

İÇİNDEKİLER

HTML	2
xHTML	3
xHTML KURALLARI	3
HTML GELİŞİMİ	3
HTML5 Temel Yapısı	3
Temel Etiketler	3
ID Attribute	7
CLASS Attribute	7
Yapısal Etiketler	8
Yapısal Etiketler	10
İçerik Etiketleri	14
Uygulama Etiketleri	16
Tablolar	19
CSS 3.0	19
CSS 3.0 Seçiciler	19
Tip Seçiciler	19
Evrensel Seçiciler	20
ID Seçiciler	21
CLASS Seçiciler	21
SOY Seçiciler	22
JAVASCRIPT	22
Javascript'ın Özellikleri	22
Javascript Kodlarına Giriş	23
Veri Türleri	23
Birincil(Primary) Veri Türleri	23
Bileşik(Composite) Veri Türleri	23
Özel(Special) Veri Türleri	23
String Veri Türü	23
Int Veri Türü	23
Boolean Veri Türü	23
NULL Veri Türü	24
Event'lar	24
OnChange Event'i	24
OnClick Event'i	24
OnMouseOver Event'i	25
OnMouseOut Event'i	25
OnKeyDown Event'i	26
OnLoad Event'i	26
Javascript'de Hata Yönetimi	27
Document.Write	28
Alert	28
Javascript'de DOM (Document Object Model)	29
GetElementById	30
Element Attribute'u Değiştirme	31
Style Değiştirme	32
Validation Uygulama(Kontrol Etmek)	32
Javascript'de Object Oriented Programming	33
Javascript İçerisinde Object Yaratmak	33
Javascript Nesne Yönelimli Programlamada Kalıtımı Kullanmak (Inheritance)	33

Local ve Session Storage	34
Local Storage	34
Session Storage	35
JQUERY	35
CDN-Content Delivery Network.....	35
Javascriptin Çalışma Mantığı.....	35
Jquery Sözdizimi (Syntax).....	36
Projede Jquery Kullanımı.....	36
1.Yol;.....	36
2.Yol;.....	37
3.Yol;.....	37
Jquery İçerisinde Hiyerarşik Yapı	38
.siblings()	38
.children()	39
.next() ve nextAll().....	39
.prev() ve prevAll().....	41
.closest() ve parents().....	41
.find().....	42
Jquery içerisinde Css.....	43
Jquery Animasyonlar ve Olaylar	45
Nerelerde Kullanılıyor?	45
BOOTSTRAP.....	47
Bootstrap Nasıl Kurulur ?.....	47
1.Yol;.....	48
2.Yol;.....	48
Bootstrap nasıl kullanılır?	48
Bootstrap Containers.....	49
Bootstrap Media Tag.....	50
Bootstrap Grid(Izgara Sistemi)	50
Bootstrap Grid Sınıfları.....	51
Bootstrap içerisinde Input ve Button Kullanımları	52
Bootstrap içerisinde İkon,Navigasyon ve Uyarılar.....	55
İkonlar	55
Navigation Bar.....	56
Uyarılar.....	56
jQuery ile Bootstrap Beraber Kullanımı	58

HTML

HTML(Hyper Text Markup Language-Zengin Metin İşaretleme Dili) web sayfaları oluşturmak, yapısal olarak metinleri düzenleyebildiğimiz metin tabanlı bir dildir. Kesinlikle bir programlama dili değildir fakat HTML'i yorumlayabilen programlama dilleri kullanılarak web ve mobil uygulamalar geliştirebilirsiniz.

HTML, içerisinde tag yani etiket denilen ifadeler ile kullanılır. Bu ifadeler ise W3C yani World Wide Web Consortium(Dünya Ağ Birliği) tarafından web sayfaları konusunda standartları belirlemek için kurulmuş bir organizasyon tarafından standartlaştırılır ve kullanıma sunulur.

Web kavramı ise, internet üzerinde ki servislerden birisidir.İnternet üzerinden yazı,grafik,resim,ses ve hareketli görüntülerden oluşan dökümanları uzakta ki bilgisayarlara iletir. 1980 yılında CERN'de görevli olan Tim Berners-Lee adlı bilgisayar programcısı tarafından ,CERN araştırmacılarının bilgilerini ve dökümanlarını birbirleriyle paylaşabilmeleri açısından HTML geliştirilmiş ve Web üzerinden kullanılmaya başlanılmıştır.

1990 yılında HTML ilk versiyonu standardize edilmiş ve WWW yani Worl Wide Web sistemini kurduktan sonra kullanıma açılmış takiben gelecek olan süreçte ise günlük hayatımızın vazgeçilmez bir parçası olan internetin başlıca temelini oluşturmuştur.

xHTML

xHTML, açılımıyla **Extensible Hyper Text Markup Language** yani Türkçe' si ile "**Geliştirilebilir Metin İşaretleme Dili**" anlamına gelen, **istemci** taraflı (**client side**) çalışan bir **HTML** dilidir. HTML ve XML'in birleşiminden oluşur. HTML'nin XML sözdizimine göre düzenlenmesidir. Bu yüzden XHTML'de kurallar katıdır.

xHTML KURALLARI

- xHTML DOCTYPE tanımlaması olmalı.
- Html etiketinde xmlns (xmlns özelliği belge içim xml ad alanını belirtir.) özelliği tanımlanmalı.
- 4 etiket(Html,head,title,body) etiketleri kesinlikle olmalı.
- İç içe oluşturulan etiketler muazzam bir şekilde kapatılmalı.
- Etiketler ve etiket özellikleri küçük harf ile tanımlanmalı.

HTML GELİŞİMİ

- HTML 1991
- HTML 2.0 1995
- HTML 3.2 1997
- HTML 4.01 1999
- XHTML 2000
- HTML5 2014

HTML5 Temel Yapısı

En yalın hali ile bir adet yapısını aşağıda görmekteyiz

HTML5 Temel Yapısı

```
<!DOCTYPE html>
<html>
<head>
  <title>Bilge Adam</title>
  <meta charset="utf-8" />
</head>
<body>
</body>
</html>
```

Yukarıda görmüş olduğumuz yapı içerisinde her bir küçüktür '<' ve büyüktür '>' sembolleri arasında kalan ifadelere tag yani etiket diyoruz. Bu etiketlerin her biri tam olarak ne anlama geliyor şimdi onlara bir gözatalım sizinle.

Temel Etiketler

```
<!DOCTYPE html>
```

Doctype, hazırladığımız sayfaların hangi tipte bir belge olduğunu tarayıcıya yani web browser a bildirmek için kullanılır. Bu sayede tarayıcı sayfayı hangi HTML standartlarına göre yorumlayacağını bilir ve ona göre yorumlar aksi durumda ise her bir tarayıcı kendi algıladığı şekilde dili yorumlar ve yazmış olduğumuz HTML tagleri tarayıcıdan tarayıcıya davranış değişikliği gösterir.

<html>

Yukarıda görmüş olduğunuz <html> etiketi diğer tüm tagleri kapsayıcı etiket olarak kullanılır. Tam olarak html taglerinden faydalananarak bir web sayfası oluşturmak istiyorsak tarayıcımıza en başta

<!DOCTYPE html> tagi kullanarak önce tip bilgisi veriyoruz devamında ise genel etiket olan <html> içerisine diğer etiketlerimizi kullanarak yapıyı oluşturuyoruz.

<head>

Hazırlayacağımız web sayfası ile ilgili bilgilerin bulunduğu sayfa başlığı(<title>), link özellikleri, siteyi tarayıcıya ve arama motorlarına tanıtmak amacıyla kullanılan <meta> etiketlerinin bulunduğu kısımdır. Bunun yanı sıra title, scripts, meta, css, link gibi etiketleri barındırmaktadır.

<meta />

Meta etiketi tam olarak yazmış olduğumuz HTML sayfamızın HEAD kısmına yerleştirilmiş bilgi etiketleridir. Arama motorlarında site ve sitemizin içeriği hakkında bilgi sunduğumuz kısımdır. Meta etiketi içerisinde kullanılan charset ise;

```
<meta charset="utf-8" />
```

Sayfada kullanılan karakter kümesini tanımlamak için kullanılır. Bildiğimiz gibi her dilde gösteriş şekli farklı olan harfler ya da bir takım formülüne edilmiş ifadeler olabilir ve her dil bunları desteklemeyebilir. Bunun önüne geçebilmek için html 5 ile beraber charset gelmiştir. Charset içerisinde utf yani Unicode Transformation Format kullanılarak sayfamıza istediğimiz karakter kümesinin desteğini sağlayabiliriz.

```
<meta name="author" content="Bilge Adam"/>
```

Yukarıda ki kısımda ise meta etiketimizin içerisine ait alt özellikleri tanımlamak için name özelliğini kullandık. Alt özelliği tanımladıktan sonra ise ona içerik atamak için content özelliği kullanılır. Bu örneğimiz de author yani üzerinde bulunduğumuz güncel sayfanın yazarının veya yazar ait iletişim bilgilerinin belirtilmesini sağlar.

```
<meta name="abstract" content="Bilişim sektöründe öncü kurum"/>
```

Üzerinde bulunduğumuz güncel sayfanın kısa özetinin belirtilmesini sağlar.

```
<meta name="content-language" content="tr"/>
```

Lisan Kodları listesinde ki dilerden uygun olan lisanı seçerek güncel sayfamızın içerik dilini belirlemiş oluruz.

```
<meta name="content-language" content="tr-TR"/>
```

Ayrıca içerisinde dil hangi ülkeye ait ise o da belirtilebilir.

```
<meta name="content-type" content="text-html" charset=utf-8/>
```

Sayfanın içerik tipini ve karakter setini belirtir.

```
<meta name="copyright" content="©2017 bilge adam-copy all right" />
```

Sayfanın telif hakkı ve bilgisinin belirtilmesini sağlar.

```
<meta name="description" content="Bu sayfa bilge adam hakkında bilgi  
vermektedir." />
```

Güncel sayfamız hakkında açıklamalara yer verilir.

```
<meta name="designer" content="bilge adam akademi-www.bilgeadam.com"/>
```

Sayfanın ya da sitenin tasarımı hakkında bilgi verir.

```
<meta name="distribution" content="local"/>
```

Distribution ise sayfanın ya da sitenin hitap ettiği bölgenin belirtilmesini sağlar. Content içerisinde alacak 3 farklı paramtre vardır.

