



Web Teknolojileri

Ders 1

Ders Tanıtımı & Giriş

**Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR**

Konu & İçerik

- Ders Değerlendirme
- Internet & Web
- Internetin Tarihçesi
- TCP / IP
- DNS
- FTP / HTTP / SMTP
- HTTP İstek Yaşam Döngüsü
- Kaynaklar



Ders Değerlendirme

- **Yıl İçi Başarı Oranı %60**
 - Vize %40
 - 1. Proje/Tasarım %20
 - 2. Proje/Tasarım %20
 - 3. Proje/Tasarım %20
- **Yıl Sonu Başarı Oranı %40**

Internet & Web

- **Internet** dünya çapında birbirine bağlı bilgisayarların oluşturduğu bir ağ sistemidir.
- Internetin veriyi bir yerden başka bir yere taşımak gibi basit bir işi vardır.
- **Web** ise, internet üzerinden erişebileceğimiz metin sayfaları, dijital fotoğraflar, müzik dosyaları, videolar, animasyonlar gibi internette çalışan pek çok şeyden ya da uygulamadan biridir.

Internetin Tarihçesi

- 1962 yılında, MIT'deki Leonard Kleinrock, verilerin paketler olarak aktarılmasını sağlayan teknoloji (packet switching theory), üzerine ilk makaleyi yayınladı.
- Aynı zamanda, J.C.R. MIT'den Licklider, insanların herhangi bir yerden bilgiye erişmelerini sağlayacak olan “Galaktik Ağ”ı tanımlayan bir dizi not yazdı.
- 1965 yılında, MIT araştırmacısı Lawrence G. Roberts, Thomas Merrill ile birlikte, Massachusetts'deki TX-2 bilgisayarını California'daki Q-32'ye düşük hızlı bir telefon hattıyla askeri bir proje olan ARPANET kapsamında bağladı.

Internetin Tarihçesi...

- 1972'de Ray Tomlinson, ARPANET için e-posta oluşturdu ve e-posta adresleri için "@" sembolünü kullanmaya başladı.
- Yine 1972 yılında, ARPANET farklı ülkelerde bulunan 23 düğümle birlikte dünyaya yayıldı ve böylece bu **internet** olarak tanındı.
- Zamanla, FTP (Dosya Aktarım Protokolü) TCP / IP protokollerı, DNS, www, tarayıcılar, betik dilleri vb. gibi yeni teknolojilerin icadıyla internet, web üzerinden bilgi yayınılmak ve erişmek için bir ortam sağladı ve web doğdu.

Internetin Tarihçesi...

- Bilgisayarların 1960-1980 yıllarında birbirleriyle bilgi alışverişinde bulunmaları çok nadir.
- 1980: IBM küçük işletmeler için kişisel bir bilgisayar tanıttı. Microsoft, tüm "IBM uyumlu" bilgisayarların aynı programları çalıştırmasına izin veren, Windows adlı yazılım geliştirdi.
- 1980'lerde, Tim Berners-Lee CERN'de farklı bilgisayarları kullanan kişilerin araştırmalarını paylaşmanın kolay bir yolunu aradı.
- Berners-Lee, ASCII (American Standard Code for Information Interchange),'yi bilgiyaların haberleşmesinde bir kural olarak kullandı ve bunu HTTP (HyperText Transfer Protocol) olarak adlandırdı. (örnek curl -IL www.google.com)
- Berners-Lee ikinci kural olarak, tüm CERN bilgisayarlarının HTML (HyperText Markup Language) adlı ortak bir dilde yazılmış dosyaları değişim tokusu etmesini sağladı. Planladığı HTML ile etiked adı verilen özel kodlarla metinleri yapılandırdı.

Internetin Tarihçesi...

- **1993:** Marc Andreessen, daha sonra Netscape ve Mozilla'ya dönüßen ilk kullanıcı dostu web tarayıcısı Mosaic'i yazdı.
- **1994:** İnsanlar kısa sürede hızla büyuyen World Wide Web'de gezinmek için yardıma ihtiyaç duyduklarını anladılar. WebCrawler ve Yahoo gibi arama motorları geliştirildi.
- **1995:** Amazon, eBay'ın kurulmasıyla beraber e-ticaret düzgün bir şekilde başladı.
- **1996:** ICQ, ile birlikte Internet'teki ilk kullanıcı dostu anında mesajlaşma sistemi geliştirildi.
- **1997:** Jorn Barger ilk blogu yayınladı (web günlüğü).
- **1998:** Larry Page ve Sergey Brin, Google'adında bir arama motoru geliştirdi.
- **1999:** Kevin Ashton, sadece bilgisayarların değil, gündelik nesnelerin Internet'in bir parçası olabileceği fikrini ortaya koydu. Bu fikir şimdi Nesnelerin Interneti olarak bilinir.

İnternetin Tarihçesi...

- **2003:** Neredeyse dünyadaki her ülke Internet'e bağlandı.
- **2004:** Harvard öğrencisi Mark Zuckerberg, insanları arkadaşlarıyla buluşturan, kullanımı kolay bir web sitesi olan Facebook ile sosyal ağlarda devrim yarattı.
- **2006:** Jack Dorsey ve Evan Williams, insanların 140 karakterlik mesajlarıyla düşüncelerini ve gözlemlerini paylaşabildikleri basit bir "mikroblog" sitesi olan Twitter'ı duyurdular.
- **2007:** Iphone ile mobile internet kavramı yaygınlaştı
- **2008:** İlk "Internet Seçimi" Amerika'da yapıldı
- **2010:** Facebook 400 milyon aktif kullanıcıya ulaştı
- **2010:** Pinterest ve Instagram sosyal medya siteleri yayınlandı

İnternetin Tarihçesi...

- **2013:** Pew Research Center'ın yaptığı bir araştırmaya göre Amerika'daki yetişkinlerin %50'si bankacılık işlemlerini online olarak yaptıkları belirlendi.
- **2015:** İlk olarak fotoğraf paylaşma sitesi olarak yayınlanan Instagram, Twitter'ı geride bırakarak 400 milyon kullanıcı ulaştı.
- **2016:** Google, ses ile aktifleşen bir kişisel asistan programı olan Google Asistan'ı yayınladı. Böylece, Google Amazon'un Alexa'sı, Apple'ın Siri'si ve Microsoft'un Cortana'sına katıldı.

Web 1.0 -> Web 2.0 -> Web 3.0 -> Web 4.0

➤ Web 1.0

➤ Kullanıcı etkileşimin olmadığı sadece statik sayfalardan oluşan, World Wide Web'in evriminin ilk aşamasını ifade eden yapıya verilen addır.

➤ Web 2.0

➤ İkinci nesil web hizmetlerini barındıran, içeriklerin kullanıcılar tarafından oluşturulduğu, katılımcı web - sosyal web olarak da adlandırılan yapıdır. Web 2.0 ile AJAX ve JavaScript frameworkleri çok popüler bir araç haline gelmiştir.

➤ Web 3.0

➤ Web'in bir veritabanına dönüştürülmesini içerir. Web'de yer alan uygulamaların farklı verilere bağlam sağladığı, bu verilerin farklı uygulamalar tarafından anlaşılıp yorumlandığı bir yapıdır.

➤ Web 4.0

➤ Simbiyotik ağ olarak da adlandırılır. Bu ağır amacı, insanlar ve yapay zeka kullanan makineler arasındaki etkileşimin arttırılmasıdır.

Web 1.0 -> Web 2.0 -> Web 3.0 -> Web 4.0



...

Web 1.0	Web 2.0	Web 3.0	Web 4.0
Çoğunlukla Salt Okunur	Çoğunlukla okuma yazma	Taşınabilir ve Kişisel	...
Ticari Odaklı	Topluluk Odaklı	Bireysellik odaklı	
Web Formları	Web Uygulamaları	Akıllı Uygulamalar	
Dizinler	Etiketler	Kullanıcı Davranışları	
İçerik Sahipliği	İçerik Paylaşımı	İçerik Birleştirme-Anlamlaştırma	
Britannica Online	Wikipedia	The Semantic Web	
HTML/Portals	XML / RSS	RDF / RDFS / OWL	

Internet

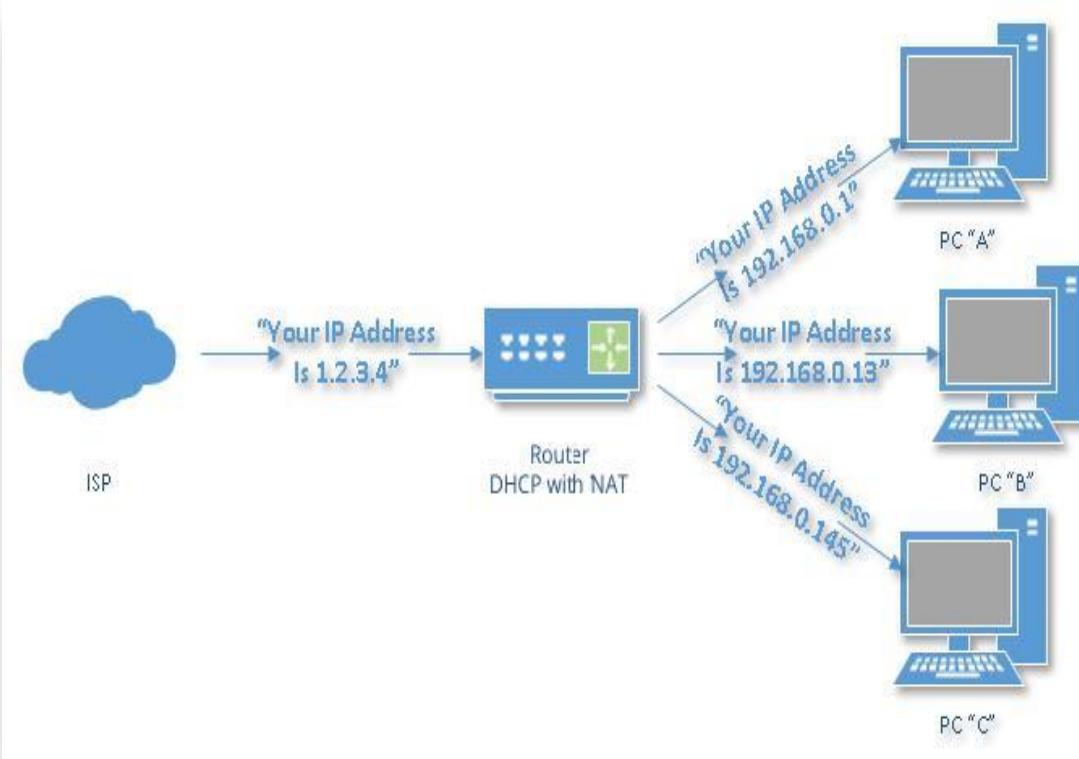
- Internet, dünya çapında birbirine bağlı bilgisayar ağları sistemidir. Internet Protokolünü (IP) kullanır.
- Internette bağlı her bir bilgisayar benzersiz bir sayı kümesinden oluşan ve IP olarak adlandırılan bir adres ile tanımlanır.
- IP adresleri için IP Sürüm 4 (IPv4) ve IP Sürüm 6 (IPv6) olmak üzere iki standart tanımlama vardır.
- IPv4; 10 luk sayı sistemi ile ifade edildiğinde 3 haneli ve her bir hanenin maksimum 255 olduğu 4 bloktan oluşur. IPv4 esasında 32 bit ile ifade edilir.
- (216.27.61.137) ==>11011000.00011011.00111101.10001001

Internet

- IPv4 ile adres ($256 * 256 * 256 * 256$) tanımlanabilir. Zamanla internete bağlı cihazların artmasıyla bu sayının yetersiz kalacağı öngörüldüğünden IPv6 kavramı ortaya atılmıştır.
- IPv6 ile ağda tek bir benzersiz adres oluşturmak için 128 ikili bit kullanılır. Bir IPv6 adresi, **2001:cdba:a93e:0000: 5f9d:81e2:3257: 9652**'de olduğu gibi, sütunlarla ayrılan sekiz onaltılık (taban-16) sayı grubu tarafından ifade edilir. Böylece 2^{128} adet adres sağlanmış olur.

DHCP

- Bir IP adresi dinamik veya statik olarak atanabilir. Statik olarak cihazın ağ ayarları ile kullanıcı tarafından düzenlenirken, dinamik adresler Dynamic Host Configuration Protocol (DHCP) tarafından atanırlar.



*Görüntü https://askleo.com/whats_the_difference_between_a_hub_a_switch_and_a_router/ adresinden alınmıştır

TCP / IP

- Transmission Control Protocol/Internet Protocol
- Application (Uygulama), Transport (Taşıma), Internet veya Network, Data Link Layer (Veri Bağlantısı Katmanı) ve Physical (Fiziksel) Layer
- **NOT:** Bazı kaynaklarda Data link Layer ve Physical Layer Network Access Layer şeklinde tek bir katman olarak tanımlanmaktadır.) olmak üzere 5 katma

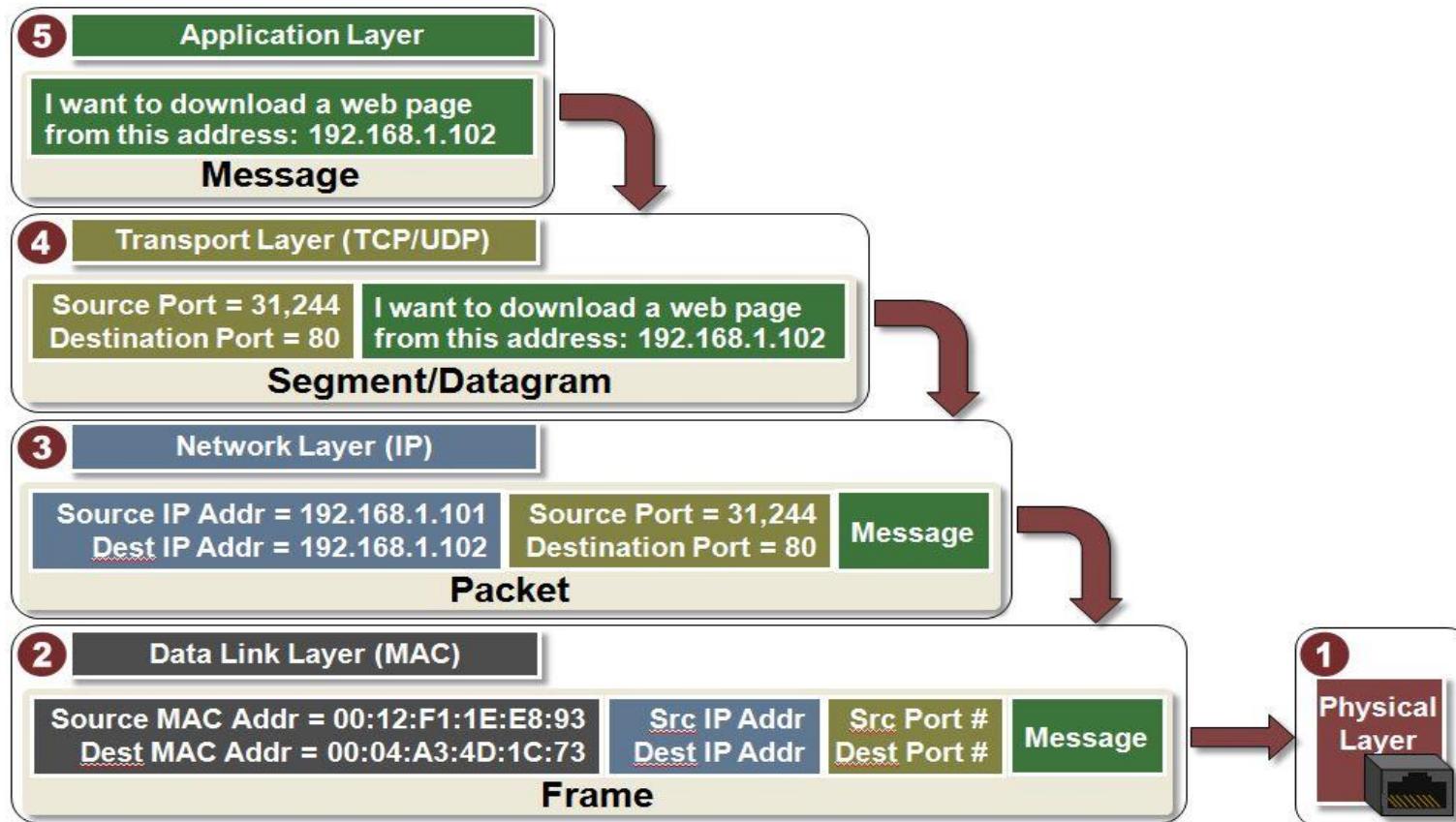
Layer #	Layer Name	Protocol	Protocol Data Unit	Addressing
5	Application	HTTP, SMTP, etc...	Messages	n/a
4	Transport	TCP/UDP	Segments/Datagrams	Port #s
3	Network or Internet	IP	Packets	IP Address
2	Data Link	Ethernet, Wi-Fi	Frames	MAC Address
1	Physical	10 Base T, 802.11	Bits	n/a

*Görüntü <http://microchipdeveloper.com> adresinden alınmıştır

TCP / IP

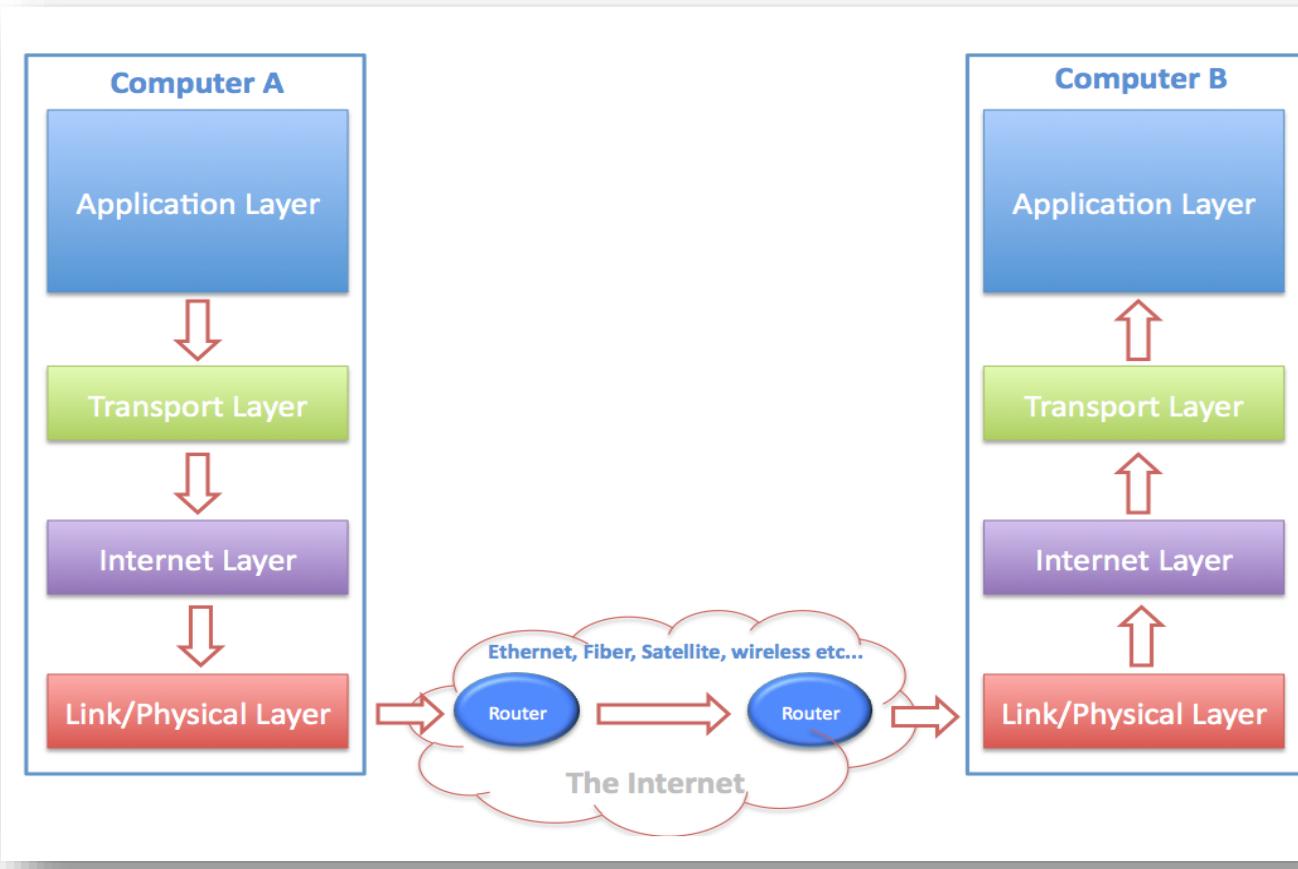
- **Application (Uygulama) Katmanı:** Bu katmanda gönderilecek veri tipi ve veriyi işleyen uygulamalar bulunur. Bilgisayarlarda işlemler tarafından kullanılan protoller tanımlar. Örneğin (HTTP, SMTP, SSH,FTP...)
- **Transport (Taşıma) Katmanı:** Taşıma katmanı, kaynak ve hedef bağlantı noktası adreslerini içeren TCP veya UDP başlığını ekler. Uçtan uca iletimden sorumludur.
- **Internet Katmanı:** Paketin hedefe en iyi yoldan gönderilmesinden sorumludur. IP adresleri veriye bu katmanda eklenir ve yönledime bu katmanda gerçekleştirilir.
- **Veri Bağlantısı Katmanı,** Aynı ağdaki cihazlar arasında veri aktarımından sorumludur. Bu katmanda MAC adres bilgisini içeren bir başlık eklenerek çerçeve verisi oluşturulur. Çerçeve daha sonra bitler şeklinde iletilmek için Fiziksel katmana gönderilir.
- **Fiziksel Katman:** Bitlerin bir ortam üzerinde fiziksel bir sinyal olarak iletilesinden sorumludur.

TCP / IP



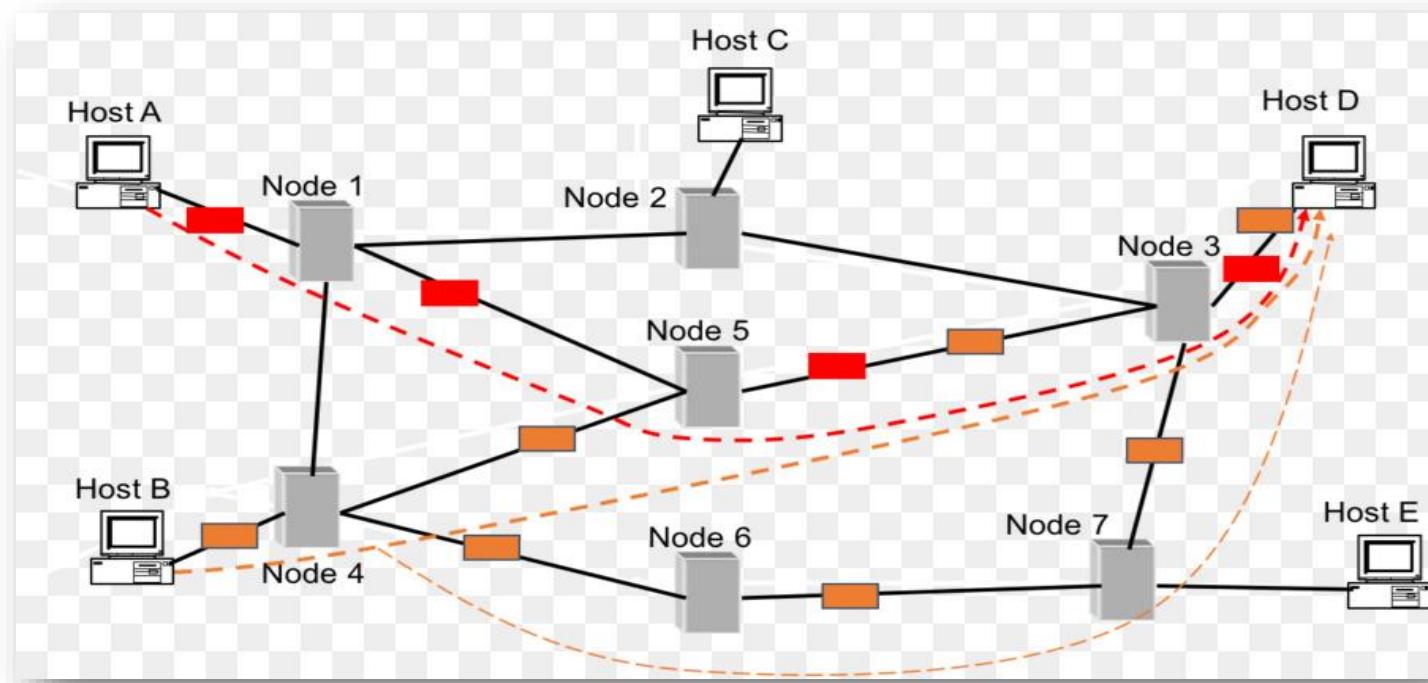
*Görüntü <http://microchipdeveloper.com> adresinden alınmıştır

TCP / IP



*Bu görüntü http://www.cellbiol.com/bioinformatics_web_development adresinden alınmıştır

Paket Anahtarlama



*Bu görüntü <https://www.kisspng.com/png-packet-switching-internet-network-packet-circuit-s-6135085/preview.html> adresinden alınmıştır

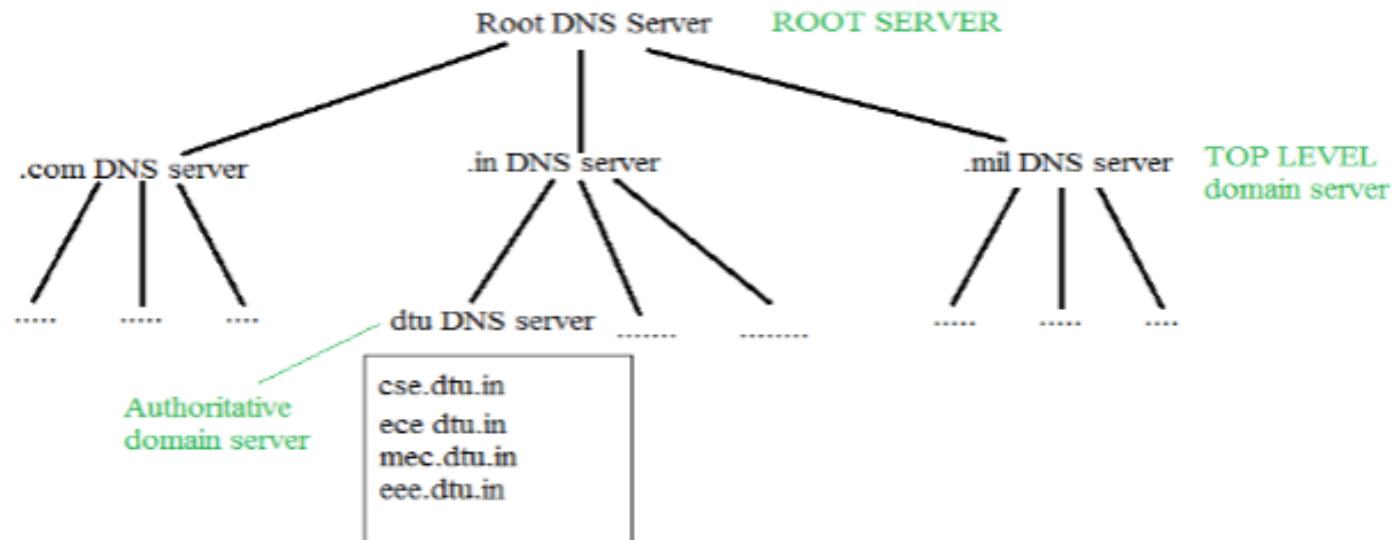
DNS (Domain Name Server)

- DNS adı verilen sunucu kümeleri, insan tarafından okunabilen alan adlarını IP adresleriyle eşleştiren basit veritabanlarıdır.
- Web sunucuları statik IP adresleri kullanır.

