



anadolum
e K a m p ü s
ve

anadolu mobil
dilediğin yerden,
dilediğin zaman,
öğrenme fırsatı!



(ekampus.anadolu.edu.tr)



(mobil.anadolu.edu.tr)

ekampus.anadolu.edu.tr



Takvim



Duyurular



Ders
Kitabı (PDF)



Epub



Html5



Video



Canlı Ders



Sesli Kitap



Ünite
Özeti



Sesli Özet



Sorularla
Öğrenelim



Alıştırma



Deneme
Sınavı



Infografik



Etkileşimli
İçerik



Bilgilendirme
Panosu



Çıkmış Sınav
Soruları



Sınav Giriş
Bilgisi



Sınav
Sonuçları



Öğrenci
Toplulukları



AOSDESTEK
AÇIKÖĞRETİM DESTEK SİSTEMİ

Açıköğretim Sistemi ile ilgili
merak ettiğiniz her şey AOS Destek Sisteminde...

- Kolay Soru Sorma ve Soru-Yanıt Takibi
- Sıkça Sorulan Sorular ve Yanıtları
- Canlı Destek (Hafta İçi Her Gün)
- Telefonla Destek

aosdestek.anadolu.edu.tr

AOS DESTEK Sistemi İletişim ve Çözüm Masası

0850 200 46 10

www.anadolu.edu.tr



/AOFAnadolum



/Anadolu_Univ



instagram.com/anadoluuniv

Bölüm 1

Temel Kavramlar

öğrenme çıktıları	1	Önemli Kavramlar	2	Bilgi İşleme Modeli ve Bilgi İşleme Süreçleri
	1	Bilgi, enformasyon ve veri kavramlarını ayırt edebilme	2	Bilgi işleme süreci ve aşamalarını örneklerle tanımlayabilme
	3	Bilgisayarların Bileşenleri	4	Bilgi İşleme ve Teknoloji
	3	Bilgisayarları oluşturan bileşenleri sıralayabilme	4	Bilgi işleme sürecinde teknolojinin oynadığı rolü açıklayabilme
	5	Sosyal Hayatta Teknoloji		
	5	Teknolojinin sosyal yaşam üzerindeki etkilerini tartışabilme		

Öğrenme çıktıları

Bölüm içinde hangi bilgi, beceri ve yeterlikleri kazanacağınızı ifade eder.

Bölüm Özeti

Bölümün kısa özetini gösterir.

Sözlük

Sözlük

Bölüm içinde geçen önemli kavramlardan oluşan sözlük ünite sonunda paylaşılır.

Karekod

Bölüm içinde verilen karekodlar, mobil cihazlarınız aracılığıyla sizi ek kaynaklara, videolara veya web adreslerine ulaştırır.

Tanım

Bölüm içinde geçen önemli kavramların tanımları verilir.

Dikkat

Konuya ilişkin önemli uyarıları gösterir.

Neler Öğrendi ve Yanıt Anahtarı
Bölüm içeriğine ilişkin 10 adet çoktan seçmeli soru ve cevapları paylaşılır.

Öğrenme Çıktısı Tablosu

Araştır/İlişkilendir/Anlat-Paylaş

İlgili konuların altında cevaplayacağınız soruları, okuyabileceğiniz ek kaynakları ve konuyla ilgili yapabileceğiniz ekstra etkinlikleri gösterir.

Yaşamlı İlişkilendir

Bölümün içeriğine uygun paylaşılan yaşama dair gerçek kesitler veya örnekleri gösterir.

Araştırmalarla İlişkilendir

Bölüm içeriği ile ilişkili araştırmaların ve bilimsel çalışmaları gösterir.

Web Tabanlı Kodlama

Editör

Doç.Dr. Yusuf Levent ŞAHİN

Yazarlar

BÖLÜM 1 Doç.Dr. Kerem KILIÇER

BÖLÜM 2, 3 Doç.Dr. Ahmet Naci ÇOKLAR

BÖLÜM 4 Doç.Dr. Serkan ŞENDAĞ

BÖLÜM 5 Dr.Öğr.Üyesi Hürşit Cem SALAR

BÖLÜM 6 Doç.Dr. Serkan ÇANKAYA

BÖLÜM 7 Dr.Öğr.Üyesi Onur DÖNMEZ

BÖLÜM 8 Doç.Dr. Cem ÇUHADAR

T.C. ANADOLU ÜNİVERSİTESİ YAYINI NO: 3518

AÇIKÖĞRETİM FAKÜLTESİ YAYINI NO: 2352

Bu kitabın basım, yayım ve satış hakları Anadolu Üniversitesine aittir.
“Uzaktan Öğretim” tekniğine uygun olarak hazırlanan bu kitabın bütün hakları saklıdır.
İlgili kuruluştan izin almadan kitabın tümü ya da bölümleri mekanik, elektronik, fotokopi, manyetik kayıt
veya başka şekillerde çoğaltılamaz, basılamaz ve dağıtılamaz.

Copyright © 2017 by Anadolu University

All rights reserved

No part of this book may be reproduced or stored in a retrieval system, or transmitted
in any form or by any means mechanical, electronic, photocopy, magnetic tape or otherwise, without
permission in writing from the University.

Grafik Tasarım ve Kapak Düzeni

Prof.Dr. Halit Turgay Ünalın

Dil ve Yazım Danışmanı

Öğr.Gör.Dr. Mustafa Ever

Ölçme Değerlendirme Sorumlusu

Emrah Emre Özkeskin

Grafiker

Gülşah Karabulut

Hilal Özcan

Ayşegül Dibek

Özlem Çayırılı

Dizgi ve Yayına Hazırlama

Kader Abpak Arul

Diğdem Aydın

Halil Kaya

Kağan Küçük

Gizem Dalmış

Zülfıye Çevir

WEB TABANLI KODLAMA

E-ISBN

978-975-06-2562-6

Bu kitabın tüm hakları Anadolu Üniversitesi'ne aittir.

ESKİŞEHİR, Ağustos 2018

3138-0-0-0-2109-V01

İçindekiler

BÖLÜM 1

Web Teknolojisinin Doğuşu ve Gelişimi



Giriş	3
Web Teknolojisinin ve Web Mimarisinin Doğuşu	3
Web Sunucularının ve Standartlarının Gelişimi	7
E-Posta Sunucuları	8
Web Sunucuları	9
DNS Sunucuları	10
FTP Sunucuları	12
Proxy Sunucuları	12
Güvenlik Sunucuları	13
Web Tarayıcılarının Gelişimi	15
Web Sayfasının Yapısı ve Temel Özellikleri	19

BÖLÜM 2

Metinlerin Biçimlendirilmesi



Giriş	29
Html'de Metin Kullanımı	29
Karakter Seti Özelliklerinin Belirlenmesi	30
Metin Özelliklerinin Düzenlenmesi	31
Paragrafların Biçimlendirilmesi	33
Listelerin Oluşturulması	34
Köprü ve Bağlantıların Kurulması	36
Metinlerde Stil Kullanımı	38
Etiketin İçerisinde Stil Kullanımı	39
Gömülü Stil Kullanımı	40
Dışarıdan CSS Dosyası Kullanma	43

BÖLÜM 3

Çokluortam Öğelerinin Sunumu



Giriş	53
Resimlerin Görüntülenmesi	53
Resim Haritalarının Oluşturulması	55
Html5'ın Getirdiği Çoklu Ortam Özellikleri	57
Html5 ile Videoların Eklenmesi	58
Html5 ile Seslerin Eklenmesi	61
Harici Uygulama ve İçeriklerin Entegrasyonu	63
Flash Animasyonlarının Eklenmesi	63
PDF Dosyalarının Eklenmesi	64
Youtube Videolarının Eklenmesi	65
Web Sitesine Konum Bilgisinin Eklenmesi	66
Çokluortam Öğelerinde Stil Kullanımı	67

BÖLÜM 4

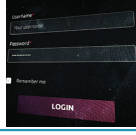
Tablo Yapılarının Oluşturulması



Giriş	81
Tabloların Tanımlanması	81
Tabloların Stil Kullanılarak Biçimlendirilmesi	84
Tablo Kenarlıklarının Oluşturulması	85
Tablolarda Veri Yerleşimi	87
Tablo Formatlarının Düzenlenmesi	88
Tabloların Bölünmesi	93

BÖLÜM 5

Form Tasarımı ve Veri İletimi



Giriş	107
Form Yapısı	107
Form Özellikleri	108
Form Elemanları	110
Input (Giriş)	112
Text (Metin Kutusu)	114
Password (Şifre)	114
Tel (Telefon Numarası)	114
Email (E-posta)	115
Date (Tarih)	115
URL (WEB Adresi)	116
Checkbox (Onay Kutusu)	116
Radio (Radyo Düğmesi)	117
Submit (Gönder Düğmesi)	117
Image (Resimli Düğme)	118
Reset (Sıfırlama Düğmesi)	118
Button (Düğme)	119
File (Dosya Yükleme)	120
Hidden (Gizli)	120
Color (Renk)	121
Range (Aralıklı Kaydırma)	122
Number (Sayı)	122
Search (Arama)	123
Button (Düğme)	123
Fieldset ve Legend	125
Select, Optgroup, Option (Açılır Liste, Gruplama Başlığı, Seçenek)	126
Textarea (Metin Alanı)	127
Datalist (Veri Listesi)	127
Label Elemanı	128

BÖLÜM 6

Bölüm ve Kanvas Yapılarının Kullanımı



Giriş	135
Div Etiketi	135
Html5 İle Sayfa Yapısı Oluşturma	137
Bölüm Özelliklerinin Belirlenmesi	139
Display Özelliği	139
Width ve Max-Width Özellikleri	140
Margin Özelliği	141
Position Özelliği	142
Overflow Özelliği	146
Float Özelliği	147
Display Özelliği İçin Inline-Block Değeri	151
Örnek Uygulama	152
Menü Uygulamaları	154
Esnek Bölüm Tasarımı	158
Viewport	159
Ortam Sorgusu	160
Grid Kullanımı	163
Resimler	165
Kanvasların Kullanımı	166

BÖLÜM 7

Önemli Web Çatıları ve Kullanımları



Giriş	175
Web Çatıları	175
Web Çatılarının Gelişimi	175
Web Çatılarının Kullanım Alanları	175
Web Çatılarının Yararları	176
Güncel Bir Web Çatısı: Bootstrap	177
Bir Bootstrap Projesinin Anatomisi	179
Bootstrap Projesi Başlatmak	180
Metin Etiketleri	182
Bootstrap Tabloları	186
Bootstrap Boyut Sistemi	188
Bootstrap Izgara Sistemi	188
Form Elementleri	192
Görsel Element Sınıfları	196
Menü Sınıfları	197
Kayan Resimler	200

BÖLÜM 8

Web Teknolojisinin Geleceği



Giriş	209
Web Teknolojisinin Başlangıcı ve Gelişimi	209
Semantik Web	211
Xml	213
Ontolojik Web	215
Bulut Bilişim	217
Mobil Uygulamalar	218

■ Önsöz

Sevgili öğrenciler,

1980'li yılların sonlarında ortaya çıkan web teknolojisi, görece kısa sayılabilecek bir süre içinde gelişip yaygınlaşarak insan yaşamında son derece önemli bir yer edinmiştir. Her gün milyarlarca insanın etkileşim kurduğu bu teknolojinin yalnızca “okuru” olmak ise, özellikle bilgi teknolojiyle iç içe yaşayan kişiler için yetersiz kalmaya başlamıştır.

Bu kitap, içeriği itibarıyla “web yazarlığı” niteliğini kazanmaya yönelik kapsamlı bir başlangıç yapmanızı amaçlamaktadır. Sözü edilen amaç doğrultusunda öncelikle web teknolojilerinin doğuşu ve gelişimi ele alınıp, bu noktaya nasıl ulaşıldığına ışık tutulmaktadır. Sonrasında ise, web içeriğinin oluşturulmasını ve sunulmasını sağlayan işaretleme dili olan HTML birçok kullanımıyla, örnekler üzerinden açıklanırken, web sayfalarının gücünü hem görsel hem de işlevsel anlamda artıran CSS'in kullanımına da yeri geldikçe değinilmektedir. Bu bağlamda, metinleri biçimlendirmek, çokluortam öğelerini sunmak, tablolar oluşturmak, veri formları tasarlamak ve bölümler oluşturmak için gerek duyacağınız bilgileri edinmeniz amaçlanmıştır.

Üretkenliğinizi artıracak düşüncesiyle, içerik sunumuna ilişkin süreçlerde hem kolaylık hem de hız sağlayan web çatısı kavramının da kitabın kapsamına alınması uygun görülmüştür. Web çatısı kavramının, sağladığı işlevlerden dolayı oldukça popülerleşmiş olan Bootstrap ile örneklendirilmiş olması, güncel web teknolojilerine ilişkin önemli bir bakış kazanmanızı sağlayacaktır.

Son olarak, web teknolojisinin geleceği olduğu söylenen semantik web ve ontolojik web kavramları üzerinde durup çeşitli kestirimlerde bulunmaya çalışan bu kitabın sizlere, istemci taraflı web teknolojilerine ilişkin kapsamlı bilgiler kazandırıp, sonraki aşama olarak değerlendirilebilecek sunucu taraflı web teknolojilerini öğrenmeniz için gereken altyapıyı sunacağı umulmaktadır.

Editör

Doç.Dr. Yusuf Levent ŞAHİN

■ Bölüm 1

Web Teknolojisinin Doğuşu ve Gelişimi

öğrenme çıktıları

1

Web Teknolojisinin ve Web Mimarisinin Doğuşu

- 1 Web teknolojinin tarihsel gelişimini açıklayabilme

2

Web Sunucularının ve Standartlarının Gelişimi

- 2 Web standartlarının gelişim sürecini açıklayabilme

3

Web Tarayıcılarının Gelişimi

- 3 Web tarayıcılarının nasıl çalıştığını açıklayabilme

4

Web Sayfasının Yapısı ve Temel Özellikleri

- 4 Web sayfalarının temel özelliklerini ifade edebilme

Anahtar Sözcükler: • İnternet • Web Tarayıcısı • Web Mimarisi • Web Standartları • Web Sayfası • WWW
• TCP/IP • HTML • HTTP • URL • W3C



http://www

GİRİŞ

Yediden yetmişe günümüz bilgi çağı insanının hayatında önemli bir yeri olan internet, 2016 yılı itibarıyla 26. yaşını kutlamaktadır. İngilizce “interconnected network” kelimelerinden türetilen internet, bir başka ifadeyle web teknolojisi, birbirine bağlı ağların oluşturduğu bir yapıyı temsil etmektedir. Başlangıçta belki ütöpik sayılabilecek bir düşünceden günümüzde kullanılan yapıya ulaşınca kadar internet farklı aşamalardan geçmiştir. Fikrî temelleri yirminci yüzyılın ortalarında atılmasına karşın internetin doğuşu ve toplumsal düzeyde yayılması yirminci yüzyılın sonlarında gerçekleşmiştir. Yaş bakımından daha genç olarak nitelendireceğimiz internet, beraberinde getirdiği yenilikler açısından günümüzde matbaanın icadından neredeyse daha önemli bir buluş olarak görülmektedir.

Günümüz bilişim toplumunda internet sadece bilgisayarların değil; cep telefonu, televizyon, buzdolabı, tost makinesi, resim çerçevesi gibi neredeyse tüm teknolojik araçların bağlandığı bir mecradır. İnternet sayesinde günümüzde neredeyse tüm işlemlerimizi (bankacılık, ticaret, iletişim, medya, kamu hizmetleri vb.) oturduğumuz yerden yapmanın keyfini sürüyoruz. Özellikle mobil internet teknolojilerinin gelişmesiyle birlikte internete daha esnek ve hızlı bir biçimde erişebiliyor ve dijital cihazlarımızı da internete bağlıyoruz. İnternete bağlı bu teknolojik cihazlar gereksinim duyduğu tüm bilgileri internetten indirerek kullanabilmekte ve kullanıcısının hayatını kolaylaştırmakta. Tam tersi olarak da kendisiyle ilgili bilgilere kullanıcıların internet üzerinden ulaşılmasına olanak vermektedir. İşte bu ve buna benzer özellikleri sayesinde artık teknolojik cihazlarımız daha da akıllanmaktadır.

Bu ünite, öncelikle web teknolojisinin ve web mimarisinin doğuşu kısaca incelenecektir. Özellikle günümüz internetinin ortaya çıkış serüveni ve bu serüvende yolculuk yapmış bilim insanları ve yapmış oldukları önemli katkılara değinilecektir. Daha sonra web standartlarının oluşumu, web standartları içerisinde yer alan sunuculardan ve türlerinden bahsedilecektir. Web teknolojilerinin gelişimiyle birlikte ortaya çıkan web tarayıcıları, çalışma mantıkları ve günümüzde yaygın kullanılan web tarayıcıları hakkında bilgi verilecektir. Son olarak ise bir web sayfasının yapısı ve web sayfasını oluşturan temel özelliklere değinilecektir.

WEB TEKNOLOJİSİNİN VE WEB MİMARİSİNİN DOĞUŞU

Hayatımızın artık vazgeçilmez bir parçası olan internetin ortaya çıkış serüveni incelendiğinde hem teknolojik hem de fikrî birçok gelişmenin bunda rol oynadığı görülmektedir. Özellikle iletişim alanındaki gelişmeler web teknolojisinin doğuşunun önünü açmıştır. Günümüz web teknolojisinin doğuşu 19. Yüzyılın başlarından itibaren ortaya konan buluşlara ve teknolojik gelişmelere dayanmaktadır. 1830 yılında elektrik akımının teller üzerinden iletimiyle bir mesajın uzaktaki başka bir noktaya taşınması, 1858 yılında Atlas Okyanusunun bir tarafından diğer tarafına iletişim amacıyla kabloların döşenmeye başlaması ve 1876 yılında telefonun icat edilmesi web teknolojisinin temelini oluşturan en önemli teknolojik gelişmelerdendir. Bu gelişmeler sonucunda ortaya çıkan merkezi telefon ağ sistemi ve bu sistemin kablolar vasıtasıyla kıtalararası genişlemesiyle ortaya çıkan iletişim ağı, günümüz internet ağ sisteminin omurgasını oluşturmaktadır.

Yirminci yüzyıla gelindiğinde, Vannevar Bush 1945 yılında yazmış olduğu “As we make think” isimli makale ile günümüz internetin çalışma prensiplerini öngörerek bir “Memex” isimli foto elektriksel temelle çalışan teorik bir makineden söz etmiştir. İnsan zihninin geliştirilmiş bir modellemesi olan bu depolama ve geri çağırma aygıtı; bir kimsenin tüm kitaplarının, kayıtlarının ve mesajlarının mikrofilm olarak depolamasına, bunların birbirleriyle ilişkilendirilmesine ve bunlar arasındaki bağlantıların olağanüstü hız ve esneklikle takip edilmesine olanak sağlamaktadır. Vannevar Bush’un öngördüğü bu ilişki yapısı bugün hipermetin (hypertext) olarak bilinen web kavramının 1965 yılında Ted Nelson tarafından ortaya konmasına katkı sağlamıştır. Web teknolojisinin gelişmesine teorik olarak katkı sağlayan bir başka isim de Amerikalı psikolog Joseph Carl Robnett Licklider’dır. 1962 yılında görev yaptığı Amerika Savunma Bakanlığına bağlı İleri Araştırma Projeleri Kurumunda (Advanced Research Projects Agency - ARPA) ortaya attığı “Intergalactic Network (İntergalaktik Ağ)” kavramı ile Licklider, dünya üzerinde bilgisayar kullanan herkesin birbiriyle bağlantılı bilgisayarlar ile bilgi alabileceği bir ağ yapısı öngörmektedir. Bu ağ yapısında uzak bilgisayar kullanan bir kullanıcı, başka bir uzak bilgisayardan veri alabilmekte veya işiyle ilgili birçok bilgisayarı kullanabilmektedir. Böyle bir ağ yapısı-

nın nasıl oluşturulacağıyla ilgili herhangi bir fikri olmamasına karşın bunun gelecek için çok önemli olduğunu öngörmüştür. Benzer şekilde Doug Engelbart bilgisayarların modern dünyanın karmaşık problemlerine çözüm olabileceğini düşünerek bilgisayarların insan aklının artırılmasında kullanılması için çalışmalar yürütmüştür. 1960 -1970 yılları arasında gerçekleştirdiği çalışmalar sonucunda NLS (oNLine System) olarak adlandırılan aynı ağ üzerinde çalışan ayrıntılı çokluortam sistemini geliştirmiştir. Bu sistem dijital kütüphanelerin ve elektronik belgelerin hipermetinler kullanarak oluşturulmasını kolaylaştırmaktadır. Bu çalışmalar Vannevar Bush'un 1945 yılında yapısını çizdiği hipermetin kavramının ilk başarılı uygulaması olmuştur. Ayrıca Engelbart bu sistem üzerinde bilgisayar etkileşimini kolaylaştırmak için ilk kez fare (mouse) kullanımını gerçekleştirmiştir. 1967 yılında Ted Nelson, Xanadu adını verdiği bir sistem tasarlamıştır. Bu sistem birbiriyle bağlantılı bilgi yığınlarının ayrı dosyalar hâlinde depolanmasına izin veren dünya çapında bir ağdan oluşmaktadır. Bu sistemde belgelere süresiz olarak erişilebilmekte ve kullanıcılar herhangi bir belgenin sanal kopyasını oluşturabilmektedir. Nelson'un ortaya koyduğu bu sistem tam olarak tamamlanamasa da, ilerleyen zamanlarda Tim Berners-Lee daha geniş ölçekli olan benzer bir yapıyı yani World Wide Web (WWW)'i oluşturmuştur.

Web teknolojisiyle ilgili teknolojik gelişmeler ağırlıklı olarak 1960'ların başlarından itibaren yaşanmaya başlamıştır. Özellikle o dönemlerde telefon ağları, bilgileri göndericiden alıcıya iletmek için devre anahtarlamayı kullanmaktadır. Buna göre ses, alıcı ve gönderici olarak belirlenen iki nokta arasında belirlenen frekans bandı üzerinden sabit bir hızla iletilmektedir. Ancak 1960'ların başından itibaren bilgisayarlara verilen önemin artmasıyla birlikte bilgisayarların birbirlerine bağlanarak coğrafi olarak farklı noktalardaki kullanıcılar arasında nasıl paylaşılacağı konusunda çalışmalar yürütülmeye başlanmıştır. Yapılan çalışmalar özellikle devre anahtarlamının bilgisayar bağlantılarındaki yetersizliklerini gidermeye yöneliktir. 1961 yılında Leonard Kleinrock devre anahtarlamının verimli ve sağlam bir alternatifi olan paket anahtarlamaya yönelik ilk çalışmaları gerçekleştirmiştir. Paket anahtarlama ileride internetin atası olarak adlandırılan ARPANET'in temel yapı taşlarından



olacaktır. Web teknolojisinin hayata geçirilmesiyle ilgili en önemli gelişme ise 1962 yılında yaşanmıştır. 1957 yılı Ekim ayında Rusların Sputnik uydusunu fırlatmasının ardından dönemin Amerika başkanı tarafından Amerika Savunma Bakanlığına bağlı İleri Araştırma Projeleri Kurumu (Advanced Research Projects Agency-ARPA) kurularak teknolojik çalışmalar başlatılmıştır. Askeri kökenli çalışmalar gerçekleştiren kurumun amacı bilimde Amerika'nın önde olduğu ve orduda teknolojinin uygulanmasına yönelik çalışmalar gerçekleştirmektir. J.C.R. Licklider'in ARPA'da görev yaptığı sırada öne sürdüğü ARPA girişimcilerinin birbirleriyle etkili bir biçimde iletişim kurabilecekleri bilgisayar ağı oluşturulması fikri, 1969 yılında Larry Roberts önderliğindeki ekip tarafından gerçekleştirilen ARPANET projesiyle hayata geçmiştir. ARPANET projesi, ARPA bünyesindeki proje girişimcilerinin bilgisayar kaynaklarını paylaşması ve benzer araştırmaların yapılmasını engellemek amacıyla bilgi paylaşımının sağlanacağı paket anahtarlama bir ağ oluşturma fikridir.

✓ ARPANET, ARPA bünyesindeki farklı bilgilerin paydaşlarla etkili bir biçimde paylaşımı için bir ağ oluşturma projesidir.

Bu ağ ile ağına bağlı tüm kullanıcılar bilgisayar kaynaklarını ve sonuçlarını birbirleriyle paylaşabileceklerdir. Arabirim mesaj işlemcileri (Interface Message Processors - IMP) olarak bilinen ilk paket anahtarları dört yerde kurulmuştur. Bunlar; Kaliforniya Üniversitesi-Los Angeles (UCLA), Stanford Araştırma Enstitüsü (SRI), Kaliforniya Üniversitesi-Santa Barbara (UCSB) ve Utah Üniversitesi'dir. ARPANET projesi kapsamında büyük ana bilgisayarlar ağı geçidi olan küçük bilgisayarlar ile birbirlerine bağlanmıştır. 1969 yılının Ağustos ayında ilk mesaj Kaliforniya Üniversitesi-Los Angeles (UCLA)'a gönderilmiştir. Bir ay sonra da Stanford Araştırma Enstitüsü (SRI)'ne ikinci mesaj gönderilerek webin atası olan ARPANET doğmuştur.

✓ Webin atası olan ARPANET, 1969 yılında hayata geçirilmiştir.

Daha sonraki yıllarda farklı iletişim kanallarını kullanan ağlar ortaya çıkmıştır. Örneğin Havaii üniversitesi'nin radyo dalgalarını kullanarak veri iletimi yaptığı ALOHANET, uyduların kullanıldığı SATNET, ticari paket anahtarlama ağı olan TELENET, Fransız paket anahtarlama ağı olan CYCLADES gibi farklı veri iletim teknolojilerini de içinde barındıran ağlar kullanılmaya başlanmıştır. Fakat oluşturulan bu ağlar veri iletimi için farklı protokoller ve standartlar kullandığı için birbirleriyle bağlantı kuramamaktadır. Bu sorunu ortadan kaldırmak için 1974 yılında Vint Cerf ve Bob Kahn ileride internet teriminin de türetileceği "A protocol for packet network interconnection" isimli bir çalışma yayınlayarak yeni bir protokol yazmışlardır. İletim Kontrol Protokolü (Transmission Control Protocol - TCP) olarak isimlendirilen bu protokol farklı ağların birbirleriyle bağlanmasının önünü açarak standart bir protokol olmuştur. 1978 yılında TCP'nin bir bölümü ayrılarak Internet protokolü (Internet Protocol - IP) olmuştur. 1982 yılında TCP/IP, internet için standart haberleşme protokolü olarak kabul edilmiştir.

1976 yılında Robert M. Metcalfe ve David R. Boggs "Ethernet: Distributed Packet Switching for Local Computer Networks" isimli bir çalışma yayınlayarak dağıtık olarak yerleşen bilgisayar sistemlerinde dijital veri paketlerini taşımak amacıyla dallanmalı yayın iletişim sistemi olan bir teknolojiyi ve protokolü ortaya koymuşlardır. Bu sistem ALOHANET'te kullanılan radyo frekanslı ağ yapısının kablo tabanlı bir yapıya dönüştürmüş şeklindedir. Bunun sonucunda kablo tabanlı paylaşımlı ağlar için yeni bir teknoloji ve günümüz web teknolojisinin vazgeçilmez aracı olan Ethernet icat edilmiştir. 1980'lere gelindiğinde ARPANET'e bağlı sistem sayısında inanılmaz büyüme yaşanmaya başlamıştır. Özellikle 1980'lerin sonunda günümüz internetine benzeyen bir ağ birleşimi olan bir yapı ortaya çıkmış ve bu yapıya bağlı ana sistem sayısı yüz bine ulaşmıştır. Bu büyümenin en önemli nedeni ARPANET'e erişimi olmayan üniversiteleri birbirine bağlamak için bilgisayar ağı oluşturmak üzere sarf edilen ayrı ayrı çabalar. Örneğin, Amerika'nın kuzeybatısındaki birçok üniversite arasında e-posta ve dosya iletimi sunan BITNET, bilgisayar bilimi araştırmacıları için kurulan CSNET, üniversite ve okulları birbirine bağlamak amacıyla ARPANET tarafından üretilen teknolojiyi kullanarak geliştirilen NSFNET bunlardan bir-

kaçıdır. 1986 yılında Amerika Ulusal Bilim Vakfı (National Science Foundation – NSF) tarafından oluşturulan NSFNET omurga hızı başlangıçta 56 Kbps iken 1988 yılında 1.5 Mbps 1991 yılında ise 45 Mbps hıza ulaşmış ve bölgesel ağları bağlamak için birincil omurga olarak kullanılmıştır. Ayrıca 1984 yılında Paul Mockapetris tarafından ortaya konulan Alan Adı Sistemi (Domain Name System – DNS) ile ağa bağlı her bir sistemin birbirlerine veri gönderip almak için kullandıkları internet protokol adresini (Internet Protocol Address – IP Address) okunabilir internet adresleriyle eşleştirilmiştir. Böylelikle günümüzde kullandığımız internet adresleri ortaya çıkmıştır.

✓ DNS(Domain Name System), web ağına bağlı her sisteme ait internet protokol adresinin kolay ve akılda kalıcı internet adresleri ile eşleştirildiği bir sistemdir.

1990'lara gelindiğinde ise internet giderek büyümeye devam etmiş ve artık ticarileşmeye başlamıştır. Ancak 1990'larda meydana gelen en büyük olay, internetin dünya çapında milyonlarca kullanıcının evine ve işyerine getiren ayrıca yüzlerce yeni uygulamanın mümkün olmasını sağlayan bir platform olan World Wide Web (WWW) uygulamasının ortaya çıkmasıdır. Bu uygulamanın ortaya çıkmasındaki başrol oyuncusu Avrupa Nükleer Araştırma Merkezi (CERN)'de yazılım mühendisi olarak çalışan Tim Berners-Lee'dir. Web'in ortaya çıkışının altında CERN'de çalışan Tim Berners-Lee'nin gördüğü temel bir problem yatmaktadır. Bu problem CERN'de çalışan araştırmacıların farklı birçok bilgisayarda bulunan verilere erişim zorluğudur. Araştırmacılar farklı bilgisayarlardaki verilere erişmek için o bilgisayarlarda oturum açmak ve hatta farklı bilgisayarlardaki farklı programları kullanmayı öğrenmek zorunda kalmaktadır. Bu sorunu gidermek için Tim Berners-Lee bir çözüm önermiştir. Zaten o dönemde bilgisayarlar birbirleriyle yüksek hızda iletişim kurarak veri transferi yapabilmektedir. Tek yapılması gereken bu bilgilerin birbirine bağlı hipermetinler aracılığıyla paylaşılmasıdır. Bunun için 1940'larda Vannevar Bush'un ve 1960'larda Ted Nelson'un öngörü ve çalışmalarını temel alarak 1989 yılında hipermetin temelli webi anlattığı "Information Management: A Proposal" isimli

çalışmayı CERN yöneticilerine sunmuştur. Tim'in sistemi internet üzerindeki verinin paylaşılmasını ve bulunmasını daha kolaylaştırmaktadır. Ancak Tim Berners-Lee'nin fikri başlangıçta kabul edilmemiştir. Bunun üzerine bireysel çalışmalar yürüten Tim, 1990 yılında günümüz webinin üç temel teknolojisini yazmıştır. Bunlar; web genelindeki bağlı kaynakların bulunup getirilmesine izin veren bir iletişim protokolü olan Hipermetin Transfer Protokolü (Hypertext Transfer Protocol – HTTP), web için bir işaretleme dili olan Hipermetin İşaretleme Dili (Hypertext Markup Language – HTML) ve webe bağlı her bir kaynağı tanımlamak için kullanılan ve daha sonraki yıllarda URL olarak adlandırılan benzersiz adres yapıları olan Benzersiz Kaynak Tanımlayıcı (Uniform Resource Identifier – URI)'dir. Ayrıca Tim, ilk web sayfası editörü (tarayıcısı) olan WorldWideWeb'i yazmış ve ilk web sunucusunu oluşturmuştur. 1990 yılının sonuna gelindiğinde Tim, ilk web sayfasını oluşturarak istemci-sunucu iletişimini gerçekleştirmiştir. 1991 yılından itibaren ise CERN dışındaki kullanıcıların da dâhil olduğu yeni bir web topluluğu oluşmaya başlamıştır.

✓ WWW yapısı CERN'de yazılım mühendisi olarak çalışan Tim Berners-Lee tarafından 1990 yılında oluşturulmuştur.

Daha sonraki yıllarda yaşanan donanımsal ve yazılımsal gelişmelerle birlikte web teknolojisi giderek büyümeye ve genişlemeye devam etmiştir. 1992'nin sonlarına gelindiğinde çalışan yaklaşık iki yüz web sunucusu bulunmaktadır ve o yıllarda Stanford Üniversitesi'nde ilk Avrupa dışı web sunucusu kurulmuştur. 1993 yılında aralarında MidasWWW, Erwise, ViolaWWW ve Samba'nın da bulunduğu birçok web tarayıcısı geliştirilmiştir. Ancak bunlardan en önemlisi Mark Andreessen tarafından Süper Bilgisayar Uygulamaları Ulusal Merkezi (National Center for Supercomputing Applications – NCSA) laboratuvarlarında geliştirilen NCSA Mosaic web tarayıcısıdır. Mosaic web tarayıcısı türünün ilk örneğidir. Tarayıcı, temel web komutlarını anlayarak GIF ve JPG uzantılı resim dosyalarını görüntülemenize daha da önemlisi dosya indirmenize izin vermektedir. Geliştirilen ilk Mosaic web tarayıcısı, grafik ekran sunabilen UNIX işletim sistemleri için oluşturulmuştur. Ardından Macintosh ve Windows sürümleri sunulmuştur. 1993 yılında CERN laboratu-

varı web protokollerini herkesin kullanımına açmış ve böylelikle webde küresel düzeyde yaratıcılık, işbirliği ve yenilik içeren gelişmeler yaşanmıştır. 1994 yılında Mark Andreessen ve Jim Clark NCSA'dan ayrılarak ileride Netscape Communication Cooperation olacak olan Mosaic Communications'ı kurmuştur. Aynı yıl Tim Berners-Lee öncülüğünde açık web standartları geliştirmek misyonuyla uluslararası bir topluluk olan W3C (World Wide Web Consortium) kurulmuştur. 1996 yılında Microsoft kendisi ile Netscape arasında, Microsoft'un birkaç yıl sonra kazanacağı tarayıcı savaşını başlatmıştır. 1995 yılında NSFNET yerini Amerika için ulusal İnternet Servis Sağlayıcısı (Internet Service Provider – ISP) ve Ağ Erişim Noktası (Network Access Point – NAP) kuruluşlarına bırakmıştır. Bu yıldan itibaren birçok özel ISP ve NAP şirketi kurulmuştur. 1996 yılında ise platform bağımsız ve web özellikleri ile genişletilmiş Java programlama dili piyasaya sürülmüştür. 1990'ların sonlarına gelindiğinde ana şirketlerin ve binlerce internet uygulamasının açılmasıyla web giderek büyümüştür. Bin yılın sonunda web; e-posta, internet, anlık mesajlaşma ve MP3 paylaşımı olmak üzere dört temel uygulamayı içine alan yüzlerce popüler uygulamayı destekler hâle gelmiştir. Gelişen mobil ve wifi ağları ile birlikte ses ve görüntü iletme, video paylaşımı, televizyon yayını gibi birçok zengin ve yüksek hızlı çokluortam uygulaması webde yerini almıştır. Ayrıca peer to peer (P2P) adı verilen, web sunucularını devre dışı bırakarak istemciler arası dosya paylaşımı, dosya dağıtımı ses ve görüntü paylaşımına olanak sağlayan ağ protokolleriyle birlikte webin potansiyeli inanılmaz artmıştır. Bu gelişmelerin yanında web sunucularına yapılan dijital saldırılar nedeniyle ağ güvenliğine yönelik günümüzde kullanılan birçok uygulama da hayat bulmuştur. Örneğin bir ağ saldırısı ile ilgili erken uyarı sağlayan saldırı tespit sistemleri ve istenmeyen ağ trafiğini ağa girmeden önce engelleyen güvenlik duvarları ağ güvenliği ile ilgili olarak geliştirilen uygulamalardandır.



Web teknolojilerinin gelişim sürecini kronolojik olarak daha ayrıntılı inceleyebilmek için <https://www.w3.org/2004/Talks/w3c10-HowItAllStarted/?n=1> adresine başvurabilirsiniz.

Başlangıçta yalnızca bilgisayarların birbiriyle bağlı olduğu bir ağ olan internet, günümüzde dijital teknolojilerdeki gelişmelerle birlikte internet ağına bağlanarak gerekli bilgiyi alan ve ona göre işlem yapan nesnelerin de dâhil olmasıyla kapsamı daha da artmıştır. IP yetenekli bu teknolojik araçlar internet ağına bağlanarak hem gerekli bilgileri sunuculardan indirerek kullanabilmekte hem de kendisiyle ilgili bilgilerin internet üzerinden kullanılmasına olanak vermektedir. Uzak sunucudan dijital fotoğrafları indirerek onları görüntüleyen bir dijital resim çerçevesi, hava durumu ile ilgili bilgileri alarak günlük hava tahminini yapılan tostun üzerine basan bir tost makinesi, trafik ve hava durumunu görüntüleyen bir web kamerası, e-posta bilgilerini alarak görüntüleyen bir cep telefonu, çeşitli bilgileri internetten alan bir çamaşır makinesi, buzdolabı veya fırın bunlardan yalnızca birkaçıdır. Yeni nesil internete bağlı nesneler binaları, köprüleri, sismik etkinlikleri, vahşi hayat alışkanlıklarını, nehir ağzlarını ve atmosferin daha düşük katmanlarını izleme olanağı sunmaktadır.



WEB SUNUCULARININ VE STANDARTLARININ GELİŞİMİ

Sunucular, 24 saat boyunca web ağına bağlı bulunan, çeşitli amaçlardaki veri dosyalarını (metin, resim, ses, video, animasyon vb.) içerisinde barındıran ve internet üzerinden bu verilere ulaşmak isteyen istemci adı verilen bilgisayarların kullanımına sunan bilgisayarlardır. Sunucular donanımsal özellikleri açısından masaüstü bilgisayarlardan daha iyidir. Sunucuların güçlü işlemci özellikleri, bellek kapasitesi ve birden çok kullanıcıya aynı anda hizmet verebilecek hızda veri aktarma özelliği bulunmaktadır. Ayrıca sunucular, kullanım amaçlarına göre üzerlerinde farklı işletim sistemleri çalıştırmaktadırlar. Genel olarak güvenlik ve maliyet açısından sunucularda Linux temelli işletim sistemleri tercih edilmektedir.

✓ Sunucu, web ağına kesintisiz bağlı olan ve 24 saat boyunca çalışan bilgisayarlardır. Kullanım amacına göre farklı türde sunucular bulunmaktadır.

Web sunucuları kendisine bağlanan birçok istemciye göre farklı işlemleri aynı anda yaptıkları için ısınma sorunu yaşamaktadırlar. Bu sorunu gidermek için sunucuların bulundukları alanda özel soğutma sistemleri çalıştırılmaktadır. Web teknolojisindeki gelişmelere paralel olarak farklı amaçlarla kullanılan sunucular bulunmaktadır. Donanımsal olarak benzer özelliklere sahip olan bu sunucular yazılımsal olarak farklı programlar ve protokoller kullanmaktadır. Bu nedenle bu bölümde sunucuların yanında ilgili iletişim protokollerinden de bahsedilecektir. Web teknolojisinin gelişimiyle paralel olarak kullanılmaya başlayan ve sırasıyla anlatılacak olan sunucular şunlardır;

- E-posta sunucuları
- Web sunucuları
- DNS sunucuları
- FTP sunucuları
- Proxy sunucuları
- Güvenlik sunucuları

E-posta Sunucuları

Elektronik posta (e-posta), web teknolojisinin başlangıcından beri var olan ve yıllar geçtikçe daha da güçlü hâle gelen günümüz internetinin en çok kullanılan uygulamalarından biridir. Sıradan posta gibi eşzamansız (asenkron) bir iletişim ortamı olan e-posta; hızlı, dağıtımı kolay ve ucuz olması nedeniyle günümüz web kullanıcılarının vazgeçilmez iletişim araçlarından bir tanesidir.

✓ E-posta, web ağı üzerinden asenkron iletişim kurmaya olanak sağlayan bir uygulamadır.

Başlangıçta yalnız metin tabanlı olan e-postalar, yaşanan gelişmelerle birlikte artık eklentiler, bağlantılar, HTML ile biçimlendirilmiş metinler veya fotoğraflar içermektedir. Web teknolojisinin tarihsel serüveninde de bahsedildiği gibi 1969 yılında webin atası olarak nitelendirilen ARPANET projesi, projeyi yürüten İleri Araştırma Projeleri Kurumu (ARPA) ile dört farklı yerde bulunan arabirim mesaj işlemcileri (IMP) arasında mesaj gönderimi ile hayata geçmiştir. O yıllarda oluşturulan arabirim mesaj işlemcileri, günümüzde ağ yönlendiricisi (router) olarak adlandırılmaktadır ve iki farklı noktadaki bilgisayarların birbirleriyle mesajlaşmasına olanak sağlamaktadır. Ancak bu mesajlaşmada aynı bilgisayar kullanıcıları arasında eşzamanlı (senkron) bir iletişim söz konusudur. Bu durumu ortadan kaldıracak yani ağ içerisindeki herhangi bir başka bilgisayara gönderilecek mesajın eşzamansız (asenkron) olarak iletilmesini ve alınmasını sağlayacak bir sistemin geliştirilmesi 1971 yılında gerçekleştirilmiştir. Bolt, Beranek and Newman (BBN) ileri teknoloji şirketinde çalışan Raymond Samuel Tomlinson, ARPANET projesini kullanan bilgisayarlar arasında iletişimi sağlayacak bir sistem geliştirmiştir. SNDMSG uygulaması ile CYPNET aktarım protokolünün birleştirilmesiyle oluşturulan bu sistemde ağ içerisindeki herhangi bir bilgisayara mesaj gönderilebilmektedir. Postaların saklandığı posta kutuları, eklenecek her bir dosyanın posta kutusunun sonuna eklenmesi ve bu eklemenin var olan dosyanın üzerine yazılmaması gibi ilkeler Tomlinson'un geliştirdiği sistem üzerindeki yazılımın temel ilkeleridir. Ayrıca gönderilen e-postanın hangi bilgisayara ve hangi kullanıcıya iletileceğini

belirtmek için @ işaretinin konulması yine Tomlinson tarafından hayat bulmuştur. Böylelikle ilerleyen yıllarda web teknolojisinin içerisinde büyük bir bölümünü oluşturan e-posta sistemi hayata geçmiştir.

✓ @ işaretinin e-posta adreslerinde ilk olarak Ray Tomlinson tarafından kullanılmıştır.

Doksanlı yıllarda web teknolojisinin yaygınlaşmasıyla birlikte e-postanın web üzerindeki popülerliği daha da artmıştır. Başlangıçta sınırlı sayıda ve sistem üzerinde çalışan e-posta programları, webin yaygınlaşmasıyla birlikte ücretsiz ve ulaşılabilir olmaya başlamıştır. Özellikle web tarayıcılarının gelişmesi ve teknoloji şirketlerinin artan yenilik üretimi, e-postanın internet kullanıcıları için haberleşme amaçlı kullanılan uygulamaların başında yer almasını sağlamıştır. Günümüzde video konferans veya sohbet programlarının yanı sıra sosyal ağ uygulamalarının yaygınlaşmasıyla birlikte e-posta, daha resmi iletişim kanalı olarak hizmet vermektedir. Peki, e-posta göndermek veya almak için nasıl bir altyapı kullanılıyor, isterseniz kısaca bu yolculuktan da biraz bahsedelim. E-posta sistemleri günümüzde ayrı sunucular üzerinden işlem görmektedir. Bunun nedeni, bu sunucular üzerinden çok büyük hesapları hızlı bir biçimde yönetebilmektir. Kullanıcılar, e-posta yazılımlarını (Outlook, Thunderbird, Mail vb.) kullanarak veya e-posta sunucularının web arayüzü üzerinden hazırladıkları e-postaları başka bir kullanıcıya göndermek için öncelikle o kişinin e-posta adresini alıcı kısmına yazması gerekmektedir. E-posta adresleri iki bölümden oluşmaktadır ve her iki bölüm arasında @ işareti bulunmaktadır (Örneğin, webtabanlikodlama@anadolu.edu.tr). Birinci bölüm (sol taraf) alıcının e-posta kutusunun adı, ikinci bölüm (sağ taraf) kullanıcının posta kutusunun konumlandığı e-posta sunucusunun adıdır. Bir kullanıcı tarafından hazırlanan e-postada alıcı adresi, konu, mesaj metni ve ekleri oluşturulduktan sonra gönder düğmesine basıldığında ilk olarak e-posta, kullanılan yazılım tarafından göndericinin e-posta sunucusunun kuyruğuna yerleştirir. Gönderici e-posta sunucusunda bulunan Basit Posta İletim Protokolü (Simple Mail Transfer Protocol – SMTP)'nün istemci tarafı alıcının e-posta adresine göre mesajın iletileceği alıcı e-posta sunucusuyla güvenilir ve doğrudan bir bağlantı kurar.

✓ SMTP (Simple Mail Transfer Protocol), e-posta iletim için kullanılan temel protokoldür.

Daha öncede bahsedildiği gibi alıcının e-posta sunucusu, alıcı e-posta adresindeki @ işaretinden sonraki kısımda belirtilmektedir. Eğer gönderici e-posta sunucusu alıcı e-posta sunucusuyla bağlantı kuramıyorsa, belirli aralıklarla bağlantı kurmayı yeniden dener. Kurulan bu bağlantı web teknolojisinin altyapısını oluşturan TCP protokolü ile gerçekleştirilmektedir. Gönderici ve alıcı e-posta sunucuları arasında oluşturulan TCP bağlantısından sonra gönderici e-posta sunucusu kuyruğunda bulunan e-postayı alıcı e-posta sunucusuna gönderir. Bu gönderme işleminden önce kurulan bağlantıda alıcının posta kutusunun alıcı e-posta sunucusunda olup olmadığı doğrulanır. Eğer alıcı posta kutusu kurulan bağlantıda doğrulanmaz ise gönderilmek istenen e-posta gönderici posta kutusuna geri iletilir. Sorunsuz bir bağlantı kurulduğunda, alıcı e-posta sunucusundaki SMTP'nin sunucu tarafı e-postayı alır ve alıcının posta kutusuna yerleştirir. Alıcı, posta kutusuna düşen e-postayı yine e-posta yazılımlarını veya e-posta sunucularının web arayüzünü kullanarak açabilir.

Web Sunucuları

Web sunucuları, hipermetin temelli World Wide Web (WWW) yapısını oluşturan her türlü dosyanın (metin, resim, video, ses, animasyon vb.) içerisinde yer aldığı güçlü bilgisayarlardır. Web sunucuları, web tarayıcılarından gelen isteklere göre ilgili web sayfasını oluşturan her türlü dosyayı istemci bilgisayara göndermektedir. Web sunucularının ortaya çıkışı, CERN'de yazılım mühendisi olarak çalışan Tim Berners-Lee tarafından geliştirilen WWW uygulamasına dayanmaktadır. Tim Berners-Lee'nin gerçekleştirdiği çalışmalar sonucunda hipermetin temelli web sayfaları oluşturularak sunucu adı verilen bilgisayara kaydedilen bu sayfaların başka bir bilgisayarda görüntülenmesi sağlanmıştır. Bunun için HTML işaretleme dili, URL adı verilen benzersiz web adresleme yapısı ve bilgisayarlar arası bağlantı için kullanılan HTTP iletim protokolü oluşturulmuştur. Oluşturulan bu yapı içerisinde; hipermetin temelli web sayfalarının yerleştirildiği ve web ağına kesintisiz bağlı olan

web sunucuları bulunmaktadır. İlk olarak 1990 yılında oluşturulan web sunucuları, o zamana kadar kullanılan web teknolojisinin yapısını oldukça değiştirmiştir. Doksanlı yılların başına kadar web teknolojisi öncelikle araştırmacılar, akademisyenler ve üniversite öğrencileri tarafından uzak ana sistemlerde oturum açmak, dosya transfer etmek, haber almak ve göndermek ve elektronik posta almak ve göndermek için kullanılmaktadır. Ancak 1990 yılında temelleri atılan WWW yapısının oluşturulmasıyla web teknolojisi herkes için erişilebilir olmuş ve birçok veri ağından tek bir veri ağına dönüşmüştür. Bu dönüşümün bir parçası da web sunucularıdır. Günümüzde web sunucuları; web içerisinde bulunan bir web sayfasına ait tüm verileri içerisinde barındıran, güçlü donanımsal özellikleri bulunan ve kesintisiz olarak web ağına bağlı bulunan bilgisayarlardır. Web sunucuları, bir internet kullanıcısının web tarayıcısında görüntülemek istediği web sayfasına ait tüm verileri internet kullanıcısının bilgisayarına iletmektedir.

✓ Web sunucuları, WWW yapısını oluşturan tüm hipermetin temelli web sayfa-larına ait dosyaların içerisinde yer alan ve kesintisiz olarak web ağına bağlı bulunan bilgisayarlardır.

Web sunucularının WWW yapısı içerisindeki temel görevi, hosting (barındırma) hizmeti vermektir. Bunun anlamı, internette yayınlanması istenen bir web sitesine ait tüm dosyaların (HTML sayfası, resim, video, ses, animasyon vb.) internet kullanıcıları tarafından erişilebilecek bir sunucuya yerleştirilmesi gerekmektedir. Web sunucuları, sahip oldukları donanımsal özelliklere bağlı olarak birçok siteyi içerisinde barındırabilmektedir. Web tasarımcıları, hazırlamış oldukları web sayfalarını web sunucularından hosting hizmeti almak (satın alarak veya kiralarak) suretiyle internette yayınlatabilmektedir. Bunun için hazırlamış oldukları web sitesine ait tüm dosyaları hosting hizmeti aldığı web sunucusunun içerisine yerleştirmesi yeterlidir. Web sitesinde yapılacak güncellemelerin de bu sunucu içerisindeki dosyalar üzerinde yapılması gerekmektedir. Günümüzde ücretli bir hizmet olan hosting, bazı siteler tarafından ücretsiz olarak da verilmektedir. Ancak ücretsiz olarak verilen hosting hizmetinde istenilen bir alan adının alınamaması,

sınırlı kapasitede veri depolanabilmesi, veritabanı desteği olmaması, dinamik web sitesi oluşturulmaması veya web sitesini kullanırken istenmeyen reklamların görüntülenmesi gibi çeşitli olumsuzlukların yaşanabileceği unutulmamalıdır.

✓ Hosting hizmeti, web sitelerinin web ağı üzerinden erişilebilmesi amacıyla web sunucuları tarafından verilen barındırma hizmetidir.

Web sunucularının istemci bilgisayarlarla iletişim kurmak için kullandığı temel protokol HTTP protokolüdür. Bu protokol 1990 yılında Tim Berners-Lee tarafından oluşturulan WWW yapısının temel iletişim protokolüdür. HTTP protokolü, WWW yapısı içerisindeki web sunucuları ile web sayfalarını görüntüleyecek olan istemci bilgisayarlar arasındaki iletişimin nasıl gerçekleştirileceğini tanımlamaktadır. HTTP iki bilgisayar arasındaki bağlantı esnasında kurulan iletişimdeki mesajların yapısını ve istemci ile sunucunun mesaj alışverişini nasıl yapacağını tanımlamaktadır. Web sunucuları ile web tarayıcıları arasındaki iletişim ve veri transferi detaylı olarak “Web Tarayıcılarının Gelişimi” bölümünde anlatılmıştır. Web sunucuları ile istemci bilgisayarlar arasında iki farklı şekilde bağlantı kurulabilmektedir. Bunlar kalıcı ve kalıcı olmayan bağlantılardır. Kalıcı bağlantıda istemci ile sunucu arasında uzun bir zaman periyodu boyunca iletişim gerçekleşmektedir. TCP üzerinden gerçekleşen kalıcı bağlantıda istemci ve sunucu arasındaki tüm istekler ve ilgili cevaplar aynı TCP bağlantısı üzerinden gönderilmektedir. Kalıcı olmayan bağlantıda ise, istemcinin bir seri istekte bulunması ve sunucunun her isteğe cevap vermesi gerekmektedir. Kalıcı olmayan bağlantı başka ifadeyle istemci sunucu arasında gerçekleştirilecek her istek ve cevap için yeni bir TCP bağlantısı oluşturulmaktadır. Bu durum, eşzamanlı olarak yapılacak çok sayıda istemciye cevap veren sunucuya önemli ölçüde yük bindirebilmektedir. Bu nedenle HTTP’nin varsayılan durumunda kalıcı bağlantılar kullanılmaktadır.

✓ HTTP (Hypertext Transfer Protocol), web ağı üzerindeki iki bilgisayarın WWW yapısına göre birbirleriyle iletişim kurmak için kullandığı temel protokoldür.

DNS Sunucuları

Alan adı sistemi olarak adlandırılan DNS (Domain Name System), web ağına bağlı her bir birimin birbirlerine veri göndermek ve almak için kullandıkları internet protokol adresini (IP) okunabilir internet adresleriyle eşleştirilmesi için kullanılan bir sistemdir. Paul Mockapetris tarafından 1984 yılında ortaya konulan bu sistem ile web tarayıcısına yazılan bir internet adresindeki alan adını barındıran sunucu bilgisayarın IP adresi çözümlenerek ilgili web sunucusuna bağlanılabilmektedir. DNS sunucuları ile internetteki alan adlarının IP adreslerine çevrim işlemi yapılmaktadır. Bu çevrim işleminin yapılmasındaki en büyük neden IP adreslerinin hatırlanmasının zor olmasına karşın alan adlarının hatırlanmasının kolay olması ve IP adreslerinin alan adlarına göre daha çok bilgi içirmesi ve sunucu bilgisayarın web ağının içerisinde nerede olduğunun kolaylıkla bulunabilmesidir.



✓ IP (Internet Protocol) adresi, web ağına bağlı olan her bir sisteme verilen ve o sistemi tanımlayan benzersiz numaralardır.

DNS sunucularının çalışma prensiplerine geçmeden önce IP adresleriyle ilgili birkaç bilgi vermekte yarar vardır. IP adresi web ağına bağlı olan her bir sisteme erişim için kullanılan benzersiz numaralardır. IP adresleri bilgisayarların TC kimlik numarası olarak düşünülebilir. Web ağına erişimi olan her bir cihazın mutlaka IP adresi olması gerekir. Günümüzde kullanılan IP adresleri IPv4 olarak adlandırılan yapıya göre tanımlanmaktadır. Bu yapıya göre adres uzunluğu 32 bit olan 2^{32} adet IP adresi bulunmaktadır. Bir başka ifadeyle yaklaşık olarak 4 milyar adet IP adresi vardır. Ancak web ağına bağlanan sistem sayısındaki artış göz önünde bulundurularak IPv6 olarak adlandırılan yeni bir yapıya geçiş yapılmaktadır. Bu yapıda ise adres uzunluğu 128 bit olan 2^{128} adet IP adresi tanımlanabilmektedir. Hiyerarşik bir yapıya sahip olan IP adresleri; dört byte’tan oluşan ve her bir byte 0 ile 255 arasında ondalık değer ile temsil edilen dört rakam grubundan oluşmaktadır. Ayrıca her sayısal değer birbiriyle nokta (.) ayracı ile ayrılmaktadır (Örneğin, 192.168.170.40). Bu adresler hem soldan sağa doğru okunduğunda hem de yukarıdan

aşağıya doğru okunduğunda birçok bilgi vermektedir. IP adresi soldan sağa doğru tarandığında bu adresin birbirine bağlı olan birçok ağ içerisinde hangi ağda olduğuna dair bilgi edinilebilmektedir. Benzer şekilde aşağıdan yukarıya doğru tarandığında adresin nerede konumlandığı ile ilgili bilgiler edinilebilmektedir. Bir başka ifadeyle IP adresine bakarak bir kullanıcının bulunduğu ülkeyi, kenti, enlem ve boylamı ve internet servis sağlayıcısını belirlemek mümkündür. Web ağında kullanılabilecek IP adresleri dünya genelinde merkezi bir kurum olan Internet Corporation for Assigned Names and Numbers – ICANN tarafından yapılmaktadır. Bu kurulun alt birimi olan Internet Assigned Numbers Authority (IANA), web ağına bağlanan tüm sistemlere ait alan adları ve IP adresleri gibi benzersiz tanımlayıcılara ait standartların oluşturulmasında görev almaktadır. Bu kurul tarafından belirlenen ve kendi altyapıları üzerinden internet ağına erişim olanağı sağlayan ticari firmalar olan internet servis sağlayıcıları ise IP adreslerini web kullanıcılarına tanımlamaktadır. Ülkemizde yetkili kurum Orta Doğu Teknik Üniversitesi (ODTÜ)'dür. Belirlenen IP adresinin başlangıç değeri hangi sınıfta (A, B, C, D, E) yer aldığını göstermektedir. Farklı sınıfların içerisinde farklı sayıda IP adresi bulunmaktadır. A sınıfı IP adresleri 1 ile 126 arasında octet değeri ile başlamakta ve teorik olarak maksimum 128 ağ ve bir ağda maksimum 16777216 bilgisayar olabilmektedir. B sınıfı IP adresleri 128 ile 191 arasında octet değeri ile başlamakta ve teorik olarak maksimum 16384 ağ ve bir ağda maksimum 65536 bilgisayar olabilmektedir. C sınıfı IP adresleri ise 192 ile 223 octet değeri ile başlamakta ve teorik olarak maksimum 2097152 ağ ve bir ağda maksimum 256 bilgisayar olabilmektedir. A sınıfı bir IP adresinin ilk kısmı, B sınıfı IP adresinin ilk iki kısmı, C sınıfı IP adresinin ilk üç kısmı ağ adresi, sonraki kısımlar/kısım ise bilgisayar adresi için kullanılmaktadır. Bir IP adresinin ait olduğu ağı temsil eden adresi bulmak için bilgisayar adresi kısmındaki tüm bitlerin 0 olarak tanımlanması gerekir. Örneğin, 193.140.180.2 IP adresinin sınıfı C sınıfında bir adrestir ve ağ adresi 193.140.180.0'dır.

✓ İnternet servis sağlayıcıları (ISS), kendi altyapıları üzerinden internet ağına erişim olanağı sağlayan ticari firmalardır.

DNS sunucularının web teknolojisi içerisindeki rolü incelendiğinde, DNS sunucularının web sunucularıyla birlikte çalıştığı görülmektedir. Bir başka ifadeyle her alan adının, hem web sunucusu hem de DNS sunucusu bulunmaktadır. Her iki sunucunun işleyişini bir örnek üzerinden anlatmak daha açıklayıcı olacaktır. Gaziosmanpaşa Üniversitesi'ndeki bir kullanıcının web tarayıcısı yardımıyla Anadolu Üniversitesi'nin alan adındaki (anadolu.edu.tr) bir web adresine erişmek istediğini düşünelim. Öncelikle Gaziosmanpaşa Üniversitesi'nin ana sistemi kendi yerel DNS sunucusuna bağlanmak istediği alan adıyla ilgili bir DNS sorgu mesajı gönderir. Sorgu mesajı, ulaşmak istenen ana sistemin alan adını içermektedir. Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusu, sorguyu hiyerarşik olarak yapının en üstünde bulunun kök DNS sunucusuna iletir. Kök DNS sunucusu bağlanmak istenen alan adının tipine (edu) ve ülke koduna (tr) göre Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusuna edu etki alanından ve tr ülke kodundan sorumlu üst seviye etki alanlarına ait IP adreslerinin listesini gönderir. Ardından Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusu, üst seviye etki alanı sunucularından birine erişmek istediği ana sistemin alan adına ait sorgu mesajını tekrar gönderir. Bu sefer, üst seviye etki alanı sunucusu Anadolu Üniversitesi'nin yetkili DNS sunucusunun IP adresini Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusuna gönderir. Son olarak Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusu sorgu mesajını doğrudan Anadolu Üniversitesi'nin IP adresini gönderecek olan ana sistemin DNS sunucusuna gönderir. Anadolu Üniversitesi'nin ana DNS sunucusu ana sistem IP adresini Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusuna gönderir. Gaziosmanpaşa Üniversitesi'nin yerel DNS sunucusu almış olduğu IP adresini web sunucusuna ileterek iki sunucu arasında HTTP bağlantısı oluşturulmasını sağlar. Oluşturulan bağlantı ve veri aktarımı sonucunda Anadolu Üniversitesi alan adı içerisindeki bir web adresi Gaziosmanpaşa Üniversitesi alan adı içerisindeki bir kullanıcının web tarayıcısında görüntülenir. Örnekte de görüldüğü gibi DNS sunucuları web sunucularının başka bilgisayarlar tarafından kolay ve rahat bir biçimde ulaşılmasında önemli bir rol üstlenmektedir.

FTP Sunucuları

Dosya transfer protokolü (File Transfer Protocol – FTP), web teknolojileri üzerinden kendine erişen istemci ile sunucu arasında dosya paylaşımını ifade etmektedir. HTTP iletim protokolünden daha eski olan FTP protokolü, 1971 yılında ARPANET projesi kapsamında yapılan çalışmalar sonucunda Abhay Bhushan tarafından yazılmıştır. FTP protokolü TCP/IP mimarisinin gelişmesine önemli katkılar sağlamış ve oluşturduğu yapı e-posta protokolünün ilk versiyonu olarak kullanılmıştır. Bhushan tarafından yazılına FTP protokolü, ARPANET protokolüne göre çalışmak üzere tasarladığı için sonraki yıllarda değişiklikler yapılarak TCP/IP protokolüne uygun hâle getirilmiştir. FTP sunucusu ise, FTP protokolü üzerinden kendisiyle iletişim kuran cihazlarla veri aktarımı ve dosya paylaşımı yapan bilgisayarlardır. FTP protokolü ile FTP sunucusuna bağlantı kurulabilmesi için ağ sistem yöneticisinin verdiği kullanıcı adı ve şifre girilmesi gerekmektedir. Ancak herkesin kullanımına açık olan yani her kullanıcının girmesine izin verilen FTP sunucuları da bulunmaktadır. FTP sunucuları üzerindeki dosya yapısı bilgisayarlardaki dosya yapısına benzerdir. Yani FTP sunucuları üzerinde klasör ve dosyalar bulunmaktadır. Klasörlerden alt klasörlere erişilebilmektedir. Dosyalar tıklandığında ise ilgili dosya istemci bilgisayara indirilmektedir.

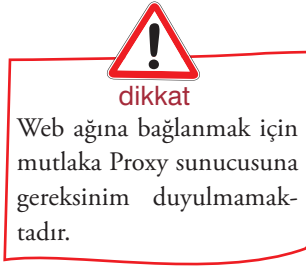
✓ FTP (File Transfer Protocol), web ağı üzerindeki bilgisayarların dosya paylaşımı için kullanılan protokoldür.

FTP sunucularına değişik işletim sistemleri çalıştıran istemci bilgisayarlar aracılığıyla bağlanılabilmektedir. Bir başka ifadeyle bir web kullanıcısı istediği herhangi bir bilgisayardan FTP sunucusuna erişebilmektedir. FTP sunucularına hem web tarayıcıları aracılığıyla hem de FTP yazılımları aracılığıyla bağlanılabilmektedir. “Web Tarayıcılarının Gelişimi” bölümünde de bahsedildiği gibi bir web tarayıcısının adres çubuğuna “ftp://anadolu.edu.tr” yazılarak bu sayfa açılmak istendiğinde, istemci bilgisayar FTP protokolünü kullanarak Anadolu Üniversitesi’nin FTP sunucusuyla iletişim kuracaktır. Kurulan iletişim sonucunda açılan sayfada klasörler ve dosyalar görüntülenecek ve istenilen dosya web sayfası üzerinden istemci bilgisayara kaydedilebilecektir. Tam tersi olarak da FTP sunucusuyla bağlantı kuran bir istemci bilgisayar sunucuya kullanıcı adı ve şifre ile bağlanarak yetkilendirilmesine bağlı olarak sunucuya veri yükleyebilmekte veya sunucuda bulunan verileri alabilmektedir. FTP sunucularına bağlanmak için web tarayıcıları kullanılabildiği gibi FTP sunucularına erişmek için tasarlanmış özel yazılımlarda (FileZilla, Cute FTP, SmartFTP) kullanılabilmektedir. Bu yazılımlar yardımıyla bağlanılacak FTP sunucusunun adresi, kullanıcı adı ve şifre bilgileri girilerek bağlantı oluşturulmakta ve istenilen dosyalar istemciden sunucuya veya sunucudan istemciye transfer edilebilmektedir.

Proxy Sunucuları

Proxy sunucuları, birden fazla kullanıcının aynı anda internet bağlantısı kullandığı yerlerde kullanılan sunuculardır. Temel görevi, internet trafiğinin daha verimli kullanılabilmesi ve aynı sayfayı görüntülemek isteyen kullanıcıların hızlı bir biçimde web sayfasına erişimini sağlamasıdır. Proxy sunucuları, kullanıcılar tarafından girilen web sayfalarının bir kopyası alınarak kendi sabit diskinde yedeklemektedir. Daha sonra eğer başka bir kullanıcı aynı web sayfasına ulaşmak isterse ulaşmak istenen web sunucusuyla tekrar bağlantı kurup ilgili web sayfasına ait verileri almak yerine, sabit diskinde yedeklediği web sayfasını kullanıcıya göndermektedir. Böylelikle aynı sayfaları web ağı üzerinden tekrar tekrar yüklemeye gerek kalmamaktadır. Ayrıca yerel ağ bağlantısı daha hızlı olduğu için yerel ağda bulunan Proxy sunucusunda yedeklenen web sayfasına ait verilere daha çabuk ulaşılabilir. Web ağına bağlantı için mutlaka Proxy sunucularına gereksinim duyulmamaktadır. Ancak çok sayıda kullanıcının aynı anda web ağına bağlandığı durumlarda ve özellikle uluslararası internet bağlantılarındaki yoğunluğu azaltmak, erişimleri hızlandırmak ve ağı daha etkin kullanmak için oldukça yararlıdır.

✓ Proxy sunucunun temel görevi, internet trafiğinin daha verimli kullanılabilmesi ve aynı sayfayı görüntülemek isteyen kullanıcıların hızlı bir biçimde web sayfasına erişimini sağlamasıdır.



Proxy sunucuları yukarıda bahsedilen yedekleme dışındaki bazı amaçlar için de kullanılabilir. Proxy sunucuları;

- Sistem kullanıcılarının hangi web sayfaları görüntülediği izlenebilir,
- Bazı web sitelerine yapılan erişim engellenebilir,
- Sistem kullanıcılarının her birine internet bağlantı kapasitesinin belirli bir oranı ayrılabilir,
- Bilgisayar virüsü içeren yazılımların internetten indirilmesi engellenebilir.

Güvenlik Sunucuları

Web ağına bağlı olan sistem sayısı arttıkça ve web ağıyla yapılabilecek işlemler yelpazesi genişledikçe web ağına daha güvenli bağlantıların gerçekleştirilmesi, web teknolojileri kapsamındaki önemli konulardan bir tanesi olmuştur. Özellikle web ağına bağlı bir bilgisayarın başka bir bilgisayara veya sunucuya rahatlıkla erişmesi, kullanıcı adı ve şifre açarak oturum açması o bilgisayar veya sunucu üzerinde istediği işlemleri yapabilmesinin önünü açmaktadır. Bu durum bilgisayar korsanı (hacker) olarak adlandırılan kişilerin, eriştiği bilgisayardaki bilgileri görüntülemesi, kopyalaması, silmesi veya değiştirmesi gibi birçok istenmeyen işlemlerin yapılmasına neden olmaktadır. Bu nedenle günümüzde web ağındaki başka bir bilgisayara yetkisiz erişimi engellemek adına çeşitli önlemler alınmaktadır. Alınan bu önlemler güvenlik yazılımları gibi

yazılımsal boyutta olabileceği gibi güvenlik sunucuları gibi donanımsal boyutta da olabilmektedir. Güvenlik sunucuları, web ağında yaşanabilecek güvenlik sorunlarını ortadan kaldırmak için kullanılan sunuculardır. İçerisinde güvenlik yazılımlarını da barındıran güvenlik sunucuları, yerel ağ ile web ağı arasında yer alarak sisteme yetkisiz erişimi, ağ saldırılarını belirleme ve engelleme amacıyla kullanılmaktadır. Web ağı üzerinden yerel sisteme giren veri paketleri veya yerel ağdan web ağına çıkan veri paketleri güvenlik sunucuları tarafından denetlenmektedir. Eğer güvenlik sunucuları tarafından şüpheli bir erişim belirlendiğinde veri paketinin gideceği yere ulaşması engellenmektedir.

✓ Güvenlik sunucuları, web ağında yaşanabilecek güvenlik sorunlarını ortadan kaldırmak için kullanılan ve içerisinde güvenlik yazılımlarını barındıran sunuculardır.

Ancak güvenlik sunucularının web ağından gelen ve web ağına çıkan her veri paketini incelemesi beraberinde bazı sorunlara neden olmaktadır. Bu sorunların en başında yaşanan zaman kaybı yer almaktadır. Güvenlik sunucusundaki yazılımın web ağını sürekli bir şekilde denetlemesi, kullanıcıların web ağındaki kaynaklara erişimlerinde hissedilir yavaşlamaya neden olmaktadır. Ayrıca kullanıcılar tarafından web ağında erişilmek istenen sayfaların denetlenmesi, izlenme fikrinin oluşmasına neden olabilmektedir. Bu nedenle web ağından gelebilecek saldırıları tespit edecek saldırı tespit sistemleri kullanılarak web ağına erişim hızındaki yavaşlama giderilebilmektedir. Saldırı tespit sistemleri, geçen veri paketlerini inceleyerek sakıncalı bir durum olup olmadığı denetlenmektedir. Sakıncalı bir durumun tespit edilmesi hâlinde buna uygun olarak güvenlik sistemleri devreye girmektedir.



yaşamla ilişkilendir

İnternete bağlı cihaz sayısı ve türü günden güne artıyor

İnternetin yaygınlığının artması ve kablosuz teknolojinin gelişmesi mobilitenin hayatımızdaki yerinin artmasına sebep olurken mobil internet teknolojisi ve akıllı telefonlar, çevrelerinde çok farklı ve ucu açık bir segmentin oluşmasına sebep oldu. Tabii bunların yanına gelişen sensör teknolojilerini de eklemek gerek. Basit bir akıllı telefonun içerisinde bile onca devre, ekran, kamera, batarya gibi bileşenler bulunuyorken aynı zamanda onlarca sensör bulunuyor. Bundan birkaç yıl önce tek başına cebimize sığamayacak boyutlardaki sensörler cep telefonumuzun içinde yer alıyor.

İştah açıcı mobil pazar için hazırlanan ve hızla geliştirilen teknolojiler, bu teknolojilerin farklı alanlarda da kendine yer bulmasını sağlıyor. Mikroçiplerin, sensörlerin ve radyo bağlantılarının gelişmesiyle beraber akıllı telefonlardan çok daha küçük, çok daha “hedefe yönelik” ürünlerin ortaya çıkmasına sebep oldu. Bu “küçük ve hedefe yönelik” cihazlar tek başlarına kullanılmak üzere tasarlanabildikleri gibi (3G destekli akıllı saatler), gücünü akıllı telefondan veya başka bir akıllı cihazdan da alıyor olabilirler (adım sayar bileklikler). Ayrıca hali hazırda bulunan bir ürün tipinin içerisine yerleştirilerek, aynı zamanda farklı cihazlarla iletişim kurması da sağlanabiliyor.

Akıllı Şehir Yaklaşımları

Akıllı şehirlerin oluşmasında büyük rol oynayan sensörler, birer “her şeyin interneti” cihazı olarak karşımıza çıkabiliyorlar. Örneğin bir akarsuyun kenarına yerleştirilen cihaz ile o noktadaki nem oranını, rüzgarı ve daha birçok bilgiyi elde edebiliyoruz. GSM şebekesi, kablo bağlantısı veya WiFi gibi bağlantılarla verileri ana merkeze gönderen bu sensör cihazları, gelecekte kendi elektriğini üretecek seviyeye de geleceklerdir.

Çevresel olayların bölge bölge, anlık olarak takip edilebiliyor olması, yerleşim yeri hakkında verilecek kararlar hakkında büyük yol gösterici

konumundalar. İnternete, dolayısıyla verilerin toplandığı ve irdelendiği merkeze bağlı olarak çalışan bu tip cihazların kullanımı günden güne artıyor.

Giyilebilir Teknolojiler

2015, giyilebilir teknolojilerin bol bol konuşulduğu ve bazı ürün tiplerinin yavaş yavaş şekillendiği bir yıl oldu. Elbette ki bunlardan en öne çıkanı akıllı saatler oldu. Artık hemen her akıllı telefon üreticisi, bir de akıllı saat üzerine çalışıyor. Akıllı saatler her serisinde büyük yeniliklerle donatılırken, arayüzleri de sürekli olarak şekilleniyor. 2015’in sonlarına doğru piyasaya sürülen Samsung Gear S2 ve Apple Watch, bu cihazların artık kullanılabilir hale geldiklerini gösteriyor. Akıllı saat modellerinin bir kısmı kendi içerisindeki sim kart yuvasıyla tamamen bağımsız bir yapıda çalışabiliyorken, kimisi de Bluetooth üzerinden bir akıllı telefonu “veri merkezi” olarak kullanıyor; akıllı telefon aksesuarı olarak çalışıyor. Akıllı saatler, sağlıkla ilgili çeşitli verilerin sağlanmasının yanı sıra indirilebilen uygulamalarıyla daha farklı işlevler de kazanabiliyorlar. Henüz uygulama alanında dikkat çeken çok fazla bir şey olmasa da 2016 yılında uygulama anlamında büyük yeniliklerle karşılaşacağız. Mobil uygulama geliştiricileri, uygulamalarının akıllı saatlerdeki yansımalarının daha kullanışlı olması için ellerinden geleni yapacaklar.

Yeni Nesil Akıllı Binalar

Gerek evlerin, gerekse işyerinin / işyerlerinin bulunduğu binalarda kullanılan akıllı teknolojiler, binanın yönetimini kolaylaştırırken buralardaki yaşam kalitesini artırıyor. Bina içerisinde tüm mekanik, elektrik ve elektronik aksamalarının bir merkez tarafından kontrol edilip yönetilebiliyor olmasının yanı sıra, her an her noktada bulunan kameralar da binanın güvenliğinin kolay bir şekilde kontrol altına alınmasını sağlıyor. Sıcaklıkların, binanın ayrı noktalarında otomatik olarak ayarlanabildiği akıllı bina sistemlerinde sensörler ve bu sensörlerin bağlı olduğu merkeze ilettiği veriler büyük rol oynuyor.

Akıllı Arabalar

Mobil teknolojilerin etkilediği bir sektör de otomotiv sektörü oldu. Akıllı arabaları, kısaca “akıllı telefonla bağlantı kurabilen, internete bağlanabilen” olarak tanımlamak mümkün; fakat bu tanımlama yeterli değil. Zira akıllı araç yaklaşımlarına baktığımızda ortada çok etkileyici teknolojilerin bulunduğunu görüyoruz.

Öncelikle akıllı telefonla bağlanabilirlik yeteneğinden bahsedelim. Günümüzde yeni model orta ve üst seviyede pek çok araçta, akıllı telefon ile bağlanabilirliğin aktif olduğunu görüyoruz. Kimi araçlarda şarj etmek ve Bluetooth üzerinden görüşmek yapmak mümkünken, bazı araçlar bu

basit işlevlerin çok daha fazlasını sunabilecek şekilde tasarlanıyor. Bunlardan en dikkat çeken ise bir kazanın yaşanması halinde bildirim gönderilmesi. En yakın polis merkezine anında bildirilen kaza, özellikle sağlık durumları için çok daha hızlı aksiyon alınmasını sağlıyor. İnternet bağlantılı akıllı arabalar, aynı zamanda yol durumları hakkında bilgi sahibi olarak güvenlikle ilgili uyarılar verebiliyor. Mesela X yolunda hız limiti 50 km/s ise bu hızı geçtiğimiz anda uyarı verilebiliyor.

Kaynak: <http://www.bthaber.com/2016-ya-teknoloji-bakisi/internete-bagli-cihaz-sayisi-ve-turu-gunden-gune-artiyor/1/16703>



Öğrenme Çıktısı

2 Web standartlarının gelişim sürecini açıklayabilme

Araştır 2

İnternet üzerindeki güvenliğin artırılması için alınabilecek diğer önlemlerin neler olabileceğini araştırınız.

İlişkilendir

Günümüzdeki web sitelerinin gelişim süreçlerini kronolojik olarak incelemek üzere archive.org adresini ziyaret ediniz.

Anlat/Paylaş

HTTP ve HTML kavramları arasındaki ilişki ve farkları açıklayınız.

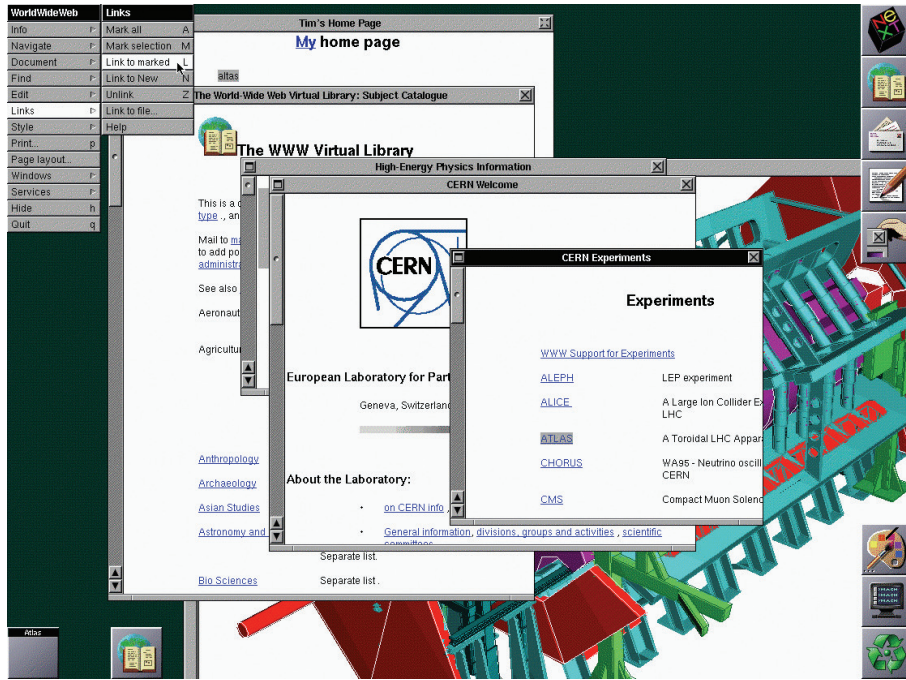
WEB TARAYICILARININ GELİŞİMİ

Web tarayıcısı (web browser); web ağına bağlanan kullanıcı ile web üzerindeki siteler arasındaki bağlantıyı sağlayan yazılımlardır. Web tarayıcıları sayesinde, sunucu adı verilen ve internet üzerindeki sitelere ait verilerin depolandığı bilgisayarlar ile internet ağına bağlanmak isteyen kullanıcıların bilgisayarları (istemci) arasında bağlantı oluşmaktadır. Kurulan bu bağlantı sayesinde tarayıcılar, web sayfalarını oluşturan HTML kodlarını web sitesinin bulunduğu web sunucusundan almakta ve gelen kodlara göre metin, resim, şekil gibi görüntüleri internet kullanıcısının bilgisayarında görüntülemektedir. Ayrıca web tarayıcısı, sunucu ile istemci bilgisayarlar arasındaki bağlantı esnasında verilerin karşılıklı olarak güvenli bir biçimde gönderilip alınması için gerekli olan şifreleme ve güvenlik önlemlerini de yerine getirmektedirler. Web tarayıcılarının gelişimini incelemekten önce web tarayıcılarının çalışma mantığından bahsetmekte yarar vardır. Bir yazılım olan web tarayıcıları, web ağına bağlı olan bir web sayfasının adresi (URL) adres çubuğuna yazılarak görüntülenmek istendiğinde öncelikle web sayfasının bulunduğu sunucuyla iletişim kurar. Web tarayıcısı, web sunucusuyla hangi protokol ile iletişim kuracağına adres çubuğuna yazılan URL adresine göre karar vermektedir. URL adresleri temelde üç bölümden oluşmaktadır. İlk bölüm iletişim protokolünü, sonraki bölüm alan adını ve son bölüm ise hiyerarşik dosya yolunu tanımlamaktadır. Örneğin, web tarayıcısının adres çubuğuna “http://www.gop.edu.tr/index.html” yazıldığında, tarayıcı URL adresinin baş kısmında bulunan “http://” ifadesi ile sunucu bilgisayara HTTP protokolü ile bağlantı kurulacağını belirler. Bağlantı kurulacak sunucunun ismi

ise “www.gop.edu.tr” ifadesiyle belirtilmektedir. Ancak sunucu bilgisayarlara bağlantı yalnız internet protokol numarası (IP adresi) ile kurulduğu için web tarayıcısının sunucu bilgisayarın IP adresini bilmesi gerekir. Bunun için ise internet adreslerinin IP adresi karşılığını kayıtlı tutan DNS sunucularına bağlanarak ilgili internet adresinin IP adresini öğrenir. Daha sonra web tarayıcısı öğrendiği IP adresine sahip web sunucusuyla (sunucu üzerindeki yazılımla) HTTP protokolü kullanarak iletişim kurar. Sunucuyla bağlantı sağlandıktan sonra web tarayıcısı, URL adresindeki son bölüm olan görüntülenecek olan dosyanın hiyerarşik yolunu kullanarak “index.html” isimli dosyayı görüntülemek üzere dosyayı oluşturan tüm nesneleri sunucu bilgisayardan ister. Burada bahsedilen nesne, sadece tek bir URL ile adreslenebilen bir dosyadır. Örneğin, bir web sayfası metin, üç JPEG resim dosyası ve bir video dosyası içeriyorsa bu dosya beş nesneye sahiptir. Yani beş farklı URL ile adreslenebilen dosyadır. Bunun karşılığında web sunucusu kendi içinde bulunan ve HTML işaretleme dili kullanılarak oluşturulmuş olan “index.html” dosyasına ait kodları web tarayıcısının bulunduğu bilgisayara gönderir. Gönderme işlemi ise TCP protokolü ile sağlanmaktadır. TCP protokolü, bir bilgisayardan diğerine gönderilecek iletiyi, ağ üzerinde gönderilebilecek uzunlukta küçük paketlere bölerek gönderilmesi işlemidir. TCP protokolü kullanılarak oluşturulan her küçük pakette gönderilecek iletinin bir parçası, hangi bilgisayardan gönderildiği, hangi bilgisayara gönderileceği gibi çeşitli bilgiler bulunmaktadır. Web tarayıcısı sunucu bilgisayarın göndermiş olduğu ilgili dosyaya ait paket verileri TCP protokolünü kullanarak bir araya getirir ve ilgili dosyaya ait HTML kodlarını oluşturur. Son olarak web tarayıcısı, web sayfası oluşturulması için standart bir işaretleme dili olan HTML dili kodlarını çözümleyerek kodların karşılığı olan metin, resim, tablo, animasyon gibi görüntüleri web sayfasında görüntüler.

Web tarayıcılarının ortaya çıkışı “Web Teknolojisinin ve Web Mimarisinin Doğuşu” başlığında da bahsedildiği gibi 1990 yılında yaşanan hipermetin temelli World Wide Web (WWW) uygulamasının geliştirilmesine dayanmaktadır. CERN’de yazılım mühendisi olarak çalışan Tim Berners-Lee’nin gerçekleştirdiği çalışmalar sonucunda web sayfalarının bulunduğu sunucu bilgisayarlar ile web sayfalarının görüntüleneceği istemci bilgisayarlar arasındaki iletişimi sağlayan HTTP protokolü oluşturulmuştur. Bu iletişim protokolü sayesinde web genelindeki bağlı tüm kaynaklar birbirleriyle iletişim kurabilmektedir. Ayrıca web sayfalarının tasarlanması için kullanılan HTML işaretleme dili ve webe bağlı her bir kaynağı tanımlamak için kullanılan web adresleme yapısı (URL) da geliştirilmiştir. Gerçekleştirilen bu çalışmalar web tarayıcılarının doğması için gerekli olan tüm şartları hazırlamıştır. Bunun sonucunda 1990 yılında ilk web tarayıcısı olan WorldWideWeb ortaya çıkmıştır. Resim 1.1’de WorldWideWeb ile hazırlanmış bir web sayfasının 1993 yılında alınmış ekran görüntüsü bulunmaktadır. Tim Berners-Lee tarafından yazılan ilk web tarayıcısı, WWW kütüphanesi için geliştirilen bir web editörüdür. Basit stil şablonlarını (yazı tipi, yazı puntosu, girinti vb.) ve hipermetin bağlantılarını görüntüleyebilmektedir. Ayrıca gezinti penceresi ile gezinti yapılan önceki ve sonraki sayfalar görüntülenebilmektedir. Tarayıcının tek sınırlılığı ise, NeXTStep işletim sisteminde çalışıyor olmasıdır. Daha sonraki yıllarda web tarayıcısının ismi Nexus olarak değiştirilmiştir.

✓ İlk web tarayıcısı olan WorldWideWeb, 1990 yılında CERN’de yazılım mühendisi olan Tim Berners-Lee tarafından yazılmıştır.



Resim 1.1 İlk web sayfasının görünümü

Kaynak: <https://www.w3.org/People/Berners-Lee/WorldWideWeb.html>

Bin dokuz yüz doksanlı yılların başlarında başlayan web tarayıcıları serüvenine onlarca tarayıcı katılmasına rağmen akıllarda yalnızca kilit noktadaki tarayıcılar kalmıştır. Bu nedenle bu bölümde yalnızca web tarayıcısı sektöründe iz bırakan web tarayıcılarından bahsedilecektir. İlk web tarayıcısı WorldWideWeb'in geliştirilmesinden sonraki yıllarda çeşitli metin temelli web tarayıcıları tasarlanmıştır. Unix işletim sistemlerinde X Windows System arayüzü kullanan bilgisayarlar için tasarlanan Erwise, ViolaWWW, Lynx; Macintosh bilgisayarlar için üretilen ilk web tarayıcısı Samba; yine Unix işletim sistemi için tasarlanan ve fizikçiler arasında yaygın kullanılan MidasWWW bunlardan bazılarıdır. Web tarayıcılarının gelişiminde önemli bir kilometre taşı olarak görülen NCSA Mosaic, 1993 yılında Mark Andreessen tarafından Süper Bilgisayar Uygulamaları Ulusal Merkezi (National Center for Supercomputing Applications – NCSA) laboratuvarlarında geliştirilmiştir. Mosaic web tarayıcısı, webin büyümesinde önemli katkılar sağlamıştır. Bunun en önemli nedeni, Mosaic web tarayıcısının grafik arayüzlü ve platformlar arası bir tarayıcı olması ve video klip, ses, formlar, yer imleri gibi daha önceki tarayıcılarda olmayan birçok yenilik sunuyor olmasıdır. Ayrıca Mosaic, dosya indirmeye de izin vermektedir. Sunduğu yenilikler

sayesinde Mosaic web tarayıcısı kullanımını hızla artırarak webin gelişmesine katkı sağlamıştır. Geliştirilen ilk Mosaic web tarayıcısı, grafik ekran sunabilen UNIX işletim sistemleri için oluşturulmuştur. Ardından Macintosh ve Windows sürümleri de piyasaya sunulmuştur. Web tarayıcılarının büyük çoğunluğu UNIX işletim sistemleri için tasarlanması nedeniyle, Windows işletim sistemleri için 1993 yılında ilk web tarayıcısı olan Cello geliştirilmiştir. Metin temelli olmasına karşın Cello diğer güçlü tarayıcıların birçok özelliğini içerisinde barındırmaktadır. GIF, XBM, BMP gibi resim formatları desteği, dosya kaydetme, yazdırma, yer işareti koyma bunlardan bazılarıdır. Ancak ilerleyen yıllarda Mosaic web tarayıcısının Macintosh ve Windows sürümlerini piyasaya çıkarmasıyla Cello popülerliğini yitirmiştir. 1994 yılında Mark Andreessen ve Jim Clark NCSA'dan ayrılarak ileride Netscape Communication Cooperation olacak olan Mosaic Communications'ı kurmuştur. Kurdukları şirket bünyesinde 1994 yılında Netscape web tarayıcısını piyasaya sürmüşlerdir. Netscape web tarayıcısının piyasaya çıkmasıyla birlikte Mosaic web tarayıcısının popülerliği giderek azalmaya başlamış ve 1995 yılında Netscape fiili standart web tarayıcısı olmayı başarmıştır. Bin dokuz yüz doksan beşin sonuna gelindiğinde Netscape web tarayıcısı, pazar payını

%90'a çıkararak web tarayıcıları sektöründe zirveye ulaşmıştır. Aynı yıl Microsoft firması günümüzdeki en yaygın web tarayıcılarından biri olan Internet Explorer'ı piyasaya sürerek Netscape ile "browser war (tarayıcı savaşı)" olarak adlandırılan web tarayıcısı rekabetini başlatmıştır. İlerleyen üç yılın sonunda Internet Explorer web tarayıcısı savaşını kazanarak Netscape web tarayıcısından liderliği almıştır. Bunda iki önemli unsur rol oynamıştır. Bunlardan birincisi Windows işletim sisteminin web kullanıcıları tarafından tercih edilen bir işletim sistemi olarak popülerliğinin artması, buna karşın Unix işletim sisteminin popülerliğinin düşmesidir. İkinci nokta ise Internet Explorer web tarayıcısının 1996 yılında Windows işletim sistemiyle bütünleşik olarak sunulmasıdır. Ancak Internet Explorer web tarayıcısı liderliğini günümüze kadar koruyamamıştır. Web tarayıcısı pazarında 2004 yılında en üst noktaya ulaşmış (%92) olmasına rağmen sonraki yıllarda yeni web tarayıcılarının çıkmasıyla birlikte sürekli düşüşe geçmiştir.

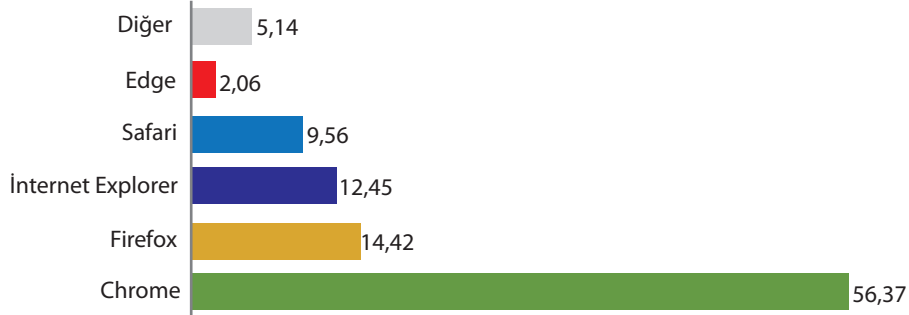
Doksanlı yılların sonlarına gelindiğinde web tarayıcısı sektörü açısından iki önemli web tarayıcısı piyasaya sürülmüştür. Bunlardan ilki Norveç'te bulunan bir telekomünikasyon firması olan Telefon tarafından 1996 yılında geliştirilen Opera web tarayıcısıdır. Piyasaya sürülen Opera web tarayıcısının HTML3 desteği, zoom özelliği ve çoklu resim, video desteği gibi birçok özelliği bulunmaktadır. Hâlâ kullanılan bir web tarayıcısı olan Opera, günümüze kadar birçok özellik eklenerek ücretsiz olarak sunulmaktadır. İkinci web tarayıcı ise, 1998 yılında Netscape Communication Corporation tarafından Netscape web tarayıcısından sonra ikinci nesil web tarayıcısı olarak piyasaya sürülen Mozilla'dır. Mozilla, Netscape firmasının kaynak kodlarını herkese açmasıyla birlikte ortaya çıkan açık kaynak yazılım ailesinin ismidir. Mozilla web tarayıcısı birçok güçlü eklentiye sahiptir ve eklentiler ileride çıkacak olan Firefox web tarayıcısının popülerliğine önemli katkılar sağlayacaktır. Bu eklentiler; e-posta ve haber grubu üyelikleri, internet sayfası geliştirme aracı, internet üzerinden sohbet etmeye olanak sağlayan IRC (Internet Relay Chat – İnternet Aktarımlı Sohbet) istemcisi bunlardan birkaçıdır.