- Global:Tüm internet ağına hitap ettiğini belirtir.
- Local:Sitenin bulunduğu IP(Internet Protocol) adres bloğunun ait olduğu ülkeye hitap ettiğini belirtir.
- IU(International Use): Kamuya hitap etmediğini belirtir.

```
<meta name="expires" content="Tue,28 feb 2017 23:59:59 gtm"/>
```

Sayfanın geçerli olduğu tarih belirtilir. Sayfaya herhangi bir geçerlilik tarihi vermek istemediğimiz yani daimi olarak kullanmak için ise content özelliğine '0'(sıfır) değeri atanmalıdır.

```
<meta name="generator" content="Visual Studio 2015"/>
```

Sayfanın oluşturulması esnasında kullanılan program isminin ve sürümünün belirtilmesini sağlar.

```
<meta name="keywords" content="html,css,javascript,jquery,bootstrap"/>
```

Sayfanın içeriğini tanımlayacak virgül ile ayrılmış anahtar kelimeleri belirtir.

```
<meta name="publisher" content="VS 2015 "/>
```

Publisher sayfayı yayınlayan uygulamanın isminin ve sürümünün belirtilmesini sağlar.

```
<meta name="reply-to" content="bence@bilgeadam.com"/>
```

Sayfa hakkında E-Posta adresi belirtilmesini sağlar.

Robots: Meta etiketler,web yöneticilerinin arama motorlarına siteleri hakkında bilgi vermesi için kullanıyoruz fakat bu meta etiketler arama motoru tarama ve dizine ekleme işlemlerinin davranışlarını kontrol edebilir ve bu sayede arama motorları tarafından uygulamamamız daha rahat dizinlenebilir. Robots kullanımı esnasında içerisinde ki content özelliği bir ya da virgül ile ayrılarak daha fazla değer alabilir.

- index:Sayfanın dizine eklenmesini bildirir.
- follow:Sayfa içinde ki bağlantıların/linklerin izlenmesini bildirir.
- archive:Sayfanın önlzleme için ekran görüntüsünün oluşturulmasını bildirir.
- noindex:Sayfanın dizine eklenmesini engeller.
- nofollow:Sayfa içinde ki bağlantıların/linklerin izlenmesini engeller.
- noarchive: Sayfanın ön izleme için ekran görüntüsünün oluşturulmasını engeller. Daha önce oluşturulduysa silinmesini sağlar. (Ön izlemenin silinmesi arama motorlarına göre farklılık gösterebilir.)
- all: Tüm olumlu bildirimleri uygular. Arama motorları index, follow ve archive bildirimlerinden farklı bildirimleri de kabul ediyor olabilir.
- none: Sayfanın göz ardı edilmesini bildirir.
- noodp: Eğer sayfa DMOZ dizininde kayıtlı ise arama sonuçlarında DMOZ dizinindeki açıklamanın yerine sayfanın kendi açıklamasının kullanılmasını sağlar.

Bir arama motoru robotu aksi belirtilmediği sürece bir sayfayı dizine ekleyebilir, sayfadaki linkleri takip edebilir ve sayfanın ön izlemesini oluşturabilir. Bu yüzden robots özelliği ile **all**, **index**, **follow** ve **archive** değerlerinin kullanımının çok özel bir anlamı yoktur. Bunlar yerine sadece robotu engellemek amacıyla, engelleme amacına göre **none**, **noindex**, **nofollow** ve **noarchive** değerlerinin kullanımı daha doğru olacaktır.

Index,follow ve archive değerlerinin uygulanması:

```
<meta name="robots" content="all" />
```

Arama motoruna sayfayı dizine eklemesini fakat sayfadaki linkleri takip etmemesini ve sayfanın ön izlemesinin oluşturulmamasının bildirilmesini göstermektedir.

```
<meta name="robots" content="index,nofollow,noarchive" />
```

- revisit-after: Arama motorlarına gün olarak sonraki ziyaretlerini gerçekleştirme zamanının bildirilmesini sağlar.

```
<meta name="revisit-after" content="4" />
```

```
<meta name="revisit-after" content="4 days" />
```

Refresh: Belirli bir süre sonunda belirtilen bir URL adresinin yüklenmesini sağlar. URL adresi belirtilmezse güncel sayfa adresi yeniden yüklenir.

```
<meta name="refresh " content="10" />
```

```
<meta name="refresh " content="60; http://www.bilgeadam.com/" />
```

<title>

Tarayıcının web sayfamızı görüntülediği anda pencere başlığında ve sekme başlığında gösterilecek olan metnin tanımlandığı etikettir.

```
<title>Bilge Adam Yazılım</title>
```

<script>

Sayfamıza javascript kod bloklarını eklememiz için kullanılan etikettir. Head etiketi içerisinde kullanılmasının yanısıra body etiketi içerisinde de kullanılabilir.

Özellikler:

src: {url} harici bir script dosyasını belgeye dâhil etmek için kullanılır. Bu özellik kullanılırsa <script></script> blokları arasına kod yazılamaz.

async: Harici bir JavaScript dosyasındaki kodların eş zamansız çalışmasını sağlayan özelliktir.

defer: Bu özellik sadece src özelliği ile harici bir JavaScript dosyası eklendiyse çalışır. Sonradan dâhil edilen JavaScript kodlarının sayfa yüklendikten sonra çalışmasını sağlar.

Script Örneği

```
<script type="text/javascript">
  var d = new Date();
  var days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];
  document.getElementById("demo").innerHTML = days[d.getDay()];
</script>
```

<style>

Head etiketleri içerisinde CSS kodları yazmak için kullanılır.

Özellikleri:

- **type:** style etiketi için içerik türü tanımlanır.

Style Örneği

```
<style type="text/css">
  body {
    font-family: Verdana;
    font-size: 9pt;
  }
```

```
</style>
```

```
<body>
```

Sayfanın gövdesini oluşturan etikettir. İki gövde arasında yazdığımız tüm kodlar kullanıcıya gösterilecektir.

Örnek

```
<body>
<p>Son Kullanıcıya gösterilecek olan içerikler bu alana yazılacak.</p>
</body>
```

ID Attribute

HTML elemanına benzersiz bir isim vermek için kullanılır.

Örnek

```
<div id="menu"></div>
```

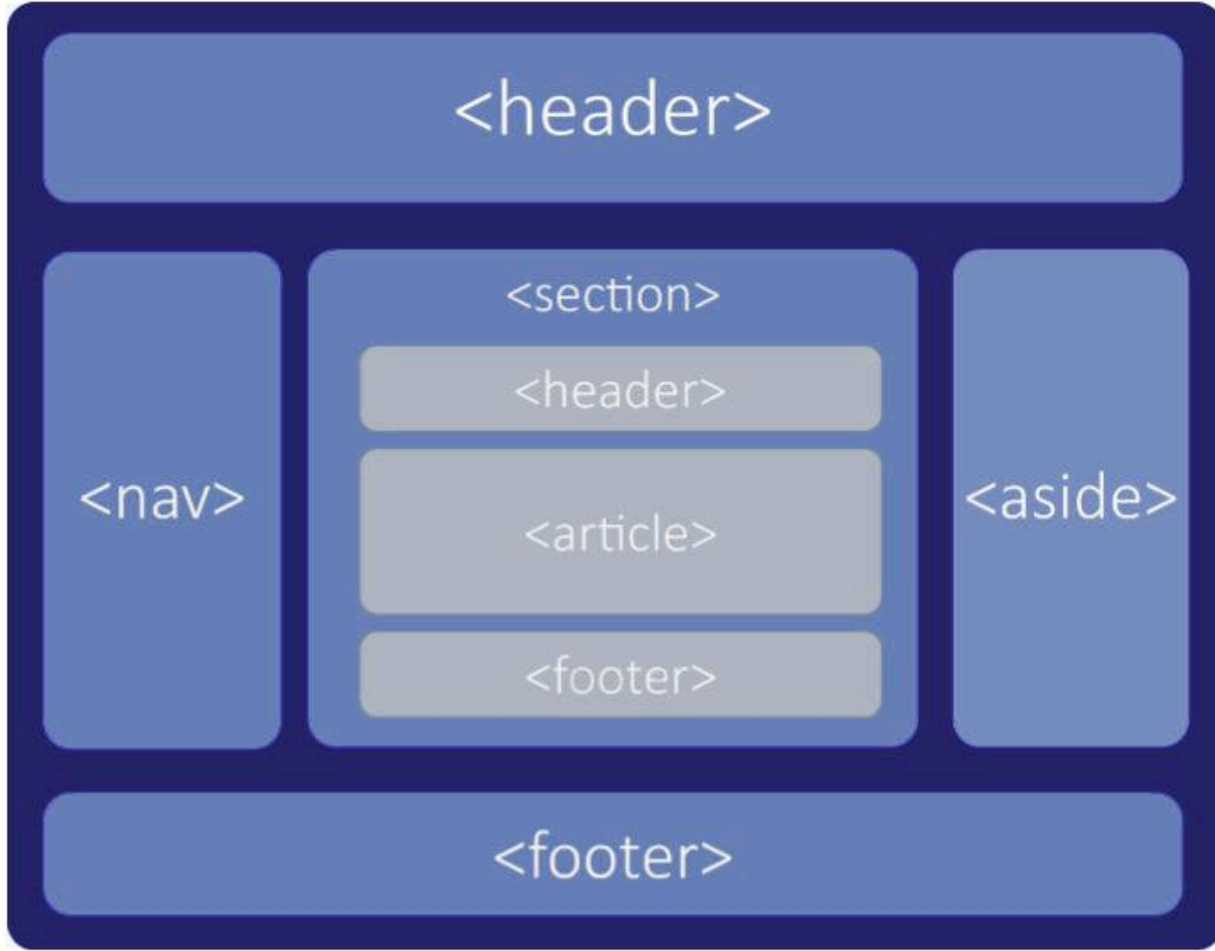
CLASS Attribute

CSS kodlarının atanması için kullanılan etikettir. Bir elemana birden fazla class tanımlayabilirsiniz.