Domain Adı	Kullanım Alanı
com	Ticari İşletmeler
edu	Eğitim Kurumları
gov	Hükümet Birimleri
net	Ağ Organizasyonları
mil	Askeri Kuruluşlar
org	Kar amacı gütmeyen kuruluşlar

Domain Adı	Ülke
tr	Türkiye
fr	Fransa
jp	Japonya
es	İspanya
cn	Çin
ca	Kanada
de	Almanya

➤ DNS Organizasyonu



HTTP – FTP – SMTP

- HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), , SMTP ise (Simple Mail Transfer Protocol) olarak bilinen üç protokol de bir bilgisayar ağı üzerinden bilgi aktarmak için kullanılır ve günümüzün internetinin ayrılmaz bir parçasıdır.
- HTTP, World Wide Web'in (WWW) ana unsurlarından biridir. Web Tarayıcılarının (Firefox, Chrome gibi) ve web sunucularıyla iletişim kurduğu, web tarayıcısının belirli bir web tarayıcısına isteğine nasıl yanıt vermesi gerektiğini de tanımlayan mesajların biçimini tanımlar. 80 numaralı portu kullanır.
- FTP, adından da anlaşılacağı gibi, dosyaları bir iletişim ağı üzerinden aktarmak için kullanılan temel protokoldür. FTP 20-21 numaralı portları kullanır.
- SMTP, dünyanın dört bir yanındaki E-posta sunucularının birbirleriyle iletişimini amacıyla kullanılır. SMTP 25 numaralı port ile bu iletişimini gerçekleştirir

HTTP İstek Yaşam Döngüsü

- URL bir domain name çeriyorsa, tarayıcı önce bir DNS sunucusuna bağlanır ve web sunucusu için karşılık gelen IP adresini alır.
- Web tarayıcısı web sunucusuna bağlanır ve istenen web sayfası için bir HTTP isteği (protokol yığını aracılığıyla) gönderir.
- Web sunucusu isteği alır ve istenen sayfayı kontrol eder. Sayfa varsa, web sunucusu istemciye sayfayı gönderir. Sunucu istenen sayfayı bulamazsa, istemciye bir HTTP 404 hata kodu gönderir.
- Web tarayıcısı sayfayı alır ve bağlantı kapatılır.
- Tarayıcı daha sonra web sayfasını tamamlamak için ihtiyaç duyduğu görüntüler, uygulamalar vb diğer sayfa öğelerini parse eder.
- Tarayıcı; gereken her öğe için , sunucuya ek bağlantılar ve HTTP istekleri yapar.
- Tarayıcı tüm görüntüleri, uygulamaları vb. yüklemeyi bitirdiğinde, sayfa tarayıcı penceresine tamamen yüklenmiş olur.

Örnek İstek / Cevap

➤ İstek

```
* Rebuilt URL to: www.google.com/
*   Trying 216.58.212.36...
* TCP_NODELAY set
* Connected to www.google.com (216.58.212.36) port 80 (#0)
> GET / HTTP/1.1
> Host: www.google.com
> User-Agent: curl/7.55.1
> Accept: */*
```

➤ Cevap

```
< HTTP/1.1 200 OK
< Date: Tue, 23 Feb 2021 19:11:58 GMT
< Expires: -1
< Cache-Control: private, max-age=0
< Content-Type: text/html; charset=ISO-8859-1
< P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
< Server: gws
< X-XSS-Protection: 0
< X-Frame-Options: SAMEORIGIN
< Set-Cookie: 1P_JAR=2021-02-23-19; expires=Thu, 25-Mar-2021 19:11:58 GMT; path=/; domain=.google.com; Secure
< Set-Cookie: NID=210=m9070yIQsKt5WIBIDt7a0e3UeIrzQ3K9bEUSM7bTmDsZMvgVoUBKfT2gstsB-041j5kL6Jq1AsqSvvpkIbTKzVgKhJhxAmSIgi
l47goF2FpRt0cbqEdKZwwumkAsp2-sKSG72PY_pDp73kJNROHS5xDWgLRH3i26qe6N9IRBI0; expires=Wed, 25-Aug-2021 19:11:58 GMT; path=/
; domain=.google.com; HttpOnly
< Accept-Ranges: none
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
```

KAYNAKLAR

- <https://www.explainthatstuff.com>
- <https://www.makeuseof.com/>
- <https://www.tutorialspoint.com>
- <https://computer.howstuffworks.com>
- <https://www.geeksforgeeks.org>



Web Teknolojileri

Ders 2

Internet & HTML

Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

- Internet
- www
- Web
- Web Sayfası Çalışma Mantığı
- HTML



Internet

- **Internet**, bilgisayar sistemlerini birbirine bağlayan elektronik iletişim ağıdır.
- Günümüzde hemen hemen tüm ülkeler Internet ağına bağlıdır.
- Internet ağında resmi kurumlar, özel kuruluşlar, kişisel bağlantılar v.b. yer almaktadır.
- Günümüzde Internet Ağına bağlı 15 milyardan fazla akıllı cihaz bulunmaktadır.
- Ağ'da TerraByte'dan daha fazla bağlantı gerçekleştirılmıştır.
- Sosyal Ağlara bağlantılarının %60'dan fazlası akıllı cihazlarla gerçekleştirılmıştır.

Web Sayfası

- Web sayfaları, yapılarına göre statik ve dinamik web sayfaları olmak üzere ikiye ayrılırlar.
- Statik web sayfaları sunucu tarafından dosyadan okunarak işlenmeden direkt olarak istemciye gönderilir. Bildiğimiz htm ya da html uzantıyla sahip sayfalar -*özel durumlar hariç*- statik sayfalardır.
- Dinamik web sayfaları sunucu taraflı ve istemci taraflı dinamik web sayfaları olmak üzere iki şekilde gerçekleştirilebilir.

Web Sayfası Çalışma Mantığı

- Web sayfaları web tarayıcılarında(Browser) görüntülenir.
- Web sayfaları HTML dili kullanılarak oluşturulur.
- Web sayfalarının transfer edilmesi için HTTP (Hypertext Transfer Protocol) ortak protokolü kullanılarak gerçekleştirilir.
- Web sayfaları Web sunucularında tutulur.
- Her sayfanın bir adresi vardır.
- İstenilen sayfa sunucu tarafından istemciye gönderilir.

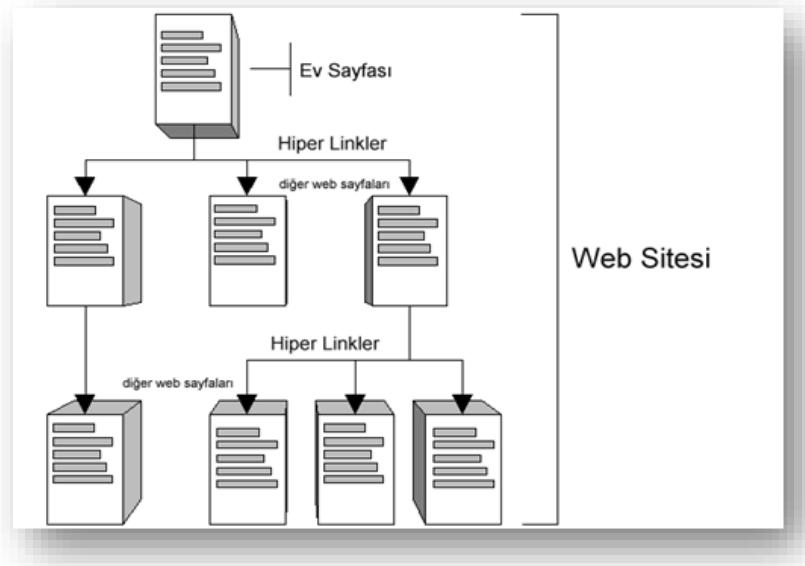
Web Sayfası Çalışma Mantığı...

➤ Web Tarayıcısının adres çubuğuına görüntülenmesi istenilen sayfanın URL adresi yazılıp Enter tuşuna basıldığında (Örnek olarak www.sakarya.edu.tr) DNS isimli sunucular bu adresin IP adresini bulur (Sakarya Üniversitesi IP adresi:193.140.253.140). Yapılan istek Web Sunucusuna iletilir.

- URL : Uniform Resource Locator
- DNS : Domain Name Server Internetteki tüm sayfaların IP adresini saklar.
- IP : Internet Protocol. Internette her bilgisayarın bir IP adresi vardır. Bilgisayarlar bu IP adresleri ile haberleşirler.

Web Sayfası Çalışma Mantığı...

- Web Sunucu yazılımı yüklü olan bilgisayarda gelen http taleplerini karşılamak için bilgisayardaki belirli bir klasör ayrılır (Örnek olarak C:/www/ gibi).



- Web sunucularında gelen istek için anasayfa index.htm, index.php gibi sayfa isimleri olabilir)

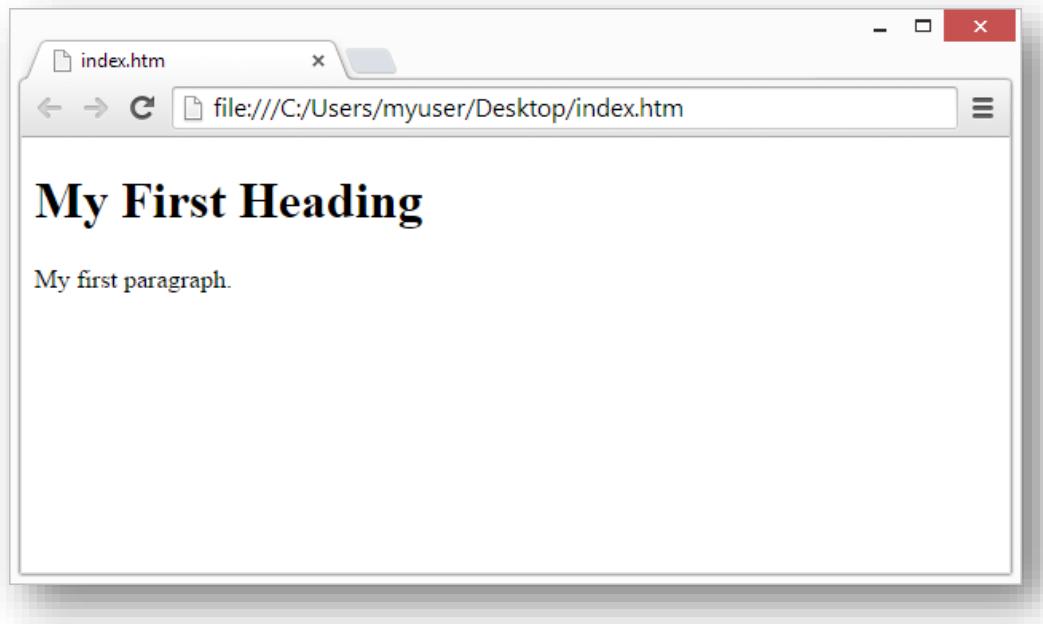
Web Sayfası Çalışma Mantığı...

- Sunucuya gelen istekler için belirli bir portu sürekli olarak dinler.
- Gelen isteğe göre, istekte bulunan sayfa, türüne göre işlenerek veya doğrudan istekte bulunana istemcinin bilgisayarına gönderilir.
- İstemci bilgisayardan gelen web sayfası tarayıcı programda (işlenerek veya doğrudan) kullanıcıya gösterilir.

□ HTTP için varsayılan port 80 numaralı porttur.

Web Tarayıcısı

- Web sayfaları web tarayıcıları tarafından yorumlanarak görüntülenir.
- Chrome, Safari, Firefox, Internet Explorer gibi.
- Web içeriklerinin kullanıcının bilgisayarına gönderilebilmesi için de web sunucusuna ihtiyaç vardır.



- 1994 yılında Tim Berners-Lee, World Wide Web Consortium (W3C) isminde bir organizasyon kurdu. Vizyonu, web'i oluşturan protokol ve teknolojilerin standardizasyonuydu.
- 1993'de HTML standardı ortaya çıkmıştır.
- 1995'de HTML 2.0 yayınlandı.
- 1997'de HTML 4.0 yayınlandı.
- 1999'da HTML 4.01 yayınlandı.
- 2000'de XHTML 1.0 yayınlandı.
- 2014'de HTML 5 yayınlandı.

HTML...

- HTML (HyperText Markup Language / Hareketli-Metin İşaretleme Dili) tarayıcıda gösterilen sayfayı oluşturmaya yarayan bir işaretleme dildir. Sayfanın hypertext olması, sayfanın ya da bir parçasının, başka bir sayfanın içinden çağrırlabilmesi nedeniyledir. Çağırılan, aynı sayfa içinde bir kısım ya da ağ üzerinde herhangi bir bilgisayarda bulunan başka bir sayfa veya sayfanın bir kısmı olabilir.

HTML...

- Hyper Text Markup Language (Hiper Metin İşaretleme Dili) web sayfalarını oluşturmak için kullanılan standart metin işaretleme dilidir.
- HTML, bir programlama dili olarak tanımlanamaz.
- HTML kodlarıyla kendi başına çalışan bir program yazılamaz.
- Ancak bu dili yorumlayabilen programlar aracılığıyla çalışabilen programlar yazılabılır.
- Web Sayfaları HTML tagları ile oluşturulur.
- Her bir tag web sayfasının farklı bir yerinin görüntülenmesi sağlar.

➤ Özellikleri

- Web sayfalarındaki yazılara değişik formatlar verilebilir (koyu renk yazı, italik yazı, yazı ortalaması, renk verme, değişik boyutlarda yazma gibi),
- Tablo, Liste, Adres Alanı, sabit genişlikli yazı alanı vb gibi özel biçimler oluşturulabilir,
- Web sayfalarında "frame" adı verilen ve birbirleri ile ilişkilendirilebilen alt-kısımlar oluşturulabilir,
- Ses, grafik, animasyon gibi uygulamaların web sayfalarından çalıştırılabilmesi için gerekli ortamlar sağlanabilir,
- Java, JavaScript gibi programlama/script dilleri ile web içeriklerinin etkileşimli kullanımı için gerekli ortamları sağlar,

➤ Döküman Yapısı

- Bir HTML dokümanı genel olarak (frameset'ler gibi istisna kullanımları da vardır) aşağıdaki şekildedir:
- <HTML>
<HEAD>
<!-- Head (başlık bölümü) elemanları -->
Burada genelde doküman içeriği dışında kalan karakter set tanımlamaları, başlık, JavaScript tanımlamaları vb elemanlar bulunur...
</HEAD>
<BODY>
<!-- Body (gövde bölümü) elemanları -->
Burada ise dokümanın asıl içeriği (görünen sayfa) vardır. İçinde metin, ses, video vb içerebilir...
</BODY>
</HTML>

➤ İlk Örnek

```
<html>
<head>
<title>Sayfa Başlığı</title>
</head>
<body>

<h1>Başlık </h1>
<p>Paragraf</p>

</body>
</html>
```

➤ Sayfa Yapısı

```
<html>

  <head>
    <title>Page title</title>
  </head>

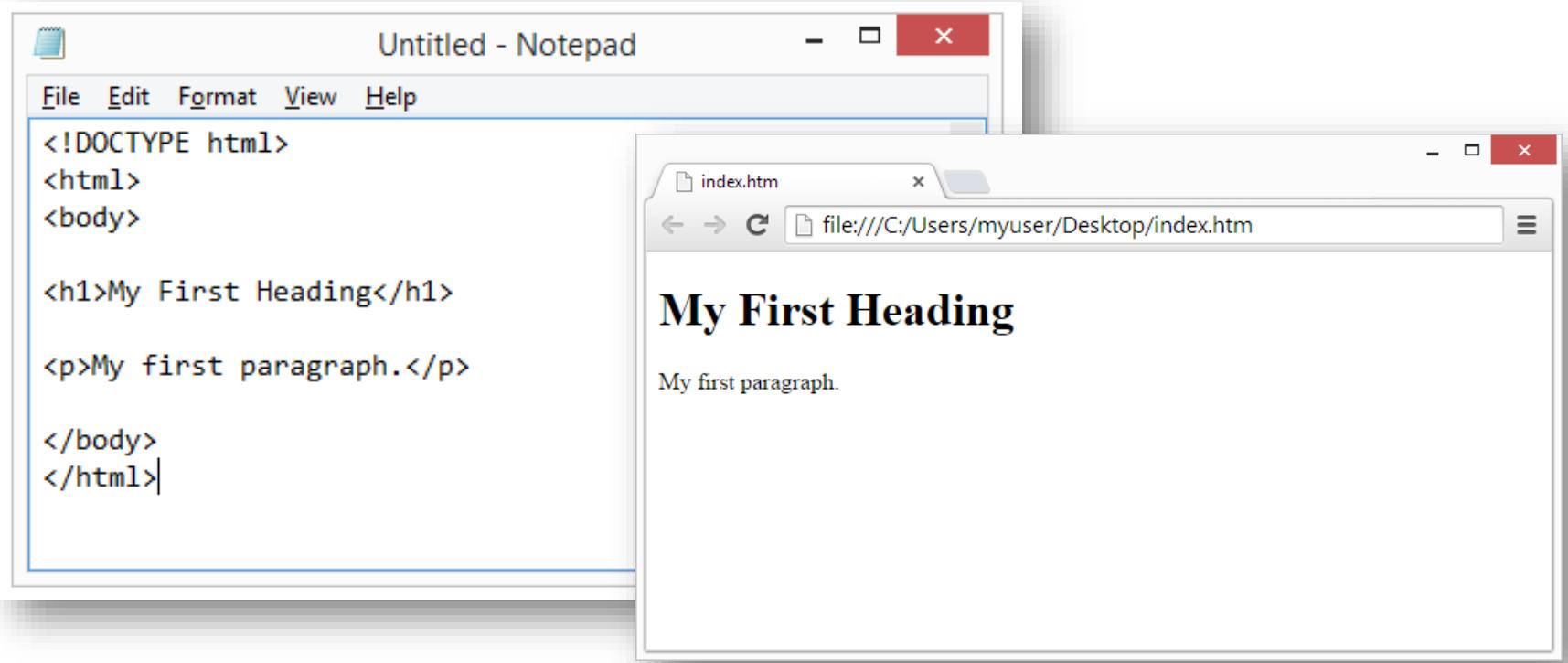
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>

</html>
```

HTML...

➤ Editörler

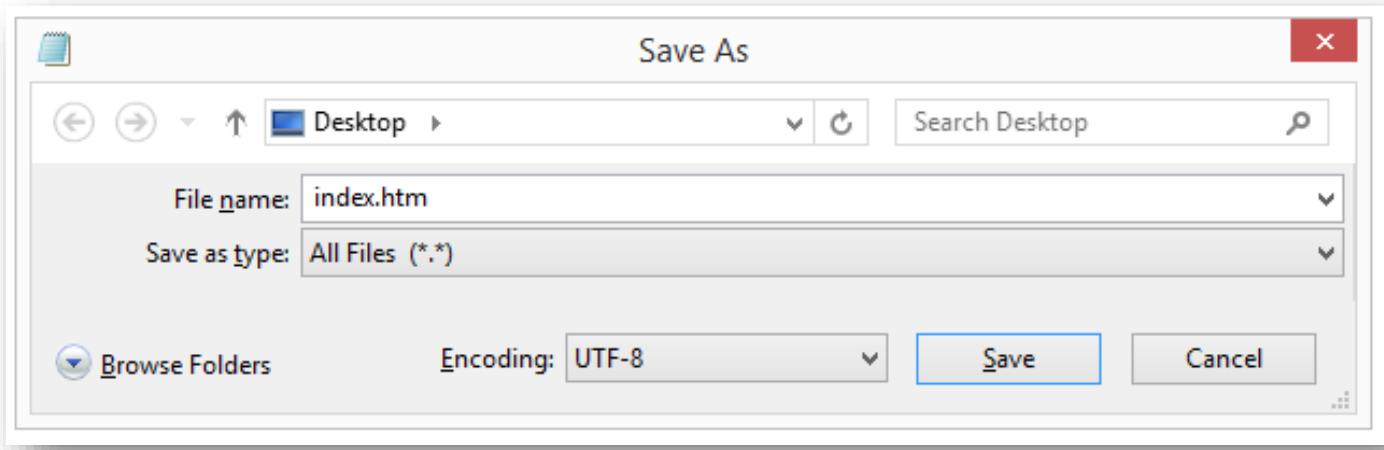
- HTML sayfaları basit editörler ile yazılabilir.
- Bunlara Notepad, Sublime Text, UltraEdit, Notepad++, Visual Studio Code verilebilir.



HTML...

➤ Sayfaları Kaydetmek

➤ Html sayfaları .html veya .htm uzantısı ile kaydedilir.



➤ <!DOCTYPE> Etiketi

- DOCTYPE web sayfası dökümanı hakkında web tarayıcılarına bilgi vermek amacıyla kullanılırlar.
- Html taglarının hepsinde büyük küçük harf duyarlılığı yoktur. Ama küçük harf tercih edilir.
- Aşağıdaki tüm kullanımlar geçerlidir.

```
<!DOCTYPE html>
```

```
<!DOCTYPE HTML>
```

```
<!doctype html>
```

```
<!Doctype Html>
```

HTML...

➤ <!DOCTYPE> Etiketi...

HTML5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```



➤ Başlangıç ve Bitiş Etiketleri

- HTML sayfalarında her içerik başlangıç ve bitiş tagları arasında gösterilir.
- Bazı taglar bitiş tagı içermez. Örnek


```
< etiket_adı > içerik < / etiket_adı >
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>

HTML...

➤ Nitelikler

- HTML elementleri niteliklere sahiptir.
- Nitelikler HTML etiketlerine ek bilgiler sağlar.
- Nitelikler her zaman başlangıç tagına yazılır.
- Nitelikler **name="value"** şeklinde verilir.
- Örneğin lang niteliği sayfanın dilini bildirmek için kullanılır.
- İlk iki harf dili gösterirken ikinci iki harf ülkeyi gösterir.

```
<!DOCTYPE html>
<html lang="tr-TR">
<body>
<h1>İlk Başlık</h1>
<p>İlk paragraf.</p>
</body>
</html>
```



➤ Nitelikler...

- Bütün nitelik içerikleri tırnak işaretleri arasında verilir.
- Eğer tırnak işareti kullanılmazsa doğru olarak gösterilemeyebilir.
- Aşağıdaki iki örnekten ikincisi boşluk içerdiği için sadece boşluktan önceki kelime gösterilecektir.
- Çünkü boşluktan sonraki kelime yeni bir nitelik olarak yorumlanacaktır.

```
<a href=http://www.w3schools.com>
```

```
<p title=About W3Schools>
```

➤ Başlık Etiketi - <h>

- Html sayfalarında başlık verebilmek için <h> etiketleri kullanılır.
- <h> etiketi 1-6 arasında farklı boyutlarda başlık vermek için kullanılır.

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

HTML...

➤ Paragraf Etiketi - <p>

➤ Html sayfalarında paragraflar <p> etiketi ile verilir.

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

➤ Bağlantı Etiketi - <a>

- Html sayfalar arasında bağlantı kurabilmek için <a> etiketi kullanılır.
- href niteliği ise bağlantının adresini bildirmek için kullanılır.

```
<a href="http://www.w3schools.com">This is a link</a>
```

- Sayfa içerisinde id verilmiş belirli bir noktaya ulaşmak için de bağlantı verilebilir

```
<a href="deneme1#orta">Burası sayfanın ortasıdır</a>
```

➤ Resim Etiketi -

- Html sayfalarda resim görüntüleyebilmek için etiketi kullanılır.
- src niteliği resmin kaynak yolunu verir.
- width niteliği resmin genişliğini pixel cinsinden tanımlar.
- height niteliği resmin yüksekliğini tanımlar.
- alt niteliği resim hakkında bilgi vermek için kullanılır. Örneğin görme engelli ziyaretçiler tarafından resmin içeriği hakkında bilgi vermek için kullanılmaktadır.

```

```

➤ "title" Özelliği/Niteligi

- title niteliği eklendiği etiketin içeriği hakkında kullanıcıya bilgi vermek için kullanılır.

```
<p title="About W3Schools">
W3Schools is a web developer's
site.
It provides tutorials and
references covering
many aspects of web programming,
including HTML, CSS, JavaScript,
XML, SQL, PHP, ASP, etc.
</p>
```

➤ HMTL Metin Biçimlendirme

- **** veya **** metni koyu yazmak için kullanılır.