İki binli yıllarda web tarayıcıları alanında yaşanan en önemli gelişme internetin açık ve herkese erişilebilir küresel bir kaynak olarak kalmasını sağlamak amacıyla kurulan Mozilla Vakfı tarafından

2002 yılında Firefox web tarayıcısının ilk sürümünün yayınlanmasıdır. Yayınlanan ilk iki sürümde farklı isimler kullanan Firefox, 2004 yılında yayınlanan üçüncü sürümünde günümüzde kullanılan ismini almıştır. Firefox web tarayıcısı, kendinden önce yayınlanan Mozilla Suite web tarayıcısının geliştirilmiş hâlidir. Firefox web tarayıcısı hızlı internet kullanımı ve gizlilik konusunda en üst düzey olması için tasarlanmıştır. Bu özellikleri sayesinde Firefox, web tarayıcısı pazarında kendine önemli bir yer bulmuş ve 17 Haziran 2008 yılında yayınlanan Firefox 3'ün bir günde sekiz milyondan fazla indirilmesiyle Guinness World Record'un "24 saatte en çok indirilen program" alanında dünya rekoru kırmıştır. Aynı dönemde Internet Explorer'da yaşanan güvenlik açıkları Firefox'un popülerliğine önemli katkı sağlamıştır. Web tarayıcıları sektöründe yaşanan ikinci gelişme ise Apple firmasının 2003 yılında yayınladığı Safari web tarayıcısıdır. Başlangıçta yalnız Macintosh bilgisayarlar için üretilen Safari web tarayıcısı 2007 yılında Windows versiyonunu yayınlamıştır. Macintosh bilgisayarlar için 2003 yılına kadar öncesinde Netscape sonrasında ise Internet Explorer'ın Mac versiyonu standart web tarayıcısı olarak kullanılırken 2003 yılında Apple kendi web tarayıcısını yayınlamakla Mac OS işletim sistemlerinde Safari standart web tarayıcısı olarak kullanılmaya başlanmıştır. Kendine web tarayıcısı pazarında önemli bir yer bulan Safari, 2008 yılında yayınlanan Safari 4 versiyonundaki SquirrelFish JavaScript motoruyla hızını artırarak oldukça beğeni toplamıştır. Günümüz web tarayıcıları içerisinde en çok kullanılan tarayıcı olan Chrome ise 2008 yılında yayınlanmıştır. Google firması tarafından ücretsiz olarak yayınlanan web tarayıcısı olan Chrome, öncelikle Windows versiyonunu sonrasında ise Macintosh ve Linux versiyonuyla piyasada yerini almıştır. Web tarayıcısı üzerinde sunduğu Google arama motoru ile kullanıcıların ilgisini oldukça çekmiştir. Web tarayıcısı piyasasına sonradan girmesine ve Safari, Internet Explorer gibi işletim sistemiyle birlikte sunulmamasına karşın Chrome kısa sürede kullanımını artırarak web tarayıcısı pazarında liderliğe oturmuştur. Web trafiği analiz hizmeti veren StatCounter'ın 2016 yılı verilerine göre dünya çapındaki masaüstü ve tablet cihazlarda kullanılan web tarayıcısı oranına göre Chrome %56 ile ilk sırada yer almaktadır (Resim 1.2). Türkiye'de ise Chrome %76,21'lik kullanım oranıyla ikinci sıradaki Internet Explorer'dan (%8,19) dokuz kattan fazladır. Günümüz web tarayıcıları içerisinde bilinen

ve kullanılan Yandex web tarayıcısı ise 2012 yılında yayınlanmıştır. Google'un açık kaynak web tarayıcısı Chromium temel alınarak tasarlanan Yandex web tarayıcısı, web sayfası güvenliğinde Yandex güvenlik sistemini ve indirilecek dosyaların antivirüs yazılımı Kaspersky tarafından incelenmesiyle güvenlik açısından önemli özellikler taşımaktadır. Ayrıca yavaş internet bağlantısı durumunda kullandığı turbo teknolojisiyle hızlı web tarama özelliği sergilemektedir.

✓ 2008 yılında yayınlanan Firefox 3 web tarayıcısı, bir günde 8.002.530 kez indirilerek Guinness World Record'un "24 saatte en çok indirilen program" kategorisinde dünya rekoru kırmıştır.



Resim 1.2 Web tarayıcılarının 2016 yılı pazar payı

Kaynak: <http://gs.statcounter.com/#desktop+tablet-browser-ww-yearly-2016-2016-bar>

Başlangıçta metin tabanlı, basit biçimsel ve gezinti işlemleri yapabilen yazılımlar olarak hayatımıza giren web tarayıcıları yıllar içerisinde grafik arayüzlü hâle gelerek görsel ve güvenlik anlamında güçlü bir hâle dönüşmüştür. Günümüz web tarayıcıları video, animasyon, ses gibi birçok çokluortam nesnesini destekleyen, VBScript ve JavaScript kodları işleyebilen, ActiveX destekleyebilen ve yüklenecek eklentilerle istenildiği gibi kişiselleştirilebilen yapıdadır.



WEB SAYFASININ YAPISI VE TEMEL ÖZELLİKLERİ

Web sayfası, web ağına bağlanan kullanıcıların web tarayıcıları aracılığıyla görüntüledikleri hipermetin dosyalarıdır. Web sayfaları, 1990 yılında Tim Berners-Lee tarafından hayata geçirilen WWW teknoloji-siyle birlikte ortaya çıkmıştır. Platform bağımsız olarak görüntülenebilen bu dosyalar, başlangıçta metin tabanlı iken günümüzde resim, video, ses, animasyon gibi çokluortam nesnelerini içinde barındıran, kullanıcılarla etkileşime girebilen ve veritabanı desteği ile kodlama (VB Script, JavaScript, SQL vb.) dillerinin kullanılmasıyla kullanıcıların istekleri doğrultusunda özelleştirilebilen bir yapıya bürünmüştür. Web sayfaları, içeriğini oluşturan tüm nesneleri ofis dosyalarında (örneğin, Word dosyası) olduğu gibi içerisinde barındırmamaktadır. Web teknolojisinin yapısı gereği web sayfaları Hipermetin İşaretleme Dili (Hyper-

text Markup Language – HTML) kullanılarak oluşturulmaktadır. Bu işaretleme dili gereği, web sayfasını oluşturan metinler dosya içerisine yerleştirilebilirken sayfada gösterilecek diğer tüm nesneler (resim, ses, video, animasyon vb.) özellikleri ve URL adresleri ile birlikte kodlar içerisinde tanımlanmaktadır. Bu durum, “Web Tarayıcılarının Gelişimi” bölümünde de bahsedildiği gibi bir web sayfasını oluşturan tüm nesnelerin ayrı birer dosya olarak yer almasını gerekli kılmaktadır. Örneğin, bir web sayfasının içeriğinde iki JPEG resim dosyası ve bir SWF (Adobe Flash) dosyası bulunuyorsa bu web sayfasının kendisinden başka üç dosyayı daha kullandığını göstermektedir. Web sayfasında bu iki resim dosyası ve bir animasyon dosyasına ait kodlar bulunmazken sadece web tarayıcılarının bu dosyalara erişmek için kullanacağı URL adresleri yer almaktadır. Bu durum özellikle web tasarımı sürecinde metin dışındaki diğer nesnelerin konumlandırılması, isimlendirilmesi ve web sunucusuna klasör yapısına uygun yerleştirilmesi gibi zorunlulukları beraberinde getirmektedir. Aksi takdirde, yapısal işaretleme doğru yapılmayan yani URL adresi üzerinden ulaşılamayan dosyalar web sayfasının içerisinde görüntülenememektedir.

✓ Web sayfası, web tarayıcıları aracılığıyla görüntülenen hipermetin dosyalarıdır.



dikkat

Bir web sayfası içerisindeki metin dışındaki çokluortam nesnelerinin (resim, video, animasyon vb.) görüntülenebilmesi için web sayfasının kodları içerisinde bu nesnelere nereden ulaşılacağı doğru bir biçimde işaretlenmelidir.

HTML diline ait kodlarla oluşturulan web sayfaları iki farklı biçimde tasarlanabilmektedir. Bunlardan ilki hazırlanan bir web sayfasının kullanıcılar tarafından değiştirilemediği ve sürekli aynı biçimde ekranda görüntülenene sabit içerikli web sayfasıdır. Bu sayfalar web tasarımcıları tarafından bir kez oluşturulup ilgili web sunucusuna yerleştirildikten sonra tüm kullanıcılar tarafından aynı içerikle görüntülenmektedir. Kullanıcıların web sayfasını görüntülemeleri sayfa üzerinde herhangi bir değişikliğe neden

olmamaktadır. Ayrıca bu tür web sayfaları kullanıcı isteği veya zamana göre değişiklik göstermemektedir. Bir diğer web sayfası biçimi ise değişken içerikli web sayfalarıdır. Bu web sayfalarının içerisindeki veriler kullanıcıların isteklerine ve çeşitli durumlara göre değişiklik göstermektedir. Değişken içerikli web sayfalarında bulunan çeşitli form elemanları (metin kutusu, düğme, açılır listeler, seçenek grubu vb.) ve bu form elemanlarından gelen verilere göre içeriği yapılandıracak kodlama dilleri (VB Script, JavaScript, PHP, ASP vb.) yardımıyla içerik kullanıcının istekleri doğrultusunda yapılandırılmaktadır. Örneğin, sınav sonuçları görüntüleme ekranında web sayfalarına girilen veriler doğrultusunda sonuçların oluşturulması, internet mağazalarında yer alan ürünlerin çeşitli kriterler sonucunda filtrelenmesi bunun en güzel örneklerindendir. Bu tür dinamik içerikli web sayfaları oluşturulurken HTML kodlarının yanı sıra çeşitli script dillerinin ve veritabanı etkileşiminin kullanılması gerektiği unutulmamalıdır.

Basit bir web sayfası oluşturmak için yalnız HTML kodları kullanılması yeterlidir. HTML ilk olarak World Wide Web (WWW) yapısı oluşturulurken ortaya konan ve daha sonraki yıllarda 1994 yılında kurulan Dünya Çapında Ağ Birliği (World Wide Web Consortium – W3C) tarafından standartlarının geliştirildiği bir işaretleme dilidir. Webin mucidi Tim Berners-Lee tarafından kurulan ve hâlen onun yöneticiliğini yaptığı W3C, web standartlarının geliştirilmesi için kurulan uluslararası bir kuruluştur. W3C ayrıca webin tüm dünya üzerindeki herkesin kullanımına açık ve webin sağladığı iletişim ve bilgi paylaşımı avantajlarından yararlanabilmesini amaçlamaktadır. W3C bünyesindeki teknoloji uzmanları tarafından webin ihtiyaçları temel alınarak geliştirilen HTML işaretleme dilinin güncel versiyonu 2014 yılında yayınlanan HTML5'tir.

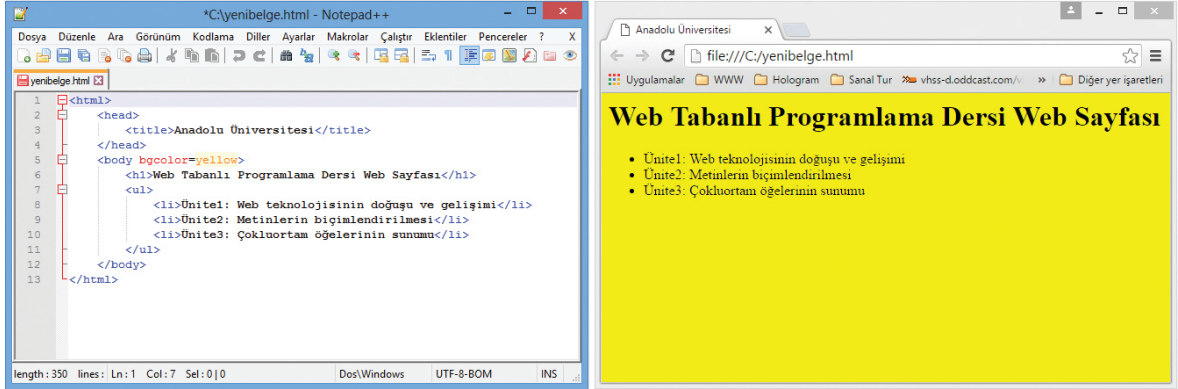


World Wide Web Consortium tarafından yayınlanan HTML5 kodlarını ayrıntılı olarak inceleyebilmek için <https://www.w3.org/TR/html5/> adresinden yararlanabilirsiniz.

HTML dili kullanılarak web sayfası oluşturmak için kullanılabilecek farklı yöntemler bulunmaktadır. Birinci ve en basit yöntem; ASCII editörü, EditPlus, Notepad, Notepad++ gibi çeşitli metin

editörleri kullanarak web sayfası oluşturmazdır. Bunun için öncelikle metin editörleri üzerinden bir sayfa oluşturulmalı, web sayfasını oluşturacak HTML kodları metin editörüne satırlar hâlinde yazılmalı ve daha sonra sayfa uzantısı “.htm veya .html” olacak şekilde değiştirilerek kaydedilmelidir. Oluşturulan bu dosya web tarayıcıları aracılığıyla açıldığında web sayfasının içeriği görüntülenmektedir. Eğer web sayfası üzerinde düzenleme yapıl-

mak isteniyorsa oluşturulan dosya tekrar metin editörleri aracılığıyla açılarak gerekli düzenlemeler yapılmalı ve yeniden kaydedilmelidir. Notepad++ metin editörü kullanarak oluşturulmuş bir web sayfasının görünümü Resim 1.3’te verilmiştir. Sol tarafta Notepad++ metin editörüyle hazırlanmış tasarım görünümü, sağ tarafta ise web tarayıcısı aracılığıyla HTML kodlarının yorumlanmasıyla oluşan web sayfasının görünümü yer almaktadır.



Resim 1.3 Bir web sayfasına ait html kodları ve bu kodların görünümü

Web sayfası oluşturmak için kullanılacak ikinci yöntem ise FrontPage, Dreamweaver gibi çeşitli HTML editörlerinin veya yardımcıların kullanılmasıdır. HTML editörleri yardımıyla web sayfasını oluşturmak metin editörleri yardımıyla web sayfası oluşturmaktan daha basit ve kolaydır. Metin editörleri kullanıldığında HTML kodları tasarımcı tarafından yazılıp kontrol edilirken, HTML editörlerinde yapılmak istenen işlemler kullanıcı tarafından menüler yardımıyla yapılmakta, HTML kodları arkaplanda editörler tarafından oluşturulmaktadır. Bir başka ifadeyle, kullanıcı web sayfasına resim eklemek istediğinde, araç çubuğunda bulunan resim ekleme düğmesini tıklayarak eklenecek resmi seçmekte, boyutu ve konumunu ayarlamakta, bunun için gerekli olan HTML kodlarını ise HTML editörü otomatik olarak kendisi oluşturmaktadır. Bu yöntemde tasarımcı, tasarlamak istediği web sayfasının asıl görünümü üzerinden tasarım yapmakta, gereksinim duyduğunda HTML kodlarına müdahale etmektedir. Web sayfası oluşturmak için kullanılabilecek üçüncü yöntem ise MS Word, Lotus gibi HTML çeviricilerinin kullanılmasıdır. Bu yöntemde kelime işlem programı ile oluşturulan bir Word dosyası veya bir taşınabilir belge formatında (PDF) hazırlanmış bir dosya HTML formatına dönüştürülebilmektedir. Böylelikle dönüştürülen dosya web ağında yayınlanarak web tarayıcıları tarafından görüntülenebilmektedir.

HTML dili, tag (belirteç/etiket) adı verilen kodların kullanıldığı, içeriğin bu kodlar aracılığıyla geliştirildiği bir dildir. Tagların yazımında büyük-küçük harf ayrımı bulunmamaktadır. Ayrıca boşluk, alt satıra inme ve sekmenin önemi yoktur. Ancak burada unutulmaması gereken durum, HTML taglarının İngilizce diline uygun olduğu, bu nedenle büyük harf kullanımında da İngilizce yazım kurallarına dikkat edilmesi gerektiğidir. HTML dili kullanılarak oluşturulan sayfalar sadece bu dili yorumlayabilen yazılımlar özellikle de web tarayıcıları tarafından okunabilmektedir. HTML dili içerisinde her tag, küçüktür ve büyüktür işaretleri arasında yazılmaktadır. Örneğin; <html>, <body>, <table> gibi. Ayrıca açılan taglar genellikle aynı tagın başına / (eğik çizgi) işareti konulmasıyla kapatılmaktadır. Resim 1.3’te bu durum açık bir biçimde görülmektedir. Web sayfası içerisinde görüntülenecek metinler, ilgili taglar arasında yazılmaktadır. Bir web sayfasına ait HTML kodları <html> başlangıç tagı ile başlamakta ve </html> bitiş tagı ile sonlandırılmaktadır. Bir web sayfası temel olarak iki ana bölümden oluşmaktadır. Birinci bölüm <head></head> tagları arasında oluşturulan ve web sayfasına ait metaverilerin tanımlandığı bölümdür. Bu bölümde; web sayfasının tarayıcıda görüntülenecek başlığı, web sayfasının tarayıcıda görüntülenecek resmi (iconu), sayfa içinde kullanılan yazı kodlama tipinin ve karakter setinin ne olduğu, sayfaya bağlı olan harici stil dosyasının konumu, sayfa içi tanımlanan stiller,

tanımlanan scriptler ve web sayfasına ait diğer meta-veriler bu bölümde tanımlanmaktadır. Bu bölümde tanımlanan özellikler, web sayfasının genel yapısıyla ilgili olan verileri içermektedir. İkinci bölüm ise <body></body> tagları arasında oluşturulan ve web sayfasının içeriğine ait verilerin tanımlandığı bölümdür. Bu bölümde; web sayfasının içeriğinde yer alan metinler, paragraf biçimleri, tablolar, bağlantılar, resimler, çokluortam nesneleri, form elemanları gibi web sayfasının içeriğini oluşturan tüm nesneler tanımlanmaktadır. Daha önce de bahsedildiği gibi, sadece sayfa içerisinde görüntülenecek metinler ilgili taglar arasında kullanılmakta, diğer çokluortam nesnelerinin (resim, video, ses, animasyon vb.) özellikleri ve URL adresleri işaretlenerek web tarayıcısının ilgili dosyayı nereden indireceği belirtilmektedir.



dikkat

HTML diline ait kodlar İngilizce olduğu için büyük harf kullanılarak yazılan HTML kodlarında İngilizce yazım kurallarına dikkat edilmesi gerekmektedir.

Her ne kadar HTML dili web teknolojilerinin gereksinimleri doğrultusunda güncellense de dinamik ve görsel öğelerle zenginleştirilmiş web sayfaları yapılabilmesi için ek dillere gereksinim duyulmaktadır. HTML dili ile beraber en çok kullanılan yapı, basamaklı stil sayfaları (Cascading Style Sheets – CSS) olarak adlandırılan biçimlendirme özellikleridir. Basamaklı terimi, stillerin sırasını ve öncelik düzeyini tanımlamaktadır. Stil, tek bir isimle tanımlanan bir dizi biçimlendirme niteliğidir. Bu biçimlendirme niteliği, web tarayıcısına bir elemanı nasıl görüntülemesi gerektiğini bildirmektedir. HTML dili kendi içerisinde metinler veya diğer nesneler

üzerinde sınırlı sayıda biçimlendirme özelliğine sahiptir. Ancak CSS yapısı kullanılarak, web sayfası, metinler, çokluortam nesneleri, form elemanları, tablolar gibi birçok nesne üzerinde esnek ve hassas biçimlendirmeler tanımlanabilmektedir. Ayrıca CSS yapısı kullanılarak oluşturulacak bir web sayfasına ait standart stil şablonu oluşturulabilmekte ve hazırlanacak her web sayfası için bu şablonlar tanımlanarak kullanılabilir. Buradan da anlaşılacağı üzere, CSS hem web sayfası içerisinde bağımsız olarak hem de tüm sayfaları etkileyecek biçimde bağlantılı biçimde de tanımlanabilmektedir. CSS yapısının kullanımı günümüzde web tasarımında olmazsa olmaz konulardan bir tanesidir. HTML diline destek amaçlı kullanılan bir diğer yapı ise farklı script dillerinin kullanılmasıdır. Özellikle günümüz web sayfalarında görünen kullanıcı etkileşimi ile oluşturulacak dinamik web sayfaları bu diller kullanılarak tasarlanmaktadır. Veritabanı desteği ile birleştirilerek kullanılan bu tür web sayfalarında kullanıcıdan gelen veriler doğrultusunda sayfa oluşturulabilmekte veya sayfa içeriği filtrelenebilmektedir. PHP, ASP, ASP.NET gibi programlama dilleri, dinamik web sayfaları oluşturmak için yararlanılan dillerin başında gelmektedir. Özellikle veritabanı kullanımı, internet tabanlı programlama için bu dillerin kullanımını zorunlu kılmaktadır. Ayrıca web teknolojisinin gelişimiyle birlikte web sayfalarının arkaplanında kodlamanın yapılabilmesine olanak sağlayan JavaScript dili, günümüz web sayfalarında yoğun bir biçimde kullanılmaktadır. Özetle, bir web sayfası oluşturmak için temel HTML kodlarının yeterli olduğu, ancak görsel olarak zengin ve etkileşimli dinamik web sayfaları oluşturmak için HTML kodlarının yanında diğer PHP, ASP, JavaScript gibi dillerin kullanılması gerektiği unutulmamalıdır.

Öğrenme Çıktısı



4 Web sayfalarının temel özelliklerini ifade edebilme

Araştır 4

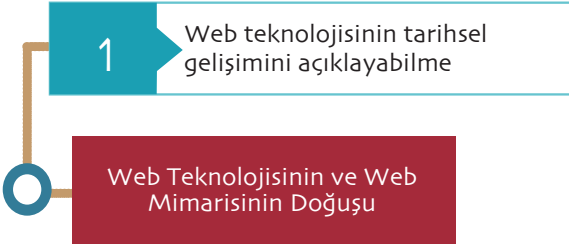
Web sayfalarının farklı işletim sistemleri üzerinde çalışan web tarayıcıları tarafından aynı şekilde gösteriliyor olmasının sebebini açıklayınız.

İlişkilendir

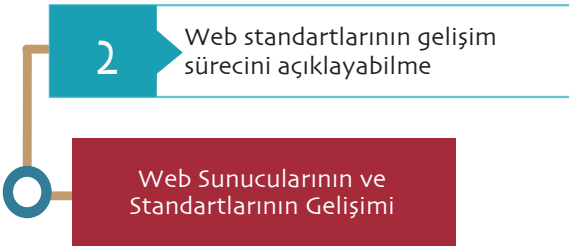
Ziyaret ettiğiniz bir web sayfasını oluşturan HTML kodlarına ulaşip inceleyiniz.

Anlat/Paylaş

CSS'in, HTML teknolojisine üzerine neler koyduğunu açıklayınız.



Web teknolojisinin tarihsel gelişimi kronolojik olarak incelendiğinde, kendi içerisinde farklı gelişim süreci takip eden üç dönemde ele alınabileceği görülmektedir. Birinci dönem, 19. yüzyılın başlarından itibaren başlayıp 20. yüzyılın ortalarına kadar devam eden bilgisayar öncesi dönemdir. Bu dönemde, elektrik akımı teller üzerinden bir noktadan başka bir noktaya iletilmiş, iletişim amaçlı geniş ölçekli kablolar döşenmiş ve telefonun icadıyla birlikte oluşturulan merkezi telefon ağ sistemleriyle kablolu iletişim omurgası oluşturulmuştur. İkinci dönem, 20. yüzyılın ortalarından itibaren başlayıp sonuna kadar devam eden bilgisayarlar dönemidir. Bu dönem, bilgisayarların kullanımının ve onlar aracılığıyla yapılabilecek işlemlerin arttığı dönemdir. Özellikle bilgisayarların modern dünyanın karmaşık problemlerine çözüm olabileceği düşüncesinin yaygınlaşmasıyla birlikte, bu dönemde farklı bilim ve araştırma kuruluşundaki bilgisayarların birbirleriyle veri paylaşımına yönelik çalışmalara ağırlık verilmiştir. 1957 yılında kurulan Amerikan Savunma Bakanlığı'na bağlı İleri Araştırma Projeleri Kurumu (ARPA) tarafından yürütülen bilgisayarların ağı projesi ARPANET, web teknolojisinin geleceğiyle ilgili önemli gelişmelerin yaşanmasına kaynaklık etmiştir. Bu dönemde bilgisayar iletişim protokolleri (TCP/IP), dosya aktarım protokolleri (FTP), e-posta protokolleri (SMTP), alan adı sistemi (DNS) gibi webin arkaplanında çalışan birçok işleğin temelli atılmıştır. Üçüncü dönem ise 1990 yılından itibaren başlayıp günümüze kadar devam eden WWW dönemidir. Bu dönemde, farklı ortamlardaki verilerin hipermetin temelli sayfalar üzerinden birbirleriyle ilişkilendirilmesi ve tüm kullanıcıların kullanımına açılması fikriyle çalışmalar başlatılarak şuan kullandığımız web'e ait birçok gelişme yaşanmıştır. Hipermetin transfer protokolleri (HTTP), hipermetin işaretleme dili (HTML), benzersiz kaynak konumlaması (URL), sunucular, web tarayıcıları ve web sayfaları gibi birçok gelişme bu dönemde yaşanmış ve özelliklerini geliştirerek devam etmiştir.



Web teknolojisinin gelişimine paralel olarak kullanılan teknoloji standartları denince akla E-posta, Web, DNS, FTP, PROXY ve güvenlik kavramlarının geldiği söylenebilir. İnternet'in ilk yıllarından bu güne yoğun bir şekilde kullanılmakta olan E-Posta teknolojisi, SMTP (Simple Mail Transfer Protocol – Basit Posta İletim Protokolü) üzerinden iletilir. Web sayfalarının gönderim ve alım süreçlerinde ise HTTP (Hypertext Transfer Protocol – Hipermetin İletim Protokolü) kullanılmaktadır. Alan adı-IP adreslerinin dönüşüm süreçlerinin yürütülmesini sağlayan DNS ise, yine kendi adıyla anılan DNS Protokolünü (DNSP) kullanır. İnternet üzerindeki dosya transferi süreçlerine ismini veren FTP (File Transfer Protocol – Dosya İletim Protokolü) ise halihazırda bir protokol isimlendirmesidir. Diğerlerine benzer şekilde PROXY ve güvenlik gerektiren süreçlerin de kendilerine özel protokolleri vardır. Söz konusu protokoller, internet'in gelişimi sürecinde ortaya çıkan ihtiyaçlar doğrultusunda değiştirilip güncellenerek günümüzdeki hallerine ulaşmışlardır. İnternet üzerinden, internet araçlarını kullanacak olan yazılımların ilgili protokole uyacak şekilde çalışması, dolayısıyla söz konusu protokoller tarafından ortaya konmuş olan standartlara uyması gerekmektedir.

3

Web tarayıcılarının nasıl çalıştığını açıklayabilme

Web Tarayıcılarının Gelişimi

Web tarayıcıları HTML dili ile oluşturulmuş hipermetin temelli sayfaları görüntüleyen yazılımlardır. Bir web tarayıcısının adres çubuğuna bir web sitesinin adresi yazılarak görüntülenmek istediğinde, öncelikle web tarayıcı web sitesine ait dosyaları barındıran sunucu bilgisayarlarla hangi protokolle iletişim kuracağına karar vermektedir. Bunun için görüntülenmek istenen adresin baş kısmında bulunan protokol türüne (http, ftp vb.) ait ifadeyi temel almaktadır. Daha sonra görüntülenmek istenen web sitesine ait alan adını temel alarak iletişim kurulacak sunucu belirlenmektedir. Alan adı, web adresinde protokol ifadesinden sonraki kısımdır. Sunucu bilgisayarın alan adı belirlendikten sonra web tarayıcısı sunucu bilgisayarın IP adresine ulaşması gerekmektedir. Bunun için web tarayıcı yerel DNS sunucusu üzerinden ulaşmak istediği alan adına sahip sunucunun DNS sunucusuna bağlanarak web sunucusunun IP adresine erişmektedir. Daha sonra eriştiği IP adresini kullanarak web sunucusuyla istenen protokolle bağlantı kurarak ilgili web sayfasını istemektedir. İstenecek web sayfası ise web adresinin alan adından sonraki kısımdır. Bu kısım hiyerarşik olarak ilgili dosyanın nerede olduğunu sunucuya bildirmektedir. Kurulan bağlantı sonucunda web sunucusundan gelen veri paketleri web tarayıcısı tarafından yorumlanarak ilgili web sayfası görüntülenmektedir.

4

Web sayfalarının temel özelliklerini ifade edebilme

Web Sayfasının Yapısı ve Temel Özellikleri

Web sayfaları, temelde HTML işaretleme dili kullanılarak oluşturulmaktadır. Bu dile göre hazırlanan web sayfaları, web tarayıcıları tarafından çözümlenerek sayfanın içeriği oluşturulmaktadır. Web sayfaları oluşturmak için metin editörlerinden, HTML editörlerinden veya HTML dönüştürücülerinden yararlanılabilmektedir. HTML dili ile tüm kullanıcılar tarafından aynı şekilde görünen sabit içerikli web sayfaları kolaylıkla oluşturulabilmektedir. Ancak, değişken içerikli web sayfaları oluşturmak için bu dilin yanında veritabanı ve veritabanı yönetiminde kullanılacak kodlama dillerinin kullanılması gerektiği unutulmamalıdır. Ayrıca çeşitli script dilleri kullanılarak web sayfalarının daha görsel olarak hazırlanması ile kullanıcıdan gelecek verilere göre yeniden yapılandırılması gibi kullanıcı etkileşimi özellikleri eklenebilmektedir. Web sayfası içerisindeki nesneler üzerinde daha esnek ve geniş biçimlendirme yapabilmek için CSS yapısı kullanılmaktadır. Web sayfalarının bir diğer özelliği ise farklı işletim sistemlerinde dahi benzer şekilde görüntülenebilmesidir. Ancak burada web sayfalarını görüntüleyen web tarayıcılarının işletim sistemleriyle uyumlu çalışması gerektiği unutulmamalıdır.

1 Web teknolojisinin atası olarak sayılan ve Amerikan İleri Araştırma Projeleri Kurumu tarafından geliştirilen proje aşağıdakilerden hangisidir?

- A. ALOHANET B. TELENET
C. ARPANET D. SATNET
E. CYCLADES

2 Alan adı sistemi olan DNS, hangi yıl hayata geçmiştir?

- A. 1969 B. 1971
C. 1976 D. 1984
E. 1990

3 World Wide Web'in mucidi olarak belirtilen ve CERN'de yazılım mühendisi olarak çalışmış kişi aşağıdakilerden hangisidir?

- A. Marc Andeeseen
B. Tim Berners - Lee
C. Ted Nelson
D. Robert M. Metcalfe
E. Vint Cerf

4 Sunucularla ilgili aşağıdaki ifadelerden hangisi **yanlıştır**?

- A. 24 saat boyunca çalışmaları
B. Web ağına kesintisiz bağlı olmaları
C. Donanımsal olarak güçlü özelliklerinin olması
D. Kullanım amacına göre farklı türlerinin bulunması
E. Sadece Linux işletim sistemi çalıştırması

5 E-posta sunucularının kullandığı iletişim protokolü aşağıdakilerden hangisidir?

- A. SMTP B. DNS
C. TCP D. HTTP
E. IP

6 Geliştirilen bir web sitesinin web ağı üzerinden yayınlanabilmesi için web sitesine ait tüm dosyalar aşağıdaki sunuculardan hangisine yüklenmelidir?

- A. Güvenlik sunucusu B. DNS sunucusu
C. Web sunucusu D. FTP sunucusu
E. Proxy sunucusu

7 Dosya transfer protokolü olarak adlandırılan ve web ağı üzerindeki sistemlerin birbirleriyle dosya paylaşımı için kullandıkları protokol aşağıdakilerden hangisidir?

- A. HTTP
B. TCP
C. DNS
D. FTP
E. Proxy

8 Bir web tarayıcısının adres çubuğuna "http://www.anadolu.edu.tr/akademik/fakulteler/152/egitim-fakultesi/genel-bilgi.html" ifadesi yazılarak ilgili sayfa görüntülenmek istediğinde web tarayıcısının bağlanmak isteyeceği sunucunun alan adı aşağıdakilerden hangisidir?

- A. anadolu.edu.tr
B. akademik
C. akademik/fakulteler
D. fakulteler/152/egitim-fakultesi
E. akademik/fakulteler/152/egitim-fakultesi/genel-bilgi.html

9 İçerisindeki verilerin kullanıcının istekleri ve çeşitli durumlara göre değişiklik gösterdiği bu tür web sayfalarına ne ad verilir?

- A. Sabit içerikli web sayfası
B. Değişken içerikli web sayfası
C. Metaveri web sayfası
D. Durağan web sayfası
E. Hareketli web sayfası

10 Bir web sayfası içerisindeki nesneler üzerinde daha hassas ve esnek biçimlendirme niteliği tanımlanmak istendiğinde aşağıdaki yapılardan hangisi kullanılmalıdır?

- A. PHP
B. ASP
C. VB Script
D. JavaScript
E. CSS

1. C

Yanıtınız yanlış ise “Web Teknolojisinin ve Web Mimarisinin Doğuşu” konusunu yeniden gözden geçiriniz.

2. D

Yanıtınız yanlış ise “Web Teknolojisinin ve Web Mimarisinin Doğuşu” konusunu yeniden gözden geçiriniz.

3. B

Yanıtınız yanlış ise “Web Teknolojisinin ve Web Mimarisinin Doğuşu” konusunu yeniden gözden geçiriniz.

4. E

Yanıtınız yanlış ise “Web Sunucularının ve Standartlarının Gelişimi” konusunu yeniden gözden geçiriniz.

5. A

Yanıtınız yanlış ise “Web Sunucularının ve Standartlarının Gelişimi” konusunu yeniden gözden geçiriniz.

6. C

Yanıtınız yanlış ise “Web Sunucularının ve Standartlarının Gelişimi” konusunu yeniden gözden geçiriniz.

7. D

Yanıtınız yanlış ise “Web Sunucularının ve Standartlarının Gelişimi” konusunu yeniden gözden geçiriniz.

8. A

Yanıtınız yanlış ise “Web Tarayıcılarının Gelişimi” konusunu yeniden gözden geçiriniz.

9. B

Yanıtınız yanlış ise “Web Sayfasının Yapısı ve Temel Özellikleri” konusunu yeniden gözden geçiriniz.

10. E

Yanıtınız yanlış ise “Web Sayfasının Yapısı ve Temel Özellikleri” konusunu yeniden gözden geçiriniz.

1

Araştır Yanıt Anahtarı

Araştır 1

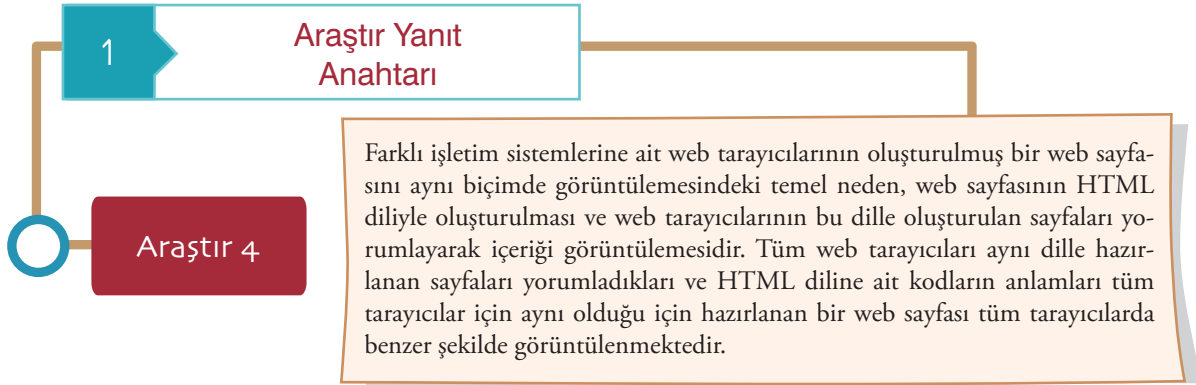
İngilizce bir kavram olan “Web”, örümcek ağı anlamına gelir. Söz konusu içerik ağlarının mantıksal yapısının örümcek ağına benzer olması, Web ismiyle anılmasını sağlamıştır.

Araştır 2

Güvenlik sunucularının dışında web ağı üzerinden gelebilecek saldırılara yönelik izinsiz girişleri engelleme sistemleri kurulabilir. Bu sistemler ticari kuruluşlar tarafından verilen bir hizmet olup, güvenlik duvarlarından daha geniş kontrol olanağı sağlamaktadır. Bunun dışında gelişmiş güvenlik ve antivirüs yazılımlarının ağ saldırılarını önlemek amaçlı birçok özelliği bulunmaktadır. Web erişiminin korunması, HTTP protokolü denetimi, URL adresi yönetimi, e-posta istemci koruması gibi birçok özelliğe sahip yazılımlar ağ saldırılarında kullanılabilmektedir.

Araştır 3

Web trafiği analiz hizmeti veren StatCounter’ın 2016 yılı verilerine göre dünya genelindeki mobil cihazlarda kullanılan web tarayıcısı oranına göre Chrome %36,02 ile ilk sırada yer almaktadır. İkinci sırada %18,3 kullanım oranı ile UC Browser, üçüncü sırada %18,02 kullanım oranı ile Safari, dördüncü sırada %10,91 kullanım oranı ile Opera ve beşinci sırada %8,61 kullanım oranı ile Android web tarayıcısı gelmektedir. Buna göre Chrome web tarayıcısı günümüzde mobil cihazlarda da en çok tercih edilen web tarayıcısıdır.



Kaynakça

- Abhay Bhushan. (t.y.). Erişim tarihi: 22.07.2016, <http://www.internet-guide.co.uk/AbhayBhushan.html>.
- Banks, M. (2008). *On the way to the Web : The secret history of the Internet and its founders*. Berkeley, CA, US: Apress.
- Berghel, H. (1998). Who won the Mosaic war? *Communications of the ACM*, 41(10), 13-16.
- Bush, V. (1945). As we may think. *The Atlantic*. 101-108.
- Çölkesen, R. (2009). *Bilgisayar ağları ve internet mühendisliği*. İstanbul: Papatya yayıncılık.
- Demir, S. (2012). İnternet teknolojisi. İ. H. Sugözü (Ed.), *Temel bilgi teknolojileri* (s.355-394). Ankara: Nobel Yayıncılık.
- Gökgül, M. (2014). *Türkiye’de elektronik ticaret ve işletmelere, tüketicilere sağladığı avantajlar, dezavantajlar*. (Yayınlanmamış Yüksek Lisans Tezi). İstanbul Kültür Üniversitesi, İstanbul.
- ICANN. (2015). *The IANA Functions: An introduction to the Internet Assigned Numbers Authority (IANA) Functions*. Erişim tarihi: 22.07.2016, <https://www.icann.org/en/system/files/files/iana-functions-18dec15-en.pdf>.
- Internet pioneers* (t.y.). Erişim tarihi: 18.07.2016, <http://www.ibiblio.org/pioneers/index.html>.
- Kaymaz, G. (2008). Ray Tomlinson’un hayatı. *ODTÜ Bilgisayar Topluluğu Elektronik Dergisi*. Erişim tarihi: 20.07.2016, <http://e-bergi.com/y/Biyografi-Ray-Tomlinson>.
- Kurose, J.F. ve Ross, K.W. (2007). *Bilgisayar ağları*. (G. Özyeri, Çev.) İstanbul: Alfa Yayınları.
- Metcalf, R.M. ve Boggs, D. R. (1976). Ethernet: Distributed packet swiching for local computer networks. *Communications of the ACM*, 19(7), 395-404.
- Page, K. A. (2006). *Macromedia Dreamweaver 8: Kaynağından eğitim*. (A. Ortakmaç, Çev.) İstanbul: Medyasoft Yayınları.
- PCNET. (2011). *İnternetin ilk çağı*. Erişim tarihi: 18.07.2016, <http://www.pcnet.com.tr/makaleler/internet/internetin-ilk-cagi/>.
- Şenel, H. (2011). İnternet’in altyapısı. C.H. Aydın, Y. Hoşcan ve A.E. Özkul (Ed.) *Temel bilgi teknolojileri* (s.241-289). Eskişehir: Anadolu Üniversitesi Yayınları. Yayın No: 1418.
- Uysal, M. (2001). *Web teknolojileri*. İstanbul: Beta yayıncılık.
- Van der Vlist, E., Ayers, D., Bruchez, E., Fawcett, J. ve Vernet, A. (2007). *Web 2.0 teknolojileri*. (G. Özyeri, Çev.) İstanbul: Alfa Yayınları.
- Wikiwand. (t.y.). *Safari (web browser)*. Erişim tarihi: 20.07.2016, [http://www.wikiwand.com/en/Safari_\(web_browser\)](http://www.wikiwand.com/en/Safari_(web_browser)).
- Wikiwand. (t.y.). *Yandex browser*. Erişim tarihi: 20.07.2016, http://www.wikiwand.com/en/Yandex_Browser.
- World Wide Web Consortium. (2004). *How it all started*. Erişim tarihi: 22.06.2016, <https://www.w3.org/2004/Talks/w3c10-HowItAllStarted>.
- World Wide Web Foundation. (t.y.). *History of the web*. Erişim tarihi: 20.07.2016, <http://webfoundation.org/about/vision/history-of-the-web/>.

■ Bölüm 2

Metinlerin Biçimlendirilmesi

öğrenme çıktıları

Html'de Metin Kullanımı

- 1 Web içeriğini oluşturan metinler üzerinde biçimsel düzenlemeler yapabilme
- 2 Paragraf yapıları oluşturabilme
- 3 Liste yapıları oluşturabilme
- 4 Web sayfalarına bağlantılar ekleyebilme

2

Metinlerde Stil Kullanımı

- 5 Metinlerde stil kullanımını açıklayabilme

Anahtar Sözcükler: • Karakter Setleri • Metin Özellikleri • HTML Yapısı • Paragraflar • Listeler • Köprüler • CSS • Metne Dayalı Stiller

<body style="margin: 0; background-color: #f0f0f0;">
<div id="main_content">
<h1 style="font-family: Courier, monospace; font-size: 24px; margin: 0; padding: 10px 0 0 0;">
<p style="font-size: 12px; margin: 0; padding: 0 0 0 20px;">
<p style="font-family: monospace; font-size: 12px; margin: 0; padding: 0 0 0 20px;">

GİRİŞ

Metin kullanımı özellikle web sayfası tasarım sürecinin ilk çıktığı yıllarda aktarılan bilginin temsilcisi durumunda iken, çoklu ortam özelliklerinin gelişmesine karşın hâlen önemli oranda web sayfalarında tercih edilmektedir. Bu tercihte muhtemelen bilgisayar ekranı da dâhil olmak üzere çoğu yerde okuma alışkanlığının baskın gelmesi gerekçe gösterilebilir. Bu açıdan günümüz web sayfası tasarımında da metinler çok sık kullanılan ortam tasarım unsurlarıdır.

HTML'DE METİN KULLANIMI

HTML kodlamaya yeni başlayanlar için metin kullanımı konusunda karşılaşılabilecek ilk sorun kelime işlemci programlarında yapılan metin kullanım etkilerinin doğrudan program ara yüzünden de beklenmesidir. Ancak metinle alakalı html yazımında maalesef html kodlarının ayrı bir yazım kuralları vardır. Aslında yazılan html (Hypertext Markup Language – Metin İşaretleme Dili) kodları bir yazım kurallar listesidir ve bu kodlar Explorer, Firefox, Chrome veya Yandex gibi tarayıcılar aracılığı ile derlenmektedir. Yani html kodları ile aslında hem tasarım yapmakta, hem de bir kod yazmış olmaktadır. Bu açıdan yazılan metinlerin nasıl, nerede ve ne şekilde görünmesi gerektiği html'de etiket olarak adlandırılan kodlarla metin içerisinde belirtilmelidir. Örneğin paragraf içerisindeki bir kelime koyu olacak ise kelime işlemci programda seçilerek kalın yapılması için ilgili sembole basılmalı veya kısayol tuşu kullanılmalıdır. Ancak html kodlama sisteminde ilgili metnin içerisinde kendisini takip eden metinden hemen önce `<b>` etiketi ilave edilir. Eğer bu etiket sonlandırılmazsa kendisinden sonra gelecek tüm metinleri koyulaştıracaktır. Bunu engellemek için `</b>` etiketi ile koyu olması istenilen yerden hemen sonra sonlandırmak gerekir. Benzer şekilde kelime işlemci programı içerisinde satır sonu vermek (yani enter tuşuna basmak) alt satıra geçişi sağlarken, html kodları için aynı anlama gelmemektedir. `<br>` etiketi html'de enter tuşunun işlevini yerine getirmekte, tarayıcı da alt satıra geçişi sağlamaktadır.

dır. Benzer şekilde bir html yapısı içerisinde boşluk tuşuna çok sayıda basılması tarayıcı da kelimeler arasında çok sayıda boşluk olacağı anlamına gelmemektedir. Html'de kelimeler arasına boşluk vermek için ` ` etiketi verilmek istenen boşluk sayısı kadar kullanılmalıdır. Bir başka örnek olarak html sisteminde `<` veya “karakterleri özel karakter grubunda yer almaktadır. Özel karakterler doğrudan html metin yazımında kullanılamazlar. Html'de özel karakterler için aşağıdaki tablo şeklinde kullanım gereklidir.

✓ Html işaretleme dilinde `<`, `,`, `“`, gibi özel karakterlerin yazımında `&` ile başlayan özel kodlar bulunmaktadır.

Tablo 2.1 Html'de bazı özel karakterlerin kullanımı

Sonuç	Açıklama	Özel Karakterleri
	Fazladan boşluk	<code>&nbsp;</code>
<code><</code>	Küçüktür	<code>&lt;</code>
<code>></code>	Büyüktür	<code>&gt;</code>
<code>&</code>	Ve	<code>&amp;</code>
<code>“</code>	Alıntı	<code>&quot;</code>



Özel karaktere ait daha fazla bilgi edinmek için <http://www.somut.net/459/webmaster/ozel-html-karakter-kodlari-special-characters.1.html> bağlantısından yararlanabilirsiniz.

Yukarıda ifade edildiği gibi html etiketleri farklı tarayıcılar üzerinde satır satır yorumlanmakta, tarayıcıda web sayfalarına dönüşmektedir. Bu nedenle farklı tarayıcılarda aynı siteler farklı şekillerde görülebilmektedirler. Kod bilgisi oyun yazımı, hataların bulunarak düzeltilmesi gibi yerlerde öneme sahiptir ve bu açıdan ileri düzey web tasarımı için önemlidir. Bu ünite de ise metin tabanlı html kodlarına yer verilmiştir.



araştırmalarla ilişkilendir

Web üzerindeki metinsel içerik, olağanüstü bir hızla artmaktadır. Bu yönüyle web, veri madenciliği alanında çalışan araştırmacılar için büyük bir ilgi odağı haline gelmiştir. HTML'nin etiketler aracılığıyla yapılandırılmış ve işaretlenmiş olması da, bu süreci kolaylaştırmaktadır. Metnin başlık, içerik veya listeler şeklinde kodlanmış olması, veri madenciliği süreçlerinin otomatikleştirilmesi anlamında oldukça yarar sağlamaktadır. Kumar ve arkadaşları (1999) tarafından yapılan çalışmada, web'e yönelik madencilik uygulamalarına ilişkin durumlar

ele alınmış ve söz konusu uygulamalar üçe ayrılmıştır. Bu sınıflandırmada Web içeriğine ilişkin madencilik uygulamaları, web yapısının ve kullanımının yanı sıra ayrı bir başlık altında ele alınmıştır. Yapılan çalışmanın sonucunda, web veri madenciliğinin gelecekte oldukça önem kazanacağı vurgulanmıştır.

Kaynak: Madria, Sanjay Kumar, et al. "Research issues in web data mining." *International Conference on Data Warehousing and Knowledge Discovery*. Springer Berlin Heidelberg, 1999.

Karakter Seti Özelliklerinin Belirlenmesi

Html kodu yazımı, belirli bir kalıp içerisinde olmaktadır. Html, head ve body olarak üç parça bilginin yer aldığı bu yapıda < ve > işaretleri içerisine ifadeler yazılırken, bu ifadelerin her birine etiket denir. Ayrıca kod yapılarının bittiğini belirtmek için </> şeklinde bir kullanım söz konusudur. Bu açıdan temel bir html kod yapısı Resim 2.1'deki gibidir.

```

1 <html>
2
3 <head>
4 <meta http-equiv=content-type content=text/html; charset=utf-8>
5
6 <title>Ahmetin Kişisel Web Sayfasıdır</title>
7
8 </head>
9
10 <body>
11 Site içerisinde yer alacak bilgiler. Yazı, resim, video, tablo vb.
12 </body>
13
14 </html>

```

Resim 2.1 Html'in temel yapısı

Resim 2.1'de görüleceği üzere bir html tasarımı <html> etiketi ile başlar ve </html> etiketi ile biter. Ayrıca şart olmamakla birlikte <head> </head> bloğu ile web sayfasına ait üst düzey ayarlar yapılabilir. Örneğin başlık vermek için <title> </title> etiketleri arasına sitenin tarayıcı açıldığında görünmesi istenen başlık kısmı bu kod bloğu ile verilir. Ayrıca sitede eğer Türkçe karakter sorunu yer alıyorsa (örneğin i yerine y görünüyorsa vb.) bu kısımda kullanılacak <meta etiketi ile Türkçe gösterim ayarı yine bu kısımdan yapılabilir. Bunun için aşağıdaki gibi bir etiketin <head> </head> etiketleri arasına yazılması yeterli olacaktır.

```
<meta http-equiv=content-type content=text/html; charset=utf-8>
```

Bazı sitelerde sadece <meta charset =UTF-8> şeklinde de kullanılmakla beraber aynı işlevi yerine getirmektedir. Ayrıca bu blokta birden fazla kez meta etiketi kullanılabilir. Bu durumda her bir etiket kendi işlevini yerine getirecektir. Örneğin sitenizi belirli kelimeler ile ön plana çıkarmak isterseniz meta etiketini aşağıdaki şekilde kullanmanız gerekir.

```
<meta="keywords" content = "oyun tasarımı, html, Ahmet hoca, tasarım, metin ekleme">
```


Benzer şekilde `<meta http-equiv="refresh" content="10; url=anasayfa.html">` etiketini kullanmanız durumunda açılan sayfa 10 saniye görünecek sonra otomatik olarak *anasayfa.html* sayfasına yönlendirilecektir.

Meta etiketleri içerisinde metin düzenleme adına özellikle charset önemli bir kullanımdır. Türkçe karakter sorunu yaşanması durumunda UTF-8 veya windows-1254 atanarak sorun çözülecektir.

Html kod yapısı içerisinde yer alan bir diğer kod bloğu ise `<body>` olup, resim, yazı, video, tablo gibi içeriğe ait tüm bilgiler bu kod bloğu arasında verilir.

✓ Türkçe sorununun çözümünde meta etiketine charset parametresi olarak UTF-8 veya Windows-1254 atamak sorunu çözecektir.

Metin Özelliklerinin Düzenlenmesi

Bu bölümü, bir metin gösteriminde olabilecek temel işlemler senaryosu üzerinden açıklamak daha doğru olacaktır. Temel işlemler olarak yazıyı başlık hâline getirmek, yazının büyüklüğünü, rengini, yazı tipini değiştirmek, kalın, altı çizili veya italik olarak vurgulamak ön plana çıkartılabilir. Resim 2.2'de metin özelliklerini içeren bir paragraf verilmiştir.

Türkiye'nin Coğrafi Özellikleri

Türkiye coğrafi bölge olarak 7 farklı bölgeden oluşmaktadır. **Büyük** olarak en büyük coğrafi bölge **Doğu Anadolu Bölgesi**, en küçük coğrafi bölge ise **Güney Doğu Anadolu** bölgesidir. Diğer yandan nüfus yoğunluğu olarak en kalabalık bölge ise **Marmara Bölgesi**dir. **İstanbul** yaklaşık olarak 14 milyon nüfusu ile **Avrupalı** en büyük, **dünyanın** ise 5. büyük ilidir. |

Resim 2.2 Metin özelliklerini içeren örnek paragraf

Bir web sayfasındaki yazının başlık türü `<h>` etiketi ile belirtilir. Bu etiket yanına 1'den 6'ya kadar sayı alır ve böylece yazı başlığının hangi derecede öneme sahip olduğu belirtilebilir. Bir kitabın içindekiler sayfası düşünüldüğünde bunların her biri bir başlıktır ve metin içinde `<h>` etiketi ile işaretlenmelidirler. Ancak başlık biter bitmez metin bilgisinin başlık olmadığını belirtmek için `</h>` etiketi kullanılmalıdır. Aynı düzeydeki başlıklar aynı numara ile, alt düzeydeki başlıklar ise daha düşük sayıdaki numara ile `<h>` etiketinde kullanılır. Resim 2.3'te başlıkların nasıl kullanıldığına dair etiketler numaraları ile birlikte görülebilir.

✓ `<h>` etiketi 1 ile 6 arasında değer alır ve sayı arttıkça başlığın büyüklüğü azalmaktadır. Ters orantı söz konusudur.

```

1 <html>
2 <head>
3   <title>Başlık Etiketleri</title>
4   <meta http-equiv=content-type con
5 </head>
6 <body>
7   <h1>Birinci derece başlık </h1>
8   <h2>İkinci derece başlık </h2>
9   <h3>Üçüncü derece başlık </h3>
10  <h4>Dördüncü derece başlık </h4>
11  <h5>Beşinci derece başlık </h5>
12  <h6>Altıncı derece başlık </h6>
13 </body>
14 </html>
```

Birinci derece başlık

İkinci derece başlık

Üçüncü derece başlık

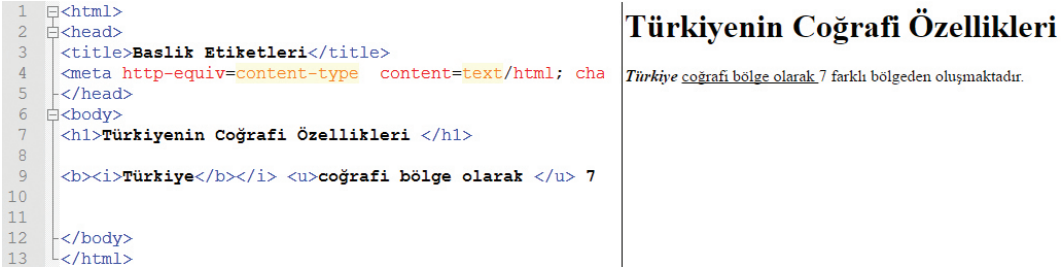
Dördüncü derece başlık

Beşinci derece başlık

Altıncı derece başlık

Resim 2.3 Metin içerisinde başlıkların kullanımı

Metin içerisindeki yazıların vurguları için çoğunlukla koyu, altı çizili ve italik olmak üzere kullanılacak üç etiket vardır. Bu etiketler koyu için `<b>` , italik için `<i>` ve altı çizili için `<u>`'dur. Vurgu yapılacak kelime veya kelimelerden önce bu etiketler kullanılmaya başlanır, bitiminde ise kullanılan etikete göre `</b>`, `</i>` veya `</u>` ile bitirilmelidir. Aksi durumda etiket etkisini sürdürmeye devam eder. Resim 2.2'deki metni oluşturmak için önce başlık, sonra Türkiye yazısının koyu ve italik olması ve devamında gelen yazının ise koyu ve italik olmayıp altı çizili olması aşağıdaki şekilde kodlanmalıdır (Resim 2.4).



Resim 2.4 Metin vurgusu kodlarının kullanım örneği

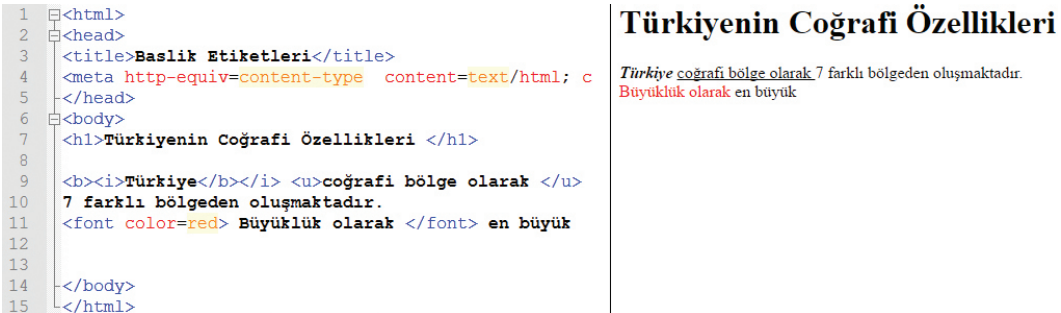
Resim 2.4'te görüleceği gibi, hangi özellik nereye eklenmek istenirse ilgili etiketi o bölümler arasında açmak `< >` ve kapatmak `</ >` yeterlidir. Bunun için koyu olması düşünülen Türkiye yazısının önüne `<b>` konulmuş bitim yerinde de `</b>` ile kapatılmıştır. Ayrıca Resim 2.2 örneğinde sıkça rastlanıldığı gibi bir metne hem kalın hem italik hem de yazı renginin farklı olması gibi birden fazla metin özelliği eklenebilir. Bu durumda her biri birbirinden bağımsızmış gibi düşünülmek üzere düzenleme yapılmalıdır. Bu nedenle Türkiye yazısının önünde hem `<b>` hem de `<i>` etiketleri birlikte kullanılmıştır.

Nadiren de olsa H_2O gibi bir formül yazmak için alt simge kullanılması gerekir ve html'de bu işlem için de kod kullanılmaktadır. `<small>` `</small>` arasına yazılacak karakterler alt simge, `<big>` `</big>` arasına yazılacak karakterler ise üst simge olarak görünecektir. H_2O örneği için aşağıdaki gibi kod yazılmalıdır.

Suyun formülü $H<small>2</small>O$ olarak ifade edilmektedir.

✓ font etiketi ile yazının rengi, büyüklüğü ve tipi değiştirilebilir. Bunun için sırası ile color, size ve face parametreleri kullanılır.

Metin özellikleri ile ilgili yapılabilecek diğer bir işlem ise yazı rengi, stili ve büyüklüğünün değiştirilmesidir. Yazının bu özelliklerinin değiştirmek için `<font>` etiketi kullanılır. Ayrıca değişikliğin etkisini sürdürmemesi için yine `</font>` etiketi ile ilgili yerde kapatma işlemi yapılmalıdır. Bu etiket hangi özellik verilmesi gerekiyorsa yanına o parametre veya parametreleri ile birlikte kullanılır. Üç parametre almaktadır. face parametresi ile yazı tipi (times new roman, arial vb.), size ile punto olarak yazı büyüklüğü, color ile de yazı rengi (red, blue vb.) değiştirilebilir. Örneğin Resim 2.2'de görülen kırmızı renkli "Büyüklik olarak" yazısı için Resim 2.5'teki şekilde `<font>` etiketi kullanılmalıdır.



Resim 2.5 Font etiketi ile yazının renginin değiştirilmesi

Yazıya verilmek istenen rengin İngilizce adı (ör: red, yellow vb.) veya hex kodu (ör: #d73b3e) internetten bulunabilir. Tablo 2.2’de temel renklere ait İngilizce ve hex kodları verilmiştir.

Tablo 2.2 Temel renkelere ait İngilizce ve Hex kodları

Renk	İngilizcesi	Hex Kodu	Renk	İngilizcesi	Hex Kodu	Renk	İngilizcesi	Hex kodu
Kırmızı	Red	#FF0000	Sarı	Yellow	#FFFF00	Mor	Purple	#800080
Siyah	Black	#000000	Yeşil	Green	#4E8975	Kahve	Brown	#804000
Beyaz	White	#FFFFFF	Mavi	Blue	#0925F4	Turkuaz	cyan	#00FFFF



Reklere ait hex kodları öğrenmek için <http://www.abecem.net/web/renk.html> bağlantısından yararlanabilirsiniz.

Renk değişikliği için kodların color parametresinden sonra sırası önemli olmaksızın atanması yeterlidir. Ayrıca yazının hem rengi hem büyüklüğü hem de stili aynı aşağıdaki kodla değiştirilebilir. Bu kodlardan önceki yazılar bu değişiklikten etkilenmeyecektir. `</font>` etiketinden sonraki stil de tekrar kullanmadan önce ayarlı olan yazım stili olarak devam edecektir. Atama işlemlerinde zorunlu olmamakla birlikte çift tırnak (" ") içinde değer gösterim kullanımı ile de karşılaşılabılır.

`<font color= "#d73b3e" size= "14" face= "tahoma">`
Etkilenecek yazı bu kısma yazılır `</font>`

Resim 2.2’de dikkati çeken bir diğer kullanım örneği ise Doğu Anadolu Bölgesi yazısının hem kırmızı renkli hem de italik ve koyu kullanımıdır. Bu durumda da `<font>` etiketi ile birlikte metinden önce `<b>` ve `<u>` ile açılmalı, metinden sonra

da `</font>`, `</i>` ve `</u>` ile kapatılmalıdır. Aksi takdirde kapatılmayan özellik devam edecektir.

...*coğrafi bölge* ** Doğu Anadolu Bölgesi*</i> *</u>*, en küçük...*

Resim 2.2’nin en alt iki satırında ise sadece arial yazı tipi ve 14 punto kullanımı söz konusudur. Bunun içinse aşağıdaki kodu bu metnin önünde kullanmak yeterli olacaktır.

`<font face= "arial" size= "14">` İstanbul yaklaşık olarak 14 ... `</font>`

✓ Çoğu html etiketinde olduğu gibi `<font>` etiketi `</font>` ile kapatılmazsa, etkisini sürdürecektir.

Paragrafların Biçimlendirilmesi

Paragrafların düzenlenmesi de yine kelime işlemci programlarından farklı bir şekilde olmaktadır. Örneğin html kodu yazarken on kez enter tuşuna basmak tarayıcıda on satır boşluk verilmesi anlamına gelmemektedir. Bunun yerine ilgili kodu kullanmak gerekir.

```

1 <html>
2 <head>
3 <title>Baslık Etiketleri</title>
4 <meta http-equiv=content-type content=text,
5 </head>
6 <body>
7
8   Günümüzde html bilgisi dinamik web sayfası
9
10
11
12
13
14   programlarının bilgisini de gerektirir.
15
16 </body>
17 </html>

```

Günümüzde html bilgisi dinamik web sayfası programlarının bilgisini de gerektirir.

Resim 2.6 Kod yazarken enter kullanımı ve tarayıcıdaki etkisi

Web tasarımlarında sıklıkla kullanılan alt satıra geçiş işlemi için ilgili metnin sonrasına `<br>` etiketi kullanılmalıdır. “Açılan etiket kapatılmalıdır” kuralı bu etiket için geçerli değildir. Yani açılan `<br>` etiketi `</br>` etiketi ile kapatılmaz. Ayrıca kaç satır boşluk verilecekse o kadar yazmak gerekir (Resim 2.7).

```

1 <html>
2 <head>
3 <title>Başlık Etiketleri</title>
4 <meta http-equiv=content-type content=text/html; charset=utf-8>
5 </head>
6 <body>
7
8   Günümüzde <br> html bilgisi dinamik web
9   sayfası <br> <br> programlarının bilgisini de gerektirir.
10
11 </body>
12 </html>

```

Günümüzde
html bilgisi dinamik web sayfası
programlarının bilgisini de gerektirir.

Resim 2.7 `<br>` kullanımı ile satır başı yapma örneği

Ayrıca yazılan metin bir paragraf ise bu metin `<p>` etiketi ile `</p>` etiketi arasına yazılmalıdır. Bu durumda ilerleyen kısımda anlatılacak stil ayarı ile ilgili paragrafa atanacak bir özellik, tüm paragraf için etkili olacaktır. Sonlandığı yerde gelen metin, satır başı olarak yazılır.

✓ ` ` etiketi kelimeler arasında boşluk vermek için kullanılırken, `<br>` etiketi alt satıra geçişte kullanılan bir html etiketidir.

`<p>` *Günümüz <i> dijital teknolojilerinden </i> en önemlilerinden biri de Tabletlerdir. Tabletler özellikle çocuklar için.... Günümüzde fiyatları da oldukça ekonomik hale gelmiştir. </p>*

Metinler için yapılabilecek bir diğer işlem ise hizalamadır. Bu amaçla metnin paragraf yani `<p>` etiketinin içerisinde `align` parametresi kullanılmalıdır. Dört değer almaktadır. Right sağa, left sola, justify iki yana, center ortaya hizalama için `align` parametresine atanmalıdır. Yazılan bir paragrafı iki yana yaslı yazmak isteyen bir kodlayıcı `align` parametresi ni aşağıdaki şekilde paragraf içerisinde yazmalıdır.

`<p align="justify">` *Günümüz <i> dijital teknolojilerinden </i> en önemlilerinden biri de Tabletlerdir. Tabletler özellikle çocuklar için.... Günümüzde fiyatları da oldukça ekonomik hale gelmiştir. </p>*

Normalde daha önce gördüğümüz bir başlık etiketi ile (örneğin `<h2> BAŞLIK </h2>`) başlık sola doğru hizalanmaktadır. Bu durumda başlığın da ortalanması istenebilir ve bu ortalama işlemi doğrudan `<center> <h2>BAŞLIK</h2></center>` şeklinde yazılarak gerçekleştirilebilir. Yani `<p>` etiketinin olmadığı yerde doğrudan `<center>` veya `<left>` parametreleri kullanılarak ta ortalama veya sola hizalama işlemleri yapılabilir. Bu ayrıca tablo, resim gibi nesnelerin ortalanmasında da sıkça kullanılan bir yöntemdir.

Listelerin Oluşturulması

Metinlerin sıralanması veya kategorilenmesi adına sıklıkla kullanılan seçeneklerden biri de listelerdir. Kelime işlemci programlarında kullanılan madde işaretleri ve madde numaraları html kodlamada listeler olarak ifade edilmektedir. Bu kapsamda madde numaraları sıralı liste, madde işaretleri ise sırasız liste olarak nitelendirilebilir. Sıralı listelerin (ordered list) ve sırasız listelerin (unordered list) oluşturulmasında İngilizce kelimelerinin baş harfleri olan `<ol>` ve `<ul>` etiketleri kullanılırken, maddelerin belirtilmesinde ise her ikisinde de `<li>` etiketi kullanılmakta ve nihai olarak hangi liste kullanılmışsa `</ol>` veya `</ul>` ile kapatılmaktadır. Bu kapsamda sıralı ve sırasız liste kullanımları aşağıdaki şekilde örneklendirilebilir (Resim 2.8).

✓ Html işaretleme dilinde sıralı liste (`<ol>`), sırasız liste (`<ul>`) ve kategorik liste (`<dl>`) olmak üzere üç farklı liste seçeneği bulunmaktadır.

6	<body>
7	
8	Sıralı Liste
9	
10	Fizik
11	Kimya
12	Biyoloji
13	
14	
15	Sırasız Liste
16	
17	Fizik
18	Kimya
19	Biyoloji
20	
21	
22	</body>

Sıralı Liste

1. Fizik
2. Kimya
3. Biyoloji

Sırasız Liste

- Fizik
- Kimya
- Biyoloji

Resim 2.8 Sıralı ve sırasız liste kullanım örnekleri

Bu durumda akla gelebilecek bir soru ise sıralı ve sırasız listelerde sunulan sıralama seçeneklerinin değiştirilmesidir. Resim 2.8'de görüleceği üzere standart olarak sıralı listeler için 1, 2, 3, sırasız listeler içinse • işareti kullanılmaktadır. Ancak ve etiketlerinin içerisine kullanılacak type parametresi ile bu seçenekler değiştirilebilmektedir. <ol type="a"> şeklinde bir atama ile gösterim değiştirilebilir. Bu durumda type için atanabilecek değerler A, a, 1, i, I şeklinde herhangi bir değer olabilir. Benzer şekilde sırasız listeler için de type parametresi ile değer atanabilmektedir. etiketinin type parametresi ise disk, circle ve square'dir. Resim 2.9'da sıralı ve sırasız listeler için html kullanım örnekleri ve tarayıcı çıktıları verilmiştir.



8	küçük harf ile sıralı Liste	küçük harf ile sıralı Liste
9	<ol type="a">	a. Fizik
10	Fizik	b. Kimya
11	Kimya	c. Biyoloji
12	Biyoloji	
13		
14		
15	romen rakami ile sıralı Liste	romen rakami ile sıralı Liste
16	<ol type="i">	i. Fizik
17	Fizik	ii. Kimya
18	Kimya	iii. Biyoloji
19	Biyoloji	
20		
8	Kare ile sırasız Liste	Kare ile sırasız Liste
9	<ul type="square">	▪ Fizik
10	Fizik	▪ Kimya
11	Kimya	▪ Biyoloji
12	Biyoloji	
13		
14		
15	İçi boş daire ile sırasız Liste	İçi boş daire ile sırasız Liste
16	<ul type="circle">	○ Fizik
17	Fizik	○ Kimya
18	Kimya	○ Biyoloji
19	Biyoloji	
20		

Resim 2.9 Sıralı ve sırasız liste kullanım örnekleri

8	İç içe liste kullanımı	İç içe liste kullanımı
9	<ol type="a">	a. Fizik
10	Fizik	o Biyo fizik
11		o Nükleer fizik
12	Biyo fizik	o Elektro fizik
13	Nükleer fizik	b. Kimya
14	Elektro fizik	c. Biyoloji
15		
16	Kimya	
17	Biyoloji	
18		

Resim 2.10 İç içe liste kullanım örneği

Bazı durumlarda ise iç içe listelerin kullanılması gerekir. Bu durumda etiketinin içerisinde yeniden veya etiketleri ile etiketini yeniden kullanmak yeterli olacaktır (Resim 2.10). Html kendi içerisinde bu kodların bir önceki listeye bağlı olduğunu anlamakta ve iç içe gösterimi sunmaktadır.

Son olarak listelerde kullanılabilecek bir liste türü de kategorik veya diğer ismi ile tanım listelerdir. Bu listelerde madde işareti ve numarası görünmez. Ancak kategorileme işlemi için ideal bir görünüm sunmaktadır (Resim 2.11). Kategorik listelerin oluşturulması için `<dl>`, kategorilerin herbirini oluşturmak için `<dt>`, kategoriler içerisinde yer alacak seçenekler için `<dd>` etiketleri kullanılır ve her bir açılan etiket yine `</dl>`, `</dt>` veya `</dd>` etiketleri ile kapatılmalıdır.

9		
10	<code><dl></code>	İlköğretim Bölümleri
11	<code><dt>İlköğretim Bölümleri</dt></code>	Sınıf öğretmenliği
12	<code><dd>Sınıf Öğretmenliği</dd></code>	Matematik
13	<code><dd>Matematik</dd></code>	Sosyal Bilgiler
14	<code><dd>Sosyal Bilgiler</dd></code>	Ortaöğretim Bölümleri
15		Tarih
16	<code><dt>Ortaöğretim Bölümleri</dt></code>	Kimya
17	<code><dd>Tarih</dd></code>	Biyoloji
18	<code><dd>Kimya</dd></code>	
19	<code><dd>Biyoloji</dd></code>	
20	<code></dl></code>	

Resim 2.11 Tanım (kategorik) liste kullanım örneği

Köprü ve Bağlantıların Kurulması

Köprü veya diğer adı ile link verme html sayfa tasarımı-
larında kullanılan en önemli özelliklerden biridir. Bu kap-
samda herhangi bir metin veya resme tıklanması durumun-
da yeni bir sayfanın açılması, sayfanın yeniden yüklenmesi,
resim, ses veya video dosyalarının açılması sağlanabilir. Bu
kapsamda kullanılması gereken etiket `<a href= "... ">...</a>`
etiketidir.

✓html'de <a> link verme etiketini kullanarak kendi menünüzü de oluşturabilirsiniz.

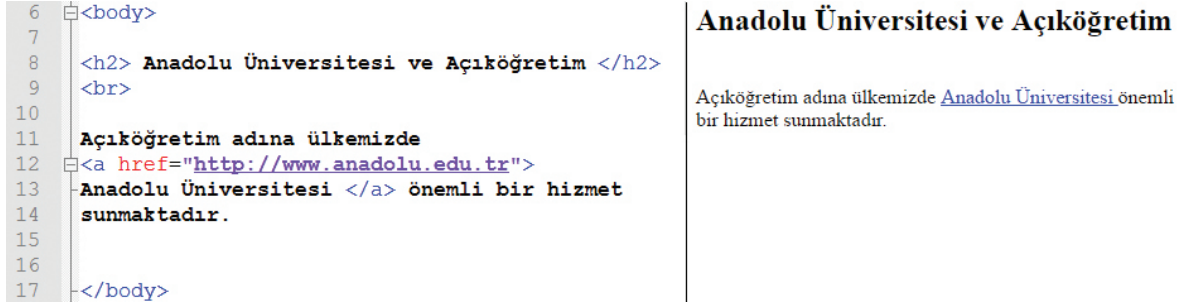
Genel kullanım şekli itibarı ile *href* parametresinden sonra açılması istenen sayfa veya resmin adresi eklenmelidir. Ayrıca *Ana Sayfaya Dön * şeklindeki kullanımda ekranda yazacak bilgi Ana Sayfaya Dön şeklinde altı çizili ve mavi bir yazıdır. Bu kullanımda tarayıcı otomatik olarak bu gösterimi kullanıcı adına sunar ve üzerine gelince fare işareti tıklanılabilirliğini gösterir. Köprüye tıklanması durumunda ise bağlantı olan *anasayfa.html* sayfasına geçiş işlemi gerçekleşir ve tarayıcı belirtilen sayfayı açar. Bu konudaki örnek uygulama Resim 2.12’de görülebilir.

[illegible]

Resim 2.12 Köprü-bağlantı kullanımına yönelik örnek kodlar

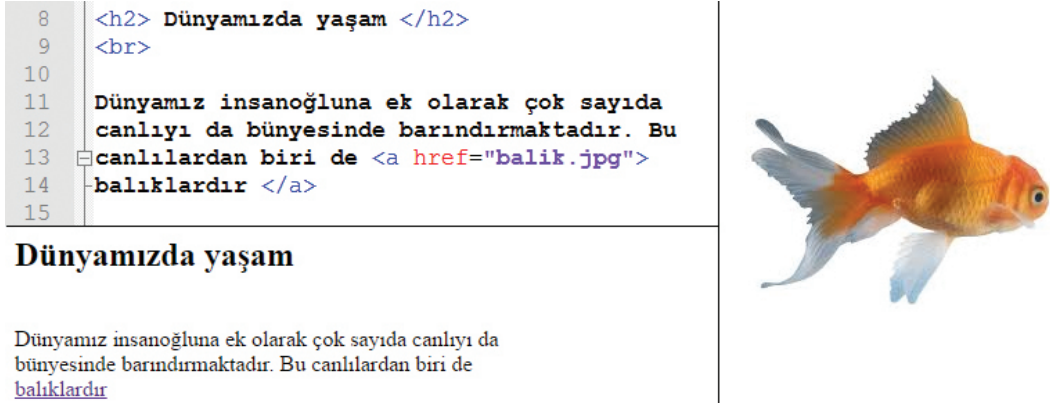
Resim 2.12’de ilk link 14. satırdaki kodlarla fotobilgi.html sayfasına, fotoğrafçılık kelimesi ile verilmiştir. Tarayıcı ekranında görüleceği üzere bu link mavi renkli ve link olduğu belirli olacak şekilde bir yazı olarak yansımaktadır. Eğer `</a>` etiketi yazılmazsa link sonrasındaki tüm kelimeleri de etkileyecektir. Bir başka deyişle link verme olayı sonlandırılmış olur. Ayrıca 18 ve 19 satırlarda da aynı site içerisinde iki

farklı sayfaya link verilmiştir. Bu şekilde site içerisinde kullanıcılar dolaşım sağlayabilir veya kendi menülerini kolaylıkla oluşturabilirler. Diğer yandan bu etiket ile mevcut bir web adresine de link verilebilir. Bu durumda ise protokolü de (ftp veya http) kullanmak gerekir. Resim 2.13'de Anadolu Üniversite'sinin web adresine (<http://www.anadolu.edu.tr>) link verilmiştir.



Resim 2.13 Mevcut bir web adresine link verme örneği

Link sadece web sayfalarına verilmekle sınırlı olmayıp, istenirse href parametresinden sonra resim, müzik vb. dosyalara da link verilebilir. Bu durumda ilgili medya ögesi açılacaktır (Resim 2.14).



Resim 2.14 Resme link verme kodları, tarayıcı ekranı ve tıklama sonuç ekranı

Burada özellikle vurgulanması gereken bir diğer konu ise hedefin konumudur. Web tasarımında genellikle düzen sağlama adına müzikler, resimler alt klasörler içerisinde saklanır. Dolayısı ile index.html gibi bir dosya kök dizinde yer alırken, zorunlu olmamakla birlikte düzeni sağlama açısından, resim dosyaları fotolar, pictures, ses dosyaları ise sesler, sounds, muzikler gibi klasörler içerisinde tutulur. Bu durumda sizin html dosyanızın bulunduğu kök dizinin içerisinde *resimler* isimli klasör dosyasında *balik.jpg* dosyanızın olduğunu varsayalım ve bu dosyaya link verelim. Bu senaryoda link komutu aşağıdaki şekilde olacaktır.

*Bu canlılardan biri de balıklardır. *

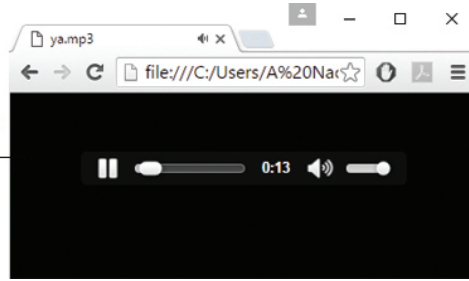
Benzer şekilde ses dosyaları için de kullanım söz konusudur. Bu durumda da tarayıcı da resim gelmesi yerine ses dosyasının oynatıldığı görülebilir (Resim 2.15). Örnekte ya.mp3 isimli dosyaya link verilmiştir. Eğer alt klasörde saklama söz konusu ise yukarıda olduğu gibi gösterim yapılmalıdır.

```

9  <br>
10
11  Ajda pekkan <a href="ya.mp3"> yakarım canını
12  </a> isimli parçası ile son yılların en çok
13  dinlenen sanatçıları arasında yerini almıştır.
14

```

Ajda pekkan [yakarım canını](#) isimli parçası ile son yılların en çok dinlenen sanatçıları arasında yerini almıştır.



Resim 2.15 Ses dosyasına link verme kodları, tarayıcı ekranı ve tıklama sonuç ekranı

Öğrenme Çıktısı



- 1 Web içeriğini oluşturan metinler üzerinde biçimsel düzenlemeler yapabilmek
- 2 Paragraf yapıları oluşturabilmek
- 3 Liste yapıları oluşturabilmek
- 4 Web sayfalarına bağlantılar ekleyebilmek

Araştır 1

Html tasarımına yeni başlayan birisi tasarımdan önce metin tabanlı hangi düzenlemeleri planlamalıdır?

İlişkilendir

Ziyaret ettiğiniz metin içerikli bir web sayfasının, sayfadaki tüm metin biçimlendirme etiketlerinin kaldırılması halinde nasıl görüneceğini kurgulayınız.

Anlat/Paylaş

Köprü ve bağlantıların Web sayfaları açısından önemini tartışınız.

METİNLERDE STİL KULLANIMI

Stil (style) web sayfalarda nesnelerin biçimlendirilmesi işlemidir. Html'de bütün sayfa için yapılması planlanan değişiklikler `<body>` etiketi içerisinde parametreler ekleyerek gerçekleştirilmektedir. Örneğin bir sayfada kullanılacak zemin rengi sarı olsun şeklinde bir ayarlama yapılmak istenirse bu durumda aşağıdaki şekilde `<body>` etiketi kullanılmalıdır. Hatta herhangi bir resim dosyası da arka plana `background` parametresi ile aşağıdaki şekilde eklenebilmektedir.

✓ Web tasarımı konusunda ilk başlayanlar için standart yazı tipi ve büyüklüğü, zemin rengi gibi ayarlamalar stillerle değiştirilebilir ve o site için bir şablon kolaylıkla oluşturulabilir.

```

<body bgcolor="#FFFF00" background="balik.jpg">

```

İçerik

```

</body>

```

Ancak bu durumda yapılacak değişik tüm sayfayı etkilemektedir. Diğer yandan sayfadaki metinler yukarıda bir kısmı ifade edilen html komutları (font, b, i vb.) ile biçimlendirilebilir. Bu sadece belirli bir metin grubu içerisinde geçerli olup, tüm değişikliklerde bu ayarlamaları yapmak oldukça zor ve yorucu bir iştir. Stil burada devreye girmektedir. Stil kullanımı site içerisinde farklı metinleri farklı şekillerde göstermek için önceden harici veya dahili kurallar tanımlama ve sonrasında bu kuralın ismine göre belirtilen yerde bu kuralın uygulanmasını sağlama işlemidir. CSS (Cascading Style Sheets – Stil Şablonları) olarak adlandırılan ve aslında web sayfalarını biçimlendirmek üzere yazılı bir dil olan stil

kullanımı metinler üzerinde etkin kullanılmasına karşın resim, tablo gibi her türlü nesnenin kullanımında da işlevseldir. Ünite metin üzerine odaklı olduğu için bu bölümde metne dayalı stil kullanımları üzerinde durulmuş, bu konudaki ilk ünite olması nedeniyle stiller hakkında detaylı bilgi verilmiştir.

Html içerisinde metin odaklı stil kullanımı birkaç yerde yapılabilir. Bunun için öncelikli olarak Türkçe olarak kuralların belirlenmesidir. Hangi şekilde yapıldığı amaca göre değişiklik gösterir. Kurallar için bir örnek vermek gerekirse bir paragraftaki metnin yazı tipi arial, 14 punto, rengi kırmızı ve iki yana hizalı bir paragraf olsun. Önemli olan kuralların belirlenmesidir. Sonrasında farklı şekillerde stiller yazılarla ilişkilendirilir. Stil kullanım türü amaca ve web tasarımıdaki deneyime göre farklılaşmaktadır. Stil kullanımı için *style* parametresi ile aşağıdaki şekilde değer atanır.

style = "özellik:değer;"

Metin kullanımında *style* parametresinin alabileceği bazı temel özellikler ve alabileceği örnek değerler Tablo 2.3'te verilmiştir.

Tablo 2.3 Stil parametresinin alabileceği bazı özellikler, işlevleri ve örnek değerler

Özellik	İşlevi	Alabileceği Örnek Değerler
color	Yazı rengi	red, blue, #FFAA00
font-family	Yazı stili	Verdana, arial
font-size	Yazı büyüklüğü	300%, x-small
text-align	Hizalama	Center, left, right, justify
text-indent	Paragraf girintisi	1em veya 50px gibi
background-color	Zemin rengi	red, blue, #FFAA00
text-decoration	Altı çizili olma	Underline, overline
text-transform	Büyük küçük harf dönüşümü	Uppercase, lowercase, capitalize

Stil kullanımı için *style* parametresinin kullanım yeri farklılaşabilmektedir; etiketin içerisinde kullanmak (inline), head bölümünde gruplayarak kullanmak (gömülü stil), dışarıdan ayrı bir stil dosyası kullanmak.

✓ Htmlde stil kullanımı etiket içerisinde, gömülü ve dışarıdan stil dosyası kullanımı olarak üç farklı şekilde gerçekleştirilebilir.

Etiketin İçerisinde Stil Kullanımı

Bu kullanım stilinde html'in temel etiketlerinin içerisinde nesnenin özelliklerinin belirlenmesidir. Örneğin bir `<h1>` başlık etiketi önceden anlatılmıştı. Başlığın kırmızı renkli ve ortalı kullanılması için *style* ile hem kırmızı renk hem de ortalama özelliklerinin tanımlanması aşağıdaki şekilde olmalıdır (Resim 2.16).

`<h1 style="color:red; text-align:center;">BAŞLIKTAKİ STİL KULLANIM ÖRNEĞİ</h1>`

```

1 <html>
2 <head>
3 <title>Baslik Etiketleri</title>
4 <meta http-equiv=content-type content=text/html;
5 </head>
6 <body>
7
8 <h1 style="color:red; text-align:center;">BAŞLIKA
9 STİL KULLANIM ÖRNEĞİ</h1>
10 <br><br>
11 Inline kullanımda doğrudan etiketin yanında style
12 parametresi kullanılır.
13 </body>
14 </html>
15

```

BAŞLIKA STİL KULLANIM ÖRNEĞİ

Inline kullanımda doğrudan etiketin yanında style parametresi kullanılır.

Resim 2.16 Inline kullanım ile stil tanımlama örneği

Inline kullanımı sadece `<h>` etiketlerinde değil, `<p>` ve `<a>` gibi çok sayıda etiket içerisinde de kolaylıkla kullanılabilir. Paragraf etiketi olan `<p>` de yapılan değişikliklerin tamamı `</p>` etiketine kadar geçerli olacaktır. Ayrıca bu paragraf içerisinde istenilen metin bölümlerinde yine `<font>` etiketi ile istenilen yazının stili istenildiği şekilde değiştirilebilir. Diğer yandan linklerin gösterim şekillerini değiştirmek isteyen bir tasarımcı da stil kullanabilir. Bu konuda aşağıdaki örnek verilebilir (Resim 2.17)

`<a href="anasayfa.html" style="background-color: #FFFF00; color: red;">`

```

2 <head>
3 <title>Baslik Etiketleri</title>
4 <meta http-equiv=content-type content=text/html; charset
5 </head>
6 <body>
7
8 <a href="anasayfa.html" style="background-color: #FFFF00;
9 color: red;"> Ana Sayfa</a>
10 <br>
11 Ana sayfaya dönmek için lütfen ilgili linki tıklayınız.
12 </body>
13 </html>

```

Ana Sayfa

Ana sayfaya dönmek için lütfen ilgili linki tıklayınız.

Resim 2.17 Inline kullanım ile linklerin özelliklerinin değiştirilmesi örneği

Yukarıdaki örnekte görüleceği üzere `<a>` etiketi ile tanımlananın dışında bir link görünümü bulunmadır. Beyaz zemin yerine sarı, mavi yazı yerine kırmızı renk ile link gösterimi söz konusudur. Tablo 2.3'teki özellikler ve değerleri kullanılarak farklı stil kullanımları da benzer şekilde yapılabilir.

Gömülü Stil Kullanımı

Etiketin içerisinde stil kullanılması durumunda stil kapatılmasına kadar o stil kullanımı geçerli olacaktır. Diğer yandan bir kullanıcı bir etiketin tüm özelliklerini sitesi için değiştirmek isteyebilir. Örneğin paragraf yani `<p>` etiketlerinde kullandığı yazının siyah yerine mor olmasını, yazı tipi olarak arial ve paragraf girintisi olarak 2 em'lik paragraf girintisine sahip olmasını ve bunun tüm `<p>` etiketi kullanılan yerlerde uygulanmasını isteyebilir. Bu durumda inline kullanım ile her `<p>` etiketinde teker teker yeniden tanımlamak yerine `<head>` `</head>` etiketleri arasında Resim 2.18'deki şekilde `<style>` `</style>` etiketleri arasına gömülü stil olarak tanımlamak yeterli olacak, tüm `<p>` etiketlerinde ilgili değişiklik otomatik olarak geçerli olacaktır (Resim 2.18). Aynı şekilde bir defada tüm etiket özellikleri için benzer tanımlamaların hepsi birlikte yapılabilir.

`<head>`

`<style type="text/css">`


```

p { text-indent: 2em; font-family:arial; color: red;>
h1 {color:green;}
i {color: #0925F4; font-family: verdana; text-decoration: underline; }
...
</style>
</head>

```

```

2 <head>
3 <title>Gömülü Stil Kullanımı</title>
4 <meta http-equiv=content-type content=text/html; charset=utf-8>
5
6 <style type="text/css">
7
8 p { text-indent: 2em; font-family:arial; color: red;" }
9 h1 {color:green}
10 </style>
11 </head>
12 <body>
13
14 <h1> Gömülü Stil Kullanımı</h1>
15 <p>Gömülü stil kullanımı çok önemli bir stil ayarıdır.
16 </p>
17 bu açıdan yeni başlayanlar için önemlidir.
18 </body>
19 </html>

```

Gömülü Stil Kullanımı

Gömülü stil kullanımı çok önemli bir stil ayarıdır.

bu açıdan yeni başlayanlar için önemlidir.

Resim 2.18 Gömülü stil kullanımı



Resim 2.18'deki gömülü stil kullanımı etiketlerin tamamı için geçerli olacaktır. Ancak bir tasarımcı bazı paragraflarda belirli kuralların, bazı paragraflarda ise diğer belirli kuralların geçerli olmasını isteyebilir. Bu durumda kurallar listesinin (class seçici veya id seçici) belirli bir isimle gömülü stil olarak tanımlanması gerekir. Programlama dillerinde özel fonksiyon tanımlama ve özel isim tanımlama ile gömülü stil kullanımı ile benzetilebilir. Bu kullanımın bir diğer yararı ise, ismi belirli olan bir stil grubunda yapılacak bir değişikliğin site içerisindeki tüm kullanıldığı yerlerde de etkisini göstermesidir.

Özel tanımlı stil kullanımına örnek olarak bir web sitesi için üç stil ismi ve bunlara ait kuralları belirleyelim. baslik1 isimli stil kırmızı, ortalı; prgrf1 ismindeki stil arial, iki yana yaslı; prgrf2 isimli stil ise times new roman, mavi renk ve sola hizalı özelliklerine sahip olsun.

Bu işlemin gerçekleştirilmesi için class ve id seçicilerin açıklanması gerekir. Class seçiciler ve id seçiciler temelde aynı işlevi yerine getirmektedir ve belirli özelliklerin bir isim altında gruplanmasını sağlamaktadırlar. Tanımlama sırasında isim önce belirlenirken, { } işaretleri arasında da kurallar belirlenir. Seçici olarak class kullanılması durumunda ismin önüne . işareti, id kullanılması durumunda ise # işareti konulur. Yukarıda ifade edilen baslik1, prgrf1 ve prgrf2 isimli stillerin class ve id seçiciler ile tanımlanması ve kullanımları ise aşağıdaki şekilde olacaktır.

✓ Stil tanımlamada class seçici ve id seçici olarak iki farklı seçici kullanılabilir ve bunlar aracılığı ile tasarımcı kendine özgü kurallar listesi belirleyerek kullanabilir.

<i>class seçici ile stil tanımlama</i>	<i>id seçici ile stil tanımlama</i>
<pre> <html> <head> <style type="text/css"> .baslik { color:red ; text-align:center; } .prgrf1 { text-align:justify; font-family:arial; } .prgrf2 { color: blue ; text-align:left; font- family:Times New Roman; } h1 {color:green} body {background-color:#666;} </style> </head> <body> <h1> seçiciler ve normal başlık </h1> <h1 class="baslik"> stil kullanımında seçiciler </ h1> <p class="prgrf1">Seçicilerin kullanımında eğer class seçici kullanılmışsa etiketin yanında class="" ile seçici adı bu kısma </p> <p class="prgrf2"> eğer seçici olarak id kullanılmışsa bu durumda da id="" ile bu kısma seçici tanımlanabilir. </p> </body> </html> </pre>	<pre> <html> <head> <style type="text/css"> #baslik { color:red ; text-align:center; } #prgrf1 { text-align:justify; font-family:arial; } #prgrf2 { color: blue ; text-align:left; font- family:Times New Roman; } h1 {color:green} body {background-color:#666;} </style> </head> <body> <h1> seçiciler ve normal başlık </h1> <h1 id="baslik"> stil kullanımında seçiciler </h1> <p id="prgrf1">Seçicilerin kullanımında eğer class seçici kullanılmışsa etiketin yanında class="" ile seçici adı bu kısma </p> <p id="prgrf2"> eğer seçici olarak id kullanılmışsa bu durumda da id="" ile bu kısma seçici tanımlanabilir. </p> </body> </html> </pre>

Bir sitede . ile class seçici, # ile id seçici birlikte kullanılabilir. Sadece paragraf içerisinde kullanımı karıştırılmamalıdır. Class seçici için etiketin içerisinde class= "" ile id seçici için id= "" ile tanımlama yapılmalıdır. Yukarıdaki her iki işaretleme etiketleri için de aynı ekran görüntüsü Resim 2.19'daki gibi olacaktır.

seçiciler ve normal başlık

stil kullanımında seçiciler

Seçicilerin kullanımında eğer class seçici kullanılmışsa etiketin yanında class="" ile seçici adı bu kısma

eğer seçici olarak id kullanılmışsa bu durumda da id="" ile bu kısma seçici tanımlanabilir.

Resim 2.19 Seçiciler ile stil kullanımı

Seçiciler ile metin kullanımı konusunda anlatılması gereken önemli bir etiket ise `<span>` `</span>` etiketleridir. Bu etiket bir yazı içerisinde istenilen aralığı belirlemek için kullanılır. `<p>` `</p>` etiketi paragrafı etki-

lerken, `<span>` `</span>` etiketleri daha özel seçimler (kelime, cümle veya paragraflar) seçiminde kullanılır ve yanına class veya id etiketi alabilmesi ile stil kullanımında önemli işleve sahiptir. Örnek olarak yukarıdaki cümlede sadece “eğer class seçici kullanılmışsa” kelimelerinin altı çizili, kırmızı renkli yapılacaksa vurgu isimli bir *class* veya *id* seçici aşağıdaki şekilde tanımlanabilir ve *span* ile kullanılabilir (Resim 2.20).

✓ `<span>` seçiciler olarak id ve class'ın metin içerisinde özel bir bölgeye tanımlanan stillerin uygulanabilmesinde kullanılan işlevsel bir html etiketidir.

```
<head>
<style type="text/css">
.vurgu { color:red; text-decoration:underline; }
.prgrfl { text-align:justify; font-family:arial; }
...
</head>
<body>
<h1> seçiciler ve normal başlık </h1>
<h1 class="baslik"> stil kullanımında seçiciler </h1>
<p class="prgrfl">Seçicilerin kullanımında <span>
class="vurgu">eğer class seçici
kullanılmışsa <span> etiketin yanında class=""
ile seçici adı bu kısma
</p>
....
```

seçiciler ve normal başlık

stil kullanımında seçiciler

Seçicilerin kullanımında eğer class seçici kullanılmışsa etiketin yanında class="" ile seçici adı bu kısma

eğer seçici id kullanılmışsa bu durumda da id="" ile bu kısma seçici tanımlanabilir.

Resim 2.20 Span ile metnin istenilen yeri için stil kullanımı

Dışarıdan CSS Dosyası Kullanma

“Kendime çok sayıda html dosyası olan bir web sitesi yapacağım ama ilk açıldığında zemin açık sarı renkli olsun, yazılar siyah değil bordo gelsin, büyük başlıklar koyu yeşil olsun ama web sitemdeki tüm sayfamda bunlar standart olarak gelsin, sürekli uğraşmayayım” şeklinde bir isteği olan tasarımcı dışarıdan css dosyası çağırarak isteklerine kavuşabilir. Bir sitenin birden fazla html’den meydana geldiği ve bunların aynı şablon üzerinden tasarlanacağı varsayılırsa her html için stil tanımı hem karmaşaya neden olacak hem de uzun satırlardan oluşan html dosyaları ortaya çıkacaktır. Bunun yerine html’in temel özellikleri body, p, h1,h2 ..h6 başlıkları, pre, ul, ol, li gibi tüm özellikleri tek bir dosyada bir kez yazılıp, css uzantılı olarak kaydedilebilir ve her html içerisinden tek bir satırla çağrılabilir. Bir css dosyası note defteri gibi basit programla da yazılabilir. Sadece dosya uzantısının .css olması gerekir. html ile dışarıdan css dosyasının çağrılması ise link ve @import etiketleri kullanılarak iki farklı şekilde gerçekleştirilmektedir. Örnek olarak ahmet.css isimli css dosyasının çağrılması aşağıdaki şekilde yapılmaktadır. Eğer bu css dosyası alt klasörler içerisinde saklanırsa bu durumda klasörle beraber adresi verilmelidir. (ör:href= “css/ahmet.css” gibi)

✓ Link ve @import dışarıdan CSS çağırma da kullanılan iki farklı etikettir. Aynı işlevi yapmakla birlikte aralarında küçük bir fark vardır.