Örnek

```
<div id="menu" class=".ImageBicim"></div>
```

Yapısal Etiketler



Yapısal etiketler tasarlayacağımız web sitesinin yapılarını belirleyen temel etiketlerdir. HTML5 ile gelen bu etiketler sitemizin daha düzenli bir yapıya sahip olmasını sağlar. Google gibi arama motorları web sitesinde başlıklara göre index'leme yaptığından dolayı bu etiketler içerisinde yer alan içerikler başlıklara göre gruplanıp kullanıcının arama sonuçlarına yansıtılmaktadır. Arama motorları tarafından tercih edilen bir sayfanız olsun istiyorsanız, uygun başlıklar içinde gerekli kodlamaları yapmalısınız.

<header>

Sayfanın başlık kısmının tanımlandığı ve içerisinde bulunması gereken yapıların gruplandığı etikettir. İçerisinde H1, H2, H3, H4, H5, H6 gibi başlık etiketlerinin yanı sıra logo banner gibi yapılarda bulunabilir. Bir sayfada birden fazla <header> etiketi kullanılabilir. Header etiketi section, aside ve article etiketleri içerisinde de kullanılabilir.

Örnek

```
<header>
  
  <h1>Bilge Adam Akademi</h1>
</header>
```

<NAV>

Nav etiketi sayfa içerisinde kullandığımız ileri, geri, başa dön gibi navigasyon bağlantılarının gruplanmasında kullanılır.

Örnek

```
<nav>
  <ul>
    <li> <a href="#">Software Development</a> </li>
    <li> <a href="#">Web Graphic</a> </li>
    <li> <a href="#">System</a> </li>
    <li> <a href="#">Microsoft Office</a> </li>
  </ul>
</nav>
```

<SECTION>

Sayfamızda içerik, navigasyon, menü ve alt menü gibi div'ler kullanırız. HTML5 ile bu gruplamaları section etiketi altında toplayarak içerisinde div kullanmaya devam edebilmekteyiz.

Örnek

```
<section>
  <header> Başlık </header>
  <article> İçerik </article>
  <footer> Alt Başlık </footer>
</section>
```

<ARTICLE>

Sayfada, uygulamada veya belgede kendi içinde düzen oluşturabilen yapısal etikettir. Article etiketi sayfa içerisinde yeniden kullanılabilir olarak tasarlanmıştır. Blog sayfasındaki yorumlar gibi ya da ürünlere yapılan yorumlar gibi sayfadan bağımsız hareket bağımsız nesneleri tutabilen etikettir.

Örnek

```
<article>
  <p> Content Text </p>
</article>
```

<ASIDE>

Sayfamızda yan menü gibi oluşturduğumuz bölümleri gruplamak için kullandığımız etikettir. İçerisinde başlık, parağraf ve tablo gibi etiketler kullanılabilir. Aside etiketi içerisinde genelde reklam alanları, banner' lar gibi yapıları gruplandırmak için kullanabiliriz.

<HGROUP>

H1, H2, H3, H4, H5, H6 başlık etiketlerinin gruplanmasını sağlayan etikettir.

Örnek

```
<hgroup>
  <h1> Yazılım </h1>
  <h2> Sistem </h2>
  <h3> Web Grafik </h3>
  <h4> Ofis </h4>
  <h5> İngilizce </h5>
```

`<h6> 3D </h6>`
`</hgroup>`

<FOOTER>

Footer' da header etiketi gibi sayfanın altında bulunması gereken yapıların gruplanması için kullanılan etikettir.

Örnek

```
<footer>
  <p>Bilge Adam Akademi</p>
  <p>© Tüm Hakları Saklıdır - 2015</p>
</footer>
```

Yapısal Etiketler

Sayfa içerisinde başlık oluşturmak için kullanılır.

Kaynak Kod	Ekran Çıktısı
<code><h1> Bilge Adam </h1></code>	Bilge Adam
<code><h2> Bilge Adam </h2></code>	Bilge Adam
<code><h3> Bilge Adam </h3></code>	Bilge Adam
<code><h4> Bilge Adam </h4></code>	Bilge Adam
<code><h5> Bilge Adam </h5></code>	Bilge Adam
<code><h6> Bilge Adam </h6></code>	Bilge Adam

<p>

Parağraf oluşturmak için kullanılır.

Kullanım şekli:

Örnek

```
<p>Kullanıcıya gösterilecek içerik.</p>
```

<HR>

Doküman içerisinde eklendiği yere yatay çizgi çeker.

Örnek

```
<hr/>
```


Satır sonu oluşturmak için yani bir alt satıra geçmek için kullanılır.

Örnek`
`****

Sıralı liste oluşturmak için kullanılır. Li elemanlarını barındırır.

Özellikleri:

- **start:** İlk liste öğesinin (li) sıra değerini tanımlamak için kullanılır

Kaynak Kod	Ekran Çıktısı
<code><ol start="10"></code> <code> Bilge Adam </code> <code></code>	1.Bilge Adam

- **reversed:** Liste elemanlarının sayısı kadar rakamsal olarak geriye doğru sıralanması için kullanılır.

Kaynak Kod	Ekran Çıktısı
<code><ol reversed></code> <code> Yazılım </code> <code> Sistem </code> <code> Web Grafik </code> <code> Ofis </code> <code> 3D </code> <code> İngilizce </code> <code></code>	6. Yazılım 5. Sistem 4. Web Grafik 3. 3D 2. Ofis 1. İngilizce

- **type:** Liste elemanları önüne koyulacak karakter tipini belirlemek için kullanılır. 1, A, a, I, i değerlerinden birini alır. Varsayılan olarak 1 gelmektedir.

Kaynak Kod	Ekran Çıktısı
<code><ol type="I"></code> <code> Yazılım </code> <code> Sistem </code> <code> Web Grafik </code> <code> Ofis </code> <code> 3D </code> <code> İngilizce </code> <code></code>	I. Yazılım II. Sistem III. Web Grafik IV. Ofis V. 3D VI. İngilizce

Sıralı olmayan liste oluşturmak için kullanılır.

li ve ol etiketleri içinde liste elemanı oluşturmak için kullanılır.

Özellikleri

- value: li elemanı ol etiketi içerisinde kullanıldığında da kullanılabilecek bir özelliktir. Kullanıldığı li elemanı için sıra değeri tanımlar.

Örnek

```
<ol>
<li value="2015"> Bilge Adam </li>
</ol>
```

<DIV>

Sayfa içeriğini birden fazla alana bölmeye ve görsel alanda bir nevi blok oluşturmak için kullandığımız etikettir.

Örnek

```
<div>
<h1> Başlık </h1>
<p> İçerik </p>

</div>
```

<A>

Sayfa içerisinde, bir projenin altındaki ayrı sayfaya bağlanmak ya da proje dışarısından bir linke bağlanmak amacıyla kullanılır. 2 farklı kullanım şekli vardır;

- href: Link üzerine tıklandığında yönlendirilecek sayfanın URL 'ini tanımlar.
- target: Açılacak olan sayfanın nerede açılacağını belirler.

_blank: Bağlantılı sayfa yeni bir pencerede açılır.

_self: Bağlantılı sayfa aynı pencerede açılır.

Örnek

```
<a href="#"> Page Title</a>
```


Sayfa içerisinde vurgulu metinler oluşturmak için kullanılır. (oluşturduğu metni italik yapar).Kısaltmasının EM olmasının sebebi ingilizce emphasize yani ansiklopedik yazı karakteri anlamına gelmesidir.

Örnek

```
<em> Öne çıkmasını istediğimiz metin. </em>
```


Etiketler içerisine yazılmış içeriğin daha kalın ve belirgin bir şekilde yazılmasını sağlar.

Örnek

**** Kalın yazılmasını istediğimiz metin. ****

<SMALL>

İçerdiği metnin küçük boyutlu olmasını sağlar.

Örnek

```
<small> Kalın yazılmasını istediğimiz metin. </small>
```

<S>

Sayfa içerisinde kullanıldığında içerdiği metnin üzerini çizerek.

Örnek

```
<s> Üzeri çizili metin. </s>
```

<I>

Sayfa içerisinde kullanıldığı alanda içeriği italik yapar.

Örnek

```
<i>İtalik yazılacak alan</i>
```


Sayfa içerisinde kullanıldığı alanda içeriği kalın yapar.