```
<p>This text is normal.</p>
```

```
<p><b>This text is bold</b>.</p>
```

- **<i>** metni yatay yazdırır.

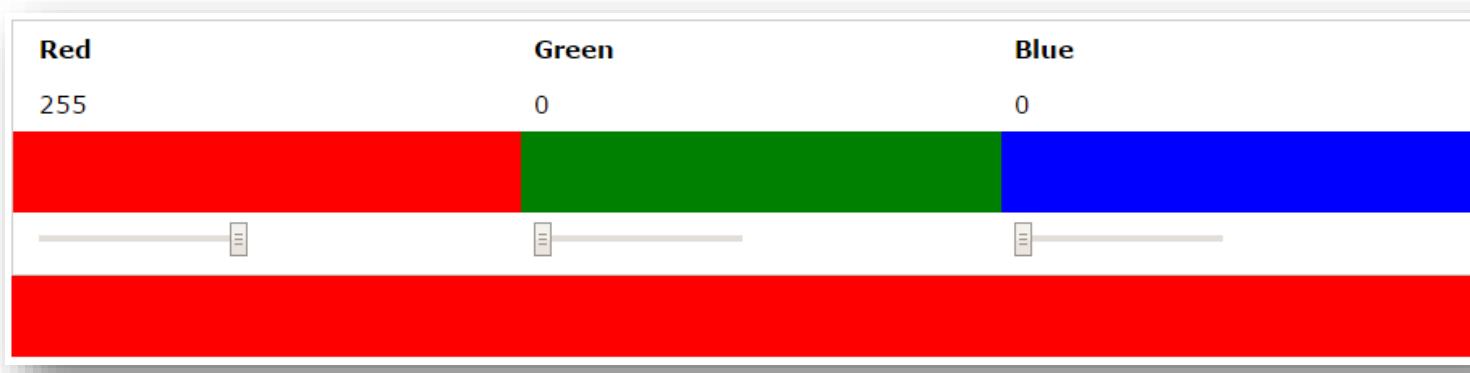
```
<p>This text is normal.</p>
```

```
<p><i>This text is italic</i>.</p>
```

Renkler

➤ Renklendirme RGB (Red, Green, Blue)

- Renkler üç ana rengin birleşmesiyle oluşur.
- Bunlar kırmızı, yeşil ve mavidir.
- Sarı rengi kırmızıdan maksimum yani 255, yeşil 255 ve maviden 0 oranında karıştırılarak oluşturulur.
- `rgb(255, 0, 0)`



Renkler...

- #000000 – Siyah
- #FFFFFF – Beyaz

#000000=black(siyah)

#0000FF=blue(mavi)

#008080=teal(koyu yeşil)

#00FFFF=aqua(turkuaz)

#800080=purple(mor)

#808080=gray(gri)

#FF0000=red(kırmızı)

#FFFF00=yellow(sarı)

#000080=navy(lacivert)

#008000=green(yeşil)

#00FF00=lime(parlak yeşil)

#800000=maroon(vişne çürüüğü)

#808000=olive(zeytuni yeşil)

#C0C0C0=silver(gümüşü gri)

#FF00FF=fuchsia(parlak pembe)



KAYNAKLAR

➤ <https://www.w3schools.com/>



Web Teknolojileri

Ders 3

Tablolar & Listeler

**Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR**

Konu & İçerik

- Tablolar
- Listeler
- <iframe> etiketi
- <head> etiketi



Tablolar

- Tablolar **<table>** etiketi ile tanımlanır.
- Tablodaki satırlar **<tr>** etiketi ile tanımlanır.
- Tablodaki sütunlar **<td>** etiketi ile tanımlanır.
- Tablodaki sütün başlıklarları **<th>** etiketi ile tanımlanır.

```
<table>
  <tr>
    <th>Adı</th>
    <th>Soyadı</th>
    <th>Notu</th>
  </tr>
  <tr>
    <td>Ahmet</td>
    <td>ÖZTÜRK</td>
    <td>90</td>
  </tr>
</table>
```

Adı	Soyadı	Notu
Ahmet	ÖZTÜRK	90



Tablolar

➤ Sütun Birleştirme

➤ İki veya daha fazla sütunu birleştirmek için **colspan** niteliği kullanılır.

```
<table>
  <tr>
    <th>Adı Soyadı</th>
    <th colspan="2">Telefon Numarası</th>
  </tr>
  <tr>
    <td>Ahmet ÖZTÜRK</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
  </tr>
</table>
```

Adı Soyadı	Telefon Numarası	
Ahmet ÖZTÜRK	555 77 854	555 77 855



Tablolar

➤ Satır Birleştirme

➤ İki veya daha fazla satırı birleştirmek için **rowspan** niteliği kullanılır.

```
<table>
  <tr>
    <th>Adı Soyadı:</th>
    <td>Ahmet ÖZTÜRK</td>
  </tr>
  <tr>
    <th rowspan="2">Telefon Numarası:</th>
    <td>555 77 854</td>
  </tr>
  <tr>
    <td>555 77 855</td>
  </tr>
</table>
```

Adı Soyadı:	Ahmet ÖZTÜRK
Telefon Numarası:	555 77 854
	555 77 855



Listeler

➤ Sırasız Listeler

- Sırasız listeler `` etiketi ile başlatılır.
- Her liste elemanı `` etiketi ile tanımlanır.

```
<ul>
  <li>Kahve</li>
  <li>Çay</li>
  <li>Şeker</li>
</ul>
```

- Kahve
- Çay
- Şeker

The CSS `list-style-type` property is used to define the style of the list item marker. It can have one of the following values:

circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Listeler

➤ Sıralı Listeler

- Sıralı listeler `` etiketi ile başlatılır.
- Her liste elemanı `` etiketi ile tanımlanır.

```
<ol>
  <li>Kahve</li>
  <li>Çay</li>
  <li>Şeker</li>
</ol>
```

1. Kahve
2. Çay
3. Şeker

Listeler

➤ Sıralı Listeler

➤ type niteliğinin değeri 1, i, a, I, A değerleri ile numaralandırma değeri belirlenebilir. Start niteliği ile numaralandırma değerinin başlangıç değeri belirlenir.

```
<ol type="A" start=3>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

- C. Coffee
- D. Tea
- E. Milk

<iframe> Etiketi

- Bir web sayfasının içerisinde belli bir kısmında başka bir web sayfası gösterilmek isteniyorsa iframe etiketi kullanılır.
- Sayfanın gösterileceği kısmın yüksekliği ve genişliği, height ve width ile belirlenir. Gösterilecek sayfanın adresi src niteliğine yazılır.

```
<iframe src="http://www.sakarya.edu.tr/
           width="600"
           height="450">
</iframe>
```

<iframe> Etiketi

- Google map üzerindeki bir adresi iframe ile göstermek için harita üzerinde bulunan paylaş sekmesinden Haritayı yerleştir seçenekinde açılan html etiketi sayfaya kopyalanır.

```
<iframe  
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d12083  
.119343158885!2d30.39719923164421!3d40.78885411917557!2m3!1f0!2f  
0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x409df633dbfb5ae7%3A0x9  
dab4f3c21ca722a!2sAdapazarı%2C+Karaosman+Mahallesi%2C+54100+Adap  
azar%C4%B1%2FSakarya!5e0!3m2!1str!2str!4v1517751745315"  
width="600" height="450" frameborder="0" style="border:0"  
allowfullscreen></iframe>
```

<iframe> Etiketi

- Youtube videolarını iframe ile göstermek için video altında bulunan paylaş sekmesinden yerleştir seçenekinde açılan html etiketi sayfaya kopyalanır.

```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/QAeQVDfLa6w" frameborder="0"  
allow="autoplay; encrypted-media" allowfullscreen></iframe>
```

<head> Etiketi

- Tarayıcıya ve sayfayı tarayacak sistemlere verilecek bilgileri gösteren etiketler <head> etiketinin içerisine yazılır.
- Bunlar; title, style, meta, link, script, base

<head> Etiketi...

- <title> Etiketi
- Tarayıcının başlık kısmında sayfa ile ilgili gözükecek bilgi title etiketleri arasında yazılacaktır.

```
<head>
  <title>Page Title</title>
</head>
```

<head> Etiketi...

➤ <meta> Etiketi

- Meta etiketleri arasında nitelikler kullanılarak sayfaların karakter setleri, sayfaya ait anahtar kelimeler, sayfa yenileme sıklığı gibi değerler belirlenebilir.

```
<head>
  <title>İlk Web Sayfam</title>
  <meta charset="UTF-8">
  <meta name="description" content="Web Teknolojileri">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="Gülüzar ÇIT">
  <meta http-equiv="refresh" content="30">
</head>
```

KAYNAKLAR

➤ <https://www.w3schools.com/>



Web Teknolojileri

HAFTA 4

Formlar

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKİ
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

- HTML Formlar
- PHP ile Form Bilgisi Gönderme/Alma



Formlar

- Html formlar kullanıcıdan veya web sayfasından bilgi almak amacıyla kullanılan html elemanlarıdır.
- Basit bir form tanımı

```
<form>  
    .  
    form elemanları  
    .  
</form>
```

- Html formlar form elemanlarını içerir.
- Form elemanları input elemanın farklı şekilleridir. checkboxes, radio buttons, submit buttons, text ve diğerleri.

Formlar

➤ <input type="text">

- Tek satırlı metin girişi elemanı tanımlar.

```
<form>
    Adı :<br>
    <input type="text" name="adi"><br>
    Soyadı :<br>
    <input type="text" name="soyadi">
</form>
```

Adı :

Soyadı :

Formlar

➤ <label>

- Form elemanı için bir etiket tanımlar

```
<form>
  <label for="adi">Adı :</label>
  <input type="text" name="adi"><br>
  <label for="soyadi">Soyadı :</label>
  <input type="text" name="soyadi">
</form>
```

The image shows a simple HTML form with two text input fields. The first field is labeled "Adı :" and the second is labeled "Soyadı :". Both labels are positioned above their respective input boxes.

Formlar

➤ <input type="radio">

- radio button tanımlar
- radio butonlar ile sadece bir seçeneği seçebilirsiniz.

```
<form>
  <input type="radio" name="cinsiyet" value="erkek" checked>Erkek<br>
  <input type="radio" name="cinsiyet" value="kadin">Kadın
</form>
```

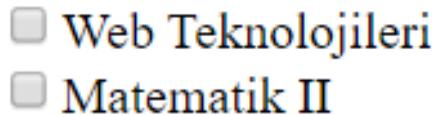
- Erkek
- Kadın

Formlar

➤ <input type="checkbox">

- Onay kutularını tanımlar
- Onay kutuları ile ya hiç ya da tüm seçenekler seçilebilir.

```
<form>
  <input type="checkbox" name="ders1"
         value="Web">Web Teknolojileri<br>
  <input type="checkbox" name="ders2"
         value="Mat">Matematik II
</form>
```



Formlar

➤ <input type="submit">

- Formu göndermek için kullanılan butondur.
- Submit butonuna tıklanınca sayfa action özelliği ile belirtilen yere yönlendirilir.

```
<form action="action_page.php">
    Adı :<input type="text" name="adi" value="ali"><br>
    Soyadı:<input type="text" name="soyadi" value="ozturk"><br>
    <input type="submit" value="GÖNDER">
</form>
```

Adı :

Soyadı:



Formlar

➤ <input type="reset">

- Form elemanlarının içeriklerinin temizleyerek varsayılan değerlerine getirilmesini sağlar

```
<input type="reset" value="BURAYA TIKLAYIN!">
```



BURAYA TIKLAYIN!

Formlar

➤ <input type="button">

- Buton tanımı yapılır

```
<input type="button" value="BURAYA TIKLAYIN!">
```



BURAYA TIKLAYIN!

Formlar

➤ <input type="password">

- Şifre metin kutusu tanımı yapılır.

```
<form>
    Kullanıcı Adı:<br>
    <input type="text" name="kullanici"><br>
    Şifre:<br>
    <input type="password" name="sifre">
</form>
```

Kullanıcı Adı:

Şifre:

Formlar

➤ <input type="email">

- Elektronik posta metin kutusu tanımı yapılır.

```
<form>
    Kullanıcı Adı:<br>
    <input type="text" name="adi_soyadi"><br>
    Elektronik Posta Adresi<br>
    <input type="email" name="email">
</form>
```

Adı Soyadı:

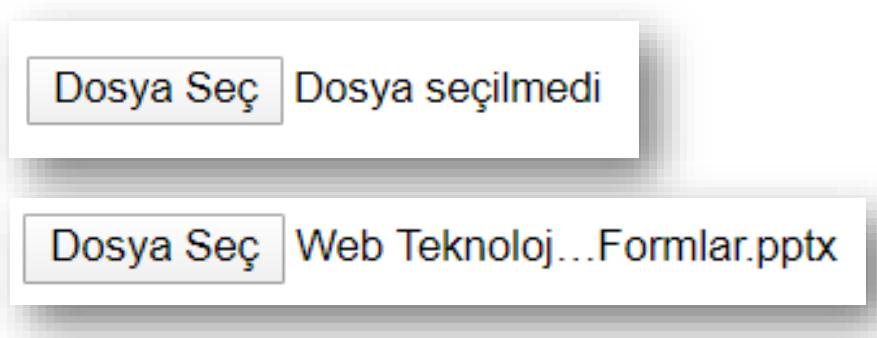
Elektronik Posta Adresi

Formlar

➤ <input type="file">

- Dosya yüklemek için dosya seçme alanı ve <gözat> butonu içerir.

```
<form>
  <input type="file" name="dosya">
</form>
```



Formlar

- **multiple** özelliği elektronik posta ve dosya veri giriş elemanlarına birden fazla veri yüklenmesini sağlar.

```
<form>
  <input type="file" name="dosya" multiple>
</form>
```

Dosyaları Seç Web Teknolo...Giris.pptx

Formlar

➤ <select> etiketi

- seçmeli (drop-down) liste tanımlanır.
- selected özelliği kullanıldığı seçeneği otomatik seçili olarak getirir.

```
<select name="arabalar">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat" selected>Fiat</option>
  <option value="audi">Audi</option>
</select>
```

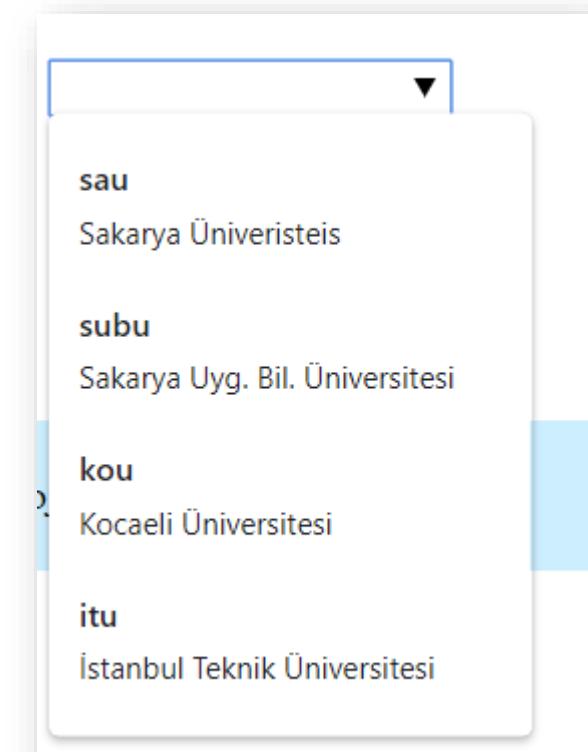


Formlar

➤ <input type="list">

- Form veri giriş elemanı için <**datalist**> etiketi içerisinde önceden tanımlanmış seçenekler içerir.

```
<input type="text" list="universiteler">
<datalist id="universiteler">
    <option value="sau">
        Sakarya Üniveristesi
    </option>
    <option value="subu">
        Sakarya Uyg. Bil. Üniversitesi
    </option>
    <option value="kou">
        Kocaeli Üniversitesi
    </option>
    <option value="itu">
        İstanbul Teknik Üniversitesi
    </option>
</datalist>
```

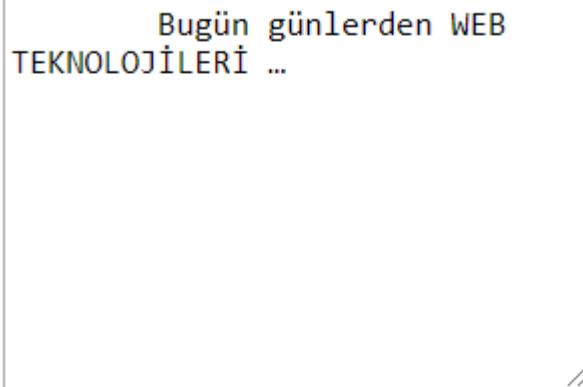


Formlar

➤ <textarea>

- Çok satırlı metin girişini tanımlar

```
<textarea name="mesaj" rows="10" cols="30">  
Bugün günlerden WEB TEKNOLOJİLERİ ...  
</textarea>
```



Formlar

➤ **value** özelliği form elemanın içeriğini belirler.

```
<form action="">
Adı:<br>
<input type="text" name="adi" value="Ali"><br>
Soyadı:<br>
<input type="text" name="soyadi">
</form>
```

Adı : ALİ

Soyadı:



Formlar

- **readonly** özniteliği kullanıldığı yerde form elemanını sadece okunabilir yapar.

```
<form action="">
Adı:<br>
<input type="text" name="adi" value="Ali" readonly><br>
Soyadı:<br>
<input type="text" name="soyadi">
</form>
```

Adı :

Soyadı:

Formlar

- **disabled** özelliği kullanıldığı yerde elemanı pasif yapar.

```
<form action="">
  Adı:<br>
  <input type="text" name="adi" value="Ali" disabled><br>
  Soyadı:<br>
  <input type="text" name="soyadi">
</form>
```

Adı : ALİ

Soyadı:

Formlar

- **required** özelliği form gönderilmeden önce doldurulması gerektiğini belirtir.

```
<form action="">
  Adı:<br>
  <input type="text" name="adi" required><br>
  Soyadı:<br>
  <input type="text" name="soyadi"><br><br>
  <input type="submit" value="GÖNDER">
</form>
```

The image shows a simple HTML form. It contains two text input fields labeled 'Adı:' and 'Soyadı:', both of which are currently empty. Below the first input field, there is a small callout-style message box with a yellow exclamation mark icon. The message text is 'Lütfen bu alanı doldurun.' (Please fill this field). At the bottom of the form is a single button labeled 'GÖNDER'.

Formlar

- **autofocus** özelliği sayfa yüklendiğinde form elemanına imlecin odaklanması sağlar.

```
<form action="">
Adı:<br>
<input type="text" name="adi" autofocus><br>
Soyadı:<br>
<input type="text" name="soyadi"><br><br>
<input type="submit" value="GÖNDER">
</form>
```

The image shows a simple HTML form. It consists of two text input fields, one for 'Adı:' and one for 'Soyadı:', both with blue borders. Below these is a large orange 'GÖNDER' button. The entire form is contained within a white rectangular area with a slight shadow.



Formlar

- **autocomplete** özelliği form elemanına veri girilirken otomatik tamamlama modunu "**on**"/"**off**" olarak belirtir.

```
<form action="">  
Adı:<br>  
<input type="text" name="adi" autocomplete="on"><br>  
Soyadı:<br>  
<input type="text" name="soyadi"><br><br>  
<input type="submit" value="GÖNDER">  
</form>
```

Adı:

Gülüzar

Can



Formlar

- **placeholder** özelliği kullanıcının veri gireceği form elemanın içerisinde o alana girilecek veri ile ilgili bir ipucu içerir.

```
<form action="">  
Adı:<br>  
<input type="text" name="adi" placeholder="Adınızı Girin..."><br>  
Soyadı:<br>  
<input type="text" name="soyadi"><br><br>  
<input type="submit" value="GÖNDER">  
</form>
```

Adı:

Soyadı:

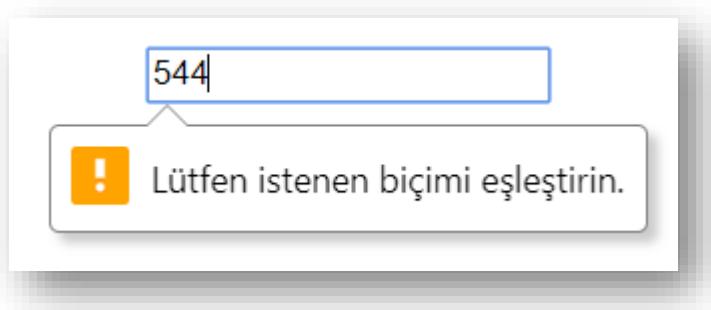
GÖNDER



Formlar

- **pattern** özelliği kullanıcının veri gireceği form elemanı için bir örüntü tanımlamasını sağlar.

```
<form action="">  
    İl Plaka Kodu:<br>  
    <input type="text" name="plaka" pattern=[0-9]{2}><br><br>  
    <input type="submit" value="GÖNDER">  
</form>
```



Formlar

➤ **size** özelliği form elemanın boyutunu belirler.

```
<form action="">  
Adı:<br>  
<input type="text" name="adi" value="Ali" size="30">  
<br>  
Soyadı:<br>  
<input type="text" name="soyadi">  
</form>
```

The image shows a simple HTML form with two text input fields. The first field is labeled "Adı:" and contains the value "ALİ". The second field is labeled "Soyadı:" and is empty. The form is enclosed in a light gray box with rounded corners.



Formlar

- **id** özelliği form elemanına erişilebilmesini (JavaScript kodu ile , vs.) sağlar.

```
<form action="">  
Adı:<br>  
<input type="text" name="adi"  
      id="adi" value="Ali" size="30"><br>  
Soyadı:<br>  
<input type="text" name="soyadi">  
</form>
```

Adı : ALİ

Soyadı:

Formlar

- Forma ait **action** özelliği form gönderildiği zaman web sayfasının yönleneceği URL adresini gösterir.

```
<form action="action_page.php">  
    Adı: <input type="text" name="adi"><br>  
    Soyadı: <input type="text" name="soyadi"><br>  
    <input type="submit" value="GÖNDER"><br>  
</form>
```



Formlar Özellikleri

- Forma ait **formenctype** özelliği formun gönderilme şeklini belirtir.
- Bu şekilde form ile bir dosya verisi gönderilebilir.

```
<form action="action_page.php" formenctype="multipart/form-data">
    Adı: <input type="text" name="adi"><br>
    Soyadı: <input type="text" name="soyadi"><br>
    <input type="submit" value="GÖNDER">
</form>
```

Form Özellikleri

- **method** özelliği formun gönderdiği bilginin gönderilme yöntemini belirtir.
- **get** ve **post** olmak üzere iki yöntem vardır.
 - get açık bir şekilde form verilerini gönderir.
 - post kapalı bir şekilde form verilerini gönderir.

```
<form action="iletisim.php" method="post">
    Adı: <input type="text" name="adi"><br>
    Soyadı: <input type="text" name="soyadi"><br>
    <input type="submit" value="GÖNDER">
</form>
```

Form Özellikleri

- **formaction** özelliği veri gönderme elemanın formun form gönderildiği zaman web sayfasının yönleneceği URL adresini gösterir.
- **formmethod** özelliği veri gönderme elemanın hangi metod ile gönderileceğini belirtir.

```
<form action="iletisim.php" method="post">
    Adı: <input type="text" name="adi"><br>
    Soyadı: <input type="text" name="soyadi"><br>
    <input type="submit" value="GÖNDER">
    <input type="submit"
        formaction="iletisim2.php"
        formmethod="post"
        value="POST İLE GÖNDER">
</form>
```

The screenshot shows a web page with a form. It has two text input fields: one for 'Adı' (Name) and one for 'Soyadı' (Surname). Below these are two submit buttons: a standard 'GÖNDER' button and a 'POST İLE GÖNDER' button. This visualizes the code where multiple submit buttons can be defined for a single form.

Form Bilgisi Gönderme/Alma

➤ PHP Nedir?

- Sunucu taraflı işlemleri yönetmek amacıyla geliştirilmiş HTML içine gömülebilen bir betik dil
 - Form verisi toplamak
 - Sunucuda dosya yönetmek
 - Veritabanlarını düzenlemek
 - ...

➤ WAMP Server Nedir?

- Windows işletim sistemine Apache, MySQL ve PHP yükleyen ve bilgisayarınızda sanal sunucu gibi davranış gösteren bir yazılım
- Wamp sunucusunu kurduktan ve ayarlarınızı yaptıktan sonra klasörün içinde **www** dizinine attığınız tüm PHP ve diğer türdeki dosyalara **http://localhost** adresinden ulaşabilirsiniz.

Form Bilgisi Gönderme/Alma

➤ Post Metodu İle Form Gönderme - 1

```
<form action="form1_post.php" method="post">
  <table cellspacing="8">
    <tr>
      <td>ADI</td><td>:</td>
      <td><input type="text" name="adi"></td>
    </tr>
    <tr>
      <td>SOYADI</td><td>:</td>
      <td><input type="text" name="soyadi"></td>
    </tr>
  <table>
<form>
```

POST METODU İLE FORM GÖNDERME

ADI :

SOYADI :

Form Bilgisi Gönderme/Alma

➤ Post Metodu İle Form Alma - 1

```
<table>
  <tr>
    <td>ADI SOYADI</td><td>:</td>
    <td><b>
      <?php
        echo $_POST['adi']." ".$_POST['soyadi'];
      ?>
      <b></td>
    </tr>
</table>
```



localhost/formlar/form1_post.php

POST METODU İLE FORM ALMA

ADI SOYADI : **Gülüzar ÇIT**

Form Bilgisi Gönderme/Alma

➤ Post Metodu İle Form Gönderme - 2

```
<form action="form2_post.php" method="post">
  <table cellspacing="8">
    <tr>
      <td>ADI</td><td>:</td>
      <td><input type="text" name="adi"></td>
    </tr>
    <tr>
      <td>SOYADI</td><td>:</td>
      <td><input type="text" name="soyadi"></td>
    </tr>
    <tr>
      <td colspan="3">Şartları Onaylıyorum:
        <input type="checkbox" name="onay" value="1">
      </td>
    </tr>
  </table>
</form>
```

POST METODU İLE FORM GÖNDERME

ADI :

SOYADI :

Şartları Onaylıyorum:

GÖNDER **TEMİZLE**

Form Bilgisi Gönderme/Alma

➤ Post Metodu İle Form Alma - 2

```
<?php
    if (isset($_POST['onay'])){
        if (isset($_POST['adi'], $_POST['soyadi']) )
            && ($_POST['adi']!='')
            && ($_POST['soyadi']!='')) {
                echo "Kayıt İşleminiz Yapıldı...";
                echo "<b>".$_POST['adi'] . ' ' . $_POST['soyadi']."</b>";
            }else{
                echo "Bilgilerinizi Kontrol Edip Tekrar Giriş Yapın.<br>
                    Eksik Bilgi Girdiniz...";
            }
    }else{
        echo "Lütfen Onay Kutusunu İşaretleyiniz!...";
    }
    echo "<p> <a href='form2_post.html'>&lt;GERİ DÖN&gt;</a></p> ";
?
}
```

POST METODU İLE FORM ALMA - 2

Kayıt İşleminiz Yapıldı...**Gülizar ÇIT**

[GERİ DÖN](#)

Form Bilgisi Gönderme/Alma

➤ Get Metodu İle Form Gönderme

```
<form action="form1_get.php" method="get">
  <table cellspacing="8">
    <tr>
      <td>ADI</td><td>:</td>
      <td><input type="text" name="adi"></td>
    </tr>
    <tr>
      <td>SOYADI</td><td>:</td>
      <td><input type="text" name="soyadi"></td>
    </tr>
  </table>
</form>
```

GET METODU İLE FORM GÖNDERME

ADI :

SOYADI :

Form Bilgisi Gönderme/Alma

➤ Get Metodu İle Form Alma

```
<table>
  <tr>
    <td>ADI SOYADI</td><td>:</td>
    <td><b>
      <?php
        echo $_GET[ 'adi' ]."&ampnbsp".$_GET[ 'soyadi' ];
      ?>
      <b></td>
    </tr>
</table>
```



KAYNAKLAR

➤ <https://www.w3schools.com/>



Web Teknolojileri

HAFTA 5

HTML 5

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKİ
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

➤ HTML5



HMTL5

- HTML'nin beşinci ve günümüzdeki son versiyonudur.
- 2014'te resmi olarak tanıtılmıştır.
- HTML5 ile gelen en dikkat çekici özellikler
 - Semantik etiketler: <header>, <footer>, <article>, <section>
 - form etiketine yeni özellikler: number, date, time, calendar
 - Grafik etiketleri: <svg>, <canvas>
 - Multimedya etiketleri: <audio> , <video>

HMTL5...