Css dosyasının link etiketi ile çağırılması

```
<link rel="stylesheet" type="text/css" href="ahmet.css" />
```

Css dosyasının @import etiketi ile çağırılması

```
<style type="text/css">
@import "style.css";
</style>
```

Her iki etikette ahmet.css dosyası içerisindeki ayarlamaları aktif hâle getirmektedir. İmport etiketi tara-
yıcı tarafından öncelikli yorumlanmakta, bu etikete kadar olan html etiketleri de ahmet.css dosyasındaki
talimatlara göre biçimlendirilmektedir. Link komutunda ise kendisinden önceki komutlar harici css dos-
yasındaki ayarlardan etkilenmemekte, ancak sonrasındaki ayarlar bu komuta göre yapılandırılmaktadır.

ahmet.css dosyasının içeriği (örnek CSS dosyası)

```
BODY { background-color: #FFFFCC; Font-Family: Verdana, Arial, Helvetica, sans-serif; font-size:
120%;background-image : zemin.jpg; }
H1 { text-align: center; color : #ff0000;}
H3,H4,H5 { color : green ; }
P {color: Brown; text-align: justify; }
PRE{ font-size : 100%; color: Blue; font : bold;}
SELECT { Font-Size: 12px; color: blue;}
A:link {COLOR: "blue"; TEXT-DECORATION: none;}
A:visited { COLOR: "red"; TEXT-DECORATION: none; }
A:hover { COLOR: "green"; }
A:active { COLOR: "black"; }
...
```

Tüm ayarlamalar

```
2 <HEAD>
3 <meta http-equiv=content-type content=text/html; charset=utf-8>
4
5 <link rel="stylesheet" type="text/css" href="ahmet.css" />
6
7 <style type="text/css">
8 #baslik { color:red ; text-align:center;> }
9 #prgrf1{ text-align:justify; font-family:arial; }
10 #prgrf2{ color: blue ; text-align:left; font-family:Times New Roman; }
11 .vurgu { color:red ; text-decoration:underline; }
12 h1 {color:green}
13 </style>
14
15 </HEAD>
16 <BODY>
17 <h1> HTML'de Metin Kullanımı </h1>
18 <p> Genel olarak HTML'de metin işlemleri kelime
19 işlemci programlarından farklı mantıkla çalışmaktadır.
20 <span class="vurgu"> Ancak korkulacak önemli birşey olmadığını</span> da gördük. </p>
21
22
23 </BODY>
```



Resim 2.21 Link ile harici css dosyası kullanımı

Resim 2.21 incelendiğinde zeminin sarı renkli olması ayarını içeren bir komut olmadığı görülebilir. Ancak link etiketi ile dışarıdan çağrılan ahmet.css dosyası incelendiğinde body bölümünde background-color: #FFFFCC; etiketi ile zeminin sarı olarak ayarlandığı, paragraflardaki rengin ise Brown yani kahve rengi olduğu ve bu yüzden <p> parametresinin bundan etkilendiği gibi çok sayıda ayarın html sayfasına yansıdığı görülebilir. Yeni bir sayfa açılması durumunda da link veya @import ile dış css dosyasının yeni sayfaya eklenilmesi, ayarların o sayfada da geçerli olmasını sağlayacaktır.

Öğrenme Çıktısı



5 Metinlerde stil kullanımını açıklayabilme

Araştır 2

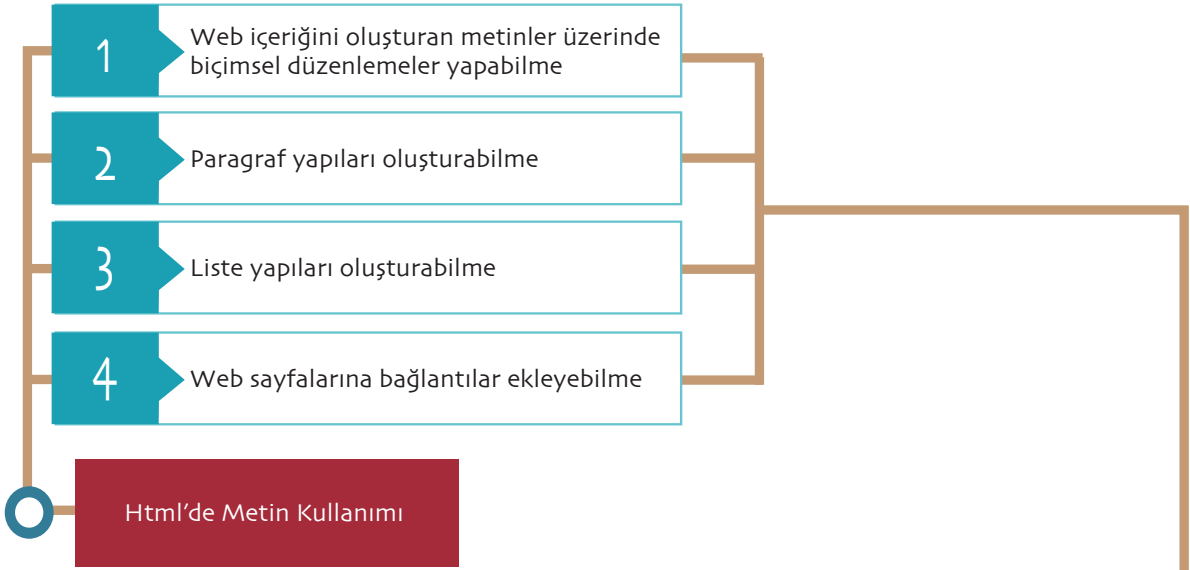
Web sitenizde kendi CSS stilinizi oluşturmanızın size ne gibi faydaları olacağını tartışınız.

İlişkilendir

Ziyaret ettiğiniz web sayfalarında görüntülenen metinler üzerinde ne gibi CSS düzenlemelerinin yapılmış olabileceğini tahmin etmeye çalışınız.

Anlat/Paylaş

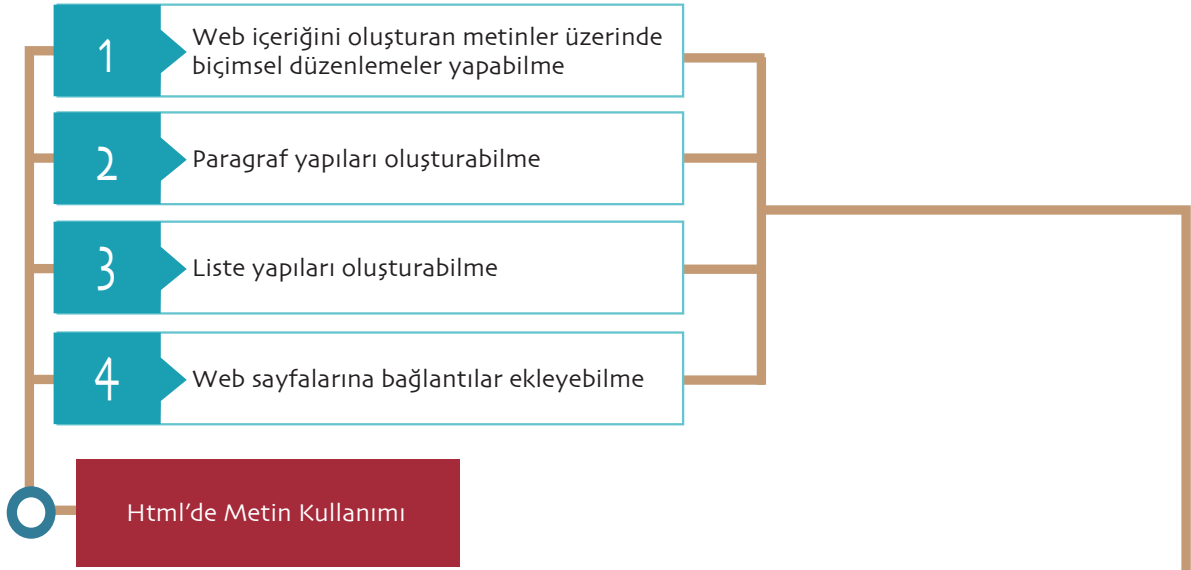
Stil kullanımının, metin biçimlendirme odaklı HTML etiketlerini hangi anlamlarda tamamlıyor olduğunu tartışınız.



- 1 Html kodu yazımı belirli kurallar çerçevesinde yapılmaktadır. Etiket olarak adlandırılan her bir kod yapısı < ve > işaretleri arasına yazılmaktadır. Ayrıca istisnai birkaç etiket haricinde her bir etiket <kod> işaretleri ile başlatılmakta, </kod> ile de sonlandırılmaktadır. Bir html sayfası temelde üç bölümden, bloktan oluşmaktadır. <html> </html> kodların başlayıp bittiğini vurgular ve ilk satır ile son satırı oluşturur. İkinci bölüm ise <head> </head> bloğu olup bu yapı arasında sitenin başlığı, dil ayarları gibi özellikleri ayarlanır. Üçüncü bölüm ise <body> </body> etiketleri arasına yazılmakla birlikte bu bölüm html'nin asıl tasarım bölümünü oluşturmaktadır.

Bu bölümde de farklı etiketler ile resim, metin, tablo gibi nesnelerin nerede ve nasıl görünmesi gerektiği belirtilir. Html bir işaretleme dilidir. Bu açıdan kullanılan etiketler ile öğelerin nerede, nasıl ve ne zaman görünmesi gerektiği gibi özellikler ayarlanmaktadır. Bu açıdan en çok kullanılan etiketler
 <p> <embed> <h> <u> <i> gibi etiketlerdir. Bu etiketler ile nasıl veya hangi nesnelerin nerede görünmesi gerektiği ayarlanabilmektedir.

- 2 Web sayfasında paragraf yazımı, kelime işlemci programlarının yazımındaki alışkanlıklar nedeniyle ilk başlangıç düzeyindeki bireyler için zor olmaktadır. Örneğin kelime işlemci programda enter tuşuna her basımda otomatik satır başı yapılabilir. Ancak html betik işaretleme dilinde bunun için bir kod kullanılmalı, bu istek tarayıcıya yorumlanmak üzere iletilmelidir. Bu açıdan bir alt satıra geçmek için
 etiketi kullanılmaktadır. Benzer şekilde bir paragraf oluşturmak ve bu paragraf kontrol altına alınmak istenirse ilgili metin <p> </p> etiketleri arasına yazılmalıdır. Böylece align parametresi gibi parametreler ile paragraf hizalama işlemleri yapılabilir. Ayrıca etiketi de paragraf içerisinde belirli bir bloğun seçilmesini sağlar. Bu özellikle CSS kullanımında ilgili paragrafın ilişkilendirilmesini sağlayan önemli bir etikettir.



3 Listeler kelime işlemci programlarında madde işareti ve madde numarası olarak adlandırılan ve kategorik gösterimlerde görsel bir düzen sağlayan önemli yazım seçenekleridir. Bu açıdan html'de de bu yazım seçenekleri özel etiketler ile kullanılabilir. Listelerin oluşturulmasında üç seçenek bulunmaktadır. `<ol>` madde numarasını yani 1,2,3 veya a,b,c veya i, ii, iii gibi gösterimleri oluşturmak için kullanılmaktadır. Bu seçenekler ise `<ol type="a">` gibi kullanımlar ile özelleştirilmektedir. Her bir seçenek, yani kategori ise `<li>` `</li>` etiketi arasına yazılmaktadır. Kare, daire ve çember şeklinde üç seçeneği bulunan sırasız listeler ise `<ul>` etiketi ile oluşturulmakta ve her bir seçenek ise yine `<li>` `</li>` etiketleri ile oluşturulmaktadır. `<ul type="circle">` gibi kullanım ile madde işaretinin circle ile çember, disk ile içi dolu daire, square ile de kare kullanılması sağlanabilir. Son olarak madde işaretinin bulunmadığı ancak gruplama işlemlerinin yapıldığı gösterimlerde tanımlama listeleri için `<dl>` `<dt>` ve `<dd>` etiketleri ile madde işareti veya numarası bulunmayan ama belirli grupları üst üste ilişkilendiren görsel gösterimler yapılabilir.

4 Köprü ve linkler sayfalar içerisinde geçişi veya sayfalar içerisindeki word belgesi, resim, video ya da müzik dosyalarına indirme özelliği kazandırabilmeyi sağlayan etiketleri kapsamaktadır. Bunun için kullanılan temel etiket ise `<a>` etiketidir. `<a href= ".link.">` link verilecek nesne `</a>` yapısında link verme işlemi yapılmaktadır. Burada href= "link" kısmı ile etkilenen yani tıklanması durumunda ne yapılması isteniyorsa o adres olarak gösterilir. Örneğin bir yazıya tıklanınca iletisim.html dosyası açılacak ise `<a href= "iletisim. html">` Biz kimiz `</a>` şeklinde etiket yazılmalıdır. Bu kısma "http://www.anadolu.edu.tr" gibi bir web sayfasının adresi yazılabileceği gibi, "belgeler/foto1.jpg" gibi bir dosya linki de verilebilir. Bu durumda ilgili dosya açılmaya, açılmazsa da indirilmeye çalışılır. Bu açıdan müzik, film gibi dosyalara da bu şekilde link vermek mümkündür.

5

Metinlerde stil kullanımını açıklayabilme

Metinlerde Stil Kullanımı

Metinlerde stil kullanımı metnin belirli bir bölümünde belirlenen kuralların geçerli olması işlemi olarak özetlenebilir. Bu işlem style parametresi ile yapılabileceği gibi, çok sık kullanılacak bir işlem ise kurallar listesi isimlendirilerek belirlenen yerlerde bu isim aracılığı ile kuralların yansıtılması şeklinde de yapılabilir. Örneğin `<h1 style= "color=blue">` şeklinde bir kullanım ile başlık metnin mavi olması sağlanabilir. Benzer şekilde `<a href= "iletişim.html" style= "color=red">` şeklindeki bir stil kullanımı ile de linkin mavi yerine kırmızı renkli görünmesi sağlanabilir.

Diğer kullanımlar ise gömülü ve seçiciler ile stil kullanımlarıdır. Gömülü kullanımda `<style type= "text/ css"> <style>` etiketleri arasında kurallar belirlenir ve tüm site için bu kurallar otomatik olarak yansıtılır. Örneğin `h1 {color:red;}` yapısı ise bu durumda site içerisindeki tüm h1 kullanılan etiketler kırmızı renkli olacaktır. Son olarak seçiciler kullanılabilir. Class seçici ve id seçici sırası ile . ve # işaretleri ile özel kuralların belirlenmesini sağlar. Belirlenen bu seçici isimleri etiketlerde kullanılarak kurallar bu kısımlara yansıtılabilir. Örneğin .ahmet class veya #ahmet id isimli kurallar tanımlanmış olsun. #ahmet {color:red; text-align:left;} şeklindeki kuralın istenilen yere yansıtılması `<h1 id= "ahmet">`, `<p id= "ahmet">`, `<span id= "ahmet">` şeklinde yapılmaktadır. Bu açıdan özellikle `<span> </span>` etiketleri ile css kullanım alanı belirlenebilmektedir. Span metinlerde stil kullanımı açısından önemli bir etikettir.

1 Aşağıdakilerden hangisi htmlde her etiketin yazımında mutlaka kullanılması gereken işarettir?

- A. = işareti
- B. " işareti
- C. < işareti
- D. ! işareti
- E. @ işareti

2 etiketi metin üzerinde aşağıdaki işlevlerden hangisini yerine getirmektedir?

- A. Alt satıra geçiş
- B. Boşluk verme
- C. Yazıyı işaretlem
- D. Koyu yapma
- E. Metni gizleme

3 Html'de kelimeler arasında boşluk sayısını artırmak isteyen bir kullanıcı hangi özel karakter setini kullanmalıdır?

- A. bsp;
- B. <
- C. >
- D. &
- E. "

4 Türkçe karakter sorunu yaşayan bir kullanıcı <meta> etiketinde aşağıdaki hangi parametreyi kullanmalıdır?

- A. http-equiv
- B. content
- C. text/style
- D. <style>
- E. Charset

5 Sayfa içerisinde kullanılacak başlık türünü ve dolayısı ile büyüklüğünü belirlemek için kullanılan etiket aşağıdakilerden hangisidir?

- A.

- B. <a>
- C. <p>
- D.
- E. <h>

6 Aşağıdakilerden hangisinde yazının tipini (arial, verdana vb.) değiştirmek isteyen bir kullanıcının kullanması gereken etiket ve parametre birlikte ve doğru olarak verilmiştir?

- A. CSS – textcolor
- B. font – color
- C. meta – backgroundcolor
- D. font – face
- E. body – style

7 etiketi aşağıdaki amaçlardan hangisi için kullanılır?

- A. Köprü oluşturmak
- B. Oluşturulan köprüye link vermek
- C. Harici CSS dosyasını bağlamak
- D. Stil dosyası eklemek
- E. Listelere seçenek eklemek

8 Web sayfasından başka bir web sayfasına geçiş sağlamak isteyen bir tasarımcı hangi html etiketini kullanmalıdır?

- A. <a href=
- B. <font link=
- C. <p link=
- D. <link href=
- E. <body link=

9 Metinlerde stil kullanımında text-indent aşağıdaki işlevlerden hangisini yerine getirmektedir?

- A. Altı çizili olma
- B. Yazı stili
- C. Paragraf girintisi
- D. Büyük küçük harf dönüşümü
- E. Hizalama

10 Bir html içerisinde oluşturacağı tüm paragraflardaki yazının kırmızı olmasını isteyen bir tasarımcı aşağıdaki metin tabanlı css stillerinden hangisini kullanmalıdır?

- A. H1 {color:red;}
- B. P {color: red;}
- C. A:link {color:red}
- D. Pre {color: red;}
- E. Select {color: red;}

1. C

Yanıtınız yanlış ise “Karakter Seti Özelliklerinin Belirlenmesi” konusunu yeniden gözden geçiriniz.

2. D

Yanıtınız yanlış ise “Html’de Metin Kullanımı” konusunu yeniden gözden geçiriniz.

3. A

Yanıtınız yanlış ise “Html’de Metin Kullanımı” konusunu yeniden gözden geçiriniz.

4. E

Yanıtınız yanlış ise “Karakter Seti Özelliklerinin Belirlenmesi” konusunu yeniden gözden geçiriniz.

5. E

Yanıtınız yanlış ise “Metin Özelliklerinin Düzenlenmesi” konusunu yeniden gözden geçiriniz.

6. D

Yanıtınız yanlış ise “Metin Özelliklerinin Düzenlenmesi” konusunu yeniden gözden geçiriniz.

7. E

Yanıtınız yanlış ise “Listelerin Oluşturulması” konusunu yeniden gözden geçiriniz.

8. A

Yanıtınız yanlış ise “Köprü ve Bağlantıların Kurulması” konusunu yeniden gözden geçiriniz.

9. C

Yanıtınız yanlış ise “Metinlerde Stil Kullanımı” konusunu yeniden gözden geçiriniz.

10. B

Yanıtınız yanlış ise “Metinlerde Stil Kullanımı” konusunu yeniden gözden geçiriniz.

2

Araştır Yanıt Anahtarı

Araştır 1

Html yazımında ilgili kuralların oluşturulması çok kolaydır. İlk olarak paragraf, başlık ve body etiketinde hangi değerlerin ele alınacağı stil ile tanımlanmalı, sayfaya genel bir düzenleme verilmelidir. Metinler ile ilgili düzenlemelerde de önemli olan bir taslağın oluşmuş olması, kullanıcının nasıl bir tasarım istediği bilincinde olmasıdır. Kelime işlemci gibi bir programla yapılacak herhangi bir yazım şeklinin önceden belirlenmesi veya oluşturulması, html etiketlerinin nerede ve nasıl kullanılacağı konusunda pratiklik sağlayacaktır. Web tasarımında tasarımcıları zorlayan çoğu zaman içeriğin ve tasarımın eş zamanlı olarak planlanmasıdır. İçeriğin ve kabaca tasarımın önceden belirlenmesi metinlerin tasarımını da çok kolay hale getirecektir.

Araştır 2

Öncelikli olarak web siteleri çoğunlukla tek bir sayfadan oluşmamaktadır. Bu açıdan bir biri ile bağlantılı ve aynı tasarıma sahip sayfalarda içerik değişmektedir. Stiller bu açıdan siteyi oluşturan sayfalar arasında belirli bir standardı sağlayacaktır. Diğer yandan çok fazla sayıda site tasarımı yapan bir kullanıcı ise bu durumda oluşturulacak CSS dosyası birden fazla sitede kullanabilecek anlamına gelmektedir. Bu ise hem pratiklik hem de tasarım sürecinde hız kazandıracaktır.

Kaynakça

- Bayrakçı, E.V. (2009). Html ile CSS Kullanım Biçimleri <http://www.sanalkurs.net/html-ile-css-kullanim-bicimleri-2917.html> Erişim Tarihi 28 Haziran 2016.
- Ece, A. (2016). ABECE renk kodları: renk kodları (html color codes), renklerin rgb eşdeğerleri, heksadesimal kodları, otomatik renk seçici. <http://www.abecem.net/web/renk.html> Erişim Tarihi 23 Haziran 2016.
- Etudio (2016). CSS Nedir? <http://www.baskent.edu.tr/~tkaracay/etudio/ders/internet/css/index.html> Erişim Tarihi 28 Haziran 2016.

Bölüm 3

Çokluortam Öğelerinin Sunumu

Öğrenme Çıktıları

- 1 Resimlerin Görüntülenmesi
1 Web sayfasına resim ekleyebilme
- 2 Resim Haritalarının Oluşturulması
2 Resim haritaları aracılığıyla link verebilme
- 3 Html5'in Getirdiği Çokluortam Özellikleri
3 Web sayfasına video ekleyebilme
4 Web sayfasına ses ekleyebilme
- 4 Harici Uygulama ve İçeriklerin Entegrasyonu
5 Harici içerikleri web sayfasına ekleyebilme
- 5 Çokluortam Öğelerinde Stil Kullanımı
6 Çokluortam nesneleri için CSS kullanabilme

Anahtar Sözcükler: • Çokluortam • Resim • Video • Flash Animasyonu • Harici Video
• Google Konumu • Resim Haritaları • Pdf Dosyaları



GİRİŞ

Html, özellikle 2000'li yılların başında teknolojik araç-gereçlerin gelişmesi ile hızlı bir gelişim göstermiş, html ifadesi içerisinde yer alan Hyper Text'in dışına çıkarak yazının yanında ses, video gibi öğeleri de kapsar hâle gelmiştir. Çokluortam, dijital ortamlarda etkileşimli yazının yanında resim, video, animasyonlar, ses gibi çok sayıda nesnenin birlikte kullanımını ifade eden bir kavramdır (Resim 3.1).



Resim 3.1 Çoklu ortam öğeleri

Kaynak: http://www.grandrapidschurchofchrist.org/?page_id=179

Günümüzde web siteleri çoğunlukla Çokluortam öğelerini tercih etmektedirler. Metin tabanlı web sitelerine göre daha eğlenceli olması ve ilgi çekmesi, çok sayıda duyu organına hitap etmesi, sözel anlatıma oranla daha kolay anlatım fırsatı sunması gibi nedenler web tasarımında çokluortam öğelerinin kullanımını önemli kılmaktadır. Bu ünite de ise teknik anlamda web sayfasında çokluortam öğelerinin nasıl siteye dâhil edileceği ele alınmıştır. Ayrıca günümüz tarayıcıları tarafından desteklenmesi ve çokluortam öğelerinin kullanımını kolaylaştırması, sahip olduğu işlevselliği artırması nedeniyle html5 komutlarından da yararlanılmıştır.

✓ Html5 video ve ses başta olmak üzere çok sayıda denetimi ile çokluortam öğelerinin kullanımını kolaylaştıran yeni bir html sürümüdür.

RESİMLERİN GÖRÜNTÜLENMESİ

Çokluortam öğeleri olarak metinden sonra en sık kullanılan öğelerin başında resim gelmektedir. Görsellik olarak gerek içerik sunumunu kolaylaştırması, gerek kolay bir şekilde kullanılabilmesi resimlerin web sayfalarında yer bulmasını sağlamıştır. Html'de resimlerin gösterimi için img etiketi aşağıdaki şekilde kullanılmaktadır.

```

```

Resmin bulunduğu konum, oluşturulan html sayfasının bulunduğu konum esas alınarak ifade edilir. Örneğin masaüstünde *web sayfam* isimli bir klasör olduğunu varsayalım. Bunun içerisinde de *iletisim.html* isimli bir sayfayı tasarladığımızı düşünelim. Bu sayfada resim göstermek istiyorsak (ör: *foto.jpg*) resmin *iletisim.html* dosyası ile aynı yerde yani *web sayfam* isimli klasörün içerisinde olması gerekir. Bu durumda *src="foto.jpg"* olarak yazılması gerekir. Ancak genellikle web sitelerinde karmaşayı önlemek adına ve tercihen çokluortam öğeleri ayrı klasörler altında tutulur. Bu açıdan web sayfam klasörünün içerisinde *resimler*, *sesler*, *videolar* gibi klasörler oluşturulur ve *foto.jpg* dosyası *resimler* klasörünün içerisinde saklanır. Bu durumda resim dosyasının konumu için yine *iletisim.html* dosyası dikkate alınarak alt klasör konumu da belirtilir. Sonuç olarak dosya konumu için *src="resimler/foto.jpg"* şeklinde bir tanımlama yapılmalıdır. Width ve Height ise kullanımı zorunlu olmamakla birlikte resmin sayfa içerisindeki büyüklüğünü *pixel* olarak ifade etmektedir. Bir tasarımcı resmin boyutlarını program aracılığı ile hiç değiştirmeden sayfasında bu iki parametreyi kullanarak istediği ebatlarda gösterebilir.

✓ Resimlerin gösterimi width ve height ile istenilen ebatta yapılabilmesine karşın, özellikle mobil uygulamalar düşünülerek profesyonel tasarımda bu tür ayarlamalar için photoshop, paint gibi programlar aracılığı ile orijinal dosya boyutu küçültülmelidir.

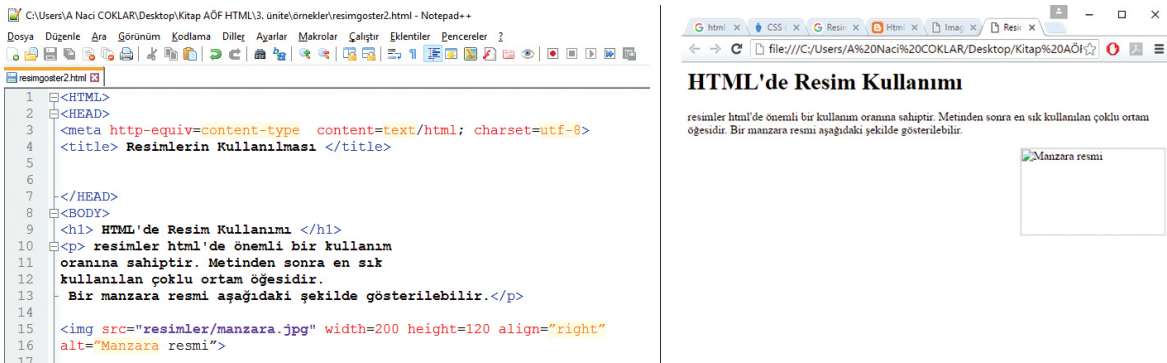
Bu kapsamda manzara.jpg isimli bir resmin gösterim örneği ve ekran görüntüsü Resim 3.2’de verilmiştir. Resim 3.2’de ilk örnekte resim orijinal boyutu ve html ile aynı klasörde (``) gösterilecek, ikinci örnekte ise resimler alt klasörünün içerisinde ve genişlik (width) olarak 200, yükseklik (height) olarak 120 piksel boyutlandırması ile (``) gösterilecektir.



Resim 3.2 Resimlerin html kodları ile kullanımı

Resim etiketi olan `<img>` etiketinin alabileceği birkaç önemli parametresi daha bulunmaktadır. Bunlara örnek olarak border, alt, align ve usemap parametreleri verilebilir. Eğer resme çerçeve verilecekse border parametresine değer atanmalıdır (ör: border= "4"). Atanan değer 0 olursa çizgi olmayacağı anlamına geleceği gibi, bu değer ne kadar fazla olursa çizginin kalınlığı da o kadar artacaktır. Alt resmin açılmaması durumunda veya üzerine gelinmesi durumunda görünmesi istenilen metnin görüntülenmesi, align ise resmin sayfaya göre hizalanması (sağa – right ya da sola – left) amacıyla kullanılır. Örneğin resim yüklenememişse "Manzara resmi" yazsın ve resim sağa hizalı olarak görünsün şeklindeki bir img etiketi aşağıdaki şekilde olmalıdır ve sayfanın kodu ile ekran görüntüsü Resim 3.3’teki gibi olacaktır.

``



Resim 3.3 img etiketinin alt ve align ile kullanımı

`<img>` etiketinin parametre olarak `align` parametresi sadece sol - left ve sağ - right değerlerini almaktadır. Ortalama işlemi için bu parametre çalışmamaktadır. Ancak resmi ortalamak isteyen bir tasarımcı `img` parametresinin önünde kullanacağı `<center>` ve sonunda kullanacağı `</center>` etiketleri ile bu sorunu aşağıdaki şekilde çözebilir.

```
<center>  </center>
```

`<img>` etiketinin alacağı bir diğer parametre ise `usemap` olup, resim haritalarının oluşturulmasında kullanılmaktadır.

✓ Hangi çokluortam özelliği olduğuna bakılmaksızın `<center>` etiketi ile istenilen tüm öğelerin sayfayı ortalaması sağlanabilir.

Öğrenme Çıktısı



1 Web sayfasına resim ekleyebilme

Araştır 1

`<div>` etiketinin kullanımı konusunu çokluortam öğelerini temel alarak tartışınız.

İlişkilendir

Web sayfalarında gezinirken, özellikle internet bağlantı hızının düşük olduğu durumlarda resimlerin metinlere göre daha geç görünülüyor olmasının sebeplerini yorumlayınız.

Anlat/Paylaş

Web tasarımında resimlerin hangi amaçlarla kullanılabileceğini anlatınız.

RESİM HARİTALARININ OLUŞTURULMASI

Resim haritaları (image maps) resimlerin üzerindeki bir veya daha fazla noktaya link verme amacı ile kullanılmaktadır. Örnek olarak yedi coğrafi bölgeden oluşan Türkiye Coğrafi Bölgesi haritası göz önünde bulundurulur ve *bolgeler.html* sayfasında kullanıldığı varsayılırsa, her bir bölge için resim üzerinde resim haritası ile belirli alan link olarak oluşturulabilir ve yedi bölgeden herhangi birini oluşturan alana tıklanması durumunda kendi bölgesine ait html (örneğin *marmara.html*, *akdeniz.html* vb.) dosyasının açılması sağlanabilir. Resim haritalarının oluşturulmasında `<img>` etiketi `usemap` parametresi ile kullanılmalıdır. Bu kullanımda ilişkilendirme değeri kullanılırken önüne # işareti unutulmamalıdır. Sonrasında ise `<map>` etiketi ile resim haritalarının alanları ve link işlemleri atanır.

```

```

```
<map name="bolgeharita">
```

```
<area shape="rect" coords="0,0,200,280" href="marmara.html">
```

```
<area shape="rect" coords="205,0,700,280" href="karadeniz.html">
```

```
<area shape="rect" coords="0,280,200,450" href="ege.html">
```

```
</map>.
```

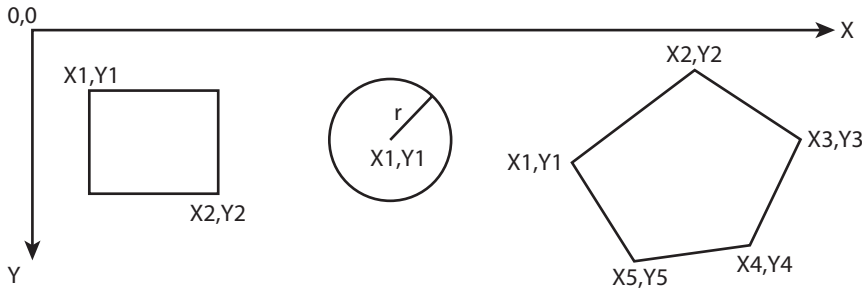

✓ Resim haritaları ile resmin istenilen yerine link verme işlemi yapılabilir. Verilecek link resim, zip, Word belgesi gibi bir doküman, tasarlanan başka bir sayfa veya başka bir sitenin web adresi olabilir.

Yukarıdaki html kodları incelendiğinde ekrana *bolgeler.png* isimli bir resim aktarılacağı ve bununla *bolgeharita* isimli bir *usemap* ile harita oluşumuna açıldığı görülebilir. Bu parametre ile `<map>` `</map>` etiketleri kullanılabilir hâle gelmiştir. `<map>` etiketinde kullanılan name parametresi `<img>` etiketi ile ekrana getirilen resim arasında bağlantı kurmaktadır. Sonrasında ise `<area>` etiketi resim üzerinde kaç farklı bölgeye link oluşturulacaksa o kadar

kullanılmalı ve her biri için ayrı tanımlanmalıdır. `<area>` etiketinde kullanılacak parametreler aşağıdaki şekilde kısaca açıklanabilir. Bir resim üzerinde tek bir seçim alanı olacak ise sadece bir `<area>` etiketi kullanımı yeterli olacaktır.

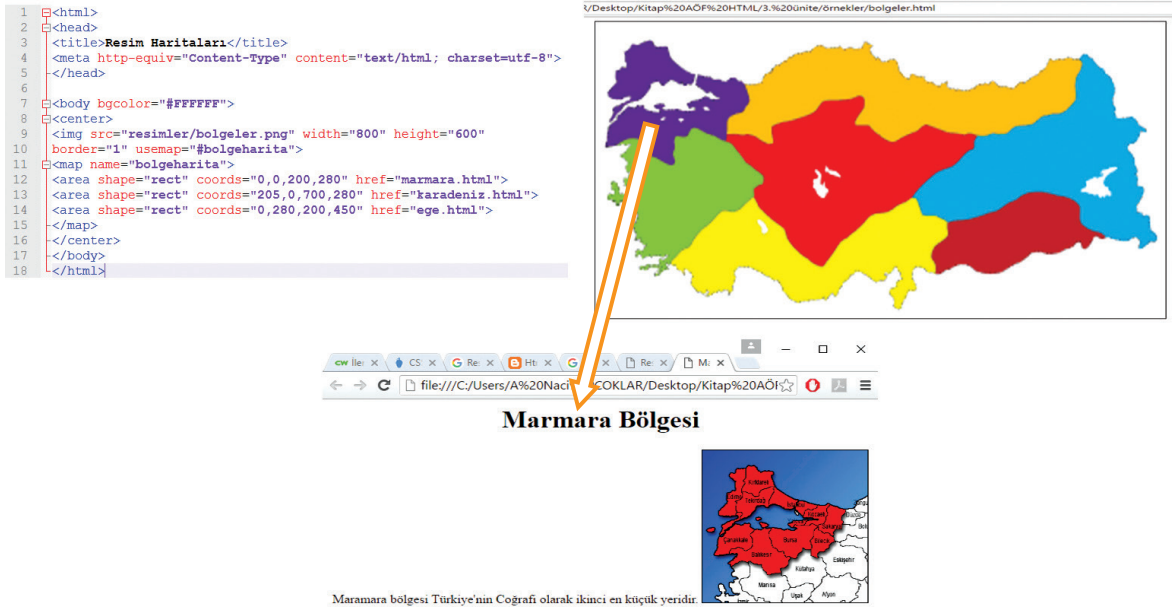
Shape parametresi, resim üzerinde işaretlenecek alan veya alanların şekillerinin belirlenmesi için kullanılır ve rect (dikdörtgen), circle (daire), poly (çok kenarlı) olmak üzere üç farklı değer alabilmektedir. *Coords parametresi* ise resim üzerinde işaretlenecek alanın genişliğini pixel olarak belirlemek için kullanılmakla beraber kullanım şekli tercih edilen shape seçeneğine göre değişmektedir. shape="rect" yani dikdörtgen alan oluşturmak için 2 nokta tanımlanmalıdır. Bunlar sol üst ve sağ alt noktadır (coords=X1,Y1,X2,Y2) ve tanımlama `<area shape="rect" coords="0,0,30,20">` şeklinde olacaktır. Büyüklük için dairesel yani shape="circle" şeklinde bir alan belirlenecek olursa bu durumda coords parametresinde (coords=X1,Y1,r) koordinatlar dairenin merkezini ve yarıçapını gösterir tanımlama ise `<area shape="rect" coords="50,50,25">` şeklinde olacaktır. Serbest olarak bir alan belirlenecekse bu durumda X ve Y koordinatları istenilen sayıda kullanılabilir (coords=X1,Y1,...,Xn,Yn) ve en son seçimdeki koordinatlar ile ilk koordinatlar birleştirilecektir, tanımlama ise `<area shape="poly" coords="0,0,25,50,48,78,75,100">` şeklindedir (Şekil 3.1).

✓ Area ile resmin üzerinde bölgeler oluşturulabilir. Rect, circle ve poly olmak üzere üç tür shape seçeneği vardır ve seçilen shape seçeneğine göre coords parametresinin kullanımı da farklılaşmaktadır.



Şekil 3.1 image map oluşturmada coords kullanım türleri

Son olarak area etiketi ile oluşturulan her bir seçenek yani link verilmiş bölgeye tıklanması durumunda ne olacağını belirlemek için href parametresi kullanılır ve link verilir. href parametresine atanacak değer ile başka bir html dosyasına yönlendirme yapılabileceği gibi, bir Word dosyasına link verme, müzik dosyasına link verme veya zip dosyası gibi bir dosyaya link verme işlemi de yapılabilir. Tıklanması durumunda da ilgili doküman belgesinin indirme işlemini yerine getirecektir. Resim 3.4'te Türkiye'nin Coğrafi Bölgeleri için *bolgeler.html* dosyasına *bolgeler.png* resim dosyası eklenmiş, bölgelerden Marmara, Karadeniz ve Ege bölgelerine image map ile rect şeklinde area oluşturularak, link verilmiştir. Kodlar ve ekran görüntüsü aşağıdaki gibidir.



Resim 3.4 Image map oluşturma örneği



Öğrenme Çıktısı

2 Resim haritaları aracılığıyla link verebilme

Araştır 2

HTML'nin resim haritalandırma özelliği olmasaydı, resim haritalamaya benzer bir uygulamayı nasıl gerçekleştirebileceğinizi kurgulayınız.

İlişkilendir

Bir resim üzerinden sizi farklı noktalara yönlendiren web gezinimi deneyimlerinizi hatırlamaya çalışınız.

Anlat/Paylaş

Resim haritalarının kullanımının web sayfasının işleyiş performansına ne gibi etkilerinin olabileceğini anlatınız.

HTML5'İN GETİRDİĞİ ÇOKLUORTAM ÖZELLİKLERİ

Web sitesinde kullanılabilecek diğer önemli çokluortam öğeleri ise ses ve videodur. Normalde html ile bu iki öğenin de kullanımı gerçekleştirilebilmektedir. Ancak web erişim aracı olarak mobil teknolojilerin önem kazanması html5 teknolojisinin geliştirilmesini sağlamış, getirdiği hazır komutlar ile bu iki öğenin kullanımı çok kolay hâle gelmiştir. Html'in zamanla sürümleri çıktığını, 1999 yılında html 4'ün geliştirildiğini, şu an kullanılmakta olan sürümün 2004 yılında geliştirildiğini ve ajax ve flash desteği sunan xhtml 2.0 sürümü olduğunu ve html5'in ise en son çıkan ve kendine yönelik standartları ve yenilikleri de beraberinde getirdiğini vurgulamak gerekir. Günümüzde hemen hemen tüm popüler tarayıcılar (firefox, chrome, yandex, internet explorer vb.) hem html4 standartlarında, hem de html5 standartlarında yazılmış etiketleri derleyebilmektedir. Öncelikli olarak html5'in biraz açılması yararlı olacaktır.

Html5 yeni sürüm olarak eski sürümlere oranla yenilikler getirmekte, buna karşın önceki sürümlerle yazılan kodları da kapsamaktadır. Html5'in getirdiği yenilikler incelendiğinde hata kontrollerinin daha gelişmiş olduğu, daha işlevsel parametreler ile yeni etiketlerin kolaylık sağlaması, yeni uygulamaları desteklemesi, ses ve video için destek sunması, veri girişi için kullanılan input type'larına özellikler eklenmesi, canvas özelliği ile iki boyutlu çizimlerin yapılabilmesi sağladığı yenilikler olarak ifade edilebilir. Özetle vurgulamak gerekirse, daha önceki sürümlerde bulunmayan yeni etiketler eklenmiş veya varolan etiketler parametreler ile zenginleştirilmiş, böylece daha zengin bir tasarım olanağı sunulmuştur. Bu yeniliklerden bir kısmı da video ve ses öğelerinde ön plana çıkmaktadır.

Html5 ile Videoların Eklenmesi

Öncelikli olarak bu işlemin daha önceki sürümlerde `<embed>` etiketi ile kullanıldığını belirtmek gerekir. Ancak html5 sahip olduğu parametre sayısı ile video ekleme ve kontrol işlemlerini çok daha pratik ve işlevsel hâle getirmiştir. Html5 ile video eklemek için kullanılan etiket `<video>` etiketi olup `</video>` etiketi ile kapatılması gerekir. Basit düzeyde video eklemek için sadece `<video>` etiketinin içerisinde `src` parametresi ile hedef belirtmek yeterli olacaktır. Ayrıca resim ekleme başlığında anlatıldığı üzere üzerinde çalışılan html'ye göre konum belirtilir. Bir başka ifade ile eğer eklenmek istenen video alt klasör veya klasörler içerisinde ise bu klasör isimleri de belirtilerek adres işaretlenmelidir. Video türleri bağlamında önemli tarayıcıların hangi sürümlerinden itibaren video ve bir sonraki konu olan etiketinin desteklendiği aşağıdaki tablodan incelenebilir (Tablo 3.1). Günümüzde kullanılan çoğu sürüm, `<video>` ve `<audio>` etiketlerinin kullanımını desteklemektedir.

Tablo 3.1 Önemli tarayıcıların video ve audio etiketlerini desteklenmeye başladığı sürümleri

Tarayıcılar	Chrome	Explorer	Firefox	Safari	Opera
Desteklenen Sürümleri	4.0	9.0	3.5	4.0	10.5

Sayfaya video eklemek için bilinmesi gereken bir diğer konu ise videonun dosya türü ve tipidir. Günümüzde web sayfası tasarımında kullanılan dosya türleri ve mime tipleri aşağıdaki gibidir (Tablo 3.2).

Tablo 3.2 Video etiketinin desteklediği video dosya uzantıları ve mime tipleri

Dosya Uzantısı	Mime Tipi
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

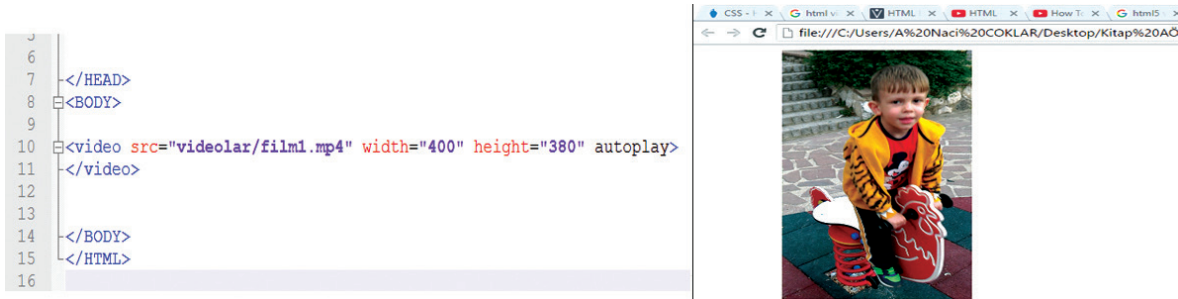
Tablo 3.2'den de görüleceği üzere eğer web sayfanıza bir video eklemek isterseniz, bu video dosyanızı mp4, webM veya ogg uzantılı dosyaya dönüştürücü programlar aracılığı ile dönüştürmeniz gerekmektedir. Mime tipi kullanımı zorunlu olmamakla birlikte `<video>` etiketinde type parametresi kullanılması durumunda atanacak değerdir. Örnek olarak en basit hâli ile tasarlanan web sayfasının içerisindeki videolar klasöründe bulunan *film1.mp4* dosyası aşağıdaki şekilde sayfaya dâhil edilebilir (Resim 3.5).



Resim 3.5 En yalın hâli ile video ekleme örneği

Resim 3.5'deki gibi video eklenmesi durumunda bazı sorunlar ortaya çıkmaktadır. Öncelikli olarak video otomatik olarak başlatılmamakta, ayrıca play, yani oynat düğmesi de bulunmamaktadır. Diğer bir sorun ise videonun kontrolsüz derecede büyük olmasıdır. Resmin büyüklüğü için de kullanılan *width* ve *height* parametreleri video için de geçerli olup benzer şekilde kullanılmaktadır. Ayrıca videonun sayfa yüklenir yüklenmez başlatılması için de *autoplay* parametresi kullanılmalıdır (Resim 3.6).

`<video src="videolar/film1.mp4" width="400" height="380" autoplay> </video>`



Resim 3.6 Videonun boyutlandırılması ve otomatik başlatılması örneği

Videolar üzerinde istenilen bir diğer önemli özellik ise video kontrolünün yapılabilmesidir. Bir başka ifade ile durdurulup başlatılabilmesi, sesin kontrol edilebilmesi veya istenilen video bölümüne geçiş yapılabilmesi ve tam ekran gösterim özellikleri önemli olup, html5'de video etiketinin içerisine yazılacak *controls* parametresi bu işlevleri yerine getirmektedir (Resim 3.7)

`<video src="videolar/film1.mp4" width="400" height="380" autoplay controls> </video>`

✓ Videolarda öne çıkan başlıca parametreler autoplay, controls, width, height, poster ve src'dir.



Resim 3.7 Video kullanıcı denetimlerinin aktif hâle getirilmesi

Ayrıca video oynatım süresinin bitmesi durumunda yeniden başlatılabilmesi için video etiketinin içerisine controls'a benzer şekilde sadece *loop* yazmak yeterli olacaktır. Diğer yandan tasarımcılar için sunulan bir diğer seçenek ise *poster* parametresidir. Poster parametresi videonun oynatılmasından önce onu temsilen bir resim kullanımı anlamına gelmektedir. Bu özelliğin kullanılması durumunda autoplay özelliğinin kaldırılması gerekir ve kullanımı Resim 3.8'deki gibi olmalıdır.



`<video src="videolar/film1.mp4" width="400" height="380" controls poster="efetatil.jpg" loop>`



Resim 3.8 Videoya başlangıç resmi atamak

Videolar ile ilgili bir diğer özellik ise tarayıcının desteklememesi durumunda hazırlanan diğer uzantılı videonun otomatik oynatılması veya hiçbir uzantının desteklenmemesi durumunda tarayıcının hata mesajı vermesidir. Bunun için video etiketinin içindeki tek bir hedef kullanılabilen src kaynak dosya gösteriminin `<video>` etiketi içinde kullanmak yerine `<source>` etiketi ile birden fazla kullanılması yeterli olacaktır. Örnek olarak film1.mp4, film1.webM dosya formatları videolar klasöründe kayıtlı olsun. Öncelikli olarak mp4 uzantılı dosya, eğer açılmazsa webM uzantılı dosya, o da açılmazsa *Tarayıcınız HTML5'i desteklememektedir* şeklinde mesaj için aşağıdaki kodlar kullanılmalıdır.

✓ Poster parametresi ile videolara başlangıçta resim atamak mümkündür.

```

<video width="400" height="380" poster="efetatil.jpg" controls>
  <source src="videolar/film1.mp4">
  <source src="videolar/film1.webM">
  Tarayıcınız HTML5'i desteklememektedir
</video>

```

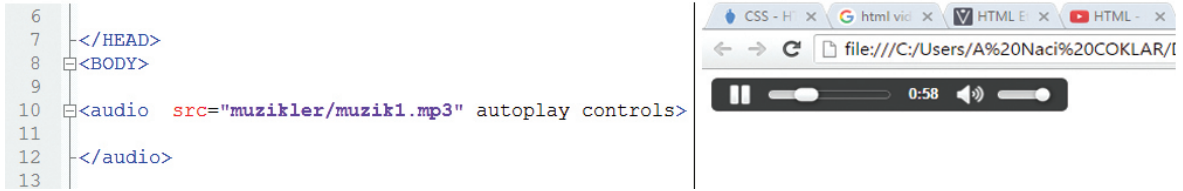

Html5 ile Seslerin Eklenmesi

Html5 ile web sayfalarına eklenebilecek ve çok sık kullanılan bir diğer çokluortam ögesi ise sestir. Html5 için desteklenen dosya uzantı türleri ise mp3, ogg ve wav'dır. Örnek olarak web sayfasına müzikler klasöründeki muzik1.mp3 isimli ses dosyasını eklemek isteyen bir tasarımcı için bu işlem çok basittir ve `<audio>` etiketi ile aşağıdaki şekilde yapılmaktadır.

```
<audio src="muzikler/muzik1.mp3"> </audio>
```

Ancak bu durumda ses duyulmayacaktır. Web sayfalarındaki ses nesneleri için de parametreler bulunmaktadır ve `<video>` etiketinde anlatılan parametrelerin çoğu bu parametre için de kullanılmaktadır. Yukarıdaki komutta ses dosyası başlatılmamış, kullanıcıya kontrol denetimi de sağlanmamıştır. Bu nedenle kullanıcı sesi aktif hâle getiremeyecektir. Bu durumda ses otomatik başlatılabileceği gibi kullanıcı kontrol özellikleri aktif hale getirilebilir (Resim 3.9).

```
<audio src="muzikler/muzik1.mp3" autoplay controls> </audio>
```



Resim 3.9 Ses dosyalarının eklenmesi

Diğer yandan ses dosyasının sürekli bittikten sonra yeniden başlamasını isteyen bir tasarımcı, loop parametresini `<audio>` etiketinin içerisine eklemelidir.

```
<audio src="muzikler/muzik1.mp3" autoplay controls loop> </audio>
```

Video dosyalarında açıklandığı gibi, ses dosyasının türünün yani uzantısının desteklenmemesi durumunda diğer dosya uzantılarının başlatılabilmesi veya hiçbirinin desteklenmemesi durumunda uyarı mesajının verilmesi de sağlanabilir. `<source>` etiketinin kullanımı aşağıdaki şekilde olacaktır.

```
<audio controls loop>
  <source src="muzikler/muzik1.mp3">
  <source src="muzikler/muzik1.ogg">
Tarayıcınız HTML5'i desteklememektedir
</audio>
```

✓ `<audio>` ve `<video>` etiketlerinde çoğu parametrenin kullanımı aynıdır.

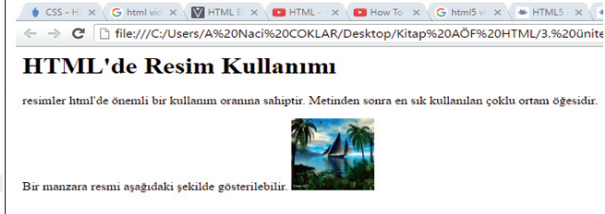
Ses, resim ve video bağlamında çokluortam öğeleri için html5 ile gelen, sıkça internet örneklerinde karşılaşılan ve düzeni sağlamayı amaçlayan iki önemli etiket olarak `<figure>` ve `<figcaption>` etiketlerini kısaca açıklamak yararlı olacaktır. `<figure>` etiketi birleştirme özelliği taşımaktadır ve içerisinde bir resim, gösterim, diyagram, kod listeleri gibi öğelerin olduğunu belirtir. `<figure>` elementi ana akış ile ilgili olsa da, konumu ana akıştan bağımsızdır ve çıkarılırsa dökümanın akışını engellemez. `<figcaption>` etiketi ise `<figure>` elementinin belirttiği çokluortam ögesine başlık eklemeye yarar. Resim 3.10'de önce `<figure>` ve `<figcaption>` etiketleri kullanılmadan, sonrasında ise kullanılarak iki resim ekleme örneği verilmiştir.

<p> resimler html'de önemli bir kullanım oranına sahiptir. Metinden sonra en sık kullanılan çokluortam öğesidir. </p>
 <p> Bir manzara resmi aşağıdaki şekilde gösterilebilir.
 <figure>

 <figcaption> Efem tatilde
 </figcaption>
 ...

✓ Html'de bir çokluortam öğesi olduğunu belirlemek için <figure> etiketi kullanılır.

```
14
15 <p> resimler html'de önemli bir kullanım
16 oranına sahiptir. Metinden sonra en sık
17 kullanılan çoklu ortam öğesidir.
18 <p> Bir manzara resmi aşağıdaki şekilde gösterilebilir.
19 
20
21
22 </BODY>
23 </HTML>
24
```



```
10
11 <p> resimler html'de önemli bir kullanım
12 oranına sahiptir. Metinden sonra en sık
13 kullanılan çoklu ortam öğesidir.
14 <p> Bir manzara resmi aşağıdaki şekilde gösterilebilir.
15 <figure>
16 
17 <figcaption> Efem tatilde
18 </figcaption>
19
```



Resim 3.10 Figure ve figcaption etiketleri ile çokluortam öğelerinin düzenlenmesi



yaşamla ilişkilendir

YouTube Artık Tamamen HTML5 Kullanıyor

YouTube artık tamamen HTML5 altyapısına geçildiğini müjdeledi.

Dünyanın en büyük video sitesi konumunda bulunan YouTube, artık resmen HTML5 altyapısına geçtiğini açıkladı. Bundan yaklaşık beş yıl önce HTML5 desteklemeye başlayan servis, aradan geçen yıllarda Flash tabanlı video oynatıcı altyapısından bir türlü vazgeçmemiştir.

Sık sık sorunlar yaşayan ve devamlı güncellenmesi gereken Flash, uzun süredir pazar payı kaybediyor. Önce Apple tarafından terk edilen Flash, zamanla yerini HTML5'e bırakmaya başladı. Ne var ki HTML5'in henüz yeterince ol-

gunlaşmadığını düşünen YouTube, geçen yıllar içinde Flash kullanmaya devam etmişti.

Bu değişiklik ne anlama geliyor? YouTube bundan sonra varsayılan olan HTML5 kodu üzerinden video gösterimi yapacak. Bunu destekleyen internet tarayıcılar ise Chrome, Internet Explorer 11, Safari 8 ve Firefox'un bazı Beta sürümleri. İnternet üzerindeki en büyük video sitesinin de Flash kullanmayı bırakması, Flash'ın artık günlerinin sayılı olduğunu düşündürüyor.

Kaynak: <https://www.technopat.net/2015/01/28/youtube-artik-tamamen-html5-kullaniyor/>

Öğrenme Çıktısı



3 Web sayfasına video ekleyebilme
4 Web sayfasına ses ekleyebilme

Araştır 3

HTML5 standardının oluşmasından önce web sayfalarına video eklemek için kullanılan yöntemleri araştırınız.

İlişkilendir

Web sayfalarının, mobil teknolojiler ile bağlandığında daha düşük çözünürlüklü video ve ses sunma eğilimi göstermelerinin nedenini tartışınız.

Anlat/Paylaş

Web sayfalarına eklenen video ve ses içeriklerinin hangi sunum özelliklerinin değiştirilebileceğini anlatınız.

HARİCİ UYGULAMA VE İÇERİKLERİN ENTEGRASYONU

Çoklu ortam olarak yukarıda anlatılan ses, video ve resim uygulamaları ile bir önceki ünite olan metin en çok kullanılan uygulamalar olarak ifade edilebilir. Ancak günümüzde zenginleşen web tasarım destek araçları (animasyonlar), farklı firmaların sunduğu hazır uygulamalar da web ortamında çoklu ortam olarak değerlendirilmektedir. Bu kapsamda popüler bazı uygulamalar aşağıdaki verilmiştir.

Flash Animasyonlarının Eklenmesi

Günümüzde animasyonlar web sayfalarına banner eklenmesi, anlatımları eğlenceli ve ilgi çekici hâle getirmesi, kullanıcının kontrol imkânı bulması gibi nedenlerle sıklıkla kullanılan çoklu ortam öğeleridir. Adobe Flash programı tarafından üretilen ve iki boyutlu tasarım ürünü olan. swf uzantılı animasyon dosyaları, web sayfalarında sıklıkla ve yaygın olarak kullanılması adına önemli çoklu ortam öğesi olarak vurgulanmalıdır. Bu kapsamda swf bir dosyanın site içerisinde kullanılması mümkün olup, bunu gerçekleştiren html etiketi ise `<embed>` etiketidir. Bir örnek üzerinden açıklamak gerekirse, tasarlanan web sayfasının içerisindeki animasyonlar alt klasörünün içerisindeki devre.swf isimli bir flash animasyon dosyası aşağıdaki kodlar ile eklenecektir (Resim 3.11).

```
<embed src="animasyonlar/devre.swf">
```

✓ Flash animasyonları ile pdf dosyaları eklemek için `<embed>` etiketi kullanılır.

```
7 </HEAD>
8 <BODY>
9 <h1> Suyun İletkenliği</h1>
10
11 <p> Suyun iletkenliği, saf su, tuzlu ve tatlı su da
12 farklılaşabilmektedir. Tuzlu suyun iletkenliği en fazladır.
13 Örnek animasyon üzerinden bu durumu inceleyelim. </p>
14 <figure>
15 <embed src="animasyonlar/devre.swf" >suyun iletkenliği</embed>
16 <figcaption> Suyun İletkenliği
17 </figure>
18
19 </BODY>
20 </HTML>
21
22
```

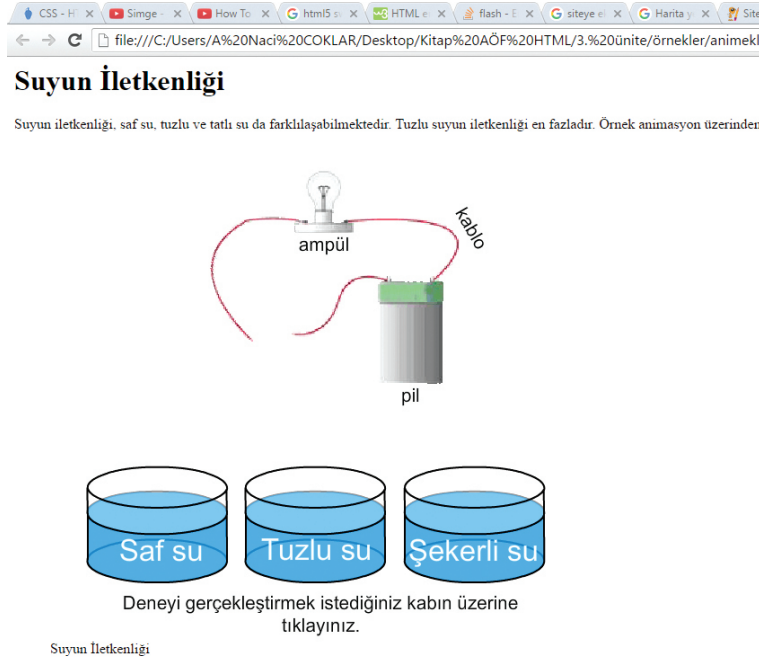


Resim 3.11 Flash animasyonlarının eklenmesi

Resim 3.11'in eklenmesi durumunda flash animasyonu web sayfasına eklenecektir. Ancak boyut sorunu bulunmaktadır. Bunun için resim ve video'da kullanıldığı şekilde width ve height komutlarının kullanılması gerekir.

```
<embed src="animasyonlar/devre.swf" width="600" height="550">
```

Animasyon boyutunun ayarlanması ile web sayfasında animasyonun görünümü Resim 3.12'deki gibi olacaktır.



Resim 3.12 Animasyonların boyutlandırılması örneği

Embed etiketinin yanında kullanılabilecek diğer bazı parametreler ise align, quality, pluginspage'dir. Align parametresi left ve right değerlerini alıp, sadece sola ve sağa hizalamak amacı ile kullanılır. Eğer animasyon ortaya hizalanacaksa daha önce bahsedildiği gibi `<embed>` etiketi `<center>` `</center>` etiketleri arasında kullanılmalıdır. Quality ise animasyonun gösterim kalitesi için kullanılır ve high, medium ve low olmak üzere üç değer alır. Pluginspage ise animasyonun oynatılması için gerekli olan player programı tarayıcı tarafında kurulu değilse, kullanıcıyı uyararak kurması için gerekli olan siteye yönlendirir. Adobe firması için bu adres <http://www.adobe.com/go/getflashplayer> adresidir. Buna göre daha detaylı bir swf dosya yükleme komutu aşağıdaki şekilde olacaktır.

```
<embed src="animasyonlar/devre.swf" width="600" height="550" quality="high" pluginspage="http://www.adobe.com/go/getflashplayer"> </embed>
```

PDF Dosyalarının Eklenmesi

Sitelerde günümüzde doküman dosyalarının doğrudan eklenmesi yerine daha güvenli olduğu düşüncesi ile pdf uzantılı Adobe Acrobat dosyaları paylaşılmaktadır. Web sayfa tasarımcıları da kolaylıkla sayfalarında pdf uzantılı dosyaları paylaşabilirler. Bunun için kullanmaları gereken etiket yine `<embed>` etiketi olup kullanım şekli animasyonların eklenmesi ile aynıdır. Resim 3.13'te embed özelliği ile eklenmiş bir pdf dosyası ve ekran görünümü verilmiştir.

```
<embed src="dokumanlar/kaynak.pdf" width="600" height="550">
```

```

6  </HEAD>
7  <BODY>
8  <h1> Suyun İletkenliği </h1>
9
10 <p> Suyun iletkenliği, saf su, tuzlu ve tatlı su da
11 farklılaşabilmektedir. Tuzlu suyun iletkenliği en fazladır.
12 Örnek animasyon üzerinden bu durumu inceleyelim. </p>
13 <figure>
14 <embed src="dokumanlar/kaynak.pdf" width="600" height="550">
15 <figcaption> Suyun İletkenliği
16 </figure>
17
18 </BODY>
19 </HTML>
20
21
22
23
24

```

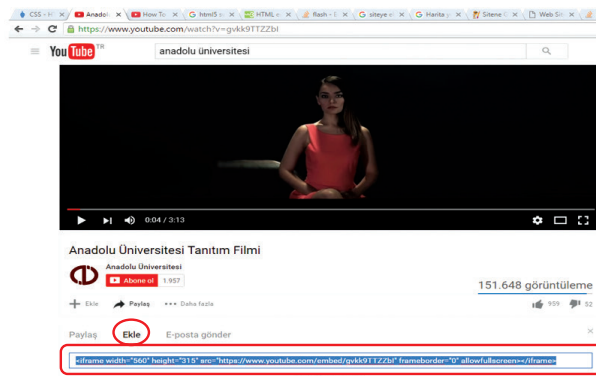


Resim 3.13 Sayfaya pdf dosyasının eklenmesi

Youtube Videolarının Eklenmesi

Youtube videolarının eklenmesi `<video>` etiketi yerine `<embed>` etiketi ile kolay bir şekilde yapılmaktadır. Hatta youtube paylaşımları sitede yapabilmek için kendisi ilgili paylaşım ile ilgili `<iframe>` oluşturmakta, gelen kod doğrudan siteye eklenerek videonun kişisel web sayfasında oynatımı mümkün hâle gelmektedir. Bunun için öncelikli olarak youtube sitesinden (www.youtube.com) paylaşılacak istenen video bulunmalı, aşağıda yer alan *Paylaş*, *Ekle* ve *E-Posta* gönder linklerinden *Ekle* tıklanmalıdır. Gelen `<iframe>` ile başlayan link kopyalanmalı ve tasarımcının html sayfasına doğrudan kopyalanmalıdır (Resim 3.14).

✓ Youtube videosu eklemek youtube'nin ürettiği birkaç satırlık `<iframe>` etiketinin site içerisine eklenmesi ile oluşur ve yapılan işlem aslında sadece link vermedir.



Resim 3.14 Youtube videolarının eklenmesi

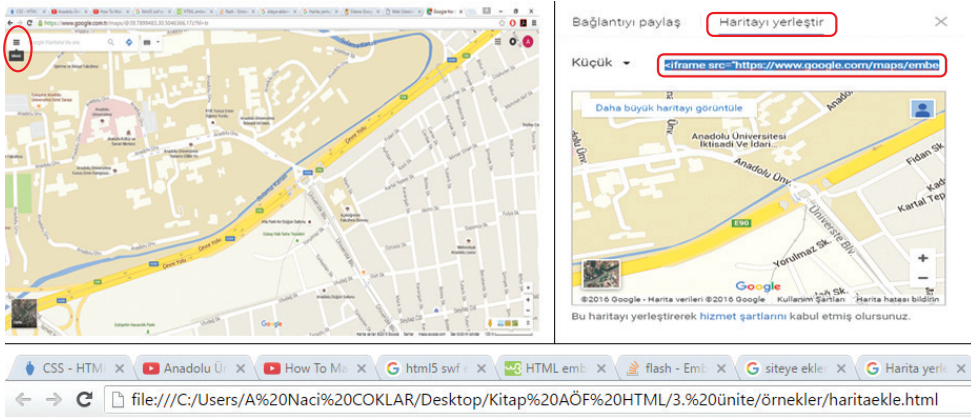
Videoların eklenmesinde <iframe> etiketinin içerisine align, width ve height gibi daha önce anlatılan komutlar ile müdahale etmek mümkündür.

Web Sitesine Konum Bilgisinin Eklenmesi

Son dönemlerde özellikle iletişim bölümlerinde kullanımı yaygınlaşan bir diğer uygulama ise adres konum bilgisinin harita olarak eklenmesidir. Aslında bu uygulama da youtube videoları gibi, Google tarafından sunulan maps.google.com hizmetinde konumun işaretlenmesi ve siteye üretilen html kodunun kopyalanıp yapıştırılması şeklindedir. Örnek olarak Anadolu Üniversitesinin ana giriş kapısına ait konum bilgisini web sayfa tasarımcısı tarafından aşağıdaki şekilde eklenmelidir.

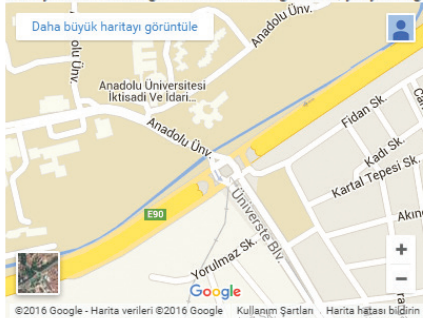
✓ Günümüzde kullanımı gittikçe yaygınlaşan harita konum bilgisi eklemekte, youtube videosu gibi Google tarafından üretilen <iframe> ile yapılmaktadır ve çok basit bir işlemdir.

maps.google.com adresinden *Anadolu Üniversitesi* konum olarak aratılmalı ve nokta olarak belirtilmelidir. Sonrasında ise sol üst köşede görünen menü seçeneğine tıklanmalı ve gelen seçenekten de *Paylaşın veya harita yerleştirin* seçeneği, Sonrasında gelen ekranda ise *bağlantı paylaşın* yerine *haritayı yerleştir* seçeneği seçilmelidir. Gelen ekranda ayrıca sayfaya yapıştırılacak kodun <iframe> olarak geldiği görülebilir. Nihai olarak bu ekrandan karar verilebilecek bir diğer konu ise haritanın gösterilecek web sayfası içerisindeki büyüklüğüdür. Bunun için sol üst köşede küçük, orta, büyük ve özel boyut şeklinde seçenekler sunulmaktadır. İstenirse değerlerin girilmesi için özel boyut seçilerek büyüklük ayarlanabilir. Son seçim işleminden sonra <iframe> etiketi güncellenir ve kopyalamaya hazırdır. Bu işlem sonrasında üretilen kod, tasarımcı tarafından siteye kopyala yapıştır yapılarak yerleştirilebilir (Resim 3.15).



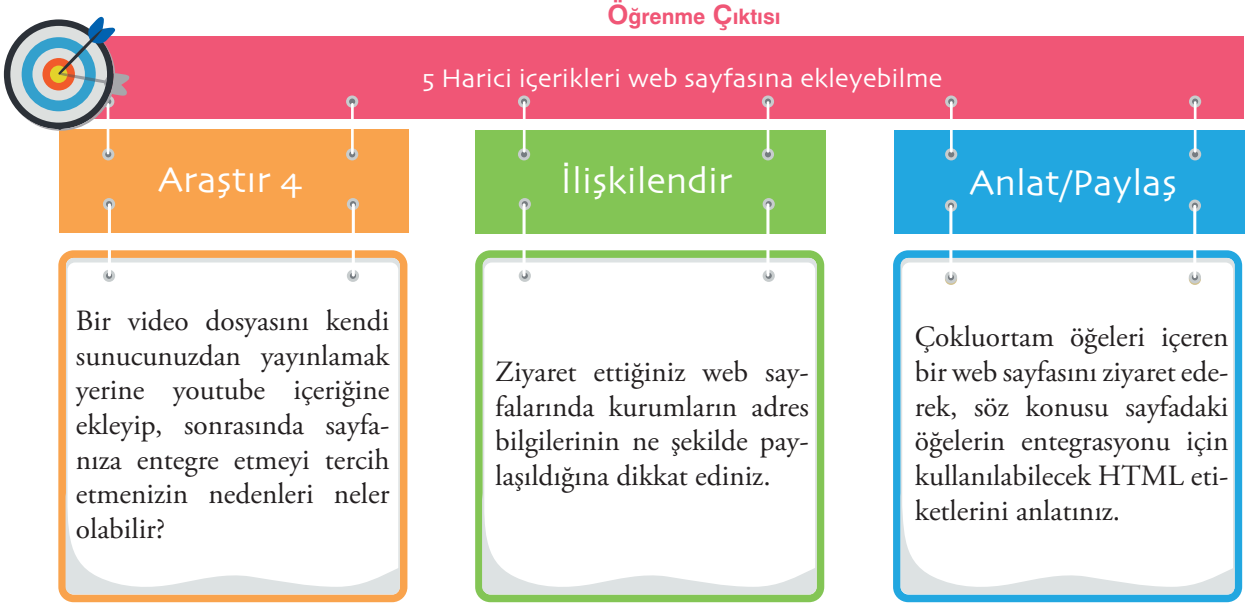
Anadolu Üniversitesi

Türkiye'nin uzaktan eğitim alanında olduğu kadar, yüz yüze eğitimi ile de örnek kampüsü ile de en güzel üniversitelerinden bir tanesidir.



Resim 3.15 Haritaların siteye eklenmesi

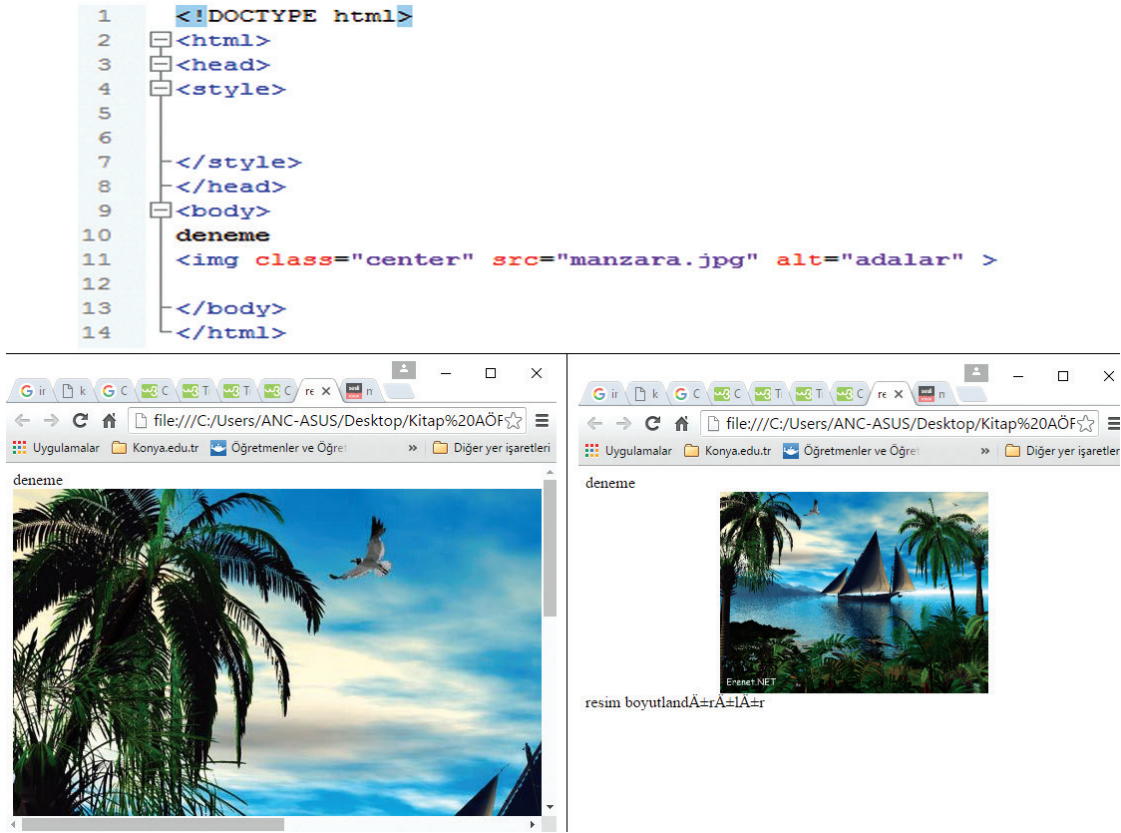
Görüleceği üzere, web sayfalarına eklenecek çoğu bilgi aslında bu hizmeti veren kişi veya kurumlar tarafından üretilmekte ve saklanılmaktadır. Aslında bu uygulamalarına yer veren çoğu tasarımcı kendi sitesinde link verme işlemi gerçekleştirmektedir.



ÇOKLUORTAM ÖĞELERİNDE STİL KULLANIMI

Metinler kadar zengin olmamakla birlikte resim, ses ve video çokluortam nesneleri için de stil tanımlamak mümkündür. Öncelikli olarak `<img>` etiketine yönelik temel CSS stil özellikleri üzerinde durulabilir. Stil tanımlaması metin konusunda detaylı bir şekilde anlatıldığı gibi burada da `style="özellik:değer;"` şeklindedir. Ayrıca değişiklik için daha çok etiketin içerisinde kullanım (inline) yerine gömülü stil kullanımı şeklinde yapılmaktadır. Bir önceki üniteden hatırlanacağı üzere gömülü değişiklik sayfada geçen tüm `<img>` etiketi kullanılan yerlerde etkisini gösterecektir. Aşağıdaki örnekte etiketin içerisinde ve gömülü olarak resim stilleri kullanılmış olup stil resmi tarayıcı da ortalama bir şekilde tarayıcı boyutunun büyütülmesi veya küçültülmesi durumunda resim de bu değişiklikten etkilenmektedir (Resim 3.16).

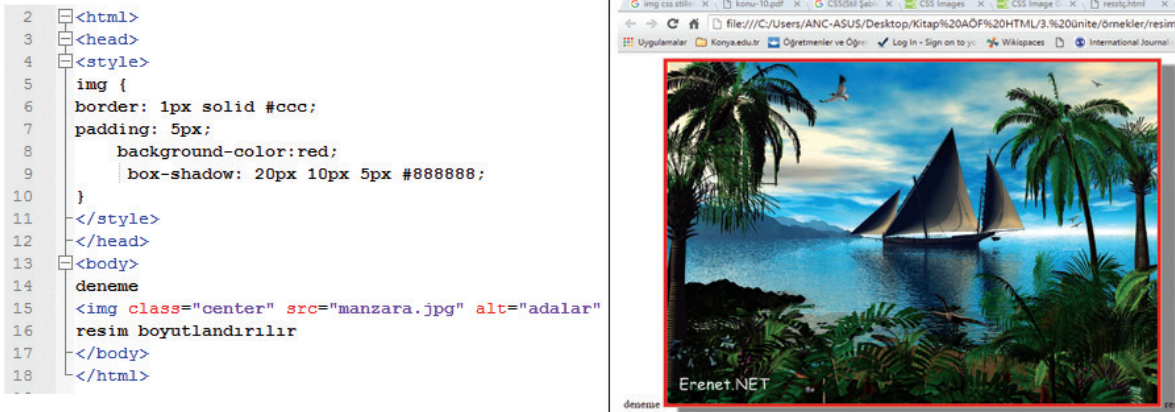
✓ Çokluortam öğelerinde de stil kullanımı metin kullanımında olduğu gibi etikette, gömülü veya harici dosya ile yapılabilir.



Resim 3.16 Otomatik boyutlandırma stili kullanmadan ve kullanarak resmin gösterilmesi örneği

Çok ihtiyaç duyulacak birkaç stil örneği daha vermek gerekirse *border: 1px solid #ccc;* stili ile resmi 1px genişliğinde ve belirtilen renkte kutucuk içerisine alma, *padding: 15px;* ile kutucuk ile resim arasındaki mesafeyi belirleme, *background-color:red;* ile arkaplanına renk verme ve *box-shadow: 10px 10px 5px #888888;* parametresi ile de gölge verme gibi temel işlemler yapılabilir (Resim 3.17). Değerler değiştirilerek stillerin gösterimleri de belirlenebilir. Örneğin en son yazılan gölge komutunda piksel değerleri değiştirilerek gölgenin uzaklığı da ayarlanabilir.

✓ Resim üzerinde istenilen bir konuma metin yazabilmek, resimler üzerine efektler eklemek gibi işlemlere izin veren çok sayıda eğlenceli ve önemli resim stilleri de bulunmaktadır.



Resim 3.17 Resmi kutuya alma, kutuya renk verme ve gölge ekleme resim stilleri

Bir diğer çokluortam ögesi olan video ile ilgili stil tanımlaması ise aşağıdaki şekilde yapılabilir.

```
<style>
video {
  video stil kuralları
}
</style>
```

Burada kullanılabilecek bazı kurallar listesi ise Tablo 3.3'te verilmiştir. Bir sitede bütün video genişlik ve yüksekliklerinin aynı değere (width=100px, height=140px) sahip olmasını isteyen bir tasarımcının stil olarak aşağıdaki tanımlamayı yapması durumunda video etiketlerinde tekrar tanımlamasına gerek kalmayacaktır.

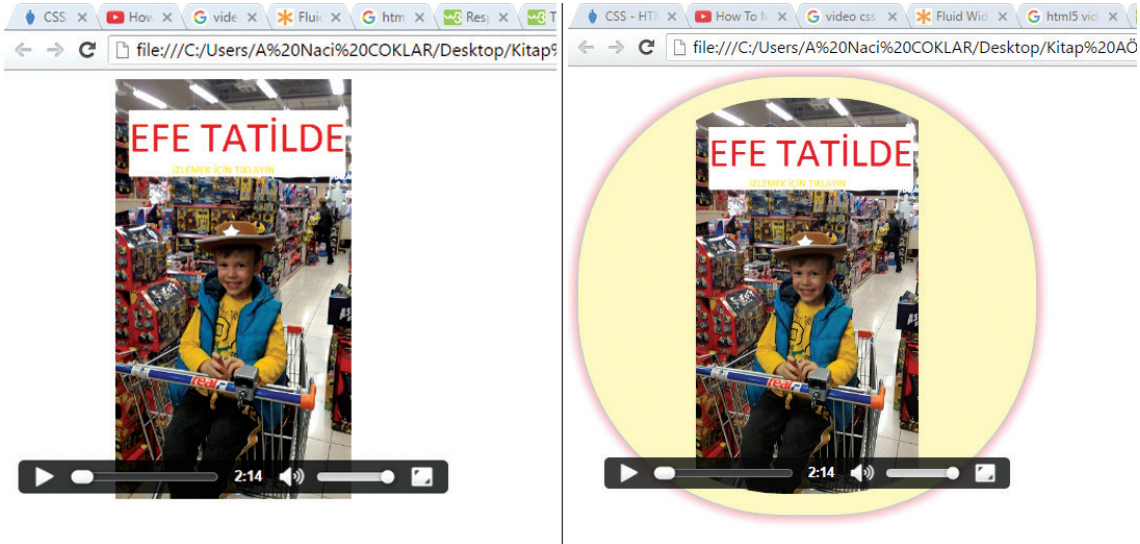
```
<head>
<style>
video {
  height:140px;
  width:100px;
}
</style>
</head>
```

Tablo 3.3 Video etiketinin alabileceği bazı stil özellikleri, işlevleri ve örnek değerler

Özellik	İşlevi	Alabileceği Örnek Değerler
width	Genişliği	250px;
height	Yüksekliği	250px;
max-width	Geniştirilebilirlik sınır oranı	100%;
border	Videoya kenarlık ekler	1px solid #ccc;
padding	Video ile kenarlık arası boşluğu ayarlar	20px;
margin	Tarayıcıdaki yazıya göre konumunu belirler.	10px;
border-radius	Kenarlığın köşesini yuvarlar. İstenirse daire yapılabilir.	20px; 200px;
background-color	Zemin rengini ayarlar	#ffcccc; red; yellow;
background-image	Özellikle geçişli renklerde kullanılır	linear-gradient(top, #fff, #fcc); -moz-linear-gradient(top, #fff, #fcc);
box-shadow	Çizilen kutuya gölge ekler.	0 0 20px #f88;

Bu tablo kullanılarak Resim 3.8'de verilen kodlardan yola çıkarak <head> </head> arasına aşağıdaki kodları yazarak video üzerinde Resim 3.18'deki gibi bir değişiklik yapılabilir.

```
<head>
<style>
video {
border:2px solid #ccc;
padding:20px;
border-radius:200px;
background-color:ffffc1;
box-shadow:0 0 20px #f88;
}
</style>
</head>
```



Resim 3.18 Videoda css uygulama örneği

Yalnız burada bilinmesi gereken nokta birden fazla özelliğe aynı anda stil ayarı verilebilir. Örneğin video kontrolü üzerindeki değişkenlere gölge vermek isteyen bir kullanıcı tek tek tanımlamak yerine, aşağıdaki kullanım ile tek bir satır ile tamamı için geçerlilik sağlayabilir. Yani #nesne adı ve arada , kullanarak yapılacak değişiklik, video nesnesi üzerindeki tüm değişkenleri etkileyecektir.

```
video, #start, #stop, #pause, #plus, #minus, #mute { box-shadow: 0 0 10px #ccc; }
```

Aşağıda daha önce verilen video ekleme örneğine stil eklenerek ön ve arka planının değiştirilmesine yönelik örnek, Resim 3.19'da ise stil kullanmadan önceki ve sonraki hâlleri karşılaştırılarak verilmiştir.


```

<HTML>
<HEAD>
<meta http-equiv=content-type content=text/html; charset=utf-8>
<title> Videolarda Stil Kullanımı </title>
<style>
video{
/* video kenarlığı */
border: 1px solid #ccc;
padding: 20px; margin: 10px;
border-radius: 20px;

/* Geçişli renklerin ayarlanması */
-moz-transition: all 1s ease-in-out;
-webkit-transition: all 1s ease-in-out;
-o-transition: all 1s ease-in-out;
-ms-transition: all 1s ease-in-out;
transition: all 1s ease-in-out; }

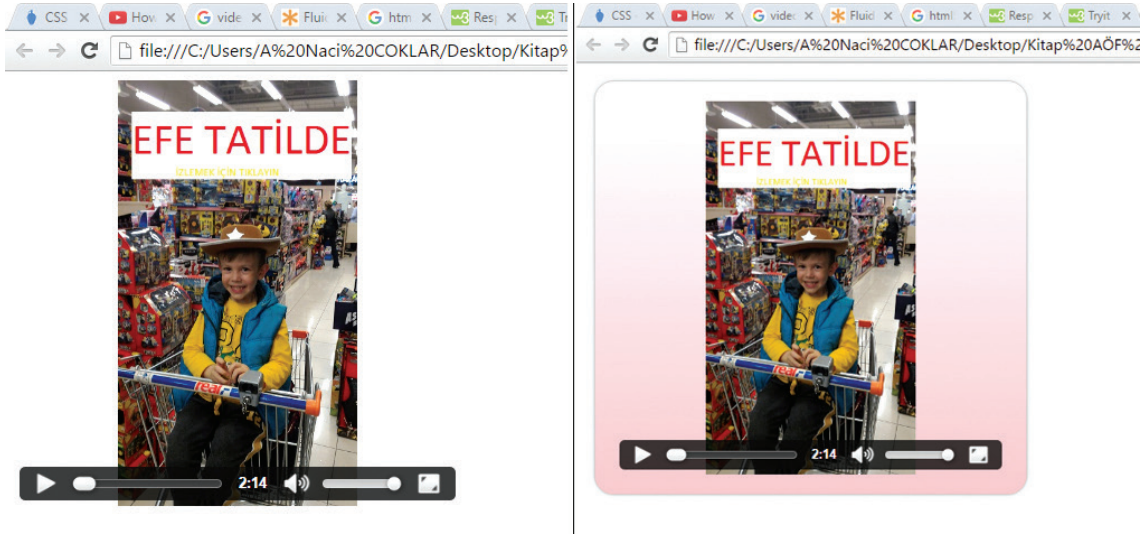
/* zemin rengi ve renk geçişleri */
video, #start, #stop, #pause, #plus, #minus, #mute {
/* background color */
background-color: #ffcccc;
/* background gradient */
background-image: linear-gradient(top, #fff, #fcc);
background-image: -moz-linear-gradient(top, #fff, #fcc);
background-image: -webkit-linear-gradient(top, #fff, #fcc);
background-image: -o-linear-gradient(top, #fff, #fcc);
background-image: -ms-linear-gradient(top, #fff, #fcc); }

/* Gölgeler kontrollerde */
video, #start, #stop, #pause, #plus, #minus, #mute { box-shadow: 0 0 10px #ccc; }
video:hover, video:focus, #start:hover, #stop:hover, #pause:hover, #plus:hover,
#minus:hover, #mute:hover {
/* glow */
box-shadow: 0 0 20px #f88; }
#controls { display: none; margin: 10px 30px; }

/* Başlangıç büyüklüğü */
height: 20px; width: 0;
}
</style>
</HEAD>
<BODY>

<video width="400" height="380" poster="efetatil.jpg" controls>
  <source src="videolar/film1.mp4" />
</video>
</BODY>
</HTML>

```



Resim 3.19 Stilleri ile videonun arka ve ön planının değiştirilmesi

Özellikle vurgulanması gereken konu ise bu sayfada tekrar bir video kullanılması <video> etiketi bu değişimden etkileneceği için aynı ayarlar diğer video için de geçerli olacaktır. Eğer bir tasarımcı web sitesindeki tüm sayfalarda aynı ayarın geçerli olmasını isterse bu durumda bir önceki üniteye detaylı olarak anlatıldığı şekilde harici bir CSS dosyası oluşturmalı ve link veya @import ile çağırmalıdır. Benzer şekilde ses için yani <audio> etiketi için de css kuralları mevcuttur. Ancak görsellik çok ön planda bulunmadığı için sadece player üzerinde işlem yapmak mümkündür. Aşağıdaki kodların eklenmesi durumunda müzikle ilgili player'ın üzerine gelme, üzerinden ayırılma durumunda efekt oluşmaktadır.

✓ Videolar ile ilgili web sayfalarında görünümleri ilgi çekici ve profesyonel görünüm özelliği kazandıracak çok sayıda stil uygulamasına internette kısa bir araştırma ile erişebilirsiniz.

```

<style>
audio:hover, audio:focus, audio:active
{
;webkit-box-shadow: 15px 15px 20px rgba(0,0, 0, 0.4-
;moz-box-shadow: 15px 15px 20px rgba(0,0, 0, 0.4-
;box-shadow: 15px 15px 20px rgba(0,0, 0, 0.4
;webkit-transform: scale(1.05-
;moz-transform: scale(1.05-
;transform: scale(1.05
{

audio
}
;webkit-transition:all 0.5s linear-
;moz-transition:all 0.5s linear-
;o-transition:all 0.5s linear-
;transition:all 0.5s linear
;moz-box-shadow: 2px 2px 4px 0px #006773-
;webkit-box-shadow: 2px 2px 4px 0px #006773-
;box-shadow: 2px 2px 4px 0px #006773
; moz-border-radius:7px 7px 7px 7px-
; webkit-border-radius:7px 7px 7px 7px-
; border-radius:7px 7px 7px 7px
{
</style>

```

Öğrenme Çıktısı

6 Çokluortam nesneleri için CSS kullanabilme

Araştır 5

Çokluortam öğeleriyle zenginleştirilmiş bir web sayfasını hızlıca yapmak için kullanabileceğiniz stil şablonları araştırınız.

İlişkilendir

Video içeriğine sahip web sitelerinin video sunum arayüzlerinin birbirlerinden farklı olmasının stil kullanımıyla olası ilişkisini yorumlayınız.

Anlat/Paylaş

Bir çokluortam öğesinin sunum arayüzünün görünümünün stil kullanımıyla ne ölçüde özelleştirilebileceğini anlatınız.

1

Web sayfasına resim ekleyebilme

Resimlerin Görüntülenmesi

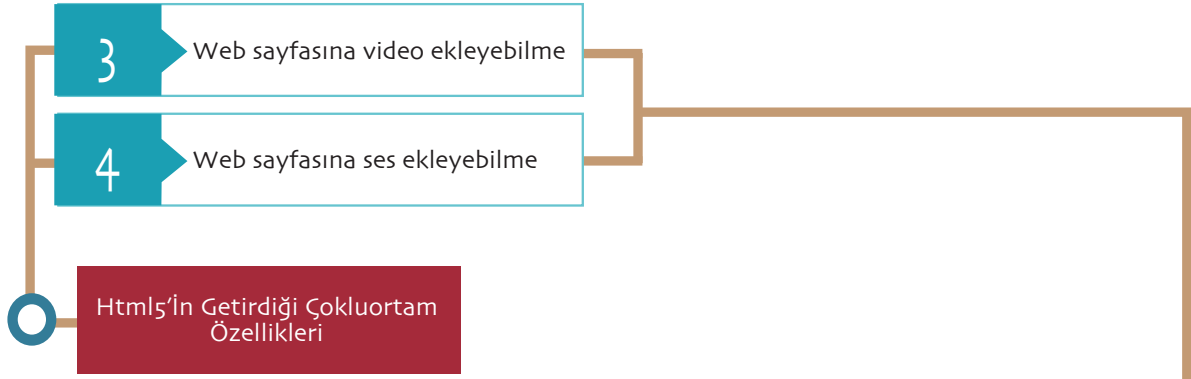
Web sayfalarının içerisinde metinden sonra en çok yer alan öğelerden biri de resimdir. Resim ekleme kelime işlemci programlarındaki ekle komutu gibi basit bir işlemle yapılamamaktadır. Bunun yerine bir klasör içerisinde yer alan bir resim dosyasına orada görünmesi şeklinde etiket ile talimat verilir ve tarayıcı ilgili konumda resmi istenilen özellikleri ile gösterecektir. Bu amaçla kullanılan etiket ise `<img>` etiketidir. `` şeklinde bir gösterim ile hangi resmin görüntüleneceği belirtilir. Tarayıcı bu satırı yorumladığı anda resmi ekrana yansıtacaktır. Ayrıca yatayda ve dikeyde belirtilen pixel ebatlarında görüntüleme işlemi için `width` ve `height` parametreleri kullanılacağı gibi, `align` parametresi ile resmin sağa, sola veya ortaya hizalaması, `alt` ile de resmin üzerine gelmesi durumunda görüntülenecek metnin ayarlanması yapılabilir.

2

Resim haritaları aracılığıyla link verebilme

Resim Haritalarının Oluşturulması

Resim haritaları resimlerin üzerinde belirli bölümler-alanlar oluşturmak ve bu bölümlere link vermek amacı ile kullanılırlar. Bunun için kullanılan etiket ise `<map>` etiketidir. Bu kullanım için öncelikli olarak `<img>` etiketi ile bir resim yansıtmak ve bunu `usemap` parametresi ile seçici hâle getirmek gerekir. Bu açıdan `` şeklinde bir gösterim kısaca örneklendirilebilir. Sonrasında ise `<map name="#harita">` etiketleri arasında `<area>` etiketi ile istenilen sayıda bölge oluşturulabilir. `<area>` etiketinin her biri ile bölge oluşturulur ve bu etiketin içerisinde `href` parametresi ile bu alana tıklanması durumunda verilmesi istenilen link atama işlemi gerçekleştirilir. Ayrıca `shape` ve `coords` parametreleri bölge oluşumunda belirleyicidirler. `Shape` parametresi `rect` (dikdörtgen), `circle` (daire) ve `poly` (çokgen) olarak üç değer almakla birlikte, bu parametreye göre koordinat tanımlama değişebilmektedir. Örneğin daire için merkez ve yarıçap verilirken, dikdörtgen için her bir köşe noktasına ait koordinatlar tanımlanmaktadır. Bu açıdan her bir bölge `<area shape="rect" coords="50,50,25" href="bolge1.html">` şeklinde bir kod yapısında olmalıdır.



3 Html 5 özellikle son yıllardaki tüm tarayıcılar tarafından desteklenen ve pratik yollarla video, müzik gibi dosyaları web sayfalarına eklemeyi sağlayan bir avantaj sunmaktadır. Bu açıdan günümüz web sitelerinde sıklıkla kullanılan videolar, pratik bir şekilde, hazır player özellikleri de yansıtılarak sitele- re dâhil edilebilmektedir. `<video src="videolar/film1.mp4" width="400" height="380" autoplay> </ video>` komut yapısı örneğinde olduğu gibi `<video>` etiketi yanına `src`, `width`, `height` ve `autoplay` gibi parametreleri de alarak, video oynatımını kolaylaştırabilmektedir. `Src` ile hangi dosyanın oynatılacağı, `width` ve `height` ile videonun yüksekliği ve genişliği, `autoplay` ile de filmin sayfa yüklenir yüklenmez oynatılmaya başlatılması sağlanmaktadır. Eğer `controls` parametresi kullanılırsa bu durumda player içerisinde bulunan kontrol butonlarının görüntülenmesi de sağlanabilir. Diğer önemli parametreler ise `loop` ve `poster` parametreleri olup, döngü ve video başlangıcına resim ekleme görevleri yapmaktadırlar.