Örnek

```
<b>Kalın yazılacak alan</b>
```

<U>

Sayfa içerisinde kullanıldığı alanda içeriğin altını çizerek.

Örnek

```
<u>Altı çizili alan</u>
```

İçerik Etiketleri

<FIGURE>

Bir bölümün yerleşik içeriğinin gruplanması için kullanılır. Örnek olarak bir ya da birden çok video veya resimlerin gruplandırılması için kullanabiliriz.

- figcaption**: Bu etiketi figure etiketi içerisinde metin eklemek için kullanılır.

Örnek

```
<figure>  
    
    
    
    
  <figcaption> Bilge Adam Bireysel Eğitimleri </figcaption>
```

</figure>

<VIDEO>

Html'in önceki sürümlerinde video etiketi diye bir şey yoktu ve herhangi bir web sayfası üzerinde video oynatabilmek için microsoft'un geliştirmiş olduğu Silverlight ya da Adobe'un geliştirmiş olduğu flashplayera ihtiyacımız vardı fakat bu durum kullanıcıların video izlerken web tarayıcılarına fazladan bir eklenti kurmalarını gerektiriyordu. HTML5 sayesinde gelen video etiketi sayesinde format ayırt etmeksizin ve eklenti gerekmeden video dosyası oynatabilmekteyiz.

Örnek

```
<video controls="controls">
  <source src="file.mp4" type="video/mp4" />
  <source src="file.webm" type="video/webm" />
  <source src="file.ogv" type="video/ogg" />
</video>
```

<AUDIO>

Multi medya dosyalarını (flash, video vs...) sitemize eklemek için kullanabileceğimiz etikettir.

Örnek

```
<embed src="file/intro.swf" type="application/x-shockwave-flash" />
```

<AUDIO>

Html sayfamızın içerisine ses dosyalarını eklememize yarayan etikettir.

Özellikleri:

- src: Audio etiketi tarafından alınacak olan ses dosyasını tanımlamak için kullanılır.
- controls: Denetim çubuğunun gösterilim gösterilmemesini ayarlamak için kullanılır.
- autoplay: Sayfa yüklendiğinde ses dosyasının otomatik olarak çalmasını sağlar.
- preload: Sayfa yüklendiğinde medya içeriğinin çalışmaya hazır olması için ön belleğe alınmasını sağlar. (autoplay değeri verilmiş ise, bu işlem otomatik olarak yapılır)
- loop: Ses dosyasının sayfa içerisinde sürekli çalmasını sağlar.

Örnek

```
<audio controls="controls">
  <source src="file.mp3" type="audio/mp3" />
  <source src="file.ogg" type="audio/ogg" />
</audio>
```

<CANVAS>

Tarayıcı üzerinde ek bir araca ihtiyaç duymadan çizim ve animasyonlar oluşturup, metinler yazdırabileceğimiz ve geçişler oluşturabileceğimiz bir kontroldür.

Örnek

```
<canvas></canvas>
```

<IFRAME>

Tarayıcı üzerinde başka bir web sayfasının içeriğini görüntülemek için kullanılan etikettir.

Örnek

```
<iframe></iframe>
```

Uygulama Etiketleri

<METER>

Belirli bir aralık içerisinde sayısal bir ölçüm ya da kesirli bir değer tanımlar.

Örnek

```
<meter value="0.5"></meter>
```

<PROGRESS>

Bir işlemin ilerleme sürecini kullanıcıya görsel olarak göstermek için kullanılır.

Örnek

```
<progress></progress>
```

<TIME>

Sayfa içerisinde kullandığımız alanda kesin bir tarih eklememize yarar.

Örnek

```
<p>Kurs<time datetime="2017-02-20"> bugün </time>başlıyor</p></head>
```

<DETAILS>

Sayfa içerisinde'ki içeriklerin gösterilip gizlenmesine yarayan etikettir.

Örnek

```
<details>
  <summary> Bilge Adam </summary>
  <p> Yazılım </p>
  <p> Sistem </p>
  <p> Web Grafik </p>
</details>
```

<FORM>

Form alanı oluşturmak için kullanılan etikettir. İçerisinde form elemanları tanımlanır.

Özellikleri:

- **accept-charset**: Sunucu tarafından kabul edilecek karakter setini tanımlar.
- **action**: Form verilerinin gönderileceği sayfayı tanımlar.

- **autocomplete**: Tarayıcı, kullanıcının daha önce veri alanına girmiş olduğu verilere göre otomatik tanımlama yapıp yapmamasını kontrol eder. (on ya da off).
- **enctype**: Şifreleme türünü ayarlar.
- **name**: Formun adını tanımlar.
- **method**: Bilgilerin gönderme şeklini ayarlar (get yada post).

Örnek

```
<form action="send.aspx" method="post">
  <label>Kullanıcı Adı :</label>
  <input type="text" id="txtKullaniciAdi" /><br />
  <label>Şifre :</label>
  <input type="text" id="txtSifre"
</form>
```

<OUTPUT>

Output etiketi bir hesaplamanın sonucunu göstermek için kullanılır.

Örnek

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">200
  <input type="range" id="a" value="50">100+
  <input type="number" id="b" value="50">=<output name="x" for="a b">
</output>
</form>
```

<SELECT>

C# tarafında ComboBox 'a benzeyen bir yapıdır. İçerisinde sıralı olarak listelemek istediğiniz bilgileri ekleyebileceğiniz bir etikettir. İçerisinde Option elemanlarını barındırır.

Özellikleri:

- **autofocus**: Sayfa yüklendiğinde seçili eleman olarak gelmesini sağlar.
- **disabled**: Elemanın pasif olmasını sağlar.
- **multiple**: Kullanıcının birden fazla seçim yapabilmesine izin verir.
- **required**: Kullanıcı için girilmesi zorunlu alan olmasını sağlar.

Örnek

```
<select>
  <optgroup>
    <option>Bilge Adam</option>
  </optgroup>
</select>
```

<OPTGROUP>

Liste elemanlarını gruplandırmak için kullanılır.

Özellikleri:

- **disabled**: Grup elemanı içerisindeki seçeneklerini pasif yapmak için kullanılır.
- **label**: Oluşturulan grup için başlık tanımlamamızı sağlar.

Örnek

```
<optgroup>
  <option>Bilge Adam</option>
</optgroup>
```

<OPTION>

Liste elemanı oluşturmak için kullandığımız etikettir.

Özellikleri:

- **disabled**: elemanı pasif yapmak için kullanılır.
- **selected**: Atanan elemanın seçili olarak gelmesini sağlar.
- **value**: Liste seçeneği için bir değer tutar. Formun sunucu tarafında çalışan sayfaya gönderildiğinde bu değer üzerinden işlem yapılır

Örnek

```
<select>
  <optgroup label="Bilge Adam">
    <option>Yazılım</option>
    <option>Sistem</option>
    <option>Web Grafik</option>
    <option>3D</option>
  </optgroup>
</select>
```

<TEXTAREA>

Tek satırdan fazlasına sahip olan metin giriş alanları oluşturmak için kullanılır.

Özellikleri:

- **cols**: Grup elemanı içerisindeki seçeneklerini pasif yapmak için kullanılır.
- **rows**: Oluşturulan grup için başlık tanımlamamızı sağlar.
- **maxlength**: Metin alanına girilecek olan maksimum karakter uzunluğunu belirler.
- **readonly**: İçeriğin sadece okunabilir olmasını sağlar.
- **required**: Zorunlu olarak doldurulmasını sağlar.
- **Autofocus**: Sayfa yüklendiğinde seçili olmasını sağlar.

Örnek

```
<textarea readonly="readonly"> </textarea>
```

<INPUT>

Sayfa ve Form içerisinde type özelliğine göre eleman oluşturur.

Örnek

```
<input type="button" />
<input type="checkbox" />
<input type="text" />
```

Tablolar

<TABLE>

Sayfa içerisinde belirli bir alanda tablolar oluşturmak için kullanılan kapsayıcı etikettir.

Örnek:

```
<table></table>
```

Tablo içeriği genelde 3 bölümden oluşur. Başlık (thead), içerik(tbody) ve alt bölüm(tfoot).

<THEAD>

Tablo üst bilgisini içeren satırları tanımlamak için kullanılır.

<TBODY>

Tablo gövdesini oluşturan satırları tanımlamak için kullanılır.

<TFOOT>

Tablo alt bölüm kısmını oluşturan satırları tanımlamak için kullanılır.

<TR>

Tablo içerisinde satır oluşturmak için kullanılır.

<TD>

<tr> etiketi ile tanımlanan satırların içerisinde hücre oluşturmak için kullanılır.

<TH>

<tr> etiketi ile tanımlanan satırların içerisinde kalın ve ortaya hizalı başlık oluşturmak için kullanılır.

CSS 3.0

Cascading Style Sheet yani Basamaklı Stil Şablonları bir diğer deyişle Basamaklı Biçim Sayfaları. Html üzerinde çalışırken görsel anlamda bir takım yetersizliklerin sonucunda ortaya çıkmış metin ve format biçimlendirme alanında fazladan olanaklar sunan WEB teknolojisidir.

CSS kodları, Html kodlarının içine yazılabildiği gibi harici bir CSS dosyası oluşturularak da işlem yapılabilir. CSS; kuralları, söz dizimi ve özellikleri olan bir yapıdır. Bu yapının CSS1, CSS2, ve son sürümü olan CSS3 yayınlanmıştır.

CSS 3.0 Seçiciler

Html sayfası üzerine yazmış olduğumuz etiketlerin özelliklerinden ulaşarak onlara css tanımlayabilmekteyiz. CSS özelliklerini tanımlayabilmemiz için etiketlere ulaşabilmemizin birden fazla farklı yolu mevcuttur.

Tip Seçiciler

HTML etiket seçiciler, html içerisinde tanımlı olan etiketlerin görsel özelliklerini değiştirmek ya da belirtilen elemanlara yeni özellik vermek için kullanılır.

Html içerisindeki tanımlı elemanlara isimleri kullanılarak ulaşılabilir.

Örnek