- HTML5 ile gelen en dikkat çekici uygulama ara yüzleri
 - HTML Geolocation: Kullanıcının konumunu almayan yarar.
 - HTML Drag ve Drop: Nesneleri sürükle bırak özelliği
 - HTML Local Storage: Tarayıcıda verinin lokal olarak saklanması
 - HTML Application Cache: Web uygulaması ön belleğe alınıp internet bağlantısı kesildikten sonra da erişilebilir.
 - HTML Web Workers: Script çalıştığı sürece sayfa devre dışı bırakılabilir.
 - HTML SSE (Server Sent Events): Sunucunun sürekli olarak DOM olaylarını (event) web tarayıcısında aktif tutabilmesine olanak sunar.

HMTL5...

- HTML5 ile kaldırılan etiketlerden bazıları
 - <center>
 -
 - <frame>
 - <frameset>
 - <big>

HMTL5...

- HTML5 ile özellikler (attribute) için yeni gelen 4 farklı yazım şekli
 - Yazma engellenmiş
 - Tırnak olmadan
 - Çift tırnak
 - Tek tırnak

```
<input type="text" value="Sakarya Üniversitesi" disabled>
  
  
<input type="text" value=Sakarya>
  
  
<input type="text" value="Sakarya Üniversitesi">
  
  
<input type="text" value='Sakarya Üniversitesi'>

```

➤ Sayfada video gösterimi

```
<video width="320" height="240" controls>
  <source src="video/example2.mp4" type="video/mp4">
  <source src="video/example3.avi" type="video/avi">
  <source src="video/example4.ogg" type="video/ogg">
  TARAYICINIZ VIDEO ETIKETINI DESTEKLEMİYOR...
</video>
```



HMTL5...

➤ Sayfada ses çalınması

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  TARAYICINIZ AUDIO ETIKETINI DESTEKLEMIYOR...
</audio>
```



➤ Sayfaya resim eklenmesi

```
<figure>
  
  <figcaption>Resim.1 - İstanbul, TÜRKİYE</figcaption>
</figure>
```



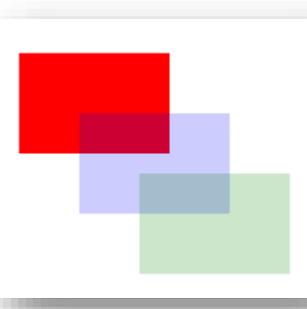
Resim.1 - İstanbul, TÜRKİYE

➤ Sayfada grafik oluşturulması

```
<!DOCTYPE html>
<html>
<body>
    <canvas id="myCanvas">
        TARAYICINIZ CANVAS ETIKETINI DESTEKLEMIYOR...
    </canvas>

    <script>
        var c = document.getElementById("myCanvas");
        var ctx = c.getContext("2d");
        ctx.fillStyle = "red";
        ctx.fillRect(20, 20, 75, 50);

        //Turn transparency on
        ctx.globalAlpha = 0.2;
        ctx.fillStyle = "blue";
        ctx.fillRect(50, 50, 75, 50);
        ctx.fillStyle = "green";
        ctx.fillRect(80, 80, 75, 50);
    </script>
</body>
</html>
```



➤ HTML4'ten HTML5'e Geçiş

HTML4	HTML5
<div id="header">	<header>
<div id="menu">	<nav>
<div id="content">	<section>
<div class="article">	<article>
<div id="footer">	<footer>

➤ HTML4'ten HTML5'e Geçiş...

HTML4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5

```
<!DOCTYPE html>
```



➤ Yeni Etiketler ile Örnek Sayfa

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
  </head>
  <body>
    <header>
      <h1>Başlık</h1>
    </header>
    <nav>
      <ul>
        <li><a href="#">Menu 1</a></li>
        <li><a href="#">Menu 2</a></li>
        <li><a href="#">Menu 3</a></li>
      </ul>
    </nav>
    <section>
      <article>
        <header>
          <h1>Makale 1</h1>
        </header>
```

➤ Yeni Etiketler ile Örnek Sayfa...

```
<section>
    Bu birinci makale. <mark>Bu kısım</mark> vurgulanmış.
</section>
</article>
<article>
    <header>
        <h1>Makale 2</h1>
    </header>
    <section>
        Bu ikinci makale. Bu sekilde makaleler devam edebilir.
    </section>
</article>
</section>
<aside>
    <section>
        <h1>Sosyal Ağ Adresleri</h1>
        <ul>
            <li><a href="#">Facebook</a></li>
            <li><a href="#">Twitter</a></li>
            <li><a href="#">Instagram</a></li>
        </ul>
    </section>
    <figure>
        
        <figcaption>HTML5</figcaption>
    </figure>
</aside>
<footer>Footer - Copyright 2017</footer>
</body>
</html>
```

➤ Yeni Etiketler ile Örnek Sayfa...

Başlık

- [Menu 1](#)
- [Menu 2](#)
- [Menu 3](#)

Makale 1

Bu birinci makale. Bu **kısim** vurgulanmış.

Makale 2

Bu ikinci makale. Bu şekilde makaleler devam edebilir.

Sosyal Ağ Adresleri

- [Facebook](#)
- [Twitter](#)
- [Instagram](#)



HTML5

Footer - Copyright 2017



KAYNAKLAR

➤ <https://www.w3schools.com/>



Web Teknolojileri

Ders 6

CSS (1/2)

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKİ
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

■ CSS – Cascading Style Sheets

- CSS Nedir?
- Stil Şablonları
- Stil Tanımlama
- Stil Hiyerarşisi
- SPAN Etiketi
- DIV Etiketi
- Örnekler

CSS Nedir?

- Cascading Style Sheets / Basamaklı Stil Şablonları(Sayfaları)
- Web sayfalarındaki HTML etiketlerinin web sayfalarının sadece içeriğini tanımlamak için oluşturulmuş bir işaretleme dili olduğundan yola çıkarak görsellik adına bu kısıtlı imkanları daha da geliştirmek, sayfa tasarımlarını daha etkili hale getirmek ve kolay tasarlanabilir, esnek yapmak amacıyla W3C (World Wide Web Consortium) tarafından geliştirilmiştir.
- HTML elemanlarının ekranda **nasıl** görüneceğini tanımlamak için kullanılırlar
- Pek çok web sayfasının düzenini aynı anda kontrol edebilir.
- Farklı cihazlar ve ekran boyutları için web sayfanızın ekranının tasarım, anahat ve versiyonlarını içeren stilleri tanımlar

Stil Şablonları

- + Stil şablonları tek bir etiket üzerinde tanımlanabileceği gibi, tüm sayfada geçerli olacak şekilde de tasarlabilir veya stil dosyaları ile istenilen sayfada stil şablonları kullanılabilir.
- + Stil şablonları tanımlandıkları yere göre üç farklı şekilde kullanılabilirler.
 - + Yerel Stil tanımlamaları (Satır İçi Stiller)
 - + Global Stil tanımlamaları (Gömülü Stiller)
 - + Bağlantılı Stil tanımlamaları (Harici Stil Dosyaları)

Yerel Stil Tanımlamaları

■ Yerel Stil tanımlamaları (Satır İçi Stiller)

- Sayfa içerisindeki etiket içerisinde **style** özelliği kullanılarak tanımlanırlar.
- Bu stil tanımları sadece tanımlandığı yerde geçerlidir.

```
<p>Stilsiz paragraf</p>
<p style="font-size:18pt; color:#C4A9A9">Stil verilmiş paragraf</p>
```

Stilsiz paragraf

Stil verilmiş paragraf

Global Stil Tanımlamaları

■ Global Stil tanımlamaları (Gömülü Stiller)

- Sayfa içerisinde geçerli olacak stil tanımlamaları için kullanılır.
- Sayfanın **head** kısmında **<style>** ile **</style>** etiketleri arasında tanımlanır.
- Etiketlere tanımlanan stiller sayfa içerisinde kullanılan aynı etiketlerin tümüne aynı şekilde etki ederler.

Global Stil Tanımlamaları...

```
<html>
<head>
<meta charset="utf-8">
<title>Örnekler</title>
<style type="text/css">
p{
    font-size:18pt;
    color:#C4A9A9
}
</style>
</head>
<body>
<p>Stil etki eden ilk paragraf</p>
stil etki etmeyen metin
<p>Stil etki eden ikinci paragraf</p>
</body>
</html>
```

Stil etki eden ilk paragraf

stil etki etmeyen metin

Stil etki eden ikinci paragraf

Bağlantılı Stil tanımlamaları

■ Bağlantılı Stil Tanımlamaları (Harici Stil Dosyaları)

- Sayfa içerisinde geçerli olacak stil tanımlamaları **.css** uzantılı dosyalara kaydedilerek sayfanın **head** kısmında **<link>** etiketi içerisinde çağrılır.
- Tanımlanan **css** dosyaları web sitesi içerisinde istenilen web sayfalarında kullanılabilir.
- Link etiketinin kullanımı:

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

Bağlantılı Stil Tanımlamaları...

```
li{  
    font-size:18pt;  
    color:#4F3977;  
}  
  
<html>  
    <head>  
        <meta charset="utf-8">  
        <title>Örnekler</title>  
        <link rel="stylesheet" type="text/css" href="sitilimiz.css">  
    </head>  
    <body>  
        <ol>Stiller  
            <li>Yerel</li>  
            <li>Global</li>  
            <li>Bağlantılı</li>  
        </ol>  
    </body>  
</html>
```



Stiller
1. Yerel
2. Global
3. Bağlantılı

Stil Tanımlama

- + Stil tanımlaması, bir seçici ve bir tanımlama bloğu kullanılarak yapılır.
- + Seçici, biçimlendirmek istediğiniz HTML elemanına işaret eder.
- + Tanımlama bloğu süslü parantezler ("{" ve "}") içerisinde yer alır.
- + "{" sembolünden sonra seçici üzerine etki edecek stilin özelliği ":" sembolünden sonra değeri belirlenir.
- + Birden fazla özellik kullanılacağı zaman ";" kullanılır.

```
seçici{  
    özellik1:değer1;  
    özellik2:değer2;  
    ....  
}
```

```
p{  
    font-size:18pt;  
    color:#C4A9A9;  
    text-align: center;  
    color: red;  
}
```

Stil Tanımlama...

- Seçici, biçimlendirmek istediğiniz HTML elemanına işaret eder.
- Seçiciler, ilgili HTML elemanlarını isimlerine, kimliklerine (id), sınıflarına, özelliklerine, vs. bakarak bulmak için kullanılırlar
- Eleman Seçicisi Kullanarak Stil Tanımlama
 - Stili tanımlanacak olan HTML elemanınının ismi kullanılır

```
p{  
    font-size:18pt;  
    color:#C4A9A9;  
    text-align: center;  
    color: red;  
}
```

Stil Tanımlama...

■ ID Seçicisi Kullanarak Stil Tanımlama

■ Sayfa içerisindeki bir öğeye ID özelliğini kullanarak stil tanımlamak için seçici isminden önce "#" simbolü kullanılır.

< p > stilden etkilenmeyen metin </ p >

< p id="idili" > stilden etkilenen metin </ p >

```
#idili{  
font-size:18pt;  
color:#48098C;  
}
```

stilden etkilenmeyen metin

stilden etkilenen metin

CLASS Stil Tanımlama

■ Sayfa içerisindeki bir öğeye CLASS özelliğini kullanarak stil tanımlamak için seçici isminden önce ":" simbolü kullanılır.

```
<p>stilden etkilenmeyen metin</p>
```

```
<p class="sinifli">stilden etkilenen metin</p>
```

```
.sinifli
```

```
{  
font-size:8pt;  
color:#06641D;  
}
```

stilden etkilenmeyen metin

stilden etkilenen metin

Stil Hiyerarşisi

- + Stil tanımlamaları yapılırken bir öğeyi etkileyen birden fazla stil tanımlaması olabilir.
- + Bir öğe içerisinde önce öğe içerisinde tanımlanmış stil özellikleri geçerli olacaktır. Eğer öğe içerisinde stil tanımlaması bulunmuyorsa öğenin içerisinde bulunduğu bölümün stil tanımlaması varsa bu stil özellikleri geçerli olacaktır.

Stil Hiyerarşisi

```
<ol id="idili">Stiller  
  <li class="sinifli">Yerel</li>  
  <li>Global</li>  
  <li>Bağlantılı</li>  
</ol>
```

Stiller

1. Yerel
2. Global
3. Bağlantılı

Sık Kullanılan Stil Özellikleri

Arka Plan Stil Özellikleri

- Stil tanımlanacak etiketlerin arka plan özellikleri belirlemek amacıyla kullanılırlar.

background-color: arka plan rengini belirlemek amacıyla kullanılır.

background-image: arka planda bulunması istenen resmin belirlenmesi için kullanılır.

Sık Kullanılan Stil Özellikleri

Arka Plan Stil Özellikleri:

```
body {background-color:#D9F7D1; }  
.arkaplan { background-color:#F0CECF; }  
.resimArkaplan {background-image:url(arkaplan.png); }
```

```
<body>  
<p class="arkaplan">Arka planı renklendirilmiş paragraf</p>  
<div class="resimArkaplan">Arkaplanda resim olan div</div>
```

Arka planı renklendirilmiş paragraf

Arkaplanda resim olan div

Sık Kullanılan Stil Özellikleri

Font Stil Özellikleri

Sayfada yer alacak olan yazıların font ayarlarını gerçekleştirmek amacıyla kullanılan stil özellikleridir.

font-family: Yazı fontunun tipini belirlemek amacıyla kullanılır. (Arial, Verdana gibi)

font-size: Yazı fontunun büyüğünü belirler. (px ile piksel, pt ile punto olarak büyülüklük belirlenebilir.)

color: Yazı rengini belirlemek için kullanılır.

Sık Kullanılan Stil Özellikleri

Font Stil Özellikleri:

```
body { background-color:#D9F7D1;  
       font-family:Cambria;  
       font-size:14px;  
       color:#48098C; }  
  
.arkaplan {  
       background-color:#F0CECF;  
       font-family:Cambria;  
       font-size:12px;  
       color:#790AF3; }  
  
.resimArkaplan {  
       background-  
       image:url(arkaplan.png);  
       font-family:Cambria;  
       font-size:10px;  
       color:#600305; }
```

```
<body>  
İçerik yazısı  
<p class="arkaplan">Arka planı renklendirilmiş  
paragraf</p>  
<div class="resimArkaplan">Arkaplanda resim  
olan div</div>
```

İçerik yazısı

Arka planı renklendirilmiş paragraf

Arkaplanda resim olan div

Sık Kullanılan Stil Özellikleri

Metin konumlandırma

 **text-align:** Metinlerin bulunduğu etiket içerisinde konumunu belirlemek amacıyla kullanılan özellikdir.

left: Yazının sola hizalanmasını sağlar.

center : Yazının ortaya hizalanmasını sağlar.

right : Yazının sağa hizalanmasını sağlar.

justify: Yazının iki yana hizalanmasını sağlar.

Sık Kullanılan Stil Özellikleri

Font Stil Özellikleri:

```
body { .... text-align:left; }  
.arkaplan { ..... text-align:center; }  
.resimArkaplan { ..... text-align:right; }
```

```
<body>  
İçerik yazısı  
<p class="arkaplan">Arka planı renklendirilmiş  
paragraf</p>  
<div class="resimArkaplan">Arkaplanda resim  
olan div</div>
```

İçerik yazısı

Arka planı renklendirilmiş paragraf

Arkaplanda resim olan div

Sık Kullanılan Stil Özellikleri

Yükseklik ve Genişlik belirleme

- width: etiketin genişlik değerini belirlemek için kullanılır.
height: etiketin yükseklik değerini belirlemek için kullanılır.
- İçerisinde bulunan değere göre yüzdelik olarak belirlenebilir.
(50%, 20% gibi)
- Piksel değeri olarak belirlenebilir. (20, 50 gibi)

Sık Kullanılan Stil Özellikleri

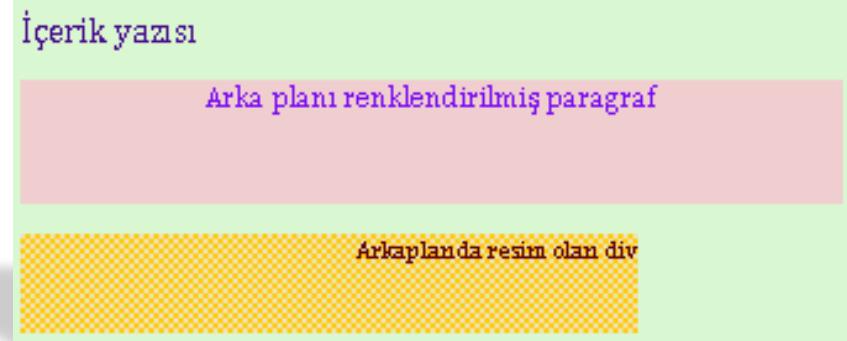
Font Stil Özellikleri:

```
body { .... text-align:left; }

.arkaplan { ..... text-align:center;
width:20%;
height:50px; }

.resimArkaplan { ..... text-align:right;
width:15%;
height:40px; }
```

```
<body>
İçerik yazısı
<p class="arkaplan">Arka planı renklendirilmiş
paragraf</p>
<div class="resimArkaplan">Arkaplanda resim
olan div</div>
```



SPAN Etiketi

- Web sayfalarını düzenlemek amacıyla kullanılan öğelerden biri de **span** elemanıdır.
- span** etiketi kullanılarak web sayfasının belirli parçalara bölünmesi ve bu parçaların içerişine içeriklerin eklenerek gösterilmesi sağlanır.
- span** etiketi satır içi temelli bir elemandır. İçerisine eklenen değerlerin büyüklüğü kadar genişliğe sahip olabilir.
- span** etiketleri art arda eklendiğinde yan yana gösterilirler.
- span** etiketleri stiller kullanarak düzenlenebilir.

SPAN Etiketi...

```
<span>Programlamaya Giriş</span>
<span>Veri Yapıları</span>
<span>Web Teknolojileri</span>
<span>Veritabanı Yönetim Sistemleri</span>
```

```
span{
    background-color:#4F3977;
    color:#ECDEDE;
    font-family:"Gill Sans", Helvetica, Arial, sans-serif;
    font-size:12px;
}
```

Programlamaya Giriş Veri Yapıları Web Teknolojileri Veritabanı Yönetim Sistemleri

DIV Etiketi

- Web sayfalarını düzenlemek amacıyla kullanılan öğelerden biri de **div** etiketidir.
- div** etiketi kullanılarak web sayfasının belirli parçalara bölünmesi ve bu parçaların içerisine içeriklerin eklenerek gösterilmesi sağlanır.
- div** etiketi blok temelli bir elemandır. İçerisine eklenen değerleri bir blok içerisinde gösterir.
- div** etiketlerini yan yana gösterebilmek için **float** elemanı kullanmak gerekmektedir.
- div** elemanları stiller kullanarak düzenlenebilir.

DIV Etiketi...

```
<div>Programlamaya Giriş</div>
<div>Veri Yapıları</div>
<div>Web Teknolojileri</div>
<div>Veritabanı Yönetim Sistemleri</div>
```

Programlamaya Giriş
Veri Yapıları
Web Teknolojileri
Veritabanı Yönetim Sistemleri

DIV Etiketi...

```
<div>Programlamaya Giriş</div>
<div>Veri Yapıları</div>
<div>Web Teknolojileri</div>
<div>Veritabanı Yönetim Sistemleri</div>
```

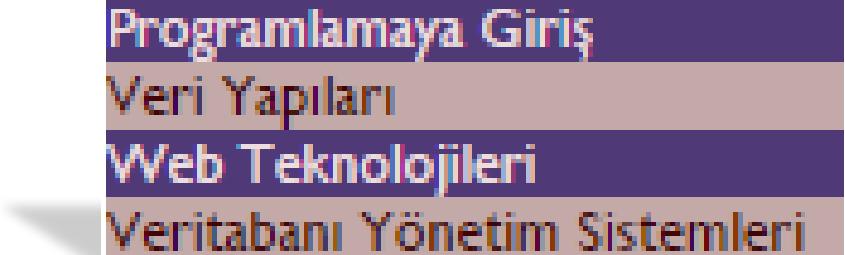
```
div{
    background-color:#7961A4;
    color:#ECDEDE;
    font-family:"Gill Sans", "Gill Sans MT", Helvetica, Arial, sans-serif;
    font-size:12px;
}
```

Programlamaya Giriş
Veri Yapıları
Web Teknolojileri
Veritabanı Yönetim Sistemleri

DIV Etiketi...

```
<div class="t">Programlamaya Giriş</div>
<div class="c">Veri Yapıları</div>
<div class="t">Web Teknolojileri</div>
<div class="c">Veritabanı Yönetim Sistemleri</div>
```

```
div{
    font-family:"Gill Sans", Helvetica, Arial, sans-serif;
    font-size:12px;
}
div.t{
    background-color:#4F3977;
    color:#ECDEDE;
}
div.c{
    background-color:#C4A9A9;
    color:#410506;
}
```



Programlamaya Giriş
Veri Yapıları
Web Teknolojileri
Veritabanı Yönetim Sistemleri

DIV Etiketi ile Yerleşim

Web sayfamızda art arda div etiketleri kullanıldığında alt alta gösterilecektir.

```
.bolumler{  
font-family:"Gill Sans", Helvetica,  
Arial, sans-serif;  
font-size:12px;  
background-color:#C4A9A9;  
color:#410506;  
width:15%;  
}
```

```
.dersler{  
font-family:"Gill Sans", Helvetica,  
Arial, sans-serif;  
font-size:10px;  
background-color:#4F3977;  
color:#ECDEDE;  
width:20%
```

DIV Etiketi ile Yerleşim...

```
<div class="bolumler">  
    <div>Bilgisayar Mühendisliği</div>  
    <div>Bilişim Sistemleri Mühendisliği</div>  
</div>  
  
<div class="dersler">  
    <div>Programlamaya Giriş</div>  
    <div>Veri Yapıları</div>  
    <div>Web Teknolojileri</div>  
    <div>Veritabanı Yönetim Sistemleri</div>  
</div>
```

Bilgisayar Mühendisliği
Bilişim Sistemleri Mühendisliği
Programlamaya Giriş
Veri Yapıları
Web Teknolojileri
Veritabanı Yönetim Sistemleri

DIV Etiketi ile Yerleşim...

- + Web sayfamızda art arda div etiketleri kullanıldığında eğer yan yana gösterilmek isteniyorsa **float** özelliği left (solda) veya right (sağda) seçilir.
- + **float** özelliğinin sıfırlanması için **clear** özelliği left veya right seçilir. Her ikisinin sıfırlanması için bu özellik both olarak seçilir.

DIV Etiketi ile Yerleşim...

```
.bolumler{  
font-family:"Gill Sans", Helvetica, Arial, sans-serif;  
font-size:12px;  
background-color:#C4A9A9;  
color:#410506;  
width:15%;  
float:left;  
}  
.dersler{  
font-family:"Gill Sans", Helvetica, Arial, sans-serif;  
font-size:10px;  
background-color:#4F3977;  
color:#ECDEDE;  
width:20%;  
float:left;  
}
```

DIV Etiketi ile Yerleşim...

```
<div class="bolumler">
    <div>Bilgisayar Mühendisliği</div>
    <div>Bilişim Sistemleri Mühendisliği</div>
</div>
<div class="dersler">
    <div>Programlamaya Giriş</div>
    <div>Veri Yapıları</div>
    <div>Web Teknolojileri</div>
    <div>Veritabanı Yönetim Sistemleri</div>
</div>
```

Bilgisayar Mühendisliği
Bilişim Sistemleri Mühendisliği

Programlamaya Giriş
Veri Yapıları
Web Teknolojileri
Veritabanı Yönetim Sistemleri

DIV Etiketi ile Yerleşim...

```
.bolumleri{  
    font-family:"Gill Sans", Helvetica, Arial, sans-serif;  
    font-size:12px;  
    background-color:#C4A9A9;  
    color:#410506;  
    width:40%;  
    float:left;  
}  
.dersleri{  
    font-family:"Gill Sans", Helvetica, Arial, sans-serif;  
    font-size:10px;  
    background-color:#4F3977;  
    color:#ECDEDE;  
    width:60%;  
    float:left;  
}
```

DIV Etiketi ile Yerleşim...

```
.universite{  
    font-family:"Gill Sans", Helvetica, Arial, sans-serif;  
    font-size:14px;  
    background-color:#7961A4;  
    color:#410506;  
    width:100%;  
    text-align:center;  
}  
.bolum{  
    font-family:"Gill Sans", Helvetica, Arial, sans-serif;  
    font-size:14px;  
    background-color:#D2EBE7;  
    color:#410506;  
    width:100%;  
    text-align:center;  
}  
.sifirla{  
    clear:both;  
}
```

DIV Etiketi ile Yerleşim...

```
<div class="universite">Sakarya Üniversitesi</div>
<div class="bolumleri">
    <div>Bilgisayar Mühendisliği</div>
    <div>Bilişim Sistemleri Mühendisliği</div>
</div>
<div class="dersleri">
    <div>Programlamaya Giriş</div>
    <div>Veri Yapıları</div>
    <div>Web Teknolojileri</div>
    <div>Veritabanı Yönetim Sistemleri</div>
</div>
<div class="sifirla"></div>
<div class="bolum">Bilgisayar ve Bilişim Bilimleri Fakültesi</div>
```



ÖRNEK

[FİLM DÜNYASI]

[VİZYONDAKİ FİMLER]

[YAKINDA]...

[EN ÇOK BEKLENENLER]...

[SON YORUMLAR]...