4 Ses dosyalarının html web sayfalarının kullanımı da son derece pratik olarak yapılabilir. Bunun için kullanılması gereken etiket ise `<audio> </audio>` etiketleridir. Bu amaçla kullanılan komut yapısı ba- sitçe `<audio src="muzikler/muzik1.mp3" >` şeklindedir. Ancak gelişmiş olarak ise `<audio src="muzikler/ muzik1.mp3" autoplay controls loop> </audio>` şeklinde parametre kullanılarak özelliği artırılabilir. Bu açıdan `src` müzik dosyasının adresini ifade ederken, `autoplay` otomatik başlatmayı, `controls` player üze- rindeki komutların gösterimini ve `loop`'ta sürekli çalma işlemini gerçekleştirmektedir.

5

Harici içerikleri web sayfasına ekleyebilme

Harici Uygulama ve İçeriklerin Entegrasyonu

Pdf ve swf videolarının eklenmesi <embed> etiketi ile kolayca yapılabilmektedir. Komutun kullanımı genellikle benzer şekildedir. Örneğin pdf için gelişmiş bir kod yapısı <embed src="dokumanlar/ kaynak.pdf" width="600" height="550"> şeklinde iken, swf yani flash animasyon dosyaları için de <embed src="animasyonlar/devre.swf" width="600" height="550"> şeklindedir ve görüleceği üzere aynı olduğu söylenebilir. Tarayıcı otomatik olarak kaynağa göre tepki gösterecektir. Bu açıdan <embed> etiketinde kullanılacak src parametresi ile hangi dosyaların ekleneceği, width ve height ile bunların yükseklik ve genişlik alanları piksel olarak tanımlanmaktadır. Ayrıca sıkça yapılan bir işlem olarak youtube videolarının site içerisinde gösterimini sağlamak mümkündür ve bunun içinse <iframe> etiketi kullanılmaktadır. Bu amaçla youtube'da ilgili videonun adresi "ekle" seçeneğinden kopyalanarak <iframe src= ""> src parametresine yapıştırılmalıdır. İstenirse burada da width, height gibi parametreler kullanılabileceği gibi, allowfullscreen parametresi ile de tam ekran gösterim izni de verilebilir.

6

Çokluortam nesneleri için CSS kullanabilme

Çokluortam Öğelerinde Stil Kullanımı

Çokluortam nesnelerinde stil kullanımı da metin kullanımına benzer şekilde olmaktadır. Çokluortam stil ayarlaması için style parametresi (style="özellik:değer;") metin özellikleri yerine, kendine has özellik ve değerler alabilmektedir. Genellikle gömülü olarak yapılan bu stil tanımlamaları ve kullanımı için <style> </style> etiketleri arasına , <video> vb. parametrelerin özellikler belirlenir ve <body> etiketi kullanımında tüm bu etiketlerin gösteriminde bu özellikler otomatik olarak yansıtılacaktır. Örnek bir stili <style> etiketinde ise aşağıdaki şekilde tanımlanabilir. <style> img{border:1px solid #ccc background-color:yellow; } </style> bu şekilde tüm ile gösterilecek resimlere kırmızı renkli bir kenarlık eklenirken, görünen zemin rengi sarı yapılmış olacaktır. İstenirse bu kısımda box-shadow parametresi ile özellik tanımlanarak kutunun gölgeli olması da sağlanabilir. Benzer şekilde <style> </style> etiketleri arasında <audio> ve <video> etiketlerinin özelliklerine de stil tanımlanarak, çok güzel görünümlü playerlar elde edilebilir.

1 Resimlerin gösterilmesi için kullanılan etiketinde dosyanın konumu aşağıdaki parametrelerden hangisi ile belirtilmektedir?

- A. align
- B. width
- C. span
- D. src
- E. link

2 Aşağıdakilerden hangisi etiketinin alabileceği bir parametrelerden biri **değildir**?

- A. tint
- B. border
- C. src
- D. alt
- E. align

3 Bir etiketinin içerisinde resim haritası olacağını belirtmek için kullanılan parametre aşağıdakilerden hangisidir?

- A. name
- B. border
- C. href
- D. assign
- E. usemap

4 Resim haritalarında alanın hangi şekle sahip olacağı aşağıdaki parametrelerden hangisi ile belirtilmektedir?

- A. rect
- B. coords
- C. area
- D. circle
- E. shape

5 Aşağıdakilerden hangisi <video> etiketi ile kullanılabilecek bir video dosya türüdür?

- A. xls
- B. ogg
- C. doc
- D. pdf
- E. mp3

6 <audio> etiketi ile müzik dosyası eklemek isteyen bir kullanıcı için aşağıdaki dosya uzantılarından hangisi doğru bir seçimdir?

- A. avi
- B. mid
- C. wav
- D. jpeg
- E. rtf

7 Video dosyalarının sitenin yüklenmesi durumunda kendiliğinden başlamasını isteyen bir kullanıcının etiket içerisinde kullanması gereken parametre aşağıdakilerden hangisidir?

- A. poster
- B. controls
- C. loop
- D. autoplay
- E. source

8 Siteye yüklenen bir müzik dosyasının sürekli olarak yeniden başlamasını sağlayan parametre aşağıdakilerden hangisidir?

- A. loop
- B. autoplay
- C. audio
- D. repeat
- E. controls

9 Web sayfasına eklenen bir çokluortam öğesine başlık eklemek için aşağıdaki etiketlerden hangisi kullanılmalıdır?

- A. <title>
- B. <figcaption>
- C. <source>
- D. <embed>
- E. <link>

10 Web sayfasında videoya link vermek, pdf dosyası veya flash animasyon dosyası eklemek için aşağıdaki etiketlerden hangisi kullanılmalıdır?

- A. <link>
- B. <source>
- C. <embed>
- D. <audio>
- E. <video>

1. D

Yanıtınız yanlış ise “Resimlerin Görüntülenmesi” konusunu yeniden gözden geçiriniz.

2. A

Yanıtınız yanlış ise “Resimlerin Görüntülenmesi” konusunu yeniden gözden geçiriniz.

3. E

Yanıtınız yanlış ise “Resim Haritalarının Oluşturulması” konusunu yeniden gözden geçiriniz.

4. E

Yanıtınız yanlış ise “Resim Haritalarının Oluşturulması” konusunu yeniden gözden geçiriniz.

5. B

Yanıtınız yanlış ise “Html5 ile Videoların Eklenmesi” konusunu yeniden gözden geçiriniz.

6. C

Yanıtınız yanlış ise “Html5 ile Seslerin Eklenmesi” konusunu yeniden gözden geçiriniz.

7. D

Yanıtınız yanlış ise “Html5 ile Videoların Eklenmesi” konusunu yeniden gözden geçiriniz.

8. A

Yanıtınız yanlış ise “Html5 ile Seslerin Eklenmesi” konusunu yeniden gözden geçiriniz.

9. B

Yanıtınız yanlış ise “Html5 ile Seslerin Eklenmesi” konusunu yeniden gözden geçiriniz.

10. C

Yanıtınız yanlış ise “Harici Uygulama ve İçeriklerin Entegrasyonu” konusunu yeniden gözden geçiriniz.

3

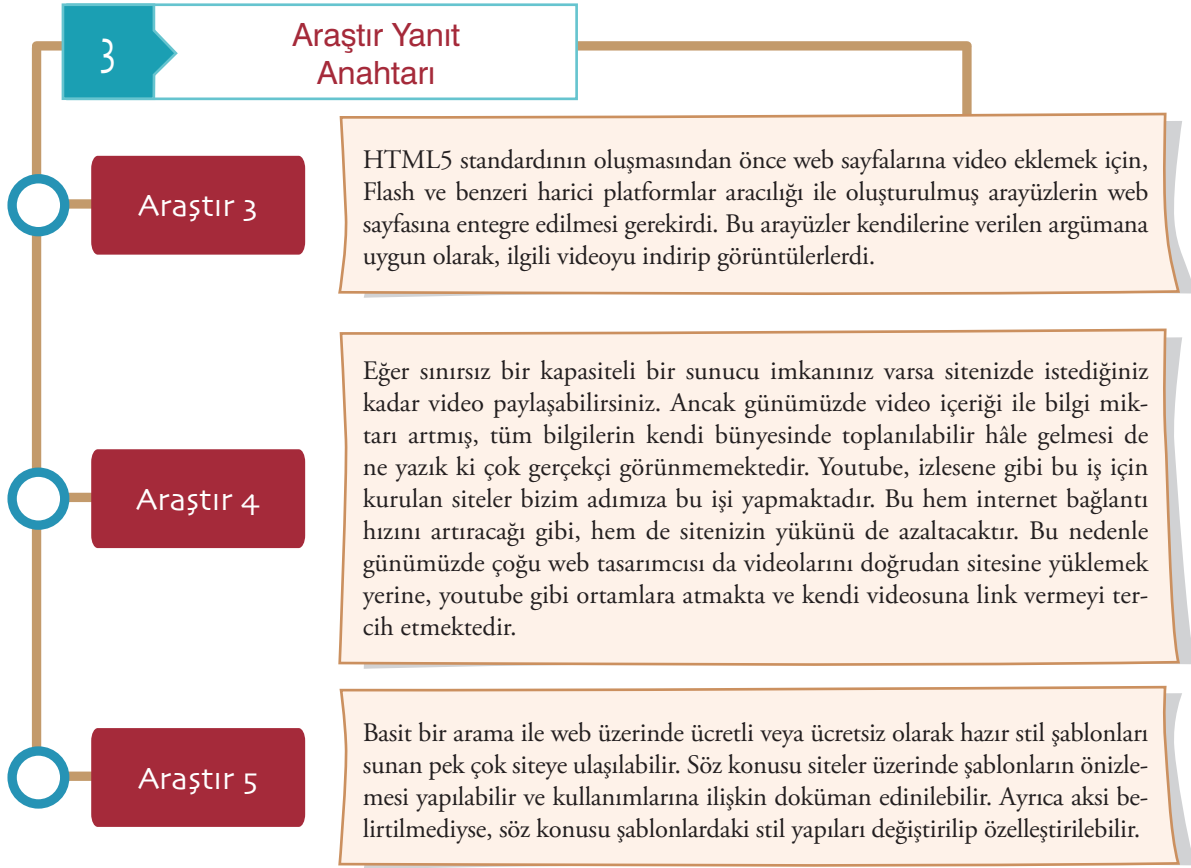
Araştır Yanıt Anahtarı

Araştır 1

<div> etiketi günümüz web sayfalarının en önemli etiketlerinden biridir. Sayfanın istenilen şekil ve düzende parçalara bölünebilmesini, yapısal olarak şekillendirilebilmesini sağlamaktadır. Eskiden <tablo> ile yapılan düzenleme artık <div> ile çok daha etkin bir şekilde yapılabilmektedir. Ayrıca aldığı parametreler ile (id, class, position, width, height, z-index, padding, border, background vb) kolayca şekillendirilebilmekte, seçici desteği sayesinde CSS kullanımını da aktif hâle getirebilmektedir. Çokluortam öğeleri açısından düşünüldüğünde sitede bu öğelerin dağınık bir şekilde durması engellenebileceği gibi, zemin rengi, çizgisi gibi tüm özellikleri de <div> ler aracılığı ile kolayca yapılabilir.

Araştır 2

Resim haritalandırma özelliğini diğer HTML etiketlerini kullanarak taklit etmek oldukça zordur. Ancak benzer bir uygulama, her biri kendi bağlantısına sahip olan farklı resimleri tablo veya bölüm yapıları aracılığıyla kusursuzca birleştirip, çeşitli bağlantılar içeren bölgelere sahip bir resim gibi görünmesi sağlanabilir.



Kaynakça

- Google, (2016). Harita yerleştirme veya konum paylaşma, <https://support.google.com/maps/answer/144361?co=GENIE.Platform%3DDesktop&hl=tr> Erişim Tarihi 01 Temmuz 2016.
- Hayrioğlu, F. (2008). CSS ile Resim Haritaları (imagemap) Yapmak. <http://fatihhayrioglu.com/css-ile-resim-haritalariimagemap-yapmak/> Erişim Tarihi 30 Haziran 2016.
- HTMLdersleri.org (2008). HTML5 Video. http://www.htmldersleri.org/index.php?getir=html5_video&ID=37 Erişim Tarihi 01 Temmuz 2016.
- _____ (2008). HTML5-Semantik Elementler. http://www.htmldersleri.org/index.php?getir=html5_semantic&ID=31 Erişim Tarihi 30 Haziran 2016.
- İşcan A. (2012). Html 5 Nedir? <http://www.ahmetiscan.web.tr/html5-nedir/> Erişim Tarihi 30 Haziran 2016.
- w3schools.com, (2016). HTML Audio/Video DOM Reference http://www.w3schools.com/tags/ref_av_dom.asp Erişim Tarihi 01 Temmuz 2016.

■ Bölüm 4

Tablo Yapılarının Oluşturulması

öğrenme çıktıları

Tabloların Tanımlanması

- 1 Html'de kullanılan tablo etiketlerinin işlevlerini açıklayabilme
- 2 Html'de kullanılan tablo etiketlerini kullanarak tablo oluşturabilme

Tabloların Stil Kullanılarak Biçimlendirilmesi

- 3 Varolan tabloların biçimlerini CSS kullanımıyla düzenleyebilme

Tabloların Bölünmesi

- 4 Tabloların satır ve sütunlarını bölebilmeye

Anahtar Sözcükler: • Html 5 • Tablo Yapısı • Tablo Etiketleri • Tablo Oluşturma • Tablo Biçimlendirme • CSS



GİRİŞ

Tablolar genellikle anlaşılabilirliği desteklemek için verilerin satır ve sütun şeklinde gruplandırılarak sunulmasını sağlayan iki boyutlu yapılardır. Html’de tablo oluşturmak için belli başlı komutlar bulunmaktadır.

✓ Tablolar genellikle anlaşılabilirliği desteklemek için verilerin satır ve sütun şeklinde gruplandırılarak sunulmasını sağlayan iki boyutlu yapılardır.

Bu bölümde çeşitli örnekler üzerinden tablo yapılarını, tablolarda kullanılan etiketleri kavramaya çalışacağız ve tablo etiketlerini ve CSS dosyalarını kullanarak etkili ve kullanışlı tablo oluşturma üzerinde duracağız. Ancak bundan önce temel tablo yapısının hangi bileşenlerden oluştuğunu anlamak için Şekil 4.1’i inceleyelim.

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.1 Tabloların Temel Yapısı

✓ Her tabloda dört temel bileşen bulunmaktadır. Bunlar: Satır, sütun, hücre ve veridir.

Şekil 4.1’de görüldüğü üzere her tabloda dört temel bileşen bulunmaktadır. Bunlar: Satır, sütun, hücre ve veridir. Satır (İng. row) tabloda soldan sağa yan yana geldiğinde bir birey, grup, nesne, lokasyon, şirket vb. anlamlı bir bütüne ait verileri içerir ve çeşitli özellikleri ile bu anlamlı bütünü

(İng. entity) tanımlar. Örneğin Şekil 4.1’de adının Mikail soyadının Ceksin olduğunu bildiğimiz kişinin vize puanının 82 olduğu bilgisini okuyabiliyoruz. Sütun (İng. Column) ise bir satırdan diğer satıra geçebilecek tek bir özelliğe ilişkin verileri içerir. Örneğin yukarıda Mikail, Adem şeklinde değişen ve bireylerin adlarına ilişkin verileri içeren bir sütundur. Yukarıdaki tablodaki her bir dikdörtgen kutucuk bir tablo hücresidir (İng. cell). Her hücrede ise veriler (İng. Data) bulunur. Örneğin ortadaki sütunun üçüncü satırındaki veri “Sendeler”dir.



dikkat

Bu bölümde buradan sonra // işareti ilgili kod satırının açıklanması amacıyla kullanılacaktır. // ifadesinin okuyucuya göre sol tarafı çalışan kod satırını, sağ tarafı ise ilgili kod satırının açıklamasını içermektedir (Örneğin: <title>Tablo Oluşturmayı Öğreniyorum</title> // Web sayfasının en üst bölümünde “Tablo Oluşturmayı Öğreniyorum” başlığının yazılmasını sağlar.)

TABLoların Tanımlanması

Bu bölümde öncelikle temel tablo komutları kullanarak tablonun nasıl tanımlandığı ve verilerin hücrelere nasıl girildiği üzerinde durulacaktır.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4  <meta charset="utf-8"/>
5  <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <link rel="stylesheet" href="tablo_gorunum.css">
7  </head>
8  <body>
9
10 <h1>Merhaba Tablolar!</h1>
11
12 <table style="width:100%">
13 <tr>
14 <td>Adı</td>
15 <td>Soyadı</td>
16 <td>Vize Notu</td>
17 </tr>
18 <tr>
19 <td>Mikail</td>
20 <td>Ceksin</td>
21 <td>82</td>
22 </tr>
23 </table>
24
25 </body>
26 </html>

```

Şekil 4.2’de verilen örnek kodlamadan da anlaşıldığı üzere bundan önceki konulardan aşına olduğunuz temel html yapısı içinde tablonun 12. satırdaki `<table>` etiketi ile açıldığı ve 23. satırdaki `</table>` etiketi ile kapatıldığını görebilirsiniz. Şimdi 12. Satır ile 23. Satır arasındaki etiketleri sırasıyla yorumlayarak tarayıcıdan (İng. browser) ne yapmasını istediğimizi açıklayalım.



dikkat

Kodlama üzerinde yaptığınız değişiklikten sonra kaydetmeyi unutmayın. Aksi hâlde değişimi tarayıcıdan izleyemezsiniz. Ayrıca dosyayı ilk kez kaydederken dosya türü olarak “html” seçmelisiniz.

Şekil 4.2 Temel Tablo Komutlarının Kullanımına İlişkin Örnek Kod

- `<table style="width:100%">` // 12. Satırda, web sayfasını enine (solda-sağa) tamamını kaplayacak (yüzde yüzünü) şekilde bir tablo oluşturuluyor. Burada “width:” işlevi ile genişlik bilgisi 100% olarak gönderiliyor.
- `<tr> ...</tr>` // 13. satırda açılıp 17. satırda kapanan bu etiket **table rows** ifadesinin kısaltmasıdır ve tabloda bir satır oluşturur. 13. Satırda kullanılan `<tr>` etiketi ile tablonun ilk satırı açılıyor.
- `<td>Adı</td>` // **table data** `<td>` etiketi daha önce `<tr>` ile açılan satırda bir hücre oluştur ve `<td>` ile `</td>` arasındaki veriyi ilgili hücreye yazar. Burada 14. Satırdaki `<td>Adı</td>` ifadesi birinci satırın ilk hücresine “Adı” verisini yazıyor. `<td>Soyadı</td>` ilk satırdaki ikinci hücreyi oluşturuyor ve içine “Soyadı” verisini yazıyor, `<td>Vize Notu</td>` ilk satırdaki üçüncü hücreyi oluşturarak içine “Vize Notu” verisini yazıyor.
- `</tr>` // 17. satırdaki etiket ile tablonun ilk satırı kapatılıyor.
- `<tr>` // 18. satırdaki etiket ile tablonun ikinci satırı oluşturuluyor.
- `<td>Mikail</td>` //19. satırdaki ifade ile ikinci satırın birinci hücresine “Mikail” verisi yazılıyor.
- `<td>Ceksin</td>` // 20. satırdaki ifade ile ikinci satırın ikinci hücresine “Ceksin” verisi yazılıyor.
- `<td>82</td>` // 21. satırdaki ifade ile ikinci satırın ikinci hücresine “82” verisi yazılıyor.
- `</tr>` // 22. satırdaki etiket ile ikinci satır kapatılıyor.
- `</table>` // 23. satırdaki ifade ile tablo kapatılıyor.

Şekil 4.2’deki kodlamaya üçüncü satır eklendiğinde ve çalıştırıldığında elde edilen ekran görüntüsü Şekil 4.3’te verilmiştir.

Tablolarda genellikle tablonun ilk satırı tablo başlığı olarak adlandırılır ve ilgili sütundaki verinin türü ve içeriği hakkında bilgi verir. Örneğin, “Vize Notu” ilk satırın son hücresinde yer almaktadır ve o sütunda sunulacak olan verinin her bir öğrenci için vize puanı olacağını belirtmektedir. Bu anlamda ilk satırdaki veri ilgili sütun için “sütun etiketi” olarak nitelendirilebilir. İlgili sütundaki veriyi betimleme özelliği olduğundan dolayı tabloların ilk satırları diğer satırlarına göre farklı yazılırlar. Bu durumu ifade etmek için html’de **table headings** `<th>` `</th>` etiketi kullanılmaktadır.

Şekil 4.2'deki örnekte tanımlanan tablonun ilk satırını tablo başlığı hâline getirmek için Şekil 4.4'teki değişiklikleri gerçekleştirelim. Kodlama çalıştırıldığında Şekil 4.5'teki ekran görüntüsü elde edilmektedir.

Merhaba Tablolar!

Adı	Soyadı	Vize Notu:
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.3 Temel tablo komutlarına ilişkin örnek kodun oluşturduğu ekran görüntüsü



dikkat

Html'de bir satır yerine bir sütunu veya yalnızca belirli hücreleri tablo başlığı olarak tanımlayabilirsiniz.

```

5  <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <link rel="stylesheet" href="tablo_gorunum.css">
7  <Style>
8  </Style>
9  </head>
10 <body>
11
12 <h1>Merhaba Tablolar!</h1>
13
14 <table style="width:100%">
15 <tr>
16 <th>Adı</th>
17 <th>Soyadı</th>
18 <th>Vize Notu</th>
19 </tr>
20 <tr>
21 <td>Mikail</td>
22 <td>Ceksin</td>
23 <td>82</td>
24 </tr>
25 <tr>
26 <td>Adem</td>
27 <td>Sendeler</td>
28 <td>75</td>
29 </tr>
30 </table>
31
32 </body>
33 </html>

```

Şekil 4.4 Tablo Başlıklandırmaya İlişkin Örnek Kod

- **R** <th>Adı</th> // 16. satırdaki ifade tablonun ilk satırının ilk hücresinin tablo başlığı olduğunu belirterek koyu ve hücrenin ortasında yazılmasını sağlıyor.
- <th>Soyadı</th> // 17. satırdaki ifade tablonun ilk satırının ikinci hücresinin tablo başlığı olduğunu belirterek koyu ve hücrenin ortasında yazılmasını sağlıyor.
- <th>Vize Notu</th> // 18. satırdaki tablonun ilk satırının üçüncü hücresinin tablo başlığı olduğunu belirterek koyu ve hücrenin ortasında yazılmasını sağlıyor.

Merhaba Tablolar!

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.5 Tablo Başlıklandırmaya İlişkin Örnek Kodun Oluşturduğu Ekran Görüntüsü

Akdeniz Bölgesi 2015 Yılı İhracat Verileri

Ürün Adı	Ürün Kodu	Miktarı (Ton)
Çilek	001	500
Muz	003	975

Şekil 4.6 Araştır Etkinliği İçin Dikkate Alınacak Ekran Görüntüsü

Öğrenme Çıktısı



- 1 Html'de kullanılan tablo etiketlerinin işlevlerini açıklayabilme
- 2 Html'de kullanılan tablo etiketlerini kullanarak tablo oluşturabilme

Araştır 1

Şekil 4.6'da verilen ekran çıktısını oluşturacak HTML kodlarını yazınız.

İlişkilendir

Birden fazla kısımdan oluşan web sayfalarındaki olası tablo yapılarının nasıl bir HTML yapısıyla oluşturulmuş olabileceğini yorumlayınız.

Anlat/Paylaş

Web sayfasına tablo eklemek için mutlaka kullanılması gereken etiketlerin neler olduğunu açıklayınız.

TABLOLARIN STİL KULLANILARAK BİÇİMLENDİRİLMESİ

Şekil 4.5'teki ekran çıktısından görüldüğü üzere tabloya başlık eklediğimizde otomatik olarak bir biçimlendirme ile başlığı ortalı ve koyu yapmaktadır. Bunun yanı sıra diğer hücrelerdeki veriler ise sola hizalanmıştır. Tablolarda daha önce öğrendiğiniz metin biçimlendirme etiketleriyle bu ve benzeri değişiklikleri yapabilirsiniz. Ancak tablo metinlerinin hizalamasının yapılmasının yanı sıra tabloların kendilerine özgü özellikleri olan tablo kenarlıkları ve kalınlığı, rengi, hücre yüksekliği gibi özelliklerin ayarlanması gerekir. Takdir edersiniz ki web sayfalarında çoğunlukla birden çok tablo kullanılmaktadır. Bu nedenle bir web sayfasındaki tabloların standart bir biçimlendirmeye sahip olmasını gerekli kılan birkaç temel nedenden bahsedilebilir. Bunlar: Sayfanın yorumlanmasını tarayıcının inisiyatifine bırakmamak, kolay yönetilebilirlik ve algılamada kolaylık olarak bilinen tasarım ilkesi olarak sıralanabilir. Bildiğiniz üzere eğer aksi belirtilmezse tarayıcılar kendi ayarlanmış değerlerine göre web sayfalarının biçimlerini yorumlarlar/gösterirler. Web sayfasında bizim biçimlendirme kullandığımız yerler ile kullanmadığımız veya göz ardı ettiğimiz bölümler farklı biçimlerde görüntülenecektir. Oysa bütüncül algılama eğiliminde olan insan zihni için (Schunk, 2011) bu olumsuz bir durumdur ve bu durum zihnin algılamak için ekstra çaba harcanmasına neden olacaktır ve zihinde karmaşa hissi uyandıracaktır. Benzer şekilde Gestalt kuramının benzerlik ilkesine göre insan benzer görüntüleri aynı algılar (Schunk, 2011) bu nedenle daha az bir çaba ile algılama-anlama süreci tamamlanır. Her tabloda farklı biçimlendirme seçenekleri kullanıldığında bu ilke ihlal edilmiş olacaktır, birey ilgili görüntünün tablo olduğunu algılamak için ekstra zihinsel çaba göstermek durumunda kalacaktır. Bunun yanı sıra bir tabloda biçimsel bir değişiklik yaptığınızda örneğin tablo kenarlık kalınlığını değiştirdiğinizde, ilgili satırlara giderek tek tek değişiklik yaparak bütün tablolara değişikliğin yansımaları sağlayabileceksiniz. Bütün bu olumsuzlukları ortadan kaldırmak için önceki bölümlerden de hatırlayacağınız stil tanımlamalarını kullanarak bir tablo biçimlendirmesi elde ettiğinizde, bu ve benzeri olumsuzlukları ortadan kaldırmış olacaksınız.

Tablo Kenarlıklarının Oluşturulması

Şekil 4.7’de verilen örnek kodlamada 7. satırda başlayan `<style>` etiketi ile açılan ve 11. satırda `</style>` etiketi ile kapanan bölümde bundan sonra oluşturacağımız tabloların stillerin nasıl olacağını tanımlıyoruz. 8. satırda başlayan ve 10. satırda `}` işareti ile biten tanımlama ile tablonun (table), tablo başlığının (th), ve hücrelerin etrafına kenarlık çizilmesi talimatı veriliyor.



dikkat

Style bölümünde yapılan belirtiler ile tüm tablolara uygulanacak bir biçimlendirme tanımlanmış olur.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4  <meta charset="utf-8"/>
5  <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <link rel="stylesheet" href="tablo_gorunum.css">
7  <style>
8  table,th, td {
9      border: 1px solid black;
10 }
11 </style>
12 </head>
13 <body>
14
15 <h1>Bilişim Teknolojileri Dersi Vize Sınavı Puanları </h1>
16
17 <table>
18 <tr>
19 <th>Adı</th>
20 <th>Soyadı</th>
21 <th>Vize Notu</th>
22 </tr>
23 <tr>
24 <td>Mikail</td>
25 <td>Ceksin</td>
26 <td>82</td>
27 </tr>
28 <tr>
29 <td>Adem</td>
30 <td>Sendeler</td>
31 <td>75</td>
32 </tr>
33 </table>
34
35 </body>
36 </html>

```

Şekil 4.7 Tablo Kenarlığının Biçimlendirilmesi

Bilişim Teknolojileri Dersi Vize Sınavı Puanları

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.8 Biçimlendirilmiş Tablo Kenarlıkları

Şekil 4.7’de verilen kod çalıştırıldığında Şekil 4.8’deki gibi bir ekran görüntüsü elde edilir. Şekil 4.8’de görüldüğü üzere tablo kenarlıklarının yanı sıra hücre kenarlıklarının tanımlanması ile çift kenarlık görüntüsü oluşmuştur.

Şekil 4.8’de verilen ekran görüntüsünde okunurluğu artırmak için birkaç küçük değişiklik yapılabilir. Bu amaçla Şekil 4.9’da verilen örnek kodlamayı inceleyelim.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4  <meta charset="utf-8"/>
5  <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <link rel="stylesheet" href="tablo_gorunum.css">
7  <Style>
8  table {
9      border-collapse:collapse;
10     width:100%;
11     }
12     th, td {
13         border: 1px solid black;
14     }
15     th { height:50px;
16     }
17     td { height:30px;
18     }
19 </Style>
20 </head>
21 <body>
22
23 <h1>Bilişim Teknolojileri Dersi Vize Sınavı Puanları </h1>
24
25 <table>
26 <tr>
27     <th>Adı</th>
28     <th>Soyadı</th>
29     <th>Vize Notu</th>
30 </tr>
31 <tr>
32     <td>Mikail</td>
33     <td>Ceksin</td>
34     <td>82</td>
35 </tr>
36 <tr>
37     <td>Adem</td>

```

Şekil 4.9 Tablo Kenarlıklarının Biçimlendirilmesine İlişkin Örnek Bir Kod

Şekil 4.9’daki stil tanımlamalarına ilişkin açıklamalar aşağıdaki gibi maddelenebilir.

- <Style> // 7. satırda stil tanımlama başlatılıyor.
- table { // 8. satırda tabloya ilişkin bir stil tanımlaması açılıyor.
 - border-collapse:collapse; //9. satırda kenarlığın kaldırılması sağlanıyor.
 - width:100%; // 10. satırda tablo genişliğinin tarayıcı sayfasının %100’ünü kaplaması sağlanıyor.
 - }// 11. satırda tablo stil tanımlaması kapatılıyor.
- th, td { // 12. satırda tablo başlığı ve hücelere ilişkin bir stil tanımlaması açılıyor.
- border: 1px solid black; // 13. satırda hücre kenarlıklarının 1 piksel kalınlığında sürekli çizgi olarak tanımlanması sağlanıyor.
- }// 14. satırda tablo başlığı ve hücelere ilişkin tanımlama kapatılıyor.

- `th { height:50px; // 15. satırda tablo başlığına ilişkin bir tanımlama satırı açılıyor ve tablo başlıklarının yükseklikleri 50 pixel'e ayarlanıyor.`
- `// 16. satırda tablo başlığına ilişkin açılan tanımlama kapatılıyor.`
- `td { height:30px; // 17. Satırda hücelere ilişkin bir tanımlama satırı açılıyor ve hücre yükseklikleri 30 pixel'e ayarlanıyor.`
- `// 18. satırda hücelere ilişkin olarak açılan tanımlama kapatılıyor.`
- `</Style> // 19. satırda stil tanımlama kapatılıyor.`

Şekil 4.9'da verilen örnek kodlama çalıştırıldığında Şekil 4.10'daki ekran görüntüsü elde edilir.

Bilişim Teknolojileri Dersi Vize Sınavı Puanları

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.10 Tablo Kenarlıklarını Biçimlendiren Örnek Kodun Oluşturduğu Ekran Görüntüsü

Tablolarda Veri Yerleşimi

Tablolarda verilerin hücrelerin içine nasıl yerleşeceği ile ilgili üç temel özellikten bahsedebiliriz bunlardan birincisi hücre kenarlığı çizgileri arasındaki boşluk **"border-spacing"** ile, ikincisi hücre kenarlığı ile veri arasında bırakılacak olan boşluk (örneğin Şekil 4.10'daki çıktıyı incellerseniz "Adem" veya "75" gibi hücre içerisindeki verilerin hücre kenarlığına bitişik olarak görüldüğünü fark edebilirsiniz) **"padding"** ile, üçüncüsü verinin hizalanması (sola, sağa, ortali) **"text-align"** ile belirtilir. Bu üç işlevi daha iyi anlamak için Şekil 4.11'de verilen örnek kodlamayı inceleyelim.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4  <meta charset="utf-8"/>
5  <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <link rel="stylesheet" href="tablo_gorunum.css">
7  <Style>
8  table {
9      border: 1px solid black;
10     border-spacing:10px;
11     width:100%;
12 }
13 th, td {
14     border: 2.5px dashed red;
15     padding-left:5px;
16 }
17 th { height:50px;
18     text-align:left
19 }
20 td { height:30px;
21 }
22 </Style>
23 </head>
24 <body>

```

Şekil 4.11 Tabloda Veri Yerleşimine İlişkin Örnek Kod

- `table { // 8. satırda tablo için tanımlama açılıyor.`
 - `border: 1px solid black; // tabloya 1 pixel kalınlığında siyah sürekli çizgiden oluşan bir kenarlık tanımlanıyor.`
 - `border-spacing: 10px; // tablo kenarlığı ile hücre arasındaki boşluk 10 pixel olarak tanımlanıyor.`
 - `width: 100%; // tablo genişliği web sayfasının enine tamamını kaplayacak şekilde ayarlanıyor`
 - `} // tablo için tanımlama kapatılıyor.`
- `th, td { // tablo başlığı ve hücreler için tanımlama satırı açılıyor.`
- `border: 2.5px dashed red; // tablo başlığı ve hücrelere 2.5 pixel kalınlığında kırmızı kesik çizgili kenarlık tanımlanıyor.`
 - `padding-left: 5px; // tablonun ilk satırı ve diğer hücrelere yazılacak verilerin sol kenarlık 5 pixel boşluk bırakılarak yazılması sağlanıyor.`
 - `}tablo başlığı ve hücreler için tanımlama satırı kapatılıyor.`
- `th { height: 50px; // tablo başlığı için tanımlama satırı açılıyor.`
 - `text-align: left; // tablo başlığı içindeki verinin sola hizalanması sağlanıyor.`
 - `} // tablo başlığı için tanımlama satırı kapatılıyor.`
- `td { height: 30px; // hücreler için tanımlama satırı açılıyor ve hücre yüksekliği 30 pixel olarak ayarlanıyor.`
 - `} // hücreler için tanımlama satırı kapatılıyor.`



dikkat

Eğer aksi belirtilmezse tablo başlığı içindeki veri, html'nin tanımlı değerlerine göre pek çok tarayıcı tarafından hücre içerisinde ortalı hizalanmış ve kalın yazılı olarak yorumlanır.

Şekil 4.11'de Tablolarda veri yerleşimi ile ilgili olarak verilen örnek kodlamayı çalıştırdığınızda Şekil 4.12'de verilen ekran görüntüsünü elde edeceksiniz.

Bilişim Teknolojileri Dersi Vize Sınavı Puanları

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.12 Tabloda Veri Yerleşimine İlişkin Örnek Kodun Ürettiği Ekran Görüntüsü

Tablo Formatlarının Düzenlenmesi

Tablolar için stil tanımlaması yapılmasının temel nedeni tüm web sayfası boyunca aynı formatta tablo kullanarak bütünlüğü sağlamaktır. Ancak bazı durumlarda iki veya daha farklı formatta tablo kullanılması gerekebilir. Bu tür durumlar için “id attribute” özelliği sayesinde tablolara kimlik numarası verilerek istenilen tablolara farklı özelliklerin eklenmesi sağlanır. Normalde bir tablo için stil tanımlaması satırı açmak için “table {” şeklinde başlayıp tanımlamalar gerçekleştiriliyordu, istediğimiz zaman çağırarak kullanabileceğimiz bir tablo özelliği eklemek istediğimizde “table” belirtiminden sonra “#” işareti ve bir etiket, örneğin “t01” belirtmemiz gerekmektedir. Şekil 4.13'te verilen örnekte iki farklı formatta tanımlanan tablo kodlarının ne ifade ettiğini inceleyelim.



“id attribute” özelliği sayesinde tablolara kimlik numarası verilerek istenilen tablolara farklı özelliklerin eklenmesi sağlanır.

```

7 <Style>
8   table, th, td {
9     border: 1px solid black;
10    border-collapse: collapse;
11  }
12  th, td {
13    padding-left: 5px;
14    text-align: left;
15  }
16  table#t01 {
17    width: 100%;
18    background-color: yellow;
19  }
20 </Style>
21 </head>
22 <body>
23 <h1>Bilişim Teknolojileri Dersi Final Sınavı Puanları </h1>
24
25 <table id="t01">
26   <tr>
27     <th>Adı</th>
28     <th>Soyadı</th>
29     <th>Final Notu</th>
30   </tr>
31   <tr>
32     <td>Mikail</td>
33     <td>Ceksin</td>
34     <td>90</td>
35   </tr>
36   <tr>
37     <td>Adem</td>
38     <td>Sendeler</td>
39     <td>88</td>
40   </tr>
41 </table>
42 <h1>Bilişim Teknolojileri Dersi Vize Sınavı Puanları </h1>
43
44 <table style="width:100%">
45   <tr>
46     <th>Adı</th>
47     <th>Soyadı</th>
48     <th>Vize Notu</th>
49   </tr>
50   <tr>
51     <td>Mikail</td>
52     <td>Ceksin</td>
53     <td>82</td>
54   </tr>
55   <tr>
56     <td>Adem</td>

```

Şekil 4.13 Tablo Formatının Değiştirilmesine İlişkin Örnek Kod

- table, th, td { // 8. satırda tablo, tablo başlığı ve hücreler için tanımlama satırı açılıyor
- border: 1px solid black; // 9. satırda tablo, tablo başlığı, ve hücre kenarlıkları 1 pixel sürekli-düz siyah çizgi olarak tanımlanıyor.

- border-collapse: collapse; // 10. satırda kenarlıklardan biri çıkartılarak tablo tek kenarlığa düşürülüyor.
- } // 11. satırda tablo, tablo başlığı ve hücrelere ilişkin tanımlama kapatılıyor.
- th, td { // 12. satırda tablo başlığı ve hücrelere ilişkin tanımlama açılıyor.
- padding-left:5px; // 13. satırda hücre başlığı ve hücre içindeki verilerin soldan 5 pixel boşluk kalacak şekilde yerleşimi sağlanıyor.
- text-align:left; // 14. satırda tablo başlığı ve hücreler içindeki verilerin sola hizalanması sağlanıyor.
- } // 15. satırda tablo başlığı ve hücrelere ilişkin tanımlama kapatılıyor.
- table#t01 { // 16. satırda tabloya istenildiğinde kullanılmak üzere t01 özelliği tanımlanıyor.
- width: 100%; // 17. satırda tablo genişliği web sayısının tamamını kaplayacak şekilde ayarlanıyor.
- background-color: yellow; // 18. satırda tablonun arka plan rengi sarı olarak ayarlanıyor.
- } // 19. satırda tanımlama kapatılıyor.

Stil bölümündeki tanımlamalardan sonra body içinde ilgili tablo özelliğinin nasıl çağırıldığına bakacak olursak, kodlamamızda iki tablo bulunduğu görülecektir. 25. Satırda <table id="t01"> // ifadesi ile aşağıda oluşturulacak tabloyu stil bölümünde tanımlanan tablo, tablo başlığı ve hücre özelliklerine ek olarak "t01" özelliği ile kullanmak istediğimizi belirtiyoruz.

İkinci tablomuz ise daha önce gördüğümüz şekilde sadece table, th ve td ile ilgili olarak stil bölümünde yapılan tanımlamaları kullanmaktadır. Ancak birinci tablo table, th, td ile ilgili olarak yapılan tanımlamalara ek olarak t01 özelliğini de kullanmaktadır. Şekil 4.14'te verilen ekran görüntüsü farklı tablolara farklı özelliklerin nasıl eklendiğine ilişkin olarak Şekil 4.13'te verilen örnek kodlamanın ekran çıktısını yansıtmaktadır.



HTML standart renk kodlarına ulaşmak için http://www.w3schools.com/colors/colors_hexadecimal.asp bağlantısından yararlanabilirsiniz.

Bilişim Teknolojileri Dersi Final Sınavı Puanları

Adı	Soyadı	Final Notu
Mikail	Ceksin	90
Adem	Sendeler	88

Bilişim Teknolojileri Dersi Vize Sınavı Puanları

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.14 Tablo Formatının Değiştirilmesine İlişkin Örnek Kodun Ürettiği Ekran Görüntüsü

Tabloları birbirinden farklı şekilde biçimlendirirken çoğu zaman tablonun arka plan rengini değiştirmekten daha fazlasını yapmak isteriz. Örneğin, tablo başlıklarının vurgulamasının farklı olması ya da tablodaki tek veya çift numaralı satır renklerinin farklı olması bunlar arasında düşünülebilir. Bu amaçla hazırlanan Şekil 4.15'teki örnek kodlamayı inceleyelim.


```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4    <meta charset="utf-8"/>
5    <title>Tablo Oluşturmayı Öğreniyorum</title>
6    <link rel="stylesheet" href="tablo_gorunum.css">
7    <Style>
8      table, th, td {
9        border: 1px solid black;
10       border-collapse: collapse;
11     }
12     th, td {
13       padding-left: 5px;
14       text-align: left;
15     }
16     th { height: 50px;
17     }
18     table#t01 {
19       width: 100%;
20     }
21     table#t01 th {
22       background-color: black;
23       color: white;
24     }
25     table#t01 tr:nth-child(odd) {
26       background-color: #9ACD32;
27     }
28     table#t01 tr:nth-child(even) {
29       background-color: #F5DEB3;
30     }
31   </Style>
32 </head>
33 <body>
34   <h1>Bilişim Teknolojileri Dersi Final Sınavı Puanları </h1>
35
36   <table id="t01">
37     <tr>
38       <th>Adı</th>
39       <th>Soyadı</th>
40       <th>Final Notu</th>
41     </tr>
42     <tr>
43       <td>Mikail</td>
44       <td>Ceksin</td>
45       <td>90</td>
46     </tr>
47     <tr>
48       <td>Adem</td>
49       <td>Sendeler</td>
50       <td>88</td>

```

Şekil 4.15 Tablolara Yeni Özellikler Eklenmesine İlişkin Örnek Kod

- table, th, td { // 8. satırda tablo, tablo başlığı ve hücreler için tanımlama açılıyor.
 - border: 1px solid black; // 9. satırda kenarlık tanımlanıyor.
 - border-collapse: collapse; // 10. satırda tablo tek kenarlığa düşürülüyor.
 - } // 11. satırda tanımlama kapatılıyor.

- th, td { // 12. satırda tablo başlığı ve hücreler için tanımlama açılıyor.
 - padding-left:5px; // 13. satırda verilerin soldan 5 pixel boşluk bırakılarak yazılması sağlanıyor.
 - text-align:left; // 14. satırda verilerin sola hizalanmış olarak yazılması sağlanıyor.
 - } // 15. Satırda tanımlama kapatılıyor.
- th { height:50px; // 16. satırda tablo başlığı için tanımlama açılıyor ve tablo başlığı satırının yüksekliği 50 pixel olarak ayarlanıyor.
 - } // 17. satırda tanımlama kapatılıyor.
- table#t01 { // 18. satırda “t01” ek özelliği için tanımla açılıyor.
 - width: 100%; // 19. satırda tablo genişliğinin web sayfasının tamamını kaplaması sağlanıyor.
 - } // 20. satırda tanımlama kapatılıyor.
- table#t01 th { // 21. satırda tablo başlığı için “t01” ek özelliği açılıyor.
 - background-color: black; // 22. satırda tablo başlığının arka plan rengi siyah olarak ayarlanıyor.
 - color:white; // 23. satırda yazı rengi beyaz olarak ayarlanıyor.
 - } // 24. satırda tanımlama kapatılıyor.
- table#t01 tr:nth-child(odd) { // 25. satırda tablo satırı için “t01” ek özelliği açılıyor ve “odd” ifadesi ile tablonun tek numaralı (1. Satır, 3. Satır, 5. Satır, vb.) satırları için ek özellik tanımlanacağı belirtiliyor.
 - background-color: #9ACD32; // 26. satırda kodu verilen (sarı-yeşil) renk tek numaralı satırlar için arka plan rengi olarak ayarlanıyor.
 - } // 27. satırda tanımlama kapatılıyor.
- table#t01 tr:nth-child(even) { // 28. satırda çift numaralı satırlar için “t01” ek özelliği tanımlanıyor.
 - background-color: #F5DEB3; // 29. satırda çift numaralı satırlar için arka plan rengi buğday rengi (İng. Wheat) olarak ayarlanıyor.
 - } // 30. satırda tanımlama kapatılıyor.

<body> bölümündeki her iki tabloya da ikişer satır daha eklenerek çalıştırıldığında Şekil 4.16'daki ekran görüntüsü elde edilir.



dikkat

html'de kullanılan renklerin hegz-desimal kodlarına karşılık gelen isimleri öğrenmek için http://www.w3schools.com/colors/colors_names.asp bağlantısından yararlanabilirsiniz.

Bilişim Teknolojileri Dersi Final Sınavı Puanları

Adı	Soyadı	Final Notu
Mikail	Ceksin	90
Adem	Sendeler	88
Danyal	Kirek	76
Bilal	Kapılar	70

Bilişim Teknolojileri Dersi Vize Sınavı Puanları

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75
Danyal	Kirek	66
Bilal	Kapılar	60

Şekil 4.16 Tablolara Yeni Özellik Eklenmesine İlişkin Örnek Kodun Oluşturduğu Ekran Görüntüsü

Bilişim Teknolojileri Dersi Final Sınavı Puanları

Adı	Soyadı	Final Notu
Mikail	Ceksin	90
Adem	Sendeler	88

Bilişim Teknolojileri Dersi Vize Sınavı Puanları

Adı	Soyadı	Vize Notu
Mikail	Ceksin	82
Adem	Sendeler	75

Şekil 4.17 Araştır Etkinliği İçin Dikkate Alınacak Ekran Görüntüsü

**Öğrenme Çıktısı**

3 Varolan tabloların biçimlerini CSS kullanımıyla düzenleyebilme

Araştır 2

Şekil 4.17’de verilen ekran çıktısını oluşturacak HTML kodlarını yazınız.

İlişkilendir

Ziyaret ettiğiniz web sayfalarında yer alan tabloların dolgu, çizgi biçimi, çizgi rengi gibi özelliklerinin büyük bir çeşitliliğe sahip olabilmesinin nasıl sağlandığını yorumlayınız.

Anlat/Paylaş

Stil özelliklerinin web tablolarıyla nasıl bütünleştirildiğini açıklayınız.

TABLORARIN BÖLÜNMESİ

Tabloları biçimlendirirken gereksinim duyulan başlıca konulardan birisi de tablo satır ve sütunlarının bölünmesidir. Bu amaçla iki önemli özellik kullanılmaktadır bunlar: colspan ve rowspan’dır. “colspan” sütunların bölünmesi için kullanılırken “rowspan” satırların bölünmesi için kullanılmaktadır. Öncelikle colspan’ın işlevini kavramak için Şekil 4.18’de verilen örnek kodlamayı inceleyelim.



Tablolarda “colspan” sütunların bölünmesi için kullanılırken, “rowspan” satırların bölünmesi için kullanılmaktadır.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4    <meta charset="utf-8"/>
5    <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <style>
7    table, th, td {
8      border: 1px solid black;
9      border-collapse: collapse;
10   }
11   th, td {
12     padding: 5px;
13     text-align: left;
14   }
15 </style>
16 </head>
17 <body>
18 <table style="width:100%">
19   <caption>Tablo 1. Kişisel Bilgiler</caption>
20   <tr>
21     <th>Adı Soyadı</th>
22     <th colspan="2">Telefon Numarası</th>
23   </tr>
24   <tr>
25     <td>Danyal Kirek</td>
26     <td>888 00 001</td>
27     <td>777 00 002</td>
28   </tr>
29 </table>
30
31 </body>
32 </html>

```

Şekil 4.18 Sütunların Bölünmesi

Şekil 4.18’de <body> ...</body> bölümünde tablo yapısı oluşturulurken bir sütunun ikiye bölünmesi işlemi gerçekleştirilmiştir. İlgili bölüm aşağıda açıklanmıştır.

- <table style="width:100%"> // 18. satırda web sayfasının tamamını enine kaplayacak tablo tanımlanıyor.
- <caption>Tablo 1. Kişisel Bilgiler</caption> // 19. satırda tablo başlığı tanımlanıyor.
- <tr> // 20. satırda tabloya satır ekleniyor.
 - <th>Adı Soyadı</th> // 21. satırda tablonun birinci satırı tablo başlığı olarak tanımlanıyor ve ilk hücreye “Adı Soyadı” verisi giriliyor.
 - <th colspan="2">Telefon Numarası</th> // 22. satırda ilk satır ikinci hücreye “Telefon Numarası” verisi giriliyor ve Telefon numarası ile başlayan sütun iki sütuna bölünüyor.
- </tr> // 23. satırda tablonun ilk satırı kapatılıyor.
- <tr> // 24. satırda tablonun ikinci satırı açılıyor.
 - <td>Danyal Kirek</td> // 25. satırda tablonun ikinci satır ilk hücreye “Danyal Kirek” verisi giriliyor.
 - <td>888 00 001</td> // 26. satırda tablonun ikinci satır ikinci hücreye “888 00 001” verisi giriliyor.
 - <td>777 00 002</td> // 27. satırda tablonun ikinci satır üçüncü hücreye “777 00 002” verisi giriliyor.

- `</tr>` // 28. satırda tablonun ikinci satırı kapatılıyor.
- `</table>` // 29. satırda tablo kapatılıyor.

Şekil 4.18'deki örnek kodlama çalıştırıldığında Şekil 4.19'da verilen ekran çıktısı elde edilmektedir.

Tablo1. Kişisel Bilgiler

Adı Soyadı	Telefon Numarası	
Danyal Kirek	888 00 001	777 00 002

Şekil 4.19 Sütunların Bölünmesine İlişkin Örnek Kod Parçasının Ürettiği Ekran Görüntüsü

Tablolarda rowspan işlevini kullanarak satırların nasıl bölündüğünü kavramak için Şekil 4.20'de verilen örnek kodlamayı inceleyelim.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4    <meta charset="utf-8"/>
5    <title>Tablo Oluşturmayı Öğreniyorum</title>
6    <style>
7      table, th, td {
8        border: 1px solid black;
9        border-collapse: collapse;
10     }
11     th, td {
12       padding: 5px;
13       text-align: left;
14     }
15   </style>
16 </head>
17 <body>
18
19   <table style="width:100%">
20     <caption>Tablo 1. Kişisel Bilgiler</caption>
21     <tr>
22       <th>Adı Soyadı</th>
23       <td>Danyal Kirek</td>
24     </tr>
25     <tr>
26       <th rowspan="2">Telefon Numarası</th>
27       <td>888 00 001</td>
28     </tr>
29     <tr>
30       <td>777 00 002</td>
31     </tr>
32   </table>
33
34 </body>
35 </html>

```

Şekil 4.20 Satırların Bölünmesine İlişkin Örnek Kod

Şekil 4.20'de `<body> ...</body>` bölümünde tablo yapısı oluşturulurken bir satırın ikiye bölünmesi işlemi gerçekleştirilmiştir. Aşağıda ilgili bölüm açıklanmıştır.

- `<table style="width:100%">` // 19. satırda web sayfasının tamamını enine kaplayacak tablo tanımlanıyor.
- `<caption>Tablo 1. Kişisel Bilgiler</caption>` // 20. satırda tablo başlığı tanımlanıyor.
- `<tr>` // 21. satırda tabloya satır ekleniyor.
 - `<th>Adı Soyadı</th>` // 22. satırda tablonun ilk satır ilk hücresi tablo başlığı olarak tanımlanıyor ve içine “Adı Soyadı” verisi giriliyor.
 - `<td>Danyal Kirek</td>` // 23. satırda tablonun ilk satırı ikinci hücreğine “Danyal Kirek” verisi giriliyor.
- `</tr>` // 24. satırda tablonun ilk satırı kapatılıyor.
- `<tr>` // 25. satırda tablonun ikinci satırı açılıyor.
 - `<th rowspan="2">Telefon Numarası</th>` // 26. satırda tablonun ikinci satırı ilk hücresi tablo başlığı olarak tanımlanıyor ve bir sonraki hücrenin iki satıra bölünmesini sağlıyor
 - `<td>888 00 001</td>` // 27. satırda ikinci satır ikinci hücrede bölünen ilk hücre “888 00 001” verisi giriliyor.
- `</tr>` // 28. satırda tablo satırı kapatılıyor.
- `<tr>` // 29. satırda tablo satırı açılıyor.
 - `<td>777 00 002</td>` // 30. satırda tablonun 3. satırının ilk hücresi olan daha önceden bölünen ikinci hücreye “777 00 002” verisi giriliyor.
- `</tr>` // 31. satırda tablo satırı kapatılıyor.
- `</table>` // 32. satırda tablo kapatılıyor.

Şekil 4.20'deki örnek kodlama çalıştırıldığında Şekil 4.21'deki ekran görüntüsü elde edilmektedir.

Tablo1. Kişisel Bilgiler

Adı Soyadı	Danyal Kirek
Telefon Numarası	888 00 001
	777 00 002

Şekil 4.21 Satırların Bölünmesine İlişkin Örnek Kod

Tablo1. Kişisel Bilgiler

Adı Soyadı	Muhammed Ali
Telefon Numarası	999 33 033
	777 33 000

Şekil 4.22 Araştır Etkinliği İçin Dikkate Alınacak Ekran Görüntüsü





araştırmalarla ilişkilendir

HTML Tablolarından Hiyerarşik Verilerin Elde Edilmesi

HTML tabloları genellikle verileri hiyerarşik bir şekilde ve diğer verilerden izole bir şekilde tutmakta ve veri sunumu için katı olmayan kurallar tanımlamaktadır. Bu yapı, araştırmacıları web tablolarından otomatik olarak veri çekme süreçlerini inceleme konusunda güdülemiştir. Seung-Jin ve Yiu-Kai Ng (1999) tarafından yapılan çalışmada, iç yapısı önceden bilinmeyen bir tablonun içerik ağacının otomatik olarak oluşturulması sağlanmıştır. Araştırmacılara göre

söz konusu uygulama, web tablosu kaynaklarına ilişkin bir sorgulama dilinin oluşturulması, html tablolarının XML ile hazırlanmış veri yığınlarına dönüştürülmesi ve tablo verilerinin veritabanlarına aktarılması için kullanılma potansiyeli taşımaktadır.

Kaynak: Lim, Seung-Jin ve Yiu-Kai Ng. "An automated approach for retrieving hierarchical data from HTML tables." Proceedings of the eighth international conference on Information and knowledge management. ACM, 1999.



Öğrenme Çıktısı

4 Tabloların satır ve sütunlarını bölebilmeye

Araştır 3

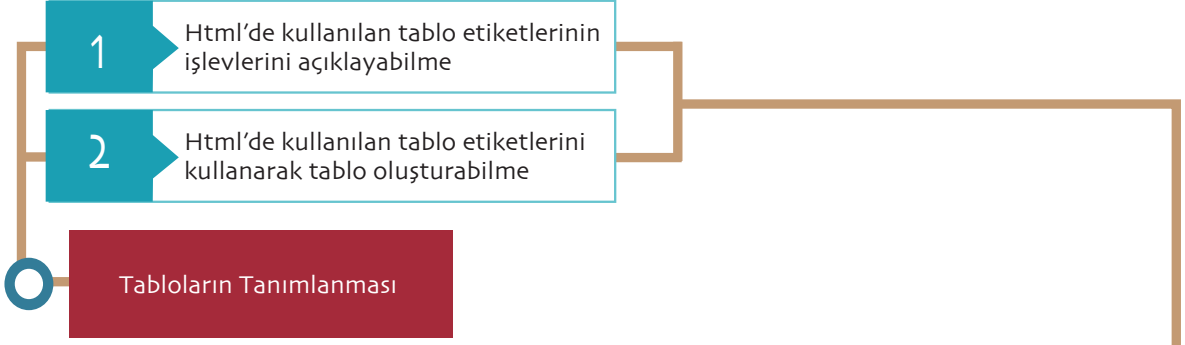
Renkler konusunda http://www.w3schools.com/colors/colors_names.asp bağlantısından yardım alarak Şekil 4.22'de verilen ekran çıktısını elde edecek HTML kodlarını yazınız.

İlişkilendir

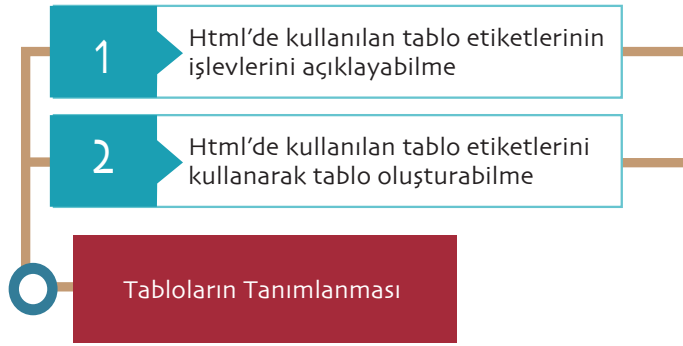
Web tablolarındaki birleştirme ve bölme işlemlerinin ofis yazılımlarındaki karşılıklarıyla ilişkilendiriniz.

Anlat/Paylaş

HTML'deki "colspan" ve "rowspan" deyimlerinin benzerlik ve farklarını açıklayınız.



- 1 Html'de tabloların oluşturulması ve düzenlenmesi için kullanılacak temel etiketler ve işlevleri aşağıdaki gibi maddelenebilir.
 - <table> elementini tabloları tanımlamak için kullanırız.
 - <tr> elementini tablo satırlarını tanımlamak için kullanırız.
 - <td> elementini tablo verilerini tanımlamak için kullanırız.
 - <th> elementini tablo başlıklarını tanımlamak için kullanırız.
 - <caption> elementini tabloya başlığı yazmak için kullanırız.
 - CSS'DE border özelliğini kenarlıkları tanımlamak için kullanırız.
 - CSS'DE border-collapse özelliğini hücre kenarlıklarını çıkartmak için kullanırız.
 - CSS'DE padding özelliğini hücre ile veri arasında bırakılacak boşluğu ayarlamak için kullanırız.
 - CSS'DE text-align özelliğini verinin hücre içerisinde hizalanması sağlamak için kullanırız.
 - CSS'DE border-spacing özelliğini hücreler arasındaki boşluğu ayarlamak için kullanırız.
 - colspan özelliğini kolanları bölmek için kullanırız.
 - rowspan özelliğini satırları bölmek için kullanırız.
 - id attribute özelliğini tablolara ek özellik eklemek için kullanırız.

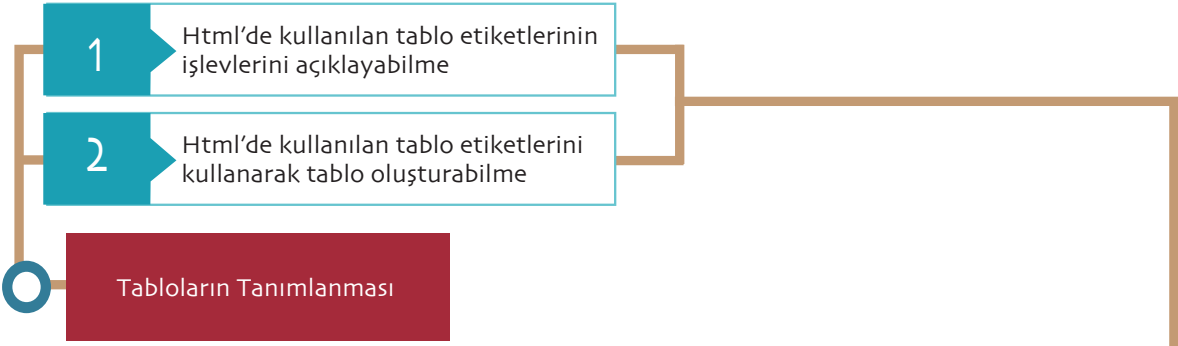


2 Şekil 4.23'te verilen örnek kodlama üzerinden tablo oluşturma ile ilgili temel elementleri ve işlevlerini hatırlamaya çalışalım.

```

2  <html lang="tr">
3  <head>
4    <meta charset="utf-8"/>
5    <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <style>
7    table, th, td {
8      border: 1px solid black;
9      border-collapse: collapse;
10   }
11   th, td {
12     padding: 5px;
13     text-align: left;
14   }
15 </style>
16 </head>
17 <body>
18
19 <table style="width:100%">
20   <caption>Tablo 1. Kişisel Bilgiler</caption>
21   <tr>
22     <th>Adı Soyadı</th>
23     <td colspan=2>Danyal Kirek</td>
24
25   </tr>
26   <tr>
27     <th rowspan="2">Telefon Numarası</th>
28     <td>888 00 001</td>
29     <td>888 00 033</td>
30   </tr>
31   <tr>
32     <td>777 00 002</td>
33     <td>777 00 033</td>
34   </tr>
35 </table>
36 </body>
37 </html>
  
```

Şekil 4.23 Tablo Elementleri ve Stilleri



2 Şekil 4.23'te verilen örnek kodlamada <body> bölümünde tablonun nasıl oluşturulduğunu inceleyelim.

- <table style="width:100%"> // 19. satırda tablo tanımlanıyor ve genişliği web sayfasının tamamını kapsayacak şekilde ayarlanıyor.
- <caption>Tablo 1. Kişisel Bilgiler</caption> // 20. satırda tabloya başlık ekleniyor.
- <tr> // 21. satırda tabloya satır ekleniyor.
- <th>Adı Soyadı</th> // 22. satırda tablo başlığı olarak tanımlanan hücreye "Adı Soyadı verisi giriliyor.
- <td colspan=2>Danyal Kirek</td> // 23. satırda tablonun birinci satır ikinci hücreye "Danyal Kirek" verisi giriliyor ve bir alt satırda tablo iki sütuna bölünüyor.
- </tr> // 25. satırda birinci satır kapatılıyor.
- <tr> // 26. satırda ikinci satır açılıyor.
- <th rowspan=2>Telefon Numarası</th> // 27. satırda tablo başlığı olarak belirlenen ikinci satır birinci hücreye "Telefon Numarası verisi giriliyor ve bir sonraki hücre iki satıra bölünüyor.
- <td>888 00 001</td> // 28. satırda daha önce 23. satırda iki sütuna bölünen hücre ilk sütununa "888 00 01" verisi giriliyor.
- <td>888 00 033</td> // 29. satırda daha önce 23. satırda iki sütuna bölünen hücrenin ikinci sütununa "888 00 033" verisi giriliyor.
- </tr> // 30. satırda ikinci satır kapatılıyor
- <tr> // 31. satırda 27 satırda rowspan ile ikiye bölünen satır daha önceden 23. satırda colspan ile de iki sütuna bölündüğünden burada ilgili hücrelere veri yazmak için ek satır açılıyor.
- <td>777 00 002</td> // 32. satırda ek olarak açılan tablo satırının birinci sütun hücreğine "777 00 002"
- <td>777 00 033</td> // 33. satırda ek olarak açılan tablo satırının ikinci sütun hücreğine "777 00 033"
- </tr> // 34. satırda tablonun ek satırı kapatılıyor
- </table> // 25. satırda tablo kapatılıyor.

Şekil 4.23'te verilen örnek kodlamanın ekran çıktısı Şekil 4.24'te görülmektedir.

3

Varolan tabloların biçimlerini CSS kullanımıyla düzenleyebilme

Tabloların Stil Kullanılarak Biçimlendirilmesi

Şekil 4.20’de verilen örnek uygulama üzerinden tablolara ilişkin stil tanımlama işlemlerini hatırlamaya çalışalım. Şekil 4.20’de stil tanımlama bölümünün 6.satırda `<style>` başlayıp `</style>` ile bittiğini görüyoruz. Stil bölümünde tablo ile ilgili olan kısımları inceleyelim:

- `table, th, td { // 7. satırda tablo, tablo başlığı ve hücreler için stil tanımlaması açılıyor.`
- `border: 1px solid black; // 8. satırda 1 pixel kalınlığında kenarlık tanımlanıyor.`
- `border-collapse: collapse; // 9. satırda tablo kenarlığı tek kenarlığa düşürülüyor.`
- `} // 10. satırda tablo, tablo başlığı ve hücreler için açılan stil tanımlaması kapatılıyor.`
- `th, td { // 11. satırda tablo başlığı ve hücreler için tanımlama satırı açılıyor.`
- `padding: 5px; // 12. satırda hücre içerisindeki veri alanı ile hücre çerçevesi arasında 5 pixel boşluk bırakılması sağlanıyor.`
- `text-align: left; // 13. satırda tablo başlığında ve hücre içerisinde verinin sola hizalanması sağlanıyor.`
- `} // 14. satırda tablo başlığı ve hücreler için açılan tanımlama kapatılıyor.`

Tablo1. Kişisel Bilgiler

Adı Soyadı	Danyal Kirek	
Telefon Numarası	888 00 001	888 00 033
	777 00 002	777 00 033

Şekil 4.24 Tablo Elementleri ve Stiller Ekran Çıktısı

4

Tabloların satır ve sütunlarını bölebilme

Tabloların Bölünmesi

HTML tablolarının satır ve sütunlarının birleştirilmesi ya da bölünmesi için kullanılabilecek iki önemli özellik tanımlaması bulunmaktadır. “`colspan`” adlı özellik, sütunlara ilişkin işlem yaparken, “`rowspan`” adlı özellik satırlara ilişkin işlem yapar. Örneğin `<td colspan="2">` şeklindeki bir hücre tanımlaması, söz konusu hücrenin iki hücre kapsayacak yükseklikte olmasını sağlar. `<td rowspan="2">` tanımlaması, hücrenin iki hücre kapsayacak genişlikte olmasını sağlar. Bu iki özellik örneklerde sunulduğu üzere yalnız başlarına kullanılabildiği gibi, birlikte de kullanılabilirler.

1 <th>Adres</th> ifadesi nasıl yorumlanır?

- A. Hücreye “Adres” verisini yaz
- B. Hücreye “Adres” verisi yaz ve tablo başlığı olarak belirle
- C. Hücreye “Adres” verisi yaz ve sütun başlığı olarak belirle
- D. Hücreye “Adres” verisini kalın ve italik olarak yaz
- E. Hücreye “Adres” verisi yaz ve kenarlık ekleme

2 table#t07 tr { ifadesi nasıl yorumlanır?

- A. Tablo satırları ile ilgili t07 numaralı işlemi gerçekleştir.
- B. Tablo da t07 numaralı satır ile ilgili tanımlama işlemi başlat.
- C. Tablo satırları ile ilgili t07 isminde bir ek özellik tanımlaması başlat.
- D. t07 özelliğinin tablo satırlarından çıkartılması ile ilgili bir işlem başlat.
- E. t07 ek özelliği dışındaki işlemlerin tablo satırlarında uygulanması ile ilgili bir işlem başlat.

3 background-color: blue; ifadesi nasıl yorumlanır?

- A. Arka-plan rengini mavi yap.
- B. Yazı rengini mavi yap.
- C. Sadece hücre içerisindeki rengi mavi yap
- D. Arka-plan rengini siyah yap
- E. Arka-plan rengini hücre başlığında mavi yap.

4 Aşağıdakilerden hangisinde stil tanımlaması yapılmaktadır?

- A. <td>Mikail</td>
- B. <h1>Bilişim Teknolojileri Dersi Vize Sınavı Puanları </h1>
- C. th { height:50px;}
- D. </Style>
- E. <table id="t01">

5 Aşağıdakilerden hangisinde body bölümünde bir tablo ek özelliğinin kullanılması söz konusudur?

- A. th, td {padding-left:5px;}
- B. table#t10 {width: 100%;}
- C. table#t10 th { background-color: black;}
- D. table#t10 th { color:white;}
- E. <table id="t33">

6 Aşağıdakilerden hangisinde komple bir tablo için çift kenarlıklı stil tanımlanmaktadır?

- A. table{border: 1px solid black;
- B. table{border: 1px solid black; border-collapse: collapse;}
- C. table{border: 1px solid black;}
- D. table, th, td {border: 1px dashed black;}
- E. table, th, td {border: 1px dashed black; border-collapse: collapse;}

7 Aşağıdakilerden hangisinde tablo başlığı için hücre içindeki verinin rengi tanımlanmaktadır?

- A. table#t01 th {color: #F5DEB3;}
- B. table#t01 th { background-color: yellow;}
- C. table#t01 td {color:white;}
- D. table#t01 tr {color:blue;}
- E. table#t01 tr { background-color: yellow;}

8 Aşağıdakilerden hangisi tablonun genişliğinin web sayfasının %75'ini kaplamasını sağlar?

- A. table#t01 {width: 100%;}
- B. <table style="width:75%">
- C. <table width:75%>
- D. <table style="width:0.75%">
- E. <table#t01 {width: 3/4%;}>

9 Aşağıdakilerden hangisi bir hücredeki sütun sayısını üçe artırmaktadır?

- A. <tr="3"> </tr>
- B. colspan=3
- C. <th colspan="3">
- D. <td colspan="3"> </td>
- E. <th colspan=3>Telefon Numarası</th>

10 Aşağıdakilerden hangisi bir hücredeki satır sayısını iki ye çıkartmaktadır?

- A. rowspan=iki;
- B. <th rowspan=2> </th>
- C. tr {rowspan=2; td}
- D. td, th, tr {rowspan:#};
- E. <th rowspan="2">Telefon Numarası</th>

1. B	Yanıtınız yanlış ise “Tabloların Tanımlaması” konusunu yeniden gözden geçiriniz.	6. D	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.
2. C	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.	7. A	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.
3. A	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.	8. B	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.
4. C	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.	9. D	Yanıtınız yanlış ise “Tabloların Bölünmesi” konusunu yeniden gözden geçiriniz.
5. E	Yanıtınız yanlış ise “Tabloların Stil Kullanılarak Biçimlendirilmesi” konusunu yeniden gözden geçiriniz.	10. E	Yanıtınız yanlış ise “Tabloların Bölünmesi” konusunu yeniden gözden geçiriniz.

4

Araştır Yanıt Anahtarı

Sunulan ekran çıktısını elde edebilecek örnek bir kod Şekil 4.25'te görülmektedir.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4  <meta charset="utf-8"/>
5  <style>
6  </style>
7  </head>
8  <body>
9
10 <table style="width:75%">
11 <tr>
12 <th>Ürün Adı</th>
13 <th>Ürün Kodu</th>
14 <th>Miktarı (ton)</th>
15 </tr>
16 <tr>
17 <td>Çilek</td>
18 <td>001</td>
19 <td>500</td>
20 </tr>
21 <tr>
22 <td>Muz</td>
23 <td>003</td>
24 <td>975</td>
25 </tr>
26 </table>
27
28 </body>
29 </html>
30

```

Şekil 4.25 Araştır etkinliği için dikkate alınacak örnek kod.

4

Araştır Yanıt Anahtarı

Araştır 2

Sunulan ekran çıktısını elde edebilecek örnek kodlamalardan biri Şekil 4.26'da verilmiştir.

```

6 <Style>
7 table, th, td { border: 1px solid black; border-collapse: collapse; }
8 th, td { padding-left: 5px; text-align: left; }
9 th { height: 50px; }
10 table#t01 { width: 100%; }
11 table#t01 th { background-color: blue; color: yellow; }
12 table#t01 td { text-align: center; }
13 table#t01 tr:nth-child(odd) { background-color: Azure; }
14 table#t01 tr:nth-child(even) { background-color: Bisque; }
15 </Style>
16 </head>
17 <body>
18 <h1>Bilişim Teknolojileri Dersi Final Sınavı Puanları </h1>
19 <table id="t01">
20 <tr>
21 <th>Adı</th>
22 <th>Soyadı</th>
23 <th>Final Notu</th>
24 </tr>
25 <tr>
26 <td>Mikail</td>
27 <td>Ceksin</td>
28 <td>90</td>
29 </tr>
30 <tr>
31 <td>Adem</td>
32 <td>Sendeler</td>
33 <td>88</td>
34 </tr>
35 </table>
36 <h1>Bilişim Teknolojileri Dersi Vize Sınavı Puanları </h1>
37 <table style="width:100%">
38 <tr>
39 <th>Adı</th>
40 <th>Soyadı</th>
41 <th>Vize Notu</th>
42 </tr>
43 <tr>
44 <td>Mikail</td>
45 <td>Ceksin</td>
46 <td>82</td>
47 </tr>
48 <tr>
49 <td>Adem</td>
50 <td>Sendeler</td>
51 <td>75</td>
52 </tr>
53 </table>
54 </body>
55 </html>

```

Şekil 4.26 Araştır etkinliği için dikkate alınacak örnek kod.

4

Araştır Yanıt
Anahtarı

Araştır 3

Sunulan ekran çıktısını elde edebilecek örnek kodlamalardan biri Şekil 4.27'de verilmiştir.

```

1  <!DOCTYPE html>
2  <html lang="tr">
3  <head>
4  <meta charset="utf-8"/>
5  <title>Tablo Oluşturmayı Öğreniyorum</title>
6  <style>
7  table, th, td {border: 1px dashed black;}
8  th, td { padding: 5px; text-align: left;}
9  table#t01 th { background-color: black; color:white; }
10 table#t01 tr:nth-child(odd) { background-color: cyan;}
11 table#t01 tr:nth-child(even) { background-color: orange;}
12 </style>
13 </head>
14 <body>
15 <table id="t01" style="width:100%">
16 <caption>Tablo 1. Kişisel Bilgiler</caption>
17 <tr>
18 <th>Adı Soyadı</th>
19 <td colspan="2">Muhammed Ali</td>
20 </tr>
21 <tr>
22 <th rowspan="2">Telefon Numarası</th>
23 <td>999 33 033</td>
24 <td>999 00 033</td>
25 </tr>
26 <tr>
27 <td>777 33 000</td>
28 <td>777 33 033</td>
29 </tr>
30 </table>
31 </body>
32 </html>

```

Şekil 4.27 Araştır etkinliği için dikkate alınacak örnek kod.

Kaynakça

- Castro, E. & Hyslop, B. (2014). *HTML and CSS: visual quickstart guide (8th Edition)*. Boston: Pearson.
- Cottrell, L. M. (2011). *HTML & XHTML DeMYSTiFieD*. New York: Mc Graw Hill.
- Çelikkilek, İ. (2015). *HTML 5: Geleceğin Web Standartları*. İstanbul: Kodlab.
- Duckett, J. (2011). *HTML & CSS: Design and build websites*. Indianapolis, John Wiley & Sons.
- Frain, B. (2015). *Responsive web design with HTML5 and CSS3 (Second Edition)*. Birmingham: Packt Publishing.
- Gauchat, J. D. (2015). *HTML5 for masterminds: How to take advantage of HTML5 to create amazing websites and revolutionary applications(2nd Edition)*. Toronto:Mink.
- Mermerkaya, A. O. (2015). *HTML 5 CSS 3*. İstanbul: Abaküs Kitap.
- Meloni, J. C. (2010). *HTML and CSS in 24 Hours, Sams Teach Yourself (Updated for HTML5 and CSS3), 9th Edition*. Boston: Pearson.
- Myers, M. (2015). *A Smarter Way to Learn HTML & CSS: Learn it faster. Remember it longer. (Volume 2)*. USA: Mark Myers.
- Schunk, D. H. (2012). *Learning theories: An educational perspective (6th ed.)*. Boston: Pearson.
- Tittel, E., & Minnick, C. (2013). *Beginning HTML5 and CSS3 for dummies*. New Jersey: Wiley
- Tokak, B. (2015). *HTML5, CSS3 ve Javascript ile web tasarımı*. İstanbul: Dikeyksen Yayınevi.
- W3Schools (ty.). *w3schools.com*.
- Wempen, F. (2011). *HTML5 step by step*. Sebastopol: O'Reilly Media.

Bölüm 5

Form Tasarımı ve Veri İletimi

öğrenme çıktıları

Form Yapısı

- 1 Formların kullanım amaçlarını ifade edebilme
- 2 Genel form yapısını tanımlayabilme
- 3 Form özelliklerinin kullanım amaçlarını ifade edebilme

Form Elemanları

- 4 Form elemanlarını ve HTML5 ile gelen form elemanlarını listeleyebilme
- 5 HTML5 kavramını açıklayabilme
- 6 Birden fazla form elemanı ile WEB formu tasarlayabilme
- 7 Form elemanlarının özelliklerini kullanabilme

Anahtar Sözcükler: • HTML • WEB • Form • Form Elemanları • Form Özellikleri • Veri İletimi

Username*

Your username

Password*

☐ Remember me

LOGIN

OR

GİRİŞ

WEB, İnternet üzerinde en çok kullanılan servislerden biridir. WEB sayfaları aracılığıyla bilgiye erişimdeki ve iletişimdeki yenilikler hayatı neredeyse her alanda değiştirmektedir. Teknolojiyle birlikte yeni iş ortamları yaratılırken mevcut işlerin yapılmasında yeni şekiller ortaya çıkmaktadır. İnsanlar sosyal ağlara dâhil olarak farklı iletişim olanaklarına kavuşmakta, bilgisayarlar da kendi aralarında haberleşebilmektedir. Bunun yanında WEB'in tarihine bakıldığında WEB 1.0, WEB 2.0, WEB 3.0 ... gibi bir sıradan söz edilebilir. İlk WEB sayfalarında bilgi girişi yapılabilecek arayüz elemanları ve teknolojileri olmadığından, İnternet kullanıcıları kendilerine sunulan bilgiye sadece bakabilmekteydiler. Siteler arasında gezinerek vakit geçirmek anlamına gelen WEB'de sörf tanımı da WEB 1.0 ile ortaya çıkan bir kavramdı. WEB 2.0, İnternet kullanıcılarının WEB sayfalarına bilgi girebildikleri yeni teknolojilerin ortaya çıkmasıyla anılmaya başlandı. Bu durum sosyal ağlar, yeni alışveriş, eğitim ve akla gelebilecek diğer sitelerin ortaya çıkmasını sağladı. Bugünün WEB anlayışında bilinen birçok sitenin içeriğini aslında kullanıcıların kendileri oluşturmaktadır. Başka bir deyişle aslında kullanıcılar olmasa bu siteler basit, belki de boş birer WEB sayfasından başka bir şey olmayacaktır. Yüklenen videolar, gönderilen mesajlar ve bilgi akışları aslında İnternet kullanıcılarının kendi ürünleridir.

Teknik anlamda WEB sayfaları, sunucu ve istemci olarak anılan bilgisayarların birbirlerine veri akışı yapmalarıyla iş görmektedir. Şöyle ki; Bir kullanıcı, bilgisayarı başında bir WEB sayfasına erişmek istediğinde, kendi bilgisayarını istemci olarak, erişmek istediği WEB sitesinin bilgisayarına yani sunucuya İnternet üstünden erişim sağlar. Sunucu üzerinde barındırdığı WEB sitesini ziyaretçinin bilgisayarı olan istemciye gönderir. Böylelikle kullanıcı istediği WEB sayfasına girmiş olur. Bu klasik akış WEB 1.0'ın çalışma biçimini de ifade etmektedir. WEB sayfalarının temelini oluşturan HTML (HyperText Markup Language) üzerinde yapılan geliştirmeler ve iyileştirmeler ve yeni WEB teknolojileri bu süreci bir adım daha ileriye götürmüştür. WEB 2.0'da sadece WEB sayfaları değil, kullanıcıya ait bilgiler de bilgisayarlar arasında dolaşmaktadır. WEB sayfaları üzerinde daha fazla etkileşim olanağı, HTML formları sayesinde gerçekleşmiştir. Yukarıda sözü geçen kullanıcıların WEB içeriğini kendilerinin oluşturmaları kavramı bu formlar sayesinde gerçekleşmiştir. Örneğin sosyal ağda kişisel sayfaya resim ekleme, alışveriş sitesinde adres girme, ya da bir haber sitesindeki habere yorum yapma formlar aracılığıyla gerçekleşir.

Kitabın bu bölümünde formlara, form elemanlarına, formların ve form elemanlarının özelliklerine yer verilmiştir. Konuyu en iyi şekilde anlatabilmek için kodlar ve bu kodların tarayıcıdaki görüntüleri alt alta konarak resimlendirilmiştir.

FORM YAPISI

Formlar genelde kullanıcıların veri girdiği ya da seçim yaptığı alanlar ile bu bilgileri sunucuya gönderen düğmelerden oluşan bir bütündür. Formlar içerisinde verilerin girildiği ya da seçimlerin yapıldığı araçlara form elemanları denir. Bir formun bir ya da birden fazla form elemanına ait verileri bir bütün olarak sunucuya gönderilebilir. Resim 5.1'de ad ve soyad bilgilerinin girilebildiği bir formun kodları yer almaktadır. Formları daha iyi anlayabilmek için bu örneğe göz atalım.

✓ Formlar genellikle kullanıcının bilgilerini ve seçimlerine ait verileri toplayan elemanlar ile bu verileri sunucuya gönderen düğmelerden oluşan bir bütün olarak düşünülebilir.

```
1 <form action="kayit.php" method="get" target="_blank">
2   Ad: <input type="text" name="isim">
3   Soyad: <input type="text" name="soyisim">
4   <input type="submit" value="Gönder">
5 </form>
```

Resim 5.1 Basit bir form örneğinin kodları

Resim 5.1'deki kayıt formunda isim, soyisim girilebilen iki metin kutusu ve bir gönder düğmesi olmak üzere üç adet form elemanını oluşturan kodlar yer almaktadır. Kodlar tarayıcıda açıldığında Resim 5.2'de ekran görüntüsü elde edilmektedir.

Ad: Soyad:

Resim 5.2 Basit bir form örneğinin tarayıcı görüntüsü

Kutucuklara girilen bilgiler Gönder düğmesiyle sunucuya gönderilmektedir. Bir siteye üye olmak istediğimiz zaman ad, soyad ve diğer bilgilerimizi bu tür formlara girerek siteye üye olmaktadır.

Resim 5.1'deki birinci satırda görüldüğü üzere `<form>` etiketi içinde action, method ve target gibi kodların bulunduğu görülmektedir. Bu kodlara özellik denilmektedir. Özellikler formlara ve form elemanlarına ek işlevler kazandırmak ya da sınırlamalar getirmek için kullanılırlar.

✓ Formlara ve form elemanlarına ek işlevler kazandırmak ve sınırlamalar getirmek için özellik olarak adlandırılan kodlar kullanılır.

Yukarıdaki form örneği formların genel yapısı hakkında bilgi vermektedir. Öyleyse bir formun genel yapısı Resim 5.3'te sunulduğu gibi özetlenebilir. Formlar `<form>` etiketiyle başlayıp klasik HTML kodlarında olduğu gibi `</form>` komutuyla sonlandırılır.

```
1 <form action="dosya yolu" method="get ya da post" ve diğer özellikler>
2 <form elemanları ve özellikleri>
3 </form>
```

Resim 5.3 Form yapısı



dikkat

Formlar `<form>` etiketiyle başlatıldıktan sonra kapatma etiketi olan `</form>` ile sonlandırılmalıdır.

Form Özellikleri

Resim 5.3'te görüldüğü gibi `<form>` etiketi içine Tablo 1'de yer alan özellikler eklenebilir. Bu özelliklerin alabilecekleri değerler, kullanım biçimleri ve tanımları tabloda yer almaktadır. Ayrıca bazı özelliklerin hemen altında ve sonraki bölümlerde görüleceği üzere HTML5 ibaresi yer almaktadır. HTML5, o özelliğin HTML5 ile yeni gelen bir özellik olduğunu göstermektedir. WEB sayfası tasarımının temelini oluşturan HTML, zaman içinde geliştirilerek yeni özellikler kazanmaktadır. Bu özellikler gerek WEB sayfası tasarımcılarına gerekse WEB kullanıcılarına kolaylıklar sağlamaktadır. Son olarak olarak HTML5 sürümü kullanılmaktadır.

Tablo 5.1 Form özellikleri

Özellik	Değer ve Kullanım	Tanımı
action	= "http://ornek.com/kayit.php" gibi	Form verilerinin sunucuda nereye gönderileceğini belirtir.
method	= "get" = "post"	Form verilerinin gönderilmesinde hangi HTTP yönteminin kullanılacağını belirtir.
enctype	= "application/x-www-form-urlencoded" = "multipart/form-data" = "text/plain"	Form verileri sunucuya gönderilirken nasıl kodlanacağını belirtir. Sadece method="post" ile kullanılabilir.
name	= "kayitformu" gibi	Forma bir isim verilmesini sağlar.
accept-charset	= "UTF-8" =	Form verilerinin hangi karakter kodlamasını kabul edeceğini belirtir.

novalidate HTML5	<i>novalidate</i>	Form verilerinin doğrulanmaması için kullanılır.
target	="_blank" (Yeni boş sayfa) ="_self" (Formun açıldığı sayfa) ="_parent" (İlk açılmış sayfa) ="_top" (En üstteki sayfa)	Form verileri sunucuya gönderildikten sonra cevabın hangi sayfada (Boş, açık olan, ana sayfa ya da en üstteki WEB sayfası) görüntüleneceğini belirtir.
autocomplete HTML5	="on" ="off"	Bu özellik form doldururken kullanıcıya önceden girilmiş bilgileri göstererek yardımcı olur.
accept	="audio/*" ses dosyaları ="video/*" görüntü dosyaları ="image/*" resim dosyaları =" .jpg, . doc" gibi (virgülle ayrılmış)	Sunucunun kabul ettiği dosya türlerini belirtir.

Kaynak: <https://www.w3.org/TR/html5/forms.html>, 2016

Tablo 5.1.'deki özellikler form yapısına uygun şekilde <form> etiketi içinde kullanılabilir. Örneğin Resim 5.4'te bir metin kutusu ve düğmeden oluşan bir formun kodları görülmektedir. Bu kodların tarayıcıdaki görüntüsü Resim 5.5'te yer almaktadır.

```

1 <form action="kayip.php" method="get" target="blank">
2   Adınız<br>
3   <input type="text" name="isim"><br>
4   <input type="submit" value="Gönder">
5 </form>

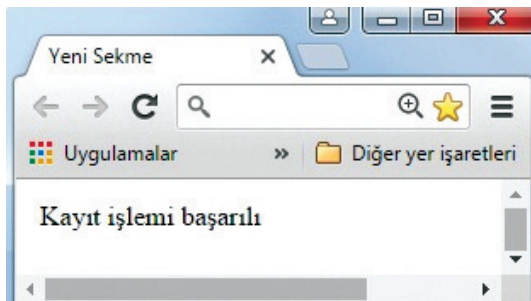
```

Resim 5.4 Formlara özellik eklemek

Adınız

Resim 5.5 Diğer form özelliklerini ekleme kodlarının tarayıcı görüntüsü

Metin kutusuna isim yazıldıktan sonra gönder düğmesine basıldığında, girilen değer isim değişkeniyle sunucuya gönderilmektedir. Sunucuda gerekli işlemler yapıldıktan sonra kayıt işleminin sonucu yeni bir pencerede "Kayıt işlemi başarılı" şeklinde gösterilmektedir. İşlemin tarayıcı görüntüsü Resim 5.6'daki gibidir.



Resim 5.6 Gönder düğmesine basıldıktan sonraki tarayıcı görüntüsü

İşlem sonucunu yeni bir pencerede açabilmek için target="_blank" komutu <form> etiketi içine eklenmiştir. Resim 5.4.'teki örnek kodlarda yer aldığı gibi aslında formların en fazla kullanılan iki özelliği vardır. Bunlar action ve method'dur. Bu iki özellik neredeyse bütün formlarda yer alır.

✓ Formların en fazla kullanılan iki özelliği action ve method'dur. Action form verilerinin sunucuda nereye gönderileceğini, method ise verilerin sunucuya nasıl gönderileceğini belirler.

Action özelliği formlardan alınan verinin sunucuda nerede değerlendirileceğini belirler. CGI, ASP veya PHP gibi uygulamalar form verilerini işlemeye kullanılır. Örneğin sunucuya gönderilen veriler bir veritabanına kaydedilebilir. Method özelliği kullanıcı verisinin sunucuya nasıl gönderileceğini, başka bir deyişle form verilerinin iletim biçimini belirler. Get ve post olmak üzere 2 değer vardır. Bu iki değer farkları şu şekilde özetlenebilir. Formlarda veri iletim metodu olarak get kullanıldığında form elemanının karşısında aldığı de-

ğer adres satırında gözükür. Örneğin Resim 5.4'teki örnekte isim kutusuna cem yazıp gönder düğmesine bastığımızda açılan sayfanın adres satırında “<http://www.webadresl.com.tr/kayit.php?isim=cem>” şeklinde bir ifade yer alacaktır. Form içerisinde başka elemanlar yer alıyorsa, onların değeri de bu şekilde adres satırında çiftler hâlinde gözükcektir. Post metodunda ise veriler adres satırında gözükmeden arka planda sunucuya gönderilir. Get metodundaki gibi şifre türü verilerin adres satırında gözükmesi, güvenlik anlamında sorunlar yaratabilir. Ayrıca get metodunda form bilgilerini içeren adres gönderimi için belirli bir karakter sayısı sınırı vardır. Bu sınır tarayıcı ya da sunucuya göre değişebilir. Başka bir deyişle uygulamada çok yeri olmasa da get metoduyla sınırlı miktarda form verisi sunucuya gönderilebilir.

Ancak get ile veri iletiminin bir avantajı vardır. Formlarla yapılmış belirli sorgulamaların sonuçları sık kullanılanlara eklenebilir. Bu bir arama motorunda yapılan araştırma ya da bir WEB sitesinde belirli değişkenlere göre yapılan arama olabilir. Örneğin bir ikinci el araç sitesinde dizel, sedan ve mavi bir araç aradığımızda, arama sonuçlarını sık kullanılanlara ekleyerek daha sonra inceleme olanağına sahip olabiliriz.

Name özelliği formların ve form elemanlarının çoğunda yer alır. Özellikle veri taşıyan ve üzerinde sunucu ya da istemcide işlem yapılacak form elemanlarında name özelliği kullanılmalıdır. Form elemanının taşıyacağı veri için kullanılacak değişkenin adı name ile belirlenir. Örneğin `<input type="text" name="ad">` ifadesinde ad isimli bir metin kutusu oluşturulmaktadır. Kutuya girilen kullanıcı ismi Ali ise, `ad=Ali` olarak sunucuya gönderilir.



dikkat

name özelliği veri barındıran bütün form elemanlarında kullanılmalıdır. Bir form elemanındaki veriyi tutacak olan değişken adı name ile belirlenir.

Öğrenme Çıktısı



- 1 Formların kullanım amaçlarını ifade edebilme.
- 2 Genel form yapısını tanımlayabilme.
- 3 Form özelliklerinin kullanım amaçlarını ifade edebilme.

Araştır 1

Bağlantı özellikli bir metnin, yönlendirdiği sayfaya aynı zamanda bir veri teslim etmesini nasıl sağlayabileceğinizi araştırınız.

İlişkilendir

Web üzerinde karşılaştığınız bir form sayfasının HTML kodlarını görüntüleyerek, üzerindeki verileri nereye yönlendirdiğini keşfetmeye çalışınız.

Anlat/Paylaş

GET ve POST metodlarının yapısını göz önünde bulundurarak, kullanıcıların parolalarının iletimi için hangisinin daha uygun olacağını yorumlayınız.

FORM ELEMANLARI

Form elemanları genellikle `<form>...</form>` etiketleri arasında kalan alana girilen, kullanıcının veri girişi ya da seçim yaptığı birimlerdir. Örneğin bir siteye kayıt olurken mail adresimizi girdiğimiz metin kutusu veya öğrenim durumumuzu belirlediğimiz açılır listeler birer form elemanıdır. Web sayfalarında yer alabilecek önemli form elemanlarının listesi Tablo 5.2'de özetlenmiştir.

- ✓ Form elemanları, WEB sitesinde metin kutuları, açılır listeler, onay kutuları, düğmeler gibi kullanıcının veri girişi ya da seçim yaptığı alanları oluşturur.

Tablo 5.2 Form elemanları

Form Elemanı	Alt Eleman	HTML Sürüm
Input (type ile)	text (metin)	
	password (şifre)	
	tel (telefon numarası)	HTML5
	email (E-posta)	HTML5
	date (tarih)	HTML5
	month (ay)	HTML5
	week (hafta)	HTML5
	time (zaman)	HTML5
	datetime-local (yerel zaman)	HTML5
	url (WEB adresi)	HTML5
	checkbox (onay kutusu)	
	radio (radyo düğmesi)	
	submit (gönder düğmesi)	
	image (resimli düğme)	
	reset (sıfırlama düğmesi)	
	button (düğme)	
	file (dosya yükleme)	
	hidden (gizli)	
	color (renk)	HTML5
	range (aralıklı kaydırma)	HTML5
	number (numara)	HTML5
	search (arama)	HTML5
button (düğme) (type ile)	Button	
	Submit	
	Reset	
fieldset (gruplayıcı)	legend (başlık)	
select (açılır liste)	optgroup (gruplama başlığı)	
	option (seçenek)	
textarea (metin alanı)	-	
datalist (veri listesi)	-	HTML5
label (etiketler)	-	
output (sonuç çıktısı)	-	HTML5
keygen (kullanıcı kimliği doğrulama)	-	HTML5
meter (gösterge)	-	HTML5
progress (ilerleme çubuğu)	-	HTML5

Kaynak: <https://www.w3.org/wiki/HTML/Elements/form>, 2016

Görüldüğü gibi button hem input elemanın bir özelliği, hem de ayrı bir form elemanıdır. Bu durumun nedeni Button (Düğme) bölümünde anlatılmıştır. Ayrıca output, progress, meter, keygen elemanları PHP, ASP gibi WEB programlama dilleriyle birlikte kullanılabildiğinden, kitapta bu elemanlara yer verilmemiştir. Bunlardan keygen ve output elemanları, kullanıcının seçim ve giriş yapabilmesi nedeniyle progress ve meter elemanlarına göre daha fazla form özelliği taşımaktadır.



Keygen ve output elemanlarının uygulamalı örneklerini incelemek için <http://huseyinbodur.net/?p=791#more-791> adresinden faydalanabilirsiniz.

Form elemanlarından bazılarının sağ tarafında yer alan HTML5 ibaresi, bu elemanların HTML5 ile gelen yeni elemanlar olduğunu göstermektedir. Bu bölümden sonra form elemanlarının her birinin kullanım amaçlarını açıklayan, bunları örneklendiren ve kendilerine verilebilecek özellikleri gösteren bölümlere yer verilmiştir.



yaşamla ilişkilendir

Tarayıcılardaki Otomatik Form Doldurma Özelliğinde Büyük Güvenlik Açığı!

Tarayıcıların kolaylık sağlamak için sunduğu otomatik form doldurma özelliğinde kişisel verileri tehlikeye atan önemli bir güvenlik açığı tespit edildi.

Tarayıcıların çoğunda bulunan otomatik form doldurma (tamamlama) özelliğini mutlaka farketmişsinizdir. İnternet sitelerinde herhangi bir form doldururken bu özellik sayesinde formdaki boşlukları otomatik olarak tamamlayabılıyorsunuz. Ancak bu özelliği kullanırken artık iki kere düşünmeniz gerekiyor. Zira bu küçük özellik büyük bir güvenlik açığını da beraberinde getiriyor.

Anttivijlami isimli bir GitHub kullanıcısının farkettiği bu güvenlik açığı, kullanıcıların otomatik doldurma (autofill) özelliği ile kaydedilmiş

tüm bilgilerin -tek bir form alanını bu özellik ile doldursa dahi- açığa çıkmasına sebep oluyor. Kötü niyetli siber saldırı gerçekleştirmek isteyenler ise bu verilere kolayca ulaşabiliyor.

Peki ne yapmak gerekiyor?

Kişisel verilerinizin tehlikeye girmemesi için ilk yapmanız gereken -ilgili güvenlik açığı kapatılana kadar- tarayıcı ayarlarından otomatik form doldurma özelliğini kapatmak. Chrome, Firefox, Safari gibi tarayıcıların ayarlarına girerek ilgili özelliği kapatabilirsiniz. Ayrıca güvenmediğiniz internet sitelerinde form doldurmaktan da kaçınmanızı tavsiye ederiz.

Kaynak: <http://www.webmasto.com/tarayicilardaki-otomatik-form-doldurma-ozelliginde-buyuk-guvenlik-acigi>

Input (Giriş)

Input elemanı genellikle <form> etiketi içinde kullanılır ve birçok formatta veri girebilmesini sağlar. Veri girişi tipi type komutuyla belirlenir ve kapama etiketi kullanılmadan özellikler eklenir.

Input elemanlarına, kullanılan veri tipine göre çeşitli özellikler yüklenebilir. İlgili input elemanında en fazla kullanılan özelliklerin nasıl kullanılacağı ve hangi değerleri alabileceği Tablo 3'te listelenmiştir. Ayrıca Tablo 1'de yer alan autocomplete ve name özelliği de Input elemanının bir özelliği olarak kullanılabilir. Yine Tablo 1 'deki action, enctype, method, validate ve target özelliklerinin başına form eki eklenerek formaction, formenctype, formmethod, formvalidate ve formtarget şeklinde kullanılabilir. Bu durumda form elemanına, ait olduğu formun özellikleri dışında kendi özellikleri eklenebilir.



dikkat

Input elemanında kapama etiketi kullanılmaz.



Input elemanı ve bu elemanın özelliklerine ilişkin örnekleri incelemek ve uygulamalar yapmak için http://www.html5dersleri.org/index.php?getir=html5_formattrs&ID=34 adresinden faydalanabilirsiniz.

Tablo 5.3 Input özellikleri

Özellik	Değer ve Kullanım	Tanım
accept	= "audio/*" ses dosyaları = "video/*" görüntü dosyaları = "image/*" resim dosyaları = ".jpg, .doc" gibi (virgülle ayrılmış)	Yüklenebilecek dosya türünü ve formatını belirler. Soldaki değerler araya virgül konarak istenilen şekilde kullanılabilir.
alt	= "Gönder" gibi alternatif değer	Resimli düğmenin resmi sayfaya yüklenemediği durumlarda metin olarak düğmenin işlevini ifade eder ve resmin yerine geçer.
checked	<i>checked</i>	Onay kutusunun varsayılan değer olarak işaretli olarak görünmesini sağlar.
height	= "20" gibi piksel cinsinden değer	Resimli düğmenin yüksekliğini piksel cinsinden belirler.
list HTML5	= "liste" gibi bir açılır liste kimliği	Önceden oluşturulmuş bir açılır listenin kimliğini kullanarak liste elemanlarına erişir.
max min	= "20" gibi sayısal değer = "1979-12-31" gibi tarih	Bir form elemanının alabileceği en yüksek veya düşük değeri belirler.
maxlength minlength	= "20" gibi sayısal değer	Bir elemanın alabileceği en yüksek değeri belirler. Pozitif tam sayı değerleri alır.
multiple HTML5	<i>multiple</i>	Kullanıcının birden fazla email girebilmesini ya da dosya yüklemelerinde birden fazla dosya seçebilmesini sağlar. E-postalar için aralarda virgül kullanılır.
pattern HTML5	= "[A-Za-z]" regexp değerler (regular expressions)	Belirli kalıplara göre veri girişi sağlar. Örneğin kullanıcı adı belirlerken büyük ve küçük harf kullanımının zorunlu tutulması. "[A-Za-z]" bu durumu sağlar.
placeholder HTML5	= "birisi@webadres.com.tr" gibi	Veri girişi öncesi ipucu sağlar. Örneğin e-posta girilecek alanda örnek bir e-posta gösterilmesi (birisi@mail.com).
readonly	<i>readonly</i>	Veri girişini kapatır. İlgili form elemanında yalnızca belirlenen değer görünür.
required HTML5	<i>required</i>	Formun doldurulması zorunlu alanlarını belirler. Bu alanlar doldurulmadan veri girişi sağlanmaz.
size	= "20" gibi sayısal değer	İlgili form elemanının kaç karakter uzunluğunda veri alabileceğini belirler.
src	= "http://ornek.com/dugme.jpg" gibi	Bir resimli düğmenin resim yolunu belirler (Resmin sunucuda bulunduğu yer).
step	= "3" = "-3" gibi sayısal değer	İlgili form elemanında yukarı ve aşağı yönlü olarak adım miktarını belirler.
width	= "20" gibi piksel cinsinden değer	Resimli düğmenin genişliğini piksel cinsinden belirler.
value	= "gonderdugmesi" gibi	Bir giriş elemanının varsayılan değerini belirler.

Kaynak: <https://www.w3.org/TR/html5/forms.html>, 2016

Pattern özelliğinde yer alan regexp (Regular Expressions, Düzenli İfadeler) terimi programlama dillerinde ve HTML'de özellikle veri girişlerinde sıklıkla kullanılmaktadır. Yandaki karekodu kullanarak, açılan sayfadan daha fazla örnek ve uygulamaya erişebilirsiniz.



Özellikler kullanıcı veri girerken ya da seçim yaparken fayda sağlamaktadır. Örneğin bir text elemanına bilgi girerken en fazla on karakter girilmesi isteniyorsa `maxlength="10"` şeklinde belirtilir. Böylelikle kullanıcı 10 karakterden fazla veri giremez. Şimdi sırasıyla bütün input elemanlarına ve özelliklerin nasıl kullanılabileceğine göz atalım.

Text (Metin Kutusu)

Tek satırlık metin girişi için kullanılır. Resim 5.7'de text için örnek kodlar yer almaktadır.