```
<head>
  <style type="text/css">
    p {
      color: red;
      font-family: Verdana;
      font-size: 9pt;
    }
  </style>
</head>
<body>
  <div><p> Bilge Adam </p></div>
</body>
```

Evrensel Seçiciler

Evrensel seçiciler <body> etiketi içerisindeki bütün etiketlere aynı özelliği vermek için kullanılır.

Örnek

```
<head>
  <style type="text/css">
    * {
      color: red;
      font-family: Verdana;
      font-size: 9pt;
    }
  </style>
</head>
<body>
  <div>
    <h1> Bilge Adam </h1>
    <p> Yazılım </p>
  </div>
</body>
```

ID Seçiciler

Sayfanızdaki elemanın ID özelliğinden ulaşabilmenizi sağlayan seçicidir. Html yapısında ID özelliği her bir eleman için bir defa kullanılmalıdır. ID benzersiz isim olarak elemana atanır.

Örnek

```
<head>
  <style type="text/css">
    #box {
      width: 150px;
      height: 150px;
      background-color: yellow;
    }
  </style>
</head> <body>
  <div id="box">
    <h1> Bilge Adam </h1>
  </div>
</body>
```

CLASS Seçiciler

Bu seçicileri kullanarak bir HTML elemanına birden fazla class ekleyebilirsiniz.

Örnek

```
<head>
  <style type="text/css">
    .Box {
      width: 150px;
      height: 150px;
      background-color: yellow;
    }
    .font {
      color: red;
      font-family: Verdana;
      font-size: 9pt;
    }
  </style>
</head> <body>
  <div class="Box font">    <h1>Bilge Adam </h1>    </div>
</body>
```

SOY Seçiciler

Html elemanlarının oluşturduğu yapının içerisinde bulunan elemanlara ulaşmak için kullanılan yapıdır.

Örnek

```
<head>
  <style type="text/css">
    table {
      border: solid 1px #000;
    }

    table tr {
      padding: 5px;
    }

    table tr td {
      background-color: yellow;
      font-family: 'Agency FB';
      font-size: 9pt;
    }
  </style>
</head>
<body>
  <div>
    <table>
      <tr>
        <td>Bilge Adam </td>
      </tr>
    </table>
  </div>
</body>
```

JAVASCRIPT

Doksanlı yıllarda web sayfalarının dinamikleştirme ve statik bir web sayfasına daha fazla iş yaptırma doğrultusunda 1995 yılında Netscape firması tarafından geliştirilmiş bir script dilidir. Client yani istemci taraflı çalışan betik bir dildir. Peki betik Nedir? Web sayfalarında dinamik içerik sağlamak ve kullanıcıyla iletişim kurmak için kullanılan, istemci tarafında çalışan bir dildir. Html etiketleri arasında tanımlanarak kullanılabilir. Kısacası istemcinin web tarayıcısı tarafından yorumlanmış bir HTML sayfası üzerinde ki script dilini yorumlanabilmesidir.

Javascript'in Özellikleri

- Etkileşimli ve Dinamik web sayfaları hazırlayabilir. Css dosyasını yönetebilir.
- Client side olarak çalışır.
- Javascript bilgisayardaki dosyalara müdahale edemez
- Html sayfalarının bir parçasıdır tek başlarına bir anlam ifade etmezler ve çalışamazlar.
- Yorumlanabilmesi için tarayıcıya ihtiyaç duyarlar
- Javascript ile sunucuya dosya transferi yapılamaz
- Javascript kodları <script> </script> tagleri arasına yazılır.
- Harici bir dosya olarak HTML sayfasına gömülebilir.
- Herhangi bir text editör ile yazılabilir.
- Csharp koduna değer gönderebilir.

Javascript Kodlarına Giriş

Javascript kodlarını üç yöntem ile yazabilirim. Birinci yöntem; bir javascript dosyası (.js uzantılı) oluşturup kodlarımı yazar ve bu sayfayı HTML sayfamın içerisine gömebilirim.

İkinci yöntem; HTML sayfamın içerisinde script etiketleri açarak javascript kodlarımı bu etiketler arasına yazabilirim. Üçüncü yöntem ise; işlem yapacağım yere doğrudan kodlarımı yazabilirim.

Javascript browser'a komut gönderen bir script dilidir. Javascript'de satır sonu ; ile belirlenir. Ama kullanılmak zorunda değildir. Javascript case sensitive bir dildir. Büyük küçük harf duyarlılığına Csharp'dan daha fazla dikkat edilmelidir. Javascriptde hata kontrolü yapmak zordur. Kod hataları mesaj kutusu ile test edilir. Javascript'de boşluk karakterleri okunmaz. Javascript dosyalarının bir sıkıştırılmış halleri vardır birde normal halleri bulunmaktadır.

Veri Türleri

JavaScript içinde, üç birincil(Primary) veri türü, iki bileşik(Composite) veri türü ve iki özel(Special) veri türü vardır.

Birincil(Primary) Veri Türleri

- String
- Number
- Boolean

Bileşik(Composite) Veri Türleri

- Object
- Array

Özel(Special) Veri Türleri

- Null
- Undefined

String Veri Türü

String veri türü bir veya birden fazla unicode karakterlerden oluşmuş bir dizidir. JS içerisinde metinsel veri tiplerini ifade etmek için kullanılır aynı zamanda yine JS altında tek bir karakteri temsil edecek tür yoktur ihtiyaç duymamız halinde ise içerisinde tek bir karakter barındırabilen string ifade kullanırız.

Int Veri Türü

JS altında int veri tipi tamsayılar ve double yani ondalıklı değerler tek bir veri tipi içerisinde tutulur. Ondalıklı değerlerin yanı sıra 16lık(hexadecimal), 8lik(octal) sayı tipleride bulundurulabilir.

Boolean Veri Türü

String ve int veri tipleri neredeyse sınırsız sayıda farklı değerler alabiliyorken boolean veri türü yalnızca iki değer alır. True ve false değerleridir. Boolean bir gerçeklik değeridir ve koşulun doğru olup olmadığını belirtir.

NULL Veri Türü

Null veri türü js içerisinde yine null değer alabilir.Null bir değişken ya da işlev adı olarak kullanılamaz.Null içeren bir değişken sayısal,metinsel boolean ,array veya object içermez. Herhangi bir değişkenin içeriğini silmeden üzerine null değer atayarak silme işlemini gerçekleştirebilirsiniz.

Event'lar

Programlama dünyasında geçerli olan eventların bir kısmı JS içerisinde de geçerli.

OnChange Event'i

Sayfa üzerinde ki textbox içerisine bir ifade yazıldığında tetiklenir.

Html Kodu

```
<div id="Ornek1">
  <h5>1.Event: onchange</h5>
  <div>Javascripte hosgeldiniz.</div>
  <input type="text" onchange="Yaz()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
  function Yaz()
  {
    alert("<h1>Sayfa uzerinde ki text'e metin giriliyor.</h1>");
  }
</script>
```

OnClick Event'i

Sayfa üzerinde ki button tıklandığında tetiklenen eventdır.

Html Kodu

```
<div id="Ornek1">
  <h5>2.Event: onclick</h5>
  <div>Javascripte hosgeldiniz.</div>
  <input type="button" name="btnTikla" value="Tikla"onclick="Click()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
```



```
function Click()
{
    alert("<h1>Hello JS !! </h1>");
}
</script>
```

OnMouseOver Event'i

Sayfa üzerinde ki button ya da ona benzer kontrollerin üzerine mouse'un imleci ile üzerine geldiğimizde tetiklenir.

Html Kodu

```
<div id="Ornek1">
<h5>3.Event: onmouseover</h5>
<div>Javascripte hosgeldiniz.</div>
<input type="button" name="btnTikla" value="Tikla"onmouseover="Click()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
function Click()
{
    alert("<h1>Imlec su an buttonun uzerinde !! </h1>");
}
</script>
```

OnMouseOut Event'i

Sayfa üzerinde ki button ya da ona benzer kontrollerin üzerinden mouse'un imlecini kullanarak geri çekildiğimiz de tetiklenen event'dir.

Html Kodu

```
<div id="Ornek1">
<h5>4.Event: onmouseout</h5>
<div>Javascripte hosgeldiniz.</div>
<input type="button" name="btnTikla" value="Tikla"onmouseout="Click()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
    function Click()
    {
        alert("<h1>İmlec su an buttonun üzerinden ayrıldı !</h1>");
    }
</script>
```

OnKeyDown Event'i

Sayfa aktif bir halde browser üzerinde açıkken kullanıcının bir tuşa basması olayıdır.

Html Kodu

```
<div id="Ornek1">
<h5>5.Event: onkeydown</h5>
<div>Javascripte hosgeldiniz.</div>
<input type="text" onKeyDown="Down()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
    function Down()
    {
        alert("<h1>klavyeyi kullanıyorsun !</h1>");
    }
</script>
```

OnLoad Event'i

Tarayıcı sayfayı load etmeyi tamamladığında tetiklenen event'dır.

Html Kodu