Copyright @Ümit KOCABIÇAK



SAKARYA ÜNİVERSİTESİ

BSM 104 Web Teknolojileri 2020-2021 Bahar

ÖRNEK...

```
<div class="ust">[FİLM DÜNYASI]</div>
<div class="bosluk">&nbsp;</div>
<div class="ortaust">[VİZYONDAKİ FİMLER]</div>
<div class="bosluk">&nbsp;</div>
<div class="ortaalt">
    <div class="ortaaltsol">[YAKINDA]...</div>
    <div class="ortabosluk"></div>
    <div class="ortaalttorta">[EN ÇOK BEKLENENLER]...</div>
    <div class="ortabosluk"></div>
    <div class="ortaaltsag">[SON YORUMLAR]...</div>
</div>
<div class="sifirla"></div>
<div class="bosluk"></div>
<div class="alt">Copyright @Ümit KOCABIÇAK</div><div class="alt"></div>
```

ÖRNEK...

```
.ust{  
background-color:#ffcccc;  
color:#660066;  
height:80px;  
padding:5px;  
}  
.ortaust{  
background-color:#eeeeee;  
color:#660066;  
height:180px;  
padding:5px;  
text-align:center;  
}  
.ortaaltsag{  
background-color:#ffcc99;  
color:#660066;  
width:30%;  
height:180px;  
padding:1%;  
float:left;  
}  
.ortaaltorta{  
background-color:#c6aad8;  
color:#660066;  
width:30%;  
height:180px;  
padding:1%;  
float:left;  
}  
}
```

ÖRNEK...

```
.ortaaltsol          }  
 {  
    background-color:#b3d9ff;  
    color:#660066;  
    width:30%;  
    height:180px;  
    padding:1%;  
    float:left;  
 }  
  
.ortabosluk        }  
 {  
    width:2%;  
    height:200px;  
    float:left;  
 }  
  
}                      .alt{  
    background-color:#ffcccc;  
    color:#660066;  
    padding:5px;  
    height:50px;  
 }  
  
.bosluk{  
    width:100%;  
    height:10px;  
 }  
  
.sifirla{clear:both;}
```

ÖRNEK

ARA...

[Üye Ol](#) | [Üye Girişi](#) | [English](#)



[FİLM DÜNYASI]

| [Vizyondakiler](#) || [Yakında](#) || [Salonlar](#) || [Filmler](#) || [İletişim](#) |



SAKARYA ÜNİVERSİTESİ

BSM 104 Web Teknolojileri 2020-2021 Bahar

ÖRNEK...

```
<div class="ust">
  <div>
    <div class="ustara">
      <form><input type="text" name="ara" value="ARA..."></form>
    </div>
    <div class="ustbosluk">&nbsp;</div>
    <div class="stsol">Üye Ol | Üye Girişи | English</div>
  </div>
  <div class="sifirla"></div>
  <div>
    <div class="ustalt">
      &nbsp;&nbsp;&nbsp;[FİLM DÜNYASI]
    </div>
```

ÖRNEK...

```
<div class="altmenu">
  <table>
    <tr>
      <td class="menu">Vizyondakiler</td>
      <td class="menu">Yakında</td>
      <td class="menu">Salonlar</td>
      <td class="menu">Filmler</td>
      <td class="menu">İletişim</td>
    </tr>
  </table>
</div>
</div>
<div class="sifirla"></div>
<div class="bosluk">&nbsp;</div>
</div>

</div>
```

ÖRNEK...

```
.ust{  
    background-color:#ffcccc;  
    color:#660066;  
    height:80px;  
    padding:5px;  
}  
  
.ustara  
{  
    background-color:#ffcccc;  
    color:#660066;  
    width:70%;  
    padding:1%;  
    float:left;  
}  
  
.ustbosluk  
{  
    width:2%;  
    float:left;  
}  
  
.ustsol  
{  
    font-size: 11px;  
    background-color:#ffcccc;  
    color:#660066;  
    padding:1%;  
    float:right;  
}
```

ÖRNEK...

```
.ortaalsol
{
    background-color:#b3d9ff;
    color:#660066;
    padding:1%;
    float:left;
}

.ustalt
{
    background-color:#ffcccc;
    color:#660066;
    padding:2px;
    float:left;
    width:60%;
}

.altmenu
{
    background-color:#ff80aa;
    color:#660066;
    padding:5px;
    float:left;
}

.menu {
    width: 80px;
    text-align: center;
    border-right: 1px solid #660066;
    border-left: 1px solid #660066;
}
```

ÖRNEK...

```
.sifirla{clear:both;}
```

```
.bosluk{  
    width:100%;  
    height:10px;  
}
```

```
input  
{  
    width:300px;  
}
```



Web Teknolojileri

Ders 7

CSS (2/2)

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKİ
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

- CSS – Cascading Style Sheets
- Stil Özellikleri

Stil Özellikleri

➤ Metin Özellikleri

➤ **text-align**

- Metinlerin bulunduğu etiket içerisinde konumunu belirlemek amacıyla kullanılan özelliktir.
- left: Yazının sola hizalanmasını sağlar.
- center : Yazının ortaya hizalanmasını sağlar.
- right : Yazının sağa hizalanmasını sağlar.
- justify: Yazının iki yana hizalanmasını sağlar.

Stil Özellikleri...

➤ Metin Özellikleri...

➤ **text-decoration**

- Metinlerin üstüne veya altına çizgi eklemek amacıyla kullanılır. Eğer altı çizgili metnin bu özelliği kaldırılmak istenirse bu özelliğin kaldırılması gereklidir.
- underline: Yazıyı altı çizgili hale getirir.
- overline: Yazının üstünün çizgili olmasını sağlar.
- line-through: Yazıyı üzeri çizgili hale getirir.
- none: Yazının çizgili halini kaldırır.

Stil Özellikleri...

➤ Metin Özellikleri...

```
.metin{  
background-color:#7961A4;  
color:#ECDEDE;  
height:50px;  
text-align:center;  
text-decoration:line-through;}  
  
a{  
text-decoration:none;  
color:red;}
```

```
<div class="metin">  
Geçici olarak değiştirilmiş metin. Yazı ortada ve  
üzeri çizili  
</div>  
<a href="http://www.sakarya.edu.tr"  
style="text-decoration:none">Sakarya  
üniversitesi linki altı çizgili değil</a>
```

Geçici olarak değiştirilmiş metin. Yazı ortada ve üzeri çizili

Sakarya Üniversitesi linki altı çizgili değil

Stil Özellikleri...

➤ Kenarlık Özellikleri

➤ **border**

- Sayfa içerisindeki bir elemanın kenarlıkla ilgili değerlerini ayarlamak amacıyla kullanılır. Kenarlık kalınlığı, rengi ve sitili ayarlanabilir.
- border-width: Kenarlık kalınlığını ayarlar.
- border-color: Kenarlık rengini ayarlar.
- border-style: Kenarlık sitilini ayarlar.

Stil Özellikleri...

➤ Kenarlık Özellikleri...

```
.kenarlik{  
background-color:#CCEB5B;  
color:#6F0B0B;  
border-width:3px;  
border-color:red;  
border-style:solid;  
width:5%;  
}
```

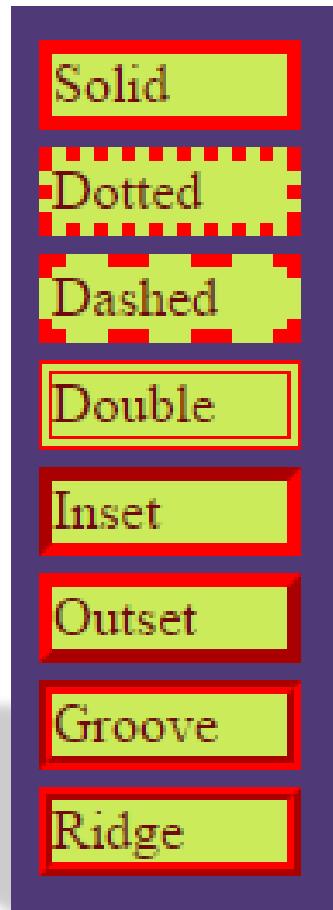
```
<div class="kenarlik">Kenarlık</div>
```

Kenarlık

Stil Özellikleri...

➤ Kenarlık Özellikleri...

➤ border-style



Stil Özellikleri...

➤ Kenarlık Özellikleri...

- CSS3 ile gelen kenar özelliklerinden biri **radius** yani kenarların yumuşatılması, yuvarlaklaştırılmasıdır. Piksel olarak belirlenebileceği gibi yüzdelik değer de verilebilir.

border-radius: 5px;



border-radius: 5%;



Stil Özellikleri...

- Birbiriyle bağlantılı birden fazla özelliği ayrı ayrı tanımlamak yerine tek bir özellik altında tanımlanabilir.
- Ayrı ayrı belirtilen özellik ana özellik altında alt özellik belirtilmeden aralara boşluk bırakılarak artarda tanımlanır.

Stil Özellikleri...

➤ **ÖRNEK:**

- Kenarlık özelliklerinin kalınlık, renk, sitil ve yuvarlaklık özellikleri tek bir özellik altında tanımlanabilir.

```
border-width:5px;  
border-color:#CCEB5B;  
border-style:solid;
```

Aynı Aynı

```
border:5px #CCEB5B double;
```

Tek

Stil Özellikleri...

➤ **Yükseklik ve Genişlik Belirleme**

- width: etiketin genişlik değerini belirlemek için kullanılır.
- height: etiketin yükseklik değerini belirlemek için kullanılır.
- İçerisinde bulunan değere göre yüzdelik olarak belirlenebilir. (50%, 20% gibi)
- Piksel değeri olarak belirlenebilir. (20px, 50px gibi)
- Bazı durumlarda sayfa elemanın genişlik ve yükseklik değeri belirlendiğinde sayfanın boyutlarının kullanıcının değiştirmesi durumunda etiketin belli bir değerin altında küçülmemesi veya belli bir değerin üzerinde büyümemesi için max-width, min-width, max-height, min-height kullanılır.

Stil Özellikleri...

➤ Yükseklik ve Genişlik Belirleme...

```
div.gvm{  
background-color:#C3CC65;  
width:400px;  
margin:10px;  
}  
  
div.gvmm{  
background-color:#C3CC65;  
max-width:450px;  
margin:10px;  
}  
  
div.gvmmm{  
background-color:#C3CC65;  
min-width:350px;  
margin:10px;  
height:  
}
```

```
<div class="gvm">Genişliği verilmiş  
metin</div>  
<div class="gvmm">Genişliği kısıtlanmış  
metin</div>  
<div class="gvmmm">Genişliği kısıtlanmış  
metin</div>
```

Genişliği verilmiş metin

Genişliği kısıtlanmış metin

Genişliği kısıtlanmış metin

Genişliği verilmiş metin

Genişliği kısıtlanmış metin

Genişliği kısıtlanmış metin

Stil Özellikleri...

➤ Taşma Özelliği

➤ **overflow**: Sayfa elemanına genişlik veya yükseklik değeri belirlendiğinde sayfanın boyutlarının kullanıcının değiştirmesi durumunda blok elemanın değeri de değişiktir. Fakat bazı durumlarda blok elemanı içerisinde bulunan elemanlar içerisinde bulunan blok elemanlarının dışına taşıabilir. Bu taşıma için dört özellik belirlenebilir.

➤ visible: taşıma gerçekleşen kısmın taşıdığı kısımda gözükmescini sağlamak için kullanılır.

auto: taşıma gerçekleşen kısmın davranışını otomatik belirlenir.

scroll: taşıma gerçekleşen kısmın dikeyde kaydırma çubuğu çıkarılmasını sağlar.

hidden: taşıma gerçekleşen kısmın taşıdığı kısımda gözükmemesini sağlamak için kullanılır.

Stil Özellikleri...

➤ Taşma Özelliği...

```
<div class="sinirli"> Sınırlı genişlik değerine sahip katmanın İçerisindeki metin</div>
```

```
div.Sinirli {  
width:10%;  
height:100px;  
overflow:visible; }
```

Sınırlı
genişlik
değerine
sahip
katmanın
İçerisindeki
metin

```
div.Sinirli {  
width:10%;  
height:100px;  
overflow:auto; }
```

Sınırlı
geniş.
değer
sahip
katma

```
div.Sinirli {  
width:10%;  
height:100px;  
overflow:scroll; }
```

Sınırlı
geniş.
değer
sahip
katma

```
div.Sinirli {  
width:10%;  
height:100px;  
overflow:hidden; }
```

Sınırlı
genişlik
değerine
sahip
katmanı
İçerisindeki
metin

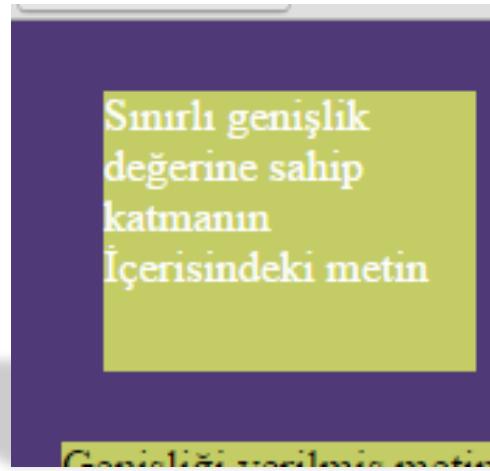
Stil Özellikleri...

➤ "margin" Özelliği

- Kenarlıktan dışarıya doğru etiketin içerisinde bulunduğu yapıdan ne kadar içerde olacağını belirlemek amacıyla kullanılır. Girilen değer üst, alt, sağ ve sol değerler için belirlenecektir.

```
div.sinirli{  
color:white;  
width:10%;  
height:100px;  
overflow:hidden;  
margin:25px;}
```

```
<div class="sinirli"> Sınırlı genişlik değerine  
sahip katmanın içerisindeki metin</div>
```



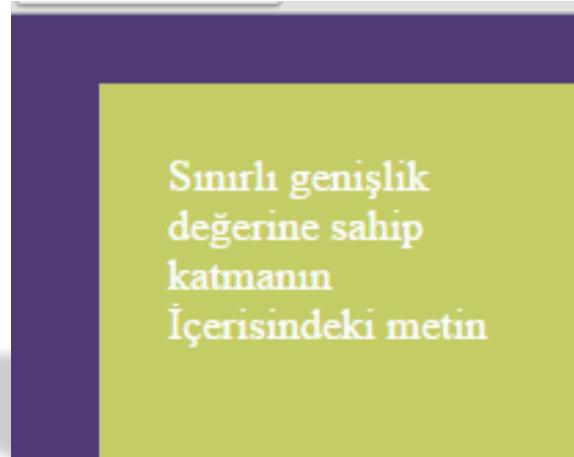
Stil Özellikleri...

➤ "padding" Özelliği

- Kenarlıktan içeriye doğru etiketin içerisinde bulunan yapıların ne kadar içeride olacağını belirlemek amacıyla kullanılır. Girilen değer üst, alt, sağ ve sol değerler için belirlenecektir.

```
div.sinirli{  
color:white;  
width:10%;  
height:100px;  
overflow:hidden;  
margin:25px;  
padding:25px;}
```

```
<div class="sinirli"> Sınırlı genişlik değerine  
sahip katmanın İçerisindeki metin</div>
```



Stil Özellikleri...

- Bir blok elemanın **border**, **margin**, **padding** gibi özellikleri üst, alt, sağ ve sol olarak ayrı ayrı tanımlanabilir.
- Bu özellikler
 - left : sol,
 - right: sağ,
 - top: üst,
 - bottom : altözellik isminden sonra ‘-’ simbolünden sonra yazılır ve sonra değeri belirlenir.

Stil Özellikleri...

➤ **border-width**, **margin**, **padding** özellikleri ayrı ayrı tanımlandığı gibi tek bir satırda toplu halde de tanımlanabilir.

➤ 4lü kullanım (ozellik : [üst] [sağ] [alt] [sol])

border-width: 5px 3px 2px 4px

➤ 2li kullanım (ozellik : [üst|alt] [sağ|sol])

margin:5px 3px (Üst ve Alt 5px, sağ sol 3px)

padding:5px auto (Üst/Alt 5px, sağ ve sol eşit mesafe)

➤ Tekli kullanım (ozellik : [üst|alt|sağ|sol])

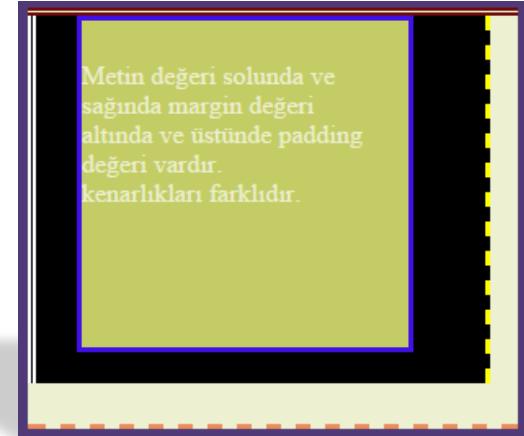
padding: 5px (Her taraf 5px)

Stil Özellikleri...

```
.dis{  
border-top-color:#6FOBOB;  
border-bottom-color:#EB8C5B;  
border-top-style:double;  
border-bottom-style:dashed;  
border-top-width:5px;  
border-bottom-width:3px;  
background-color:#EDF0D1;  
color:black;  
width:300px;  
height:250px;}  
.ic{  
border:#4110E9 3px solid;  
width:200px;  
height:150px;  
margin-left:25px;  
margin-right:25px;  
padding-top:25px;  
padding-bottom:25px; }
```

```
.orta{  
border-left-color:white;  
border-right-color:yellow;  
border-left-style:double;  
border-right-style:dashed;  
border-left-width:5px;  
border-right-width:3px;  
background-color:black;  
color:#EDF0D1;  
width:275px;  
height:225px;}
```

```
<div class="dis">  
 <div class="orta">  
   <div class="ic">  
     Metin değeri solunda ve  
sağında margin değeri<br>altında ve  
üstünde padding değeri  
vardır.<br>kenarlıklar farklıdır.  
   </div>  
 </div>  
</div>
```



Stil Özellikleri...

➤ Bağlantılara Stil Verme

- Bağlantılara stil vermek için '**a**' etiketine stil özellikleri tanımlamak gerekmektedir.

```
a {  
background-color:#4110E9;  
color:#CCEB5B;  
font-family:verdana;  
text-decoration:none;  
font-size:14px;  
}
```

```
<a href="http://www.sakarya.edu.tr">Sakarya  
Üniversitesi</a>  
<br>  
<a href="http://www.cs.sakarya.edu.tr">Bilgisayar  
Mühendisliği Bölümü</a>
```

Sakarya Üniversitesi
Bilgisayar Mühendisliği Bölümü

Stil Özellikleri...

➤ Bağlantılara Stil Verme...

➤ Bağlantılar durumlarına göre farklı şekillerde bulunabilir. 5 farklı durum vardır.

link: bağlantının kendisi,

visited: ziyaret edilmiş bağlantı,

focus: odaklanılmış bağlantı,

hover: üzerine fare ile gelinmiş bağlantı,

active: o an üzerine tıklanmış bağlantı.

➤ Bağlantı etiketinden sonra ‘:’ simbolü kullanılarak yukarıdaki özelliklerin her birine ayrı ayrı stil tanımlanabilir.

Stil Özellikleri...

➤ Bağlantılara Stil Verme...

```
a:hover  
{  
color:yellow;  
text-decoration:underline;  
}  
  
a:visited  
{  
color:red;  
}
```

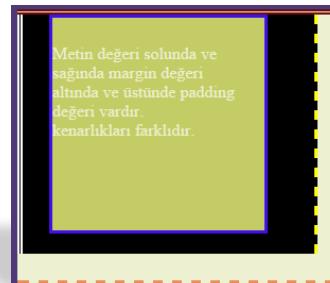
```
<a href="http://www.sakarya.edu.tr">Sakarya  
Üniversitesi</a>  
<br>  
<a href="http://www.cs.sakarya.edu.tr">Bilgisayar  
Mühendisliği Bölümü</a>
```

[Sakarya Üniversitesi](http://www.sakarya.edu.tr)
[Bilgisayar Mühendisliği Bölümü](http://www.cs.sakarya.edu.tr)

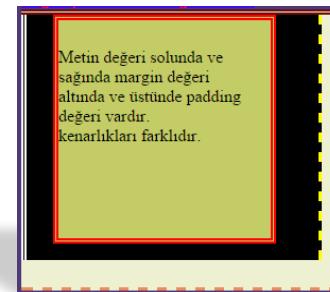
Stil Özellikleri...

➤ ÖRNEK:

```
.ic{  
border:#4110E9 3px solid;  
width:200px;  
height:150px;  
margin-left:25px;  
margin-right:25px;  
padding-top:25px;  
padding-bottom:25px; }
```



```
.ic:hover{  
border:red 5px double;  
color:black;}
```



```
<div class="dis">  
  <div class="orta">  
    <div class="ic">  
      Metin değeri solunda ve sağında margin değeri<br>altında  
      ve üstünde padding değeri vardır.<br>kenarlıkları farklıdır.  
    </div> </div> </div>
```

Stil Özellikleri...

➤ ÖRNEK:

```
.resim:hover  
{  
border:#EB8C5B 3px solid;  
width:210px;  
height:160px;  
}
```

```

```



Stil Özellikleri...

➤ Seçici Çeşitleri

- . ve # ile tanımlanan seçicilerin dışında da seçici tanımlamaları gerçekleştirilebilir.
- Sayfadaki tüm etiketlere aynı stilleri tanımlamak için * kullanılır.
* { border: 1px solid red;}

Stil Özellikleri...

➤ Seçici Çeşitleri...

➤ Özelliğin Seçicisi

- Bir seçici ile tanımlanmış olan bileşen özelliklerinin değerine göre stil belirlemek için kullanılır. Seçicini sahip olduğu özellik '[]' içerisinde belirtilir.

```
<p>Alt özelliksiz</p>
<p align="center">Alt özellik
değerine sahip</p>
```

```
p[align="center"]{background-color:#D2EBE7}
```

Alt özelliksiz

Alt özellik **değerine sahip**



Stil Özellikleri...

➤ Seçici Çeşitleri...

```
div[class="deneme"]{  
background-color:#C3CC65;  
width:50%;  
}
```

```
<div class="deneme">Selam</div>
```



Selam

Stil Özellikleri...

➤ Seçici Çeşitleri...

- Bir etiket içerisinde geçerli olacak **id** veya **class** seçicisi tanımlama
 - Etiket ismi belirtildikten sonra **id** için '#' **class** için '.' simbolü kullanılarak seçici ismi yazılır. Bu tanım sadece belirtilen etikette geçerlidir.

```
p.buyukfont{  
font-size:18pt;  
background-color:#C3CC65;  
}
```

```
<h4 class="buyukfont">sitilsiz başlık</h4>  
<p class="buyukfont">Sitilli paragraf</p>
```

sitilsiz başlık

Sitilli paragraf

Stil Özellikleri...

➤ Seçici Çeşitleri

- Seçici içerisindeki alt seçici sitilleri
 - Bir seçici içerisinde tanımlanan seçenekler üzerinde geçerli olacak sitiller tanımlamak için seçici isminden sonra boşluk bırakılarak alt seçici adı yazılır ve stilleri belirtilir.

```
div p{  
background-color:red;  
color:white;  
}
```

```
<p>Alt özellsiz</p>  
<div><p>Div içerisindeki paragraf</p></div>
```

Alt özellsiz

Div içerisindeki paragraf

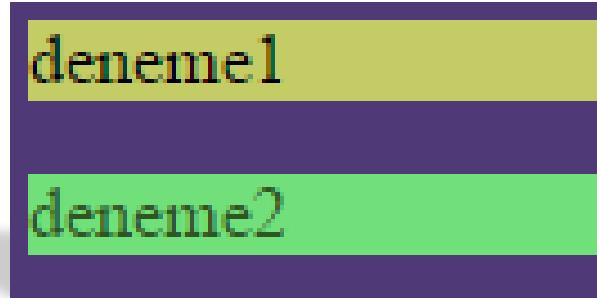
Stil Özellikleri...

➤ Seçici Çeşitleri

- Seçici içerisindeki alt seçici sitilleri

```
.deneme .deneme2{  
background-color:#71DF7A;  
color:#2D5621;  
}
```

```
<div class="deneme2">deneme1</div>  
<div class="deneme"><p  
class="deneme2">deneme2</p></div>
```



Stil Özellikleri...

➤ ÖRNEK:

The screenshot displays the homepage of the 'Film Dünyası' website. At the top, there is a navigation bar with a search bar, social media icons (Facebook, Twitter, LinkedIn), a language switcher (TR), and user links (Üye Giriş, Üye Ol). Below the header is a main title 'Film Dünyası' with a video camera icon. A large, semi-transparent watermark reading 'Vizyondaki Filmler' is centered over the page. The main content area features three colored boxes: a pink 'YAKINDA' box, a blue 'EN ÇOK BEKLЕНENLER' box, and an orange 'SON YORUMLAR' box, each containing five placeholder movie cards. Each card includes a small thumbnail and a 'Tüm Liste' button. At the bottom, there is a footer bar with the text 'Copyright Ü.Kocabıçak' and icons for Android and Apple mobile devices.

Stil Özellikleri...

➤ ÖRNEK:

[FİLM DÜNYASI]

[VİZYONDAKİ FİMLER]

[YAKINDA]...

[EN ÇOK BEKLENENLER]...

[SON YORUMLAR]...

Copyright @Ümit KOCABIÇAK



SAKARYA ÜNİVERSİTESİ

BSM 104 Web Teknolojileri 2020-2021 Bahar

Stil Özellikleri...

➤ ÖRNEK:...

```
.ust{  
    background-color:#ffcccc;  
    color:#660066;  
    height:60px;  
    padding-top:25px;  
    padding-bottom:25px;  
    margin-left:20px;  
    margin-right:20px;  
}  
.ortaust{  
    background-color:#eeeeee;  
    color:#660066;  
    height:180px;  
    padding-top:25px;  
    padding-bottom:25px;  
    margin-left:20px;  
    margin-right:20px;  
    text-align:center;  
}  
.ortaalt{  
    margin-left:20px;  
    margin-right:20px;  
}  
.ortaaltsol{  
    background-color:#b3d9ff;  
    color:#660066;  
    width:31%;  
    height:180px;  
    margin-right:5px;  
    margin-bottom:20px;  
    float:left;  
}
```

Stil Özellikleri...