```
1 <form>
2   İsim<br>
3   <input type="text" name="ad"><br>
4   Soyisim<br>
5   <input type="text" name="soyad">
6 </form>
```

Resim 5.7 Input text örneği

İsim

Soyisim

Resim 5.8 Input Text Örneğinin Tarayıcı Görüntüsü

Resim 5.7'deki kodların sırasıyla üçüncü ve beşinci satırlarında ad ve soyad isimli birer metin kutusu oluşturulmaktadır. İsim ve Soyisim bilgileri sunucuya, ad ve soyad isimli değişkenler olarak gönderilmektedir. Text için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.8'deki gibidir.

Text elemanına autocomplete, list, maxlength, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir. Örneğin formda ad ve soyad bölümünün mutlaka doldurulması isteniyorsa `<input type="text" name="ad" required>` şeklinde required komutu eklenmelidir. Tablo 3'te görüldüğü gibi required özelliğinin bir değeri yoktur. Input etiketi içine doğrudan yazılabilir.

Password (Şifre)

Kullanıcının şifre girebilmesi için bir metin alanı oluşturur. Resim 5.9'da password için örnek kodlar yer almaktadır.

```
1 <form>
2   Şifrenizi giriniz <input type="password" name="sifre">
3 </form>
```

Resim 5.9 Input password örneği

Resim 5.9'daki kodların ikinci satırında şifre isimli bir şifre kutusu oluşturulmaktadır. Bu alana girilen karakterler güvenlik amacıyla ekranda * ya da gibi sembollerle gizlenir. Forma ait bir gönder düğmesi var ise, gönder düğmesine basıldığında şifre, şifre isimli değişken olarak sunucuya gönderilmektedir. Password için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.10'daki gibidir.

Şifrenizi giriniz

Resim 5.10 Input password örneğinin tarayıcı görüntüsü

name="ad" autocomplete="off"> şeklinde düzenlenmelidir. Tablo 3'te görüldüğü gibi autocomplete özelliğinin "on" ve "off" olmak üzere iki değeri vardır.

Password elemanına autocomplete, maxlength, name, pattern, placeholder, readonly, required, ve size özellikleri verilebilir. Örneğin önceden ziyaret edilmiş ve şifre girilmiş bir sitede, şifre alanının otomatik olarak doldurulması istenmiyorsa, kodlar `<input type="password"`

Tel (Telefon Numarası)

Kullanıcının telefon numarası girebilmesi için bir alan oluşturur. Ancak bazı tarayıcılar bu özelliği desteklemez. Resim 5.11'de tel için örnek kodlar yer almaktadır.

```

1 <form action="form.php">
2   Telefon Numaranız: <input type="tel" name="telno"><br>
3   <input type="submit">
4 </form>

```

Resim 5.11 Input tel örneği

Resim 5.11'deki kodların ikinci satırında telefon numarası girilebilecek, telno isimli bir numara alanı oluşturulmaktadır. Tel için yazılmış söz konusu kodların tarayıcıdaki görüntüsü Resim 5.12'deki gibidir. Gönder Düğmesine basıldığında telefon numarası, telno isimli değişken ile sunucuya gönderilmektedir.

Telefon Numaranız:
Gönder

Resim 5.12 Input tel örneğinin tarayıcı görüntüsü

Tel elemanına autocomplete, list, maxlength, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir. Telefon numarasının belirli bir şekilde girilmesi isteniyorsa kullanıcıya bilgi verebilmek için `<input type="tel" name="telno" placeholder="0 5XX XXX XX XX">` şeklinde düzenlenebilir. Burada telefon numarasının Türkiye'deki numaralandırma biçimi gösterilmiştir. Tarayıcı görüntüsü Resim 5.13'te görülmektedir.

Telefon numaranız Gönder

Resim 5.13 Diğer bir input tel örneği

Email (E-posta)

Kullanıcının e-posta adresini girebilmesi için bir e-posta alanı oluşturur. Resim 5.14'te email için örnek kodlar yer almaktadır.

```

1 <form>
2   E-posta adresiniz: <input type="email" name="email">
3   <input type="submit">
4 </form>

```

Resim 5.14 Input email örneği

Resim 5.14'teki kodların ikinci satırında email isimli bir e-posta alanı oluşturulmaktadır. Gönder düğmesine basıldığında bu alana girilen e-posta adresi, email isimli değişken ile sunucuya gönderilmektedir. Email için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.15'teki gibidir. Email, e-postalardaki @ simgesi kontrolü yaparak verileri gönderir. Alana girilen karakterlerde @ yoksa Resim 5.15'teki tarayıcı uyarısını verir. Email elemanına autocomplete, list, maxlength, multiple, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir.

E-mail: Gönder

! Lütfen e-posta adresine bir "@" işareti ekleyin. "hcs" adresinde "@" eksik.

Resim 5.15 Input email örneğinin tarayıcı görüntüsü

Date (Tarih)

Kullanıcının tarih girebilmesi için bir alan oluşturur. Bununla birlikte Form Elemanları adlı bölümde listelenen diğer tarih ve zaman elemanları (datetime-local, month, week, time) benzer şekilde kullanıldığından kitapta bu elemanlara ayrıca yer verilmeyecektir. Resim 5.16'da date için örnek kodlar yer almaktadır.

```

1 <form action="demo.asp">
2   Doğum tarihiniz: <input type="date" name="dogumgunu">
3   <input type="submit">
4 </form>

```

Resim 5.16 Input date örneği

Doğum tarihiniz: gg . aa . yyyy

Haziran 2016

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Resim 5.17 Input date örneğinin tarayıcı görüntüsü

Resim 5.16'daki kodların ikinci satırında dogumgunu isimli bir tarih alanı oluşturulmaktadır. Date için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.17'deki gibidir. Gönder düğmesine basıldığında bu alanda seçilen tarih, dogumgunu isimli değişken ile sunucuya gönderilmektedir.

Date elemanına autocomplete, list, max, min, name, readonly, required, step ve value özellikleri verilebilir. Örneğin tarih alanına 1 Ocak 2000 tarihinden önceki tarihlerin girilmemesi isteniyorsa <input type="date" name="dogumgunu" min="2000-01-01"> şeklinde min özelliği kullanılabilir.

URL (WEB Adresi)

Kullanıcının bir WEB adresi girebilmesi için alan oluşturur. Resim 5.18'de url için örnek kodlar yer almaktadır.

```

1 <form action="sayfaniz.asp">
2   WEB adresiniz: <input type="url" name="sayfa"><br>
3   <input type="submit">
4 </form>

```

Resim 5.18 Input url örneği

Resim 5.18'deki kodların ikinci satırında sayfa isimli bir url alanı oluşturulmaktadır. URL için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.19'daki gibidir. Gönder düğmesine basıldığında bu alana girilen WEB adresi, sayfa isimli değişken ile sunucuya gönderilmektedir. Url elemanı, girilen metinde http:// belirtecini kontrol eder. Alana girilen karakterlerde http:// belirteci yoksa Resim 5.19'daki tarayıcı uyarısını verir. Url elemanına autocomplete, list, maxlength, name, pattern, readonly, required, size, ve value özellikleri tanımlanabilir.

WEB adresiniz:

! Lütfen bir URL girin.

Resim 5.19 Input url örneğinin tarayıcı görüntüsü

Checkbox (Onay Kutusu)

Kullanıcının işaretleyebilmesi için onay kutuları oluşturur. Resim 5.20'de checkbox için örnek kodlar yer almaktadır.

```

1 <form action="demo_form.asp">
2   <input type="checkbox" name="arac1" value="bisiklet" checked> Bisiklet<br>
3   <input type="checkbox" name="arac2" value="taksi"> Taksi<br>
4   <input type="checkbox" name="arac3" value="tekne"> Tekne<br>
5   <input type="submit" value="Gönder">
6 </form>

```

Resim 5.20 Input checkbox örneği

Resim 5.20'deki kodların ikinci ve dördüncü satırları arasında arac1, arac2 ve arac3 isimli onay kutuları oluşturulmaktadır. Checkbox için yazılmış bu kodların tarayıcı görüntüsü Resim 5.21'deki gibidir.

☒ Bisiklet
☐ Taksi
☐ Tekne

Resim 5.21 Input checkbox örneğinin tarayıcı görüntüsü

Kullanıcı kendine uyan seçenek ya da seçenekleri işaretleyebilir. Gönder düğmesine basıldığında bisiklet değeri arac1, taksi değeri arac2 ve tekne değeri arac3 isimli değişkenlerle sunucuya gönderilmektedir. Checkbox elemanına checked, name, required, ve value özellikleri verilebilir. Örneğin bisiklette olduğu gibi kullanıcıya WEB sayfası açıldığında onay kutusu işaretli gösterilecekse checked özelliği eklenebilir.

Radio (Radyo Düğmesi)

Kullanıcının seçim yapabilmesi için radyo düğmeleri oluşturur. Resim 5.22'de radio için örnek kodlar yer almaktadır.

```
1 <form action="kayit.php">
2   Cinsiyetiniz<br>
3   <input type="radio" name="cinsiyet" value="bay"> Bay<br>
4   <input type="radio" name="cinsiyet" value="bayan"> Bayan<br>
5   <input type="submit" value="Gönder">
6 </form>
```

Resim 5.22 Input radio örneği

Cinsiyetiniz

☐ Bay

☐ Bayan

Resim 5.23 Input radio örneğinin tarayıcı görüntüsü

Resim 5.22'deki kodların üçüncü ve dördüncü satırlarında cinsiyet isimli radyo düğmeleri oluşturulmaktadır. Radio için yazılmış bu kodların tarayıcı görüntüsü Resim 5.23'teki gibidir.

Kullanıcı kendine uyan seçeneklerden birisini seçebilir. Gönder düğmesine basıldığında cinsiyet değişkeni bay veya bayan değerlerinden biriyle sunucuya gönderilmektedir. Radio elemanına checked, name, required, ve value özellikleri verilebilir.

Submit (Gönder Düğmesi)

Submit, bundan önceki form elemanları için verilen örneklerde görüldüğü gibi, genel olarak kullanıcının form verilerini sunucuya göndermesi için kullanılır. Resim 5.24'te submit için örnek kodlar yer almaktadır.

```
1 <form action="kayit.php">
2   Cinsiyetiniz:<br>
3   <input type="radio" name="cinsiyet" value="bay">Bay<br>
4   <input type="radio" name="cinsiyet" value="bayan">Bayan<br>
5   <input type="submit" name="gonderdugmesi" formaction="cinsiyet.php" value="Gönder">
6 </form>
```

Resim 5.24 Input submit örneği

Resim 5.24'teki kodların beşinci satırında gonderdugmesi isimli bir gönder düğmesi oluşturulmaktadır. Value değeri düğmenin üstünde beliren yazıdır. Bu örnekte Gönder olarak belirlenmiştir. Submit için yazılmış bu kodların tarayıcı görüntüsü Resim 5.25'teki gibidir. Kullanıcının cinsiyet seçimi gonderdugmesi isimli düğmeyle sunucuya gönderilmektedir. Submit elemanına formaction, formenctype, formmethod, formvalidate, formtarget, name, ve value özellikleri verilebilir.

Cinsiyetiniz

☐ Bay

☐ Bayan

Resim 5.25 Input submit örneğinin tarayıcı görüntüsü

Ancak örnekte olduğu gibi formaction özelliğiyle cinsiyet bilgisi, form etiketinde yer alan kayıt.php adresine değil, düğme içinde tanımlanan cinsiyet.php adresine gönderilmektedir. Formaction özelliği submit için, kendi veri gönderme yolunu ya da dosyasını belirlemesini sağlar.

Image (Resimli Düğme)

Image aslında bir submit elemanıdır. Genel olarak kullanıcının form verilerini sunucuya göndermesi için kullanılır. Tek farkı gönder düğmesi yerine bir resim kullanılmasıdır. Resim 26'da image için örnek kodlar yer almaktadır.

```

1 <form action="adsoyad.asp">
2   Adınız: <br>
3   <input type="text" name="isim"><br>
4   Soyadınız: <br>
5   <input type="text" name="soyisim"><br>
6   <input type="image" src="gondersimgesi.png" alt="Gönder" width="55" height="55">
7 </form>

```

Resim 5. 26 Input image örneği

Resim 5.26'daki kodların altıncı satırında image tipinde düğme oluşturulmaktadır. Type, image değişkeniyle düğme türünü (bu örnekte resimli düğme) belirlemektedir. Src resmin sunucudaki konumunu göstermektedir. Örnekte olduğu gibi resim ögesi olarak gondersimgesi.png isimli resim kullanılmıştır. Bu resmin sayfada görüntülenememesi durumunda, alt özelliğiyle sayfada tıklanabilir Gönder yazısı belirecektir. Bu yazıya basıldığında bilgi adsoyad.asp adresine gönderilecektir. Resmin piksel cinsinden eni ve boyu sırasıyla width ve height özellikleriyle belirlenmiştir.



Resim 5. 27 Input image örneğinin tarayıcı görüntüsü

Image için yazılmış bu kodların tarayıcı görüntüsü Resim 5.27'deki gibidir. Kullanıcının ad ve soyad bilgisi image elemanıya, yani mavi ok simgesiyle sunucuya gönderilmektedir. Image elemanına alt, formaction, formenctype, formmethod, formvalidate, formtarget, name, src ve value özellikleri verilebilir.

Reset (Sıfırlama Düğmesi)

Reset genel olarak çok fazla veri girilmesi gereken formlarda önceden girilmiş verilerin tarayıcı üstünde silinmesi için kullanılır. Başka bir deyişle form elemanlarına veri girilmişse, form elemanlarını sayfa yüklendiğindeki değerlerine döndürür. Ancak submit gibi düğmelerle veriler sunucuya gönderilmediği sürece, bilgiler sadece tarayıcı üstünde ilk değerlerine döner. Resim 5.28'de reset için örnek kodlar yer almaktadır.

```

1 <form action="bilgiler.php">
2   E-postanız: <input type="text" name="email"><br>
3   Adresiniz: <input type="text" name="adres"><br>
4   <input type="radio" name="ogrenimdurumu" value="ilkokul"> ilkokul<br>
5   <input type="radio" name="ogrenimdurumu" value="ortaokul" checked=""> Ortaokul<br>
6   <input type="radio" name="ogrenimdurumu" value="Diger"> Diğer<br>
7   <input type="submit" value="Gönder">
8   <input type="reset" value="Resetle">
9 </form>

```

Resim 5. 28 Input reset örneği

Resim 5.28'deki kodların sekizinci satırında type özelliğinin reset değeriyle reset tipinde düğme oluşturulmaktadır. Reset için yazılmış bu kodların tarayıcı görüntüsü Resim 5.29'deki gibidir.

E-postanız:

Adresiniz:

☐ İlkokul
☒ Ortaokul
☐ Diğer

Durum 1.
WEB sayfası açıldığında form değerleri

E-postanız:

Adresiniz:

☐ İlkokul
☐ Ortaokul
☒ Diğer

Durum 2.
Kullanıcı bilgileri girişi sonrası form değerleri

Resim 5.29 Input reset örneği tarayıcı görüntüsü

Resim 5.29 'da iki durum vardır. Kullanıcı form alanlarını doldururken yanlış veriler girmişse, resetle düğmesine basarak Durum 1'e dönebilir. Dikkat edilirse sadece e-posta ve adres bilgilerinin girildiği metin alanları değil, öğrenim durumunun belirlendiği seçim düğmeleri de ilk değerlerine dönmektedir. Reset elemanın herhangi bir özelliği yoktur.

Button (Düğme)

Button öntanımlı olmayan düğmeler oluşturur. Başka deyişle gerçekleştirecekleri eylem sayfa tasarımcısı tarafından belirlenir. Örneğin oluşturulan düğme sayfada bir javascript kodunu etkinleştirebilir. Resim 5.30'da button için örnek kodlar yer almaktadır.

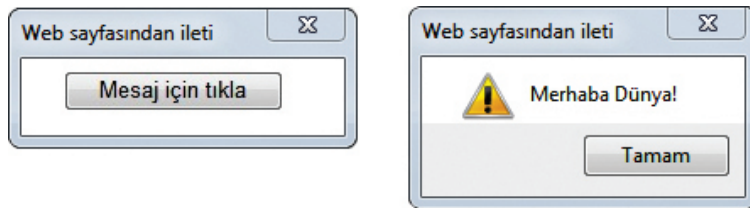
```

1 <form>
2   <input type="button" value="Mesaj için tıkla" onclick="mesaj()">
3 </form>
4
5 <script>
6 function mesaj() {
7   alert("Merhaba Dünya!");
8 }
9 </script>

```

Resim 5.30 Input button örneği

Resim 5.30'daki kodların ikinci satırında type özelliğinin button değeriyle, button tipinde düğme oluşturulmaktadır. Value değeri düğme üstünde görünen ifadeyi belirlemektedir. Onclick ise altıncı ve dokuzuncu satırlar arasında yer alan mesaj isimli javascript kodunu çalıştırmaktadır. Kodların tarayıcı görüntüsü Resim 5.31'deki gibidir. Resim 5.31'de görüldüğü gibi sol tarafta sayfa içinde yer alan Mesaj için tıkla düğmesine basıldığında, tarayıcının mesaj penceresinden "Merhaba Dünya!" mesajı ekranda belirmektedir.



Resim 5.31 Input button örneğinin tarayıcı görüntüsü

Button elemanına autofocus, disabled, name ve value özellikleri verilebilir. Örneğin disabled özelliği, kullanıldığı düğme elemanını etkisizleştirerek, tıklanamaz duruma getirir.

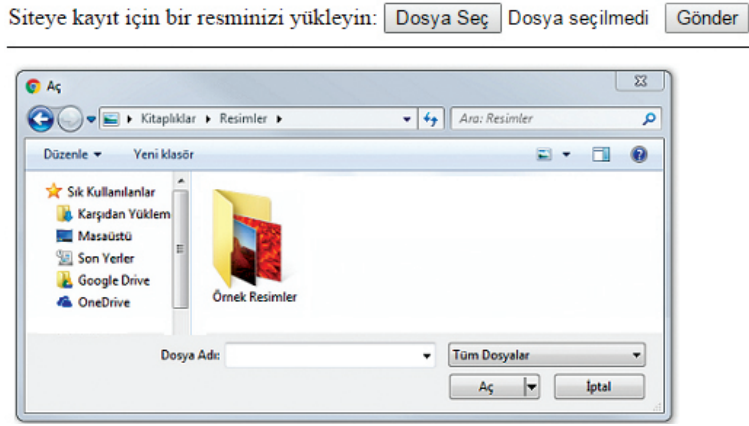
File (Dosya Yükleme)

File, kullanıcının tarayıcıda dosya/dosyalar seçebilmesi için kullanılır. Seçim yapıldıktan sonra forma ait submit düğmesiyle dosya sunucuya gönderilebilir. Resim 5.32'de file için örnek kodlar yer almaktadır.

```
1 <form action="resim.php">
2   Siteye kayıt için bir resminizi yükleyin:
3   <input type="file" name="resim">
4   <input type="submit" name="gonderdugmesi" value="Gönder">
5 </form>
```

Resim 5.32 Input file örneği

Resim 5.32'deki kodların üçüncü satırında sunucuya dosya yükleyebilmek resim isimli dosya yükleme elemanı oluşturulmaktadır. Resim 5.33'te kullanıcının resim yükleyebilmesi için oluşturulan formun tarayıcı görüntüsü yer almaktadır.



Resim 5.33 Input file örneğinin tarayıcı görüntüsü

File tarayıcıya göre farklı isimlerde dosya seçim düğmesi oluşturur. Resim 5.33'te Google Chrome'da Dosya Seç düğmesinin kullanıldığı görülmektedir. İnternet Explorer'da ise bu düğmenin adı Gözet'dir. Kullanıcı dosya seç düğmesine bastığında dosya seçim penceresi belirir. Resim seçildikten sonra Gönder düğmesiyle resmi sunucuya göndermektedir.

✓ Input file ile yaratılan ve dosya seçme penceresini açmak için kullanılan düğme, tarayıcılarda farklı isimlerde gösterilir. Örneğin Google Chrome'da Dosya Seç, İnternet Explorer'da ise Gözet olarak sayfada belirir.

File elemanına accept, multiple, name, required ve value özellikleri verilebilir. Örneğin birden fazla dosyanın seçilebilmesi isteniyorsa <input type="file" name="resim" multiple> şeklinde multiple özelliği eklenebilir.

Hidden (Gizli)

Hidden, WEB sayfasındaki formlarda gerekli ancak kullanıcının görmemesi gereken ya da görmesinin bir anlam taşımadığı verileri gizlemek için kullanılır. Formda yer alan gizli veriler de sunucuya gönderilir. Bu elemanı kullanabilmek için javascript ve WEB programlamasındaki dillerden yararlanılmaktadır. Resim 5.34'te hidden için örnek kodlar yer almaktadır.

```

1 <form action="takim.asp">
2   Tuttuğunuz takım: <input type="text" name="takim"><br>
3   <input type="hidden" name="ulke" value="Türkiye">
4   <input type="submit" value="Gönder">
5 </form>

```

Resim 5.34 Input hidden örneği

Resim 5.34'teki kodların üçüncü satırında ulke isminde bir gizli eleman oluşturulmaktadır. Hidden için yazılmış örnek kodların tarayıcı görüntüsü Resim 5.35'teki gibidir. Bu örnekte kullanıcının tuttuğu takımı girebilmesi için bir form hazırlanmıştır. Ancak formda, kullanıcının ülkesini kendisinin girebileceği bir alan yoktur. Bunun yerine ülke bilgisi, IP ya da tarayıcı bilgileri kullanılarak otomatik olarak form verisine eklenebilir. Kullanıcı gönder düğmesine bastığında hem tuttuğu takım hem de yaşadığı ülke sunucuya gönderilebilir. Hidden elemanına name ve value özellikleri verilebilir.

Tuttuğunuz takım:

Resim 5.35 Input hidden örneğinin tarayıcı görüntüsü

Color (Renk)

Color kullanıcının bir renk seçebilmesi için kullanılır. Seçilen renk, rengin sayısal ifadesi olarak sunucuya gönderilir. Resim 5.36'da color için örnek kodlar yer almaktadır.

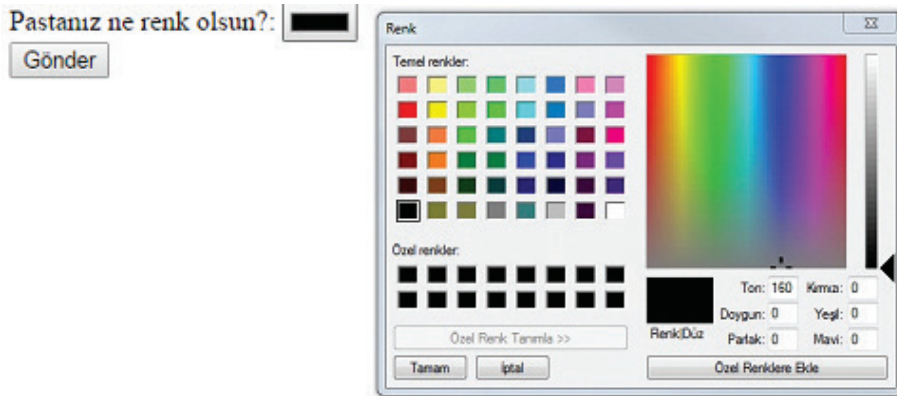
```

1 <form action="siparis.asp">
2   Pastanız ne renk olsun?: <input type="color" name="pastarengi"><br>
3   <input type="submit">
4 </form>

```

Resim 5.36 Input color örneği

Resim 5.36'daki kod örneğinin ikinci satırında pastarengi isminde bir renk seçim düğmesi oluşturulmaktadır. Kullanıcının pasta siparişi için renk belirleyebilmesini sağlayan kodların tarayıcı görüntüsü Resim 5.37'deki gibidir. Kullanıcı Pastanız ne renk olsun? yazısının yanındaki renk seçim düğmesine tıkladığında resimde görülen bir renk paleti belirlemektedir. Kullanıcı istediği rengi bu paletten seçebilmektedir.



Resim 5.37 Input color örneğinin tarayıcı görüntüsü

Color elemanına autocomplete, list, name ve value özellikleri verilebilir. Örneğin resimde gönderilen renk seçim düğmesinin renginin kırmızı olması isteniyorsa `<input type="color" name="pastarengi" value="#ff0000">` şeklinde value özelliği kullanılabilir.



WEB tasarımında ve programlamada renkler sayısal olarak ifade edilirler. Renklerin sayısal değerleri genellikle onaltılık (Hexadecimal) sistemle gösterilir. Konu ile ilgili daha fazla bilgi ve örneği içeren WEB sayfasına erişim için yandaki karekodu kullanabilirsiniz.

Range (Aralıklı Kaydırma)

Kullanıcının sayı değeri belirleyebilmesi için bir kaydırma çubuğu oluşturur. Belirlenen değer bir form verisi olarak sunucuya gönderilir. Resim 5.38'de range için örnek kodlar yer almaktadır.

```
1 <form action="siparis.asp">
2   Ürün sayısı? 1<input type="range" name="adet" value="1" min="1" max="5">5
3   <input type="submit" value="Ürünü onayla">
4 </form>
```

Resim 5.38 Input range örneği

Resim 5.38'deki kodların ikinci satırında adet isminde bir aralıklı kaydırma elemanı oluşturulmaktadır. Kullanıcının aynı üründen kaç adet sipariş vereceğini belirleyen formda, değer adet değişkeniyle sunucuya gönderilmektedir. Ürün sayısı aralıklı kaydırma elemanı ile belirlendikten sonra, ürünü onayla isimli submit düğmesiyle ürün sayısı sunucuya iletilmektedir. Min kaydırma çubuğundaki en küçük değeri, max en büyük değeri belirler. Value değeri sayfa açıldığında kaydırma çubuğu işaretçisinin nerede (Hangi değerde)

Ürün sayısı? 1 5 Ürünü onayla

Resim 5.39 Input range örneğinin tarayıcı görüntüsü

olacağını belirler. Bu örnekte en küçük değer olan 1 değeri verilmiştir. Kodların tarayıcı görüntüsü Resim 5.39'daki gibidir. Range elemanına autocomplete, disabled, form, list, max, min, name, step, time ve value özellikleri verilebilir.

Number (Sayı)

Kullanıcının değer belirleyebilmesi ya da girebilmesi için bir sayı alanı oluşturur. Bu alan içinde yukarı ve aşağı oklar bulunur. Böylelikle kullanıcı girdiği ya da sayfada beliren değeri yukarı veya aşağı değiştirebilir. Belirlenen değer bir form verisi olarak sunucuya gönderilir. Resim 5.40'ta number için örnek kodlar yer almaktadır.

```
1 <form action="adet.asp">
2   Bir sayı girin:<br>
3   <input type="number" value="4" name="sayi" min="1" max="10" step="2">
4   <input type="submit">
5 </form>
```

Resim 5.40 Input number örneği

Resim 5.40'taki kodların üçüncü satırında sayi isimli bir sayı alanı oluşturulmaktadır. Belirlenen ya da kutuya girilen değer sayi değişkeniyle sunucuya gönderilmektedir. Min sayı alanındaki en küçük değeri, max en büyük değeri belirler. Value değeri sayfa açıldığında sayı alanında hangi sayının olacağını belirler. Bu örnekte 4 değeri verilmiştir. Step ise değerlerin kaçar kaçar artacağını ya da azalacağını belirler. Resim 5.41'de kodların tarayıcı görüntüsü yer almaktadır. Number elemanına autocomplete, disabled, form, list, max, min, name, placeholder, readonly, required, step, ve value özellikleri verilebilir.

Bir sayı girin:
4 Gönder

Resim 5.41 Input number örneğinin tarayıcı görüntüsü

Search (Arama)

Kullanıcının arama yapabileceği bir alan oluşturur. Aslında bu eleman İnternet üzerinden ya da diğer aramalar için gerekli özelliklere sahip değildir. Mevcut hâliyle yapabildiği tek şey, alana girilen değeri bir arama değeri olarak sunucuya göndermek, kullanıcı bilgisayarında gerekli WEB programlama dilleriyle çeşitli aramalar yaptırmaktır. Ayrıca x simgesiyle arama satırındaki mevcut değeri siler. Resim 5.42'de search için örnek kodlar yer almaktadır.



dikkat

tel (Telefon Numarası) elemanında olduğu gibi search de tarayıcılar tarafından desteklenmez.

```
1 <form action="arama.asp">
2   Ara: <br>
3   <input type="search" name="aramayap">
4   <input type="submit">
5 </form>
```

Resim 5.42 Input search örneği

Ara:

Resim 5.43 Input search örneğinin tarayıcı görüntüsü

Resim 5.42'deki kodların üçüncü satırında aramayap isimli bir arama elemanı oluşturulmaktadır. Alana girilen değer aramayap değişkeniyle sunucuya gönderilmektedir. Kodların tarayıcıdaki görüntüsü Resim 5.43'teki gibidir. Search elemanına autocomplete, list, maxlength, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir.

Button (Düğme)

Button daha önceki input bölümünde bahsedilen, <form> etiketi dışında da kullanılabilen genel amaçlı düğme türleri yaratır. Button için kullanılabilecek özellikler Tablo 4'te özetlenmiştir.

Tablo 5.4 Button özellikleri

Özellik	Değer ve Kullanım	Tanım
type	= "button" = "submit" = "reset"	Ne tür düğme yaratılacağını belirler.
autofocus HTML5	autofocus	Sayfa açıldığında kullanıcıyı ilgili düğmeye yönlendirir.
disabled	disabled	Düğmeyi tıklanamaz duruma getirir. Düğmenin etkisiz olmasını sağlar.
form HTML5	= "formkimligi" gibi	İlgili formun kimliğini kullanarak, o form için işlem yapılacağını belirler.
formaction Submit için HTML5	= "http://ornek.com/kayit.php" gibi	Form verilerinin formun kendi yolu dışında başka bir yola gönderilmesini sağlar.
formenctype Submit için HTML5	= "application/x-www-form-urlencoded" = "multipart/form-data" = "text/plain"	Form verileri sunucuya gönderilirken nasıl kodlanacağını belirtir.
formmethod Submit için HTML5	= get = post	Form verilerinin formun kendi iletim metodu dışında bir yöntemle gönderilmesini sağlar.

formvalidate Submit için HTML5	<i>formvalidate</i>	Formun kendi veri doğrulama özelliğini devre dışı bırakarak kendi değerine göre doğrulama işlemi yapar.
formtarget Submit için HTML5	=_blank (Yeni boş sayfa) =_self (Formun açıldığı sayfa) =_parent (İlk açılmış sayfa) =_top (En üstteki sayfa)	Form kendi gösterim özelliğini devre dışı bırakarak cevabın hangi sayfada gösterileceğini belirtir. Sayfada (Boş, açık olan, ana sayfa ya da en üstteki WEB sayfası)
value	= <i>"gonderdugmesi"</i> gibi	Bir giriş elemanının varsayılan değerini belirler.
name	= <i>"resetle"</i> gibi	Düğmelere isim vermek için kullanılır.

Kaynak: <https://www.w3.org/TR/html-markup/button.html>, 2016)

Button elemanının type özelliği, oluşturulacak olan düğmenin türünü belirlemektedir. Submit, reset ve button olmak üzere üç çeşit button vardır. Tablo 4'te görüldüğü gibi formaction, formenctype, formmethod, formvalidate ve formtarget özellikleri sadece submit tipi için kullanılabilir. Button elemanının input ile yaratılan button ile işlevleri aynıdır. Ancak input buttondan farklı olarak *<button özellikler>* şeklinde kullanılır ve HTML kapama etiketi *</>* ile sonlandırılmaları gerekir.

Bununla birlikte button elemanı ile yaratılan düğmelerin en büyük özelliği resim ve metinlerle şekillendirilebilmeleridir. Özellikle CSS (Cascading Style Sheet) ile biçimlendirilme konusunda oldukça kullanışlıdır. Resim 5.44'te submit, reset ve button türünde hazırlanmış button elemanlarına ait örnek yer almaktadır.



dikkat

Button (düğme), input ile oluşturulan düğmeden farklı olarak *<button>* etiketiyle başlatılıp, klasik HTML kodlarında olduğu gibi bitirme etiketi olan *</button>* komutuyla sonlandırılır.



CSS ile button şekillendirme örneklerini incelemek için <http://fatihhayrioglu.com/css-ile-buton-yapmak/> adresinden faydalanabilirsiniz.

```

1 <form action="hesap.php" method="post">
2   <p>
3     Adınız: <br><input type="text" name="firstname"><br>
4     Soyadınız: <br><input type="text" name="lastname"><br>
5     Eposta adresiniz: <br><input type="text" name="email"><br>
6   <br>
7   <button name="submit" type="submit" value="gonder">
8     Gönder
9   </button>
10  <button name="reset" type="reset" value="resetle">
11    Sıfırla
12  </button>
13  <button name="merhaba" type="button" value="merhaba" onclick="mesaj()">
14    Merhaba de
15  </button>
16  </p>
17 </form>
18
19
20 <script>
21 function mesaj () {
22   alert("Merhaba");
23 }
24 </script>

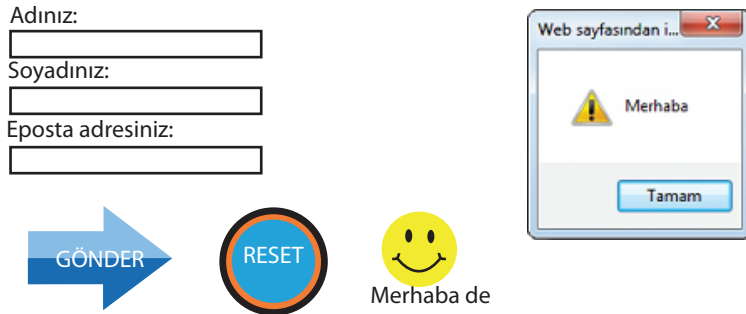
```

Resim 5.44 Submit reset button örneği

Resim 5.44'te yedinci ve dokuzuncu satırlar arasında submit isimli düğme tanımlanmıştır. Sekizinci satırda HTML'nin resim ekleme etiketi olan kullanılarak bu düğmeye jpg uzantılı gonder ismindeki resim eklenmiştir. Böylelikle tarayıcıda bir düğme yerine bu resim gözükmektedir. Resmin boyutuna bağlı olarak düğmenin genişliği width="70", yüksekliği ise height="55" kodlarıyla 70 ve 55 piksel olarak ayarlanmıştır.

Benzer şekilde onuncu ve on ikinci satırlar arasında tanımlı reset isimli düğmeye, gif uzantılı reset ismindeki resim eklenmiştir. Bu düğmenin ise yüksekliği ve genişliği reset resminin şekline göre 55 piksel olarak belirlenmiştir. Düğmeye gif uzantılı reset ismindeki resim eklenmiştir.

Merhaba de düğmesi ise, onüç ve onbeşinci satırlar arasında, merhaba.gif isimli resim dosyasından görüntüsünü almaktadır. 55 piksel yüksekliğinde ve genişliğindedir. Kodların tarayıcı görüntüsü Resim 5.45'teki gibidir.



Resim 5.45 Submit reset button örneğinin tarayıcı görüntüsü

Düğmelerin işlevleri input button ile yaratılan düğmelerle aynıdır. Gönder düğmesi kullanıcı bilgilerini sunucuya göndermektedir. Reset, Adınız, Soyadınız ve E-posta adresiniz isimli metin kutularındaki bilgileri silmekte ya da bir başlangıç değeri varsa o değere döndürmektedir. Merhaba de düğmesi ise, Resim 5.45'te görüldüğü gibi ekrana Merhaba yazan pencereyi açan (20. satırdan itibaren) javascript kodunu çalıştırmaktadır.

Fieldset ve Legend (Gruplayıcı ve Başlık)

Fieldset form elemanlarını görsel açıdan gruplamak için kullanılır. Fieldset içindeki elemanlar bir kutu içinde gözükmürler. Legend ise gruplara başlık vermek için kullanılır. Resim 5.46'da fieldset ve legend için örnek kodlar yer almaktadır.

```

1 <fieldset>
2   <legend>Kişisel Bilgiler</legend><br/>
3   Adı: <input name="ad" type="text">
4   Soyadı: <input name="soyad" type="text"><br/><br/>
5 </fieldset>
6 <fieldset>
7   <legend>İletişim bilgileri</legend><br/>
8   Adresi: <input name="personal_address" type="text" size="55" tabindex="3"/><br/>
9   Telefon No: <input type="tel" name="telno">
10 </fieldset>

```

Resim 5.46 Fieldset legend örneği

Resim 5.46'daki kodların birinci ve beşinci satırlarında ad ve soyad elemanlarından bir grup oluşturmak için fieldset kullanılmıştır. Bu gruba ikinci satırdaki legend elemanı ile kişisel bilgiler başlığı verilmiştir. Aynı şekilde altı ve onuncu satırlardaki fieldset adres ve telefon elemanlarından bir grup oluşturmak için kullanılmıştır ve yedinci satırdaki legend ile bu gruba İletişim bilgileri başlığı verilmiştir. Kodların tarayıcı görüntüsü Resim 47'deki gibidir. Fieldset elemanına disabled, form, name özellikleri verilebilir.

Resim 5.47 Fieldset legend örneğinin tarayıcı görüntüsü

Select, Optgroup, Option (Açılır Liste, Gruplama Başlığı, Seçenek)

Select açılır liste oluşturmak, option açılır liste elemanlarını belirlemek, optgroup ise gerekli durumlarda bu elemanları gruplandırmak için kullanılır. Resim 5.48'de select, optgroup ve option için örnek kodlar yer almaktadır.

```

1  Gitmek için bir ülke seçin...
2  <select name="ulkelistesi" form="ulkeler">
3    <optgroup label="Avrupa Ülkeleri">
4      <option value="1">Almanya</option>
5      <option value="2">Fransa</option>
6      <option value="3">İtalya</option>
7    </optgroup>
8    <optgroup label="Amerika Ülkeleri">
9      <option value="4">ABD</option>
10     <option value="5">Meksika</option>
11     <option value="6">Brezilya</option>
12   </optgroup>
13   <optgroup label="Asya ülkeleri">
14     <option value="7">Hindistan</option>
15     <option value="8">Çin</option>
16     <option value="9">Afganistan</option>
17   </optgroup>
18 </select>
19
20 <form action="ulke.php" id="ulkeler">
21   isim:<input type="text" name="ad">
22   <input type="submit">
23 </form>

```

Resim 5.48 Select optgroup option örneği

Resim 5.48'deki kodların ikinci ve on sekizinci satırlarında ülkeler ve bu ülkelerin yer aldığı kıtaya göre gruplandırıldığı ulkelistesi adında bir açılır liste oluşturulmaktadır. Açılır listede Avrupa, Amerika ve Asya olmak üzere ülkeler üç grupta listelenmiştir. Gruplama işlemi <optgroup> etiketiyle yapılmış ve her gruptan önce kullanılmıştır. Ayrıca bu grupların başlıklarını oluşturmak için <optgroup> etiketinin label özelliği kullanılmıştır.

Yirminci satırdan itibaren kimliği (id'si) ülkeler olan form bulunmaktadır. Görüldüğü gibi açılır liste bu formun dışındadır. Liste elemanlarını forma dâhil edebilmek için listenin form özelliğine, formun kimliği verilmiştir. Kullanıcı açılır listeden bir eleman seçip gönder düğmesine bastığında, seçilen değer sunucuya bu listeden form aracılığıyla gönderilir. Select elemanı doğrudan bir form içinde de yaratılabilir. Böylelikle select o formun elemanı olur. Kodların tarayıcı görüntüsü Resim 5.49'daki gibidir. Kullanıcı açılır listeden bir ülke seçip gönder düğmesine bastığında seçilen ülke sunucuya gönderilir. Select elemanına autofocus, disabled, form, name, required, size özellikleri verilebilir.

Resim 5.49 Select optgroup option örneğinin tarayıcı görüntüsü

Textarea (Metin Alanı)

Textarea kullanıcının yazı yazabileceği çok satırlı metin kutuları oluşturur. Satır sayısı rows değişkeniyle belirlenir. Resim 5.50'de textarea için örnek kodlar yer almaktadır.

```

1 Gezi izlenimlerinizi burada bizimle paylaşın
2 <textarea rows="5" cols="50" form="gezi">
3 Buraya düşüncelerinizi yazınız
4 </textarea>
5
6 <form action="ulke.php" id="gezi">
7   <input type="submit">
8 </form>

```

Resim 5.50 Textarea örneği

Resim 5.50'deki kodların ikinci ve dördüncü satırları arasında textarea ile bir metin alanı oluşturulmuştur. Metin alanına rows="5" kodu ile beş satır yükseklik, "cols="50" komutu ile elli karakter alabilecek genişlik verilmiştir. Altıncı ve sekizinci satırlar arasındaki kodlar ise metin alanına girilen bilgileri sunucuya göndermek için kullanılmıştır. Bu örnekte de textarea, kendi form özelliği olarak, formun kimliği (id'sini) yani "gezi" değerini kullanmaktadır. Böylelikle bu metin alanı, gezi kimlikli formun elemanı olmuştur. Kullanıcı metin alanına düşüncelerini girip Gönder düğmesine bastığında, form bilgileri sunucuya gönderilmektedir. Textarea, bu örnekten farklı olarak, doğrudan bir formun elemanı olarak bir form içinde yaratılabilir. Böylelikle o formun elemanı olur. Örnekteki kodların tarayıcı görüntüsü Resim 5.51'deki gibidir. Textarea elemanına autofocus cols, disabled, form, name, placeholder, readonly, required, rows özellikleri verilebilir.

Gezi izlenimlerinizi burada bizimle paylaşın

Buraya düşüncelerinizi yazınız

Gönder

Resim 5.51 Textarea örneğinin tarayıcı görüntüsü

Datalist (Veri Listesi)

Datalist diğer elemanlar için listelenmiş veri sağlar. Bu yönüyle select elemanına benzer. Ancak kullanıcı liste elemanlarına doğrudan göremez. Input elemanı olan list gibi elemanlarla bu elemanlara ulaşılabilir ve kullanıcılara liste elemanları gösterilebilir. Resim 5.52'de datalist için örnek kodlar yer almaktadır.

```

1 Gittiğiniz şehir:<br>
2 <input type="text" name="sehir" list="sehirismi" form="gezi">
3
4 <datalist id="sehirismi">
5   <option value="Boston">
6   <option value="New York">
7 </datalist>
8
9 <form action="sehir.asp" id="gezi">
10   <input type="submit">
11 </form>

```

Resim 5.52 Datalist örneği

Resim 5.52'de dördüncü ve yedinci satırlarda datalist kullanılarak sehirismi kimliğini (id) taşıyan bir veri listesi oluşturulmuştur. Bu veri listesinde Boston ve New York olmak üzere iki değer yer almaktadır. İkinci satırda bu veri listesindeki şehir isimlerini görüntüleyen sehir isimli bir metin alanı oluşturulmuştur. Metin kutusu içinde veri listesindeki şehirleri görüntüleyebilmek için, input text elemanının list özelliğine, datalist elemanının kimliği, yani sehirismi değeri verilmiştir. Normal görünümde sayfa yüklendiğinde aç-

lır liste elemanları kullanıcıya görünmemektedir. Liste elemanları, fare imleci metin kutusunun üzerine götürüldüğünde ortaya çıkar. Datalist aynı zamanda bir form elemanı olarak list elemanı ile birlikte formun içinde yaratılabilir. Bununla birlikte metin kutusuna tıklanıldığında açılır liste elemanlarından farklı değerler de bu metin kutusu içine girilebilir. Resim 5.53'te kodların tarayıcı görüntüsü yer almaktadır.

✓ Datalist hem kullanıcıların seçim yapabilmeleri için bir açılır liste hem de seçim listesi elemanlarından farklı değerler girebilmeleri için bir metin kutusu olarak iş görür.

Resim 5.53 Datalist örneğinin tarayıcı görüntüsü

Örnekte veri listesi ve metin kutusu yanında dokuzuncu ve on birinci satırlarda kimliği (id) gezi olan bir form tanımlanmıştır. Input text elemanının form özelliğinde bu kimliğin kullanılması, onun bu forma ait olduğunu gösterir. Kullanıcı şehir seçip veya başka bir şehir girdiğinde, formdaki gönder düğmesi girilen ya da seçilen şehir ismini sunucuya göndermektedir. Datalist elemanına verilebilecek bir özellik yoktur.

Label Elemanı

Label elemanı diğer elemanları etiketlemek için kullanılır. Sadece ilişkilendirildiği form elemanı ile bir bütün olarak, o form elemanının özelliğini kazanır. Resim 5.54'deki label elemanı için hazırlanmış örnek kod ve tarayıcıdaki görüntüsü, bu durumu daha iyi anlayabilmek için yardımcı olacaktır.

```

1 <form action="vergi.asp">
2   <label for="binek">taksi</label>
3   <input type="radio" name="arac" id="binek" value="taksi"><br>
4
5   ticari<input type="radio" name="arac" id="ticari" value="otobüs"><br>
6   <input type="submit" value="Gönder">
7 </form>

```

Resim 5.54 Label örneği

Resim 5.54'te üçüncü satırda arac isimli radyo düğmelerinden ilki için, kimliği (id'si) binek olan seçenek oluşturulmuştur. İkinci satırda ise bu radyo düğmesi için taksi etiketi (label) oluşturulmuştur. Dikkat edilirse kimliği binek olan radyo seçeneğinin id değeri ile label etiketinin for değeri aynıdır. Bu durum binek kimliği taşıyan radyo düğmesinin başına, taksi yazısının (etiketinin) gelmesini sağlamıştır. Böylelikle yazı ve düğme bir bütün olmuştur. Ancak beşinci satırda yer alan arac isimli radyo düğmelerinden ikincisi için oluşturulan ticari düğmesi için aynı durum söz konusu değildir. Ticari düğmesinin başında yer alan ticari ibaresi sadece bir metindir ve bir etiket değildir. Farklı özellikler taşıyan bu radyo düğmeleri için kullanıcı açısından uygulamada fazla fark yoktur. Tek fark taksi yazısına tıklanıldığında radyo düğmesinin işaretlenebilmesidir. Ancak ticari yazısına tıklanıldığında, yazı ile düğme ilişkilendirilmediği için düğme işaretlenemez. Label elemanına verilebilecek bir özellik yoktur.

Resim 5.55 Label örneğinin tarayıcı görüntüsü

Öğrenme Çıktısı



- 4 Form elemanlarını ve HTML5 ile gelen form elemanlarını listeleyebilme.
- 5 HTML5 kavramını açıklayabilme.
- 6 Birden fazla form elemanı ile WEB formu tasarlayabilme.
- 7 Form elemanlarının özelliklerini kullanabilme.

Araştır 2

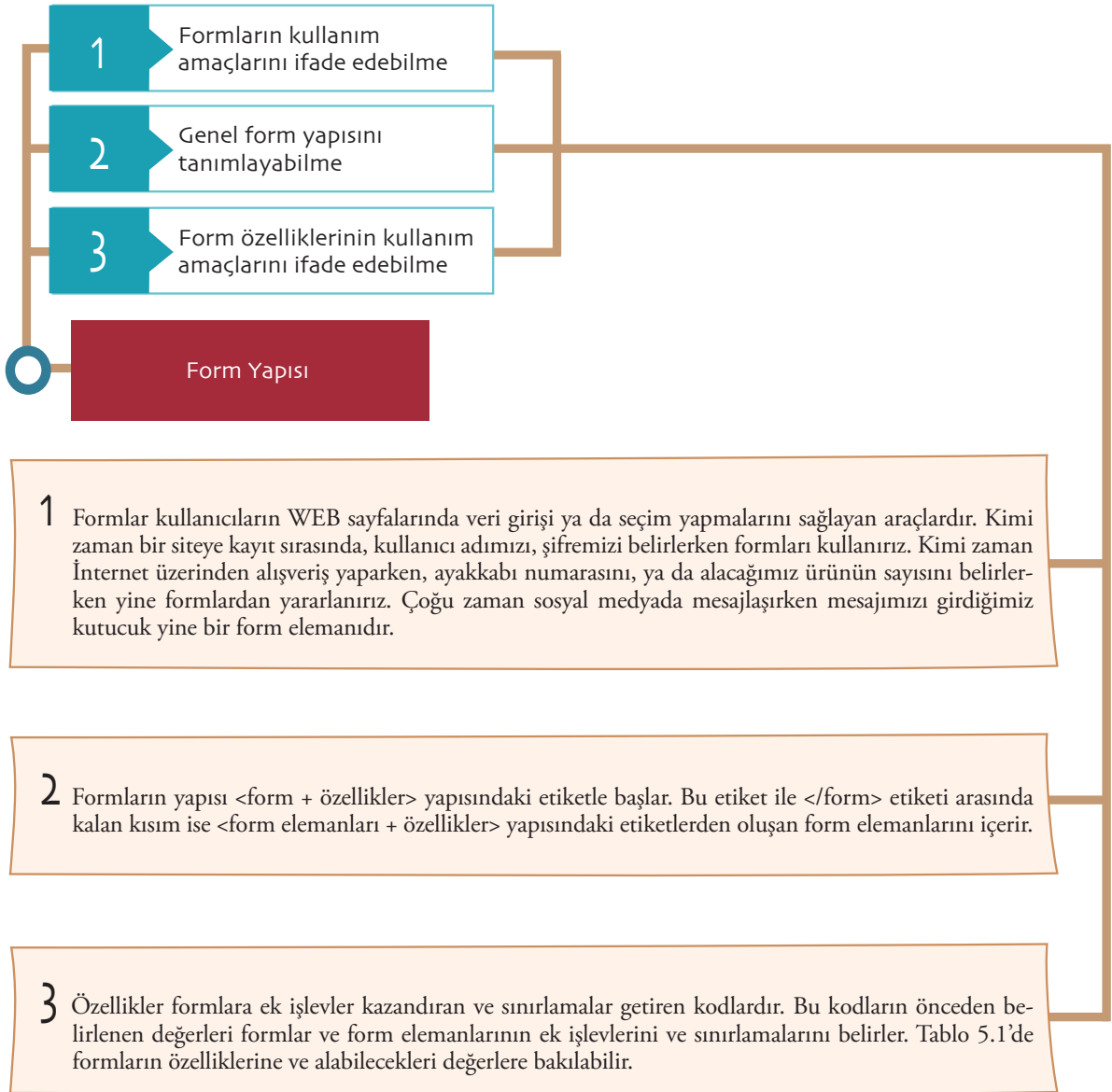
Bir e-posta alanına birden fazla e-posta adresi girilebilmesi için Resim 5.14.'de yer alan email elemanına hangi özelliğin eklenmesi gerektiğini bulunuz.

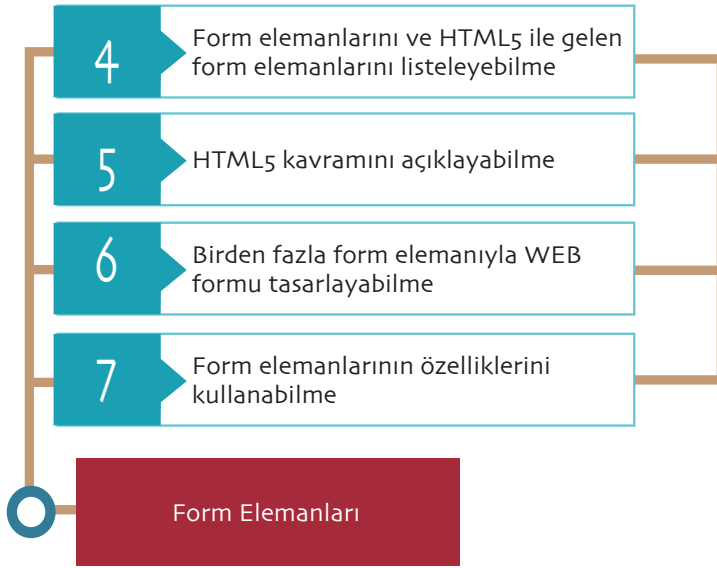
İlişkilendir

Şimdiye kadar ziyaret ettiğiniz web sayfalarında ne tür form elemanlarıyla karşılaşmış olduğunuzu hatırlamaya çalışınız.

Anlat/Paylaş

“Button” tipli form elemanı ile “Submit” tipli form elemanının farklarını tartışınız.





4 Başlıca form elemanları input (giriş), button (düğme), fieldset (gruplayıcı), select (Açılır liste), textarea (Çok satırlı metin alanı), datalist (Veri listesi-HTML5), label (Etiket), output (Sonuç çıktısı-HTML5), progress (İlerleme çubuğu-HTML5), meter (gösterge- HTML5) ve keygen (kullanıcı kimliği doğrulama- HTML5) olarak listelenebilir.

Listedeki input (giriş) elemanı type özelliği ile diğer form elemanlarından farklı olarak birçok veri girişi ve seçim alanı oluşturabilir. Bunlar text (metin), password (şifre), tel (telefon numarası-HTML5), email (E-posta-HTML5), date (tarih-HTML5), month (ay-HTML5), week (hafta-HTML5), time (zaman-HTML5), datetime-local (yerel zaman-HTML5), url (WEB adresi-HTML5), checkbox (onay kutusu), radio (radyo düğmesi), submit (gönder düğmesi), image (resimli düğme), reset (sıfırlama düğmesi), button (düğme), file (dosya yükleme), hidden (gizli), color (renk-HTML5), range (aralıklı kaydırma-HTML5), number (numara-HTML5) ve search (arama- HTML5) şeklindedir.

5 Görüldüğü gibi bazı form elemanlarının yanında HTML5 ifadesi kullanılmaktadır. Bunun sebebi HTML'nin bir WEB sayfası oluşturma dili olarak zaman içinde geliştirilmesidir. Böylelikle HTML hem kullanıcıların hem de WEB sayfası tasarımcılarının işlerini kolaylaştıracak ek işlevler kazanmaktadır. Bu kapsamda son olarak HTML5 olarak adlandırılan yükseltme sürümüne ait kodlar, etiketler ve özellikler kullanılmaktadır.

6 Bir Web sitesinde kullanıcının veri girmesi, seçimler yapması ve bunları sunucuya göndermesi için birden fazla form elemanı birlikte kullanılabilir. Tasarımcının toplamak istediği bilgiler için kullanacağı form elemanlarının nasıl kullanılacağı ilgili form elemanı bölümünden bakılabilir. Örneğin bir alışveriş sitesinde müşteri bilgilerinin toplanacağı bir form tasarlanabilir. Form elemanları olarak ad, soyad bilgileri için text, telefon bilgileri için tel, adres bilgileri için textarea gibi elemanlar kullanılabilir. Müşterinin alışveriş sırasında seçeceği ürünün sayısı number elemanı ile belirlenebilir. Renk seçiminde color elemanı kullanılabilir. Müşteri tarafından girilen bütün bu bilgiler gönder düğmesiyle sunucuya gönderilir.

7 Form elemanlarının özellikleri, elemanlara ek işlevler kazandırmak ve sınırlamalar getirmek için kullanılır. Hangi elemana hangi özelliğin verilebileceği ilgili form elemanı bölümünde yer almaktadır. Örneğin multiple özelliği file ve email elemanlarında kullanılır. File elemanında kullanıldığında kullanıcı bir siteye resim yüklerken birden fazla resmi aynı anda seçebilir. Böylelikle büyük bir kolaylık elde etmiş olur. Email elemanında ise birden fazla kişiye tek seferde mail adresi yapılarak mail gönderilebilir.

1 Form ve form elemanlarına ek işlevler kazandırmak ya da sınırlamalar getirmek için aşağıdakilerden hangisi kullanılır?

- A. Gruplayıcılar
- B. Şablonlar
- C. Veri kalıpları
- D. Özellikler
- E. Biçimlendiriciler

2 Form özelliklerinden biri olan action ne amaçla kullanılır?

- A. Form verilerini temizlemek
- B. Form verilerinin sunucuya nasıl gönderileceğini belirlemek
- C. Form verilerinin sunucudaki yolunu belirlemek
- D. Form elemanlarının doğrulanmasını sağlamak
- E. Forma isim vermek

3 Aşağıdakilerden hangisi bir input elemanı **değildir**?

- A. Datalist
- B. Checkbox
- C. Text
- D. Email
- E. Password

4 Required özelliği ne amaçla kullanılır?

- A. Form elemanını etkisiz hale getirmek
- B. Form elemanına çoklu giriş yapabilmek
- C. İlgili form elemanının kaç karakter uzunluğunda veri alabileceğini belirler.
- D. Form elemanının mutlaka doldurulmasını sağlamak
- E. Form elemanına isim vermek

5 Aşağıdakilerden hangisi HTML5 ile gelen bir form elemanıdır?

- A. Range
- B. File
- C. Submit
- D. Text
- E. Checkbox

6 Textarea elemanının satır sayısı aşağıdakilerden hangisi ile belirlenir?

- A. Cols
- B. Rows
- C. Min
- D. Max
- E. Step

7 `<input type="range"...>` komutu hangi form elemanını yaratmak için kullanılır?

- A. Onay kutusu
- B. Radyo düğmesi
- C. WEB adresi
- D. Düğme
- E. Aralıklı kaydırma

8 Aşağıdaki ifadelerden hangisinde yanlışlık vardır?

- A. `<input type="range" autocomplete>`
- B. `<input type="button"...></input>`
- C. `<button type="reset"> </button>`
- D. `<datalist></datalist>`
- E. `<textarea></textarea>`

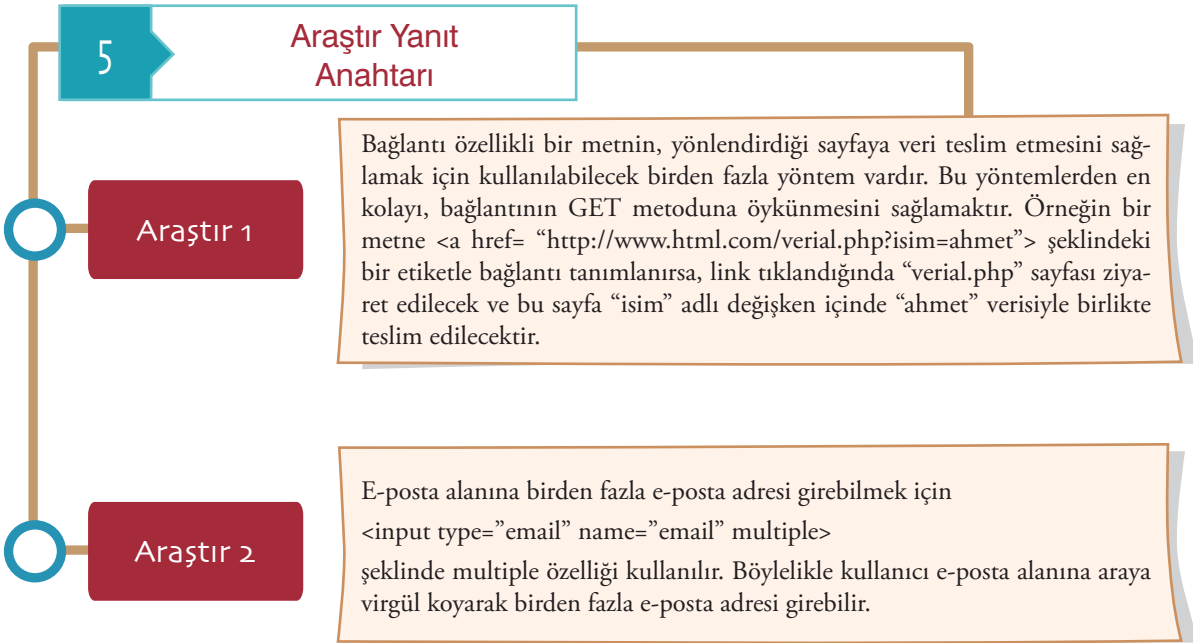
9 Aşağıdakilerden hangisinde belirli bir tarihten sonra veri girişi yapılabilmesini sağlayan form elemanı ve özelliği kullanılmıştır?

- A. `<input type="date" max="2010-02-04">`
- B. `<input type="checkbox" step="2010-02-04">`
- C. `<input type="date" min="2010-02-04">`
- D. `<input type="radio" maxlenght="2010-02-04">`
- E. `<input type="url" minlenght="2010-02-04">`

10 Aşağıdakilerden hangisi tarayıcılar tarafından desteklenmeyen bir form elemanıdır?

- A. Checkbox
- B. Datalist
- C. Password
- D. Date
- E. Tel

1. D	Yanıtınız yanlış ise “Form Yapısı” konusunu yeniden gözden geçiriniz.	6. B	Yanıtınız yanlış ise “Textarea (Metin Alanı)” konusunu yeniden gözden geçiriniz.
2. C	Yanıtınız yanlış ise “Form Özellikleri” konusunu yeniden gözden geçiriniz.	7. E	Yanıtınız yanlış ise “Range (Aralıklı Kaydırma)” konusunu yeniden gözden geçiriniz.
3. A	Yanıtınız yanlış ise “Form Elemanları” konusunu yeniden gözden geçiriniz.	8. B	Yanıtınız yanlış ise “Button (Düğme)” konusunu yeniden gözden geçiriniz.
4. D	Yanıtınız yanlış ise “Input (Giriş)”, “Select, Optgroup, Option (Açılır Liste, Gruplama Başlığı, Seçenek)” ve “Textarea (Metin Alanı)” konusunu yeniden gözden geçiriniz.	9. C	Yanıtınız yanlış ise “Date (Tarih)” ve “Input (Giriş)” konusunu yeniden gözden geçiriniz.
5. A	Yanıtınız yanlış ise “Form Elemanları” konusunu yeniden gözden geçiriniz.	10. E	Yanıtınız yanlış ise “Tel (Telefon Numarası)” konusunu yeniden gözden geçiriniz.



internet kaynakları

<https://www.w3.org/TR/html5/forms.html>, Erişim Tarihi: 30.06.2016.

<https://www.w3.org/wiki/HTML/Elements/form>, Erişim Tarihi: 30.06.2016.

<https://www.w3.org/TR/html-markup/button.html>, Erişim Tarihi: 30.06.2016.

<http://www.koodla.com/2008/07/20/css-ile-form-tasarim-mantigi/>, 02.07.2016.

Bölüm 6

Bölüm ve Kanvas Yapılarının Kullanımı

öğrenme çıktıları

Bölüm Özelliklerinin Belirlenmesi

- 1 Bölüm oluşturma etiketlerini kullanabilme
- 2 Bölüm oluşturma etiketlerini biçimlendirebilme

Menü Uygulamaları

- 3 Yatay, dikey, açılır menüler oluşturabilme

Esnek Bölüm Tasarımı

- 4 Ortam sorgusu kullanarak esnek bölüm tasarımı yapabilme

Kanvasların Kullanımı

- 5 Grid yapısı kullanarak sayfa tasarımı yapabilme
- 6 Kanvas üzerine çizim yapabilme

Anahtar Sözcükler: • Bölüm • Div • Float • Display • Position • Overflow • Menü • Esnek Tasarım • Ortam Sorgusu • Grid • Kanvas



GİRİŞ

Web sayfalarının estetik açıdan güzel olması ve kullanıcılar tarafından kolay kullanılabilir olması daha çok kullanıcıya ulaşması bakımından önemlidir. Eğer kullanıcılar web sayfasının tasarımını beğenmezlerse o web sayfasını ziyaret etmekten vazgeçebilirler. Bu bakımdan web sayfasındaki elemanların görsel tasarımının yapılması ve sayfada yerleşimlerinin doğru şekilde ayarlanması gerekir. HTML etiketleme dilinde sayfa içi elemanların yerleşimini yapmak için bölümlerin oluşturulması gerekmektedir. Bölümlerin düzgün şekilde oluşturulması, web sayfasının istenen görünümde olabildiğini sağlamak açısından büyük öneme sahiptir. Günümüzde web sayfaları, sadece düz metinden oluşmamaktadır ve sadece masaüstü/dizüstü bilgisayarlar ile görüntülenmemektedirler. Web sayfaları; başlık, alt bilgi, gezinti, temasal bölümler, makale, yan bilgi, vb. bir çok bölümden oluşabilirler ve masaüstü/dizüstü bilgisayar, tablet bilgisayar ve cep telefonları gibi farklı ortamlarda görüntülenmeleri gerekebilir. İşte bu noktada web sayfalarının farklı ortamlarda istenen şekilde görüntülenebilmesi için görsel tasarımının yapılması, bölümlerinin çok iyi tasarlanması ve biçimlendirilmesi gerekmektedir. Özellikle HTML5 ve CSS3 sürümleri ile bölüm tasarımına yönelik olarak web tasarımcılarının kullanabilecekleri bir çok yenilik gelmiştir. HTML dilinde sadece bölüm oluşturmaya yönelik div, header, footer, article gibi çeşitli etiketler bulunmaktadır. Bu tür etiketler kapsayıcı etiketlerdir ve diğer etiketleri kendi etiket bloğu içinde bulundurabilirler. Bu etiketlerden en önemlisi, genel amaçlı olarak kullanılan div etiketidir. Bu bölümde HTML dilinde bölüm oluşturma etiketlerinin kullanımı ve CSS dili ile biçimlendirilmesi konusunda bilgi verilecektir.

Div Etiketi

Div etiketi HTML dilinde sıklıkla bölümler oluşturmak için kullanılan, varsayılan display özelliği block olan kapsayıcı bir etikettir. Div etiketinin kullanım amacı sayfa içinde bölümler oluşturmaktır. Sayfa tasarımına göre iç içe bölümler de oluşturulabilir.

Şekil 6.1'de görüldüğü üzere tüm içerik id'si sayfa olan bir div içine alınmıştır. Sayfa id'li div içinde ise ustbolum, ortabolum ve altbolum id'li div'ler bulunmaktadır. Ortabolum id'li div içinde ise menu ve icerik id'li div'ler bulunmaktadır. İcerik id'li div'in içinde ise class'ı ozet olan div'ler yer yer almaktadır. Görüldüğü üzere div'ler kullanılarak bir HTML sayfası bölümlendirilmiştir. Tabi bu bölümlendirmeler CSS ile biçimlendirildiği zaman daha da anlam kazanacaktır. Şekil 6.2'de yukarıdaki kodun oluşturabileceği web sayfasının taslağı verilmiştir.



dikkat

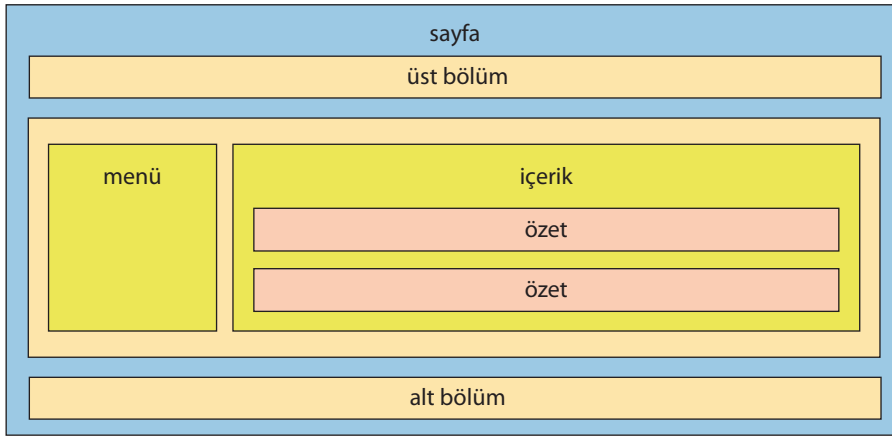
Genellikle div ile bölümler oluşturulurken, oluşturulan bölüm, id ya da class özelliği ile işaretlenir. Böylece CSS kullanılarak bu bölümler rahatlıkla biçimlendirilebilecektir.

```

1 <div id="sayfa">
2   <div id="ustbolum">...</div>
3   <div id="ortabolum">
4     <div id="menu">...</div>
5     <div id="icerik">
6       <div class="ozet">...</div>
7       <div class="ozet">...</div>
8     </div>
9   </div>
10  <div id="altbolum">...</div>
11 </div>

```

Şekil 6.1 Div etiketi örnek kullanımı



Şekil 6.2 Sayfa bölümleri örneği

Şekil 6.2’de görülen sayfa elemanlarının bu şekilde yerleşebilmelerini sağlamak için ilgili elemanların CSS ile biçimlendirilmesi gerekmektedir. İlerleyen bölümlerde bu biçimlendirme hakkında bilgi verilecektir.

Div etiketi dışında bölüm oluşturmak amacıyla kullanılacak HTML5 sürümü ile birlikte gelen yeni etiketler de vardır. Bu etiketlerden bazıları header, footer, section, article, nav, aside, main gibi etiketleridir. HTML5 sürümü ile, yazılan kodların daha anlamsal (semantik) olması amaçlanmıştır. Aslında işlevsellik olarak bu etiketlerin div etiketinden farklı yoktur. Div ve span gibi etiketler anlamsal etiketler değildir. Ancak header, footer, article gibi etiketler anlamsal etiketlerdir ve bu etiketlerin içine yazılması gereken bilgi konusunda açık şekilde bir tanımlama yapmaktadırlar. HTML5 ile gelen bu tür anlamsal etiketleri kullanmanın en önemli avantajı yazılan kodun hem programcılar tarafından hem de tarayıcı tarafından çok daha rahat şekilde anlaşılabilir olması ve bir standart getirmiş olmasıdır. HTML4 ile farklı programcılar başlık bölümü için `<div id="header">`, `<div id="ustbolum">` gibi farklı tanımlamalar yapabilirken HTML5 ile tüm programcılar `<header>` şeklinde bir tanımlama yapabilecektir ve bu durum web sayfalarında bir standardın oluşması sağlamaktadır. Ayrıca arama motorları web sayfalarını indekslerken HTML5’in anlamsal etiketlerinden yararlanmaktadırlar. Böylece geliştirilen web sayfası arama motorlarında en doğru şekilde indekslenebilecektir. HTML5’in anlamsal etiketleri tüm modern tarayıcılar tarafından desteklenmektedir.

✓ Div kelimesi İngilizce division kelimesinin kısaltmasıdır ve bölüm, ayırım, kısım anlamlarına gelmektedir.

Bölüm oluşturmak için kullanılacak anlamsal etiketlerden biri olan Header etiketi; web sayfasının başlık bölümü, logo, üst menü gibi içerikleri gruplandırmak için kullanılabilir. Genellikle header etiketi web sayfasının en üstün yer alır ve web sayfasının tüm başlık bölümünü kapsar. Ancak web sayfasının tasarımına göre header etiketinin kullanıldığı yerler değişiklik gösterebilir. Web sayfasının farklı bölümlerindeki içerikler için özel başlık alanları oluşturmak istenebilir. Bu durumda her bölümün başında bir header etiketi konulabilir.

Header etiketine benzer şekilde footer etiketi de sayfa sonunda bulunan içeriği gruplamak için kullanılmaktadır. Genellikle her sayfanın sonundaki içeriği gruplamak için bir footer etiketi kullanılır. Ancak web sayfasının tasarımına göre footer etiketinin kullanıldığı yerler değişiklik gösterebilir. Web sayfasının farklı bölümlerindeki içerikler için özel sayfa sonu alanları oluşturmak istenebilir. Bu durumda her bölümün sonuna bir footer etiketi konulabilir. Footer etiketi çoğunlukla sayfa sonunda ya da bölümde sonunda kullanılır. Ancak bazı durumlarda sayfa başında ya da bölüm başında da kullanılması mümkün olabilir. Örneğin bir blog yazısı ile ilgili yazar bilgileri blogun başında verilmek istenebilir. Bu

✓ Header kelimesi başlık, footer kelimesi ise alt bilgi anlamlarına gelmektedir.

durumda anlamsal olarak bu tür bilgileri footer etiketi içinde vermek daha uygun olacağı için footer etiketi bu amaçla kullanılabilir.

Bir başka anlamsal etiket olan section etiketi, web sayfasındaki içeriğin tematik olarak gruplandırılmasına yönelik bir etikettir. İlgisiz içeriğin gruplamasında kullanılmamalıdır. Eğer web sayfasında section etiketi kullanılıyorsa, section etiketinin içinde header ve footer etiketlerinin kullanılması uygun olacaktır, çünkü her tema kendi içinde bir başlık bölümü ve alt bilgi bölümü barındırabilir. Örneğin; bir gazete web sayfasındaki yazılar; siyaset, ekonomi, spor gibi temalandırılması durumunda her bir tema section etiketleri içinde gruplandırılabilir.

✓ Section kelimesi kesit, kısım, bölüm anlamına gelmektedir.

Article etiketi, kendi başına anlamlı ve diğer içeriklerden bağımsız olan içeriğin gruplandırılmasında kullanılır. Örneğin; gazete web sayfasındaki bir yazı, bir blog sayfasındaki bir yazı article olarak değerlendirilebilir.

✓ Article kelimesi makale anlamına gelmektedir.

Nav etiketi web sayfasındaki web bağlantılarını gruplandırmak için kullanılması gereken bir etikettir. Web sayfasındaki menü bağlantıları nav etiketi içinde gruplandırılabilir. Web sayfasında birden fazla bu etiketten kullanmak mümkündür. Nav etiketinin web sayfasındaki veya web sayfasının bir bölümündeki menü veya alt menü bağlantılarını gruplamak için kullanılması tavsiye edilmektedir (Goldstein, Lazaris ve Weyl, 2015). Makale içinde geçen bağlantılar, önemsiz bağlantılar veya ana menünün web sayfasının en altında tekrarlanması durumunda nav etiketinin kullanımından kaçınılmalıdır.

✓ Nav kelimesi navigation kelimesinin kısaltmasıdır ve gezinti, yön bulma anlamına gelir.

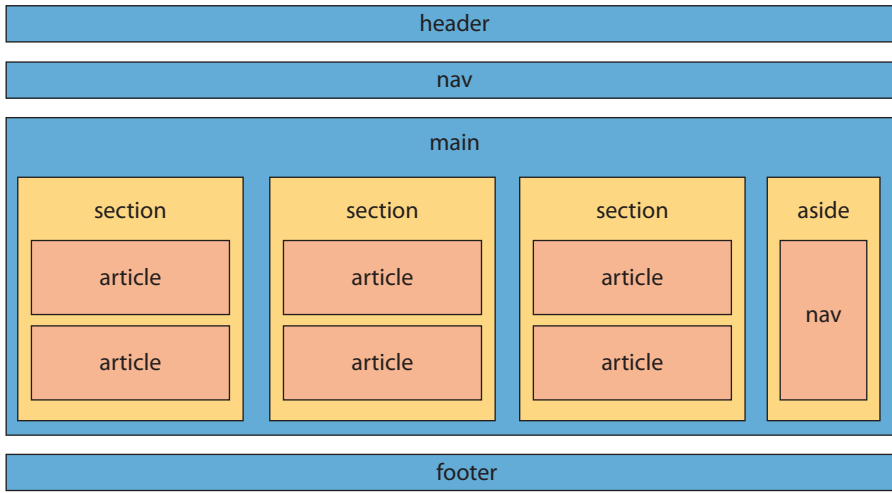
Aside etiketi, web sayfasında bulunan bir içerik ile ilgili olup da içerikten ayrı şekilde durması gereken bilgiler için kullanılmalıdır. Bu içerik tüm web sayfası olabileceği gibi web sayfası içerisinde bir article ya da section bölümleri de olabilir. Örneğin aside etiketi, bir makalenin yayınlandığı article etiketi ile birlikte kullanılabilir ve makalenin yazarı ile ilgili kısa özgeçmiş bilgisi barındırabilir. Aside etiketinin bir başka kullanımı ise web sayfasının bir bölümü ile ilgili bağlantıları barındırmasıdır. Örneğin canlılar alemi ile ilgili bir web sitesinin hayvanlar alemi ile ilgili bölümünde hayvanlar alemi ile ilgili alt menü bağlantıları aside etiketi içinde gruplandırılabilir.

✓ Aside kelimesi; ayrı, bir tarafa anlamlarına gelmektedir.

Main kelimesi; esas, başlıca, ana anlamlarına gelmektedir. Main etiketi bir belgenin ya da uygulamanın ana içeriğini temsil etmektedir. Bu bakımdan web sayfalarının ana içeriği main etiketi içinde gruplandırılabilir. Main etiketinin kullanımı ile ilgili farklı görüşler olmasına rağmen W3 Konsorsiyumuna göre; web sayfalarında sadece bir adet main etiketi kullanılmalıdır ve article, aside, footer, header, veya nav etiketleri içinde main etiketi kullanılmamalıdır. Ayrıca W3 Konsorsiyumu main etiketi için role="main" özelliğini kullanmayı tavsiye etmektedir.

HTML5 ile Sayfa Yapısı Oluşturma

HTML5 ile gelen anlamsal etiketleri kullanarak örnek bir web sayfasının nasıl oluşturulabileceğini inceleyelim. Çeşitli konularda günlük haberler veren bir web sitesinin Spor Haberleri sayfası geliştirilecek olsun. Geliştirilecek web sayfasının taslağı Şekil 6.3'deki gibi olsun.



Şekil 6.3 Haber web sitesinin taslağı

Web sitesinin en başında logo ve isim bilgilerinin bulunduğu header etiketi yerleştirilebilir. Header etiketinin altında web sitesinin ana bağlantılarının bulunduğu (Örneğin Ekonomi, Siyaset, Güncel, Spor, Finans, vb.) nav etiketi konulabilir. Nav etiketinin altında ise sayfanın ana içeriği olan bilgileri main etiketi içinde gruplandırılmıştır. Haberler 3 sütun şeklinde verilmektedir ve her bir sütun section etiketi içinde gruplandırılmıştır. Hatırlanacağı üzere section etiketi içinde gruplandırılacak içeriğin temasal bir bağının olması gerekir. Bu bağlamda birinci section etiketi içinde futbol haberleri, ikinci section etiketi içinde basketbol haberleri, üçüncü section etiketi içinde ise voleybol haberleri verilebilir. Eğer sütunlar arasında böyle temasal bir ayrım yapılmak istenmiyorsa section etiketi kullanılmamalı, bunun yerine div etiketi tercih edilmelidir. Section etiketlerinden sonra ise aside etiketine yer verilmiştir. Hatırlanacağı üzere aside etiketi tüm sayfa ya da sayfa içinde bir bölümle ilgili yan bilgi ya da ikincil bağlantılar için kullanılabilecek bir etikettir. Bu bağlamda aside etiketi içinde sayfanın tamamı spor haberleri ile ilgili olduğu için spor haberlerine yönelik alt menü amacıyla kullanılabilir. Sayfanın en sonunda ise künye, telif hakkı, vb. bilgiler footer içinde yer alabilir. Bu sayfanın kodlanmış şekli Şekil 6.4'te gösterilmiştir.

```

1 <body>
2   <header>...</header>
3   <nav>...</nav>
4   <main role="main">
5     <section>
6       <article>...</article>
7       <article>...</article>
8     </section>
9     <section>
10      <article>...</article>
11      <article>...</article>
12    </section>
13    <section>
14      <article>...</article>
15      <article>...</article>
16    </section>
17    <aside>
18      <nav>...</nav>
19    </aside>
20  </main>
21  <footer>...</footer>
22 </body>

```

Şekil 6.4 Haber web sitesi kodları

BÖLÜM ÖZELLİKLERİNİN BELİRLENMESİ

Bölümlerin HTML etiketleriyle nasıl oluşturulabileceği yukarıda anlatılmıştır. Güzel bir görüntü elde edebilmek için HTML etiketleri ile oluşturulan bu bölümlerin CSS ile biçimlendirilmesi gerekmektedir. Bu bölümde bölümlerin sayfa içindeki yerleşimlerinin CSS ile nasıl ayarlanabileceği konusu işlenecektir.



HTML etiketlerinin referansına <https://developer.mozilla.org/en-US/docs/Web/HTML/Reference> bağlantısını kullanarak ulaşabilirsiniz.

Display Özelliği

Display özelliği sayfa yerleşimini ayarlamak için kullanılan en önemli CSS özelliklerinden biridir. HTML dilinde tüm etiketler için varsayılan olarak tanımlanmış display isminde bir CSS özelliği bulunmaktadır. Varsayılan olarak atanan bu değer, genellikle inline veya block olmaktadır. Örneğin; Div, section, article, nav, p gibi etiketleri için varsayılan display özelliği block iken, span, i, img, a gibi etiketleri için ise varsayılan display özelliği inline olarak tanımlanmıştır. Display özelliği block olarak tanımlanan elemanlar yeni satırda başlarken, display özelliği inline olan elemanlar satır içinde yer almaktadırlar. Şekil 6.5'teki kodun denemesini yaparak sonuçlarını gözleyebilirsiniz.

```
1 <p>Display özelliği inline olan <i>i etiketi</i> satır
   içinde yer alır.</p><p>Display özelliği block olan p
   etiketi ise yeni satırdan başlar</p>
```

Şekil 6.5 Varsayılan display özelliği değerleri

CSS kullanarak etiketlerin varsayılan display özelliğini değiştirmek mümkündür. Örneğin; `a{display:block;}` CSS kodu kullanarak normalde display özelliği inline olan `<a>` etiketinin sayfada blok olarak görüntülenmesi sağlanabilir. Eğer bir etikete `display:none;` CSS özelliği eklenirse, bu etiket sayfada hiç görüntülenmeyecek, hatta hiç yer dahi kaplamayacaktır. Bu özellik JavaScript diliyle değiştirilebilir ve sayfa içinde bazı etiketlerin kullanıcı etkileşime bağlı olarak görünür ya da görünmez olmasını sağlamak için kullanılabilir. `visibility: hidden;` özelliği de etiketlerin sayfa da görünmez olması sağlamak için kullanılan bir etikettir. Ancak `visibility: hidden;` ile gizlenen etiketler sayfada yer kaplamaya devam ederler. Ancak görünmez olurlar. Şekil 6.6'daki örnekte id'si "deneme" olan div etiketi için CSS özelliğini değiştirerek sonuçlarını gözleyiniz.

```
1 <style >
2     #deneme {
3         display: none;
4     }
5 </style>
6 <div>Blok 1</div>
7 <div id="deneme">Blok 2</div>
8 <div>Blok 3</div>
```

Şekil 6.6 Display özelliği ile ilgili deneme

Display özelliği kullanarak bir etiketin tablo veya tablo hücresi gibi davranması da sağlanabilir. Bunun için de `display: table;`, `display: table-row;` ve `display: table-cell;` ifadeleri kullanılabilir. Şekil 6.7'de bununla ilgili bir örneğe yer verilmiştir.

```

1 <style>
2   #tablo{
3     display: table-cell;
4     border: 1px solid;
5   }
6   .satir{
7     display: table-row;
8   }
9   .hucre{
10    display: table-cell;
11    border: 1px solid;
12  }
13 </style>
14 <div id="tablo">
15   <div class="satir">
16     <div class="hucre">Hücre 1</div>
17     <div class="hucre">Hücre 3</div>
18     <div class="hucre">Hücre 3</div>
19   </div>
20   <div class="satir">
21     <div class="hucre">Hücre 4</div>
22     <div class="hucre">Hücre 5</div>
23     <div class="hucre">Hücre 6</div>
24   </div>
25 </div>

```

Şekil 6.7 Div etiketi ile tablo oluşturma

Örnekteki kodlar çalıştırıldığında 2 satır ve 3 sütundan oluşan bir tablo olduğu görülecektir. Div etiketlerinin CSS ile tablo, tablo satırı ve tablo hücresi gibi davranması sağlanmıştır.

Width ve Max-Width Özellikleri

Display özelliği block olan etiketler için width ve max-width değerlerini ayarlamak mümkündür. Varsayılan olarak bu etiketler bulunduğu ortamın tüm genişliğini kullanırlar. Eğer bu etiket doğrudan body etiketinin altında ise tarayıcı penceresinin en solundan en sağına kadar yer kaplayacaktır. Eğer bu etiket başka bir kapsayıcı etiketin altına yer alıyorsa, kapsayıcı etiketin sınırları dâhilinde olabilecek en yüksek genişliğe ulaşacaktır. Width özelliği ile ise etiketler için sabit bir genişlik tanımlaması yapılır ve bu etiket ve içindeki bileşenler bu genişlik içinde yer alırlar.

Bazı durumlar tanımlanan genişlik tarayıcının sayfa genişliğinden fazla da olabilir. Özellikle mobil cihazlarda ekran boyutlarının küçük olmasından dolayı bu durum söz konusu olabilir. Bu durumda elemanın genişliği tarayıcı penceresine sığmayacaktır ve tarayıcı tarafından yatay kaydırma çubukları görülmenecektir. Web sayfalarında yatay kaydırma çubukları web sayfasının kullanılabilirliğini düşürdüğü için tercih edilmemektedirler. Bu durumda width yerine max-width özelliği kullanıldığında elemanların genişlikleri sabit olmak yerine esnek olacaktır ve bulundukları ortamın genişliğine uyum sağlayarak kullanıcılara daha iyi bir kullanma deneyimi sunabilecektir. Max-width ile belirlenen değer, elemanın genişliği için en büyük değeri gösterir. Ancak tarayıcının sayfa genişliğinin değişmesine bağlı olarak bu değer tarayıcının sayfasına uyum sağlamak amacıyla azalabilir. Bu özellikler ile ilgili örnek kod Şekil 6.8'de verilmiştir.

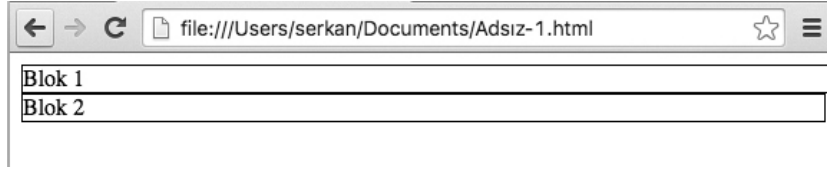
```

1 <style>
2   #deneme1 {
3     width: 700px;
4     border: 1px solid black;
5   }
6   #deneme2 {
7     max-width: 700px;
8     border: 1px solid black;
9   }
10 </style>
11 <div id="deneme1">Blok 1</div>
12 <div id="deneme2">Blok 2</div>

```

Şekil 6.8 Width ve max-width özelliklerinin kullanımı

Şekil 6.8'deki kodun çıktısı Şekil 6.9'te verilmiştir.



Şekil 6.9 Width ve Max-width özelliklerinin kullanımı ekran çıktısı

Yukarıdaki örnekte kodda görülen deneme1 ve deneme2 id'li div'ler sayfa genişliği 700px'in üzerinde olan tarayıcı penceresi için 700px'lik aynı görüntüye sahip olacaklardır. Ancak tarayıcı penceresinin genişliği azaltılırsa deneme1 id'li div'in genişliği sabit kalırken deneme2 id'li div'in genişliği tarayıcı penceresine orantılı olarak azalacaktır. Şekil'de tarayıcı penceresinin genişliği azaltılmıştır ve aradaki fark görülebilir.

Margin Özelliği

Margin özelliği bildiğiniz üzere HTML elemanlarının diğer elemanlarla arasındaki mesafeyi ayarlamak için kullanılmaktadır. Ancak margin özelliğinin bir başka kullanımı ise blok düzeyindeki etiketlerin yatay olarak ortalanmasını sağlamaktır. Bunun için etiketin margin-left ve margin-right özelliklerine ya da sadece margin özelliğine auto değerinin verilmesi gerekmektedir. Şekil 6.10'da bu işlemin nasıl yapılabileceği ile ilgili bir kod örneği yer almaktadır.

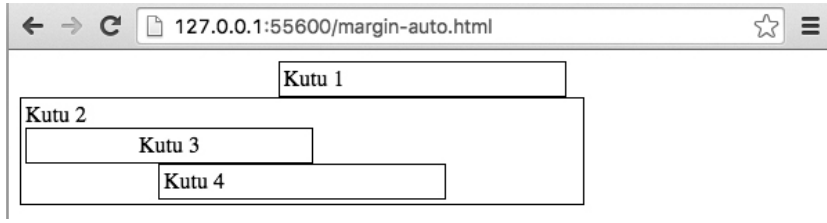
```

1  <style>
2      div {
3          padding: 3px;
4          border: 1px solid black;
5          width: 200px;
6      }
7      #kutu1 {
8          margin-left: auto;
9          margin-right: auto;
10     }
11     #kutu2 {
12         width: 400px;
13     }
14     #kutu3 {
15         text-align: center;
16     }
17     #kutu4 {
18         margin: auto;
19     }
20 </style>
21 <div id="kutu1">Kutu 1</div>
22 <div id="kutu2">
23     Kutu 2
24     <div id="kutu3">Kutu 3</div>
25     <div id="kutu4">Kutu 4</div>
26 </div>

```

Şekil 6.10 Margin özelliğinin kullanımı

Şekil 6.10'da verilen kodun çıktısı Şekil 6.4'te verilmiştir.



Şekil 6.11 Margin özelliğinin kullanımı ekran çıktısı

Şekilde de görüldüğü üzere kutu1 id'li div etiketi doğrudan body etiketinin altında yer aldığı için tarayıcı penceresinde yatay olarak ortalanmıştır. Kutu2 ve kutu3 id'li etiketlere margin özelliği ile ortalama işlemi uygulanmamıştır. Kutu3 id'li etikete text-align: center; özelliği uygulanmıştır. Şekilde de görüldüğü üzere kutu içindeki metin yatay olarak ortalanmıştır. Kutu4 id'li etikete ise margin: auto; ile ortalama uygulanmıştır. Kutu4 id'li div etiketi, kutu2 id'li etiketin çocuğu olduğu için kutu4 id'li etiket kutu2 id'li etiketin sınırları kapsamında ortalanmıştır.

Position Özelliği

Position kelimesi mevki, konum anlamları gelmektedir. Position özelliği ile elemanların ne tür bir konuma sahip olacağı belirlenir. Position özelliğine static, relative, fixed ve absolute değerleri verilebilmektedir. HTML sayfasındaki elemanların konumları left, right, top ve bottom özellikleri ile ayarlanabilir. Ancak bu ayarlamaların istenilen şekilde çalışabilmesini sağlamak için öncelikle position özelliği ile konum türü belirlenmelidir.

HTML elemanlarının varsayılan konum türü static'tir. Static kelimesi sabit, durağan anlamlarına gelmektedir. Bu tür elemanlar left (sol), right (sağ), top (üst) ve bottom (alt) özelliklerinden etkilenmezler ve sayfa akışına göre konumları otomatik olarak ayarlanır. Örneğin static özelliği sahip bir paragrafın üst tarafına başka bir static özelliğe sahip paragraf eklendiğinde altta kalan paragraf sayfada aşağı doğru kayar. Özetle static özelliğe sahip elemanların sayfada sabit bir konumları yoktur ve sayfa akışına göre konumları tarayıcı tarafından ayarlanır.

Relative kelimesi göreceli anlamına gelmektedir. Position özelliği relative olarak ayarlanan bir eleman için left, right, top ve bottom özelliklerine değer atanmaz ise static ile aynı şekilde davranış gösterecek ve sayfa akışına göre elemanın yerleşimi tarayıcı tarafından ayarlanacaktır. Ancak left, right, top veya bottom özelliklerine bir değer ataması yapılması durumunda bu eleman static olarak düşünüldüğünde normalde bulunması gereken konumdan belirtilen değerler kadar sapma meydana gelecektir. Şekil 6.12'de relative değeri ile ilgili kod örneğinin yer verilmiştir.