```
<div id="Ornek1">
<h5>6.Event: onload</h5>
<div>Javascripte hosgeldiniz.</div>
<input type="text" onKeyDown="Load()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
    function Load()
    {
        alert("<h1>Sayfa tamaiyle load oldu</h1>");
    }
</script>
```

Javascript'de Hata Yönetimi

Javascript içerisinde hata yönetimi c# da kullandığımız gibi Try-Catch blokları sayesinde yakalanabilir. Try bloğunun içerisine test etmek yada , çalıştırıldığında hata yapılabilecek olan kodu,catch bloğunun içerisine hataya düğüldüğünde kontrol altına almak ya da hata karşılığında ekran gösterilecek olan uyarıyı hazırlamıza,throw kısmında ise istersek geliştirdiğimiz uygulamaya özgü hatalar ortaya çıkartmamıza son olarak finally bloğunda ise kodumuz başarılı bir şekilde çalıştırılısada-çalıştırılmasada hata alsak da almasakda ekrana gösterilecek ifadeleri hazırlamamıza olanak tanır. Aşağıda ki örnekte en basit hali ile Try-Catch bloklarının kullanımını görmekteyiz.

Javascript Kodu

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

<p>5 ile 10 arasında bir sayı giriniz:</p>

<input id="demo" type="text">
<button type="button" onclick="myFunction()">Test</button>
<p id="message"></p>

<script type="text/javascript">
function myFunction() {
    var message, x;
    message = document.getElementById("message");
    message.innerHTML = "";
    x = document.getElementById("demo").value;
    try {
        if(x == "") throw "degeri bos biraktiniz";
        if(isNaN(x)) throw "girilen deger sayi degil";
        x = Number(x);
        if(x < 5) throw "5 den kucuk bir deger girdiniz";
        if(x > 10) throw "10 dan buyuk bir deger girdiniz";
    }
    catch(err) {
        message.innerHTML = "Input is " + err;
    }
    finally{
        message.innerHTML = "Islem basarili ";
    }
}
</script>

</body>
</html>
```

Document.Write

Document Write komutu ekrana yazı yazdırmak için kullanılır. Document bizim HTML dosyamızı temsil eder. Csharpda ki this gibidir.

Document.Write Örneği

```
Document.write("<h1>Bu bir basliktir.</h1>");
```

Document.Write dikkat edilmesi gereken bir metotdur. Bu kodu bir fonksiyonelliğe bağlarsam (butona), fonksiyon çalıştığında Document.Write tüm sayfayı siler ve içerisindeki veriyi sayfaya yazar.

Fonksiyonlar javascript'de tetiklenmeden çağırılmayan yapılardır.

Html Kodu

```
<div id="Ornek1">
  <h5>1.Ornek: Document.Write</h5>
  <div>Javascripte hosgeldiniz.</div>
  <input type="button" name="btnTikla" value="Tikla" onclick="Yaz()" />
</div>
```

Javascript Kodu

```
<script type="text/javascript">
  function Yaz()
  {
    document.write("<h1>Bu bir basliktir.</h1>");
  }
</script>
```

Alert

C Sharp'da ki MessageBox penceresi gibidir. Tarayıcı tarafından tanımlıdır. Her tarayıcıda farklı görünüm sunar. Kullanıcıya mesaj vermek için kullanılabilir.

Mesaj Kutusu

```
function mesajKutusu()
{
  alert("Mesaj kutusu");
}
```

Html Kodu

```
<div id="Ornek2">
  <h5>2.Ornek: Alert-Confirm</h5>
  <input type="button" name="btnTikla2" value="Beni Tikla!" onclick="mesajKutusu()" />
</div>
```

</div>

Kontrolün içerisinde Alert Kullanımı

<input type="button" name="btnTikla2" value="Beni Tikla!" onclick="return alert('Mesaj Kutusu')" />

Mesaj Kutusu-Dialog Result

```
function mesajKutusu()
{
    var result = confirm("evet hayir mesaj kutusu");
    if (result)
    {
        alert("tamama basildi");
    }
    else
    {
        alert("iptale basildi")
    }
}
```

Html Kodu

```
<div id="Ornek2">
<h5>2.Ornek: Alert-Confirm</h5>
<input type="button" name="btnTikla2" value="Beni Tikla!" onclick="mesajKutusu()" />
</div>
```

Kontrolün içerisinde Dialog Result Kullanımı

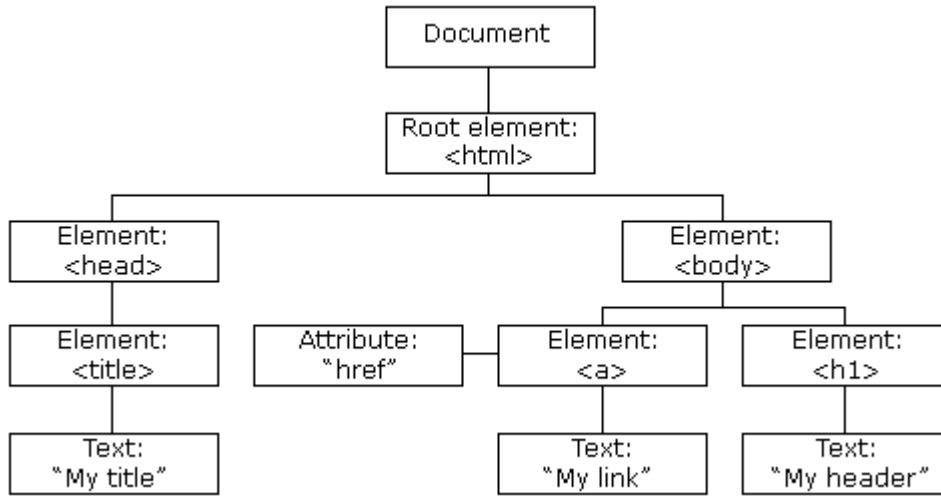
<input type="button" name="btnTikla2" value="Beni Tikla!" onclick="confirm('Evet Hayir Mesaj Kutusu')" />

Yukarıdaki gösterim c sharp tarafında metot tetikleteceğimiz zaman kullanılır.

Html'de çift tırnak ve tek tırnak string ifadeleri temsil eder. Eğer ifadeye çift tırnak ile başlamışsak içeride kullanacağımız string'i tek tırnak olarak kullanmalıyız.

Javascript'de DOM (Document Object Model)

Javascripte ilk giriş yaptığımızda dediğimiz gibi betik bir dildir yani son kullanıcının web tarayıcısında yorumlanabilen web sayfasına iş yaptırabilen yetenekli bir dil. HTML de ise önyüzü düzenler ve kullanıcıya sunarınız ama kullandığımız javascripte HTML kontrolleri tarafından bilgi aktarmak istediğimizde ya da arka planda javascript çalıştıktan sonra HTML ile düzenlenilmiş bir sayfa üzerinde herhangi bir ifade göstermek istediğimizde Document Object Model kullanırız.En basit hali ile DOM aşağıda ki resimde tam olarak gösterildiği gibidir.



GetElementById

Javascript ile herhangi bir elementin içeriğini değiştirebilirsiniz.

HTML Kodu

```
<div id="Ornek3">
  <h5>3.Ornek: getElementById</h5>
  <p id="paragraf">Javascript ile herhangi bir elementin icerigini degistirebilirsiniz.</p>
  <input type="button" name="btn" id="btnDegistir" value="Paragrafi Degistir" onclick="IcerigiDegistir()" />
</div>
```

Javascript Kodu

```
var yeniHalMi = false;
function IcerigiDegistir()
{
  var x = document.getElementById("paragraf");
  x.innerHTML = "Bu deęişiklik javascript tarafından yapıldı.";
}
```

NOT: Javascript'de Global Alan Fonksiyonların Dışındır.

Aşağıdaki örnekte javascript'in son versiyonunda çıkan, id ye göre element yakalama özelliğini kullanarak HTML etiketimizi yakalayabiliriz. Daha kısa bir yöntemdir.

Javascript'de etiketin id'sini yazdıktan sonra . dediğimizde javascript tarafında tanımlı fonksiyonlar gelecektir.

HTML Kodu

```
<div id="Ornek3">
```

```
<h5>3.Ornek: getElementById</h5>
<p id="paragraf">Javascript ile herhangi bir elementin icerigini degistirebilirsiniz.</p>
<input type="button" name="btn" id="btnDegistir" value="Paragrafi Degistir" onclick="IcerigiDegistir()" />
</div>
```

Javascript Kodu

```
var yeniHalMi = false; function IcerigiDegistir()
{
    if (yeniHalMi)
    {
        btnDegistir.value = "Paragrafi Degistir";
        paragraf.innerHTML = "Javascript ile herhangi bir elementin icerigini degistirebilirsiniz.";
        yeniHalMi = false;
    }
    else
    {
        btnDegistir.value = "Ilk Hale Don"; paragraf.innerHTML = "bu degisiklik javascript tarafından yapildi"; yeniHalMi = true;
    }
}
```

Element Attribute'u Değiştirme

Javascript ile bir element'e ait attribute değerini değiştirebiliriz. Aşağıdaki örnekte bir resmin src attribute'unun değerini değiştireceğiz.