➤ ÖRNEK:...

```
.ortaaltorta{  
    background-color:#c6aad8;  
    color:#660066;  
    width:32%;  
    height:180px;  
    margin-bottom:20px;  
    float:left;  
}  
  
.ortaaltsag{  
    background-color:#ffcc99;  
    color:#660066;  
    width:32%;  
    height:180px;  
    margin-left:5px;  
    margin-bottom:20px;  
    float:left;  
}  
  
.ortabosluk{  
    width:2%;  
    height:200px;  
    float:left;  
}  
  
.alt{  
    background-color:#ffcccc;  
    color:#660066;  
    padding:3px;  
    height:50px;  
    margin-left:20px;  
    margin-right:20px;  
}  
  
.bosluk{  
    width:100%;  
    height:10px;  
}  
  
.sifirla{clear:both;}
```

Stil Özellikleri...

➤ ÖRNEK:...

```
<div class="ust">[FİLM DÜNYASI]</div>
<div class="bosluk">&nbsp;</div>
<div class="ortaust">[VİZYONDAKİ FİMLER]</div>
<div class="bosluk">&nbsp;</div>
<div class="ortaalt">
    <div class="ortaaltsol">[YAKINDA]...</div>
    <div class="ortabosluk"></div>
    <div class="ortaaltorta">[EN ÇOK BEKLENENLER]...</div>
    <div class="ortabosluk"></div>
    <div class="ortaaltsag">[SON YORUMLAR]...</div>
</div>
<div class="sifirla"></div>
<div class="alt">Copyright @Ümit KOCABIÇAK</div>
```



Web Teknolojileri

Ders 8

Bootstrap 1/2

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

- Bootstrap
- Bootstrap & CSS

Bootstrap Nedir?

- Web sayfası geliştiricileri için twitter çalışanları tarafından 2010 yılından itibaren geliştirilmesi başlanmış bir uygulama çatısıdır.
- Bootstrap kullanılarak ;
 - web sayfalarının farklı cihazlarda düzgün, uygun bir biçimde (responsive) gözükmesi sağlanabilir
 - İçerisinde bir çok web elemanı için hazır stil şablonları barındırır
 - Tasarımın kolayca geliştirilmesi sağlanır
- css ve js dosyalarından oluşur.

Bootstrap Nedir?

➤ Bootstrap ile online çalışılabileceği gibi gerekli dosyalar indirilerek de çalışılabilir.

➤ Bootstrap CSS ile online çalışabilmek için:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
```

➤ Bootstrap JavaScript ile online çalışabilmek için:

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
```

➤ Satırlarının ilgili web sayfasının başlık(head) kısmına eklenmesi gereklidir.

Bootstrap Nedir?

- Bootstrap ile kendi bilgisayarımıza indireceğimiz dosyalar ile çalıştmak için önce gerekli dosyaları sitesinden indirmemiz gerekmektedir.
- Şu an itibarı ile geçerli son versiyonu 4.4.1'dır.
- Bu versiyonun dosyalarını getbootstrap.com sitesinden indirebiliriz.
- Sayfa açıldıktan sonra karşımıza gelen pencereden **download** butonuna tıklayarak açılan pencereden gerekli dosyalar indirilir.

Bootstrap Nedir?

- Dosyaları sayfalarımızın olduğu klasöre açtıktan sonra ilgili web sayfamızın başlık(head) kısmına kullanacağımız dosyaları ekleriz.
- CSS ile çalışmak için:

```
<link rel="stylesheet" href="css/bootstrap.css">
```
- CSS içerisinde boşlukların kaldırılarak dosyanın sıkıştırılmış hali ile çalışmak için:

```
<link rel="stylesheet" href="css/bootstrap.min.css">
```
- JavaScript ile çalışmak için:

```
<script type="text/javascript" src="js/bootstrap.js"></script>
```
- JavaScript içerisinde boşlukların kaldırılarak dosyanın sıkıştırılmış hali ile çalışmak için:

```
<script type="text/javascript" src="js/bootstrap.min.js"></script>
```

Bootstrap Nedir?

- Bootstrap HTML5 sayfalarında çalıştırılacaktır. Bunun için sayfanın en üstüne

```
<!DOCTYPE html>
```

satırı eklenerek doküman HTML5 dokümanı haline getirilir.

- Karakter setini ayarlamak için Başlık kısmına

```
<meta charset="UTF-8">
```

satırı eklenir.

Bootstrap Nedir?

- Sayfanın mobil cihazlarda gösterilebilmesi ve kullanıcıların bu cihazlarda sayfayı büyütüp küçültebilmesi ile ilgili ayarlar için başlık kısmına

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- Daha sonra Bootstrap .css ve .js dosyalarının olduğu satır eklenir.

Bootstrap Nedir?

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>ÖRNEK</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>
  <div class="container-fluid">
    <h1>Sayfamız</h1>
    <p>İlk Örneğimiz...</p>
  </div>
</body>
```

Sayfamız
İlk Örneğimiz...

Bootstrap Grid (Izgara) Yapısı

- Bootstraptta web sayfasındaki elemanların nasıl yerleşeceğini belirlenmesi amacıyla grid(ızgara) sistemini kullanılır.
- Web sayfasını yatay olarak 12 eşit parçaaya yani sütuna ayırarak elemanların yerleştirilmesi sağlanır.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1			
span 4				span 4				span 4							
span 4				span 8											
span 6						span 6									
span 12															

Bootstrap Grid (Izgara) Yapısı...

- Bootstraptta web sayfasındaki elemanların nasıl yerleşeceğini belirlenmesi amacıyla grid sistemine ek olarak farklı cihazlar için farklı alt sınıflar bulunmaktadır.
- 4 farklı ekran çözünürlüğü için farklı alt sınıflar kullanılır.
 - xs: cep telefonları için,
 - sm: tabletler için,
 - md: dizüstü bilgisayarlar için,
 - lg: masaüstü bilgisayarlar için.

Bootstrap Grid (Izgara) Yapısı...



Bootstrap Grid (Izgara) Yapısı...

- 5 farklı izgara sınıfı vardır
 - **col** → ekstra küçük cihazlar (ekran genişliği < 576px)
 - **col-sm** → küçük cihazlar (ekran genişliği \geq 576px)
 - **col-md** → orta cihazlar (ekran genişliği \geq 768px)
 - **col-lg** → geniş cihazlar (ekran genişliği \geq 992px)
 - **col-xl** → xlarge cihazlar (ekran genişliği \geq 1200px)

Bootstrap Grid (Izgara) Yapısı...

- Izgara yapısının kullanılabilmesi için önce izgaranın içinde bulunacağı row sınıfı kullanılarak satır tanımı yapılır.

```
<div class="row"> </div>
```

- Sonra her sütun için div tanımlanır.
- div içerisinde **col-çözünürlük türü-birleştirilecek sütun sayısı** belirtilir.

```
<div class="col-sm-4">Küçük cihazlar için 4 lü sütun</div>
```

Bootstrap Grid (Izgara) Yapısı...

➤ ÖRNEK:

- Tablet için üç eşit sütun

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="UTF-8">
<title>ÖRNEKLER</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>
<div class="container">
<div class="row">
<div class="col-sm-4" style="background-color: yellowgreen">4'lü sütun</div>
<div class="col-sm-4" style="background-color: greenyellow">4'lü sütun</div>
<div class="col-sm-4" style="background-color: yellowgreen">4'lü sütun</div>
</div>
</div>
</body>
```

4'lü Sütun

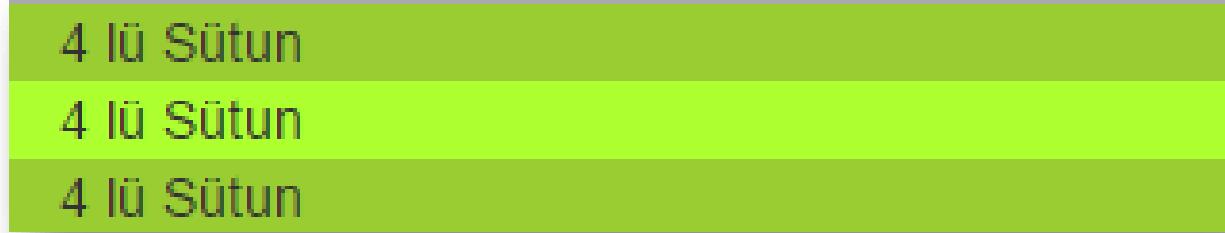
4'lü Sütun

4'lü Sütun



Bootstrap Grid (Izgara) Yapısı...

- Eğer satırda 12 sütundan daha fazla sütun bulunursa 12. sütundan sonraki sütunlar alt satırda gösterilir. Eğer sütunların yerleşimi bir satıra sağlamaz ise gösterime alt satırda devam edilir.
- Örnek olarak bir önceki uygulamada ekran küçültülürse veya daha küçük çözünürlüklü (cep telefonu gibi) ekranda gösterilirse alt alta gösterilir.



Bootstrap Grid (Izgara) Yapısı...

```
<div class="container">
  <div class="row">
    <div class="col-sm-4" style="background-color: yellowgreen">4'lü 3 sütun</div>
    <div class="col-sm-4" style="background-color: greenyellow">4'lü 3 sütun</div>
    <div class="col-sm-4" style="background-color: yellowgreen">4'lü 3 sütun</div>
  </div>
  <div class="row">
    <div class="col-sm-12" style="background-color: lightgreen">12'li tek sütun</div>
  </div>
  <div class="row">
    <div class="col-sm-3" style="background-color: yellowgreen">3'lü 4 sütun</div>
    <div class="col-sm-3" style="background-color: greenyellow">3'lü 4 sütun</div>
    <div class="col-sm-3" style="background-color: yellowgreen">3'lü 4 sütun</div>
    <div class="col-sm-3" style="background-color: greenyellow">3'lü 4 sütun</div>
  </div>
</div>
```

4'lü 3 Sütun

4'lü 3 Sütun

4'lü 3 Sütun

12'li tek Sütun

3'lü 4 Sütun

3'lü 4 Sütun

3'lü 4 Sütun

3'lü 4 Sütun

Bootstrap Grid (Izgara) Yapısı...

- İstenirse birden fazla çözünürlük türü sınıfı araya boşluklar bırakarak değerler verilerek birlikte kullanılabilir
- Ekran çözünürlüğüne bakarak hangi css'i kullanacağını kendisi seçer.

Bootstrap Grid (Izgara) Yapısı...

```
<div class="container">
  <div class="row">
    <div class="col-md-6 col-lg-3" style="background-color:yellowgreen">
      Dizüstü 2, masaüstü 4 sütun
    </div>
    <div class="col-md-6 col-lg-3" style="background-color:greenyellow">
      Dizüstü 2, masaüstü 4 sütun
    </div>
    <div class="col-md-6 col-lg-3" style="background-color:yellowgreen">
      Dizüstü 2, masaüstü 4 sütun
    </div>
    <div class="col-md-6 col-lg-3" style="background-color:greenyellow">
      Dizüstü 2, masaüstü 4 sütun
    </div>
  </div>
</div>
```

Dizüstü 2 sütun Masaüstü 4 sütun

Dizüstü 2 sütun Masaüstü 4 sütun

Dizüstü 2 sütun Masaüstü 4 sütun

Dizüstü 2 sütun Masaüstü 4 sütun

Dizüstü 2 sütun Masaüstü 4 sütun

Dizüstü 2 sütun Masaüstü 4 sütun

Dizüstü 2 sütun Masaüstü 4 sütun

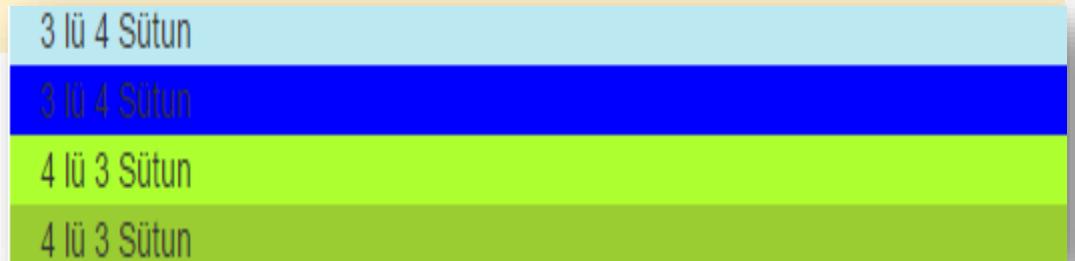
Dizüstü 2 sütun Masaüstü 4 sütun



Bootstrap Grid (Izgara) Yapısı...

- İstenirse birden fazla izgara yapısı iç içe kullanılabilir.
- Kullanım sırasında her defa satır tanımı yapmak gerekmektedir.

```
<div class="container">
  <div class="row">
    <div class="col-sm-4" style="background-color:yellowgreen">
      <div class="row">
        <div class="col-sm-6" style="background-color:#bce8f1">6'lı 2 sütun</div>
        <div class="col-sm-6" style="background-color:blue">6'lı 2 sütun</div>
      </div>
    </div>
    <div class="col-sm-4" style="background-color:greenyellow">4'lü 3 sütun</div>
    <div class="col-sm-4" style="background-color:yellowgreen">4'lü 3 sütun</div>
  </div>
</div>
```



3 lü 4 Sütun

3 lü 4 Sütun

4 lü 3 Sütun

4 lü 3 Sütun



Bootstrap Metin ve Arkaplan Renkleri

➤ Bootstrap ile metinlere verilebilecek belirlenmiş renk sınıfları bulunmaktadır.

- text-muted
- text-primary
- text-success
- text-info
- text-warning
- text-danger
- text-secondary
- text-white
- text-dark
- text-body
- text-light

Bootstrap Metin ve Arkaplan Renkleri...

```
<p class="text-muted">sessiz</p>
<p class="text-primary">önemli</p>
<p class="text-success">başarı</p>
<p class="text-info">bilgi</p>
<p class="text-warning">uyarı</p>
<p class="text-danger">tehlike</p>
<p class="text-secondary">ikinci metin</p>
<p class="text-dark">dark grey</p>
<p class="text-body">gövde rengi(genellikle siyah)</p>
<p class="text-light">açık gri(beyaz arkaplanda)</p>
<p class="text-white">beyaz(siyah arkaplanda)</p>
```

sessiz
önemli
başarı
bilgi
uyarı
tehlike
ikinci metin
dark grey
gövde rengi(genellikle siyah)
açık gri(beyaz arkaplanda)
beyaz(siyah arkaplanda)

Bootstrap Metin ve Arkaplan Renkleri...

➤ Bootstrap ile arkaplan için verilebilecek belirlenmiş renk sınıfları bulunmaktadır.

- bg-primary
- bg-success
- bg-info
- bg-warning
- bg-danger
- bg-secondary
- bg-dark
- bg-light

➤ Arkaplan renkleri ile birlikte metin renklerini de belirtmek için **.text-*** sınıfını kullanabilirsiniz.

Bootstrap Metin ve Arkaplan Renkleri...

```
<p class="bg-primary text-white">ÖNEMLİ</p>
<p class="bg-success text-white">BAŞARILI</p>
<p class="bg-info text-white">BİLGİ</p>
<p class="bg-warning text-white">UYARI</p>
<p class="bg-danger text-white">TEHLİKE</p>
<p class="bg-secondary text-white">İKİNCİ ARKAPLAN RENGİ</p>
<p class="bg-dark text-white">KOYU GRİ ARKAPLAN RENGİ</p>
<p class="bg-light text-dark">AÇIK GRİ ARKAPLAN RENGİ</p>
```

ÖNEMLİ

BAŞARILI

BİLGİ

UYARI

TEHLİKE

İKİNCİ ARKAPLAN RENGİ

KOYU GRİ ARKAPLAN RENGİ

AÇIK GRİ ARKAPLAN RENGİ

Bootstrap Tablolar

- **table** etiketi içerisinde **.table** sınıfı kullanıldığında iç satır kenarlığı olan tablo oluşturulur.

```
<table class="table">
  <tr>
    <td>Ad</td><td>Soyad</td><td>Ortalama</td>
  </tr>
  <tr>
    <td>Ali</td><td>YILMAZ</td><td>75</td>
  </tr>
  <tr>
    <td>Veli</td><td>ÖZTÜRK</td><td>90</td>
  </tr>
</table>
```

Ad	Soyad	Ortalama
Ali	Gel	75
Veli	Git	90

Bootstrap Tablolar...

- Tablo kenarlıklarının olması için ilgili tabloya **.table-bordered**, tabloda zebra şeritler oluşturmak için **.table-striped**, satır üzerinde renk değişikliği yapmak için **.table-hover**, çözünürlüklerle duyarlı tablo için **.table-responsive** sınıfları eklenebilir.
- Tablo satır ve(ya) sütunlarına arkaplan rengi vermek için kullanılabilecek sınıflar:

- table-primary
- table-success
- table-info
- table-warning
- table-danger
- table-active
- table-secondary
- table-dark
- table-light

Bootstrap Tablolar...

➤ ÖRNEK:

```
<table class="table">
  <tr class="table-warning">
    <td>Ad</td><td>Soyad</td><td>Ortalama</td>
  </tr>
  <tr>
    <td class="table-active">Ali</td>
    <td class="table-success">YILMAZ</td>
    <td class="table-info">75</td>
  </tr>
  <tr>
    <td>Veli</td><td>ÖZTÜRK</td><td>90</td>
  </tr>
</table>
```

Ad	Soyad	Ortalama
Ali	YILMAZ	75
Veli	ÖZTÜRK	90

Bootstrap Tablolar...

➤ ÖRNEK:

```
<div class="table-responsive">
  <table class="table table-striped">
    <tr class="table-warning">
      <td>Ad</td><td>Soyad</td><td>Ortalama</td>
    </tr>
    <tr>
      <td class="table-active">Ali</td>
      <td class="table-success">YILMAZ</td>
      <td class="table-info">75</td>
    </tr>
    <tr>
      <td>Veli</td><td>ÖZTÜRK</td><td>90</td>
    </tr>
  </table>
</div>
```

Ad	Soyad	Ortalama
Ali	YILMAZ	75
Veli	ÖZTÜRK	90

Bootstrap Resimler

- Resmin kenarları yuvarlatılmış gösterebilmek için **.rounded**, dairenin içerisinde gösterebilmek için **.rounded-circle**, tırnaklı kenarlıkla gösterebilmek için **.img-thumbnail**, çözünürlük duyarlı resimler oluşturmak için **.img-fluid** sınıfları kullanılır.

```
<div class="container">
  <div class="row">
    <div class="col-lg-4">
      
    </div>
    <div class="col-lg-4">
      
    </div>
    <div class="col-lg-4">
      
    </div>
  </div>
</div>
```

Bootstrap Resimler...

➤ ÖRNEK:

```
<div class="container">
  <div class="row">
    <div class="col-lg-4">
      
    </div>
    <div class="col-lg-4">
      
    </div>
    <div class="col-lg-4">
      
    </div>
  </div>
</div>
```



Bootstrap Alert Sınıfı

- Bootstrap 4 önceden tanımlı uyarı mesajlarına izin verir.
- Basit uyarı mesajları için **.alert** sınıfı kullanılır.
- Arka plan renklerini farklı tanımlamak için aşağıdaki sınıflar kullanılabilir:

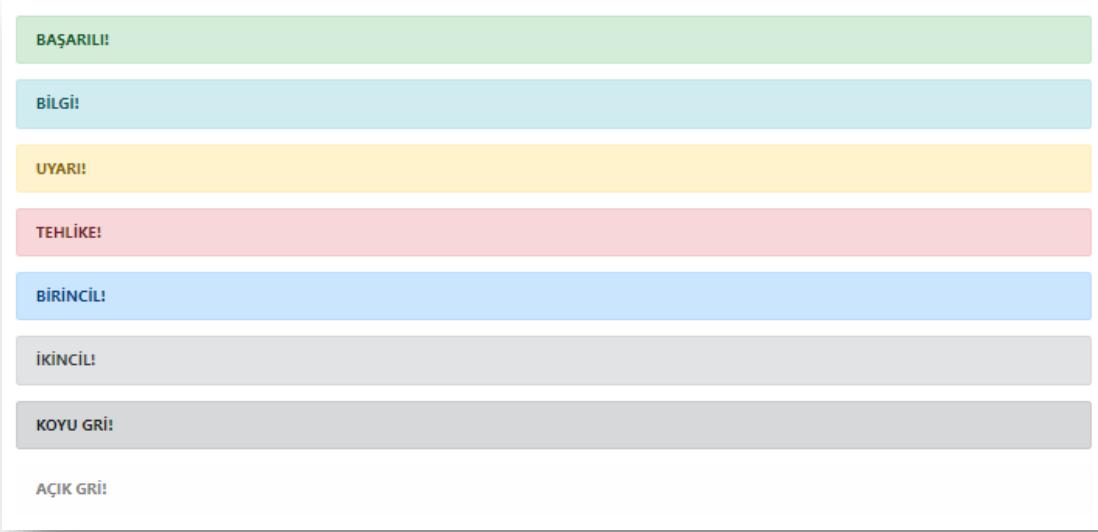
- alert-primary
- alert-success
- alert-info
- alert-warning
- alert-danger
- alert-secondary
- alert-dark
- alert-light

- **NOT:** Bootstrap Javascript kullanılarak kapatılabilir uyarı mesajları da oluşturulabilir. Bu kısım daha sonra anlatılacaktır....

Bootstrap Alert Sınıfı

➤ ÖRNEK:

```
<div class="alert alert-success"><strong>BAŞARILI!</strong></div>
<div class="alert alert-info"><strong>BİLGİ!</strong></div>
<div class="alert alert-warning"><strong>UYARI!</strong></div>
<div class="alert alert-danger"><strong>TEHLİKE!</strong></div>
<div class="alert alert-primary"><strong>BİRİNCİL!</strong></div>
<div class="alert alert-secondary"><strong>İKİNCİL!</strong></div>
<div class="alert alert-dark"><strong>KOYU GRİ!</strong></div>
<div class="alert alert-light"><strong>AÇIK GRİ!</strong></div>
```



Bootstrap Butonlar

- Basit butonlar için .btn sınıfı kullanılır
- Buton sınıfları **<a>**, **<button>** veya **<input>** elemanları için kullanılabilir.

```
<a href="#" class="btn btn-info" role="button">Bağlantı Butonu</a>
<button type="button" class="btn btn-info">Buton</button>
<input type="button" class="btn btn-info" value="Giriş Butonu">
<input type="submit" class="btn btn-info" value="Submit Butonu">
```

Link Button

Button

Input Button

Submit Button

Bootstrap Butonlar...

- Web sayfalarında arka plan renkleri farklı butonlar tanımlamak için aşağıda solda verilen sınıflar, kenarlık renkleri farklı butonlar tanımlamak için ise aşağıda sağda verilen sınıflar kullanılabilir:

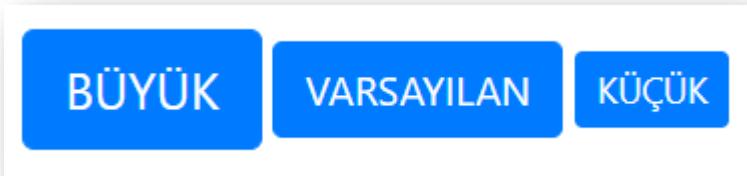
- btn-primary
- btn-secondary
- btn-success
- btn-info
- btn-warning
- btn-danger
- btn-dark
- btn-light
- btn-link

- btn-outline-primary
- btn-outline-secondary
- btn-outline-success
- btn-outline-info
- btn-outline-warning
- btn-outline-danger
- btn-outline-dark
- btn-outline-light

Bootstrap Butonlar...

- Varsayılan boyutundan daha büyük butonlar oluşturmak için **.btn-lg**, daha küçük butonlar oluşturmak için ise **.btn-sm** sınıfları kullanılır.

```
<button type="button" class="btn btn-primary btn-lg">BÜYÜK</button>
<button type="button" class="btn btn-primary btn-md">VAR SAYILAN</button>
<button type="button" class="btn btn-primary btn-sm">KÜÇÜK</button>
```



Bootstrap Butonlar...

- Butonların aktif/pasif olması sağlanabilir. Bir butonun aktif, yani tıklanabilir olması için **.active** sınıfı, tıklanamaz olması için de **disabled** özelliği kullanılır.
- Bağlantı etiketi **disabled** özelliğini desteklemediğinden dolayı bu etiketi tıklanamaz yapmak için **.disabled** sınıfı kullanılır.

```
<button type="button" class="btn btn-primary">BUTON</button>
<button type="button" class="btn btn-primary active">AKTİF BUTON</button>
<button type="button" class="btn btn-primary" disabled>ENGELLENMİŞ BUTON</button>
<a href="#" class="btn btn-primary disabled">ENGELLENMİŞ BAĞLANTI</a>
```

BUTON

AKTİF BUTON

ENGELLENMİŞ BUTON

ENGELLENMİŞ BAĞLANTI

Bootstrap Butonlar

- Web sayfasındaki butonları ayrı ayrı değil de tek bir satırda grüplamak için butonlar **.btn-group** sınıfı içerisinde tanımlanır.
- Bunu gerçekleştirmek için butonlar bir **.btn-group** sınıfı kullanılarak tanımlanmış bir **<div>** etiketi içerisinde tanımlanır.

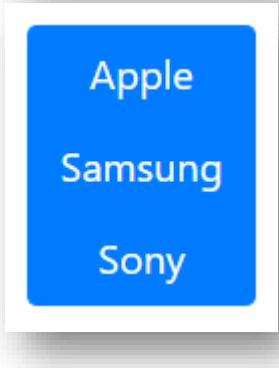
```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>
```

Apple Samsung Sony

Bootstrap Butonlar

- Butonları dikey olarak grüplamak için **.btn-group-vertical** sınıfı kullanılır.

```
<div class="btn-group-vertical">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>
```



Apple

Samsung

Sony



Bootstrap Badge Sınıfı

- İşaretler herhangi bir içerik için ilave bilgi vermek için kullanılır.
- Dikdörgensel işaretler oluşturmak için **** etiketi içerişimde **.badge** sınıfı kullanılır.
- **.badge** sınıfı içerisinde kullanılabilecek içerik sınıfları aşağıda listelenmiştir.

- badge-primary
- badge-secondary
- badge-success
- badge-info
- badge-warning
- badge-danger
- badge-dark
- badge-light
- badge-link

- **.badge-pill** sınıfı kullanılarak daha yuvarlatılmış kenarlıklı işaretler oluşturulur.