```

1  <style>
2      #div2 {
3          position: relative;
4          top: 10px;
5          left: 20px;
6      }
7  </style>
8  <div>İçerik 1</div>
9  <div id="div2">İçerik 2</div>
10 <div>İçerik 3</div>

```

Şekil 6.12 Relative değerinin kullanımı

Şekil 6.13'te, Şekil 6.12'de verilen kodun çıktısı verilmiştir.



Şekil 6.13 Relative değerinin kullanımı ekran çıktısı

Şekilde de görüldüğü üzere div2 id'li div elemanı normalde bulunması gereken yerden 10px aşağıda ve 20px sağda yer almıştır. Sayfadaki diğer elamanlar div2 id'li elemanın boşalttığı yeri doldurmaya çalışmamıştır veya yeri değişen div2 id'li elemana göre yer değiştirmemişlerdir.

Absolute kelimesinin anlamı mutlaktır. Position özelliği absolute olarak ayarlanan bir eleman için left, right, top veya bottom özelliklerine değer atanmalıdır. Böylece bu eleman için web sayfasında veya bulunduğu blok içinde sabit bir konum belirlenmiş olur. Mantık olarak left ile right ve top ile bottom özellikleri aynı anda kullanılmamalıdır. Bir başka deyişle bir elemanın hem sol koordinatını hem de sağ koordinatını ayarlamak mantıklı olmayacaktır. Birini ayarlayınca diğer koordinat zaten otomatik olarak belirlenmiş olacaktır. Eğer position özelliği absolute olarak ayarlanan bir eleman bir blok içinde yer almıyorsa koordinatları tarayıcı penceresi referans alınarak belirlenir, eğer position özelliği static olmayan başka bir blok içinde yer alıyorsa koordinatları bu bloğun sınırları referans alınarak belirlenir. Position özelliği absolute olarak ayarlanan bir eleman kaydırma çubuğu ile hareket eder. Ayrıca bu özelliğe sahip elemanlar sayfa akışından bağımsız oldukları için sayfadaki diğer elemanlar ile üst üste gelebilirler. Şekil 6.14'de position özelliğinin absolute olarak kullanımıyla ilgili örnek kod yer almaktadır.

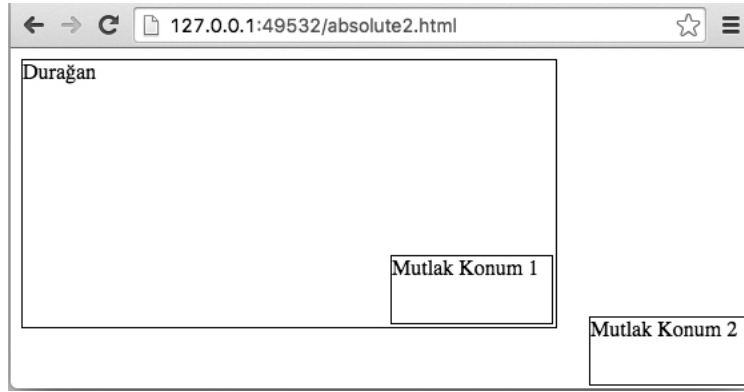
```

1  <style>
2      #kutu{
3          position: relative;
4          border: 1px solid black;
5          width: 400;
6          height: 200;
7      }
8      .mutlak{
9          border: 1px solid black;
10         position: absolute;
11         right: 2px;
12         bottom: 2px;
13         width: 100px;
14         height: 50px;
15     }
16 </style>
17 <div id="kutu">
18     Durağan
19     <div class="mutlak">
20         Mutlak Konum 1
21     </div>
22 </div>
23 <div class="mutlak">
24     Mutlak Konum 2
25 </div>

```

Şekil 6.14 Absolute değerinin kullanımı

Şekil 6.14'de verilen kodun çıktısı Şekil 6.15'de verilmiştir.



Şekil 6.15 Absolute değerinin kullanımı ekran çıktısı

Şekilde de görüldüğü üzere eğer position özelliği absolute olan eleman position özelliği static olmayan (örnekte relative kullanılmış) başka bir kapsayıcı elemanın içinde kullanıldığında, koordinatları kapsayıcı elemanın sınırları referans alınarak belirlenir. Eğer position özelliği absolute olan elemanın bir kapsayıcısı yok ise koordinatları tarayıcı penceresinin sınırları referans alınarak belirlenir. Daha öncede belirtildiği üzere eğer dikey kaydırma çubuğu kullanılırsa position özelliği absolute olan elemanlar sayfa içinde hareket edeceklerdir.

Fixed kelimesi sabitlenmiş anlamına gelmektedir. Position özelliği fixed olan bir elemanın koordinatları tarayıcı penceresi referans alınarak belirlenir ve bu eleman tarayıcı penceresine sabitlenir. Bir başka deyişle kaydırma çubukları ile sayfa içinde hareket edildiğinde bu elemanların tarayıcı penceresine göre yerleri değişmez. Bu eleman için left, right, top veya bottom özelliklerine değer atanmalıdır. Şekil 6.16'da fixed değerinin kullanımı ile ilgili kod örneği bulunmaktadır.

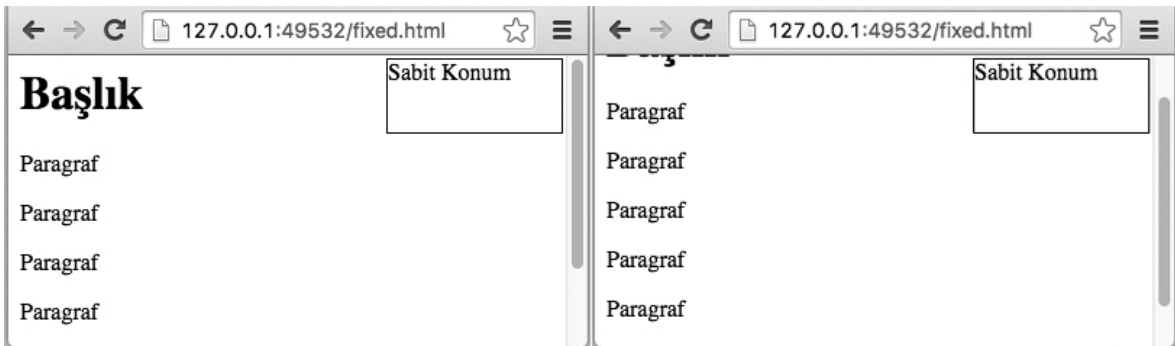
```

1  <style>
2      #sabit{
3          border: 1px solid black;
4          position: fixed;
5          right: 2px;
6          top: 2px;
7          width: 120;
8          height: 50;
9      }
10 </style>
11 <h1>Başlık</h1>
12 <p>Paragraf</p><p>Paragraf</p><p>Paragraf</p><p>Paragraf</p>
13 <div id="sabit">Sabit Konum</div>

```

Şekil 6.16 Fixed değerinin kullanımı

Şekil 6.17'de, Şekil 6.16'da verilen kodun çıktısına yer verilmiştir.



Şekil 6.17 Fixed değerinin kullanımı ekran çıktısı

Şekilde sol pencerede dikey kaydırma çubuğu görüldüğü üzere en üste bulunmaktadır. Sağdaki pencerede ise kaydırma çubuğu bir miktar aşağıya çekilmiştir. Ancak görüldüğü üzere position özelliği fixed olarak ayarlanmış olan elemanın yeri pencereye göre sabit kalmış ve kaydırma çubuklarından etkilenmemiştir.

Sayfa içinde position özelliği absolute, relative ve fixed olan elemanlar olduğu zaman, bu elemanların koordinatlarının kesişme ve üst üste gelme durumları söz konusu olabilir. Hangi elemanın üstte hangi elemanın altta görüneceğini ise html kodu içinde yazılma sırası ve z-index özellikleri ile belirlenmektedir. Yazılma sırası dikkate alındığında kodda en son yazılan eleman en üstte görünecektir. Bu durum z-index özelliği ile değiştirilebilir. Z-index özelliğine tam sayı değerler verilmelidir. Z-index değeri yüksek olan eleman, düşük olan elemanın üstünde görünecektir. Z-index özelliğine negatif tamsayı değerlerde vermek mümkündür. Bir elemanın z-index özelliğine negatif bir değer verildiği zaman varsayılan olarak z-index değerleri sıfır olan elemanların altında kalacaktır. Z-index özelliğinin kullanımı ile ilgili Şekil 6.18'de kod örneği verilmiştir.

```

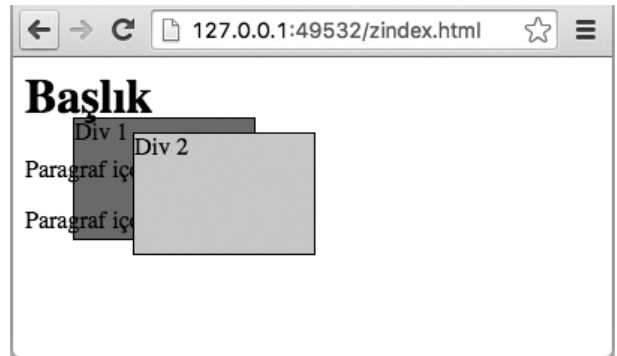
1  <style>
2      #div1{
3          border: 1px solid black;
4          position: absolute;
5          left: 40px;
6          top: 40px;
7          width: 120;
8          height: 80;
9          background-color: gray;
10         z-index: -1;
11     }
12     #div2{
13         border: 1px solid black;
14         position: absolute;
15         left: 80px;
16         top: 50px;
17         width: 120;
18         height: 80;
19         background-color: lightgray;
20         z-index: 1;
21     }
22 </style>
23 <h1>Başlık</h1>
24 <p>Paragraf içeriği</p>
25 <p>Paragraf içeriği</p>
26 <div id="div1">Div 1</div>
27 <div id="div2">Div 2</div>

```

Şekil 6.18 Z-index özelliğinin kullanımı

Şekil 6.19'da, Şekil 6.18'de verilen kodun çıktısına yer verilmiştir.

Şekilde de görüldüğü üzere div1 id'li elemanın z-index özelliğine -1 değeri verilmiş ve hem normal sayfa içeriğinin hem de div2 id'li elemanın altında kalmıştır. Div2 id'li elemana ise z-index özelliğine 1 değeri verilmiştir ve en büyük z-index değerine sahip olduğu için en üstte görünmektedir.



Şekil 6.19 Z-index özelliğinin kullanımı ekran çıktısı

Overflow Özelliği

Overflow kelimesi taşmak, dışına taşmak anlamlarına gelmektedir. Bu özellik, genişlik ve yükseklik özellikleri belirtilen elemanların içeriğinin sığmaması, taşması durumunda nasıl davranılacağı ile ilgilidir. Bu özellik visible, hidden, scroll ve auto olmak üzere 4 değer alabilmektedir. Varsayılan değer visible değeridir. Overflow özelliği visible olarak ayarlandığı zaman taşan içerik görünür durumda olur. Şekil 6.20’de visible değeri ile ilgili kod örneği verilmiştir. Div1 id’li etiket için overflow özelliği belirtilmemiştir. Bu durumda varsayılan değer olan visible değeri geçerli olacaktır.

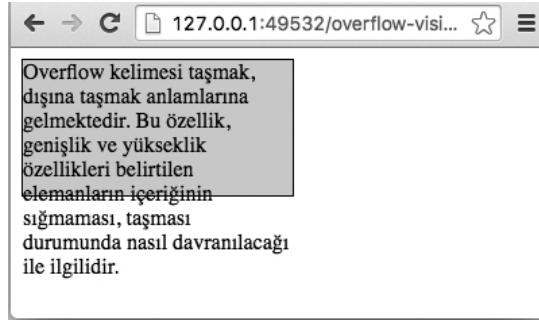
```

1 <style>
2   #div1{
3     border: 1px solid black;
4     width: 200;
5     height: 100;
6     background-color: lightgray;
7   }
8 </style>
9 <div id="div1">
10  Overflow kelimesi taşmak, dışına taşmak anlamlarına
11  gelmektedir. Bu özellik, genişlik ve yükseklik
    özellikleri belirtilen elemanların içeriğinin
    sığmaması, taşması durumunda nasıl davranılacağı ile
    ilgilidir.
12 </div>

```

Şekil 6.20 Overflow özelliği için visible değerinin kullanımı

Şekil 6.20’de, Şekil 6.21’de verilen kodun çıktısına yer verilmiştir.



Şekil 6.21 Overflow özelliği için visible değerinin kullanımı ekran çıktısı

Şekilde de görüldüğü üzere taşan içerik gizlenmemiş, sayda görünür şekilde yer almıştır. Overflow özelliğine hidden değerinin verilmesi durumunda ise taşan içerik gizlenir. Aşağıda hidden değerinin kullanımı ile ilgili örneğe yer verilmiştir.

```

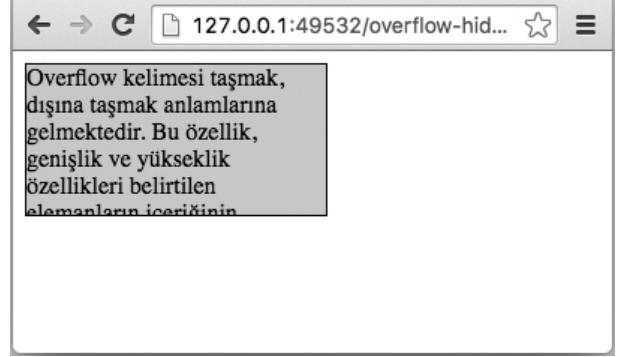
1 <style>
2   #div1{
3     border: 1px solid black;
4     width: 200;
5     height: 100;
6     background-color: lightgray;
7     overflow: hidden;
8   }
9 </style>
10 <div id="div1">
11  Overflow kelimesi taşmak, dışına taşmak anlamlarına
12  gelmektedir. Bu özellik, genişlik ve yükseklik
    özellikleri belirtilen elemanların içeriğinin
    sığmaması, taşması durumunda nasıl davranılacağı ile
    ilgilidir.
13 </div>

```

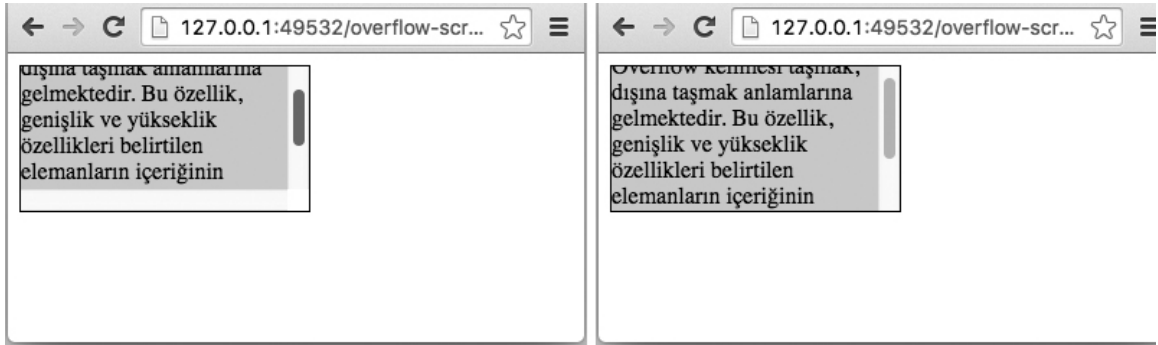
Şekil 6.22 Overflow özelliği için hidden değerinin kullanımı

Şekil 6.10'da yukarıdaki kodun çıktısına yer verilmiştir.

Şekilde de görüldüğü üzere taşan içerik gizlenmiştir, sayfada yer kaplamamaktadır. Overflow özelliğine scroll değerinin verilmesi durumunda ise ilgili elemanın sağ ve alt bölümünde kaydırma çubukları ortaya çıkar ve kullanıcı bu kaydırma çubuklarını kullanarak içeriği inceleyebilir. Şekil 6.24'de, Şekil 6.22'de verilen örnekteki overflow özelliğine scroll ve auto değerinin verilmesi durumunda oluşacak ekran çıktıları verilmiştir.



Şekil 6.23 Overflow özelliği için visible değerinin kullanımı ekran çıktısı



Şekil 6.24 Overflow özelliği için scroll ve auto değerinin kullanımı ekran çıktısı

Şekilde de görüldüğü üzere taşan içerik gizlenmiştir ve elemanın sağ ve alt bölümünde kaydırma çubukları için yerler açılmıştır. Yatay olarak taşan bir içerik bulunmadığı için yatay kaydırma çubuğu görüntülenmemiştir. Auto değeri de scroll değeri ile benzer şekilde çalışmaktadır. Auto değeri, kaydırma çubuklarının içeriğin büyüklüğüne göre otomatik olarak görünür olmasını veya gizlenmesini sağlar. Şekil 6.11'in sağ bölümünde overflow özelliğine auto değeri verilmesi durumunda ortaya çıkan görüntüye yer verilmiştir. Yatay olarak taşma olmadığı için yatay kaydırma çubuğu gizlenmiş, dikey taşma olduğu için ise dikey kaydırma çubuğu otomatik olarak yerleştirilmiştir.

Overflow ile benzer olan overflow-x ve overflow-y özellikleri de bulunmaktadır. Bu iki özellikte overflow ile aynı değerleri almaktadır. Adından da anlaşılacağı üzere overflow-x yatay eksenindeki davranışı, overflow-y ise dikey eksenindeki davranışı belirlemektedir.

Float Özelliği

Float kelimesi yüzmek, süzülme anlamına gelmektedir. Float özelliği ile elemanların sayfa içinde yüzmesi sağlanabilmektedir. None, left, right değerleri alabilmektedir. HTML etiketlerinin varsayılan float değeri none'dır. En basit kullanımı ile float özelliği kullanılarak sağ ya da sol tarafa yerleştirilen bir resmin etrafından metnin akması sağlanabilir. Bunun ile ilgili kod örneği Şekil 6.25'te verilmiştir.

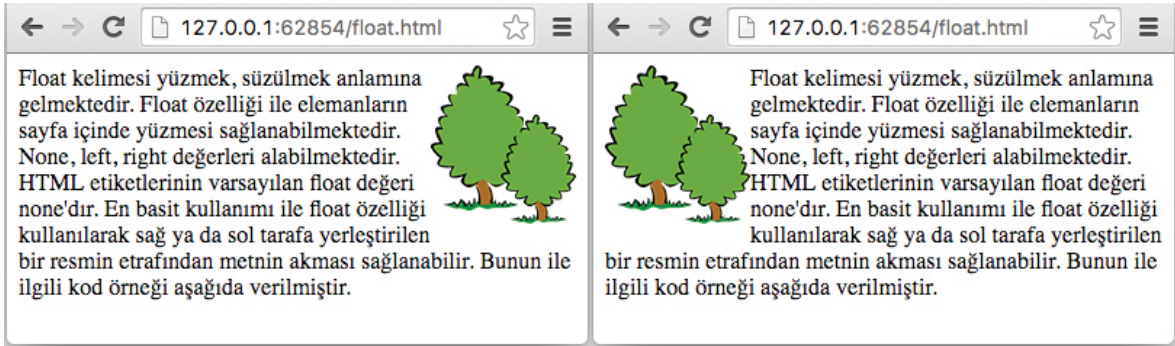
```

1 <style>
2     #resim{
3         float: right;
4     }
5 </style>
6 
7 Float kelimesi yüzmek, süzülme anlamına
   gelmektedir...

```

Şekil 6.25 Float özelliğinin kullanımı

Şekil 6.25'in sol bölümünde Şekil 6.25'teki kodun çıktısına yer verilmiştir.



Şekil 6.26 Float özelliğinin kullanımı ekran çıktısı

Şekilde de görüldüğü gibi resim için `float: right;` özelliği kullanılmıştır. Bu durumda resim sayfanın sağına yerleşmiş ve sonrasında gelen metin resmin etrafından akmıştır. Şekil 6.26'nın sağ bölümünde ise `float: left;` özelliği kullanılmıştır. Bu durumda resim sol tarafa yerleşmiş ve sonrasında gelen metin resmin etrafından akmıştır. Float özelliği sadece resim için kullanılan bir özellik değildir. Farklı etiketler içinde float özelliğinden yararlanılabilir. Örneğin Şekil 6.27'de verilen örnek kodda `div` etiketi için float özelliği kullanılmıştır.

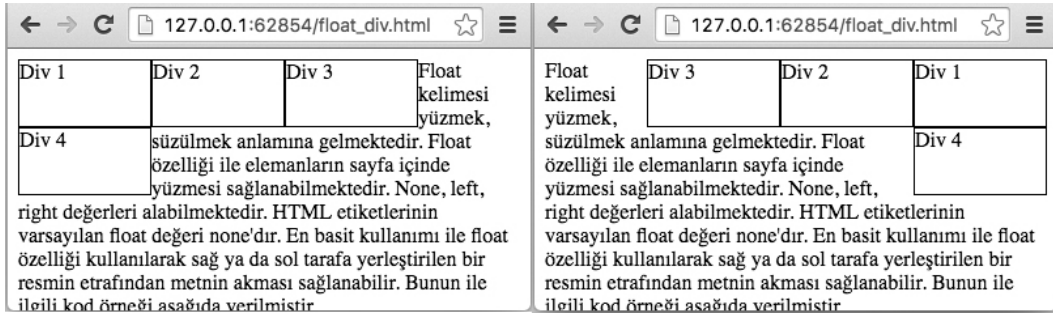
```

1 <style>
2     .solda{
3         float: left;
4         width:100px;
5         height: 50px;
6         border: 1px solid;
7     }
8 </style>
9 <div class="solda">Div 1</div>
10 <div class="solda">Div 2</div>
11 <div class="solda">Div 3</div>
12 <div class="solda">Div 4</div>
13 Float kelimesi yüzmek, süzülme anlamına gelmektedir ...

```

Şekil 6.27 Div etiketi için float özelliğinin kullanımı

Şekil 6.28'in sol bölümünde Şekil 6.27'de verilen kodun çıktısına yer verilmiştir.



Şekil 6.28 Div etiketi için float özelliğinin kullanımı ekran çıktısı

Soldaki şekilde de görüldüğü üzere 4 adet float özelliği left olan div etiketi bulunmaktadır. Bu elemanlar sola hizalı şekilde yan yana yerleşmişlerdir. Ancak 4. div elemanı sayfaya sığmadığı için alta geçmiştir. Div etiketlerinden sonra gelen metin ise boşlukları dolduracak şekilde div elemanlarının etrafından akmıştır. Sağ taraftaki resimde ise float: left; yerine float: right; kullanılmıştır. Bu durumda ise div etiketlerin sayfanın sağına hizalı şekilde yerleştikleri görülmektedir.

Yukarıdaki örneğe göre div etiketlerinde sonra gelen metnin daha önce kullanılan float özelliklerinden etkilenmeden yeni bir satırdan başlaması istenirse clear özelliği kullanılabilir. Clear özelliği left, right ve both değerleri almaktadır. Left değeri float: left; özelliğini, right değeri float: right; özelliğini, both değeri ise hem float: left; hem de float: right; değerini temizler. Clear özelliği ile ilgili kod örneği aşağıda verilmiştir.

```

1 <style>
2   .solda{
3     float: left;
4     width: 100px;
5     height: 50px;
6     border: 1px solid;
7   }
8   .temizle{
9     clear: left;
10  }
11 </style>
12 <div class="solda">Div 1</div>
13 <div class="solda">Div 2</div>
14 <div class="solda">Div 3</div>
15 <div class="solda">Div 4</div>
16 <p class="temizle">Float kelimesi yüzmek, süzülme
kelimesi anlamına gelmektedir... </p>

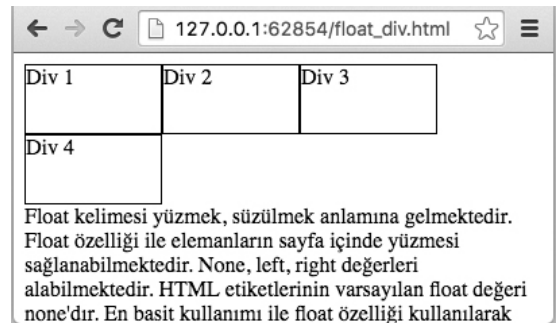
```

Şekil 6.29 Clear özelliğinin kullanımı

Şekil 6.29'da verilen kodun çıktısı Şekilde 6.30'da verilmiştir.

Şekilde de görüldüğü üzere metin için clear: left; kullanıldığında yukarıda kullanılan float: left; özellikleri temizlenmiş ve metin yeni bir satırdan başlamıştır.

Eğer float özelliği atanan bir eleman başka bir elemanın içinde yer alıyorsa ve bu elemanın uzunluğu kapsayan elemanın uzunluğundan fazla ise kapsayan elemanın sınırlarının dışına taşma durumu söz konusu olacaktır. Bu problemle ilgili kod örneği Şekil 6.31'de verilmiştir.



Şekil 6.30 Clear özelliğinin kullanımı ekran çıktısı

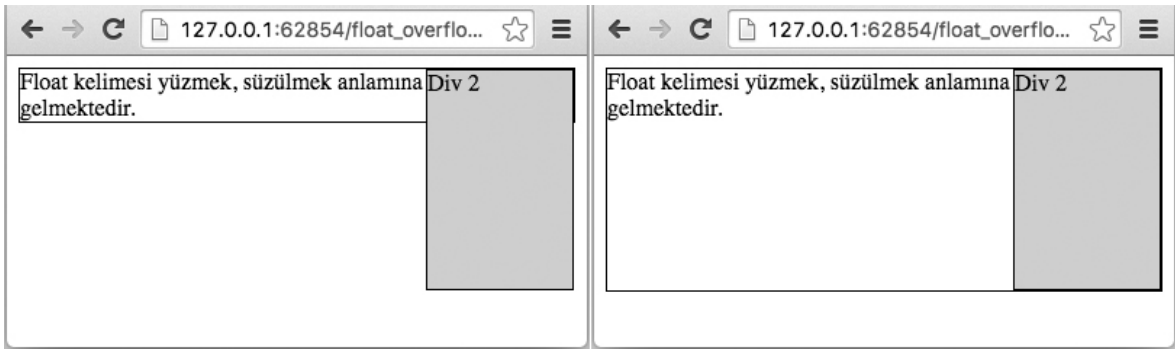

```

1 <style>
2   .sagda{
3       float: right;
4       width: 100px;
5       height: 150px;
6       border: 1px solid;
7       background-color: lightgray;
8   }
9   #kapsayici{
10      border: 1px solid;
11  }
12 </style>
13 <div id="kapsayici">
14   <div class="sagda">Div 2</div>
15   Float kelimesi yüzmek, süzölmek anlamına gelmektedir.
16 </div>

```

Şekil 6.31 Float özelliği kullanılan elemanların kapsayıcının sınırlarını aşması

Şekil 6.31’de verilen kodun çıktısı Şekil 6.32’nin sol bölümünde verilmiştir.



Şekil 6.32 Float özelliği kullanılan elemanların kapsayıcının sınırlarını aşması ekran çıktısı

Şekilde de görüldüğü üzere kapsayıcı id’li eleman class’ı sagda olan elemanı kapsıyor olmasına rağmen class’ı sagda olan eleman sınırlarını aşmıştır ya da bir başka ifade ile id’si kapsayıcı olan eleman, sınırlarını class’ı sagda olan elemanı kapsayacak şekilde genişletmemiştir. Bu durumu düzeltmek için id’si kapsayıcı olan eleman için `overflow: auto;` özelliğinin eklenmesi gerekmektedir. id’si kapsayıcı olan eleman için `overflow: auto;` özelliği eklendiğinde oluşacak görüntü Şeklin sağ bölümünde verilmiştir. Bu şekilde de görüldüğü üzere id’si kapsayıcı olan eleman sınırlarını uygun şekilde büyötmüştür. Float özelliği kullanarak rahatlıkla güzel görünümlü bir resim galerisi yapılabilir. Bunun ile ilgili örnek bir kod Şekil 6.33’de verilmiştir.

```

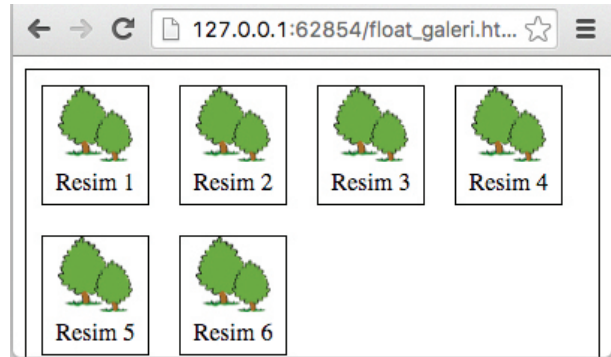
1  <style>
2      .solda{
3          float: left;
4          width: 70px;
5          border: 1px solid;
6          margin: 10px;
7      }
8      .solda img{
9          margin-left: 10px;
10         width: 50px;
11         height: 50px;
12     }
13     .solda p{
14         margin: 5px;
15         text-align: center;
16     }
17     #kapsayici{
18         border: 1px solid;
19         overflow: auto;
20     }
21 </style>
22 <div id="kapsayici">
23     <div class="solda"><p>Resim 1</p></div>
24     <div class="solda"><p>Resim 2</p></div>
25     <div class="solda"><p>Resim 3</p></div>
26     <div class="solda"><p>Resim 4</p></div>
27     <div class="solda"><p>Resim 5</p></div>
28     <div class="solda"><p>Resim 6</p></div>
29 </div>

```

Şekil 6.33 Float özelliği ile resim galerisi oluşturma

Şekil 6.33’de verilen kodun çıktısı Şekil 6.34’de verilmiştir.

Şekilde de görüldüğü üzere resimleri kapsayan div’in float özelliği left olarak ayarlanmıştır. Böylece resimleri içeren kutular sola hizalı olarak yan yana yer almışlardır. Ayrıca sayfanın genişliği arttırıldığında sayfaya sığacak kutusu sayısı artacağı için sayfa genişliğine göre Resim 5 ve Resim 6 üst satıra geçebileceklerdir. Böylece mobil cihazlar gibi ekran büyüklüğü farklı olan cihazlarda resim galerisi uygun şekilde görüntülenebilecektir.



Şekil 6.34 Float özelliği ile resim galerisi oluşturma ekran çıktısı

Display Özelliği İçin Inline-Block Değeri

Float özelliği dışında resim galerisinde olduğu gibi sayfada yüzen elemanlar oluşturmanın bir başka yolu da display özelliğine inline-block değeri vermektir. Bu özellik kullanıldığında float özelliğini kullmaya gerek kalmamaktadır. Ayrıca normalde float özelliği kullanıldıktan sonra clear özelliği kullanılarak temizlik yapmak gerekmektedir. Ancak inline-block ile buna da gerek yoktur. Inline-block değerinin kullanımı ile ilgili kod örneği Şekil 6.35’te verilmiştir.

```

1 <style>
2   .kutu{
3     display: inline-block;
4     width: 70px;
5     height: 30;
6     border: 1px solid;
7     margin: 2px;
8   }
9 </style>
10 <div class="kutu">Kutu 1</div>
11 <div class="kutu">Kutu 2</div>
12 <div class="kutu">Kutu 3</div>
13 <div class="kutu">Kutu 4</div>
14 <div class="kutu">Kutu 5</div>
15 <div class="kutu">Kutu 6</div>
16 <div>Daha sonra gelen içerik</div>

```

Şekil 6.35 Inline-block değerinin kullanımı

Şekil 6.35'te verilen kodun çıktısı Şekil 6.36'da verilmiştir.



Şekil 6.36 Inline-block değerinin kullanımı ekran çıktısı

Şekilde de görüldüğü üzere display özelliği inline-block olan div elemanları sayfada yan yana durmaktadır. Ayrıca float özelliğinde olduğu gibi daha sonra gelen içerik için clear özelliğini kullanmaya gerek kalmamıştır.



dikkat

CSS özelliklerinin alabileceği değerler ile ilgili olarak bir CSS referansından yararlanılması uygun olacaktır.



CSS Referansı <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

Örnek Uygulama

Buraya kadar web sayfasında nasıl bölümler oluşturulabileceği ve bu bölümlerin nasıl biçimlendirilebileceği konularına değinilmiştir. Bu bölümde ise HTML5'in anlamsal etiketlerinin kullanıldığı bir web sayfasının genel yapısının nasıl oluşturulabileceği ile ilgili bir örneğe yer verilmiştir. Bu örnek uygulama ile ilgili kodlar Şekil 6.37'de verilmiştir.

```

1  <header>
2  <div>
3      
4      <h1>Site Başlığı</h1>
5  </div>
6  </header>
7  <nav class="anamenu">Sitenin Ana Menüsü Burada Olacaktır.</nav>
8  <main role="main">
9  <div>
10     <section>
11         <h2>Bölüm 1</h2>
12         <article>Makale 1</article>
13         <article>Makale 2</article>
14     </section>
15     <section>
16         <h2>Bölüm 2</h2>
17         <article>Makale 1</article>
18         <article>Makale 2</article>
19     </section>
20     <aside>
21         <h3>Alt Menü</h3>
22         <nav class="altmenu">İlgili Bölüme Ait Alt Menü Seçenekleri Yer Alacaktır.</nav>
23     </aside>
24 </div>
25 </main>
26 <footer>
27     <div>Telif Hakkı Bilgisi &copy; 2016</div>
28 </footer>

```

Şekil 6.37 HTML5 bölüm oluşturma etiketleri ile örnek uygulama

Şekil 6.37’de verilen kodların çıktısı Şekil 6.38’de verilmiştir.



Şekil 6.38 HTML5 bölüm oluşturma etiketleri ile örnek uygulama ekran çıktısı

Şekilde de görüldüğü üzere yazılan HTML etiketleri yukarıdaki bölümlerde anlatılan biçimlendirme özellikleri kullanılarak CSS ile biçimlendirilmiştir. Ana menü ve alt menülerin nasıl oluşturulabileceği bir sonraki bölümde anlatılmıştır. Yukarıdaki örneğe benzer olarak HTML ve CSS kodları düzenlenerek çok farklı sayfa tasarımları yapılabilir.

Öğrenme Çıktısı



MENÜ UYGULAMALARI

Web sitelerinde kullanıcıların rahatlıkla gezinebilmesi için menü veya menüler oluşturmak gerekmektedir. Bu bölümde nasıl menü oluşturulabileceği konusunda bilgi verilecektir. Daha önce de bahsedildiği üzere web sayfasının ana menüsünü veya web sayfasının bir bölümü ile ilgili menüleri oluştururken nav etiketini kullanmak tavsiye edilmektedir. Oluşturulacak menü sayfa tasarımına göre yatay ya da dikey olabilmektedir. Menüler aslında bağlantılardan oluşan bir listedir. Bu bağlamda menü oluşturmak için ul etiketi ile bağlantılardan oluşan sırasız bir liste oluşturmak kabul görmüş bir yöntemdir. Menü oluşturmak için kullanılacak kod örneği Şekil 6.39’da verilmiştir.

```

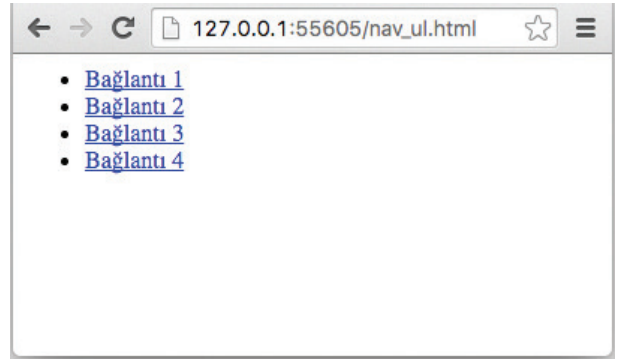
1 <nav class="anamenu">
2   <ul>
3     <li><a href="#" class="etkin">Bağlantı 1</a></li>
4     <li><a href="#">Bağlantı 2</a></li>
5     <li><a href="#">Bağlantı 3</a></li>
6     <li><a href="#">Bağlantı 4</a></li>
7   </ul>
8 </nav>

```

Şekil 6.39 Biçimlendirilmemiş menü örneği

Yukarıdaki kod incelendiğinde nav etiketinin içinde sırasız liste şeklinde oluşturulmuş bağlantı listesi olduğu görülecektir. Şekil 6.39'da verilen kodun çıktısı Şekil 6.40'daki gibidir.

Şekilde de görüldüğü üzere CSS ile biçimlendirme yapılmadığı zaman oluşturulan menü standart bir sırasız liste şeklinde görülmektedir. Bu listeyi web sayfasının tasarımına uygun hâle getirmek için CSS ile biçimlendirmek gerekir. Örneğin yukarıdaki kodları verilen listenin 900px genişliğinde yatay bir listeye dönüştürülmek istenmesi durumunda Şekil 6.41'deki css kodları kullanılabilir.



Şekil 6.40 Biçimlendirilmemiş menü örneği ekran çıktısı

✓ ul etiketi kullanmadan da CSS ile uygun şekilde biçimlendirerek benzer menülerin oluşturulması mümkündür.

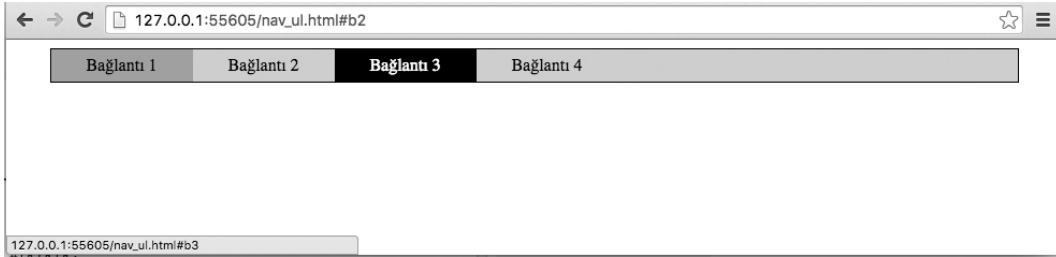
```

1 <style>
2   nav.anamenu {
3     max-width: 900px;
4     background-color: lightgrey;
5     border: thin solid;
6     overflow: auto;
7     margin: auto;
8   }
9   nav.anamenu ul {
10    list-style-type: none;
11    margin: 0;
12    padding: 0;
13  }
14  nav.anamenu li {
15    display: inline;
16    float: left;
17  }
18  nav.anamenu a {
19    display: block;
20    width: 120px;
21    text-align: center;
22    padding: 6px;
23    text-decoration: none;
24    color: black;
25  }
26  nav.anamenu a.etkin {
27    background-color: darkgrey;
28    color: white;
29  }
30  nav.anamenu a:hover:not(.etkin) {
31    color: white;
32    background-color: black;
33  }
34 </style>

```

Şekil 6.41 Yatay menü için örnek css kodları

Şekil 6.39'da verilen örneğe Şekil 6.41'deki CSS kodları uygulandığında Şekil 6.42'deki görüntü elde edilecektir.



Şekil 6.42 Yatay menü ekran çıktısı

Kodlar incelendiğinde öncelikle tüm seçicilerin nav. anamenu ile başladığı görülecektir. Böylece yapılan bu işlemde sayfa içinde diğer bağlantılar veya menüler etkilenmeyecektir. Sadece class'ı anamenu olan nav etiketi içindeki etiketler etkilenecektir. Öncelikle nav etiketi biçimlendirilmiş ve genişliği 900px, sayfada ortalanmış, arka plan rengi lightgrey olan bir kutu oluşturulmuştur. Daha sonra ul etiket biçimlendirilmiştir. Ul etiketinin list-style-type özelliği değiştirilerek varsayılan olarak madde işaretleri gösterme davranışı değiştirilmiş ve margin ve padding değerleri sıfırlanmıştır. Li etiketinin display özelliği inline olarak değiştirilerek yatay menü oluşması sağlanmıştır. Ayrıca float özelliğine left değeri verilerek li etiketlerinin sayfada yan yana gelmesi sağlanmıştır.

```

1 <style>
2   nav.anamenu {
3     max-width: 900px;
4     background-color: lightgrey;
5     border: thin solid;
6     overflow: auto;
7     float: left;
8   }
9   nav.anamenu ul {
10    list-style-type: none;
11    margin: 0;
12    padding: 0;
13  }
14  nav.anamenu li {
15    display: inline;
16  }
17  nav.anamenu a {
18    display: block;
19    width: 120px;
20    text-align: center;
21    padding: 6px;
22    text-decoration: none;
23    color: black;
24  }
25  nav.anamenu a.etkin {
26    background-color: darkgrey;
27    color: white;
28  }
29  nav.anamenu a:hover:not(.etkin) {
30    color: white;
31    background-color: black;
32  }
33 </style>

```

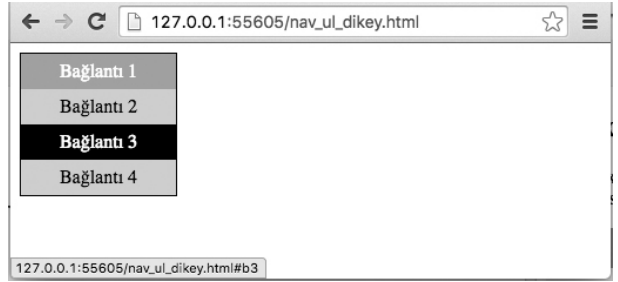
Şekil 6.43 Dikey menü için örnek css kodları

A etiketinin display özelliği block yapılmış ve bu sayede padding değeri atanabilmektedir. Ayrıca a etiketine metin ve renk ile ilgili özellikler verilmiştir. Class'ı etkin olan a etiketinin renkleri değiştirilmiştir. Genellikle menülerde kullanıcının o an bulunduğu web sayfasını gösteren bağlantı, kullanıcıyı nerede olduğunu anlayabilmesi için farklı şekilde renklendirilir. a:hover:not(.etkin) seçicisi ile class'ı etkin olmayan bağlantıların fare ile üzerine gelindiğinde farklı renklerde görünmesi sağlanmıştır. Örnekte Bağlantı 3'ün üzerine fare ile gelinmiştir. Benzer şekilde bu menü sadece css kodları kullanılarak dikey hâle dönüştürülebilir. Dikey menü oluşturmak için gerekli CC kodları aşağıda verilmiştir.

Şekil 6.39'da verilen örneğe Şekli 6.43'deki CSS kodları uygulandığında Şekil 6.44'deki görüntü elde edilecektir.

Kodlar incelendiğinde yatay menü kodları ile hemen hemen aynı olduğu görülecektir. Sadece iki fark mevcuttur. Nav etiketi sayfayı ortalayıp 900px'lik bir yer kaplamamakta ve float:left; ile sayfada sol hizalı şekilde yüzen bir blok olarak durmaktadır. Ayrıca li etiketinden float:left özelliği kaldırılmıştır. Böylece bu küçük değişikliklerle menü yataydan dikeye dönüştürülmüştür.

Görüldüğü üzere HTML kodları sırasız liste şeklinde yazılan menüler CSS ile rahatlıkla biçimlendirilebilir. CSS ile farklı davranış gösteren etkileyici menüler tasarlamak mümkündür. Örneğin yukarıdaki yatay menünün bulunduğu nav etiketi için position: fixed; özelliği eklenerek her zaman sayfanın en üstüne kalması ve kaydırma çubuğundan etkilenmemesi sağlanabilir. Ayrıca sadece CSS kullanarak fare üzerine geldiğinde açılır menü yapmakta mümkündür. Bununla ilgili kod örneğine Şekil 6.45'de yer verilmiştir.



Şekil 6.44 Dikey menü ekran çıktısı

```

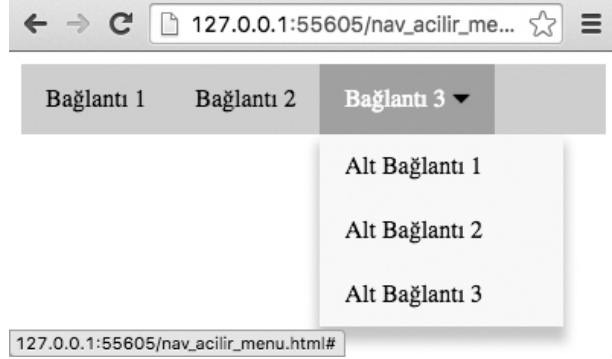
1 <style>
2   nav.anamenu ul {
3     list-style-type: none;
4     margin: 0;
5     padding: 0;
6     overflow: hidden;
7     background-color: lightgray;
8   }
9   nav.anamenu li { float: left; }
10  nav.anamenu li a {
11    display: inline-block;
12    color: black;
13    text-align: center;
14    padding: 14px 16px;
15    text-decoration: none;
16  }
17  nav.anamenu li a:hover {
18    background-color: darkgray;
19    color: white;
20  }
21  nav.anamenu li.acilirmenu { display: inline-block; }
22  nav.anamenu div.altmenu {
23    display: none;
24    position: absolute;
25    background-color: #f9f9f9;
26    min-width: 160px;
27    box-shadow: 0px 8px 10px 2px rgba(0,0,0,0.2);
28  }
29  nav.anamenu div.altmenu a {
30    color: black;
31    padding: 12px 16px;
32    text-decoration: none;
33    display: block;
34    text-align: left;
35  }
36  nav.anamenu div.altmenu a:hover { background-color: #f1f1f1 }
37  nav.anamenu li.acilirmenu:hover div.altmenu { display: block; }
38 </style>
39 <nav class="anamenu">
40   <ul>
41     <li><a href="#">Bağlantı 1</a></li>
42     <li><a href="#">Bağlantı 2</a></li>
43     <li class="acilirmenu">
44       <a href="#">Bağlantı 3
45         
46       </a>
47       <div class="altmenu">
48         <a href="#">Alt Bağlantı 1</a>
49         <a href="#">Alt Bağlantı 2</a>
50         <a href="#">Alt Bağlantı 3</a>
51       </div>
52     </li>
53   </ul>
54 </nav>

```

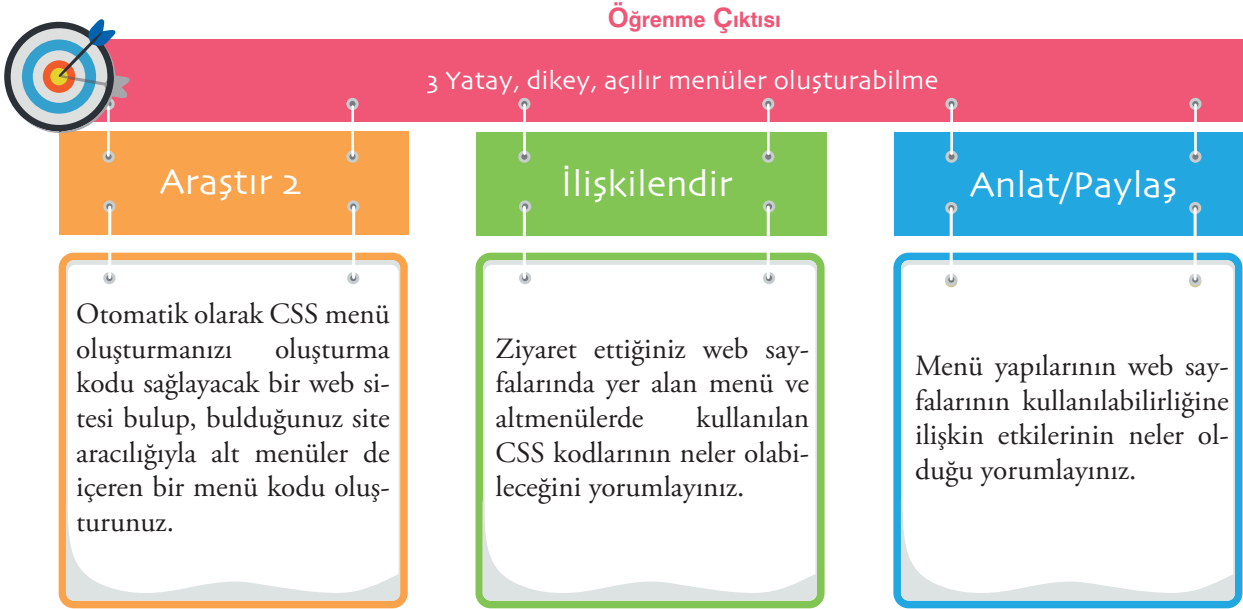
Şekil 6.45 Açılır menü kod örneği

Şekil 6.45'deki HTML ve CSS kodları uygulandığında Şekil 6.46'daki görüntü elde edilecektir.

HTML etiketleri incelendiğinde öncekinden farklı olarak açılır liste olacak li etiketi için bir class tanımlaması yapılmış ve bu li etiketinin içine ise class'ı altmenu olan div bloğu yerleştirilmiştir. CSS kodları incelendiğinde ise öncelikle nav etiketi için overflow: hidden; yapılmıştır. Böylece taşan içerik bir başka deyişle alt menü için nav etiketinin yüksekliği arttırılmayacaktır. Class'ı altmenu olan div için display: none; yapılmıştır. Böylece sayfa açıldığında alt menü sayfada yer kaplamayacak ve gizli olacaktır. nav.anamenu li.acilirmenu: hover div.altmenu seçicisi sayesinde class'ı acilirmenu olan li'nin üzerine fare ile gelindiğinde class'ı altmenu olan div için display: block; ifadesi çalışacaktır. Bir başka deyişle fare ile Bağlantı 3'ün üzerine gelindiğinde alt menü görünür hâle gelecektir.

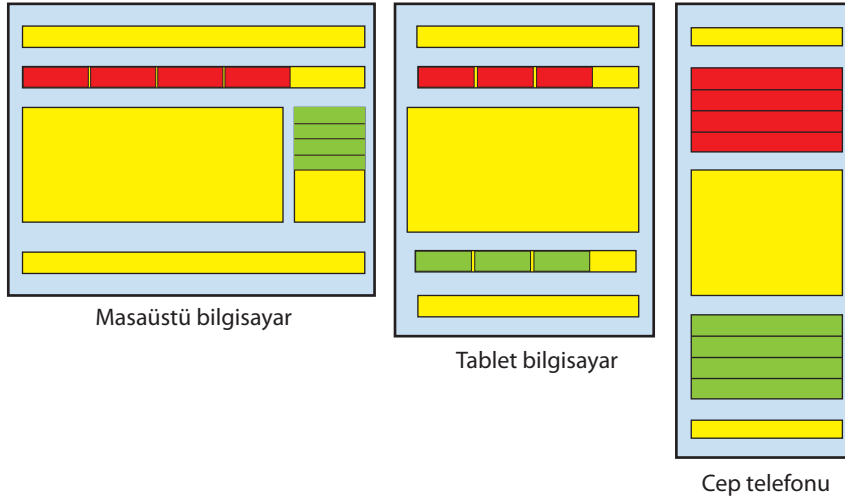


Şekil 6.46 Açılır menü ekran çıktısı



ESNEK BÖLÜM TASARIMI

Bilindiği üzere son zamanlarda mobil cihazların kullanımı oldukça artmıştır. Özellikle cep telefonlarından internete girenlerin sayısında önemli bir artış yaşanmıştır. Hatta dünya genelinde İnternet erişimi ile ilgili istatistikler incelendiğinde, mobil cihazların masaüstü bilgisayarlardan daha fazla kullanıldığı görülmektedir (Chaffey, 2016). Dolayısıyla cep telefonundan İnternet'e ulaşan kullanıcılar kesinlikle göz ardı edilemez. Web sayfası geliştiricileri için bu noktada en önemli unsur ekran büyüklüğüdür. Cep telefonlarının ekranları normal olarak masaüstü/dizüstü bilgisayar monitörlerinden çok daha küçüktür ve genellikle dikey olarak kullanılmaktadırlar. Eğer web sayfasında mobil cihazlara yönelik bir tasarım yapılmaz ise, kullanıcı cep telefonu ekranından web sayfasını gezinmekte zorluk yaşayacaktır. Bu bakımdan geliştirilecek web sayfalarının cep telefonları ile uyumlu olması, cep telefonu kullanıcılarının web sayfalarında rahatlıkla gezinebilmelerini sağlamak büyük önem arz etmektedir. Bunun için web sayfası tasarımı yaparken farklı ekran büyüklüklerinde nasıl görünmesi gerektiğini de tasarlamak gerekir. Farklı ekran büyüklüklerine otomatik olarak uyum sağlayabilen tasarımlara, esnek (responsive) tasarım adı verilmektedir. Şekil 6.47'te farklı ekran büyüklükleri için nasıl farklı tasarımlar yapılabileceği ile ilgili bir örneğe yer verilmiştir.



Şekil 6.47 Farklı ekran büyüklükleri için esnek tasarımlar

Şekilde görülebileceği üzere masaüstü/dizüstü bilgisayar monitörleri yeterince geniş olduğu için rahatlıkla bir yatay menü konulabilir ve içeriğin yanında bir yan menü kullanılabilir. Tablet kullanan kullanıcıların ekranları masa üstü bilgisayarların monitörlerinden daha küçük olmasına rağmen yatay menü kullanılabilir. Ancak yan menüyü içeriğin altına almak uygun bir çözüm olarak görülebilir. Cep telefonu ekranları ise daha da küçük olduğu için menüler yatay değil dikey olarak yerleştirilebilir, Çünkü yatay menü o ekran genişliğine sığmayacaktır. Benzer şekilde yan menü de içerikten sonra dikey olarak verilebilir.

✓ Mobil cihazlar için açılır menü uygulamaları da yapılabilir. Ancak bunun için JavaScript dilinden yararlanmak gerekmektedir.

Viewport

Viewport, web tarayıcısının web sayfalarını gösterdiği alana verilen isimdir. Bu alan masaüstü/dizüstü bilgisayarlarda daha büyük iken cep telefonlarında doğal olarak daha küçüktür. Tablet ve akıllı cep telefonu kullanımı bu kadar yaygınlaşmadan önce web sayfaları sadece masaüstü/dizüstü bilgisayar monitörlerinde kullanılmaya yönelik olarak geliştirilmekteydi. Bu bakımdan genellikle web sayfaları için sabit genişlikli bir tasarım yapılırdı. Mobil cihazlar göz önünde bulundurulmadan geliştirilen web sayfaları aksi belirtilmez ise mobil cihazlar tarafından küçültülerek sunulmaktadır. Bir başka deyişle, sayfa genişliği ekranın genişliğine sığacak şekilde sayfada küçültme işlemi uygulanır. Bu durumda sayfa okunamayacak kadar çok küçüleceği için kullanıcı sayfayı inceleyebilmek için sayfayı iki parmağıyla büyütüp yatay ve dikey olarak sayfayı kaydırması gerekecektir. Sayfayı incelemek için büyütme ve yatay olarak kaydırmak zorunda olmak kullanılabilirlik açısından iyi değildir. Bu bakımdan mobil cihazlar ile uyumlu olarak geliştirilen web sayfalarında, tarayıcıya web sayfasının mobil cihazlar ile uyumlu olduğuna yönelik bilgi vermekte gerekmektedir. Bunun için de meta etiketinden yararlanılır (Şekil 6.48).

```
1 <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
```

Şekil 6.48 Tarayıcıyı viewport konusunda bilgilendirme

width=device-width ifadesi ile sayfanın genişliğinin kullanılan cihazın ekran genişliği ile aynı olacağı belirtilmiş olur. initial-scale=1.0 ifade ile ise yakınlaştırma oranı 1 olarak ayarlanmıştır. Bir başka deyişle büyütme ya da küçültme işleminin yapılmayacağı belirtilmiştir olur.

Ortam Sorgusu

Ortam sorgusu (media query) CSS'nin 3. sürümü ile gelen yeniliklerden biridir. Ortam sorgularının en önemli amacı tarayıcı penceresinin genişliği ile ilgili bilgi almaktır. Bunun dışında tarayıcı penceresinin yüksekliği, yerleşimi (yatay veya dikey), çözünürlüğü gibi bilgileri sorgulamakta mümkündür. Ortam sorgusunun yazım şekli aşağıdaki gibidir.

```
1  @media [not|only] [ortam türü and] (ortam özelliği) {
2      CSS kodları;
3  }
```

Şekil 6.49 Ortam sorgusu yazım şekli

Ortam sorgularının yazım şekli incelendiğinde öncelikle isteğe bağlı olarak not veya only kelimelerinin kullanılabileceği görülmektedir (Şekil 6.49). Not kelimesi olumsuzluk anlamına gelir ve mantıksal olarak düşünüldüğünde doğruyu yanlış, yanlış ise doğruya dönüştürür. Bu durumda eğer not kelimesinden sonra gelen ifade yanlış sonucunu döndürür ise not kelimesi genel sonucu doğruya dönüştüreceği için küme parantezi içindeki CSS kodları çalıştırılacaktır. Only ifadesinin kullanım amacı ise küme parantezi içindeki CSS kodlarını eski tarayıcılardan gizlemektir. Bir başka ifadeyle eski tarayıcılar only kelimesini yorumlayamadıkları için küme parantezi içinde kalan CSS kodlarını göz ardı edeceklerdir. Ortam sorgusunda isteğe bağlı olarak ortam türü belirtilebilir. Eğer ortam türü belirtilmez ise yazılan sorgu tüm ortam türleri için geçerli olacaktır. Kullanılabilecek ortam türlerinden bazılarının listesi aşağıda verilmiştir:

- all: Tüm ortamlar
- screen: Bilgisayar, tablet, cep telefonu gibi cihazların ekranları
- print: Yazıcı çıktısı
- speech: Ekrandaki metni sesli şekilde okuyan ekran okuyucuları

Ortam türü kararlaştırıldıktan sonra parantez içinde ortam özelliği belirtilmelidir. Ortam özelliği olarak min-width, max-width, min-height, max-height, min-resolution, max-resolution, min-device-width, max-device-width, orientation gibi çeşitli özelliklerin değerleri belirlenebilir. Şekil 6.50'de ortam sorguları ile ilgili örnek bir kullanıma yer verilmiştir.

```
1  nav {
2      width:30%;
3  }
4  @media only screen and (max-width: 500px) {
5      nav {
6          width:100%;
7      }
8  }
```

Şekil 6.50 Ortam sorgusu örneği

Bu ortam sorgusunda bilgisayar, tablet, cep telefonu gibi cihazların ekranlarının kullanılması durumunda (screen) ve tarayıcı penceresinin genişliğinin en fazla 500px olması durumunda nav etiketinin genişliğinin 100% olacağı belirtilmiştir. Only ifadesi sayesinde güncel olmayan tarayıcılarda küme parantezi içindeki kodlar dikkate alınmayacaktır. Örnekte de görüldüğü üzere öncelikle nav etiketi için varsayılan bir değer belirtilmiştir. Daha sonra bir ortam sorgusu oluşturulmuştur. Eğer ortam sorgusu doğru sonucunu üretirse nav etiketi için varsayılan değer üzerine yazılmış olacaktır. Bir başka deyişle tarayıcının ekran genişliği 500px veya daha küçük olan ortamlarda nav etiketinin genişliği 100% değerini alırken, 500px'den büyük olması durumunda ise 30% değerini alacaktır.

```

1  @media (max-width: 600px) {
2    #reklam {
3      display: none;
4    }
5  }

```

Şekil 6.51 Ortam sorgusu örneği

Şekil 6.51'deki ortam sorgusu incelendiğinde, ortam türü belirtilmediği için tüm ortamlarda eğer ortamın genişliği en fazla 600px ise reklam id'li eleman sayfadan kaldırılmaktadır.

```

1  @media screen and (min-width: 700px) and (orientation:
2    landscape){
3    #reklam {
4      position: fixed;
5      top:0px;
6      right:0px;
7    }

```

Şekil 6.52 Ortam sorgusu örneği

Şekil 6.52'deki ortam sorgusu incelendiğinde iki ayrı ortam özelliği belirtildiği ve and (ve) mantıksal bağlacı ile bağlandığı görülmektedir. Bu şekilde istendiği kadar ortam özelliği and bağlacı ile birleştirilebilir. And bağlacının kullanıldığı durumlarda her iki koşulunda doğru olması durumunda sonuç doğru olarak üretilcektir. Örnek düşünüldüğünde ortamın genişliği en az 700px ise ve ortamın yerleşimi landscape (yatay) ise id'si reklam olan eleman sayfasının sağ üst bölümüne sabitlenmektedir. İfadeden de anlaşılacağı üzere her iki koşulunda sağlanmış olması gerekir.

```

1  @media all and (max-width: 1000px) and (min-width:
2    700px) {
3    #iletisim ul li a:before {
4      content: "E-posta: ";
5    }

```

Şekil 6.53 Ortam sorgusu örneği

Şekil 6.53'deki and kullanımına yönelik bir örnek daha verilmiştir. Bu örnek incelendiğinde, tüm ortamlar için eğer ortam genişliği en az 700px ve en fazla 1000px ise, bir başka deyişle ortam genişliği 700px ile 1000px arasında bir değere sahip ise iletişim id'li elemanın içindeki bağlantı listesindeki bağlantıların başında "E-posta:" ifadesi konulmaktadır.

```

1  @media screen and (min-width: 700px), screen and
2    (orientation: landscape){
3    #reklam {
4      position: fixed;
5      top:0px;
6      right:0px;
7    }

```

Şekil 6.54 Ortam sorgusu örneği

Şekil 6.54'deki ortam sorgusu incelendiğinde iki ayrı sorgunun virgül ile birleştirildiği görülmektedir. Virgül, or (veya) mantıksal bağlacı olarak düşünülebilir. Başka bir ifadeyle virgül ile ayrılmış sorgulardan her hangi birinin doğru olması durumunda küme parantezi içindeki CSS kodları çalıştırılacaktır. Örnek düşünüldüğünde ekranın genişliği en az 700px ise veya ekranın yerleşimi landscape (yatay) ise id'si reklam olan eleman sayfasının sağ üst bölümüne sabitlenmektedir. İfadeden de anlaşılacağı üzere her iki koşuldan bir tanesinin sağlanmış olması yeterlidir.


```

1  @media not screen and (min-width: 700px) and
2  (orientation: landscape){
3      #reklam {
4          display: none;
5      }
6  }

```

Şekil 6.55 Ortam sorgusu örneği

Şekil 6.55'deki ortam sorgusunda not kelimesi kullanılmıştır. Not kelimesi daha önce belirtildiği üzere mantıksal olarak doğruyu yanlışla, yanlış ise doğruya dönüştürmektedir. Ayrıca not, sadece screen kelimesini değil tüm sorguyu etkiler. Örnek incelendiğinde not kelimesi düşünülmez ise genişliğin en az 700px olduğu ve yerleşimin yatay olduğu durumlarda sorgu ifadesi doğru sonucunu, diğer tüm durumlarda ise yanlış sonucunu üretir. Not kelimesi ise durumu tam tersine çevirmiş olur. Bu durumda örneği not kelimesi ile birlikte düşündüğümüzde, genişliği en az 700px olmayan veya yerleşimi yatay olmayan durumlarda reklam id'li elaman sayfadan kaldırılır.

Şekil 6.47'de verilen tasarıma yönelik yazılabilecek örnek kodlar Şekil 6.56 ve Şekil 6.57'de verilmiştir.

```

1  <style>
2      header, footer{
3          max-width: 1000px;
4          background-color: lightgrey;
5          margin: auto;
6          text-align: center;
7          padding: 8px 0px 8px 0px;
8      }
9      nav.anamenu {
10         max-width: 1000px;
11         background-color: darkgrey;
12         overflow: auto;
13         margin: auto;
14     }
15     nav.anamenu ul {
16         list-style-type: none;
17         margin: 0;
18         padding: 0;
19     }
20     nav.anamenu li {
21         display: inline;
22         float: left;
23         width: 100%;
24     }
25     nav.anamenu a {
26         display: block;
27         width: 100%;
28         text-align: center;
29         padding: 6px 0px 6px 0px;
30         text-decoration: none;
31         color: black;
32     }
33     nav.anamenu a:hover {
34         color: white;
35         background-color: black;
36     }
37     nav.yanmenu {
38         max-width: 1000px;
39         background-color: darkgrey;
40         overflow: auto;
41         margin: auto;
42     }
43     nav.yanmenu ul {
44         list-style-type: none;
45         margin: 0;
46         padding: 0;
47     }
48     nav.yanmenu li {
49         display: inline;
50         float: left;
51         width: 100%;
52     }
53     nav.yanmenu a {
54         display: block;
55         width: 100%;
56         text-align: center;
57         padding: 6px 0px 6px 0px;
58         text-decoration: none;
59         color: black;
60     }
61     nav.yanmenu a:hover {
62         color: white;
63         background-color: black;
64     }
65     main{
66         max-width: 1000px;
67         margin: auto;
68     }
69
70     @media screen and (min-width: 600px){
71         nav.anamenu li { width: 120px; }
72     }
73     @media screen and (min-width: 600px) and
74     (max-width:724px){
75         nav.yanmenu li { width: 120px; }
76     }
77     @media screen and (min-width: 725px) {
78         main{ overflow: auto; }
79         section {
80             width: 70%;
81             float: left;
82         }
83         aside {
84             width: 30%;
85             float: left;
86         }
87     }
88 </style>

```

Şekil 6.56 Esnek bölüm tasarımı örnek kodlar

```

1  <header>BAŞLIK</header>
2  <nav class="anamenu">
3  <ul>
4      <li><a href="#">Menü 1</a></li>
5      <li><a href="#">Menü 2</a></li>
6      <li><a href="#">Menü 3</a></li>
7      <li><a href="#">Menü 4</a></li>
8  </ul>
9  </nav>
10 <main role="main">
11 <section>
12     <h1>Başlık</h1>
13     <p>İçerik</p>
14 </section>
15 <aside>
16     <nav class="yanmenu">
17     <ul>
18         <li><a href="#">Alt Menü 1</a></li>
19         <li><a href="#">Alt Menü 2</a></li>
20         <li><a href="#">Alt Menü 3</a></li>
21     </ul>
22     </nav>
23 </aside>
24 </main>
25 <footer>Alt Bilgi</footer>

```

Şekil 6.57 Esnek bölüm tasarımı örnek kodların devamı

Şekil 6.56 ve Şekil 6.57’de verilen örnek kodlar incelendiğinde ortam sorguları düşünülmediğinde ilk tasarımın mobil cihazlara yönelik yapıldığı anlaşılabacaktır. Böylece kodların mobil cihazlarda daha hızlı çalışması sağlanmış olur. Daha sonra @media screen and (min-width: 600px) sorgusu ile ana menünün ekran genişliği 600px’in üzerindeki cihazlarda yatay olması sağlanmıştır. @media screen and (min-width: 600px) and (max-width: 724px) sorgusu ise tablet bilgisayarlara yöneliktir. Bu sorgu ile alt menünün tablet bilgisayarlarda yatay olması sağlanmıştır. Son olarak @media screen and (min-width: 725px) sorgusu ile ise daha büyük olan bilgisayar ekranları için section ve aside etiketlerine float:left; ve genişlik özellikleri verilerek yan yana gelmesi sağlanmıştır. Böylece yan menü sayfanın sağ tarafında dikey olarak yer alacaktır.

Grid Kullanımı

Günümüzde bir çok web sayfası tasarımında grid yapısı kullanılmaktadır. Hatta bir çok CSS kütüphanesi grid kullanımını destekleyecek araçlar sunmaktadırlar. Grid kullanımı ile genel olarak sayfa genişliği 12 eşit parçaya bölünür ve tarayıcı penceresinin tüm genişliği kullanılır. Grid kullanımı, tasarım öğelerini sayfa içinde yerleştirmeyi kolaylaştırır. 12 parçanın toplam genişliği yüzde yüz olacak şekilde ayarlanmaktadır. Bu durumda her bir parça yüzde 8,33’lük bir genişliğe sahip olacaktır. Aşağıda grid yapısı kullanılarak geliştirilmiş bir web sayfası için örnek kodlar yer almaktadır.

```

1 <style>
2 * { box-sizing: border-box; }
3 [class*="col-"] {
4     float: left;
5     padding: 15px;
6 }
7 .col-1 {width: 8.33%;}
8 .col-2 {width: 16.66%;}
9 .col-3 {width: 25%;}
10 .col-4 {width: 33.33%;}
11 .col-5 {width: 41.66%;}
12 .col-6 {width: 50%;}
13 .col-7 {width: 58.33%;}
14 .col-8 {width: 66.66%;}
15 .col-9 {width: 75%;}
16 .col-10 {width: 83.33%;}
17 .col-11 {width: 91.66%;}
18 .col-12 {width: 100%;}
19 .satir{ overflow: auto; }
20 header {
21     background-color: lightgray;
22     padding: 15px;
23 }
24 nav.menu ul {
25     list-style-type: none;
26     margin: 0;
27     padding: 0;
28 }
29 nav.menu li {
30     padding: 7px;
31     margin-bottom: 10px;
32     background-color :lightgray;
33     color: black;
34     box-shadow: 0 1px 3px rgba(0,0,0,0.12) , 0 1px 2px
35         rgba(0,0,0,0.24);
36 }
37 nav.menu li:hover {
38     background-color: black;
39     color: white;
40 }
41 </style>
42
43 <header>
44     <h1>Site Başlığı</h1>
45 </header>
46 <div class="satir">
47     <nav class="col-3 menu">
48         <ul>
49             <li>Bağlantı 1</li>
50             <li>Bağlantı 2</li>
51             <li>Bağlantı 3</li>
52             <li>Bağlantı 4</li>
53         </ul>
54     </nav>
55     <section class="col-9">
56         <h1>Başlık</h1>
57         <p>İçerik</p>
58     </section>
59 </div>

```

Şekil 6.58 Grid kullanımı örnek kodlar

Şekil 6. 58’de verilen kodların çıktısı Şekilde 6.59’da verilmiştir.



Şekil 6.59 Grid kullanım örneği

Kodlar incelendiğinde öncelikle tüm etiketler için `box-sizing: border-box;` özelliği belirlenmiştir. Varsayılan olarak bir elemana genişlik özelliği tanımlandığında elemanın içerik bölümü için tanımlanmış olur, padding ile border-width değerleri bu genişliğin dışında tutulur. Dolayısıyla eğer padding ve border-width değerleri sıfırdan farklı ise elemanın sayfada kaplayacağı genişlik tanımlanan genişlikten daha fazla olacaktır.

Bu durumda da yan yana gelen elemanların toplam genişlikleri yüzde yüz olsa bile padding ve border-width değerleri dolayısıyla yüzde yüzden daha fazla yer kaplayacak ve alt satıra geçerek istenmedik bir duruma yol açacaktır. Bu durumda box-sizing özelliğine border-box değeri verilerek elemana atanan genişliğin sadece içerik değil padding ve border-width bölümlerini de kapsadığı sağlanır. Böylece yan yana gelen elemanların toplam genişliği yüzde yüz olduğunda tarayıcı sayfasının genişliğine tam olarak yerleşebilmiş olacaktır. Daha sonra col ile başlayan 12 sütun tanımlaması yapılmıştır. Tüm sütunları için ayrıca float: left; özelliği tanımlanmıştır. Sayfa tasarımcısı bu sütun tanımlarını kullanarak sayfa elemanlarını yan yana yerleştirebilir. Burada dikkat edilmesi için gereken nokta yerleştirilen elemanların toplam genişliğinin yüzde yüz olması gerektiğidir. Örnekte de görüldüğü üzere menüyü yerleştirmek için col-3 (25%), içeriği yerleştirmek için col-9 (75%) kullanılmıştır. Tarayıcı penceresinin genişliği değiştirildiğinde sayfa elemanlarının genişliklerinin otomatik olarak ayarlandığı gözlenecektir. Yukarıdaki örnekte mobil cihazlar ile de kullanılabilir esnek bir tasarım yapılmamıştır. Ancak bu bölümde öğrenilen bilgiler ile bir önceki bölümde öğrendiğiniz bilgileri birleştirerek mobil cihazlara yönelik rahatlıkla esnek bir grid tasarımı yapmak mümkündür.

Resimler

Esnek tasarım yaparken dikkat edilmesi gereken bir başka konu ise resimlerdir. Eğer resimlerin genişlik veya yükseklik özelliklerine değer atanmaz ise resimler web sayfasında gerçek boyutlarında görüntülenirler. Eğer resmin genişliği tarayıcı penceresinin genişliğinden daha fazla ise bu durum resmin taşması ve yatay kaydırma çubuğunun çıkmasına neden olacaktır. Daha önce de belirtildiği üzere web sayfalarında yatay kaydırma çubuğunun olması istenilen bir durum değildir. Bu durumda resmin boyutlarının tarayıcı penceresinin genişliğine veya bulunduğu bloğun genişliğine uygun olması gerekmektedir. Bunu ise Şekil 6.60'daki kodla ile sağlamak mümkündür.

```
1  {
2     max-width: 100%;
3     height: auto;
4 }
```

Şekil 6.60 Resimler için esnek tasarım

Yukarıdaki kod ile sayfa içindeki tüm resimler içinde max-width: 100%; özelliği tanımlanmıştır. Böylece resimlerin genişliğinin en fazla bulunduğu bloğun genişliği kadar olması sağlanmış olur. Bu sayede resimler taşma yapmayacaktır. Ayrıca max-width özelliği kullanıldığında eğer resim bulunduğu ortam resmin gerçek genişliğinden fazla olursa, resmin genişliği arttırılmaz, resim gerçek boyutlarında görüntülenecektir. Eğer resmin genişliğinin bulunduğu ortamın genişliği ile her zaman aynı olması istenir ise o zaman width: 100%; özelliği tercih edilmelidir. Bunun yanında height: auto; özelliği kullanılarak resmin yüksekliğinin her zaman genişliğine bağlı olarak orantılanması sağlanmış olur.

Öğrenme Çıktısı



4 Ortam sorgusu kullanarak esnek bölüm tasarımı yapabilme

Araştır 3

Cihazların sahip oldukları ekran özelliklerine göre biçim değiştiren hazır bir web şablonu edinin deneyiniz.

İlişkilendir

Esnek bölüm tasarımına sahip web sitelerinin, tarayıcı penceresinin büyüklüğüne göre nasıl şekil aldığına dikkat ediniz.

Anlat/Paylaş

Bir web sitesinin esnek bölüm tasarımına sahip olmasının sağlayacağı erişim kolaylıklarını yorumlayınız.

KANVASLARIN KULLANIMI

Canvas etiketi HTML5 ile birlikte gelmiştir ve bu etiketin HTML5 için en önemli yeniliklerden biri olduğu söylenebilir. Canvas etiketi kullanılarak web sayfası içinde boş bir tuval oluşturulmuş olur. Bu tuval üzerinde etkileşimli canlandırmalar, oyunlar, 3 boyutlu çalışmalar yapmak mümkündür. HTML5 ile gelen bu çok önemli özellik sayesinde Adobe Flash programı ile web sayfalarında kullanılmak üzere geliştirilen bir çok içerik ve oyunlar yerini HTML5 Canvas içeriklerine ve oyunlarına bırakmıştır. Adobe Flash ile geliştirilen içeriklerin çalışabilmesi içinde tarayıcıda Adobe Flash Player eklentisinin kurulu olması gerekmektedir. Bu tür eklentiler, güvenlik açığı ve yavaşlama gibi bazı sorunların beraberinde getirebilmektedir. Bu bakımda bir süredir mobil cihazlarda kullanılan tarayıcılar Adobe Flash Player eklentisini desteklememektedirler. Bu durum Canvas kullanımının önemini daha da arttırmıştır. Çünkü hiçbir tasarımcı mobil kullanıcıları göz ardı edemez. Canvas etiketinin kullanımı Şekil 6.61’de verilmiştir.

```
1 <canvas id="tuval" width="200" height="200"></canvas>
```

Şekil 6.61 Canvas etiketinin kullanımı

Canvas etiketi ile oluşturulan tuvalin üzerine JavaScript programlama dili kullanılarak çizim yapılabilir, etkileşimli içerikler oluşturulabilir. Tabi bunu yapabilmek için JavaScript programlama diline hakim olmak gerekecektir. Bu bölümde örnek olması açısından birkaç sabit çizim örneğine yer verilmiştir. Şekil 6.62’de JavaScript kodu ile birkaç geometrik çizim örneği oluşturulmuştur.