HTML Kodu

```
<div id="Ornek4">
<h5>4.Ornek: resmin source'unu degistirme(bir elementin attribute'unu degistirme)</h5>

<p id="cop">Cop kutusunu bosaltmak icin resme tiklayiniz.</p>
</div>
```

Javascript Kodu

```
var doluMu = true;
function ResmiDegistir()
{
    if (doluMu)
    { // cop.style.backgroundColor = "red";
        var result = confirm('Cop kutusunu bosaltmak istediginizden emin misiniz?');
        if (result)
        {
            copKutusu.src = "images/bos_cop_kutusu.png";
            copKutusu.title = "Bos cop kutusu";
        }
    }
}
```

```
        cop.innerHTML = "Cop kutusu temizlendi.";
        copKutusu.style.cursor = "default";
        doluMu = false;
    }
}
}
```

Style Değiştirme

Javascript ile bir html elementinin style'ı nı değiştirebiliriz. Aşağıdaki örnekte mouse yazının (p etiketinin) üzerine gelince yazı rengi kırmızı olarak değişecektir.

HTML Kodu

```
<div id="Ornek5">
<h5>5.Ornek: style degistirme</h5>
<p id="renk" onmouseover="RenkDegistir('#ff0000','pointer')" onmouseout="RenkDegistir('black','default')">Javascript ile bir
elementin style bilgisini degistirebilirsiniz.</p>
</div>
```

Javascript Kodu

```
function RenkDegistir(yaziRengi, cursorSekli)
{
    renk.style.color = yaziRengi; renk.style.cursor = cursorSekli;
}
```

Validation Uygulama(Kontrol Etmek)

Verilerimizi javascript ile kullanıcı bilgisayarında kontrol edebiliriz. Eğer bilgiler doğru formatta değilse veya eksikse sayfanın post back olmasını engelleyerek sunucuya binecek yükü azaltabiliriz. Ama burada dikkat edilmesi gereken nokta javascript kullanıcının bilgisayarında çalışacağı için kullanıcı dilerse javascript çalışmasını engelleyerek bu güvenliği aşabilir.

Bu sebepten bilgilerimizi hem client hem server tarafında kontrol etmeliyiz. Client tarafında kontrol etmenin bize sağladığı faydalar;

- Kullanıcıya daha interaktif bir şekilde bilgi verme
- Kullanıcı Javascript kapatmamışsa gereksiz yere postback olmasını engellemek

HTML Kodu

```
<div id="Ornek6">
<h5>6.Ornek: Validation uygulama(kontrol etme)</h5>
<p>Lutfen bir sayi giriniz</p>
<input type="text" id="txtYazi" />
<input type="button" id="btnKontrol" value="Kontrol Et" onclick="KontrolEt()" />
```


</div>

Javascript Kodu

```
function KontrolEt() {  
    var deger = txtYazi.value;  
    if (deger == null || deger == "" || isNaN(deger)) {  
        //isNaN=> icerisine verilen degerin alfa numeric olup olmadigini kontrol eder  
        //isFinite ==> icerisine verilen degerin sayisal olup olmadigini kontrol eder  
        alert("Sayisal bir deger girmediniz.Lutfen sayisal bir deger giriniz");  
        txtYazi.value = "";  
    }  
}
```

Javascript'de Object Oriented Programming

Son yıllarda javascriptin iyice popüler olması ile beraber js kullanan geliştiricilerin sayısı ciddi oranda arttı. Beraberinde ise javascripti kullanmanın yanısıra javascripti object oriented olarak kullanmak işverenler tarafından sorgulanır oldu. Javascript içerisinde, c# dilinde ki gibi geniş kapsamlı bir object oriented olmasa bile ihtiyaçlarımızı karşılayacak kadar bir object oriented sistemi bulunmakta şimdi ona bir gözatalım.

Javascript İçerisinde Object Yaratmak

Asagida ki örnekte javascript içerisinde basit bir örnek nasıl yaratılıyor onu görmekteyiz.

Javascript Kodu

```
<script type="text/javascript">  
    var kisi = {  
        isim: ['Bob', 'Smith'],  
        yas: 32,  
        cinsiyet: 'male',  
        ilgiAlanlari: ['music', 'skiing'],  
        bio: function () {  
            alert(this.isim[0] + ' ' + this.isim[1] + ' is ' + this.yas + ' years old. He likes ' + this.ilgiAnlari[0] + ' and ' +  
this.ilgiAlanlari[1] + '.');  
        },  
        greeting: function () {  
            alert('Hi! I\'m ' + this.isim[0] + '.');  
        }  
    };  
</script>
```

Javascript Nesne Yönelimli Programlamada Kalıtımı Kullanmak (Inheritance)

C#, C++, Java gibi programlama dilleri içerisinde yani imperative dillerin içerisinde nesne yönelimli programlama temellerinin prensiplerini tam anlamıyla kullanabiliyoruz. Javascript içerisinde ise tam anlamıyla kullanılabilen ve

standardize edilmiş bir prensip var o da inheritance. Inheritance bildiğimiz gibi OOP içerisinde kalıtım almak anlamına gelir. JS içerisinde ise aşağıda ki gibi kullanılır.

Javascript Kodu

```
<script type="text/javascript">
  var ClassA = function () {
    this.name = "Bilge Adam";
  }

  var ClassB = function () {
    this.name = "bilgeadam";
    this.surname = "ben türeyen classim";
  }

  inheritsFrom(ClassB, ClassA);
</script>
```

Local ve Session Storage

Local Storage

Local storage ile birlikte web uygulamaları son kullanıcının tarayıcısının içerisinde verileri saklayabilir. HTML5'ten önce web uygulamaları cookie içerisinde saklıyordu. Local storage ise daha güvenli ve web uygulamasının performansına fazladan bir yük bindirmeden büyük miktarlarda veriler saklayabiliyor.

Javascript Kodu

```
<script type="text/javascript">
  localStorage.setItem("numItems", 1);
  var numItems = localStorage.getItem("numItems");
  var numItems = parseInt(localStorage.getItem("numItems"));
  numItems = numItems + 1;
  localStorage.setItem("numItems", numItems);
  localStorage.setItem("price", 9.99);
  var price = parseFloat(localStorage.getItem("price"));
</script>
```

Session Storage

Session storage ile local storage arasında aslında temelde bir fark yoktur fakat session storage içerisinde saklanılmasını istediğimiz bilgiler sadece bir sessionluk tutulur. Bu demektir ki son kullanıcı web tarayıcısını kapattığında tarayıcıyı kapatmasa bile web uygulamasına ait sekmeyi kapatırsa tüm bilgiler gidecek anlamına gelir.

Javascript Kodu

```
<script type="text/javascript">
if (localStorage.clickcount) {
    localStorage.clickcount = Number(localStorage.clickcount) + 1;
} else {
    localStorage.clickcount = 1;
}
document.getElementById("sonuc").innerHTML = "butonu tiklama sayiniz " +
    localStorage.clickcount + " time(s).";
</script>
```

JQUERY

John Resing tarafından 2006 yılında geliştirilmeye başlanan JQuery açık kaynak kodlu bir Javascript kütüphanesidir. JQuery kütüphanesi 2013'den beri 16 kişi tarafından geliştirilmekte ve 2.0 versiyonu yayınlanmaktadır. Kütüphanenin Sıkıştırılmış boyutu 86kb'dır. Javascriptin özelleşmiş bir kütüphanesidir. Az kod çok iş mantığındadır.

Jquery'i ikiye ayırabiliriz.

- Selector: Html'de ki dom yapısını kullanarak html elementlerini yakalayabiliriz.

Jquery'i kullanabilmek için kütüphaneyi projeme referans olarak vermem gerekmektedir. Javascript'de referans sırası çok önemlidir.

CDN-Content Delivery Network

Javascript dosyalarını kendi bilgisayarımız yerine yüksek hızlı sunuculardanda çekebiliriz. Bunun bize sağladığı yararlar vardır. Örneğin amerikadan bir ziyaretçi web sitemizi ziyaret ettiğinde, jquery dosyasını bizim sunucumuzdan çekmek yerine google'ın yüksek hızlı serverlarından çekebilir.

Javascriptin Çalışma Mantığı

Javascript kodlarının çalışabilmesi için web sayfasının tamamen indirilmiş ve çalışmaya hazır olması gerekmektedir. Sayfa hazır olduğunda kodları hazır olarak bekletmek için \$ sembolü kullanılır. DOM yapısı üzerindeki bir html etiketini seçmek için \$ sembolü kullanılır.

\$(document).ready() içerisinde yazılacak fonksiyonların aktif hale gelmesi ve hazır olarak kullanılmayı beklemesi için kullanılır.

Javascript'de işlemler üçe ayrılır.

- Get => text(),html(),val(),attr()

- Set => text(),html(),val(),attr()
- Add => append(),addClass(),removeClass()

Jquery Sözdizimi (Syntax)

jQuery sizin kolayca kullanabileceğiniz bir yapıya sahiptir. Nesnenin herhangi bir özelliğini kullanarak seçim yapabilirsiniz. Bu seçtiğiniz nesne veya nesnelere eylemler verebilirsiniz.