Bootstrap Badge Sınıfı

➤ ÖRNEK:

```
<span class="badge">İŞARET</span>
<span class="badge badge-warning">UYARI İŞARETİ</span>
<span class="badge badge-pill badge-danger">YUVARLATILMIŞ UYARI İŞARETİ</span>
```

İŞARET **UYARI İŞARETİ** YUVARLATILMIŞ UYARI İŞARETİ

```
<button type="button" class="btn btn-primary">
  MESAJLAR <span class="badge badge-light">4</span>
</button>
<button type="button" class="btn btn-danger">
  UYARILAR <span class="badge badge-light">7</span>
</button>
```

MESAJLAR 4

UYARILAR 7

Bootstrap Spinner Sınıfı

- Bir döndürücü/yükleyici oluşturmak için, **.spinner-border** sınıfı kullanılır.
- Dönme efekti yerine büyümeye efekti oluşturmak için **.spinner-grow** sınıfı kullanılır.
- **ÖRNEK:**

```
<div class="spinner-border text-primary"></div>
```



```
<button class="btn btn-primary">
  <span class="spinner-border spinner-border-sm"></span>
  YÜKLENİYOR..
</button>
```

Loading..

A larger blue rectangular button with rounded corners. It features a white circular loading icon on the left and the text "Loading.." to its right.

Bootstrap Sayfalama

- Birden fazla sayfadan oluşan yapılarda sayfalama yapabilmek amacıyla **** liste elemanı içerisinde **.pagination** sınıfı kullanılır.
- Her liste elamanı için **.page-item**, liste elemanı içerisindeki her bağlantı elemanına da **.page-link** sınıfı kullanılır.
- Geçerli olan sayfanın sınıfı **.active** yapılır.

```
<ul class="pagination">
  <li class="page-item"><a class="page-link" href="#">&lt;</a></li>
  <li class="page-item active"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">&gt;</a></li>
</ul>
```



Bootstrap Sayfalama...

- Birden fazla sayfadan oluşan yapılarda sayfalama yapabilmek için kutular yerine "/" kullanmak amacıyla **** listesi içerisinde **.breadcrumb** sınıfı kullanılır.
- Her liste elamanı için **.breadcrumb-item** sınıfı kullanılır.
- Geçerli olan sayfanın sınıfı **.active** yapılır.

```
<ul class="breadcrumb">
  <li class="breadcrumb-item"><a href="#">1</a></li>
  <li class="breadcrumb-item"><a href="#">2</a></li>
  <li class="breadcrumb-item"><a href="#">3</a></li>
  <li class="breadcrumb-item active">4</li>
</ul>
```

1 / 2 / 3 / 4

Bootstrap Listeler

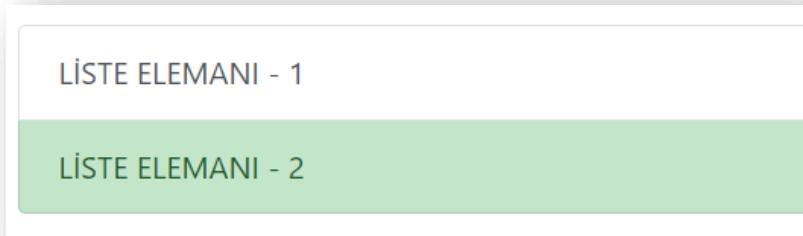
- Liste kutuları ile çalışmak için **** etiketi içerisinde **.list-group** sınıfı, liste elemanları içerisinde de **.list-group-item** sınıfı kullanılır.
- Geçerli olan liste elemanı için **.active** sınıfı kullanılır.
- Liste elemanın mouse ile üzerine gelindiğinde renk değişikliği yapmak için **.list-group-item-action** sınıfı kullanılır.
- Liste elemanlarının arkaplanını renklendirmek için ise aşağıdaki sınıflar kullanılır:

- **list-group-item -primary**
- **list-group-item-success**
- **list-group-item -secondary**
- **list-group-item -info**
- **list-group-item -warning**
- **list-group-item -danger**
- **list-group-item -dark**
- **list-group-item -light**

Bootstrap Listeler...

➤ ÖRNEK:

```
<div class="list-group">
  <a href="#" class="list-group-item
    list-group-item-action">LİSTE ELEMANI - 1</a>
  <a href="#" class="list-group-item
    list-group-item-action
    list-group-item-success">LİSTE ELEMANI - 2</a>
</div>
```



Bootstrap Listeler...

➤ ÖRNEK:...

```
<ul class="list-group">
  <li class="list-group-item
    list-group-item-action
    list-group-item-success">LİSTE ELEMANI - 1</li>
  <li class="list-group-item
    list-group-item-secondary">LİSTE ELEMANI - 2</li>
</ul>
```

LİSTE ELEMANI - 1

LİSTE ELEMANI - 2



Kaynaklar:

- <http://www.w3schools.com/>
- <http://ozgrozer.com/2015/06/08/bootstrap-izgara-sistemi>



Web Teknolojileri

JavaScript

1/2

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

➤ JavaScript

JavaScript

- JavaScript, yaygın olarak web tarayıcılarında kullanılmakta olan dinamik bir script dilidir.
- JavaScript ile yazılan **istemci tarafı betikler** sayesinde tarayıcının kullanıcıyla etkileşimde bulunması, tarayıcının kontrol edilmesi, asenkron bir şekilde sunucu ile iletişime geçilmesi ve web sayfası içeriğinin değiştirilmesi gibi işlevler sağlanır.

JavaScript...

- JavaScript, yaygın olarak web tarayıcılarında kullanılmakta olan dinamik bir **script dili**dir.
- Script Nedir?
 - Herhangi bir programlama dilinde yazılmış, uygulama bölümlerinin tüm kodlarını içinde bulunduran hazır kod blogları
 - Farklı program dillerinde hazırlanan scriptler bu dil ile bağlantılı olarak farklı özellikler taşırlar ve bu dilin özelliklerine göre kullanılırlar.
 - Bu nedenle scriptler isimlendirilirken, kullanılan programlama dili de belirtilir.
- JavaScript ile yazılan **istemci tarafı scriptler(betikler)** sayesinde tarayıcının kullanıcıyla etkileşimde bulunması, tarayıcının kontrol edilmesi, asenkron bir şekilde sunucu ile iletişime geçilmesi ve web sayfası içeriğinin değiştirilmesi gibi işlevler sağlanır.

JavaScript...

- JavaScript kodları **script** etiketleri ile tanımlanır.
- JavaScript 3 şekilde kullanılabilir.
 - <head> etiketleri arasında
 - <body> etiketleri arasında
 - Harici dosyaya yazılabilir.

```
<script>
document.getElementById("demo").innerHTML = "İlk Javascript Örneğim";
</script>
```

JavaScript...

- JavaScript <head> etiketi örneği.

```
<!DOCTYPE html>
<html><head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraf değişti.";
}
</script>
</head>
<body>
<h1>Web Sayfam</h1>
<p id="demo">Paragraf Metni</p>
<button type="button" onclick="myFunction()">DENE</button>
</body>
</html>
```

JavaScript...

- JavaScript <body> etiketi örneği.

```
<!DOCTYPE html>
<html>
<body>
<h1>Web Sayfam</h1>
<p id="demo">Paragraf Metni</p>

<button type="button" onclick="myFunction()">DENE</button>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraf Değişti.";
}
</script>
</body>
</html>
```



JavaScript...

➤ JavaScript'in harici dosya ile kullanımı.

➤ myScript.js

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraf değişti.";  
}
```

➤ <body> etiketi içerisinde çağrıma

```
<!DOCTYPE html>  
<html>  
<body>  
<script src="myScript.js"></script>  
</body>  
</html>
```

JavaScript...

➤ JavaScript harici dosya ile kullanımı.

➤ myScript.js

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraf değişti.";  
}
```

➤ <head> etiketi içerisinde çağırma

```
<!DOCTYPE html>  
<html>  
<head>  
<script src="myScript.js"></script>  
</head>  
<body>  
</body>  
</html>
```



JavaScript...

- JavaScript ile HTML sayfasına farklı şekillerde yazdırma işlemi yapılabilir.
 - Uyarı penceresi → **window.alert();**
 - HTML sayfaya yazdırma → **document.write();**
 - HTML tagi içine yazdırma → **innerHTML**
 - Tarayıcı konsol içine yazdırma → **console.log()**

JavaScript...

➤ **window.alert()**

```
<!DOCTYPE html>
<html>
<body>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

JavaScript...

➤ **document.write()**

```
<!DOCTYPE html>
<html>
<body>

<script>
    document.write(7 + 6);
</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<h1>İlk Web Sayfam</h1>
<p>İlk Paragrafım</p>

<button
    onclick="document.write(5 + 6)">DENE
</button>

</body>
</html>
```



JavaScript...

➤ **innerHTML**

- `document.getElementById (id)` – Bir HTML öğesine erişmek için kullanılır.

```
<!DOCTYPE html>
<html>
<body>

<h1>İlk Web Sayfam</h1>
<p>İlk Paragrafım</p>

<p id="demo"></p>

<script>
  document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```



JavaScript...

➤ **console.Log()**

- Tarayıcıdaki konsola log yazdırma^kmak için kullanılır.
- F12 ile tarayıcı konsolunu etkinleştirin ve sekmele^rden **console** seçiniz.

```
<!DOCTYPE html>
<html>
<body>

<h1>İlk Web Sayfam</h1>
<p>İlk Paragrafım</p>

<script>
  console.log(7 + 4);
</script>

</body>
</html>
```



JavaScript Sözdizimi

- Her bir ifade ; ile ayrılır.
- Değişkenler **var** deyimi ile tanımlanır.
- Atama işlemi = ile yapılır

```
var x = 5;  
var y = 6;  
var z = x + y;
```

- Metinler çift veya tek tırnak içinde yazılır.

"Sakarya Üniversitesi"

'Sakarya Üniversitesi'

JavaScript Sözdizimi...

- Değişkenler HTML içeriğine aktarılabilir.

```
<p id="demo"></p>

<script>
var carName = "Volvo";
document.getElementById("demo").innerHTML = carName;
</script>
```

JavaScript Sözdizimi

- Açıklama satırı için // veya /* */ ifadesi kullanılır.

```
var x = 5;    // Komut çalışır  
// var x = 6;  Komut Çalışmaz  
  
/*Birden fazla satır  
Yorum bu şekilde yapılır*/
```

- Değişken tanımlarken alt çizgi yada camelcase metodu kullanılabilir.

```
var ad_soyad, kullanici_adi;  
  
var AdSoyad, KullaniciAdi;
```

JavaScript Sözdizimi...

- Aritmetik operatörler hesaplama işlemleri için kullanılır.

```
(5 + 6) * 10
```

- Değişkenler ile aritmetiksel işlemler yapılabilir.

```
x* (5 + 6) * 10
```

- Metinler + ile birleştirilebilir.

```
"Sakarya" + " " + "Üniversitesi"
```

JavaScript Operatörler

Operatör	Açıklama
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Mod Alma
++	Artırma
--	Azaltma
=	Atama

// Arttırma

```
var x = 5;  
x++;  
var z = x;
```

// Toplama

```
var x = 5;  
var y = 2;  
var z = x+y;
```

// Atama

```
var x = 10;
```

// Atama

```
var x = 10;  
x += 5;
```

// Azaltma

```
var x = 5;  
x--;  
var z = x;
```

// Çıkarma

```
var x = 5;  
var y = 2;  
var z = x-y;
```

// Atama

```
var x = 10;  
x /= 5;
```

// Atama

```
var x = 10;  
x -= 5;
```

//Çarpma

```
var x = 5;  
var y = 2;  
var z = x*y;
```

// Bölme

```
var x = 5;  
var y = 2;  
var z = x/y;
```

// Atama

```
var x = 10;  
x *= 5;
```

// Atama

```
var x = 10;  
x %= 5;
```

JavaScript Veri Tipleri

➤ Dinamik Veri Tipleri

```
var x;           // x tanımsız  
var x = 5;      // x sayı  
var x = "Web"; // x metin
```

➤ Ondalık ve Tamsayı Tipler

```
var x1 = 34.00;    // x1 Ondalık değişken  
var x2 = 34;       // x2 tamsayı değişken, ondalık değil
```

➤ Boolean veri tipleri

```
var x = true;  
var y = false;
```

JavaScript Diziler

- Diziler [] içinde tanımlanır.

```
var arabalar= ["Mercedes", "BMW", "Audi"];
```

JavaScript Fonksiyonlar

- Bir JavaScript fonksiyonu belirli bir görevi yerine getirmek için tasarlanmış bir kod bloğudur
- Javascript fonksiyonu **function** anahtar kelimesi ile tanımlanır.
- Fonksiyon isimleri harfler, sayılar, altçizgi içerebilir.
- Fonksiyonlar birden fazla parametre alabilir.

```
function name(parametre1, parametre2, parametre3) {  
    //Komutlar  
}
```

JavaScript Fonksiyonlar...

- Javascript fonksiyonları **return** anahtar kelimesi ile geri değer döndürürüler.

```
// myFunction isimli fonksiyon çağrırlır,  
// fonksiyondan geri dönen değer x değişkenine atanır  
var x = myFunction(4, 3);  
-----  
// p1 ve p2'nin çarpımını döndüren fonksiyon  
function myFunction(a, b) {  
    return a * b;  
// Fonksiyon a ve b değerinin çarpımını geri gönderir  
}
```

JavaScript Fonksiyonlar...

➤ ÖRNEK:

➤fahrenhayt değerini derece dereğine çeviren fonksiyon örneği

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}
```

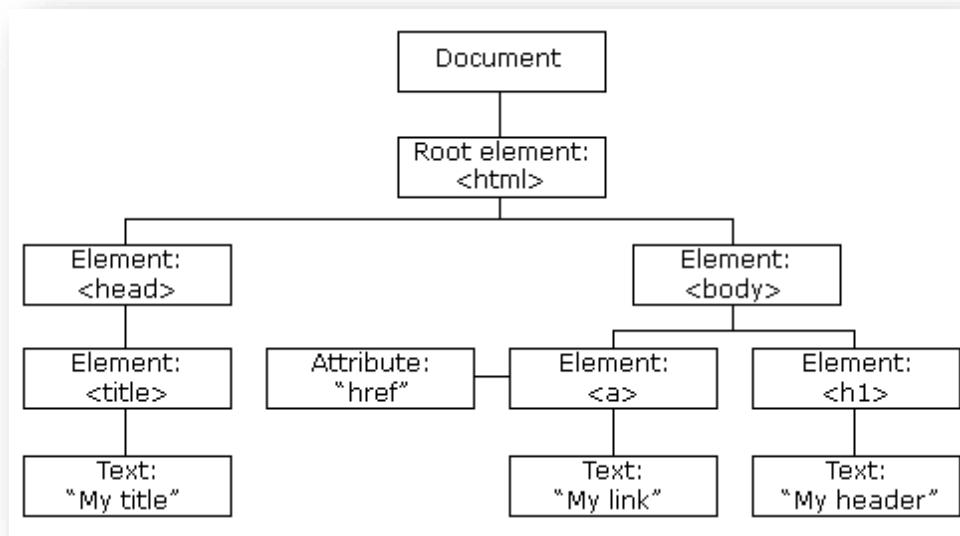
```
// Fonksiyon Kullanımı - Örnek 1  
document.getElementById("demo").innerHTML = toCelsius(77);
```

```
// Fonksiyon Kullanımı - Örnek 2  
var text = "Sıcaklık :" + toCelsius(77) + " Celsius";
```

```
// Fonksiyon Kullanımı - Örnek 3  
var x = toCelsius(32);  
var text = "Sıcaklık :" + x + " Celsius";
```

JavaScript – HTML DOM

- Bir web sayfası yüklendiğinde, tarayıcı sayfanın bir doküman nesne modelini (DOM – Document Object Model) oluşturur.
- Bir HTML Dom modeli bir nesneler ağacı olarak oluşturulur.



- HTML DOM ile, JavaScript bir HTML dökümanındaki tüm elemlara erişebilir ve değiştirebilir.

JavaScript – HTML DOM...

- W3C (World Wide Web Consortium) standarı
- DOM HTML belgelerine erişmek için bir standart tanımlar
- HTML DOM aşağıdakileri yapabilir
 - HTML elemanlarını **nesne** olarak tanımlar
 - HTML elemanlarının **özelliklerini** kullanır.
 - HTML öğelerin **yöntemlerine** erişebilir.
 - HTML elemanlarının **olaylarını** tanımlar.

JavaScript – HTML DOM...

- HTML DOM ile JavaScript aşağıdaki işlevleri yapabilir
 - JavaScript sayfasındaki tüm HTML öğeleri değiştirebilir
 - JavaScript sayfasındaki tüm CSS stilleri değiştirebilir
 - JavaScript ile HTML elemanları ve özelliklerini kaldırılabilir
 - Yeni HTML öğeleri ve özellikleri eklenebilir
 - JavaScript ile yeni bir HTML olayı oluşturulabilir

JavaScript – HTML DOM...

- HTML DOM ile HTML içeriği değiştirebilir.
- **innerHTML** özelliği ile içerik değiştirilebilir.

```
<html>
<body>

<p id="demo"></p>

<script>
  document.getElementById("demo").innerHTML = "Merhaba Arkadaşlar";
</script>

</body>
</html>
```

JavaScript – HTML DOM...

- **getElementById** yöntemi ile HTML elemanlarına **id** özelliği aracılığıyla erişilebilir.

```
<html>
<body>

<p id="demo"></p>

<script>
  document.getElementById("demo").innerHTML="Merhaba Arkadaşlar";
</script>

</body>
</html>
```

JavaScript – HTML DOM...

➤ **getElementById** farklı çeşitleri vardır.

Metot	Açıklama
<code>document.getElementById(<i>id</i>)</code>	id ile elemana erişim
<code>document.getElementsByTagName(<i>name</i>)</code>	Etiket ismi ile elemana erişim
<code>document.getElementsByClassName(<i>name</i>)</code>	Class ismi ile elemana erişim

JavaScript – HTML DOM...

- HTML elemanlarının özelliklerini değiştirebilir.

Metot	Açıklama
<code>element.innerHTML = yeni html içeriği</code>	Elemanın Html içeriği değiştirilir
<code>element.ozellik = yeni değer</code>	Elemanların özelliklerine değer atama
<code>element.setAttribute(ozellik, deger)</code>	Elemanların özelliklerine değer atama
<code>element.style.property = yeni stil</code>	Elemanlara stil tanımlama

JavaScript – HTML DOM...

- Yeni HTML elemanı ekler yada mevcut HTML elemanını silebilir.

Metot	Açıklama
document.createElement(<i>eleman</i>)	Yeni HTML elemanı oluşturma
document.removeChild(<i>eleman</i>)	Mevcut HTML elemanını siler
document.appendChild(<i>eleman</i>)	Varolan elemana, alt HTML elemanı ekler
document.replaceChild(Yeni, <i>degisecek</i>)	HTML elemanını değiştirir
document.write(<i>text</i>)	<i>text</i> 'i ekrana yazdırır

JavaScript – HTML DOM...

- Yeni HTML olayı ekleyebilir.

Metot	Açıklama
document.getElementById(<i>id</i>).onclick = function(){ <i>komutlar</i> }	id'si verilen elemana tıklama özelliğine fonksiyon eklenir

JavaScript – HTML DOM...

- HTML etiketlerine erişerek içeriği, özellikleri, vs. değiştirilebilir.

```
<!DOCTYPE html>
<html>
<body>

<h1 id="header">Eski Başlık</h1>

<script>
  var element = document.getElementById("header");
  element.innerHTML = "Yeni Başlık";
</script>

</body>
</html>
```

JavaScript – HTML DOM...

- HTML etiketlerinin özniteliklerine erişerek değiştirilebilir.

```
<!DOCTYPE html>
<html>
<body>



<script>
  document.getElementById("myImage").src = "landscape.jpg";
</script>

</body>
</html>
```



JavaScript Olayları

➤ HTML DOM Olayları

- Her bir HTML elemanına olay eklenebilir.
- Olaylara tepki verilebilir.

➤ HTML olaylarına bazı örnekler

- Kullanıcı fareyi tıkladığında
- Bir web sayfası yüklendiğinde
- Bir görüntü yükledikten sonra
- Bir ögenin üstündeyken fare hareket ettiğinde
- Bir giriş alanı değiştirildiğinde
- HTML form gönderildiğinde
- Kullanıcı bir tuşa bastığı zaman

JavaScript Olayları

➤ OnClick

- Bir öğeye tıklayınca çalışır

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="this.innerHTML='Ooops ! '">Buraya TIKLA</h1>

</body>
</html>
```



JavaScript Olayları...

➤ **OnLoad**

- Kullanıcı web sayfasına girdiği anda sayfa yüklenince çalışır.

```
<body onload="checkCookies()">
```

➤ **Onunload**

- Kullanıcı web sayfasından ayrılınca çalışır.

```
<body onunload="close()" >
```

JavaScript Olayları...

➤ **OnChange**

- Html ögesinin içeriği değiştirildiğinde çalışır.

```
<input type="text" id="fname" onchange="toUpperCase()">
```

➤ **OnMouseOver**

- Html ögesine fare ile üzerine gelindiğinde çalışır.

➤ **OnMouseOut**

- Html ögesinden fare ile üzerinden çıktıduğunda çalışır

JavaScript – ÖRNEKLER...

➤ OnMouseOver & OnMouseOut

```
<!DOCTYPE html>
<html>
<body>
<div onmouseover="mOver(this)" onmouseout="mOut(this)"
      style="backgroundcolor:#D94A38;width:120px;height:20px;padding:40px;">
    Mouse Üzerime Getir</div>
<script>
  function mOver(obj) {
    obj.innerHTML = "TEŞEKKÜRLER..."
  }
  function mOut(obj) {
    obj.innerHTML = "Mouse Üzerime Getir..."
  }
</script>
</body>
</html>
```

JavaScript – ÖRNEKLER...

➤ Form Elemanları

- frm1 isimli formdaki tüm elemanların içindeki değerleri demo Id değere sahip p etiketinin içerisine yazar.

```
<p id="demo">İçerikler buraya gelecek.</p>
```

```
var x = document.forms["frm1"];
var text = "";
var i;
for (i = 0; i < x.length; i++) {
    text += x.elements[i].value + "<br>";
}
document.getElementById("demo").innerHTML = text;
```

JavaScript – ÖRNEKLER...

➤ Form Doğrulama

```
function validateForm() {  
    var x = document.forms["myForm"]["fname"].value;  
    if (x == null || x == "") {  
        alert("Adınızı Giriniz");  
        return false;  
    }  
}
```

```
<form name="myForm" action="demo_form.asp"  
      onsubmit="return validateForm()" method="post">  
    Adı Soyadı: <input type="text" name="fname">  
    <input type="submit" value="Submit">  
</form>
```

Kaynaklar

➤ <http://www.w3schools.com/>



Web Teknolojileri

JavaScript

2/2

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

➤ JavaScript

JavaScript Nesneleri

- Objeler(Nesneler) gerçek hayatı varlıklarını modelleyen değişkenlerdir.
- Örnek olarak bir arabayı obje olarak modellersek;
 - Bir arabanın ağırlık, renk gibi özellikleri varken çalıştır ve durdur şeklinde metotları bulunmaktadır.

JavaScript Nesneleri...

- obje ⇒ araba



Özellik	Metod
araba.marka=Fiat	araba.calistir()
araba.model=500	araba.sur()
araba.agirlik=850kg	araba.rolanti()
araba.renk=beyaz	araba.durdur()

- Tanımlanan bir nesne ile birden fazla araba oluşturabilir.
- Oluşturulan her araba farklı özelliklere sahip olabilir.
- Örneğin bir arabanın rengi siyahken diğer beyaz olabilir.

JavaScript Nesneleri...

- Aşağıdaki kodda **araba** değişkenine basit bir Fiat string değeri aktarıldı.

```
var araba = "Fiat";
```

- İkinci kodda araba değişkenine birden fazla değer aktarılmaktadır.

```
var araba = {marka:"Fiat", model:"500", renk:"beyaz"};
```

- JavaScript nesnelerine **isim:değer** şeklinde değer aktarılır. Virgüllerle birden fazla değer birbirinden ayrılır.

```
var kisi = {ad:"Ayşe", soyad:"Yılmaz", yas:50, gozRengi:"mavi"};
```

JavaScript Nesneleri...

- Nesne özelliklerine iki farklı yolla erişilebilir:

nesneAdı.özellikAdı veya ***nesneAdı["özellikAdı"]***

```
var kisi = {ad:"Ayşe", soyad:"Yılmaz", yas:50, gozRengi:"mavi"};  
...  
// Nesne özelliğine erişme  
kisi.soyad;           //1. yol  
kisi["soyad"];        //2. yol
```

JavaScript Nesneleri...

- Nesne metodlarına erişim ***nesneAdı.metotAdı();***

```
var kişi = {  
    ad: "Ahmet",  
    soyad : "Yılmaz",  
    id      : 5566,  
    adSoyad : function() {  
        return this.ad + " " + this.soyad;  
    }  
};  
...  
var isim=kisi.adSoyad();           //DOĞRU ERIŞİM  
var isim=kisi.adSoyad;            //YANLIŞ ERIŞİM
```

JavaScript Nesneleri...

➤ **ÖRNEK:** Nesne özellik erişimi

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Sonucu Burada Göster...</p>
<script>
  var kisi = {ad: "Ahmet", soyad : "Yılmaz"};
  document.getElementById("demo").innerHTML = kisi.ad;
</script>
</body>
</html>
```

JavaScript Nesneleri...

➤ ÖRNEK: Nesne metot erişimi

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Sonucu Burada Göster...</p>
<script>
var kisi = {
    ad: "Ahmet",
    soyad : "Yılmaz",
    id      : 5566,
    adSoyad : function() {
        return this.ad + " " + this.soyad; } };
document.getElementById("demo").innerHTML = kisi.adSoyad();
</script>
</body>
</html>
```

Değişkenlerin Yaşam Alanları

- Değişkenler yaşam alanlarına göre iki türdedir.
 - Global değişkenler
 - Lokal değişkenler
- Global değişkenlere javascript kodunda her yerden ulaşılabilir ve yaşam alanları tüm javascript kodu kadardır.
- Lokal değişkenler ise sadece tanımlandığı bloktan erişilebilir ve yaşam alanları blok içindedir. Bloğun dışına çıkışınca ölürlər.

Değişkenlerin Yaşam Alanları...

➤ Global Değişkenler

- Global değişkenlere javascript kodunda her yerden ulaşılabilir ve yaşam alanları tüm javascript kodu kadardır.

```
var araba = "Volvo";  
...  
// araba değişkenine buradan erişilebilir.  
  
function myFunction() {  
    ...  
    // araba değişkenine buradan da erişilebilir  
}
```

Değişkenlerin Yaşam Alanları...

➤ Global Değişkenler...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>
<p>Bir GLOBAL değişken herhangi bir script veya fonksiyonda
    kullanılabilir</p>
<p id="demo">Sonucu Burada Göster...</p>
<script>
    var araba = "Volvo";
    myFunction();

    function myFunction() {
        document.getElementById("demo").innerHTML = "Bu araba" + araba;
    }
</script>
</body>
</html>
```

Değişkenlerin Yaşam Alanları...

➤ Lokal Değişkenler

- Lokal değişkenler ise sadece tanımlandığı bloktan erişilebilir ve yaşam alanları blok içindedir. Bloğun dışına çıkışınca ölürlər.