```
1 <script>
2   var tuval = document.getElementById("tuval");
3   var ortam = tuval.getContext("2d");
4   //Düz Çizgi
5   ortam.moveTo(0,0);
6   ortam.lineTo(100,100);
7   ortam.stroke();
8   //Dikdörtgen
9   ortam.fillStyle = "#FF0000";
10  ortam.fillRect(100,100,50,50);
11  //Daire
12  ortam.beginPath();
13  ortam.arc(100,50,48,0,2*Math.PI);
14  ortam.stroke();
15  //Yazı
16  ortam.font = "24px Verdana";
17  ortam.fillText("Merhaba Dünya",10,50);
18  ortam.font = "24px Courier";
19  ortam.strokeText("Merhaba Dünya",10,180);
20  //Gradyan
21  var grd = ortam.createLinearGradient(10,100,90,150);
22  grd.addColorStop(0,"red");
23  grd.addColorStop(0.5,"blue");
24  grd.addColorStop(1,"yellow");
25  ortam.fillStyle = grd;
26  ortam.fillRect(10,100,80,50);
27 </script>
```

Şekil 6.62 Canvas üzerine javascript ile çizim yapma



yaşamla ilişkilendir

HTML 5 Platformundaki Oyunlar 100 Milyar Dolarlık Bir Pazar Vaat Ediyorlar

İki boyutlu vektörel grafiklerin dinamik bir biçimde kullanılarak geliştirildiği HTML 5 oyun platformu son 4 yıldır Flash oyunların da gözden düşmesiyle hızlı bir çıkış yakalamıştı. Önümüzdeki bir yıl içerisinde HTML 5 oyunların mobil platformlarda bu çıkışını sürdüreceği ve oyun pazarına 100 milyar dolarlık bir potansiyel katacağı söyleniyor.

Hyper Text Markup Language tamlamasının kısaltılmışı olan web programlama dilinin 5'inci

nesil hali olarak da bilinen HTML 5, yükselişine Facebook'un Canvas alt yapısıyla başlamış ve kısa sürede Flash oyunları tahtından etmeyi başarmıştı. HTML5'in en önemli özelliği, Canvas üzerinde 2D veya 3D çizim yapılabilmesi ve video-audio desteği ile görüntü oynatabiliyor olmasıydı. Bu özellikler de oyunlarda son derece akıcı ve renkli tecrübelerin kapısını aralamıştı.

Kaynak: https://www.tamindir.com/html-5-platformundaki-oyunlar-100-milyar-dolarlik-bir-pazar-vaat-ediyorlar_h-18281/

Öğrenme Çıktısı



5 Grid yapısı kullanarak sayfa tasarımı yapabilme.
6 Kanvas üzerine çizim yapabilme.

Araştır 4

“SVG” kavramını araştırınız.

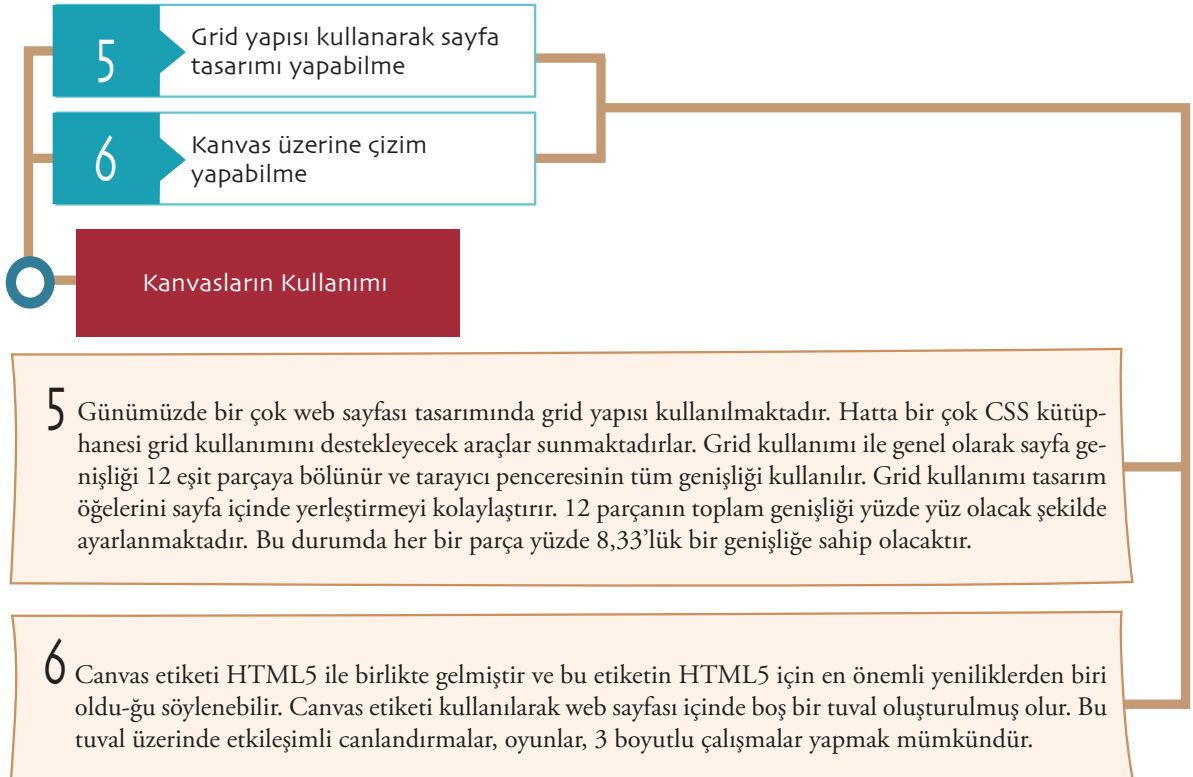
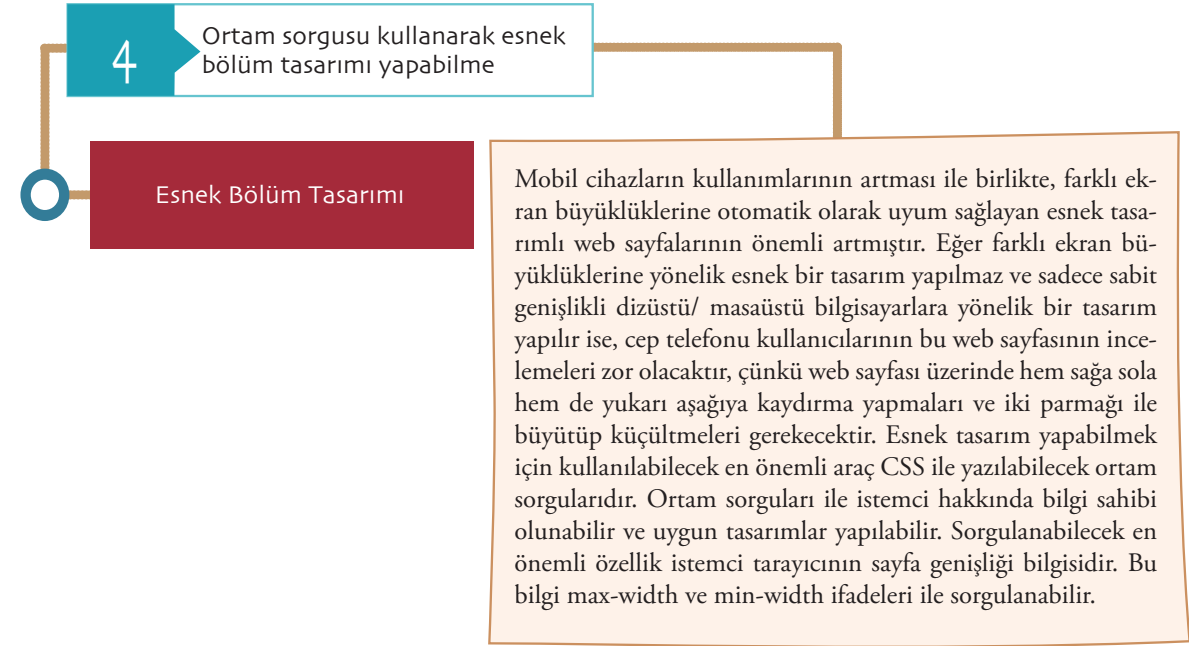
İlişkilendir

Kanvas tarafından sağlanan olanakları Flash platformunun sağladıklarıyla karşılaştırınız.

Anlat/Paylaş

Web kanvaslarını kullanarak web üzerinden oynanan oyunların oluşturulabilmesinin ne ölçüde mümkün olduğunu tartışınız.





1 Aşağıdakilerden hangisi genel amaçlı bölüm oluşturmak için kullanılan etikettir?

- A. Div B. Section
C. Aside D. Span
E. Main

2 Aşağıdakilerden hangisi temasal bölüm oluşturmak için kullanılan etikettir?

- A. Div B. Section
C. Aside D. Span
E. Main

3 Aşağıdakilerden hangisi HTML5 ile gelen anlamsal bölüm oluşturma etiketlerinden biri değildir?

- A. Section B. Aside
C. Span D. Main
E. Header

4 Aşağıdakilerden hangisi ile display özelliği block olan elemanların bulunduğu ortamın sağında ya da solunda yüzmesini sağlar?

- A. Text-align B. Margin
C. Padding D. Left
E. Float

5 Aşağıdaki özelliklerden hangisi display özelliği block olan elemanları yatay olarak ortalamak için kullanılır?

- A. Max-width
B. Text-align
C. Padding
D. Margin
E. Float

6 Bir elemanın sürekli olarak tarayıcı ekranının sağ üst köşesinde konumlanması isteniyor ise top ve right özellikleri ile birlikte hangi özelliğin ayarlanması gerekir?

- A. Overflow B. Position
C. Padding D. Margin
E. Float

7 Float:left; özelliği verilen bir elemandan sonra gelen elemanın yeni bir satırdan başlamasını sağlamak için aşağıdaki özelliklerden hangisi kullanılmalıdır?

- A. Overflow: auto;
B. Position: absolute;
C. Clear: left;
D. Margin: auto;
E. Display: inline;

8 İçinde float özelliği olan elemanların bulunduğu kapsayıcı bir etiketin yüksekliğinin doğru şekilde ayarlanabilmesi için aşağıdaki özelliklerden hangisi yazılmalıdır?

- A. Overflow: auto;
B. Position: absolute;
C. Clear: left;
D. Margin: auto;
E. Display: inline;

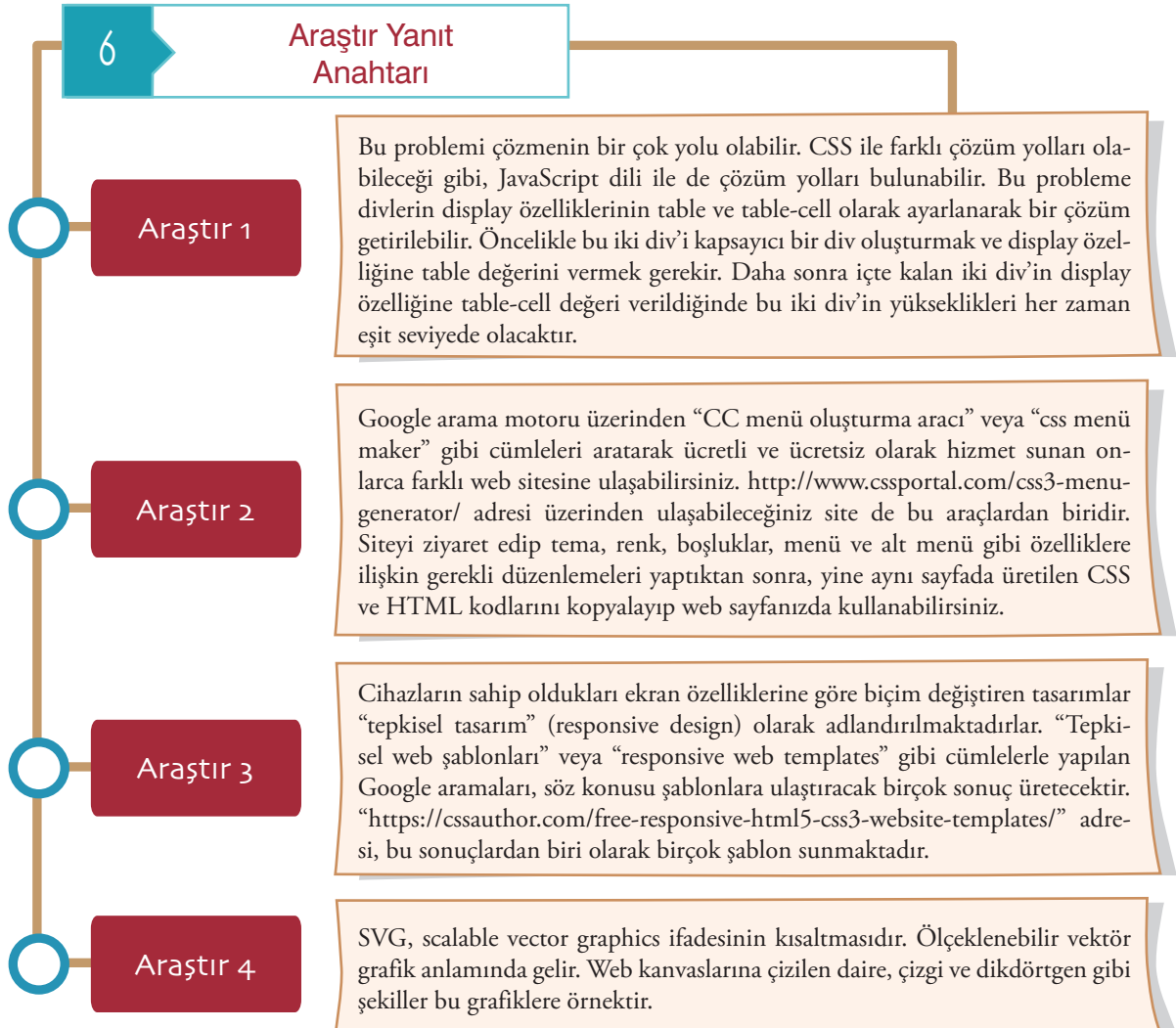
9 Web sitelerinin ana menülerinin HTML5 standartlarına göre hangi anlamsal etiket içinde yer alması gerekir?

- A. Ol B. Aside
C. Article D. Header
E. Nav

10 Aşağıdakilerden hangisi 500 px'den küçük veya 1000 px'den büyük ekran genişliğine sahip cihazları tespit etmek için kullanılan ortam sorgusudur?

- A. @media screen and (min-width: 500x and max-width: 1000px)
B. @media only screen and (max-width: 500x and min-width: 1000px)
C. @media not media screen and (max-width: 500x), screen and (min-width: 1000px)
D. @media screen and (max-width: 500x), screen and (min-width: 1000px)
E. @media screen and (max-width: 500x and min-width: 1000px)

1. A	Yanıtınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.	6. B	Yanıtınız yanlış ise “Position Özelliği” konusunu yeniden gözden geçiriniz.
2. B	Yanıtınız yanlış ise “Div Etiket” konusunu yeniden gözden geçiriniz.	7. C	Yanıtınız yanlış ise “Float Özelliği” konusunu yeniden gözden geçiriniz.
3. C	Yanıtınız yanlış ise “Div Etiket” konusunu yeniden gözden geçiriniz.	8. A	Yanıtınız yanlış ise “Overflow Özelliği” konusunu yeniden gözden geçiriniz.
4. E	Yanıtınız yanlış ise “Float Özelliği” konusunu yeniden gözden geçiriniz.	9. E	Yanıtınız yanlış ise “Div Etiket” konusunu yeniden gözden geçiriniz.
5. D	Yanıtınız yanlış ise “Margin Özelliği” konusunu yeniden gözden geçiriniz.	10. D	Yanıtınız yanlış ise “Ortam Sorgusu” konusunu yeniden gözden geçiriniz.



Kaynakça

- Bowers, M., Synodinos, D. ve Summer, V. (2011). *Pro HTML5 and CSS3 Design Patterns*. Apress: New York.
- Chaffey, D. (2016). Mobile Marketing Statistics compilation. Erişim Tarihi: 01.07.2016. <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- Çelikbilek, İ. (2014). *Her Yönüyle HTML5*. Kodlab Yayın: İstanbul.
- Erdinç, F. (2016). *Yeni Başlayanlar İçin HTML5 - CSS3*. Pusula Yayıncılık ve İletişim: İstanbul.
- Goldstein, A., Lazaris, L., ve Weyl, E. (2015). *HTML5 & CSS3 for the Real World*. Sitepoint: VIC Australia.
- Harris, A. (2014). *HTML5 and CSS3 All-in-One For Dummies, 3rd Edition*. John Wiley & Sons, Inc: Hoboken, NJ.
- Haskan, E. (2011). *HTML5 ve CSS3*. Dikeyksen Yayıncılık: İstanbul.
- Mermerkaya, A.O., (2015). *HTML5 CSS3*. Abaküs: İstanbul
- Pfaff, J. (2016). *HTML5 Development: The Beginning Beginner's Guide*. LearnToProgram, Inc: Vermon, CT.
- Tittel, E. ve Minnick, C. (2013). *Beginning HTML5 & CSS3 For Dummies*. John Wiley & Sons, Inc: Hoboken, NJ.
- Tokak, B. (2015). *HTML5 CSS3 ve JavaScript ile Web Tasarımı*. Dikeyksen Yayıncılık: İstanbul.



Bölüm 7

Önemli Web Çatıları ve Kullanımları

öğrenme çıktıları

Web Çatıları

- 1 Web çatısı kavramını açıklayabilme
- 2 Web çatılarının kullanım alanlarını tanımlayabilme
- 3 Web çatılarının yararlarını açıklayabilme

Güncel Bir Web Çatısı: Bootstrap

- 4 Bootstrap projelerinin temel bileşenlerini açıklayabilme
- 5 Bootstrap projelerinde kullanılan elementleri açıklayabilme

Anahtar Sözcükler: • Bootstrap • Izgara Sistemi • Mobil Tasarım • Ön Yüz • Tepkisel Tasarım • Web Çatıları



GİRİŞ

İnternet siteleri ve servisleri tıpkı diğer teknolojik araçlar gibi belirli üretim süreçleri gerektirir. Fikir aşamasından itibaren bir internet sitesinin (ör: firmanızın tanıtım sitesi) ya da servisinin (ör: sosyal bir ağ) hizmete alınması için geçen her dakika; gözlemlenen bir hatanın ayıklanması için geçen her saniye; güncel tasarım gereklerine uyum için hazırlanan her yeni tasarım iş gücü ve para gerektirir. Firmaların dünyaya açılan kapıları olan internet sitelerinin ya da milyonlarca kullanıcısı olan sosyal ağ sitelerinin hizmet dışı kalmaları günümüz koşullarında bir seçenek değildir. Bu çerçevede yazılım geliştiriciler üretim, düzenleme, bakım ve yenileme süreçlerini hızlandırmak için yardımcı araçlara ihtiyaç duymaktadır. Bugün pek çok ortamda duyduğumuz AngularJS, Django, Bootstrap gibi projeler bu sorunu çözmek için üretilmiş web çatılarıdır. Bu ünite, öncelikle web çatısı kavramı tanıtılacak, tarihsel gelişimleri, faydaları ve kullanım alanları incelenecektir. Ardından kullanıcı arayüzlerinin geliştirilmesi için kullanılan güncel bir web çatısı olan Bootstrap tanıtılacaktır.

WEB ÇATILARI

Web çatısı (web framework) ya da web uygulaması çatıları (web application framework) internet ortamında çalışan siteler, servisler, kaynaklar ve arayüzler üretmek için tasarlanmış yazılımlardır. Bu ünite, söz edilen uygulamaların tümü web uygulaması adı altında ele alınmaktadır. Web çatılarının temel amacı, web uygulamalarının geliştirilmesinde önemli bir iş yükü yaratan rutin işlemlerin otomatikleştirilmesidir. Örneğin, çoğu web uygulaması için önemli bir görev olan kullanıcı kaydı, yönetimi ve yetkilendirmesi için giriş formları, veri tabanı erişimi, oturum yönetimi, form verilerinin kontrolü gibi rutin işlemler gerekmektedir. Web çatıları bu işlerin tümü ya da bir bölümü için hazır kontroller ya da servisler sunmaktadır. Web çatılarının temel amacı programcılar üzerindeki rutin iş yükünü en aza indirgeyerek çevik ve dinamik web uygulamalarının geliştirilmesine destek olmaktır.

Web Çatılarının Gelişimi

WWW özünde dinamik bir yapı değildir. İnternet sayfaları hiper metinler hâlinde üretilerek sunucular üzerinden dağıtılmaktadır. Bu çer-



çevede bir internet sitesinin sıfırdan üretilmesi ve yayınlanması önemli düzeyde bilgi ve iş yükü gerektirmektedir. İnternetin erken dönemlerinde, yeni bir sayfa üretmek için yazarların/tasarımcıların tüm içerikleri el ile kodlaması, birleştirmesi ve sunucuya göndermesi gerekiyordu. Benzer şekilde, sayfaların düzenlenmesi de önemli bir iş yükü olarak ortaya çıkmaktaydı. Yayınlanmış bir sitede düzenleme yapmak için, kaynak dosyaların yazar/tasarımcı bilgisayarına indirilmesi, değişikliklerin yapılması ve bu dosyaların tekrar sunucuya gönderilmesi gerekiyordu. Bu çalışma yapısı kullanıcı etkileşimine engel olmanın yanında, internetin gelişimini de sınırlamaktaydı.

Basit bir hesap makinesi gibi, kullanıcı girdisine cevap verecek dinamik sayfaların üretilmesi için CGI (Common Gateway Interface) standardı tanıtıldı. Bu sistemde, sunucu bilgisayarda çalışan programlar kullanıcıdan gelen veriyi işleyerek statik sonuç sayfaları üretebilmekteydi. Bu yaklaşımın en büyük problemi her bir istek için sunucu bilgisayarda ayrı bir süreç (process) çalıştırılmasıydı. Basit işlemler için oluşturulmuş olsalar bile, tüm süreçlerin toplam iş yükü sunucularda önemli bir darboğaz oluşturmaktaydı. Bu darboğazdan hareketle, internet sunucuları ile entegre çalışabilen betik (script) dilleri geliştirildi. WebBase, ColdFusion, PHP ve ASP gibi işleyiciler sunucu yükü sorununa çözüm getirdi.

Bir sonraki adımda, geliştirici ve tasarımcıların iş yüklerini azaltmaya yönelik web çatıları geliştirilmeye başlandı. PHP ve ASP gibi diller veri değiş tokuşu, veri tabanı erişimi gibi işlemler için temel işlevler sağlasalar da, kullanıcı denetimi, formatlanmış çıktı oluşturma gibi işlemlerin programcı tarafından yapılması gerekmektedir. Her projede gerekli bu işlevler, tasarımcı ve geliştiricilerin iş zamanını asıl proje sorunları yerine rutin işlemlere kaydırmaktaydı. Güncel yaklaşımda bu işlemlerin gerçekleştirilmesi için üst düzey kütüphaneler kullanılmaktadır. Web çatısı olarak adlandırdığımız bu kütüphaneler arasında ASP.NET, Ruby on Rails, Django, Zend Framework, Yii, CakePHP ve Symfony sayılabilir.

Web Çatılarının Kullanım Alanları

Web çatıları, bir programlama diline dayalı olarak internet uygulamalarının geliştirmesine destek olmak için üretilmektedir. Web çatılarının kullanım alanları bir dilin yeteneklerini geliştirmekten,

uygulama yazılımları ile programlama üretmeye kadar değişkenlik gösterebilmektedir. Örnek vermek gerekirse; Rails, Ruby dilinin yeteneklerini geliştirmek için hazırlanmış bir web çatısıdır. MVC modelinde çalışan Rails, veri tabanı erişimi, web servisleri ve web sayfaları için hazır yapılar sunmaktadır.

Web çatıları, internetin geleneksel yapısındaki sunucular, protokoller ve tarayıcılar üzerinde çalışır. Bu nedenle çatılar bu işleyişteki yapısal kurallara uymak zorundadır. Sayfalar sunucular tarafından dağıtılır, tarayıcılar tarafından kullanıcıya iletirilir. Bu çerçevede, web çatılarının çalışabileceği iki nokta sunucular ve tarayıcılardır. Sunucu tarafındaki işlemler tipik olarak sayfaların yeniden yüklenmesini gerektirir. Sunucu tarafı üretimde pek çok dil ve araç kullanılabilmektedir. Bu araçlar arasında Django, “Zend Framework” ve “Ruby on Rails” sayılabilir.

✓ MVC (model-view-controller) bir yazılım geliştirme desendir. Bu desen veri soyutlanması esasına dayanır. Buna göre, kontrolcü (controller) adı verilen bir ara sistem, veri (model) ve arayüzler (view) arasında köprülük görevi görmektedir. Sistemin avantajı, yazılan kodların tekrar tekrar kullanılabilmesi ve çeşitli uygulamalar için arayüzler (views) sağlayabilmesidir. Örneğin internet sitesi kullanıcıları için görsel bir arayüz oluşturulabilirken, uzaktaki uygulamalarla veri alışverişi için farklı bir arayüz (web servisleri) üretilebilmektedir.

Tarayıcı tarafı yaklaşımında ise sayfanın tamamı yerine küçük bölümlerinde düzenlemeler ve güncelleştirmeler yapılabilmektedir. Bu durum, sayfaların masaüstü uygulamalar gibi etkileşimli çalışmasına izin vermektedir. Bu yaklaşımın dezavantajı kullanılabilecek tek dilin Javascript olması ve işlem gücünün kullanıcı bilgisayarı ve tarayıcısı ile sınırlı olmasıdır. Ağır veri trafiği ve işlem gücü gerektiren uygulamaların özellikle eskiyen mobil cihazlarda yavaş çalışması ve kilitlenmesi sıklıkla karşılaşılan bir durumdur. Bu senaryoda kullanılabilecek web çatıları arasında Backbone.js, AngularJS, EmberJS, ReactJS ve Bootstrap sayılabilir.

Web Çatılarının Yararları

Bir web çatısı kullanmanın en önemli avantajı, kodların tekrar kullanılabilir olmasıdır. Web çatıları deneyimli geliştiriciler ve geliştirici takımları tarafından tekrar kullanılabilirlik ilkesi ile üretilir. Üretilen kodlar çeşitli projelerde işe koşulur ve son kullanıcının hizmetine sunulur. Bu projelerin üretimi ve proje çıktılarının kullanımı sırasında web çatısı kodları defalarca test edilerek sorunları ortaya çıkarılır. Ortaya çıkan sorunlar geliştiriciler tarafından giderilerek web çatısının güvenilirliği ve bütünlüğü ileri seviyelere taşınır. Bu süreç sonunda, yazılım üretmek için mükemmel yakın araçlar oluşturulmuş olur. Geliştiriciler bu araçları kullanarak proje güvenliğini artırırken, yazılım geliştirme süresini kısaltabilir. Yazılım geliştirme projelerinde en önemli giderin zaman olduğu unutulmamalıdır.



Web çatısı kullanmanın bir diğer yararı ise ücretsiz yardım kanallarını kullanıma sunmasıdır. Güncel web çatıları dünya genelinde pek çok yazılımcı tarafından kullanılmakta ve geliştirilmektedir. Bir web çatısı kullanmaya başladığınızda bu çatıyı kullanan tüm geliştiricilerin ve yazılımcıların desteğini almış olursunuz. Güvenlik, tasarım, yerelleştirme ya da çoklu dil desteği gibi konular için zaman ayırmanıza gerek kalmaz. Bu tür sorunlar hâli hazırda diğer geliştiriciler tarafından çözülmüş olarak karşınıza geleceğinden zaman ve para harcamanıza gerek kalmayacaktır. Bunun yanında geliştirici forumlarında sıklıkla karşılaşılan sorunlar ya da özel durumunuz ile ilgili pek çok yardım dokümanına ulaşılabilir.

Web çatıları genellikle MVC gibi yazılım geliştirme yaklaşımlarını kullanmaktadır. Bu durum projenizin temel kabul görmüş geliştirme yaklaşımları çerçevesinde ilerlemesini sağlayacaktır. Bu sayede, proje takımına yeni katılan bir geliştiricinin uyum sağlaması daha kısa zaman alacaktır. Bunun yanında, çatıların benzer geliştirme yaklaşımlarını paylaşmaları öğrenilmelerini kolaylaştırmaktadır.

Kullanıcı girişi, veri tabanı yönetimi gibi pek çok proje için gerekli işlevleri hazır olarak sunması da web çatılarının yararlarından biridir. Bu sayede, programcıların iş gücü projenin ana problemlerine yönlendirilebilmektedir. Ayrıca web çatılarının sü-

rekli gelişim hâlinde olduğu söylenebilir. Çatıların yeni versiyonları ile birlikte güvenlik güncelleştirmelerinin yanında uygulamalarınıza güç katacak modüller gelebilmektedir. Örneğin bir e-ticaret uygulaması için kullandığınız web çatısının yeni versiyonuyla farklı bir ödeme yöntemine destek vermeye başlayabilirsiniz.

Öğrenme Çıktısı



- 1 Web çatısı kavramını açıklayabilme
- 2 Web çatılarının kullanım alanlarını tanımlayabilme
- 3 Web çatılarının yararlarını açıklayabilme

Araştır 1

Web çatılarının kullanımının beraberinde ne gibi sınırlıklar getirebileceğini araştırınız.

İlişkilendir

Web üzerinde yayınlanacak içeriği hazır olan kişi veya kurumlara ilişkin web sitelerinin yapımının birkaç saat içinde nasıl tamamlanabildiğini yorumlayınız.

Anlat/Paylaş

Kullanacağınız web çatısını seçerken hangi unsurları göz önünde bulunduracağınızı paylaşınız.

GÜNCEL BİR WEB ÇATISI: BOOTSTRAP

Bootstrap, internet siteleri ve internet tabanlı uygulamaların kullanıcı arayüzlerini (front-end) geliştirmek için kullanılabilecek bir web çatısıdır. Bootstrap, tarayıcı tarafında çalışan bir web çatısı olduğundan, temelinde HTML, CSS ve Javascript teknolojileri yer almaktadır. Bu sistemde, HTML gösterilecek içeriğin işaretlenmesini, CSS içeriğin sunumunu, Javascript ise bileşenlerin programlanmasını sağlamaktadır. Bootstrap, web siteleri ve uygulamaların kullanıcı arayüzlerini hızlı ve kolay geliştirmek için gerekli araçları sağlamaktadır. Oldukça popüler olan çatı, pek çok projede kullanılmış ve kendini kanıtlamıştır. Bu ünitenin hazırlandığı Temmuz 2016 döneminde çatının 3.3.6 versiyonu sunulmuştur.



internet

Bootstrap hakkında daha fazla bilgi edinmek için <https://getbootstrap.com/> adresini ziyaret edebilirsiniz.

Bootstrap öncelikle akıllı telefon, tablet bilgisayar gibi mobil cihazları hedef alan bir felsefe ile geliştirilmiştir. Bootstrap ile üretilen internet sayfaları ve uygulamalar mobil cihazların internet tarayıcılarında sorunsuz bir şekilde çalışmakta ve görüntülenebilmektedir. Oldukça çeşitli ekran boyutları ve piksel yoğunlukları olan mobil cihazlarda içeriklerin okunabilir ve düzgün gösterilebilmesi için metin (tipografi), yerleşim ve ölçekleme özellikleri sunulmaktadır.

✓ Programlama literatüründe kullanıcının gördüğü arayüzlere ön yüz (front-end) denmektedir. Bunun yanında, uygulamaların arkasında çalışan veri tabanı erişimi, hesaplama rutinleri gibi işlemlere ise arka yüz (back-end) adı verilmektedir. Büyük ölçekli uygulamaların ön yüz ve arka yüz geliştirici takımları işbirliği içinde çalışmaktadır.

✓ Birleşmiş Milletler'in teknoloji alanındaki araştırma kolu olan ITU'ya (2015) göre, dünya genelinde mobil internet kullanıcısının oranı (%47,2), sabit bağlantı kullanan internet kullanıcısı oranından (%46,4) daha fazladır. Bu durum, üretilen içeriklerin mobil cihazlarda düzgün görünmesini zorunlu kılmaktadır.

✓ Tepkisel tasarım (responsive design), sitelerin ve uygulamaların çeşitli ekran ve tarayıcı genişliklerine göre kendilerini yeniden düzenleyebildiği bir tasarım yaklaşımıdır. Bu yaklaşımın amacı, sitenin tasarım ve ölçeklendirmesini değiştirerek, okunurluğu desteklemek ve kullanıcı etkileşimini iyileştirmektir. Yaklaşımın temelinde CSS 3 tanımlamaları ile birlikte gelen medya sorguları (media queries) yer almaktadır.

Günümüz tarayıcı ve ekran boyutları ekosistemi oldukça dinamik ve çeşitlidir. Ekranların boyutları, en boy oranları ve tarayıcı genişlikleri cihazdan cihaza önemli farklılıklar göstermektedir. Hatta, mobil cihazların yatay ve dikey yönelimleri bile tasarımlarda ebat ve yerleşimin değişmesini gerektirmektedir. Sonuç olarak, geliştirilen sitelerin ve uygulamaların bu dinamik yapının tüm örneklerinde çalışması gerekmektedir. Bootstrap tepkisel

tasarım özellikleriyle, ürünlerin tarayıcı boyutlarına göre ölçeklendirilmesi ya da farklı yerleşimlerle sunulmasını sağlayabilmektedir.

Bootstrap'te yazı tipleri, yazı büyüklükleri, tablolar, formlar ve düğmeler tüm tarayıcılar ve ekranlar için hazır olarak gelmektedir. Bunun yanında internet siteleri ve web uygulamalarında sıklıkla kullanılan ikonlar, menüler, kayan resimler, açılır düğmeler gibi pek çok bileşen hazır olarak sunulmaktadır. Bu sayede, kullanıcının tarayıcısı, işletim sistemi ve cihazından (masaüstü, akıllı telefon, tablet) etkilenmeden, aynı görüntü ve kullanım deneyimi kolaylıkla oluşturulabilmektedir.



Bootstrap sistemiyle birlikte gelen bileşenleri incelemek için <http://trbootstrap.com/components/> adresini ziyaret ediniz.



yaşamla ilişkilendir

Ruby on Rails içinde Twitter Bootstrap kullanımını kolaylaştıran kütüphane: 'twitter-bootstrap-rails'

Seyhun Akyürek tarafından Kasım 2011'de geliştirilmeye başlanan twitter-bootstrap-rails kütüphanesi, GitHub üzerinde topladığı 4 bine yakın 'star' ve 900'ün üzerinde 'fork' rakamlarıyla dikkat çekiyor. Bunların dışında açık kaynak olarak geliştirilen kütüphane bugüne kadar 180'den fazla bağımsız geliştiricinin desteğini almayı başarmış.

Twitter tarafından geliştirilen ve oldukça popüler bir frontend (önyüz) geliştirme aracı olan Twitter Bootstrap'in Ruby on Rails üzerindeki kurulumunu kolaylaştıran kütüphane, bugüne kadar RubyGems'de 1 milyon 500 binin üzerinde indirilmiş.

RoR (Ruby on Rails) geliştiricileri için Twitter Bootstrap entegrasyonunu en kolay hale getirmeyi amaçlayan twitter-bootstrap-rails kütüphanesi içinde hazır less uyumluluğu, ikon desteği ve arayüz elementleri için helper'lar gibi çok sayıda bileşen yer alıyor.

Kaynak: <http://webrazzi.com/2015/02/24/twitter-bootstrap-rails/>

Bir Bootstrap Projesinin Anatomisi

Bootstrap projeleri, bir HTML dokümanına gerekli dosyaların bağlanması ve bu dokümanda kodların işlenmesiyle gerçekleştirilir. Temel bir Bootstrap projesinde aşağıdaki bileşenler yer almalıdır.



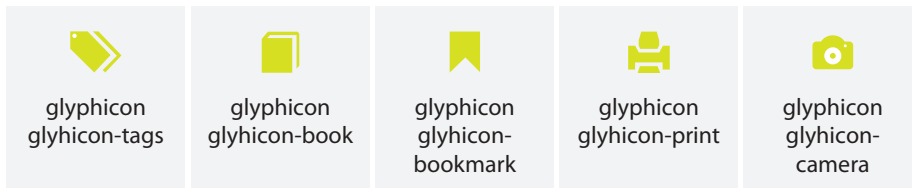
Şekil 7.1 Bir Bootstrap Projesinin Temel Bileşenleri

HTML Dokümanı: Temelinde internet teknolojileri yatan Bootstrap projeleri, HTML dokümanları üzerinden sunulur. Bu nedenle HTML dokümanı tüm içerikler için kapsayıcı rolündedir. Öncelikle, Bootstrap sisteminin çalışması için gerekli tüm dosyalar bu dokümana bağlanır. Bunun yanında, projede gösterilecek olan metinler ve tüm medya elementleri (ör: resimler, videolar) bu dosya içine eklenir ya da bağlanır. HTML dokümanı projenin kök dizininde yer alır ve giriş sayfası olarak kullanılır.

CSS Dokümanı: Bootstrap ile birlikte gelen görsel özellikler (ör: fontlar, düğmeler, ızgara sistemi) CSS dosyasında tutulmaktadır. Zorunlu hâller dışında bu dosya üzerinde değişiklik yapılmamalıdır. CSS dokümanı içindeki bir bağlantı nedeniyle bazı tarayıcılar `.map` uzantılı bu dosyayı bulamadıklarında hata mesajı verebilmektedir. Sistemin çalışırılığını etkilemeyen bu hata görmezden gelinebilir. Dosyanın asıl işlevi, tarayıcılardaki geliştirme araçlarına css dokümanının üretildiği LESS dilindeki kaynak kodlarını sağlamaktır. CSS dokümanı linki HTML dokümanının `<head>` bölümüne eklenmelidir.

Javascript Dokümanları: Bootstrap projeleri açılır menüler ve kayan resimler gibi kullanıcı etkileşimli bileşenler içerebilir. Bu bileşenlerin doğru çalıştırılabilmesi için `bootstrap.js` dosyası projeye eklenmelidir. Bunun yanında, `bootstrap.js` dosyasındaki kodların düzgün çalışabilmesi için, başka bir web çatısı olan JQuery'e ihtiyaç duyulmaktadır. Bu dosya `bootstrap.js` dosyasından önce projeye eklenmelidir. Aksi durumda sistem çalışmayacaktır. Performans kayıplarını önlemek için Javascript dokümanları projede `</body>` etiketinin önüne eklenmelidir.

Simgeler: Bootstrap ile birlikte gelen görsellerin kullanılabilmesi için `fonts` klasörü projeye dâhil edilmelidir. Simgelerin projeye eklenmesi için herhangi bir kod yazılmaz. Simgeler, `css` dosyası içinden otomatik olarak çağırılır. Bununla birlikte, sistemin düzgün çalışabilmesi için, yerel projelerde `fonts` klasörü `css` klasörü ile aynı dizinde yer almalıdır. Her projede bu simgeler kullanılmadığından bu klasörün tutulması programcının seçimine bağlıdır.



Resim 7.1 Bootstrap Sisteminde Gelen Simgelere Örnekler

Bootstrap Projesi Başlatmak

Bootstrap projeleri için gerekli dokümanlar projeye iki yöntemle eklenebilir. Bu yöntemlerin kendilerine özgü avantaj ve dezavantajları bulunmaktadır. Kullanılabilecek ilk yöntem, içerik dağıtım ağları (Content Delivery Network) üzerinden sunulan **uzak** dosyaları projeye eklemektir. İkinci yöntemde ise, gerekli dosyalar sunuculardan indirilerek **yerel** dosyalar hâlinde projeye eklenir. Aşağıda bu yöntemlerin avantajları, sınırlılıkları ve gerçekleştirilme yolları sunulmuştur.

Bootstrap projeleri uzak yöntem ile başlatılırken, uzak sunucularda barındırılan dosyalar kullanılır. Bu dosyalar içerik dağıtım ağları üzerinde ücretsiz olarak sunulmaktadır. Bu yöntem için gerekli dosyalar ve erişim adresleri Tablo 7.1'de verilmiştir.

✓ Bootstrap projelerinde kullanılan dosyalar uzak ve yerel olmak üzere iki yöntemle projeye eklenebilir.

Tablo 7.1 Bootstrap Projelerinde Kullanılan Uzak Dosyaların Bağlantı Adresleri

Dosya	Erişim Adresi
bootstrap.css	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css
jquery.js	https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js
bootstrap.js	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js

```

1 <!DOCTYPE html>
2 <html lang="tr">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <!-- HEAD bölümünde öncelikle yukarıdaki üç meta etiketinin gelmesi zorunludur -->
8     <title>Temel Bootstrap Şablonu</title>
9     <!-- bootstrap.css dosyasının bağlantısı -->
10    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" rel="stylesheet">
11  </head>
12  <body>
13    <h1>Merhaba</h1>
14    <!-- jquery.js dosyasının bağlantısı -->
15    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
16    <!-- bootstrap.js dosyasının bağlantısı -->
17    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
18  </body>
19 </html>
20

```

Resim 7.2 Uzak Yöntemle Oluşturulmuş Temel Bir Bootstrap Projesinin Kodları

Uzak yöntemle oluşturulacak bir Bootstrap projesinin giriş sayfası Resim 7.2'de görüldüğü şekilde kodlanmalıdır. İlk satırdaki doctype etiketi ile bu dokümanın bir HTML 5 dokümanı olduğu tarayıcıya bildirilmektedir. İkinci satırda dokümanın içeriğinin Türkçe dilinde olduğu belirtilmektedir. Üçüncü satırda başlayan HEAD bloğu 11. satıra kadar devam etmektedir. HEAD bloğundaki ilk üç meta etiketinin öncelikli yazılması zorunludur. Dördüncü satırda dokümanın dil kodlamasının UTF-8 olduğu belirtilmektedir. Beşinci satır ise dokümanın Internet Explorer ile görüntülenmesi durumunda, en son sürüm olan EDGE motoru ile yorumlanması direktifini vermektedir. Altıncı satırdaki viewport etiketi sayfanın mobil tarayıcılarda düzgün görüntülenmesi için gereklidir. Sekizinci satırdaki title etiketi ile tarayıcı başlığında görüntülenecek başlık metni belirlenmektedir. 10. satırda uzak sunucudaki bootstrap.css dosyası dokümana eklenirken, 11. satırda HEAD bloğu kapatılmaktadır. 12. Satırda başlayan BODY bloğunda öncelikle kullanıcıya h1 etiketleri içinde Merhaba mesajı verilmektedir. BODY bloğunun kapatılmasından hemen önce 15. satırda jquery.js, 16. satırda ise bootstrap.js dosyaları dokümana eklenmektedir.

✓ Bu dosyaların isimlerindeki min ifadesi dosyaların küçültülmüş (minified) sürümlerini ifade etmektedir. Bu sürümlerde dosya boyutuna katkı getirebilecek boşluk ve paragraf sonu karakterlerinin hepsi kaldırılır. Bu nedenle dosya içerikleri çok uzun bir satır olarak gösterilir. Bu dosyalar daha küçük boyutlu olsa da, içeriklerinin okunması ve düzenlenmesi daha zordur. Dosyaları incelemek ve düzenlemek isteyen programcılar isminde “min” ifadesi bulunmayan dosyaları tercih etmelidir.

Uzak yöntemi kullanmanın ilk avantajı, bu dosyalar için sunucu disk alanı harcanmamasıdır. Bunun yanında, Bootstrap pek çok projede kullanılan bir web çatısı olduğundan, bu dosyaların projeler arasında paylaşılması olasıdır. Projenizin kullanıcı bilgisayarına yüklenmesi sırasında, başka bir proje için zaten indirilmiş olan dosyalar tekrar kullanılabilir. Bu sayede projenizin yüklenme süresi kısalarak performansı artacaktır. Bununla birlikte, uzak dosya kullanımının önemli sınırlılıkları da bulunmaktadır. Öncelikle uzak sistemde proje geliştirmek için internet bağlantısı gerekmektedir. Sistemdeki her geliştirme adımında sonuçların izlenebilmesi için tarayıcının gerekli dosyaları içerik dağıtım ağından çekmesi gerekecektir. Bu çerçevede internet erişimi sağlanamayan durumlarda geliştirme süreci aksayabilecektir. Bir diğer sınırlılık ise projenin dış kaynağa bağımlı olmasıdır. Her ne kadar içerik dağıtım ağları güvenilir çalışan araçlar olsalar da, proje başarısı için elzem olan bu dosyaların dış kaynaklara bağımlı bırakılması önerilen bir yaklaşım değildir. Kullanıcıların sizin sayfanıza erişebildiği, fakat içerik dağıtım ağlarına erişemediği senaryolarda projeniz düzgün görüntülenemeyecektir. Uzak dosya kullanma yaklaşımının son sınırlılığı bu dosyaları kendi ihtiyacınıza göre düzenleyememenizdir. Örneğin css dosyasındaki font ve renk değişiklikleri, özel bir element için tanımlayacağınız kurallar ya da kullanmayacağınız özellikler için gereksiz kodları silmek gibi işlemler uzak dosyalarla çalışılırken yapılamaz. Projenizin bu sınırlılıklardan etkileneceğini düşünüyorsanız, yerel yöntemi tercih etmeniz faydalı olacaktır.

Yerel yöntem ile proje başlatmak için Bootstrap dosyaları ve JQuery dosyalarının ilgili sunuculardan indirilmesi gerekmektedir. JQuery dosyalarını indirmek için Tablo 7.1'deki bağlantı adresi kullanılabilir. Bu adresteki 1.11.3 sürümü Internet Explorer tarayıcısının 6/7/8 sürümlerine destek vermektedir. Eski tarayıcılara destek verilmek istenmiyorsa <http://jquery.com/download/> adresinden 2.x ya da 3.x sürümlerinden biri indirilebilir. Bootstrap dosyaları <http://trbootstrap.com/getting-started/#download> adresinden temin edilebilir. Buradaki “Bootstrap İndir” bağlantısı tıklandığında gerekli tüm dosyalar sıkıştırılmış bir paket hâlinde bilgisayarınıza transfer edilir.

✓ Dreamweaver, Aptana ve Netbeans gibi modern editörler yeni Bootstrap projelerini otomatik olarak başlatabilmektedir. Bu sistemler, dosyaları indirerek bağlantı kodlarını ekleyebilmektedir.

Bu paketin içinde css, fonts ve js isimli üç klasör gelmektedir. CSS klasörünün içinde bootstrap.css dosyası, sıkıştırılmış (min) hâli ve map dosyası bulunabilir. Bunun yanında görsel olarak daha zengin bir deneyim sunan bootstrap-theme.css dosyası, sıkıştırılmış hâli ve map dosyası bulunmaktadır. Bu dosyalar içinden sadece bootstrap.css ya da sıkıştırılmış hâli olan bootstrap.min.css dosyasının kullanılması yeterlidir. İnternet programcılığı gelenekleri gereği css uzantılı dosyaların css isimli bir klasörde tutulması önerilen bir yaklaşımdır. Fonts klasörü içinde kullanılabilecek simgeler pek çok formatta sunulmaktadır. Bu dosyalara dokunulmadan fonts klasörü, css klasörü ile aynı yerde tutulmalıdır. js klasörü içinde bootstrap.js ve sıkıştırılmış hâli olan bootstrap.min.js dosyaları bulunmaktadır. Bunlardan biri jquery dosyasından sonra projeye eklenmelidir. Yine internet programcılığı gelenekleri gereği javascript dosyalarının js isimli bir klasörde tutulması önerilmektedir. Bu dosyaların bağlanmasıyla oluşturulacak bir projenin kodları Resim 7.3'te görülmektedir.

```

1 <!DOCTYPE html>
2 <html lang="tr">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <!-- HEAD bölümünde öncelikle yukarıdaki üç meta etiketinin gelmesi zorunludur -->
8     <title>Temel Bootstrap Şablonu</title>
9     <!-- bootstrap.css dosyasının bağlantısı -->
10    <link href="css/bootstrap.min.css" rel="stylesheet">
11  </head>
12  <body>
13    <h1>Merhaba</h1>
14    <!-- jquery.js dosyasının bağlantısı -->
15    <script src="js/jquery.min.js"></script>
16    <!-- bootstrap.js dosyasının bağlantısı -->
17    <script src="js/bootstrap.min.js"></script>
18  </body>
19 </html>

```

Resim 7.3 Yerel Yöntemle Oluşturulmuş Temel Bir Bootstrap Projesinin Kodları

Merhaba

Resim 7.4 Temel Bir Bootstrap Projesinin Çıktısı

Bu proje çalıştırıldığında tarayıcıda Resim 7.4'tekine benzer bir çıktı oluşmalı ve tarayıcı konsolunda bir hata mesajı gözlenmemelidir. Bu proje ünitemizin bundan sonraki bölümlerinde kullanılacak temel kodları ve bağlantıları içermektedir. Bundan sonra göreceğiniz tüm kodları BODY etiketi içine yerleştirdiğinizde gerekli etkileri inceleyebilirsiniz.

Şimdi Bootstrap sisteminin getirdiği hazır elementleri ve etkileri inceleyelim.

✓ Tipografi (typography) yazılı materyallerin okunaklı, kavranabilir ve çekici hazırlanması ile ilgilenen bir sanat dalıdır. Tipografinin araçları arasında, yazı tipi (font), yazı yüksekliği, satır yüksekliği, satırlar arası boşluklar, metin elementleri arasındaki boşluklar gibi özellikler yer almaktadır.

Metin Etiketleri

Bootstrap, sayfalardaki metinlerin okunurluğunu arttırmak için CSS kuralları ve pek çok uygulama sınıfı sağlamaktadır.

Temel Metin Etiketleri

Bootstrap HTML dilindeki tüm başlık düzeylerini destekler. Bu başlık düzeylerinin kodlanmasını ve etkilerini incelemek için Resim 7.5'i inceleyiniz. Bunun yanında, düz metinler üzerinde başlıklardaki görsel etkileri oluşturan araç sınıfları bulunmaktadır. Bilindiği gibi HTML dilindeki başlık etiketleri blok elementlerdir. Bu elementlerin etkilerini metin içinde kullanmak için aynı isimli araç sınıfları kullanılabilir. Örneğin, aşağıdaki tasarımda son p etiketinin içindeki span elementine dördüncü düzey başlık etkileri kazandırılmıştır.

1. Düzey Başlık

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque rutrum lorem eget dictum mattis. Etiam sed leo ut felis luctus condimentum. Ut dapibus, lacus quis pretium feugiat, dolor risus malesuada urna, vel tincidunt lacus ex sit amet neque.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque rutrum lorem eget dictum mattis. Etiam sed leo ut felis luctus condimentum. Ut dapibus, lacus quis pretium feugiat, dolor risus malesuada urna, vel tincidunt lacus ex sit amet neque. Suspendisse sagittis mi vitae justo congue, id hendrerit sapien venenatis. Vivamus pretium est nec nunc accumsan feugiat.

2. Düzey Başlık

3. Düzey Başlık

4. Düzey Başlık

5. Düzey Başlık

6. Düzey Başlık

Metin içi 4. düzey başlık Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque rutrum lorem eget dictum mattis. Etiam sed leo ut felis luctus condimentum.

```
<div class="container">
  <h1>1. Düzey Başlık</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque rutrum lorem eget dictum mattis. Etiam sed leo ut felis luctus condimentum. Ut dapibus, lacus quis pretium feugiat, dolor risus malesuada urna, vel tincidunt lacus ex sit amet neque.</p>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque rutrum lorem eget dictum mattis. Etiam sed leo ut felis luctus condimentum. Ut dapibus, lacus quis pretium feugiat, dolor risus malesuada urna, vel tincidunt lacus ex sit amet neque. Suspendisse sagittis mi vitae justo congue, id hendrerit sapien venenatis. Vivamus pretium est nec nunc accumsan feugiat.</p>
  <h2>2. Düzey Başlık</h2>
  <h3>3. Düzey Başlık</h3>
  <h4>4. Düzey Başlık</h4>
  <h5>5. Düzey Başlık</h5>
  <h6>6. Düzey Başlık</h6>
  <p><span class="h4">Metin içi 4. düzey başlık</span> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque rutrum lorem eget dictum mattis. Etiam sed leo ut felis luctus condimentum.</p>
</div>
```

Resim 7.5 Bootstrap Sistemindeki Temel Tipografik Elementler ve Etkileri

Paragraf elementleri okunurluğu destekleyecek şekilde düzenlenmiştir. Yazı yüksekliği 14 piksel, satır yüksekliği 1,428 olarak belirlenmiştir. Paragraflar, 10 piksel yüksekliğindeki alt boşluklarla (margin) birbirlerinden ayrılmıştır.

İşaretleme Etiketleri

Bootstrap düz metinleri işaretleme için pek çok HTML etiketini desteklemektedir. Resim 7.6'da bu etiketler ve etkileri örneklenmiştir. Etiketlerin açıklamaları ve kullanımları Tablo 7.2'de sunulmuştur. Bu etiketleri kullanmak için bir p etiketinin içinde aşağıdaki metinleri kullanabilirsiniz. Örneğin mark etiketi aşağıdaki şekilde kullanılır:

<p>Metnin bir bölümünü öne çıkarmak için <mark>mark etiketi</mark> kullanılabilir.</p>

Metnin bir bölümünü öne çıkarmak için **mark etiketi** kullanılabilir.

Metnin bir bölümünün üstünü çizmek için **s-etiketi** kullanılabilir.

Silinmiş metinleri işaretlemek için **del-etiketi** kullanılabilir.

Sonradan eklenmiş metinleri işaretlemek için **ins etiketi** kullanılabilir.

Metnin bir bölümünün altını çizmek için **u etiketi** kullanılabilir.

Metnin bir bölümünü daha küçük yazmak için **small etiketi** kullanılabilir.

Metnin bir bölümü kalın olarak vurgulamak yazmak için **strong etiketi** kullanılabilir.

Metnin bir bölümünü yatık olarak vurgulamak için **em etiketi** kullanılabilir.

Resim 7.6 Metinleri İşaretlemek İçin Kullanılabilecek Etiketler ve Etkileri

Tablo 7.2 Metinleri İşaretlemek İçin Kullanılabilecek Etiketler ve Açıklamaları

Etiket	Açıklama	Kullanım
mark	Metnin arkasını boyayarak vurgular.	<code><mark>metin</mark></code>
s	Metnin üstünü çizer.	<code><s >metin</ s></code>
del	Silinmiş metni vurgulamak için üstünü çizer.	<code>< del>metin</code>
ins	Eklenmiş metni vurgulamak için altını çizer.	<code><ins >metin</ins ></code>
u	Metni vurgulamak için altını çizer.	<code>< u>metin</ u></code>
small	Metni bulunduğu ortamda %85 oranında küçültür.	<code><small >metin</small ></code>
strong	Metni vurgulamak için kalın yazar.	<code>metin</ strong></code>
em	Metni vurgulamak için yatık yazar.	<code>< em>metin</code>

Hizalama Sınıfları

Metinleri bulundukları ortamda hizalamak için beş adet hizalama sınıfı sağlanmaktadır. Bu sınıflar metin elementlerine eklenerek kullanılır. Örneğin bir paragrafın içeriğini sağa hizalamak için açılış etiketi `<p class='text-right'>` şeklinde düzenlenmelidir. Bu sınıfların açıklamaları Tablo 7.3'te, etkileri Resim 7.7'de sunulmuştur.

Tablo 7.3 Hizalama Sınıfları ve Açıklamaları

Etiket	Açıklama
text-left	Element içindeki metni sola hizalar. Bootstrap sisteminde varsayılan hizalama türüdür.
text-right	Element içindeki metni sağa hizalar.
text-center	Element içindeki metni yatay ortaya hizalar.
text-justify	Element içindeki metni iki yana yaslar.
text-nowrap	Element içindeki metnin tarayıcı genişliğinden daha geniş olması durumunda otomatik olarak alt satıra alınmasını engeller. Bu özellik programlama kodları gibi, tek satır hâlinde okunması daha kolay olan metinleri göstermek için kullanılır.

Sola hizalı metin. Sola hizalı metin. Sola hizalı metin. Sola hizalı metin. Sola hizalı metin.

Sağa hizalı metin. Sağa hizalı metin. Sağa hizalı metin. Sağa hizalı metin. Sağa hizalı metin.

Ortaya hizalı metin. Ortaya hizalı metin. Ortaya hizalı metin. Ortaya hizalı metin. Ortaya hizalı metin.

İki yana yaslanmış metin. İki yana yaslanmış metin. İki yana yaslanmış metin. İki yana yaslanmış metin. İki yana yaslanmış metin.

Tarayıcı genişliğinden etkilenmeden uzatılacak metin. Tarayıcı genişliğinden etki

Resim 7.7 Bootstrap Hizalama Sınıflarının Metin Üzerindeki Etkileri

Vurgulama Sınıfları

Metinleri renklendirmek için altı adet vurgulama sınıfı kullanılabilir. Bu sınıflar, bağlantılar da dâhil olmak üzere, tüm metin elementlerine uygulanabilir. Kavramsal sınıflar genellikle metin elementlerinin tümüne (ör: <p>) ya da bu elementlerin içindeki belirli bir alana (ör:) uygulanır. Alanları işaretlemek için, metin içindeki bölüm span etiketleri içine alınarak bu etiketlere vurgulama sınıflar uygulanır. Tüm kullanımlar Resim 7.8'de sunulmuştur.

Yardımcı sınıf uygulanmamış metin	<code><p>Yardımcı sınıf uygulanmamış metin</p></code>
text-muted	<code><p class="text-muted">text-muted</p></code>
text-primary	<code><p class="text-primary">text-primary</p></code>
text-success	<code><p class="text-success">text-success</p></code>
text-info	<code><p class="text-info">text-info</p></code>
text-warning	<code><p class="text-warning">text-warning</p></code>
text-danger	<code><p class="text-danger">text-danger</p></code>
işlem başarılı.	<code><p>İşlem başarılı.</p></code>

Resim 7.8 Bootstrap Sistemindeki Yardımcı Sınıflar ve Kullanım Yolları



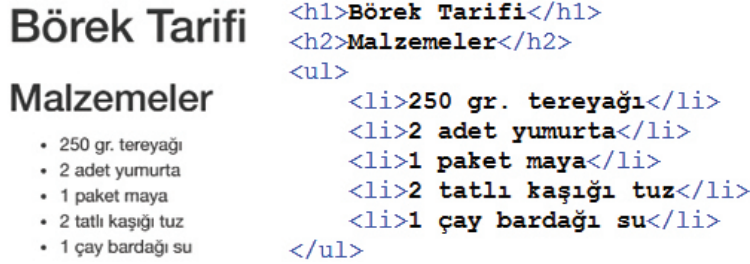
dikkat

Vurgulama sınıflarına benzer olarak, nesnelerin artalanlarını boyamak için kullanılacak kavramsal sınıflar da bulunmaktadır. Bunlar vurgulama sınıflarına benzer şekilde isimlendirilmektedir. Ancak, kavramsal sınıflar text- ön eki yerine, bg-ön eki ile başlar. Bu sınıfların etkilerini görmek için bir p etiketine bg-primary, bg-success, bg-info, bg-warning ve bg-danger sınıflarını uygulayabilirsiniz. Kavramsal artalan sınıfları arasında bg-muted olmadığına dikkat ediniz.

Bootstrap Listeleri

Bootstrap, HTML dilinde tanımlı üç liste türüne destek vermektedir: sırasız listeler (), sıralı listeler () ve tanımlama listeleri (<dl>).

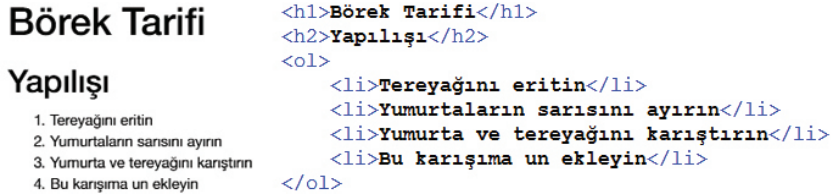
Sırasız listelerde (unordered list) elementlerin birbirleri ile herhangi bir sıra ya da hiyerarşi ilişkisi yoktur. Bu nedenle veriliş sıraları önemli değildir. Sırasız listelerde elementler madde işaretleri ile belirtilir. Örneğin bir yemek tarifinde gerekli malzemeler sırasız listelerle sunulabilir (Resim 7.9).



Resim 7.9 Sırasız Liste Kullanımı

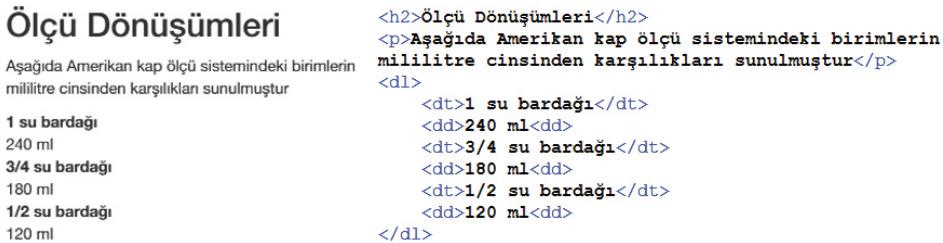
İkinci liste türü elementler arasında hiyerarşi ya da sıralama ilişkisinin bulunduğu sıralı listelerdir (ordered list). Örneğin bir yemek tarifindeki işlem basamakları sıralı listeler ile verilebilir (Resim 7.10). Bu listeler varsayılan olarak numaralandırılmış olarak gelmekle beraber, ol etiketinin type özelliği ile bu davranış değiştirilebilir.

✓ Madde-tanım çiftleri için Tablo elementi sıklıkla kullanılmaktadır. Bununla birlikte, semantik olarak tanımlama listelerinin kullanılması daha uygundur.



Resim 7.10 Sıralı Liste Kullanımı

Tanımlama listeleri (description list) başlık ve açıklama çiftlerinden oluşan listelerdir. Örnek olarak, sayfada geçen terimlerden oluşan küçük bir sözlük oluşturmak için tanımlama listeleri kullanılabilir (Resim 7.11).



Resim 7.11 Tanımlama Listesi Kullanımı

Bootstrap Tabloları

Tablolar en çok kullanılan HTML yapıları arasındadır. Bootstrap tabloların stillenmesi için pek çok seçenek sunmaktadır. Yeni bir tablo üretildiğinde Bootstrap temel tipografik öğeler dışında herhangi bir düzenleme yapmaz (Resim 7.12- Sol). Table etiketine table sınıfı eklendiğinde tablonun sunumunda önemli

değişiklikler gözlenecektir (Resim 7.12-Sağ). Öncelikle tablo bulunduğu ortamı kapsayacak kadar geniş-letilir. Bunun yanında hücrelere padding eklenerek hücreler arası boşluklar yükseltilir. Son olarak satırlara alt kenarlık eklenerek tablonun okunması kolaylaştırılır.

Öğrenci No	Adı	Soyadı	Öğrenci No	Adı	Soyadı
346	Ali	Köse	346	Ali	Köse
394	Ayşe	Biber	394	Ayşe	Biber
600	Cemil	Can	600	Cemil	Can
747	Semih	Aycan	747	Semih	Aycan

Resim 7.12 Bootstrap ile Hazırlanmış Tablolar

Bu temel düzenlemelerin yanında, aşağıda verilen sınıflar table etiketine eklenerek tabloların sunumları zenginleştirilebilir (Tablo 7.4).

Tablo 7.4 Table Etiketinde Kullanılabilecek Sınıflar

Sınıf	Açıklama
table-striped	Tablo satırlarını bir koyu, bir açık olacak şekilde boyar.
table-bordered	Tablonun dış kenarlıklarını ve hücreler arasındaki dikey kenarlıkları ekler.
table-hover	Fare üzerine geldiğinde tablo satırlarını vurgular.
table-condensed	Hücreler arası boşlukları azaltarak daha yoğun bir tablo oluşturur.

Bu sınıflar table sınıfı ile birlikte tablolara eklenerek oldukça zengin tablo sunumları oluşturulabilir. Resim 7.13'teki örnekte table-striped sınıfının etkileri gösterilmiştir.

Öğrenci No	Adı	Soyadı
346	Ali	Köse
394	Ayşe	Biber
600	Cemil	Can
747	Semih	Aycan

```

<table class="table table-striped">
  <thead>
    <tr><th>Öğrenci No</th><th>Adı</th><th>Soyadı</th></tr>
  </thead>
  <tbody>
    <tr><td>346</td><td>Ali</td><td>Köse</td></tr>
    <tr><td>394</td><td>Ayşe</td><td>Biber</td></tr>
    <tr><td>600</td><td>Cemil</td><td>Can</td></tr>
    <tr><td>747</td><td>Semih</td><td>Aycan</td></tr>
  </tbody>
</table>

```

Resim 7.13 Table-Striped Sınıfı ve Kullanımı

Tabloların satırları ve hücreleri kavramsal sınıflar kullanılarak boyanabilir. Kullanılabilecek kavramsal sınıflar ve anlamları Tablo 7.5'te sunulmuştur.

Tablo 7.5 Tablolarda Kullanılabilecek Kavramsal Sınıflar

Sınıf	Açıklama
active	İçeriği vurgulamak için kullanılır.
success	Olumlu ya da başarılı bir içeriği belirtir.
info	Düşük vurgulu, bilgilendirici içerikleri belirtir.
warning	Okuyucu dikkatini gerektiren içerikleri belirtir.
danger	Olumsuz ya da tehlikeli içerikleri belirtir.

Resim 7.14'te kavramsal sınıfların tablo satırlarına uygulanması örneklenmektedir.

Kavramsal sınıflar ve etkileri

Kavramsal sınıf yok

active

success

info

warning

danger

```
<table class="table">
  <thead>
    <tr><th>Kavramsal sınıflar ve etkileri</th></tr>
  </thead>
  <tbody>
    <tr><td>Kavramsal sınıf yok</td></tr>
    <tr class="active"><td>active</td></tr>
    <tr class="success"><td>success</td></tr>
    <tr class="info"><td>info</td></tr>
    <tr class="warning"><td>warning</td></tr>
    <tr class="danger"><td>danger</td></tr>
  </tbody>
</table>
```

Resim 7.14 Tablo Satırlarına Kavramsal Sınıfların Uygulanması

Bootstrap Boyut Sistemi

Bootstrap öncelikle mobil cihazları hedef alan tepkisel bir sistemdir. Bu sistemde içerikler farklı ekran boyutları, oranları ve tarayıcı genişliklerinde tepki verecek şekilde tasarlanabilmektedir. Örneğin bir menü çubuğu 1024 piksel genişliğindeki bir tarayıcıda yatay görünürken, 600 piksel genişliğindeki bir tarayıcıda dikey görünebilmektedir. Bu sayede, Bootstrap ile hazırlanan projeleri, bilgisayar ekranı, tablet bilgisayar ve akıllı telefon üzerinde farklı şekillerde sunulabilmektedir. Bu tepkisel sistemin yönetilmesi için boyut sisteminin anlaşılması oldukça önemlidir.

Boyut sistemi ön tanımlı kırılma noktaları ve bu noktalar arasındaki aralıklara verilen kısa belirteçlerle işlemektedir (Tablo 7.6). Bu sisteme göre, tarayıcı genişliğinin 768 pikselden küçük olduğu durumları ifade etmek için xs (Extra Small) belirteci kullanılır. Örneğin bir satın alma formunun 768 pikselden küçük tarayıcı genişliklerinde görülmesi istenmiyorsa, bu formu kapsayan etikete hidden-xs sınıfının eklenmesi yeterlidir. Tarayıcı genişliğinin 768px – 991px arasında olduğu durumları ifade etmek için sm (Small); 992px – 1199px arasında olduğu durumlar için md (Medium); 1200 pikselden büyük olduğu durumlar içinse lg (Large) belirteci kullanılır. Bu belirteçlerin kullanımı sonraki bölümlerde detaylı olarak incelenecektir.

Tablo 7.6 Bootstrap Sistemindeki Kırılma Noktaları ve Genişlik Belirteçleri

Genişlik	xs	sm	md	lg
Tarayıcı Genişliği (x)	$x < 768\text{px}$	$768\text{px} \leq x \leq 991\text{px}$	$992\text{px} \leq x \leq 1199\text{px}$	$x \geq 1200\text{px}$

Bootstrap Izgara Sistemi

Bootstrap sisteminin en önemli elementlerinden biri sayfa içeriğinin yerleşimini düzenleyen ızgara (grid) sistemidir. Bootstrap ızgara sistemi container, row ve column olmak üzere üç temel bileşenden oluşmaktadır.

Container

Container sınıfı sayfa içeriğini ve kurulacak ızgara sistemini kapsamak için üretilmiştir. Sayfa içerikleri ve ızgara sistemi container özellikli bir element içine yerleştirilmelidir (Resim 7.15). Akışkan ve sabit genişlikli olmak üzere iki container sınıfı bulunmaktadır. Akışkan container sınıfı (container-fluid) her zaman tarayıcı genişliğinin tümünü kapsayacak şekilde boyutlandırılır. Sabit genişlikli container sınıfı (container) ise tarayıcı genişliğine tepki verecek şekilde boyutlandırılır. Tablo 7.7’te farklı tarayıcı boyutlarında container sınıfının alacağı boyutlar sunulmuştur.

Tablo 7.7 Bootstrap Sisteminin Kırılma Noktaları ve Container Sınıfının Bu Noktalarındaki Genişlikleri

Genişlik	xs	sm	md	lg
Tarayıcı Genişliği (x)	$x < 768\text{px}$	$768\text{px} \leq x \leq 991\text{px}$	$992\text{px} \leq x \leq 1199\text{px}$	$x \geq 1200\text{px}$
container	Otomatik	750px	970px	1170px

Görüldüğü gibi xs tarayıcı genişliğinde container sınıfı, container-fluid sınıfı gibi davranmaktadır. Bu genişlikte container sınıfı da tüm tarayıcı genişliğini kapsayacak şekilde tasarlanmıştır. sm tarayıcı genişliğine geçildiğinde container sınıfı 750 piksel genişliğinde sabitlenir. sm ve üzerindeki tüm tarayıcı genişliklerinde container sınıfı sayfada ortalanmış olarak gösterilir. container sınıfı md tarayıcı genişliğinde 950 piksel, lg tarayıcı genişliğindeyse 1170 piksel genişliğinde gösterilecektir. 1200 pikselin üzerindeki tüm tarayıcı genişlikleri lg olarak ele alındığından container sınıfının en yüksek genişliği 1170 piksel ile sınırlandırılmıştır. Bununla birlikte container-fluid sınıfı için herhangi bir üst limit yoktur. İçeriklerin yerleşimi ve okunurluğun desteklenmesi için her iki sınıfta da sağdan ve soldan 15 piksel genişliğinde boşluklar (padding) bırakılmaktadır.

```

<div class="container">
  <p>İçerik, içerik, içerik.</p>
  <p>İçerik, içerik, içerik.</p>
</div>
<div class="container-fluid">
  <p>İçerik, içerik, içerik.</p>
  <p>İçerik, içerik, içerik.</p>
</div>

```

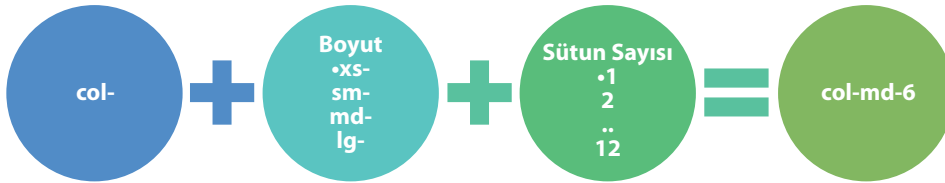
Resim 7.15 Container ve Container-Fluid Sınıflarının Kullanımı

Row

Satırlar (row) yatay sütun grupları oluşturmak için kullanılır. Düzgün hizalama ve boşluk bırakılabilmesi için her satır sabit genişlikli (container) ya da akışkan (container-fluid) bir container element içine girmelidir. Yatay genişliği bölümlmek için kullanılan sütunlar row sınıfının hemen içine yerleştirilmelidir.

Column

Sayfadaki içeriği dikey olarak düzenlemek için sütunlar kullanılır. Bootstrap sisteminde yatay genişlikler 12 parçaya bölünerek işlenir. Sütunlar oluşturulurken hangi ekran genişliğinde kaç yatay dilim kapsayacağı belirlenir. Bu amaçla Şekil 7.2'deki şema kullanılır.

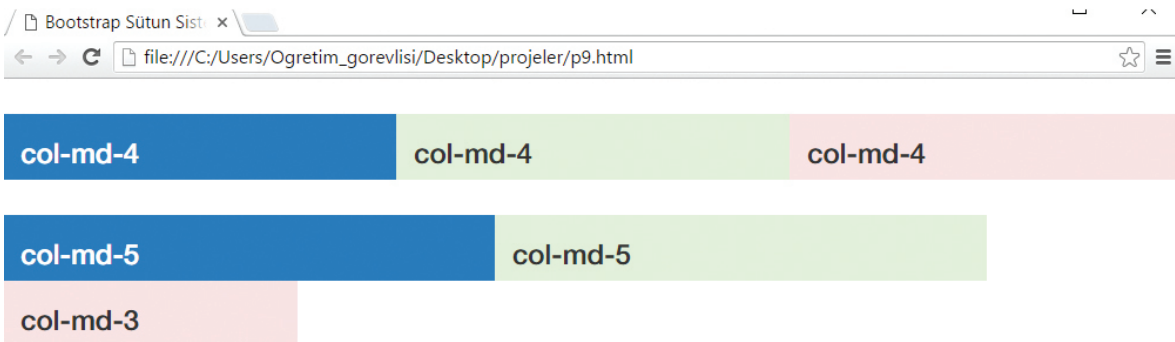


Şekil 7.2 Bootstrap Sisteminde Sütun Belirteci Oluşturma Şeması

Her sütun isimlendirmesi “col-“ ön eki ile başlar. Bundan sonra yapılan belirlemenin hangi tarayıcı genişliği için geçerli olduğu, genişlik belirteçleri ile bildirilir (Ör: col-md). Bunun ardından bu sütunun belirlenen tarayıcı genişliğinde ne kadar yer kaplayacağı sayısal olarak verilir. Örneğin, col-md-4 belirteci, sütunun 992px – 1199px aralığındaki md tarayıcı boyutunda, genişliğin üçte birini (4/12) kaplayacağını belirtmektedir. Benzer şekilde col-sm-6 belirteci, sütunun 768px-991px aralığındaki sm tarayıcı boyutunda, genişliğin yarısını (6/12) kaplayacağını belirtmektedir.

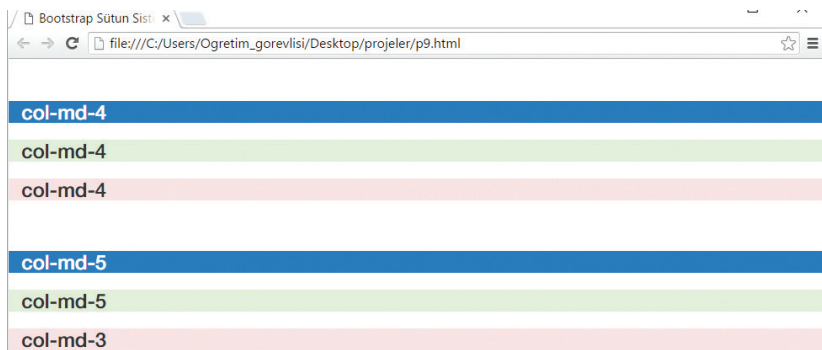
Sütunlar belirli bir tarayıcı boyutu için sütun sayıları 12'yi bulana kadar yan yana getirilir. Örneğin üç adet col-md-4 sınıfı sütun, md tarayıcı boyutunda tüm genişliği kaplayacaktır (Resim 7.16 – Üst). Bununla birlikte, iki adet col-md-5 sınıfı sütun, yan yana geldiğinde ekranın tümünü kaplayamayacaktır. Bu sütunlar sola yaslanarak, sağ taraflarında iki birimlik boşluk bırakılacaktır. Bu iki sütunun yanında col-md-3 belirteçli bir sütun getirilmesi durumunda, sütun sayıları toplamı 12'yi geçeceğinden son sütun alta gönderilecektir (Resim 7.16 - Alt). Bu senaryo aşağıda görselleştirilmiştir. Bootstrap sisteminde sütun ya da satırlar varsayılan olarak renksiz gelmektedir. Bu örnekte sütunların sınırlarının belirlenmesi için kavramsal sınıflar kullanılarak art alanlar boyanmıştır.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-md-4 bg-primary"><h3>col-md-4</h3></div>
    <div class="col-md-4 bg-success"><h3>col-md-4</h3></div>
    <div class="col-md-4 bg-danger"><h3>col-md-4</h3></div>
  </div>
  <p>&nbsp;</p>
  <div class="row">
    <div class="col-md-5 bg-primary"><h3>col-md-5</h3></div>
    <div class="col-md-5 bg-success"><h3>col-md-5</h3></div>
    <div class="col-md-3 bg-danger"><h3>col-md-3</h3></div>
  </div>
</div>
```



Resim 7. 16 Bootstrap Sütun Sistemi

Bootstrap sisteminde bir tarayıcı boyutu için belirlenen yerleşim kuralları üst tarayıcı boyutları için geçerli kalır. Örneğin sm tarayıcı boyutunda oluşturulan iki sütunlu bir yapı (Ör: col-sm-6 , col-sm-6), üst tarayıcı boyutları olan md ve lg için farklı bir kural belirtilmemişse aynen korunur. Sitenin sunumu md ve lg boyutları için iki sütunlu olacak şekilde düzenlenecektir. Alt tarayıcı boyutlarına geçildiğinde, bu boyutlar için farklı kurallar belirtilmemişse en küçük tarayıcı boyutu olan xs boyutundaki kurallar işletilir. xs boyutu için herhangi bir sütun tanımlaması yapılmadıysa, varsayılan olarak her sütun ekranı kaplayacak şekilde düzenlenecektir. Bu nedenle, sunum yapısı tek sütunlu hâle geçirilir. Yukarıdaki örnekten devam edecek olursak, bu tarayıcıyı biraz daraltarak sm boyutuna düşürdüğümüzde sütun yapısı kırılacaktır. sm tarayıcı boyutu için herhangi bir kural tanımlanmadığından, her sütun tüm genişliği kapsayacak şekilde genişletilecektir (Resim 7.17).

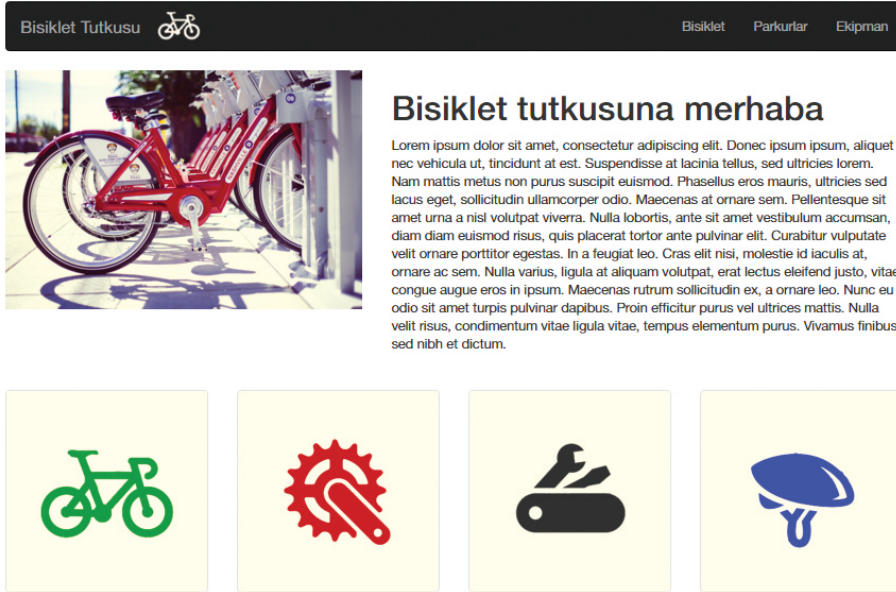


Resim 7.17 md Tarayıcı Boyutu İçin Hazırlanmış Bir Sütun Yapısının sm Tarayıcı Boyutunda Görünümü

Bootstrap tepkisel sistemi kullanılarak farklı tarayıcı boyutlarında arayüz sunumları değiştirilebilir. Örnek olarak, bir sütun lg ekran genişliğinde 2, md ekran genişliğinde 4, sm ekran genişliğinde ise 6 sütun kaplayabilir. Bu amaçla, sütun olarak kullanılacak div elementine col-lg-2, col-md-4 ve col-sm-6 sınıfları aynı anda atanır. Bootstrap tarayıcı genişliğini sorgulayarak gerekli sınıfı seçip sütuna uygulayacaktır. Bu yolla farklı tarayıcı ya da ekran boyutları için sunum sistemleri oluşturmak mümkündür.

Örneğin, Bisiklet Tutkusu (Resim 7.18) adlı si-temiz bir menü çubuğu, bir fotoğraf, bir yazı alanı ve dört büyük ikondan oluşmaktadır. Resim 7.18 ve Resim 7.19'da bu basit sitenin md, sm ve xs tarayıcı boyutlarındaki görüntüleri verilmiştir. Menü çubuğu md ve sm tarayıcı boyutlarında aynı görünürken, xs tarayıcı boyutunda menü elementleri saklanarak üç çubuklu (hamburger) düğme altına saklanmak-

tadır. Bu sistem, özellikle cep telefonları gibi dar ekranlarda oldukça kullanışlıdır. Normalde yatay görünen menü çubuğu okunurluğu desteklemek için dar ekranlarda dikey gösterilmek zorundadır. Ekran alanını ekonomik kullanmak adına çubuk kullanıcı etkileşimiyle görülebilir hâle getirilmektedir. Resim elementi farklı ekran büyüklüklerine tepki vermektedir. Bu element lg ekran boyutunda 4, md ekran boyutunda 5, sm ve xs ekran boyutlarında 12 sütun kaplamaktadır. Bu sayede resim geniş ekranlarda yazının solunda, dar ekranlarda ise yazının üstünde görüntülenmektedir. Yazı elementi bisiklet ile birlikte görüntülenecek şekilde tasarlanmıştır. Bu elementin genişlikleri lg ekran boyutunda 8, md ekran boyutunda 7, sm ve xs ekran boyutlarında 12 olarak belirlenmiştir. Bu sayede yazı ve resim, md ve lg ekran boyutlarında yan yana (sütun belirteçleri toplamalarının 12 olduğuna dikkat ediniz) olacak şekilde görüntülenmektedir.



Resim 7.18 Örnek Sitenin md Ekran Boyutunda Görünümü

Sitede verilen hizmetlere rahat ulaşım sağlamak amacıyla oluşturulan dört ikonlu bağlantı md ekran boyutunda üç, sm ekran boyutunda altı sütun kaplamaktadır. Bu sayede bağlantılar md ve lg ekran boyutlarında ekranı kaplayacak şekilde yan yana ($4 \times 3 = 12$) görüntülenecektir. Ekran boyutu sm olduğunda bağlantılar altı sütun kaplayacağından iki bağlantı yan yana görüntülenecektir (Resim 7.19 – Sol). Kalan iki bağlantı ise otomatik olarak alt satıra taşınacaktır. xs ekran boyutu için herhangi bir tanımlama yapılmadığından bağlantılar otomatik olarak 12 sütun kaplayacak şekilde alt alta gösterilecektir.



Resim 7.19 Örnek Sitenin sm (Sol) ve xs (Sağ) Ekran Boyutlarında Görünümü

Form Elementleri

Bootstrap HTML dilindeki form elementleri için önemli zenginleştirme seçenekleri sunmaktadır. Bu elementler Bootstrap ile işlendiklerinde tüm tarayıcılarda aynı etki ve görünümlere kavuşur. Elementler form etiketi kullanılarak yatay ve dikey giriş formları şeklinde düzenlenebilir. Öncelikle kullanılacak giriş elementlerini inceleyelim.

Düğmeler

Üç farklı HTML elementi düğme olarak kullanılabilir: `<a>`, `<button>` ve `<input>`. Aşağıda bu elementlerin düğme görünümü kazanması için gerekli işaretlemeler ve etkiler gösterilmiştir (Resim 7.20).

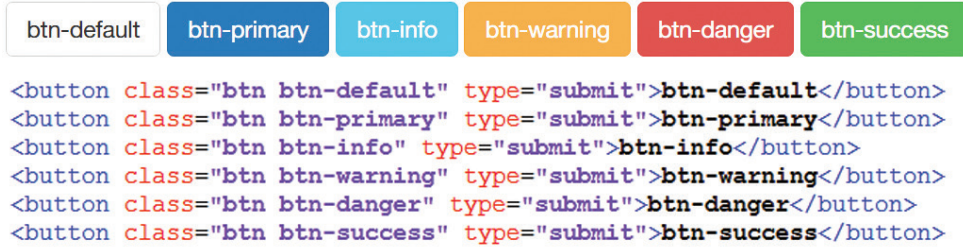
✓ Düğmeler oluşturulurken `btn` sınıfı ile birlikte, en az bir kavramsal sınıf ya da `btn-default` sınıfının eklenmesi gerekmektedir.



```
<a href="#" class="btn btn-default" role="button">Düğme 1</a>
<input type="submit" class="btn btn-default" value="Düğme 2"></input>
<input type="button" class="btn btn-default" value="Düğme 3"></input>
<button class="btn btn-default" type="submit">Düğme 4</button>
```

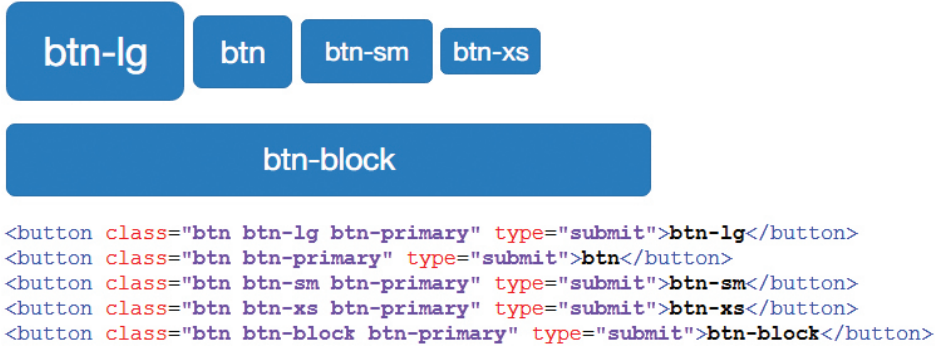
Resim 7.20 HTML Elementlerini Düğmeye Dönüştürmek

Görüldüğü gibi btn ve btn-default sınıflarının eklenmesiyle tüm elementler aynı görünüm ve etkileşim özelliklerini kazanabilmektedir. Burada, btn sınıfı elemente düğme özelliklerini kazandırmaktadır. btn-default sınıfı ise düğmeye varsayılan düğme görüntüsü kazandırmak için kullanılmaktadır. Varsayılan olarak Bootstrap düğmeleri beyaz zeminde siyah metin ile gelmekteyken, Resim 7.21'deki kavramsal sınıflarla farklı sunumlar oluşturulabilmektedir.



Resim 7.21 Düğme Oluşturulurken Kullanılabilecek Kavramsal Sınıflar

Kavramsal sınıfların yanında, düğmelerin ebatlarını belirlemek için kullanılabilecek dört sınıf daha bulunmaktadır. Ebat sınıflarının kullanımı kavramsal sınıflar gibi zorunlu değildir. Bu durumda btn sınıfından gelen değerler kullanılır. Gerekli olması hâlinde ebat sınıfları btn ve kavramsal sınıflar ile birlikte kullanılmalıdır. Resim 7.22'de ebat sınıfları btn-primary sınıfı ile birlikte kullanılmıştır. Özellikle mobil cihazlarda kullanılabilecek, tüm genişliği kaplayan blok düğmeler üretmek için btn-block sınıfı kullanılabılır. Blok düğmelerin mobil ekranlarda tıklanmaları diğer düğmelere oranla daha kolaydır.



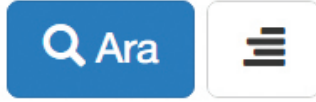
Resim 7.22 Düğmelerde Kullanılabilecek Ebat Sınıfları

Sayfa tasarımlarında daha az yer kaplaması için düğmeler gruplanarak araç kutuları oluşturulabilir. Bu amaçla, gruplanacak düğmeler btn-group sınıfı bir div elementinin içine eklenebilir. Dört düğmenin birleştirilmesi ile oluşturulmuş bir araç kutusu Resim 7.23'te görülmektedir. btn-group yerine, btn-group-vertical sınıfı kullanılarak dikey araç çubukları oluşturulabilmektedir.



Resim 7.23 Araç Kutusu Örneği

Düğme sunumlarını zenginleştirmek için simgeler kullanılabilir. Bootstrap sisteminde simgeler span elementleri kullanılarak eklenmektedir. Bu elementlere glyphicon sınıfı ile birlikte simgeyi belirten bir sınıf (ör: glyphicon-search) eklenir. Bir düğme simge ve metin birlikte kullanılarak oluşturulabileceği gibi, sadece simgeler kullanılarak da oluşturulabilir (Resim 7.24).



```
<button class="btn btn-primary" type="submit">
  <span class="glyphicon glyphicon-search"></span> Ara
</button>
<button class="btn btn-default" type="submit">
  <span class="glyphicon glyphicon-align-right"></span>
</button>
```

Resim 7.24 Simgeli Düğme Örnekleri

✓ Yerel projelerde simgelerin kullanılabilmesi için fonts klasörünün projedeki css klasörü ile aynı dizinde yer alması gerekmektedir.



Kullanılabilecek simge sınıflarının tam listesini <http://trbootstrap.com/components/#glyphicons> adresinde inceleyebilirsiniz.

Giriş Elementleri

Bootstrap sisteminde input, textarea ve select elementleri formlarda kullanılmak üzere desteklenmektedir. Elementler için açıklama metinleri label etiketleri içinde verilmelidir. Label etiketleri ile input elementleri bağlantısı id-for eşleşmesi ile sağlanır. input elementine verilen id değeri, label elementinin for özelliğine yazılarak eşleşme sağlanır. Bu kullanım sayesinde açıklama metnine tıklandığında ilgili element aktif hâle geçebilmektedir. Özellikle işaret kutusu (checkbox) ve radyo düğmeleri gibi küçük elementler için bu kullanım oldukça kullanıcı dostudur. Her label-input çifti form-group sınıflı div elementleri içine alınmalıdır. Bu yolla form elementlerinin düzgün yerleşimi sağlanır. input elementlerinin Bootstrap görüntü ve etkileşim özelliklerini kazanması için form-control sınıfı eklenir. Bu sınıf eklendiğinde element bulunduğu ortamdaki genişliğin tamamını kaplayacaktır (Resim 7.25).

Kayıt Formu

Adınız:

```
<h3>Kayıt Formu</h3>
<form>
  <div class="form-group">
    <label for="isim">Adınız:</label>
    <input class="form-control" type="text" id="isim" />
  </div>
</form>
```

Resim 7.25 Bootstrap Dikey Form Örneği

Label ve input yerleşimi varsayılan olarak dikeydir. Bu yerleşimin yataya çevrilmesi için form elementine `form-inline` sınıfı eklenmelidir (Resim 7.26). Bu durumda label ve input aynı satıra getirilecek, input nesnesinin genişliği daraltılacaktır. Bu görünüm xs tarayıcı boyutu dışındaki tüm boyutlarda korunur. Tarayıcı boyutu xs olduğunda, form otomatik olarak bir önceki örneğimizdeki dikey hâle geçecektir. Yatay formların amacı elementleri olabildiğince küçük bir alanda sunmaktır. Bu çerçevede label elementleri yerine HTML 5 ile birlikte gelen placeholder özellikleri kullanılabilir.

Giriş Formu

```
<h3>Kayıt Formu</h3>
<form class="form-inline">
  <div class="form-group">
    <label for="isim">e-Posta:</label>
    <input class="form-control" type="text" id="isim" />
  </div>
  <div class="form-group">
    <label for="isim">Şifre:</label>
    <input class="form-control" type="password" id="isim" />
  </div>
  <button class="btn btn-success" type="submit">Giriş</button>
</form>
```

Resim 7.26 Bootstrap Yatay Form Örneği

Giriş Formu

Resim 7.27 Placeholder Kullanımı

Element Grupları

Ekran alanını daha ekonomik kullanmak için form elementleri birleştirilebilir. Bu sistemde düğmeler ve giriş elementleri tek element olarak gösterilir (Resim 7.28). Giriş elementlerinin her iki tarafına da düğme eklenebilir. Düğmenin ekleneceği taraf kodların yazılış sırasına göre belirlenir. input etiketinden önce yazılan eklentiler sola, sonra yazılan eklentiler sağa eklenecektir.

```
<form>
  <div class="form-group">
    <div class="input-group">
      <input type="text" placeholder="Arama terimleri" class="form-control"></input>
      <span class="input-group-btn">
        <button class="btn btn-primary" type="button">
          <span class="glyphicon glyphicon-search"></span>
        </button>
      </span>
    </div>
  </div>
</form>
```

Resim 7.28 Bootstrap Element Grupları ile Oluşturulmuş Arama Formu

Element gruplarında düğmelerin yanında işaretleme kutuları ve radyo düğmeleri gibi elementler de kullanılabilir. Ek olarak bilgilendirme amaçlı karakterler ya da simgeler de eklenebilir. Bu durumda input-group-btn sınıfı yerine input-group-addon sınıfı kullanılır (Resim 7.29).

@

➤

```
<form>
  <div class="form-group">
    <div class="input-group">
      <span class="input-group-addon">@</span>
      <input type="text" placeholder="Kullanıcı adı" class="form-control"></input>
      <span class="input-group-btn">
        <button class="btn btn-info" type="button">
          <span class="glyphicon glyphicon-send"></span>
        </button>
      </span>
    </div>
  </div>
</form>
```

Resim 7.29 Bir Düğme ve Bilgilendirici Simge İçeren Element Grubu

Görsel Element Sınıfları

Görsel elementler en önemli tasarım elemanları arasında yer almaktadır. Bootstrap görsellerin sunumu ve boyutlandırılması için çeşitli olanaklar sunmaktadır. Bunların ilki görsellerin bulundukları ortam ve ekran boyutlarına göre boyutlandırılmasını sağlayan img-responsive sınıftır. Bisiklet tutkusu örneğinde (Resim 7.30) 1200x450 piksel boyutlarında oldukça yatay bir görsel kullanılmıştır. Resim 7.30'daki görseller xs tarayıcı boyutunda (700px) oluşturulmuştur.

Bisiklet tutkusu



En iyi rotalar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Bisiklet tutkusu



En iyi rotalar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
<h1>Bisiklet tutkusu</h1>

<h2>En iyi rotalar</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
```

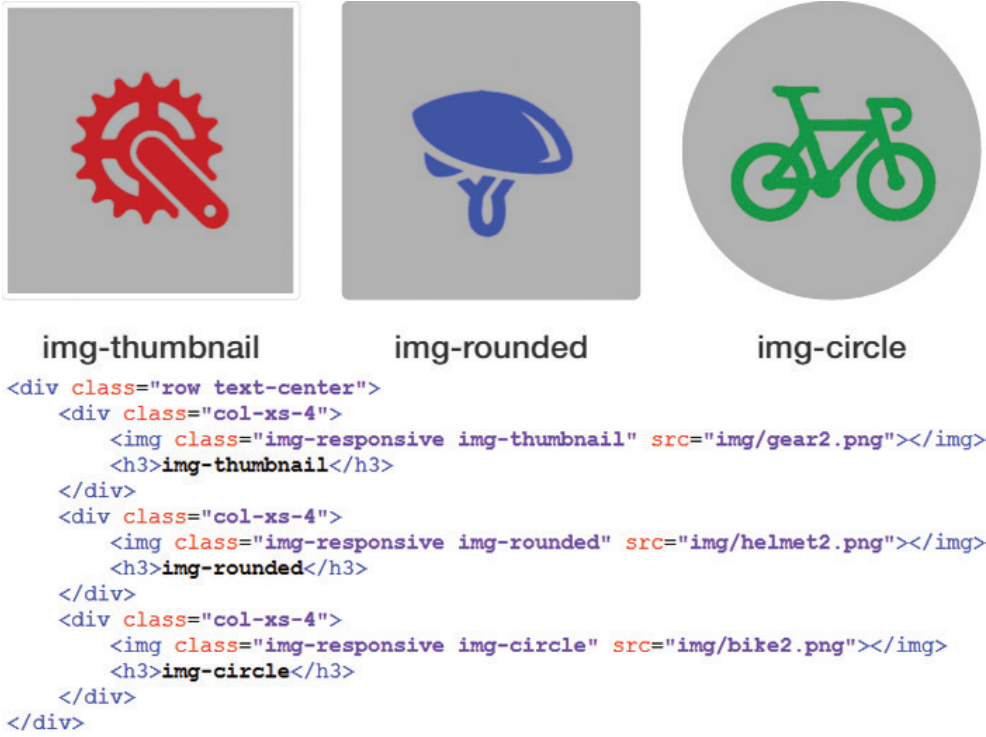
Resim 7.30 Img-Responsive Sınıfının Etkileri

Soldaki resimde img-responsive sınıfı kullanılmamıştır. Bu nedenle görsel sayfaya olduğu gibi yerleştirilmiş, tarayıcı boyutunu aşan kısımları görünmemiştir. Görselin yüksekliğinden herhangi bir kayıp yaşanmamıştır. Bununla birlikte sağdaki görselde img-responsive sınıfı kullanıldığından, görsel genişliği sayfaya uyacak şekilde daraltılmıştır. Bunun yanında yüksekliği de oranlı olacak şekilde azaltılmıştır. Bu

✓ img-responsive sınıfı görselleri block olarak göstermektedir. Bu nedenle bu sınıf kullanıldığında resmin ortalanması için center-block sınıfı kullanılmalıdır.

sayede hem görselin tamamı sayfada görülebilmüş, hem de en boy oranı sabit kaldığından görüntüsü bozulmamıştır. `img-responsive` sınıfı görsellerin bulundukları ortamın genişliğinde ölçeklenmesini sağlayan son derece faydalı bir sınıftır. Burada tüm sayfa genişliği kullanılmasına rağmen, ızgara sistemindeki küçük hücrelerde bile başarılı sunumlar sağlanabilmektedir.

Resimlerin sunumlarını düzenlemek için kullanılacak üç sınıf daha bulunmaktadır. Bunlar resimleri çerçeveler içine yerleştirmek için kullanılır. Aşağıda (Resim 7.31) bu sınıfların kullanımları ve etkileri gösterilmiştir.



Resim 7.31 Görsel Sunumlarını Düzenleyen Sınıflar

`img-thumbnail` sınıfı, görseli beyaz zeminli ve gri çerçeveli bir kutu içine alır. Özellikle artalan rengi olmayan ya da beyaz olan görsellerin sınırlarının belirtilmesi için kullanılacak bir sınıftır. `img-rounded` sınıfı görsellerin kenarlarında 6px yarıçaplı bir yumuşatma uygular. `img-circle` sınıfı ise görseli bir daire içinde sunmak için kullanılır. Özellikle ikonlar ve bireysel fotoğrafları sunmak için kullanılacak bir sınıftır.

Menü Sınıfları

Bootstrap, sitelerdeki yönlendirme işlemleri için kullanılacak menü seçenekleri sunmaktadır. Menüler ul elementlerine uygulanan sınıflar yoluyla oluşturulmaktadır. Bu işlemleri Bisiklet tutkusu adlı sitemiz için üç elementli basit bir menü yaparak örnekleyelim. HTML 5 ile birlikte sayfalardaki yönlendirme elementleri için nav etiketi getirilmiştir. Oluşturulacak tüm menülerin nav etiketlerine yerleştirilmesi semantik olarak iyi bir uygulamadır. Menü için öncelikle bir ul elementi oluşturulur. Bu elemente her bir menü elemanı için li etiketleri eklenir. Bu etiketlerin içine bağlantıları sağlayan a elementleri eklenir (Resim 7.32).


```
<nav>
  <ul>
    <li><a href="#">Bisiklet</a></li>
    <li><a href="#">Parkurlar</a></li>
    <li><a href="#">Ekipman</a></li>
  </ul>
</nav>
```

- Bisiklet
- Parkurlar
- Ekipman

Resim 7.32 HTML5 Temel Menü Şablonu

Bu durumda semantik olarak menü oluşturulmuş olsa da, görünüş olarak Bootstrap menüsü hâline getirmek için ul etiketine nav sınıfı eklenmelidir. Bu işlemle birlikte li etiketlerinin madde işaretleri kaldırılacak, menü elementleri arası boşluklar ayarlanacak ve her elementin üzerine gelindiğinde açık gri artalan ile vurgulama yapılacaktır. Bu menü Bootstrap sistemindeki en temel menüdür (Resim 7.33).

```
<nav>
  <ul class="nav">
    <li><a href="#">Bisiklet</a></li>
    <li><a href="#">Parkurlar</a></li>
    <li><a href="#">Ekipman</a></li>
  </ul>
</nav>
```

Bisiklet

Parkurlar

Ekipman

Resim 7.33 Dikey Bootstrap Menüsü

Kullanılabilecek ikinci menü, internet tarayıcılarında sayfalar arası geçişlere olanak sağlayan sekmelere benzeyen sekmeli menülerdir. Bu menüler belirli bir alan içinde tıklanan başlığın içeriğini gösterecek şekilde çalışmaktadır. Aktif menü elementi kenarlıklar yoluyla öne çıkarılmaktadır (Resim 7.34).

Bisiklet

Parkurlar

Ekipman

Parkurlar

```
<nav>
  <ul class="nav nav-tabs">
    <li class="active"><a data-toggle="tab" href="#bis">Bisiklet</a></li>
    <li><a data-toggle="tab" href="#par">Parkurlar</a></li>
    <li><a data-toggle="tab" href="#eki">Ekipman</a></li>
  </ul>
</nav>
<div class="tab-content">
  <div class="tab-pane active" id="bis">
    <h3>Bisiklet</h3>
  </div>
  <div class="tab-pane" id="par">
    <h3>Parkurlar</h3>
  </div>
  <div class="tab-pane" id="eki">
    <h3>Ekipmanlar</h3>
  </div>
</div>
```

Resim 7.34 Sekmeli Menü Örneği

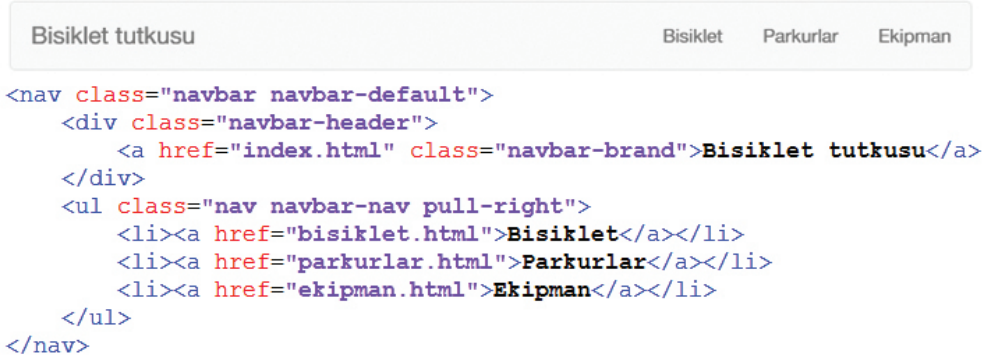
Bir sekmeli menü, sekmeler ve içerikler olmak üzere iki bölümden oluşur. Sekmeler bölümü temel menü kodlarına nav ve nav-tabs sınıflarının eklenmesi yoluyla oluşturulur. Bunun yanında her bir a elementine data-toggle="tab" ifadesi eklenir. Bu ekleme yapıldığında fare ile tıklandığında sekmelelerin değiştiği gözlemlenebilir. Ancak, içerik bölümü hazırlanıp bağlanmadan menü işlevsel olmayacaktır. Son olarak, sayfa açıldığında öne çıkarılması gereken sekmenin li etiketine "active" sınıfı eklenmelidir. Bundan sonra içerik bölümünün oluşturulmasına geçilir.

Sekme içerikleri div elementleri içine hazırlanır. Tüm içerik div'leri kapsayıcı bir div etiketi içine yerleştirilir. Kapsayıcı div etiketine "tab-content" sınıfı eklenir. İçeriklerin yerleştirileceği div etiketlerine ise tab-pane sınıfı eklenir. Sekmeler bölümünde olduğu gibi, sayfa açıldığında gösterilmesi istenen içeriğin div elementine active sınıfı eklenmelidir (Resim 7.34).

Bundan sonra sekme-içerik bağlantısının yapılmasına geçilir. İlk olarak içerik bölümündeki her div elementine bir id verilir. Bunlar içeriği işaret eden kısa ifadeler olarak belirlenmelidir.

Örneğimizde her içeriğin başlığının ilk üç harfi kullanılmıştır. Örneğin, Bisiklet sayfası için id özelliği "bis" olarak belirlenmiştir. Bu içerikler ve sekmeler arasındaki bağlantıların kurulması için, sekmelerdeki a elementlerinin href özelliklerine belirlenen id ifadeleri başlarına diyez "#" eklenerek yazılır. Örneğin Bisiklet sayfası için, bu sayfayı açması istenen a elementinin href özelliğine "#bis" yazılır. Bu düzenlemelerden sonra sekmeli menü işlevsel hâle gelecektir. Çalışma prensipleri sekmeli menü ile aynı olan düğmeli bir menü tasarımı da bulunmaktadır. Bu menünün oluşturulması için ul elementine nav-tabs sınıfı yerine nav-pills sınıfı eklenmelidir.

Kullanılabilecek bir diğer menü ise sitelerde ana menü olarak kullanılan menü çubuklarıdır. Menü çubukları sitenin istenen yerine yerleştirilebilir. İsteğe bağlı olarak sayfanın altında ya da üstünde sabit kalacak şekilde düzenlenebilir. Menü çubuğunda formlar, ikonlar, butonlar gibi pek çok element kullanılabilir. Bununla birlikte, sayfanın ana menüsü olan temel bir menü çubuğunda, sayfa bağlantıları ve ana sayfaya dönüş bağlantısı yer almalıdır (Resim 7.35).



Resim 7.35 Bootstrap Menü Çubuğu

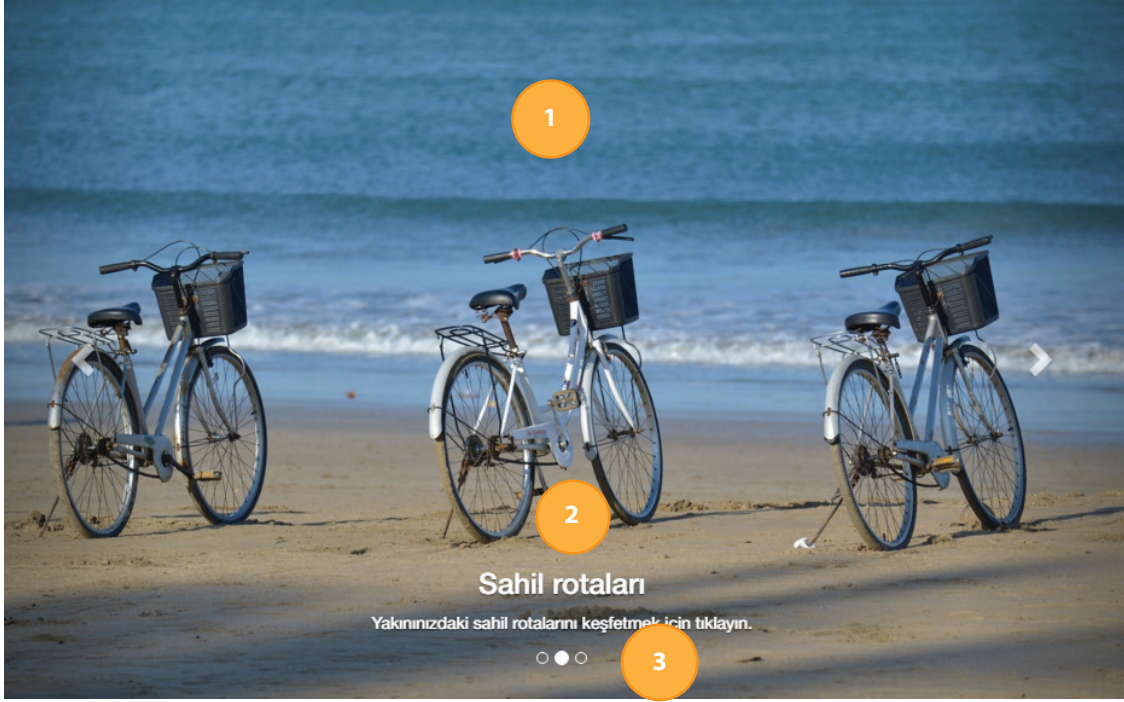
Menü çubukları nav elementleri içinde oluşturulmalıdır. nav elementine Bootstrap menü çubuğu görüntüsü kazanmak için "navbar" sınıfının yanında bir de tema sınıfı eklenir. Menü çubuğunda açık (navbar-default) ve koyu (navbar-inverse) olmak üzere iki tema kullanılabilir. Buradaki örnekte Bootstrap sisteminin varsayılan menü çubuğu teması olan navbar-default kullanılmıştır.

Menü çubuğu içinde kullanılacak ilk element site kimliğini belirleyen metin ya da ikonları tutan, başlık bölümü olarak düşünülebilecek "navbar-header" elementidir. Bu örnekte başlık bölümünde sadece sitemizin ismi yazılarak ana sayfaya yönlendirilmiş bir bağlantı içine eklenmiştir. Bu bağlantının menüdeki diğer elementlere oranla daha büyük yazılması için navbar-brand sınıfı kullanılmıştır.

Menü çubuğunda asıl menü bölümü ul elementi ile oluşturulur. Daha önceki menülerde olduğu gibi bu elemente öncelikle nav sınıfı eklenir. Bu sınıf eklendiğinde menü basit dikey menü (Resim 7.33) olarak işlemeye başlar. Yatay menü çubuğu görüntüsünü sağlamak için navbar-nav sınıfı eklenmelidir. Son olarak menüyü çubuğun sağına çekmek için pull-right sınıfı kullanılmıştır.

Kayan Resimler

İnternet sitelerinde marka imajları, reklamlar, duyurular, ürün tanıtımları gibi amaçlarla kullanılan, en önemli görsel araçlardan biri kayan resimlerdir. Bootstrap sisteminde kayan resim oluşturmak için Carousel bileşeni kullanılmaktadır (Resim 7.36). Kayan resim bileşeni dört bölümden oluşmaktadır. Hazırlanacak içeriğe göre kayan resim bileşeninin kodları olabildiğince uzun olabilmektedir. Bu bölümde üç resimden oluşan temel bir kayan resim elementinin oluşturulması anlatılacaktır.



Resim 7.36 Kayan Resim (Carousel) Elementi Örneği

Kayan resim elementi kapsayıcı bir div elementi içinde oluşturulur (Resim 7.37). Daha sonra oluşturulacak oklar ve madde işaretleri gibi kontrol elementleriyle kullanmak için bu div elementine bir id verilmelidir. Örneğimizde id olarak “resimler” belirlenmiştir. Kapsayıcı div elementine ilk olarak carousel sınıfı eklenir. İsteğe bağlı olarak, resimlerin kayarak geçişini sağlamak için slide sınıfı kullanılabilir. Slide sınıfının kullanılması zorunlu değildir, bu sınıf kullanılmadığında resimler kaymadan değişmeye devam edebilir. Son olarak div elementinin data-ride özelliği carousel olarak belirlenir. Bu sayede Bootstrap’in Javascript sınıfları elemente uygulanarak resimlerin otomatik değişimini sağlayacaktır.

Kayan resim elementinde temel olarak dört bileşen kullanılabilir. Bu elementlerin oluşturulma yöntemleri ve amaçları Resim 7.37’de sunulmuştur.

```

18 <div id="resimler" class="carousel slide" data-ride="carousel">
19
20 <div class="carousel-inner" role="listbox">
21 <div class="item active">
22 
23 </div>
24 <div class="item">
25 
26 <div class="carousel-caption">
27 <h3>Sahil rotaları</h3>
28 <p>Yakınınızdaki sahil rotalarını keşfetmek için tıklayın.</p>
29 </div>
30 </div>
31 <div class="item">
32 
33 </div>
34 </div>
35
36 <ol class="carousel-indicators">
37 <li data-target="#resimler" data-slide-to="0" class="active"></li>
38 <li data-target="#resimler" data-slide-to="1"></li>
39 <li data-target="#resimler" data-slide-to="2"></li>
40 </ol>
41
42 <a class="left carousel-control" href="#resimler" role="button" data-slide="prev">
43 <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
44 </a>
45 <a class="right carousel-control" href="#resimler" role="button" data-slide="next">
46 <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
47 </a>
48
49 </div>

```

Resim 7.37 Kayan Resim (Carousel) Elementinin Kodları

Kapsayıcı elementin içine ilk olarak kaydırılacak resimler eklenir. Tüm resimleri tutmak için kapsayıcı bir div elementi oluşturulur. Bu div elementine carousel-innersınıfı eklenir (20. satır). Bundan sonra her bir resim için bu tutucu elementin içine item sınıfı yeni div elementleri oluşturulur. Bu örneğimizde üç resim bulunduğu için 21, 24 ve 31. satırlarda item sınıfı div elementleri oluşturulmuştur. Bunun ardından div elementi içine img etiketi ile resim linkleri eklenir. Bu resimlerle ilgili en önemli nokta resim ebatlarının eşit tutulmasıdır. Kayan resim elementinin boyutları gösterilen resimlere göre ayarlandığından, farklı ebatlardaki resimlerde sayfada kaymalar yaşanabilmektedir. Örneğin, ilk resmin yüksekliği 500 piksel, ikinci resmin yüksekliği 600 piksel ise; ikinci resme geçişte kayan resim bileşeninin yüksekliği otomatik olarak 600 piksele çıkarılarak sayfanın kalan bölümü aşağı doğru kaydırılacaktır. Benzer şekilde, ilk resmin genişliği 1000 piksel, ikinci resmin genişliği ise 900 piksel olarak ayarlandığında; ilk resim sayfa genişliğini kapsarken ikinci resim sayfada ortalanacak, sağ ve solunda 50 piksellik beyaz boşluklar bırakılacaktır. Bu çerçevede, kayan resim bileşenine eklenecek resimler için 1200x500 gibi standart bir ebat belirlenmesi önerilen bir yaklaşımdır.

İsteğe bağlı her bir resmin üstüne açıklama metinleri, bağlantılar ve düğmeler eklenebilir. Bu elementlerin doğru yerleştirilmesi için item sınıfı div elementlerinin içine carousel-caption sınıfı div elementleri eklenir. Örneğimizde 26. satırda ikinci resim için açıklama metinleri eklenmiştir. Bu metinler Resim 7.36'daki 2 numaralı alanda gösterilmektedir.

İsteğe bağlı olarak resimler arası geçişleri sağlamak için madde işaretleri kullanılabilir. Madde işaretleri bölümü Resim 7.36'da 3 numaralı alanda görülmektedir. Madde işaretleri bölümü bir ol elementi içinde oluşturulur. Bu bölümün kodları Resim 7.37'de 36 – 40 satırlar arasında görülmektedir. Görsel düzenlemelerin sağlanması için ol elementine carousel-indicators sınıfı eklenmelidir. Bundan sonra her bir resim için ol elementine li etiketleri eklenir. Örneğimizde üç resim bulunduğu için, üç adet li etiketi oluşturulmuştur. Madde işaretlerinin yöneteceği kayan resim elementini belirtmek için data-target özelliği kullanılır. Bu yöntem pratik görünmese de sayfada birden fazla kayan resim elementine olabileceğini göz önünde tutunuz. Örneğimizde kayan resim elementine id olarak resimler verdiğimizden tüm li etiketlerinin data-

target özelliklerine #resimler belirteci eklenmiştir. Belirteç yazılırken hedef elementin id'sinin başına # eklendiğine dikkat ediniz. Bunun ardından madde işaretinin elementti kaçınıcı resme kaydıracağını belirten data-slide-to özellikleri doldurulur. Bu özellik genellikle sırayla gitmekle birlikte (ikinci madde işareti ikinci resmi gösterir), reklam kullanımı gibi bazı özel durumlarda resimler arasında atlama yapılabilir.

İsteğe bağlı olarak eklenebilecek son kontroller elementin sağ ve solda gösterilen ok kontrolleridir. Ok kontrolleri ile önceki ve sonraki resimlere kullanıcı denetiminde geçilebilir. Bu kontroller a elementleri içinde oluşturulur. Resim 7.37'de bu kontroller için yazılmış kodlar 42 ve 47. satırlar arasında gösterilmektedir. Bu amaçla sol ve sağ kontroller için birer adet a elementi oluşturulmalıdır. Bu elementlerin kontrol görünümünü kazanması için carousel-control sınıfı eklenir. Bunun

yanında, elementin yönünü belirtmek için left (sol) ve right (sağ) sınıfları eklenmelidir. Madde işaretlerinde olduğu gibi kontrollerin yöneteceği kayan resimler elementinin id'si başına # getirilerek href özelliklerine verilmelidir. Örneğimizdeki kayan resimler elementinin id özelliği resimler olduğundan, kontrollerin href özellikleri #resimler olarak belirlenmiştir. Kontrolün işlevsellik kazanması için data-slide özelliği kullanılmalıdır. Bu özellik yoluyla kontrolün tıklanması hâlinde hangi fonksiyonun çalışacağı belirtilmektedir. data-slide özelliği, soldaki kontrol için prev (önceki), sağdaki kontrol için de next (sonraki) olarak belirlenmelidir. Bura ya kadar gerçekleştirilen işlemler ile kayan resimler elementinin sağ ve soluna kontroller yerleştirilmiş olur. Kontroller üzerindeki okların gösterilmesi için a elementlerinin içine span elementleri eklenir. Burada, düğmeler başlığında anlatılan simge kullanım kuralları geçerlidir.

Öğrenme Çıktısı



- 4 Bootstrap projelerinin temel bileşenlerini açıklayabilme
- 5 Bootstrap projelerinde kullanılan elementleri açıklayabilme

Araştır 2

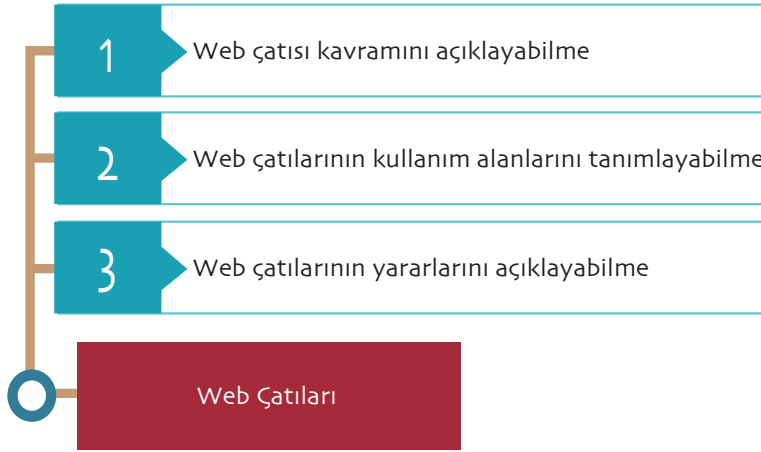
Bootstrap ile Twitter arasındaki ilişkiyi araştırınız.

İlişkilendir

Bootstrap ile yapılmış web sitelerine ulaşip inceleyiniz.

Anlat/Paylaş

Bootstrap dosyalarının projelere eklenmesi için kullanılacak iki yöntemi yarar ve sınırlılıkları bağlamında karşılaştırınız.



1 Web çatıları, internet siteleri ve servislerinin üretimini hızlandırmak için oluşturulmuş geliştirme yazılımlarıdır. Web çatılarının amacı internet sitelerinin ve servislerinin tasarım, üretim, onarım, bakım ve yenileme süreçlerinin hızlandırılmasıdır.

2 İnternet istemci-sunucu mimarisinde çalışan bir sistemdir. Bu çerçevede web çatıları sunucu ya da istemci tarafı olarak çalışabilmektedir. Sunucu tarafı çalışan web çatılarının görevi, sunucu kaynaklarının en etkin şekilde kullanılması ve istemlere en hızlı şekilde yanıt verilmesidir. Bu alanda kullanılabilecek çatılar arasında Django, “Zend Framework” ve “Ruby on Rails” sayılabilir. İstemci ortamı ise kullanıcıların bilgisayarlarındaki ya da mobil cihazlarındaki tarayıcılar ya da uygulamalar olabilir. Web çatılarının bu taraftaki rolü zengin internet uygulamaları oluşturmaktır. Bu uygulamaların içerikleri kullanıcı etkileşimine göre düzenlenerek kullanıcı deneyimi zenginleştirmeye çalışılır. Bu alanda kullanılabilecek araçlar arasında Bootstrap, Angular.js, Backbone.js gibi uygulamalar sayılabilir.

3 Bir projede web çatısı kullanmanın ilk yararı uzmanlar tarafından yazılan kodların tekrar kullanılmasıdır. Web çatıları sayesinde arayüz tasarımının gerçekleştirilmesi ya da veri tabanı bağlantısının en etkin biçimde oluşturulması gibi her projede gerekli, tekrarlanan adımların baştan yazılması gerekmez. Bu araçların oluşturulması için harcanacak programcı zamanı, projenin asıl sorunlarına aktararak etkin zaman yönetimi sağlanabilir. Bunun yanında, bu araçlar binlerce uygulamada kullanılmış olduğundan güvenilirlikleri ve performansları kanıtlanmıştır. Web çatıları açık kaynak felsefesi ile geliştirilen araçlar olduğundan ve ücretsiz dağıtıldığından geniş bir kullanıcı ağına sahiptir. Bu sayede kullanıcı forumları, yardım portalları gibi kanallardan yardıma erişilebilmektedir. Web çatılarında güncel yazılım geliştirme yaklaşımları kullanılmaktadır. Web çatısı kullanmak MVC gibi yaklaşımları kullanmayı gerektirdiğinden, projenizin güncelliği ve standartlara uyumluluğu otomatikman sağlanmış olur. Web çatıları geliştiriciler ve uygulayıcılar tarafından sürekli test edilmekte ve güncellenmektedir. Çoğunlukla geriye uyumlu yürütülen bu çalışmalar sonucunda, web çatısının yeni versiyonlarıyla gelen yeniliklerin projeye aktarılması sorunsuz bir şekilde gerçekleşebilecektir.

4

Bootstrap projelerinin temel bileşenlerini açıklayabilme

5

Bootstrap projelerinde kullanılan elementleri açıklayabilme

Güncel Bir Web Çatısı:
Bootstrap

4

Bootstrap HTML tabanlı bir web çatısıdır. HTML dilinin olanakları ve bu ortamda kullanılabilen teknolojilerden yararlanmaktadır. Bu çerçevede bir Bootstrap projesinin ilk temel bileşeni, projenin sunumu için kullanılan HTML dosyasıdır. Kullanıcı bilgisayarına bu dosya gönderilir. Projenin ikinci bileşeni elementlerin sunum biçimini belirleyen CSS dosyasıdır. Yazı tipleri, kenarlıklar, gölgelendirmeler gibi tüm etkiler CSS bileşeni sayesinde ayarlanır. Projenin üçüncü bileşeni Javascript dosyalarıdır. Bootstrap fonksiyonları JQuery gerektirdiğinden, projeye öncelikle JQuery. js dosyası eklenmelidir. Bunun ardından eklenecek Bootstrap.js dosyası menüler, kayan resimler gibi etkileşimli elementlerin çalışabilmesi için gereklidir. Son olarak, Bootstrap sistemiyle gelen simgelerin kullanılabilmesi için fonts klasörünün CSS klasörü ile aynı dizinde bulunması gerekmektedir.

5

Bu ünite öncelikle metin elementleri, listeler ve tablolar ele alınmıştır. Bootstrap sayesinde bu elementlerin tipografik özelliklerinin nasıl değiştirildiği gösterilmiştir. Kavramsal sınıflar kullanılarak hem görsel hem de semantik mesajların kullanıcıya nasıl iletileceği aktarılmıştır. Bootstrap projelerinin sunumlarını etkileyen en önemli araçlar boyut sistemi ve ızgara sistemidir. Bu sistemlerin birleştirilmesiyle tepkisel (responsive) web siteleri kolaylıkla oluşturulabilmektedir. Ünite ayrıca kullanılacak form elementleri ve bunların sunumlarının düzenlenmesi için kullanılacak araç sınıfları sunulmuştur. Sayfalara eklenecek görsellerin boyutlandırılması ve sunumlarını değiştirmek için kullanılacak görsel sınıfları örneklerle açıklanmıştır. Web sitelerinin en önemli bileşenleri olan yönlendirme elementleri sunulmuştur. Sekmeli ve düğmeli menülerin yanı sıra menü çubuğu oluşturmak için kullanılacak sınıflar aktarılmıştır. Son olarak, sitelerin en çok izlenen alanları olan kayan resim elementlerini oluşturmak ve düzenlemek için kullanılacak sınıflar anlatılmıştır.

Bu ünite yer kısıtlamaları nedeniyle tüm elementler ele alınamamıştır. Bunun yanında, kullanılacak elementlerin her gün güncellendiği ve yeni elementlerin sunulduğu unutulmamalıdır. Bu çerçevede, güncel elementleri izlemek için okuyucuların metin içinde verilen adresleri takip etmeleri önerilmektedir.

1 Aşağıdakilerden hangisi web çatılarının faydalarından biri **değildir**?

- A. Geniş yardım ağlarına ulaşabilmek
- B. Çalışırılığı kanıtlanmış araçları yeniden kullanabilmek
- C. Projelerde güncel tasarım yaklaşımlarını uygulayabilmek
- D. Proje üretme, düzenleme, yenileme maliyetlerini arttırabilmek
- E. Yeni versiyonlarla gelen iyileştirmelerden yararlanabilmek

2 Projelerin kontrolcü, model ve arayüzler yoluyla geliştirildiği yazılım geliştirme deseni aşağıdakilerden hangisidir?

- A. KMA
- B. JQuery
- C. Bootstrap
- D. MVC
- E. HTML

3 Aşağıdakilerden hangisi sunucu tarafında çalışan bir web çatısıdır?

- A. Django
- B. JQuery
- C. Bootstrap
- D. HTML
- E. AngularJS

4 Bootstrap ile ilgili olarak aşağıdaki ifadelerden hangisi doğrudur?

- A. Bootstrap tipografik özelliklere etki etmez.
- B. Bootstrap sadece mobil cihazlara yönelik bir web çatısıdır.
- C. Bootstrap istemci tarafında çalışan bir web çatısıdır.
- D. Bootstrap MVC yaklaşımıyla çalışır.
- E. Bootstrap projelerinde Bootstrap.js dosyası jquery.js dosyasından önce bağlanmalıdır.

5 Bir Bootstrap projesinde fonts klasörünün görevi aşağıdakilerden hangisidir?

- A. Projede kullanılan fontları tutar.
- B. Bootstrap ile gelen simgeleri tutar.
- C. CSS dosyalarını tutar.
- D. Javascript dosyalarını tutar.
- E. Proje kök dizinidir.

6 Yazılı materyallerin okunaklı, kavranabilir ve çekici hazırlanması ile ilgilenen bir sanat dalı aşağıdakilerden hangisidir?

- A. Tipografi
- B. Kaligrafi
- C. Grafik
- D. Görsel tasarım
- E. İkonografi

7 Bootstrap sisteminde $768px \leq x \leq 991px$ aralığını ifade etmek için hangi kısaltma kullanılır?

- A. xs
- B. md
- C. lg
- D. xl
- E. sm

8 900px genişliğindeki bir tarayıcıda col-sm-6 sınıfı ile işaretlenmiş bir sütunun genişliği ne kadar olur?

- A. 600px
- B. 450px
- C. 900px
- D. 300px
- E. 240px

9 Bir satırın içeriğinin tüm tarayıcı genişliklerinde iki sütun hâlinde gösterilmesi için bu sütunlara hangi sınıf eklenmelidir?

- A. col-xs-6
- B. col-xs-3
- C. col-sm-6
- D. col-lg-6
- E. Hepsi

10 Bir menü çubuğunu koyu renk teması ile sunmak için aşağıdaki sınıflardan hangisi kullanılır?

- A. inverse
- B. navbar
- C. navbar-inverse
- D. navbar-nav
- E. navbar-dark

1. D

Yanıtınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.

2. D

Yanıtınız yanlış ise “Web Çatılarının Kullanım Alanları” konusunu yeniden gözden geçiriniz.

3. A

Yanıtınız yanlış ise “Web Çatılarının Kullanım Alanları” konusunu yeniden gözden geçiriniz.

4. C

Yanıtınız yanlış ise “Güncel Bir Web Çatısı: Bootstrap” konusunu yeniden gözden geçiriniz.

5. B

Yanıtınız yanlış ise “Bir Bootstrap Projesinin Anatomisi” konusunu yeniden gözden geçiriniz.

6. A

Yanıtınız yanlış ise “Güncel Bir Web Çatısı: Bootstrap” konusunu yeniden gözden geçiriniz.

7. E

Yanıtınız yanlış ise “Bootstrap Boyut Sistemi” konusunu yeniden gözden geçiriniz.

8. B

Yanıtınız yanlış ise “Bootstrap Boyut Sistemi” konusunu yeniden gözden geçiriniz.

9. A

Yanıtınız yanlış ise “Bootstrap Izgara Sistemi” konusunu yeniden gözden geçiriniz.

10. C

Yanıtınız yanlış ise “Menü Sınıfları” konusunu yeniden gözden geçiriniz.

7

Araştır Yanıt Anahtarı

Araştır 1

Birçok yararıdan söz edilen web çatılarını kullanmak, beraberinde birtakım sınırlılıklar da getirebilmektedir. Örneğin, bir web çatısının beraberinde uzun kod yığınlarıyla geliyor olması, oluşturulan sistemin bir ölçüde daha yavaş çalışmasını sağlayabilir. Ayrıca geliştiricilerin web çatısı tarafından sağlanan belirli bir çerçevede çalışmaları, özel durumlara ilişkin esnek çözüm üretme süreçlerini zorlaştırmaktadır. Geliştiricilerin web çatısı tarafından sağlanan çalışma ortamına alışması da, temel geliştirme dil ve araçlarından uzaklaşmalarını sağlayabilir.

Araştır 2

Günümüzün en yaygın kullanıma sahip web çatılarından biri olan Bootstrap, en popüler sosyal ağlardan biri olan Twitter bünyesinde çalışan bir ekip tarafından geliştirilmeye başlanmış ve ücretsiz olarak diğer geliştiricilerin kullanımına sunulmuştur.

Kaynakça

- Components (n.d.). Bootstrap. <http://getbootstrap.com/components/> adresinden 26.09.2016 tarihinde erişilmiştir.
- CSS (n.d.). Bootstrap. <http://getbootstrap.com/css/> adresinden 21.09.2016 tarihinde erişilmiştir.
- Getting started. (n.d.). Bootstrap. <http://getbootstrap.com/getting-started> adresinden 23.09.2016 tarihinde erişilmiştir.
- I.T.U. (2015). ICT Facts and Figures, The World in 2015. <https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf> adresinden 03.07.2016 tarihinde erişilmiştir.
- Web framework. (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Web_framework adresinden 27.09.2016 tarihinde erişilmiştir.

Bölüm 8

Web Teknolojisinin Geleceği

öğrenme çıktıları

1 Web Teknolojisinin Başlangıcı ve Gelişimi
1 Web teknolojisinin ortaya çıkışını ve gelişim sürecini açıklayabilme

3 Ontolojik Web
3 Ontoloji oluşturma süreçlerini açıklayabilme

5 Mobil Uygulamalar
5 Mobil teknolojilerin web uygulamalarının gelişimi için önemini açıklayabilme

2 Semantik Web
2 Semantik web teknolojisini oluşturan temel özellikleri maddeleyebilme

4 Bulut Bilişim
4 Bulut bilişim teknolojisinin özelliklerini açıklayabilme

Anahtar Sözcükler: • Web • World Wide Web • İnternet • HTML • Semantik Web • Anlamsal Web
• Ontolojik Web • Bulut Bilişim • Mobil Web • Web'in Geleceği



GİRİŞ

1960'lı yıllar, günümüz dünyasını başta ekonomik, sosyolojik ve felsefi temelli olmak üzere pek çok alanda etkisi altına alan ve biçimlendiren küresel bir dönüşümün başlangıcına sahne olmuştur. Dünya tarihinde insanoğlunun var olduğu önceki bin yıllarda ortaya konan tüm bilgi birikimi, bilgi çağı olarak adlandırılacak bu yeni dönüşüm süreci ile gerek miktar olarak artmış gerekse çok daha kolay ulaşılabilir ve yayılabilir bir hâle gelmiştir. Bu dönüşümün baş aktörü, hiç kuşkusuz bugün tüm dünyada her yaştan, her sosyal statüden, her sosyoekonomik düzeyden, her eğitim seviyesinden, kısaca farklı arka planlara ve karakteristik özelliklere sahip milyarlarca bireyden oluşan bir kullanıcı kitlesine sahip "İnternet'tir". İnternet, ortaya çıkışının ardından çok kısa bir zaman dilimi içerisinde insan aklı ve yaratıcılığının sınırlarını zorlayacak biçimde bir gelişim göstermiştir. Bu sürecin başlangıcında "küresel bir nükleer savaş sırasında kesintiye uğramayacak bir iletişim ağı" olarak geliştirilen İnternet, sonraki on yıllarda sahip olduğu servisleriyle/hizmetleriyle (http, ftp, IRC, vs.) özellikle bilgiye dayalı çalışma ve üretim alanında günlük yaşamı yerel ve küresel bağlamlarda kolaylaştıracak dünya çapında bir ağa dönüşmüştür. İnternet'in gelişim süreci çerçevesinde bugün geline nokta bakıldığında söz konusu servislerin belki en önde geleninin World Wide Web ya da daha bilinen kısa adı ile "Web" teknolojisi olduğunu söylemek yanlış bir ifade olmayacaktır.

WEB TEKNOLOJİSİNİN BAŞLANGICI VE GELİŞİMİ

Web teknolojisi, İngiliz Profesör Tim Berners-Lee'nin 1989 yılında CERN (Conseil Européen pour la Recherche Nucléaire – Avrupa Nükleer Araştırma Merkezi) laboratuvarlarında HTML (Hyper Text Markup Language) adı verilen ve bugün hâlen Web tasarımında kullanılmaya devam eden işaretleme dilini geliştirmesi ile ortaya çıkmıştır. Berners-Lee, hemen ardından Web sayfalarının İnternet aracılığı ile görüntülenmesini sağlayan Web tarayıcısını da geliştirmiş ve günümüzde faaliyetlerine hâlen devam eden W3C (World Wide Web Consortium - Dünya Çapında Ağ Birliği) isimli organizasyonu kurmuştur (<http://tr.wikipedia.org>). Ortaya çıkışının ardından günümüze kadar 20-30 yıllık görece kısa sayılabilecek bir zaman diliminde Web nasıl bu denli yaygın ve popüler bir hâle gelmiştir? Nasıl bir gelişim sürecinden

geçmiş, tüm dünyadaki yaklaşık 8 milyar insan nüfusunun en az yarısının nasıl vazgeçilmezi olmuştur? Peki, Web teknolojisinin geleceği nasıl olacaktır? Bu soruların cevabı için Web'in mucidi Profesör Berners-Lee'nin şu sözü bir ipucu olabilir: "Hiç kimse bugünün teknoloji dünyasını beklemiyordu. Web size, çığırın fikirleri yapabilme olanağı tanır ve bunu yalnızca sizin kendi yaratıcılığınız sınırlandırabilir". Gerçekten de Web teknolojisinin, ilk ortaya çıktığı ve İnternet kullanıcıları ile ilk tanıştığı zamanlarındaki yalnızca yazı ve resimlerden meydana gelen statik ve etkileşimsiz yapısından; bugün Web 2.0 adı verilen, kullanıcıların da aynı zamanda içerik sağlayıcı olduğu, yazı ve resimlerin çok ötesinde çoklu ortam öğeleri içerebilen, veri tabanı bağlantılı dinamik ve etkileşimli bir yapıya dönüştüğü görülmektedir. Web'in teknolojik gelişimi bu hâli ile de sınırlı olmayıp anlamsal/semantik Web olarak tanımlanan bir tür "akıllı Web" uygulaması hâlen mühendislerin ve diğer paydaşların üzerinde çalıştığı gerçekleştirilebilir bir hedefi işaret etmektedir.



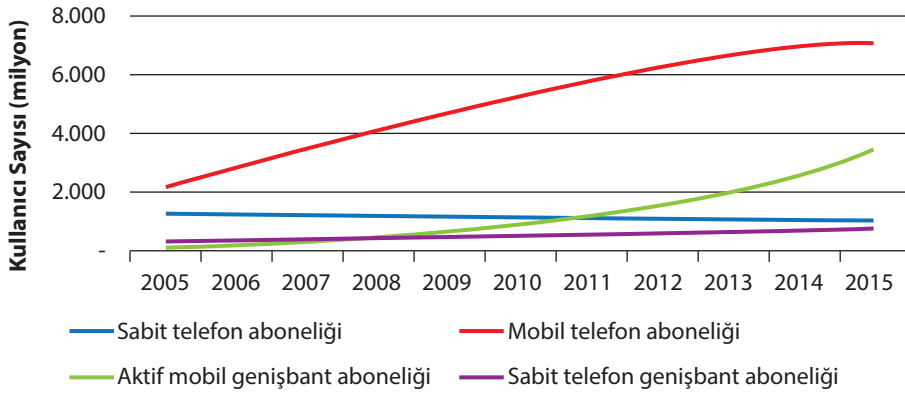
Web'in ortaya çıkış sürecini "The World Wide Web Consortium (W3C)" sitesinden daha ayrıntılı inceleyebilmek için "<http://www.w3.org>" adresine başvurabilirsiniz.

Genel olarak İnternet'in, özelde ise Web teknolojisinin gelmiş olduğu bu noktayı sayısal bir takım göstergeler ile ortaya koymak, bu teknolojinin geleceğine ışık tutabilmek açısından faydalı olabilir. İnternet Dünya İstatistikleri 2010 yılı verilerine göre Çin 420 milyon kullanıcı ile tüm dünyada İnternet kullanımı konusunda ilk sırada yer almaktadır. Yaklaşık 240 milyon kullanıcı ile ABD ve yine yaklaşık 100 milyon kullanıcı ile Japonya İnternet kullanıcı sayısı temelinde ikinci ve üçüncü sırada yer almaktadır. Türkiye 35 milyon İnternet kullanıcısı ile 2010 yılı verilerine göre on ikinci sıradadır. Türkiye İstatistik Kurumu'nun (TÜİK-<http://www.tuik.gov.tr>) gerçekleştirmiş olduğu Hane Halkı Bilişim Teknolojileri Kullanım Anketi ulusal düzeyde İnternet kullanım verileri ile bir anlamda bilgi ve iletişim teknolojilerinin toplumsal dönüşüme yansımalarını ortaya koymaktadır. Buna göre Türkiye'de İnternet'in en çok 16-24 yaş arası bireylerde yaygın olarak kullanıldığı (%62,9), bu duruma paralel olarak kullanıcı

cıların %89,5'inin öğrencilerden meydana geldiği görülmektedir. Öğrenim durumuna göre ise İnternet kullanıcılarının %89,6'sının yüksekokul, fakülte ve üzeri bir öğrenim düzeyine sahip olduğu ortaya konmakta.

Günümüzde Web teknolojisinin geldiği noktayı biraz daha somutlaştırabilmek açısından tüm dünyadaki mobil cihaz ve sabit telefon üzerinden geniş bant İnternet abone sayılarına göz atmak yararlı olabilir. Uluslararası Telekomünikasyon Birliği tarafından ortaya konan 2005-2015 yılları arasındaki veriler incelendiğinde mobil telefon aboneliğinin büyük bir ivme ile artmakta olduğu; buna karşın sabit hat aboneliklerinde ise düşük miktarda da olsa bir azalma olduğu görülmektedir. Şekil 1'deki geniş bant İnternet abone sayıları incelendiğinde ise mobil cihazlar üzerinden gerçekleştirilen iletişimin yalnızca bireyler arası sesli ve görüntülü görüşme amaçlı olmadığı; aynı zamanda zaman ver yerden bağımsız bir biçimde, iletişim gereksinimine ek olarak farklı uygulamalara erişimin amaçlandığı da açıkça görülmektedir. Doğaldır ki kullanıcıların sahip oldukları taşınabilir cihazlar bu karşılaştırmada mobil iletişim ve kablosuz ağ erişimini öne çıkaran bir faktördür. Ancak burada temel belirleyici faktör, aynı zamanda geniş bant internetin kullanımına neden olan, özellikle Web 2.0 teknolojilerinin temel oluşturduğu sosyal ağlar (Facebook, Twitter, vb.), web tabanlı çok kullanıcıli çevrimiçi oyunlar ve benzeri diğer etkileşimli teknolojilerdir. Buna ek olarak bölümün ilerleyen kısımlarında da değinileceği gibi Bulut Bilişim/programlama (Cloud Computing) gibi yine etkileşimli Web üzerinden hizmet veren teknolojiler World Wide Web için hiç düşmeyen bir ivme ile gelişime ve yaygınlaşmaya olanak vermektedir.

✓ Sabit telefonlar ve kablolu ağ yapılarına alternatif olarak, internet bağlantı hızının artması ve mobil teknolojilerin gelişimi ile birlikte Web 2.0 tabanlı etkileşimli uygulamaların özellikle genç kullanıcılar başta olmak üzere çok hızlı bir biçimde artış gösterdiği ve yaygınlaştığı görülmektedir.



Şekil 8.1 Sabit/mobil telefon kullanım ve internet bağlantı verileri

Kaynak: <http://www.itu.int>

Web'in bilgi çağına damgasını vuran söz konusu hızlı gelişim, daha önce de sorulmuş olan bir sorunun yeniden dile getirilmesini gerekli kılmaktadır: Web teknolojisi'nin geleceği nasıl olacaktır? En azından tahmin edilebilir geleceği... Bölümün bundan sonraki kısımlarında bu çerçevede öncelikle Web teknolojisi için şimdilerde ve yakın gelecekte öne çıkan Web 3.0 olarak nitelendirebileceğimiz Semantik Web veya bir diğer deyişle Anlamsal Web konusu ele alınacaktır. Bunun ardından Bulut Bilişim ve mobil Web konuları üzerinde durulacak ve Web teknolojisinin gelecekteki yapısına yönelik teknik ve sosyal yansımalar bağlamında bir bakış açısı oluşturulmaya çalışılacaktır.

Öğrenme Çıktısı



1 Web teknolojisinin ortaya çıkışını ve gelişim sürecini açıklayabilme

Araştır 1

Semantik Web'in hangi temel bileşenlerden oluştuğunu araştırınız.

İlişkilendir

Web teknolojilerinin kullanım yoğunluğunuzu, internet tarafından sunulan diğer teknolojileri kullanım yoğunluğunuzla karşılaştırınız.

Anlat/Paylaş

Web 2.0 teknolojisinin web kullanımını ne şekilde değiştirdiğini yorumlayınız.

SEMANTİK WEB

Geleceğin Web teknolojisi olarak ifade edilen semantik Web, “web içeriklerinin sadece doğal dillerde değil, aynı zamanda ilgili yazılımlar tarafından anlaşılabilir, yorumlanabilir ve kullanılabilir bir biçimde ifade edilebileceği, böylece söz konusu yazılımların veriyi kolayca bulmasını, paylaşmasını ve bilgiyi birleştirmesini sağlamayı amaçlayan, bir İnternet eklentisi” olarak tanımlanmaktadır. Esasen Fransızca olan semantik kelimesi, Türk Dil Kurumu Güncel Türkçe sözlüğünde, “anlam bilimi” ve “anlam bilimsel” olarak ifade edilmektedir. Semantik Web ile ilgili alan yazın incelendiğinde kavramın anlamsal web, anlamsal ağ, akıllı web, bilgili web ve Web 3.0 şeklinde adlandırıldığı da görülmektedir. Bu genel tanım çerçevesinde semantik Web'in daha önceki sürümleri olan statik Web ve Web 2.0 teknolojilerine ek olarak bir insanın hatırlama, kaydetme, yorumlama, öğrenme gibi düşünme biçimini ve/veya zihinsel süreçlerini belirli bir düzeyde de olsa taklit edebilecek bir yapıda olduğu anlaşılmaktadır.

Semantik Web, anahtar kelimelerden daha ziyade anlamsal içerik tarafından Web kaynaklarının erişimine olanak tanımaktadır. Bu kavram, “İnternet üzerindeki tüm bilgilerin ve bunların birbirleriyle ilişkilerinin yalnızca insanlar tarafından değil, makinelerce de anlaşılabilir; özellikle gereksinime odaklı bilgiye erişimin kolaylaştığı; amaca yönelik bağlamlarda bu veri ilişkilerinin rahatlıkla kullanılabilir olduğu bir ağ” biçiminde tanımlanmaktadır. Web servisleri olarak ifade edilebilecek olan kaynaklar, bilgisayar tarafından otomatik olarak anlaşılabilir ve çalıştırılabilir biçimdedirler.

World Wide Web Konsorsiyumu, semantik Web'in uygulamalar arasında ve topluluk içinde paylaşılabilir ve yeniden kullanılabilir ortak bir yapı sağladığını belirtmektedir.

Semantik Web'in ortaya çıkış amacı, işlevi veya kullanıcılar için sağladığı olanaklar nelerdir? Bir diğer deyişle önceki Web sürümlerine ek olarak semantik Web kullanıcılarına neyi vaat etmekte, hangi kolaylığı sağlamaktadır? Bu sorulara cevap olabilecek belki de en uygun kavram semantik Web'in “akıllı veya düşünebilen yapısı, ya da bir tür yapay zeka destekli Web olabilir. Statik Web 1.0 sürümü bir yana, etkileşimli Web olarak adlandırılan Web 2.0 teknolojisi dahi bugün ancak belirli bir oranda kullanıcı tercihleri ve taleplerini anlayabilmekte ve bunlar temelinde kullanıcı isteklerini yorumlayabilmektedir. Oysa semantik Web, sahip olduğu geniş veri tabanı, prosedürler, uygulamalar ve ontolojik yapısı ile kullanıcılar ile ilgili tüm parametreleri işleyebilir, hatırlayabilir ve anlamlandırabilir. Semantik Web, popüler olacağı günlerin belki de çok daha öncesinde Web'in mucidi olan Tim Berners-Lee tarafından şu şekilde dile getirilmiştir:

✓ Web teknolojisinin statik içeriğe sahip ilk sürümünün ardından etkileşimli Web 2.0 sürümünün ortaya çıkmıştır. Web 3.0 sürümü olan Semantik Web ise en genel biçimi ile bir tür yapay zeka ile desteklenen, akıllı, ve etkileşimli bir Web sürümü olarak düşünülebilir.

“Web için bir hayalim var, öyle ki bilgisayarlar web üzerindeki bütün veriyi, içerikler, linkler ve insanlarla bilgisayarlar arasındaki bütün işlemler gibi, analiz etmeye muktedir olacaklar. Henüz ortaya çıkmamış olsa da, ortaya çıktığı zaman anlamsal ağ ticaretin günlük mekanizmaları, bürokrasi ve günlük yaşamlarımız birbiri ile konuşan makineler tarafından yürütülecek. İnsanlığın asırlardır konuşup durduğu “akıllı ajanlar” nihayet gerçekleşecek.”



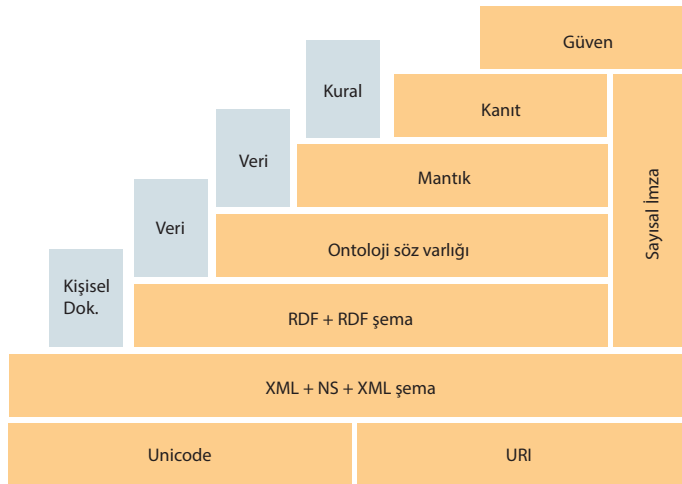
Tablo 8.1 Günümüzdeki Web teknolojisi ile Semantik Web teknolojisinin karşılaştırılması

Özellik	Günümüzdeki Web Teknolojisi	Semantik Web Teknolojisi
Mimari	Kullanıcı-Sunucu	Uçlar arası
Kullanıcı	İnsan	Makine
Veri	Yapılandırılmamış	Semantik yapılandırma
Arama	Söz dizim tabanlı	İçerik tabanlı
Arama Materyali	Doküman	Gerçek dünyada var olan tüm objeler
Dil	HTML	RDF, OWL, RSS
Linkler	Sade kavram	Kompleks kavram
İndeksleme	Kelime tabanlı	Kelime ve n-gram tabanlı
Veri Çağırma	Link tabanlı	Link tabanlı
Sınıflandırma	Bağlantı tabanlı (sayfa sınıflama)	Bağlantı tabanlı (Ontoloji sınıflama)
Sorgu Genişliği	Kullanıcı arama davranışına göre	Ontoloji tabanlı

Kaynak: Mahmoudi ve diğ., 2009; aktr: Beden, 2012

Semantik Web'in işleyişinde verinin anlamının ve doğal dil aramalarının yazılım tarafından anlamlandırılabilirdiği bir teknoloji söz konusudur. Bu durum, semantik web mimarisi içerisinde kullanıcı profillerinin, deneyimlerinin, reel uygulamalarda işe koşulan birtakım üst verilerin (metadata), arama stratejilerinin ve anahtar kelimelerin bir veri tabanında bulunmasını gerektirmektedir. Esasen başlı başına bir veri tabanının varlığı da semantik Web mimarisini tamamlamak adına yeterli olmayacaktır. Bu bağlamda mimariyi oluşturan belirli söz dizim kurallarına (XML), semantik Web yapısının omurgasını oluşturacak olan ve ağ kapsamındaki bilgi yapısının işlenmesini sağlayacak ontolojinin geliştirilmesine gereksinim duyulmaktadır. Temel bileşenlerini XML, Web servisleri ve ontoloji olarak ortaya koyduğu semantik Web işleyişinin, yalnızca Web sayfalarında yer alan bilgilerin kullanıldığı bir insan-bilgisayar ilişkisi olmadığı, verinin yer aldığı veri tabanı, servisler, programlar, vb. uygulamalara ait verilerin de işlenebildiği karmaşık bir ortam olarak tanımlanmaktadır. Şekil 8.2'de Tim Berners-Lee tarafından semantik Web için önerilen bir mimari sunulmaktadır.

✓ Semantik Web'i meydana getiren en önemli temel bileşenler Web servisleri, XML ve ontoloji olarak ifade edilebilir.



Şekil 8.2 Tim Berners Lee tarafından önerilen semantik web mimarisi

Kaynak: Çakır, 2013

XML

Geleceğin Web teknolojisi olarak, Semantik Web için en önemli bileşenlerden birisi XML (Extensible Markup Language - Genişletilebilir İşaretleme Dili) adı verilen bir işaretleme dilidir. XML, insanlar ve bilgi işlem sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan bir işaretleme dili ve aynı zamanda W3C tarafından tanımlanmış bir standarttır. Bu özelliği ile XML, veri saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara format görevi de görmektedir. XML sayesinde birbirinden çok farklı verilerin bir araya getirilerek bir etiket sistemi sayesinde orijinal formatları bozulmadan, tek bir havuzda tutulduğu; böylece verilerin hızlı, kolay ve ortamdan bağımsız bir biçimde erişilebilir hâle geldiği söylenmektedir. XML işaretleme dilinin genel karakteristiği aşağıdaki gibi maddelenebilir.

- XML yapılandırılmış belge ve verilerin evrensel formatıdır.
- XML metin tabanlı işaretleme dilidir ve veri alış verişinde kullanılan bir standart'tır.
- XML bilginin yapısını tanımlamak için kullanılan bir teknolojidir.
- XML bilgiyi tanımlayan ve Web'de bilgi alış verişi için kullanılan standart bir biçimdir.
- XML işaretleme dillerini tanımlayan bir meta dilidir.
- XML verinin yapılandırılması ve tanımlanması için kullanılan bir teknolojidir.
- XML herhangi bir verinin biçimlenmesi, tanımlanması için kullanılan bir teknolojidir.



dikkat

XML, yalnızca HTML benzeri bir işaretleme dili değildir. XML aynı zamanda bilgi yapısını tanımlayan ve standardize eden bir meta dildir.

Web tasarımında yaygın ve temel olarak kullanılan HTML'den (Hyper Text Markup Language – Hyper Metin İşaretleme Dili) farklı bir biçimde XML işaretleme dili aynı zamanda verinin içeriğini de tanımlamaktadır. XML işaretleme dili İnternet ortamında, önemlileri aşağıda maddelenmiş olan çeşitli amaçlarla kullanılabilir.

- İnternet arama motorları
- Aygıt ve uygulamadan bağımsız veri erişimi
- Elektronik veri değişimi ve elektronik ticaret
- Basın ve yayın
- Şirket uygulamaları standardizasyonu
- Yazılım geliştirme

XML işaretleme dili ile oluşturulmuş olan belgeler yazım biçimi olarak HTML ile benzerlik gösterse de mantıksal olarak farklı amaca hizmet etmekte ve üst veri olarak nitelendirilebilmektedir. Genel olarak belirli bir ağaç yapısı içerisinde tanımlanan verilerden oluşan XML belgeleri içerisinde yer alan veriler şemalar veya tanımlama dosyalarında yer almaktadır. Bu bilgiler çerçevesinde örnek bir XML belge içeriği şöyle Şekil 8.3'teki gibi örneklendirilebilir.

```

1 <kullanici>
2   <kullanici id="1">
3     <ad>A</ad>
4     <soyad>B</soyad>
5   </kullanici>
6   <kullanici id="2">
7     <ad>C</ad>
8     <soyad>D</soyad>
9   </kullanici>
10  <kullanici id="5">
11    <ad>E</ad>
12    <soyad>F</soyad>
13  </kullanici>
14  <kullanici id="8">
15    <ad>G</ad>
16    <soyad>H</soyad>
17  </kullanici>
18 </kullanici>

```

Şekil 8.3 Örnek bir XML belge içeriği

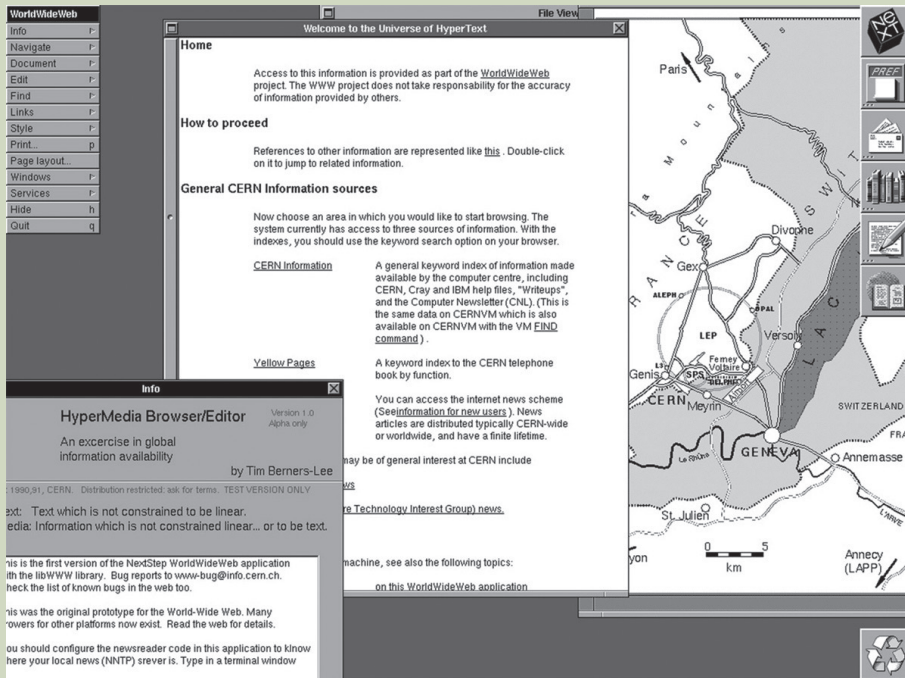


yaşamla ilişkilendir

Dünyanın ilk web sitesi 25 yaşında

22 Aralık 2015 - 10:33 Son Güncelleme: 22 Aralık 2015 - 10:35

Hürriyet



Dünya'nın ilk web sitesi, 25 yıl önce bugün açılmıştı! İşte o site ve hikayesi...

İlk web sitesi, yani Tim Berners-Lee'nin World Wide Web'i, 25 yıl önce bugün açılmıştı.

Popüler sayfa, CERN'de 20 Aralık 1990'da yayınlanmaya başlandı ancak gerçek anlamda halka açılması, Ağustos 1991'de gerçekleşti. Sayfa, hypertext tabanlı WWW projesinin nasıl çalıştığını anlatmaktan fazlasını yapmıyordu; ancak Berners-Lee'nin sayfasının bugünkü internetin altyapısını oluşturduğunu rahatça söyleyebiliriz. Bugün bile web'in internetle aynı şey olduğunu düşünen birçok kişiye rastlamanız mümkün.

Peki web'in mucidi bugün ne yapıyor? Berners-Lee web ile hâlen yakından ilgili ve kurulmasına yardımcı olduğu World Wide Web Consortium'u yönetiyor. Berners-Lee'nin web'i devletlerin sansür girişimlerinden korumaya ve iletişim firmalarının ağ eşitliğini bozma girişimlerini engellemeye yönelik önemli çalışmaları bulunuyor. Bununla birlikte CERN'in web'deki rolü bugün biraz değişti. Ağ araştırmalarına devam etse de CERN bugün daha çok parçacıkları çarpıştırmakla ilgileniyor.

Web, bilgi paylaşımı aracı olmayı bugün de sürdürüyor ve muhtemelen bu, yakın bir gelecekte değişmeyecek.

Kaynak: Hurriyet Teknoloji,

<http://www.hurriyet.com.tr/dunyanin-ilk-web-sitesi-25-yasinda-40030155>, Erişim Tarihi: 10.08.2016.

Öğrenme Çıktısı



2 Semantik web teknolojisini oluşturan temel özellikleri maddeleyebilme

Araştır 2

“Sayısal imza” kavramını araştırınız.

İlişkilendir

Kullanıcılarının alışkanlıklarına ve tercihlerine göre özelleşen web sayfalarının nasıl bir deneyim sağlayabileceğini yorumlayınız.

Anlat/Paylaş

Tim Berners Lee tarafından önerilen semantik web mimarisini açıklayınız.

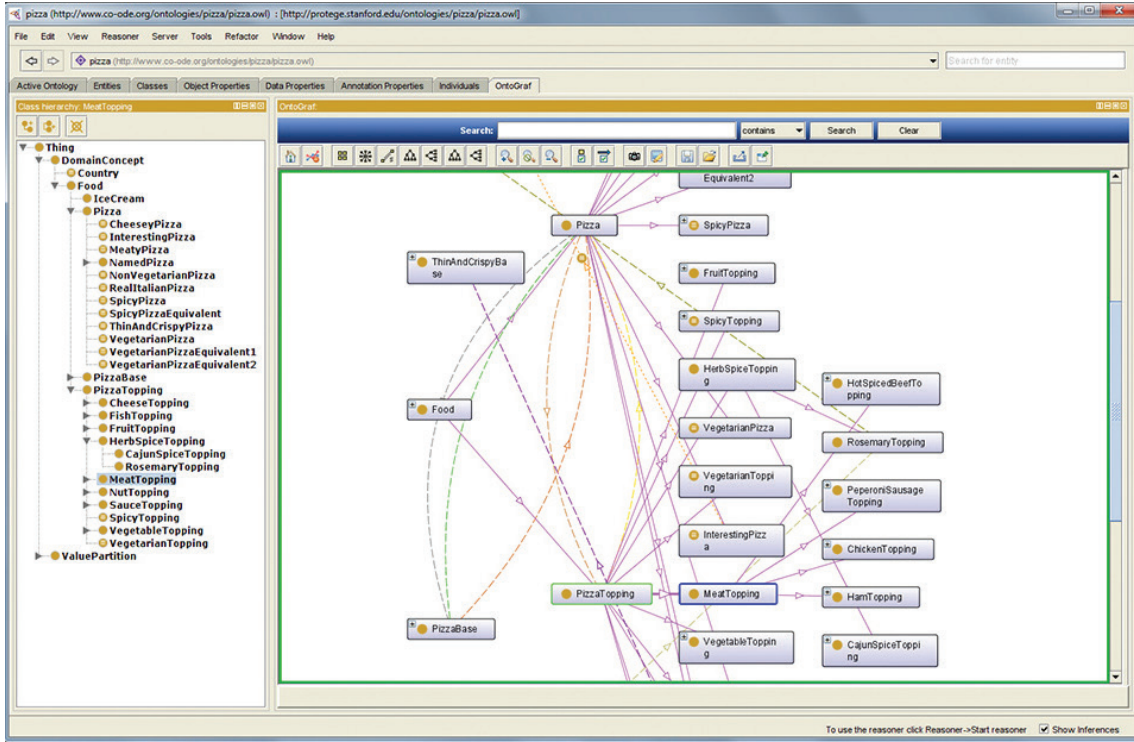
ONTOLOJİK WEB

Kelime anlamı olarak “Varlık bilimi” şeklinde tanımlanan ontoloji kavramı, semantik web uygulamalarının en önemli bileşeni sayılabilir. Ontolojiler, “bir alana ait kavramlar kümesini ve kavramlar arasındaki ilişkilerin biçimsel olarak tanımlanması” şeklinde ifade edilmektedir. Semantik web uygulamalarında farklı ağ kullanıcıları tarafından sağlanan çeşitli bağlamlardaki içeriğin, akıllı web olarak da adlandırılan söz konusu ortam tarafından anlaşılabilirliğini sağlayıp, kullanıcının isteği doğrultusunda cevap oluşturabilmek için anlamları ortak hâle getirebilecek bir dil yapısına gereksinim duyulmaktadır. Bu noktada ontolojiler, varlıklar için ortak tanımlamalar olarak karşımıza çıkmakta ve semantik Web ortamında farklı terimleri açıklamak için ontolojilere gereksinim duyulmaktadır. Benzer biçimde semantik Web için ontoloji “bilgi koleksiyonu” ve “kavramlar arasındaki ilişkileri biçimsel olarak ifade eden bir doküman” biçiminde de tanımlanabilmektedir.

Yayınlayıcı için bulunabilirlik, arama yapan için ise doğru bilgiye hızlı erişime olanak tanıyan bir çözüm olarak ifade edilen semantik web teknolojisi, içerikler arası ilişkilendirmeler sayesinde her bir web sayfasının başka sayfalarla ve insanlarla konuşabilen bir yardımcı rolünü üstlenmektedir. Öte taraftan, semantik web için her ne kadar “akıllı” veya “zeki” web kavramları kullanılıyor olsa da, büyük olasılıkla farklı amaçlarla ve çok fazla varyasyona sahip kullanıcı girdilerini sistemin anlayabileceği, bir diğer deyişle işleyebileceği hâle getirmek, ancak belirli ölçüde standartlaştırma ile olanaklıdır. Bu durumda pratikteki uygulamalar ve alanyazın ele alındığında ontolojiler için bu standartlaştırmanın veri girişinin standartlaştırılarak giderilmeye çalışıldığı görülmektedir. Bu kapsamda ontolojiler genellikle belirli bir konu alanlarına yönelik oluşturulmakta ve bilgisayarlar tarafından işlenebilecek biçimde sunulmaktadır.

Ontolojiler, kısaca OWL (Web Ontoloji Dili - Web Ontology Language) ve RDF (Kaynak Tanımlama Çerçevesi - Resource Description Framework) adı verilen ontoloji geliştirme dilleri yardımı ile oluşturulmaktadır. OWL ile bir ontoloji oluşturulması öncesinde kavram haritalarına benzer biçimde ontoloji içerisindeki nesneler/terimler tanımlanır. Bu terimlerin belirli veya sonlu sayıda olması gerekir. Ayrıca seçilen alanı oluşturan ve ontolojiyi meydana getirecek olan terimlerin birbiriyle olan hiyerarşileri ve ilişkileri ortaya konur. Ontoloji geliştirmek üzere yazılımcılar çeşitli araçlardan yararlanabilmektedir. Bunlardan en sık kullanılanlarından birisi <http://protege.stanford.edu> bağlantısı üzerinden indirilebilecek Protege adlı yazılımdır (Şekil 8.4).

✓ Ontolojiler, semantik web ortamında farklı tür ve özellikteki verilerin sistem tarafından anlaşılmasını ve anlamlandırılmasını sağlayarak kullanıcıların isteklerine cevap oluşturmayı amaçlamaktadır.



Şekil 8.4 Protege adlı yazılımın arayüzüne ilişkin bir örnek

Kaynak: <http://protege.stanford.edu>

Öğrenme Çıktısı

3 Ontoloji oluşturma süreçlerini açıklayabilme

Araştır 3

Ontoloji kavramının felsefi kökenini araştırınız

İlişkilendir

Öğrenme süreçlerinde kavram haritalarının kullanımının yarar ve sınırlılıklarını yorumlayınız.

Anlat/Paylaş

“Akıllı web” ifadesini yorumlayınız.

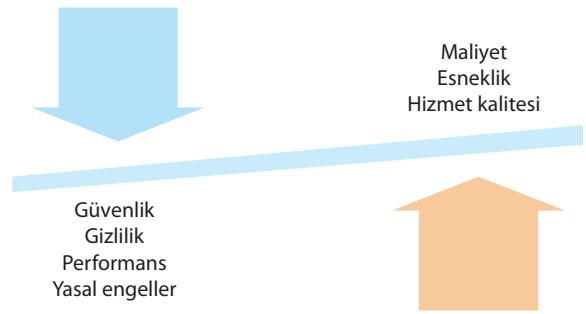
BULUT BİLİŞİM

Bulut bilişim (Cloud Computing), İnternet ve ağ teknolojileri kapsamında en güncel ve ilgi çeken konulardan biri olarak karşımıza çıkmaktadır. Özellikle taşınabilir bilgisayarlar, akıllı telefonlar gibi mobil teknolojiler ve aynı zamanda kablosuz iletişim olanaklarının yaygınlaşması ile çok daha geniş bir kullanıcı kitlesi tarafından bilinir ve kullanılır bir kavram olmuştur. Bu denli yaygın bir kullanım alanı bulan ve her düzey kullanıcı için İnternet ortamında çeşitli ve kullanışlı çözümler vaat eden bulut bilişim alanyazında, “web servisleri aracılığıyla internet üzerinde veri depolayabilen ve aynı zamanda ortak bilgi paylaşımına izin veren oluşum” şeklinde tanımlanmaktadır.

Bulut bilişim teknolojisi temel olarak bilgisayar, akıllı telefon, vb. birimlerden oluşan fiziksel bir yapı olmayıp, özünde bilgi ve iletişim teknolojilerinin bir nevi sanal kullanımına dayalı bir hizmet yapısı olarak anlaşılmalıdır. Gündelik yaşamda kullanıcıların, bilgisayar ortamında kullanmak üzere veri taşıma amacıyla edindikleri USB belleklerin veya taşınabilir sabit disklerin yerine örneğin, Google Drive, Dropbox gibi çok çeşitli bulut bilişim uygulamaları, her düzeyde İnternet kullanıcısı tarafından veri kaydetme ve taşıma amaçlı aktif olarak kullanılan bulut bilişim uygulamaları arasında yer almaktadır. Bu teknoloji sayesinde, kurumların bilgi teknolojileri altyapılarında gerekli olan sunucu, bilgi depolama ünitesi ve yedekleme sistemleri gibi fiziki ortam ve personel için yatırım yapmak yerine, bu altyapıyı kurmuş olan firmalardan kiralama sistemine dayalı hizmet satın alması olanaklı hâle gelmiştir.

Bulut bilişim teknolojisi, IaaS, PaaS ve SaaS adı verilen üç temel hizmet modeline dayalı olarak çalışmaktadır (Şekil 8.5). Servis olarak yazılım modelinde (Software as a Services, SaaS), kullanıcılar kendi bilgisayarlarına herhangi kurulum yapmaksızın, internet aracılığı ile bulut ortamında yer alan yazılımları kullanabilmektedir. Microsoft Office yazılım paketinin bulut bilişim sürümleri buna örnek olarak verilebilir. Eğer kullanıcıların bulut ortamında hazır bir yazılım kullanmak yerine kendi yazılımlarını geliştirmek gibi bir niyetleri var ise, bu kez kullanılacak model, Servis olarak platform modeli (Platform as a Service, PaaS) olacaktır. Google Drive örneğinde açıklamaya çalıştığımız gibi servis olarak altyapı modeli (Infrastructure as a Service, IaaS) kapsamında ise kullanıcılara veri depolama hizmeti gibi bulut hizmetleri sunulabilmektedir.

Bulut bilişim uygulamaları gerek bireysel gerekse kurumsal anlamda kullanıcılarına çeşitli avantajlar sağlamaktadır. Özellikle büyük ölçekli organizasyonlar için maliyet en önemli avantaj ve dezavantaj olarak öne çıkmaktadır. Fiziksel donanım ve bu donanımın yer alacağı ortam, bakım, güncelleme, barındırma vb. farklı modeller kapsamında ele alınabilecek hizmetlerin bulut bilişim hizmeti biçiminde sağlanması maliyeti düşüren ve optimize eden bir boyuttur. Diğer bazı avantajlar esneklik ve hizmet kalitesi olarak ortaya konmaktadır. Bulut bilişim hizmetlerine, fiziksel bir takım donanım ve hizmetlerin sınırlayıcı faktörlerinden kaçınarak özellikle İnternet ve intranet erişimleri sayesinde zaman ve yerden bağımsız biçimde erişim olanaklıdır. Bulut bilişim teknolojilerine ilişkin tüm bu avantajların yanı sıra bir takım sınırlılıkların söz konusu olduğu da ifade edilmektedir. Bulut bilişim teknolojisinin en fazla üzerinde durulabilecek sınırlılıkları arasında güvenlik ve gizlilik yer almaktadır. Buna ek olarak sistem performansının düşük olması ve bazı yasal engeller de sınırlılıklar arasında gösterilmektedir.



Şekil 8.5 Bulut bilişim teknolojisinin avantaj ve sınırlılıkları

✓ Bulut bilişim, sanal depolama gibi servisleri, sanal yazılım kullanımı, kurulumu ve kullanımını içerisinde barındıran bir teknoloji örüntüsünü temsil etmektedir.

Öğrenme Çıktısı

4 Bulut bilişim teknolojisinin özelliklerini açıklayabilme

Araştır 4

Bulut bilişim olanaklarıyla web sunucusu hizmeti sağlayan önemli servis sağlayıcılarını araştırınız.

İlişkilendir

Google Drive ve Dropbox gibi hizmetlerin bulut bilişim olanaklarından nasıl yararlandığını yorumlayınız.

Anlat/Paylaş

Bulut bilişim tarafından sağlanan olanakları tartışınız.

MOBİL UYGULAMALAR

Mobil teknolojilerin her geçen gün hızla yaygınlaşması ile birlikte Web tabanlı olanlar da dahil olmak üzere bilgisayar ortamında kullandığımız neredeyse tüm uygulamalar mobil ortama taşınmaktadır. Mobil teknolojilerin bu denli hızlı gelişimi ve yaygınlaşması ile birlikte bireyler arası iletişim sabit bilgisayarlardan ve kablolu ağ yapılarından; taşınabilir aygıtlar ve kablosuz ağlar aracılığı ile gerçekleşen mobil iletişim boyutuna taşınmıştır. Bu hâliyle Web tabanlı uygulamalar da dahil olmak üzere bireylere sağlanan bilgiye erişim ve iletişim olanakları gerçek anlamda zaman ve yerden bağımsız bir biçime dönüşmüştür. Bu çerçevede bireylerin artık kablosuz ağ erişim olanağına sahip akıllı telefon, tablet PC vb. taşınabilir cihazlar yardımı ile Web sitelerinde gezinebildikleri, sosyal ağlar aracılığı ile paylaşımlarda bulunabildikleri, bankacılık, alışveriş gibi ticari işlemler yapabildikleri, özetle ceplerine sığdırbildikleri küçük bir araç yardımı ile tüm dünyaya erişebildikleri mobil internet ve mobil uygulamalar çağına girilmiştir. Mobil teknolojilerin kullanımına yönelik gerçekleştirilen çeşitli araştırmalar, bu yüksek ivmeli gelişimin son kullanıcıya yansımalarını ve mobil teknolojilerin gelişimini veriye dayalı bir biçimde ortaya koymaktadır. Buna göre dünyada 5 milyardan fazla mobil telefon kullanımı olduğu ve bu sayının kişisel bilgisayar kullanımının kat kat üzerinde olduğu belirtilmektedir. Yapılan kapsamlı araştırmaların sonuçlarına göre:

✓ Mobil teknolojiler ve kablosuz internet erişimi alanlarındaki gelişmeler dünya üzerinde çok büyük bir hedef kitlenin sabit bilgisayarlar ve kablolu ağ erişiminden mobil ortama geçişine neden olmuştur.

- Kullanıcıların %76'sı akıllı telefonları olmaksızın evlerinden çıkmamaktadırlar.
- Araştırmanın gerçekleştirildiği tarih itibarıyla son altı ayda İnternet erişimini akıllı telefonları üzerinden sağlayan kullanıcılar toplam örneklemin %43'ünü oluşturmaktadır.
- Kullanıcıların %42'si akıllı telefon kullanmayı bırakmak yerine televizyon izlemeyi bırakmayı tercih etmektedir.
- Akıllı telefon kullanımının %91'i eğlence, %84'ü iletişim ve %68'i bilgi edinme amaçlı gerçekleşmektedir.
- Kullanıcıların %88'i akıllı telefonlar ile video izlemektedirler.
- Kullanıcıların %96'sı akıllı telefonları ile sosyal ağları ziyaret etmekte; bunun %75'i günde en az bir kez gerçekleşmektedir.

Mobil tabanlı uygulamaların, taşınabilir olmayan bilişim sistemlerine göre en önemli üstünlüğünün, kullanıcılar için gerçek anlamda "zaman ve yer-

den bağımsız” olabilmeye olanak sağlaması olduğu ifade edilebilir. Ayrıca, mobil cihazların herkes tarafından satın alınabilir bir fiyat aralığında olması, kolay kullanım özellikleri, belki de çok daha önemlisi her düzeyde kullanıcı için sürekli iletişim hâlinde olabilme ve günlük hayatın her alanında fayda sağlayacak mobil çözümler sunması gibi durumlar, mobil uygulamaları her geçen gün daha da cazip ve popüler bir hâle getirmektedir. Bu durumla eş zamanlı bir biçimde mobil Web ve uygulamalarının gelecek nesillerin temel bilişim uygulamaları olabileceğini var saymak olanaklıdır. Bununla birlikte her teknolojinin kendine özgü üstünlüklerinin olmasının yanı sıra mobil Web ve genel olarak mobil uygulamaların bazı sınırlılıklarında da bahsetmek yararlı olacaktır. Mobil Web uygulamalarının üzerinde çalıştığı taşınabilir cihazlar hâlen boyut olarak küçük, kablolulu ağ teknolojilerine göre nispeten daha düşük bant genişliğine sahip ve işlemci vb. donanım faktörlerine bağlı olarak çoklu görev desteği olmaması gibi nedenlerle önemli sayılabilecek sınırlılıklara sahiptir. Çoğu mobil Web uygulaması içerisinde yine görece de olsa navigasyon (dolaşım) konusunda yaşanan sorunlar ve çok farklı özelliklerdeki cihazlara göre standardize edilmiş Web uygulamalarının geliştirilmesindeki sıkıntılar diğer temel sınırlılıklar arasında sayılabilmektedir. Doğaldır ki, bu sınırlılıklar diğer teknolojilerde olduğu gibi bilimsel gelişmeler ile ortadan kalkabilmektedir.



Öğrenme Çıktısı

5 Mobil teknolojilerin web uygulamalarının gelişimi için önemini açıklayabilme

Araştır 5

Mobil Web uygulamalarının, geleneksel masaüstü uygulamalarından ayrıldığı noktaların neler olabileceğini araştırınız.

İlişkilendir

Akıllı telefon kullanıcılarının telefon kullanımlarının ne kadarlık bir kısmını web gezintisi için ayırıyor olabileceklerini yorumlayınız.

Anlat/Paylaş

Cep telefonlarında bulunan GPS teknolojilerinin web kullanımı açısından nasıl bir potansiyel taşıyor olduğunu tartışınız.

1

Web teknolojisinin ortaya çıkışını ve gelişim sürecini açıklayabilme

Web Teknolojisinin Başlangıcı ve Gelişimi

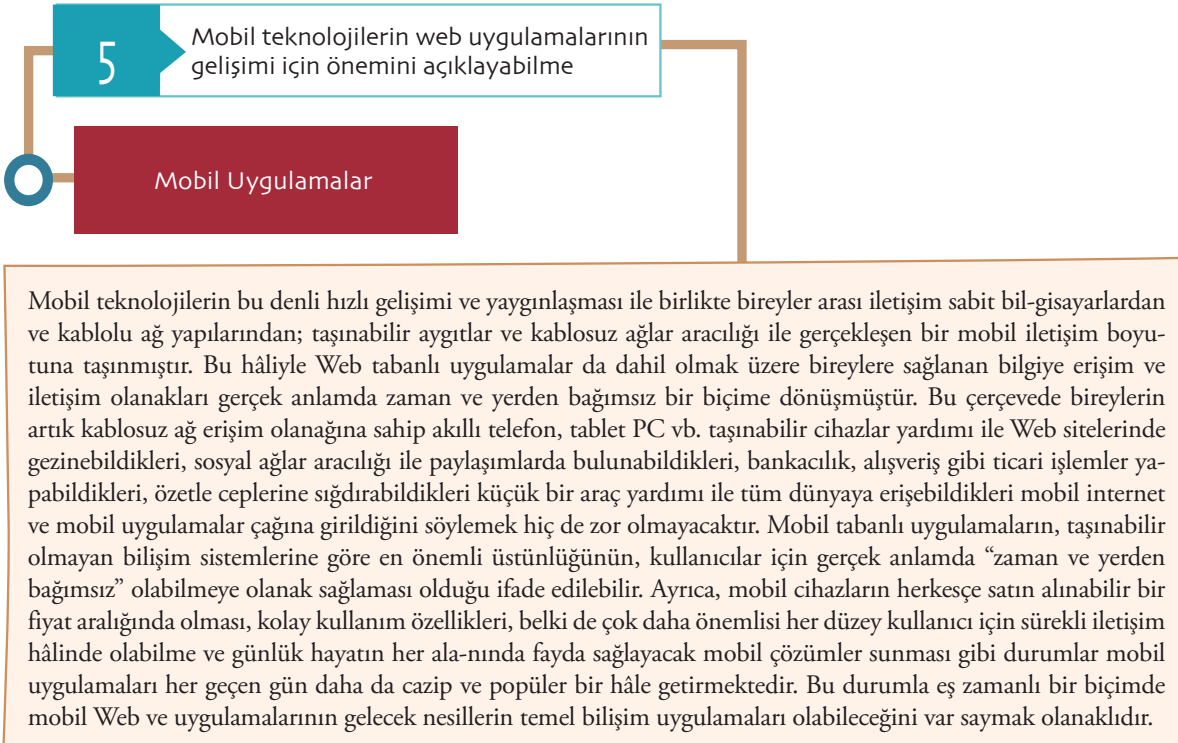
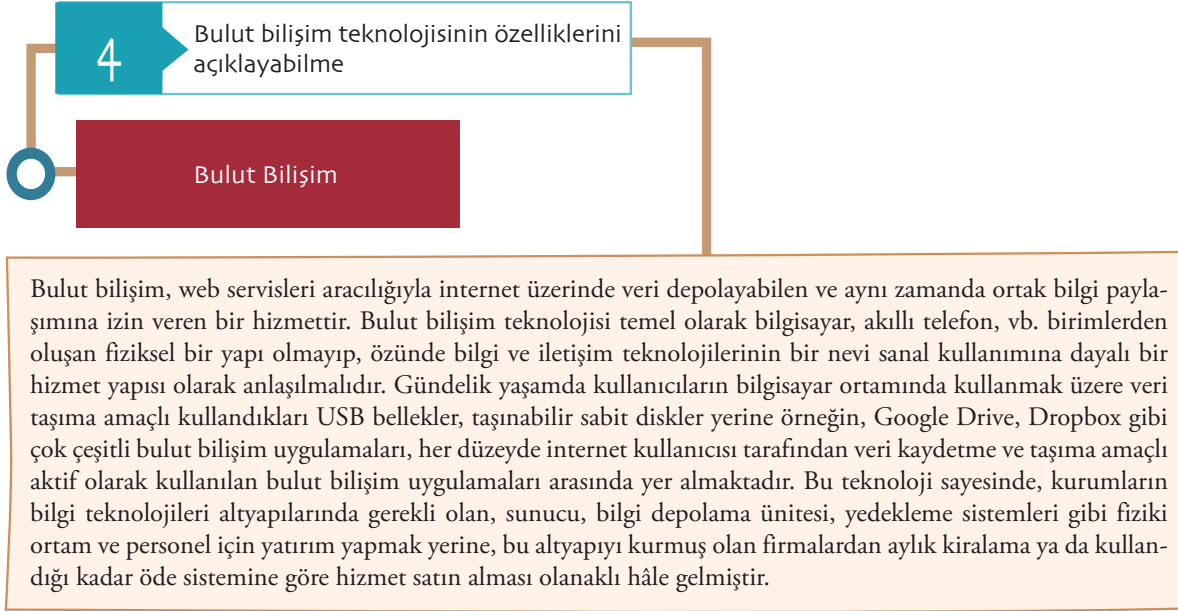
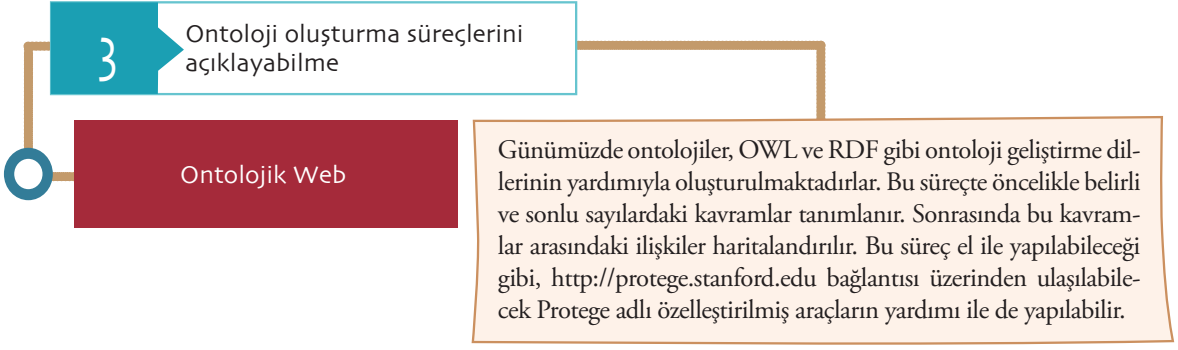
Bilgi ve iletişim teknolojilerinin gelişimini kronolojik olarak incelediğimizde 1960'lı yıllardan bu yana belki de daha önce hiç olmadığı kadar büyük bir ivme yakalandığı görülmektedir. Web'in yalnızca yazı ve resimlerden oluşan statik sürümünü, günümüzün sosyal ağ, blog vb. etkileşimli uygulamalarına alt yapı teşkil eden Web 2.0 izlemektedir. Bu bölümün semantik Web ile ilgili başlıkları altında yer alan Web 3.0 veya akıllı Web şeklinde de adlandırılabilir sürümü gerek sahadaki uygulamalarda gerekse alanyazın temelinde gelinen bir noktadır. Web'in dünü, bugünü ve bu tarihsel gelişim sürecinden hareketle geleceğini ele aldığımız bu bölümün yazımı sırasında değinmediğimiz, ancak Web'in gelecek sürümü için belirtilmesi gereken Web 4.0 üzerine çalışmalar sürmektedir. Yapay zeka, bulut bilişim ve artırılmış gerçeklik uygulamaları gibi araç, hizmet ve teknolojiler ile sanal bir işletim sistemi olarak genel bir tanımlaması yapılan Web 4.0, en azından şimdilik, Web'in geleceği olarak ifade edilebilir. İnsanoğlunun bir yeniliği ortaya koyabilmesi için İnternet ve Web'in icadında olduğu gibi hayal etmesi yeterlidir. Bu bağlamda Web'in 1980'ler de başlayan gelişim sürecinin nereye kadar devam edeceğini kestirmek çok kolay değildir. Ancak bu gelişimin Web 4.0 sürümü ile sınırlı kalmayacağı da açıktır.

2

Semantik web teknolojisini oluşturan temel özellikleri maddeleyebilme

Semantik Web

İnternet üzerindeki tüm bilgilerin ve bunların birbirleriyle ilişkilerinin yalnızca insanlar tarafından değil, makinelerce de anlaşılabilirdiği; özellikle gereksinime odaklı bilgiye erişimin kolaylaştığı; amaca yönelik bağlamlarda bu veri ilişkilerinin rahatlıkla kullanılabilirdiği bir ağ olarak tanımlanabilir. semantik Web'in işleyişinde verinin anlamının ve doğal dil aramalarının yazılım tarafından anlamlandırılabilirdiği bir teknolojiyi işaret etmektedir. Bu durum semantik web mimarisi içerisinde kullanıcı profillerinin, deneyimlerinin, reel uygulamalarda işe koşulan birtakım üst veri (metadata), arama stratejileri ve anahtar kelimelerin bir veri tabanında var olmasını gerektirmektedir. Esasen başlı başına bir veri tabanının varlığı da semantik Web mimarisini tamamlamak adına yeterli olmayacaktır. Bu bağlamda mimariyi oluşturan belirli söz dizim kurallarına (XML), semantik Web yapısının omurgasını oluşturacak olan ve ağ kapsamındaki bilgi yapısının işlenmesini sağlayacak ontolojinin geliştirilmesine gereksinim duyulmaktadır. XML, insanlar ve bilgi işlem sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan bir işaretleme dili ve aynı zamanda W3C tarafından tanımlanmış bir standarttır. Ek olarak ontolojiler ise, varlıklar için ortak tanımlamalar olarak karşımıza çıkmakta ve semantik Web ortamında farklı terimleri açıklamak için ontolojilere gereksinim duyulmaktadır.



1 Aşağıdakilerden hangisi World Wide Web' in kavramsal olarak gelişim sırasını doğru bir biçimde temsil etmektedir?

- A. Statik Web – Etkileşimli Web – Akıllı Web
- B. Statik Web – Akıllı Web – Etkileşimli Web
- C. Etkileşimli Web – Akıllı Web – Statik Web
- D. Akıllı Web – Statik Web – Etkileşimli Web
- E. Akıllı Web – Etkileşimli Web – Statik Web

2 “Yalnızca HTML benzeri bir işaretleme dili olmayan aynı zamanda bilgi yapısını tanımlayan ve standardize eden bir meta dildir.” ifadesinde yer alan boşluğa aşağıdaki terimlerden hangisi gelebilir?

- A. Ontoloji
- B. XML
- C. Web 2.0
- D. Semantik Web
- E. HTML

3 Aşağıdaki ifadelerden hangisi bulut bilişim kavramı ile ilişkili servis olarak yazılım modeli için doğrudur?

- A. Kullanıcılar İnternet yardımı ile bulut ortamında yer alan bir yazılıma erişerek kullanılabilir.
- B. Kullanıcılar bulut ortamında kendi yazılımlarını geliştirmesi
- C. Kullanıcıları veri depolama gibi sanal hizmetlerden yararlanabilir.
- D. Kullanıcıları akıllı telefonlar gibi mobil cihazlarla bulut bilişim servislerine erişebilmeleri
- E. Kullanıcılar bulut bilişim hizmetlerini ücretsiz olarak kullanabilir.

4 Aşağıdakilerden hangisi semantik Web uygulamalarının özelliklerinden biri **değildir**?

- A. Ontoloji tabanlı olması
- B. Semantik yapılandırma
- C. Gerçek dünyada var olan tüm objeleri içermesi
- D. HTML tabanlı olması
- E. XML tabanlı olması

5 Aşağıdakilerden hangisi bulut bilişim teknolojisi ile ilgili üstünlükler arasında **yer almaz**?

- A. Maliyet
- B. Esneklik
- C. Hizmet kalitesi
- D. Zaman ve yerden bağımsız erişim olanağı
- E. Performans

6 Aşağıdakilerden hangisi ontoloji kavramını tanımlamaktadır?

- A. Semantik Web ortamında yapılandırılmış belge ve verilerin evrensel format
- B. Bilgi ve iletişim teknolojilerinin bir tür sanal kullanımına dayalı bir hizmet yapısı
- C. Semantik Web ortamında kavramlar arasındaki ilişkileri biçimsel olarak ifade eden bir doküman
- D. Web tasarımı için kullanılan gelişmiş bir dil
- E. Sanal ortamda veri depolamaya yarayan bir araç

7 Mobil Web uygulamalarının, masaüstü Web uygulamalarına göre en önemli üstünlüğü aşağıdakilerden hangisidir?

- A. Web uygulamalarına farklı mobil cihazlardan erişebilmesi
- B. Web uygulamalarına zaman ve yerden bağımsız biçimde erişebilmesi
- C. Mobil cihazların kullanımının masaüstü bilgisayarlara göre daha kolay olması
- D. Mobil cihazların İnternet erişiminin masaüstü bilgisayarlara göre daha hızlı olması
- E. Mobil cihazların masaüstü bilgisayarlara göre daha düşük maliyetli olması

8 Web ortamında geniş bir kullanıcı kitlesi olan sosyal ağlar (Facebook, Twitter vb.) hangi Web sürümünü temel almaktadır?

- A. Semantik Web
- B. Ontolojik Web
- C. Web 1.0
- D. Web 2.0
- E. Web 3.0

9 Aşağıdakilerden hangisi kullanıcılara sanal ortamda veri depolama olanağı sağlayan bulut bilişim uygulamasıdır?

- A. Ms Office
- B. Facebook
- C. Twitter
- D. Akıllı telefon
- E. Google Drive

10 Aşağıda verilen ifadelerden hangisi doğrudur?

- A. XML ve HTML birbirlerinin benzeri iki işaretleme dilidir.
- B. Bulut bilişim tümüyle yazılım tabanlı hizmetlerden oluşan sanal bir teknolojidir.
- C. Semantik Web'i meydana getiren en önemli temel bileşenlerden birisi XML olarak ifade edilebilir.
- D. Ontoloji kavramının da kullanıldığı Web 1.0 sürümü ile Web teknolojisi etkileşimli bir yapı kazanmıştır.
- E. Protege yazılımı ile kullanıcılar verilerini sanal ortamda depolayabilmektedir.

1. A

Yanıtınız yanlış ise “Web Teknolojisinin Başlangıcı ve Gelişimi” konusunu yeniden gözden geçiriniz..

2. B

Yanıtınız yanlış ise “XML” konusunu yeniden gözden geçiriniz.

3. A

Yanıtınız yanlış ise “Bulut Bilişim” konusunu yeniden gözden geçiriniz.

4. D

Yanıtınız yanlış ise “Semantik Web” konusunu yeniden gözden geçiriniz.

5. B

Yanıtınız yanlış ise “Bulut Bilişim” konusunu yeniden gözden geçiriniz.

6. C

Yanıtınız yanlış ise “Ontolojik Web” konusunu yeniden gözden geçiriniz.

7. B

Yanıtınız yanlış ise “Mobil Uygulamalar” konusunu yeniden gözden geçiriniz.

8. D

Yanıtınız yanlış ise “Web Teknolojisinin Başlangıcı ve Gelişimi” konusunu yeniden gözden geçiriniz.

9. E

Yanıtınız yanlış ise “Bulut Bilişim” konusunu yeniden gözden geçiriniz.

10. C

Yanıtınız yanlış ise “Semantik Web” konusunu yeniden gözden geçiriniz.

8

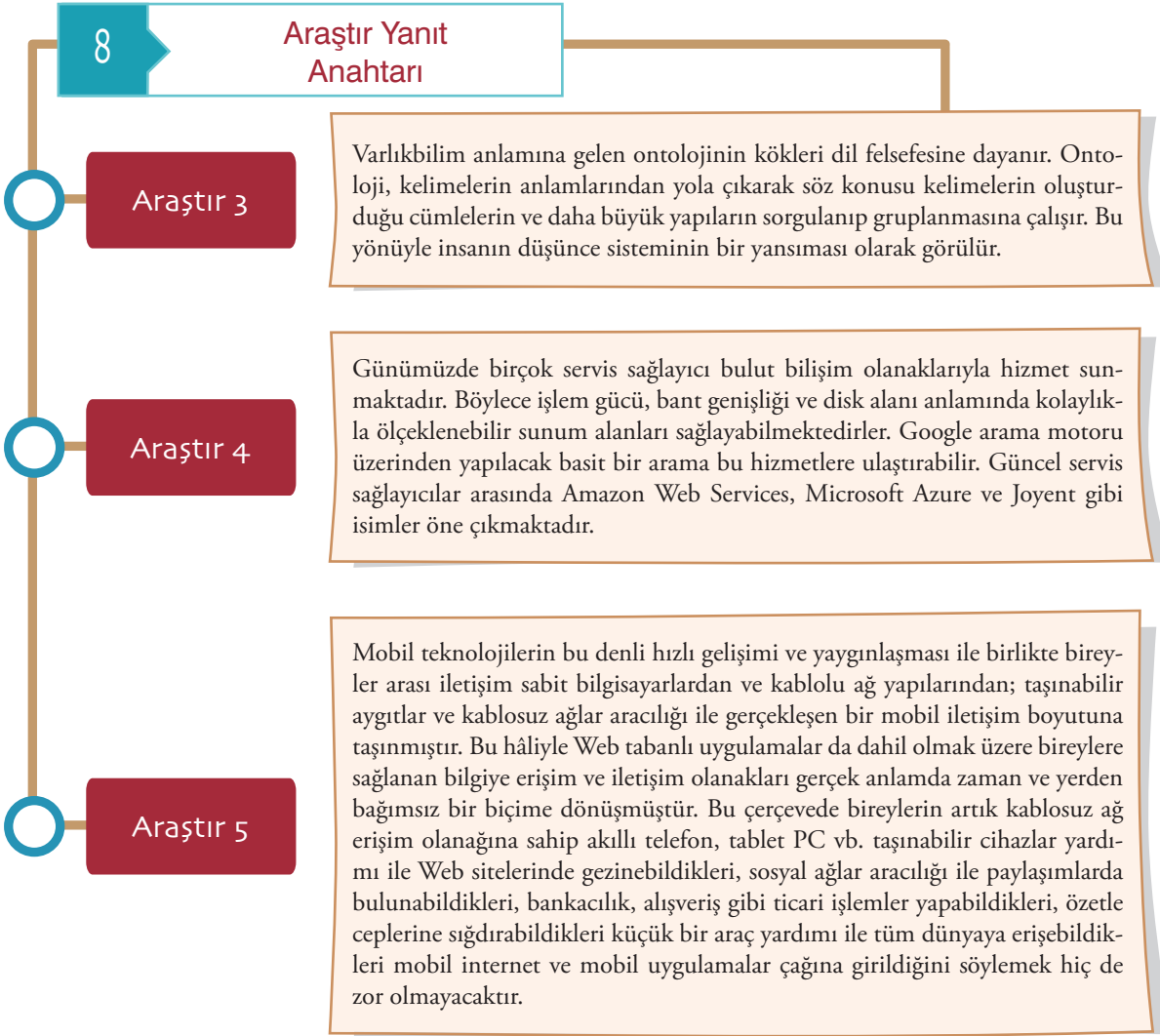
Araştır Yanıt Anahtarı

Araştır 1

Semantik Web, verinin anlamının ve doğal dil aramalarının yazılım tarafından anlamlandırılabilirdiği bir teknolojiyi işaret etmektedir. Bu durum semantik web mimarisi içerisinde kullanıcı profillerinin, deneyimlerinin, reel uygulamalarda işe koşulan birtakım üst veri, arama stratejileri ve anahtar kelimelerin bir veri tabanında var olmasını gerektirmektedir. Esasen başlı başına bir veri tabanının varlığı da semantik Web mimarisini tamamlamak adına yeterli olmayacaktır. Bu bağlamda mimariyi oluşturan belirli söz dizim kurallarına, semantik Web yapısının omurgasını oluşturacak olan ve ağ kapsamındaki bilgi yapısının işlenmesini sağlayacak ontolojinin geliştirilmesine gereksinim duyulmaktadır. Temel bileşenleri XML, Web servisleri ve ontoloji olarak ortaya konulan semantik Web işleyişini, yalnızca Web sayfalarında yer alan bilgilerin kullanıldığı bir insan-bilgisayar ilişkisi olmadığını, verinin yer aldığı, veri tabanı, servisler, programlar, vb. uygulamalara ait verilerin de işlenebildiği karmaşık bir ortam olarak tanımlamaktadır.

Araştır 2

Sayısal imza, kimlik doğrulamak için kullanılan serfitika temelli bir sistemdir. Dijital ortamda kimlik doğrulamayı sağlayan kullanıcı adı-parola uygulamalarından farklı olarak yasa ile belirtilmiş bağlayıcılara sahiptir.



Kaynakça

- BBC (2010), Over 5 billion mobile phone connections Worldwide, <http://www.bbc.com/news/10569081>, Erişim tarihi: 04.07.2016
- Beden, Ş. (2012) Bir Semantik Web-Tabanlı Öğrenme Yönetim Sistemi Modeli, *İstanbul Üniversitesi Fen Bilimleri Enstitüsü*, Yayınlanmamış Yüksek Lisans Tezi.
- Berners-Lee, Tim ve Fischetti, Mark (1999). *Weaving the Web*. Harper San Francisco. s. chapter 12. ISBN 9780062515872.
- Can, Ö. & Ünalır, M.O. (2010) Ontoloji Tabanlı Erişim Denetimi, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 16(2), 197-206.
- Çakır, H. (2013) Semantik Web (Web 3.0) Teknolojisi Kullanılarak Örnek Bir E-Ders İçeriği Hazırlama, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yayınlanmamış Yüksek Lisans Tezi.
- Davies, N. J., Fensel, D. & Richardson, M. (2004) The future of Web Services, *BT Technology Journal*, 22(1), 118-130.
- Demirli, C. & Kütük, Ö.F. (2010) Anlamsal Web (Web 3.0) Ve Ontolojilerine Genel Bir Bakış, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 9(18), 97-107.
- Emiroğlu, G. B. (2009) Semantic Web (Anlamsal Ağ) Yapıları ve Yansımaları, *Akademik Bilişim'09*, Harran Üniversitesi, Şanlıurfa.
- Green, M. (2011) Better, Smarter, Faster: Web 3.0 and the Future of Learning, The Association for Talent Development (ATD), <https://www.td.org/Publications/Magazines/TD/TD-Archive/2011/04/Better-Smarter-Faster-Web-30-and-the-Future-of-Learning>, Erişim Tarihi: 11.07.2016.
- Güldal, H. (2014) Bulut Tabanlı Bir Ders Yönetim Sistemi Yazılımının Geliştirilmesine Dayalı Olarak Öğretim Elemanı ve Öğrencilerin Teknoloji Kabullerinin İncelenmesi, *Trakya Üniversitesi Fen Bilimleri Enstitüsü*, Yayınlanmamış Doktora Tezi.
- Gültepe, Y. & Memiş Kabataş, E. (2014) Kavram Haritalarının Ontoloji Tabanlı Oluşturulması: Kuvvet Konusu Uygulama Örneği, *Journal of Instructional Technologies & Teacher Education*, 3(1), 24-33.
- Gürünlü, Ö. (2002) XML Tabanlı Web Uygulaması, Muğla Üniversitesi, Fen Bilimleri Enstitüsü, Yayınlanmamış Yüksek Lisans Tezi.
- International Telecommunication Union, Sabit/mobil telefon kullanım ve internet bağlantı verileri, <http://www.itu.int/en/ITU-D/Statistics/Pages/stat>, Erişim tarihi: 01.07.2016.
- Internet World Stats, <http://www.internetworldstats.com>, Erişim Tarihi: 01.07.2016 ODTÜ Bilgisayar Topluluğu, Web teknolojisinin gelişimi, <http://e-bergi.com>, Erişim Tarihi: 08.06.2016.
- Ipsos MediaCT (2013) “Mobil Tüketiciyi Anlama”, Our Mobile Planet: Türkiye, <http://www.slideshare.net/edizdar/trkiyede-mobil-tketiciyi-anlama-mays-2013>, Erişim tarihi: 10.07.2016.
- Kurtel, K. (2008) Web'in Geleceği: Anlamsal Web, *Ege Akademik Bakış*, 8(1), 205-213.
- Küçükşille, E.U., Özger, F. & Genç, S. (2013) Mobil Bulut Bilişim ve Geleceği, *Akademik Bilişim 2013 – XV. Akademik Bilişim Konferansı Bildirileri*, Akdeniz Üniversitesi, Antalya.
- Protégé, <http://protege.stanford.edu>, Erişim tarihi: 20.07.2016. The World Wide Web Consortium (W3C), <http://www.w3.org>, Erişim tarihi: 15.06.2016.
- Reid, C. (2016) Tim Berners-Lee Offers His View of the Future of Publishing, *Publishers Weekly*, 23.
- Seyrek, İ.H. (2011) Bulut Bilişim: İşletmeler için Fırsatlar ve Zorluklar, *Gaziantep Üniversitesi Sosyal Bilimler Dergisi*, 10(2), 701 -713.
- Şendağ, S. (2008) Web'de yeni eğilimler: öğrenme ortamlarına entegrasyonu, *Proceedings of 8th International Educational Technology*, Anadolu Üniversitesi, 995-1001.
- Türk Dil Kurumu, <http://www.tdk.gov.tr>, Erişim tarihi: 12.06.2016.
- Türkiye İstatistik Kurumu (TÜİK), Hane Halkı Bilişim Teknolojileri Kullanım Anketi, <http://www.tuik.gov.tr>, Erişim tarihi: 15.06.2016.
- Wikipedia, Anlamsal Ağ, https://tr.wikipedia.org/wiki/Anlamsal_ağ, Erişim tarihi: 01.07.2016.
- Wikipedia, XML, <http://tr.wikipedia.org/wiki/XML>, Erişim tarihi: 01.07.2016.
- Yaşar, C. (2007) Wordnet Üzerinde Türkçe Bilişim Ontolojisinin Oluşturulması, Çanakkale Onsekiz Mart Üniversitesi, Fen Bilimleri Enstitüsü, Yayınlanmamış Yüksek Lisans Tezi.