```
// araba değişkenine buradan ulaşılamaz.  
// Burada yaşamamaktadır.  
function myFunction() {  
    var araba = "Volvo";  
  
    ...  
    // araba değişkenine buradan erişilebilir.  
    // Değişken bu blok için yaşar.  
}
```

Değişkenlerin Yaşam Alanları...

➤ Lokal Değişkenler...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>
<p>Bir LOKAL değişken sadece tanımlandığı fonksiyon içerisinde erişilebilmektedir</p>
<p id="demo">Sonucu Burada Göster...</p>
<script>
myFunction();
document.getElementById("demo").innerHTML="Bu araba" + typeof araba;

function myFunction() {
    var araba = "Volvo";
}
</script>
</body>
</html>
```

JavaScript String

- Stringler metinsel bilgileri saklayan değişken tipleridir.
- İki şekilde tanımlanabilir; tek tırnak veya çift tırnak içinde.

```
var araba = "Volvo XC60";
var araba = 'Volvo XC60';
```

- String uzunluğu length özelliği ile bulunur.

```
var metin = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = metin.length;
```

- String özel karakterleri escape karetleri ile yazdırabilirsiniz.

```
var x = 'It\'s alright';
var y = "\"Sakarya Üniversitesi\"ne Hoşgeldiniz."
```

JavaScript String...

➤ Özel Karakterler

KOD	ÇIKTI
\'	Tek tırnak
\\"	Çift tırnak
\\"\\	backslash
\n	Yeni satır
\r	Satır başı
\t	tab
\b	backspace
\f	Sayfa başı

JavaScript String...

➤ String Metotları

METOT	AÇIKLAMA
charAt()	İndeksi verilen karakteri döndürür(pozisyon)
indexOf()	String'te verilen değerin ilk bulunduğu indis döndürür
concat()	İki veya daha fazla string'i birleştirir ve birleşmiş string'i döndürür
lastIndexOf()	Stringte verilen değerin son bulunduğu indis döndürür
replace()	Stringte yer bir metni bulup değiştirmek için kullanılır
search()	String içinde bir metnin pozisyonunu döndürür
slice()	Metinde Başlangıç ve bitiş değeri verilen aralıktaki metni döndürür
split()	verilen karaktere göre metni bölerek diziye dönüştürür
substr()	Metinde Başlangıç ve uzunluğu verilen aralıktaki metni döndürür
substring()	Metinde Başlangıç ve bitiş değeri verilen aralıktaki metni döndürür

JavaScript String...

➤ String Metotları...

METOT	AÇIKLAMA
toLocaleLowerCase()	Sunucunun bölge ayarlarına referans alarak metni küçük harfe çevirir
toLocaleUpperCase()	Sunucunun bölge ayarlarına referans alarak metni büyük harfe çevirir
toLowerCase()	Metni küçük harfe çevirir
toString()	Nesneyi string ifadeye dönüştürür
toUpperCase()	Metni büyük harfe çevirir
trim()	Metnin başındaki ve sonundaki boşlukları siler
valueOf()	Nesnenin string değerini döndürür

JavaScript String...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>
<p id="p1">Bu köşe kış köşesi, bu köşe yaz köşesi...</p>
<button onclick="myFunction()">DENE</button>
<p id="demo"></p>
<script>
    function myFunction() {
        var str = document.getElementById("p1").innerHTML;
        var pos = str.indexOf("köşe");
        document.getElementById("demo").innerHTML = pos;
    }
</script>
</body>
</html>
```

Bu köşe kış köşesi, bu köşe yaz köşesi...

DENE



Bu köşe kış köşesi, bu köşe yaz köşesi...

DENE

3

JavaScript String...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>

<p>substr() metodu, bir karakter katarının bir parçasını çıkartır ve
çıkartılan parçayı yeni bir karakter katarında döndürür:</p>
<p id="demo"></p>

<script>
var str = "Elma, Muz, Kivi";
document.getElementById("demo").innerHTML = str.substring(6,10);
</script>

</body>
</html>
```

substr() metodu, bir karakter katarının bir parçasını çıkartır ve çıkartılan parçayı yeni bir karakter katarında döndürür:

Muz,



JavaScript Number

- **parseInt** tamsayı tipine dönüştürür.

```
parseInt("10");           // 10 döndürür
parseInt("10.33");        // 10 döndürür
parseInt("10 20 30");     // 10 döndürür
parseInt("10 yıl");       // 10 döndürür
parseInt("yıl 10");       // NaN döndürür
```

- **parseFloat()** ondalık sayı tipine dönüştürür.

```
parseFloat("10");          // 10 döndürür
parseFloat("10.33");        // 10.33 döndürür
parseFloat("10 20 30");     // 10 döndürür
parseFloat("10 yıl");       // 10 döndürür
parseFloat("yıl 10");       // NaN döndürür
```

JavaScript Number

- **valueOf()** sayısal değerini geri gönderir.

```
var x = 123;  
x.valueOf();           // x değişkeninden 123 döndürür  
(123).valueOf();    // tam sayı 123 den 123 döndürür  
(100 + 23).valueOf(); // 100+23 ifadesinden 123 döndürür
```

JavaScript Tarih&Saat

```
new Date()
new Date(milliseconds)    //86400000
new Date(dateString)      //"October 13, 2014 11:13:00"
new Date(year, month, day, hours, minutes, seconds, milliseconds)
                        //99, 5, 24, 11, 33, 30, 0
```

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var d = new Date();
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```

Wed Apr 29 2020 16:01:58 GMT+0300 (GMT+03:00)

JavaScript Diziler

➤ **Söz Dizimi (Syntax)**

```
var dizi-adi = [item1, item2, ...];
```

➤ **ÖRNEK:**

```
<!DOCTYPE html>
<html>
<body>

<p id="demo1"></p>
<p id="demo2"></p>

<script>
  var arabalar = ["Saab","Volvo","BMW"];
  document.getElementById("demo1").innerHTML = arabalar;
  document.getElementById("demo2").innerHTML = arabalar[0];
</script>

</body>
</html>
```

Saab,Volvo,BMW

Saab



JavaScript Diziler...

- Farklı veri tipleri tek bir dizide barındırılabilir.

```
var kisi = ["Ayşe", "Yılmaz", 46];
```

- **length()** özelliği ile dizinin uzunluğu bulunabilir.

```
var meyveler = ["Muz", "Portakal", "Elma", "Mango"];
meyveler.length; // meyveler dizisinin uzunluğu:4
```

- Dizi elemanlarında döngü yardımıyla dolaşılabilir.

```
var metin="";
var index;
var meyveler = ["Muz", "Portakal", "Elma", "Mango"];
for (index = 0; index < meyveler.length; index++) {
    metin += meyveler[index];
}
```

JavaScript Diziler...

- Dizilere iki yolla eleman eklenebilir.
- Son eleman olarak ekleme

```
var meyveler = ["Muz", "Portakal", "Elma", "Mango"];
meyveler[meyveler.length] = "Limon"; //meyveler dizisine
//(Limon) ekleme
```

- **push** metoduyla ekleme

```
var meyveler = ["Muz", "Portakal", "Elma", "Mango"];
meyveler.push("Limon"); //meyveler dizisine (Limon) ekleme
```

Javascript Boolean

- İki değere sahip olan programlamada sıkılıkla kullanılan değişken tipidir.

YES / NO
ON / OFF
TRUE / FALSE

- **Boolean()** fonksiyonu karşılaştırmanın sonucunu verir.

```
Boolean(10 > 9)           // true döndürür
```

- 0 değeri **false** olarak değerlendirilir.

```
var x = 0;  
Boolean(x);           // false döndürür
```

Karşılaştırma Operatörleri

OPERATÖR	AÇIKLAMA	KARŞILAŞTIRMA	DÖNEN DEĞER
==	eşit	x == 8	false
		x == 5	true
		x == "5"	true
====	değer ve tipi eşit	x === 5	true
		x === "5"	false
!=	eşit değil	x != 8	true
!==	değeri veya tipi eşit değil	x !== 5	false
		x !== "5"	true
		x !== 8	true
>	büyük	x > 8	false
<	küçük	x < 8	true
>=	Büyük veya eşit	x >= 8	false
<=	Küçük veya eşit	x <= 8	true

var x = 5;

Mantıksal Operatörler

OPERATÖR	AÇIKLAMA	ÖRNEK
&&	AND	$(x < 10 \&\& y > 1)$ is true
	OR	$(x == 5 \mid\mid y == 5)$ is false
!	NOT	$!(x == y)$ is true

```
var x = 5;  
var y = 5;
```

Karşılaştırma

➤ **Söz Dizimi (Syntax)**

```
if (koşul) {  
    koşul doğruysa yapılacaklar  
}
```

➤ **ÖRNEK:**

```
if (saat < 18) {  
    mesaj = "İyi Günler...";  
}
```



Karşılaştırma...

➤ **Söz Dizimi (Syntax)**

```
if (koşul) {  
    koşul doğruysa yapılacaklar  
} else {  
    koşul yanlışsa yapılacaklar  
}
```

➤ **ÖRNEK:**

```
if (saat < 18) {  
    mesaj = "İyi Günler...";  
} else {  
    mesaj = "İyi Akşamlar...";  
}
```



Karşılaştırma...

➤ **Söz Dizimi (Syntax)**

```
if (koşul1) {  
    koşul1 doğruysa yapılacaklar  
} else if (koşul2) {  
    koşul1 yanlış ve koşul2 doğruysa yapılacaklar  
} else {  
    koşul1 ve koşul2 yanlışsa yapılacaklar  
}
```

➤ **ÖRNEK:**

```
if (saat < 10)          {mesaj = "Günaydın...";}  
else if (saat < 20)     {mesaj = "İyi Günler...";}  
else                      {mesaj = "İyi Akşamlar...";}
```



Karşılaştırma...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Sonucu Burada Göster.</p>
<script>
    var mesaj;
    var saat = new Date().getHours();
    if (saat < 18) {
        mesaj = "İyi Günler...";
    }
    else {
        mesaj = "İyi Akşamlar...";
    }
    document.getElementById("demo").innerHTML = mesaj;
</script>

</body>
</html>
```

İyi Akşamlar...

Switch

➤ Söz Dizimi (Syntax)

```
switch(ifade) {  
    case n:  
        kod bloğu  
        break;  
    case n:  
        kod bloğu  
        break;  
    default:  
        varsayılan kod bloğu  
}
```



Switch...

➤ ÖRNEK:

```
switch (new Date().getDay()) {  
    case 0: gun = "Pazar"; break;  
    case 1: gun = "Pazartesi"; break;  
    case 2: gun = "Salı"; break;  
    case 3: gun = "Çarşamba"; break;  
    case 4: gun = "Perşembe"; break;  
    case 5: gun = "Cuma"; break;  
    case 6: gun = "Cumartesi"; break;  
    default: alert("!!!!"); break;  
}
```

Döngüler – FOR

➤ **Söz Dizimi (Syntax)**

```
for (ifade 1; ifade 2; ifade 3) {  
    gerçekleştirilecek kod bloğu  
}
```

➤ **ÖRNEK:**

```
for (i = 0; i < 5; i++) {  
    text += "Sayı " + i + "<br>";  
}
```

```
for (i = 0, len = arabalar.length, metin = ""; i < len; i++)  
{  
    metin += arabalar[i] + "<br>";  
}
```

Döngüler – FOR...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>
<script>
    var metin = "";
    var i;
    for (i = 1; i < 10; i = i + 2) {
        metin += i + "<br>";
    }
    document.getElementById("demo").innerHTML = metin;
</script>

</body>
</html>
```

1
3
5
7
9

Döngüler – WHILE

➤ **Söz Dizimi (Syntax)**

```
while (koşul) {  
    gerçekleştirilecek kod bloğu  
}
```

➤ **ÖRNEK:**

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```



Döngüler – DO...WHILE

➤ **Söz Dizimi (Syntax)**

```
do {  
    gerçekleştirilecek kod bloğu  
}  
while (koşul);
```

➤ **ÖRNEK:**

```
do {  
    text += "Sayı " + i;  
    i++;  
}  
while (i < 10);
```

Break & Continue

➤ **break** komutu döngüyü kullanıldığı yerde kırarak sonlandırır.

```
for (i = 0; i < 10; i++) {  
    if (i == 3) { break; }  
    text += "Sayı: " + i + "<br>";  
}
```

➤ **continue** komutu döngüyü kullanıldığı yerde işlem yaptırmadan bir sonraki iterasyona yönlendirir.

```
for (i = 0; i < 10; i++) {  
    if (i == 3) { continue; }  
    text += "Sayı: " + i + "<br>";  
}
```

JavaScript JSON

- JSON saklamak ve taşımak için kullanılan bir veri biçimidir.
- Genellikle veriler bir sunucudan web sayfasına gönderildiğinde kullanılır.
- JSON, programlama dilinden bağımsız olan XML'e alternatif olarak kullanılan javascript tabanlı veri değişim formatıdır.
- JSON'un amacı veri alış verisi yaparken daha küçük boyutlarda veri alıp göndermektir.
- Bu özellikleri sayesinde JSON ile çok hızlı web uygulamaları oluşturabilir.

JavaScript JSON...

➤ **JSON sözdizimi kuralları**

- Veri ad / değer çiftleri içinde yazılır
- Veri virgül ile ayrılır
- Köşeli parantezler diziler tutar

```
{  
  "calisan": [  
    {"ad": "Ayşe", "soyad": "Yılmaz"},  
    {"ad": "Mehmet", "soyad": "Öztürk"},  
    {"ad": "Ömer", "soyad": "Çetin"}]  
}
```

JavaScript JSON...

➤ JSON verisi

- Javascript nesne özellikleri gibi ad / değer çiftleri içinde yazılır.

```
"ad": "Ayşe"
```

➤ JSON nesneleri

- JSON nesneleri kümeye parantezi içine yazılır.
- Sadece JavaScript gibi nesneler birden fazla ad / değer çiftlerini içerebilir:

```
{"ad": "Ayşe", "soyad": "Yılmaz"}
```

JavaScript JSON...

➤ **JSON dizisi**

- JSON nesne dizileri köşeli parantez içinde yazılır.
- JavaScript gibi, bir JSON dizisi nesneler içerebilir:

```
{  
  "calisan": [  
    {"ad": "Ayşe", "soyad": "Yılmaz"},  
    {"ad": "Mehmet", "soyad": "Öztürk"},  
    {"ad": "Ömer", "soyad": "Çetin"}]  
}
```

JavaScript JSON...

➤ Bir JSON metnini JavaScript nesnesine dönüştürme

- JSON'un genel kullanım amaçlarından birisi bir web sunucusundan veri okumak ve bu veriyi web sayfasında görüntülemektir.
- Kolaylık içim, giriş olarak string kullanarak örneklenenecektir.
- İlk önce, JSON sözdizimi içeren bir Javascript string oluşturulur.

```
var metin = '{"calisan":[' +  
    '{"ad":"Ayşe", "soyad":"Yılmaz"},' +  
    '{"ad":"Mehmet", "soyad":"Öztürk"},' +  
    '{"ad":"Ömer", "soyad":"Çetin"}]  
}';
```

- Daha sonra, string'i bir JavaScript nesnesine dönüştürmek için dahili fonksiyonu olan **JSON.parse()** kullanılır:

```
var nesne = JSON.parse(metin);
```

JavaScript JSON...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>
<script>
  var metin = '{"calisan":[' +
    '{"ad":"Ayşe","soyad":"Yılmaz"},' +
    '{"ad":"Mehmet","soyad":"Öztürk"},' +
    '{"ad":"Ömer","soyad":"Çetin"}]}';
  obj = JSON.parse(metin);
  document.getElementById("demo").innerHTML =
    obj.calisan[1].ad + " " + obj.calisan[1].soyad;
</script>
</body>
</html>
```

Mehmet Öztürk

JavaScript JSON...

➤ ÖRNEK:

```
<!DOCTYPE html>
<html>
<body>

<table id="calisanlar">
    <tr><td>ADI</td><td>SOYADI</td><td>BÖLÜMÜ</td></tr>
</table>

<p id="demo"></p>
<script>
    var metin = '{"calisan":[' +
        '{"ad":"Ayşe","soyad":"Çetin", "bolumu":"İmalat"},' +
        '{"ad":"Mehmet","soyad":"Öztürk", "bolumu":"Bilgisayar"},' +
        '{"ad":"Ömer","soyad":"Yılmaz", "bolumu":"İmalat"}]';
    personel = JSON.parse(metin);
    var tablo = document.getElementById("calisanlar");
    for (i in personel.calisan) {
        var row = tablo.insertRow(-1);
        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        var cell3 = row.insertCell(2);
        cell1.innerHTML = personel.calisan[i].ad;
        cell2.innerHTML = personel.calisan[i].soyad;
        cell3.innerHTML = personel.calisan[i].bolumu;}
</script>
</body>
</html>
```

ADI	SOYADI	BÖLÜMÜ
Ayşe	Çetin	İmalat
Mehmet	Öztürk	Bilgisayar
Ömer	Yılmaz	İmalat

Kaynaklar

➤ <http://www.w3schools.com/>



Web Teknolojileri

Ders 11

Api Kullanımı

Dr. Öğretim Üyesi Gülüzar ÇİT
Dr. Öğretim Üyesi Hüseyin ESKI
Öğr. Gör. Dr. Can YÜZKOLLAR

Konu & İçerik

- Api Nedir
- Api Kullanımı
- Örnek API Kullanımı-openweathermap

Api Nedir?

- API: Application Programming Interface'in ilk harflerinden oluşmaktadır. **Uygulama Programlama Arayüzü** anlamına gelmektedir
- Uygulamanın işlevlerine dışarıdan/uzaktan erişilip bu işlevlerin kullanılmasını sağlayan arayüzdür
- Bir sunucuda çalışan uygulamaya farklı platformlardan ulaşılmasını ve cevap dönmesine olanak sağlar
- Web API'lerinin tamamı REST(REpresentational State Transfer) mimarisi üzerinde dizayn edilir
- Web API çıktıları talebe göre JSON, XML gibi çeşitli çıktıları olabilir

Api Nedir?

- Bir uygulamanın bütün veya bazı metodlarını diğer uygulamalara kullanıma açmak için kullanılır
- Kullanıcı uzaktan veri ve bilgi talebinde bulunur
- Verilen izinler mertebesinde erişim sağlanır
- İstenilen parametrelere göre sonuçlar kullanıcıya iletilir
- Kullanıcıya key(anahtar sözcük) ve şifre verilerek kullanıcı sınırlaması yapılmaktadır

Api Çalışma Yapısı

- Kullanıcı API'nin çalıştığı sunucuya talepte bulunur
 - Sunucu talebi yazılmış olan API uygulamasına iletir
 - API talep sonucunu, sunucuya iletir
 - Sunucu API'den gelen cevabı kullanıcıya iletir
-
- Yazılan API uygulamasına göre, cevap olarak bazen veri kümesi olabilir bazen de başarılı/başarısız bilgisi döner

Tarayıcı API'leri - Geolocation

- Tüm tarayıcılarda, karmaşık işlemleri desteklemek ve verilere erişmeye yardımcı olmak için bir dizi yerleşik Web API'si bulunmaktadır
- Örneğin, Geolocation API tarayıcının bulunduğu yerin koordinatlarını döndürebilir.

```
function getLocation() {  
    if (navigator.geolocation) {  
        navigator.geolocation.getCurrentPosition(showPosition);  
    } else {  
        x.innerHTML = "Geolocation tarayıcınız tarafından desteklenmemektedir.";  
    }  
}
```

Tarayıcı (Browser) API'leri

```
<!DOCTYPE html>
<html>
<body>

<p>Konumuzu öğrenmek için tıklayın</p>

<button onclick="getLocation()">Konum İçin Tıklayın</button>

<p id="demo"></p>

<script>
var x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Tarayıcınız geolocation servisini desteklememektedir";
  }
}
function showPosition(position) {
  x.innerHTML = "Enlem: " + position.coords.latitude +
  "<br>Boylam: " + position.coords.longitude;
}
</script>

</body>
</html>
```

Konumuzu öğrenmek için tıklayın

[Konum İçin Tıklayın](#)

Enlem: 40.7657515

Boylam: 30.365904

Tarayıcı API'leri - windows.history

- Web Geçmişi API'si, windows.history nesnesine erişmek için kolay yöntemler sağlar.
- Window.history nesnesi, kullanıcı tarafından ziyaret edilen URL'leri (Web Siteleri) içerir.

.back: Bir önceki sayfa

```
<button onclick="myFunction()">Go Back</button>
```

```
<script>
function myFunction() {
    window.history.back();
}
</script>
```

.go: Yazılan sayı kadar
ileri/geri gider

```
<button onclick="myFunction()">Go Back 2 Pages</button>
```

```
<script>
function myFunction() {
    window.history.go(-2);
}
</script>
```

Örnek API Kullanımı

- API olarak hizmet veren birçok servis vardır. Bunların başlıcaları hava durumu, haberler gibi apilerdir.
- API hizmetlerinin bir kısmı ücretli bir kısmı da ücretsizdir.
- Kullanıcılara bir tane key (anahtar kelime) verilerek kimlik doğrulaması yapılmış olur

Örnek API Kullanımı-openweathermap

- Openweathermap yaygın kullanılan hava durumu api hizmetlerinden biridir
- <http://openweathermap.org/appid> adresine girilerek sisteme üye olunması gerekmektedir.
- Sisteme giriş yapıldıktan sonra My API Keys kısmından anahtar sözcüklerinizi listelenmektedir

We have sent the confirmation link to heski@sakarya.edu.tr. Please check your email.

New Products Services API keys Billing plans Payments Block logs My orders My profile

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Create key
e569[REDACTED]e6b1	WebTek	<input type="button" value="Generate"/>

Örnek API Kullanımı-openweathermap

- Havadurumu bilgilerini sorgularken bu API key'i kullanılacaktır.
- Aşağıdaki link kullanılarak apiid'niz ve istediğiniz bölgeyi (q parametresi ile) parametre olarak göndererek JSON formatında sonucu görebilirsiniz.

[http://samples.openweathermap.org/data/2.5/forecast?q=İstenilen
İlveYayılÇe&appid=KendiAPIKeyiniz](http://samples.openweathermap.org/data/2.5/forecast?q=İstenilenİlveYayılÇe&appid=KendiAPIKeyiniz)

- Aşağıdaki örnekte de koordinata göre bilgi alınabilmektedir.
 - lat: enlem bilgisini, lon: boylam bilgisini ifade etmektedir

[http://samples.openweathermap.org/data/2.5/weather?lat=41.1
11721&lon=29.022345&appid=KendiAPIKeyiniz](http://samples.openweathermap.org/data/2.5/weather?lat=41.111721&lon=29.022345&appid=KendiAPIKeyiniz)

Örnek API Kullanımı-openweathermap

- Kırmızı ile yazılmış yerleri değiştirerek tarayıcınıza girdiğinizde aşağıdakine benzer sonuç görünür.

```
JSON Ham veri Üst bilgiler
Kaydet Kopyala Tümünü daralt Tümünü genişlet JSON'ı filtrele
cod: "200"
message: 0.0032
cnt: 36
list:
  0:
    dt: 1487246400
    main:
      temp: 286.67
      temp_min: 281.556
      temp_max: 286.67
      pressure: 972.73
      sea_level: 1046.46
      grnd_level: 972.73
      humidity: 75
      temp_kf: 5.11
    weather:
      0:
        id: 800
        main: "Clear"
        description: "clear sky"
        icon: "01d"
```

Örnek API Kullanımı-openweathermap

- Web sayfamızda istediğimiz yere is'si **acıklama, sıcaklık ve konum** olan divler yerleştirilir. Kırmızı ile verilen yerleri kendiniz girmeniz gerekmektedir.

```
<button id = "Goster" onclick = "HavaDurumuGetir('sakarya','kendi api keyiniz')">Hava Durumunu Göster</button>
<div>
    <div id="aciklama"></div>
    <h1 id="sicaklik"></h1>
    <div id="konum"></div>
</div>
```

Örnek API Kullanımı-openweathermap

- Javascript Kodları: Butona tıklandığında api'den bilgileri alıp istenilen formata dönüştüren Javascript kodları

```
function HavaDurumuGetir( cityID, key ) {  
    fetch('https://api.openweathermap.org/data/2.5/weather?q=' + cityID+ '&lang=tr&appid=' + key )  
    .then(function(resp) { return resp.json() })  
    .then(function(data) {  
        HavaDurumuGoster(data);  
    });  
}  
  
function HavaDurumuGoster( d ) {  
    var celcius = Math.round(parseFloat(d.main.temp)-273.15);  
    var fahrenheit = Math.round(((parseFloat(d.main.temp)-273.15)*1.8)+32);  
    var aciklama = d.weather[0].description;  
    document.getElementById('aciklama').innerHTML = aciklama;  
    document.getElementById('sicaklik').innerHTML = celcius + '&deg;';  
    document.getElementById('konum').innerHTML = d.name;  
}
```

Örnek API Kullanımı-openweathermap

Açılış ekranı

Hava Durumunu Göster

Butona Tıklandığında

Hava Durumunu Göster

parçalı az bulutlu

23°

Sakarya

Örnek API Kullanımı-openweathermap

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
<script>
function HavaDurumuGetir( cityID, key ) {
  fetch('https://api.openweathermap.org/data/2.5/weather?q=' + cityID+ '&lang=tr&appid=' + key )
    .then(function(resp) { return resp.json() })
    .then(function(data) {
      HavaDurumuGoster(data);
    });
}

function HavaDurumuGoster( d ) {
  var celcius = Math.round(parseFloat(d.main.temp)-273.15);
  var fahrenheit = Math.round(((parseFloat(d.main.temp)-273.15)*1.8)+32);
  var aciklama = d.weather[0].description;
  document.getElementById('aciklama').innerHTML = aciklama;
  document.getElementById('sicaklik').innerHTML = celcius + '&deg;';
  document.getElementById('konum').innerHTML = d.name;
}
</script>
</head>
<body>
<button id = "Goster" onclick = "HavaDurumuGetir('sakarya','e566b1')">Hava Durumunu Göster</button>
<div>
  <div id="aciklama"></div>
  <h1 id="sicaklik"></h1>
  <div id="konum"></div>
</div>
</body>
</html>
```



Kaynaklar:

- <http://www.w3schools.com/>
- <https://medium.com/@akanesen/api-nedir-entegrasyonu-nas%C4%B1l-yap%C4%B1%C4%B1r-578f93578e46>
- <http://bahadir.almaci.com/2016/01/hava-durumu-apisi/>