Temel seçim şöyledir: **\$(seçici).eylem()**

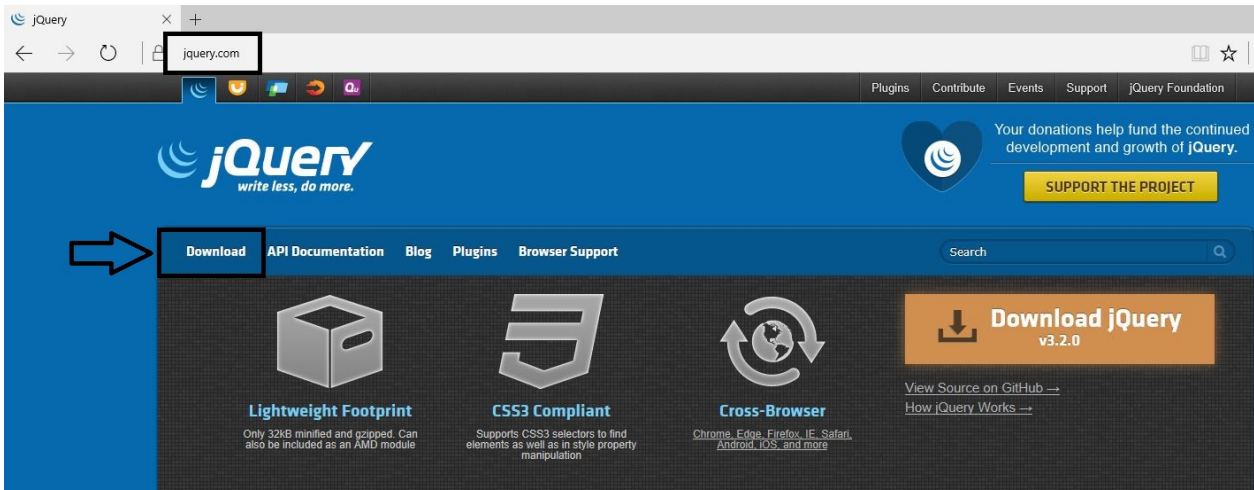
- \$(this).hide() - güncel nesneyi gizler.
- \$(p).hide() - tüm <p> nesnelerini gizler.
- \$(".test").hide() - CSS sınıf ismi (class'ı) "test" olan tüm nesneleri gizler.
- \$("#test").hide() - ID'si "test" olan nesneyi gizler.

Projede JQuery Kullanımı

Herhangi bir proje içerisinde jquery kullanmak istediğimiz noktada üç farklı şekilde kullanabiliriz. Bu yollar neler şimdi onlara bir gözatalım.

1.Yol;

www.jquery.com adresine girip ekran görüntüsünde ki download sekmesinden:



Download Sekmesine tıkladıktan sonra ise açılan sayfada:

Downloading jQuery using Bower

jQuery is also registered as a package with [Bower](#). You can install the latest version of jQuery with the command:

```
1 | bower install jquery
```

This will install jQuery to Bower's install directory, the default being `bower_components`. Within `bower_components/jquery/dist/` you will find an uncompressed release, a compressed release, and a map file.

The jQuery Bower package contains additional files besides the default distribution. In most cases you can ignore these files, however if you wish to download the default release on its own you can use Bower to install jQuery from one of the above urls instead of the registered package. For example, if you wish to install just the compressed jQuery file, you can install just that file with the following command:

```
1 | bower install https://code.jquery.com/jquery-3.2.0.min.js
```

Bower install ile başlayan satır üzerinde ki linki kopyalayıp aşağıda ki ekran görüntüsünde ki gibi projemizin içerisine yapııştırıp kullanabiliriz.

```

7      <meta name="viewport" content="width=device-width" />
8      <title>Login</title>
9      <style type="text/css">
10     table tr td
11     {
12         padding:5px;
13     }
14     table tr td .txtLog
15     {
16         width:200px;
17         height:20px;
18     }
19
20     #button
21     {
22         padding:5px;
23     }
24 </style>
25 <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
26 <script type="text/javascript">
27     function Kontrol()
28     {
29         var uyarı = "";
30         var kadi = $("#txtKadi").val();
31         kadi = $.trim(kadi);
32         var sifre = $("#txtSifre").val();
33         sifre = $.trim(sifre);

```

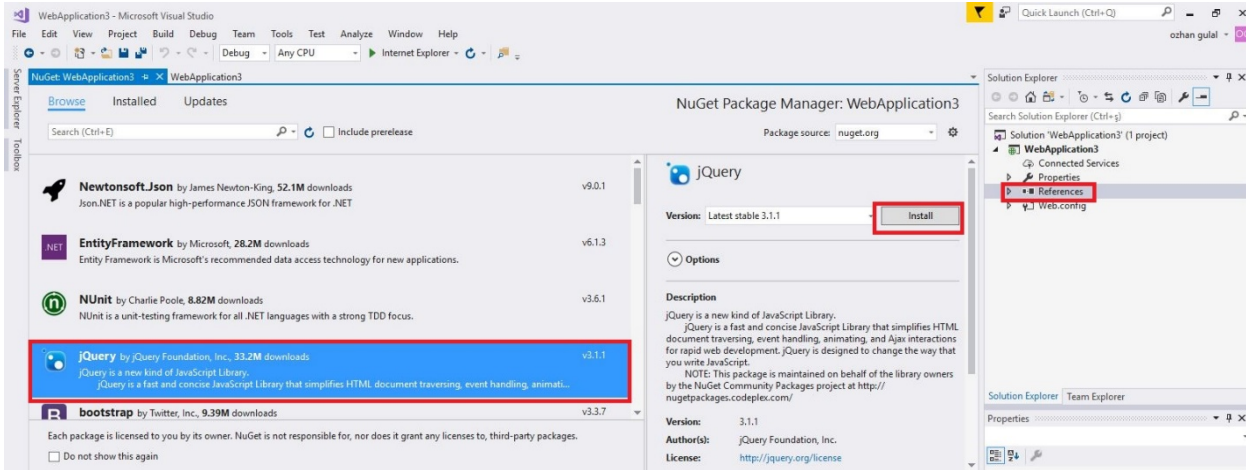
2.Yol;

www.jquery.com adresine girip ekran görüntüsünde ki download butonuna tıklayıp indirilen dosyayı projemizin içerisine sürükleyip bırakarak kullanabiliriz:



3.Yol;

Son yol ise en çok kullanılan yollardan biridir. Visual Studio içerisinde solution explorer penceresinde proje altında referanslara sağ tık yaptıktan sonra "Manage Nuget Packages" sekmesinden gelen pencerede doğrudan JQuery alanını tıklayıp "install" butonuna tıkladıktan sonra kurulumu başlatıyoruz bir kaç saniye içerisinde kurulum tamamlanmış oluyor ve bu sayede jquery artık uygulamamız içerisinde kullanılabilir durumda.



Jquery İçerisinde Hiyerarşik Yapı

Hiyerarşi, bir toplulukta veya bir kuruluştaki yer alan kişileri alt-üst ilişkileri, görev ve yetkilerine göre sınıflandıran sistemdir. Web Dünyasında ise HTML içerisinde özellikle bazı taglerin kullanılabilmesi başka bir tage bağlıdır. Mesela `<tr>` ve `<td>` tagini amacına uygun olarak kullanmak istersek en başa `<table>` tagini tanımlamak zorundayız sebebi ise `<tr>` ve `<td>` `<table>` tagine bağımlı bir tagdir ve o olmadan bir anlam ifade etmez.

Hiyerarşik yapı içerisinde 6 farklı öge mevcut:

- `.siblings()`
- `.children()`
- `.next()` ve `.nextAll()`
- `.prev()` ve `.prevAll()`
- `.closest()` ve `.parents()`
- `.find()`

.siblings()

`.siblings` methodu ile bir HTML yapısındaki istenilen elementin tagi ya da tagleri seçilebilir ve gerekli düzenlemeler yapılabilir.

HTML Kodu

```
<form id="form1" runat="server">
  <div>
<ul>
  <li>liste elemanı 1</li>
  <li>liste elemanı 2</li>
  <li>liste elemanı 3</li>
  <li>liste elemanı 4</li>
  <li>liste elemanı 5</li>
</ul>
  </div>
```

Javascript Kodu

```
<script type="text/html">
    $('li:nth-child(2)').siblings().css('color', 'mediumslateblue');
</script>
```

.children()

.children methodu ile bir elementin alt elemanlarını seçmeyi ve bu elemanları düzenlemeye yarar.

HTML Kodu

```
<form id="form1" runat="server">
<form id="form1" runat="server">
    <div>
        <ul>
            <li>liste elemanı 1</li>
            <li>liste elemanı 2</li>
            <li>liste elemanı 3</li>
            <li>liste elemanı 4</li>
            <li>liste elemanı 5
        </ul>
        <ul>
            <li>alt liste elemanı 1</li>
            <li>alt liste elemanı 2</li>
            <li>alt liste elemanı 3</li>
            <li>alt liste elemanı 4</li>
            <li>alt liste elemanı 5
        </ul>
        <ul>
            <li>liste elemanı 6</li>
            <li>liste elemanı 7</li>
        </ul>
    </div>
</form>
```

Javascript Kodu

```
<script type="text/javascript">
    $('li').children().css('color', 'mediumslateblue');
</script>
```

.next() ve nextAll()

.next() seçilen elemandan bir sonraki elementi, .nextAll() ise seçilen elemandan sonraki tüm elementleri seçmek ve düzenlemek için kullanılır.

HTML Kodu

```
<body>
    <form id="form1" runat="server">
        <div>
            <ul>
                <li>liste elemanı 1</li>
                <li>liste elemanı 2</li>
                <li class="deneme">liste elemanı 3</li>
                <li>liste elemanı 4</li>
                <li>liste elemanı 5
```

```
<ul>
  <li>alt liste elemanı 1</li>
  <li>alt liste elemanı 2</li>
  <li>alt liste elemanı 3</li>
  <li>alt liste elemanı 4</li>
  <li>alt liste elemanı 5</li>
</ul>
  <li>liste elemanı 6</li>
  <li>liste elemanı 7</li>
</ul>
</div>
</form>
</body>
```

Javascript Kodu

```
<script type="text/javascript">
  $('li.deneme').next().css('color', 'mediumslateblue');
</script>
```

HTML Kodu

```
<body>
<form id="form1" runat="server">
  <ul>
    <li>liste elemanı 1</li>
    <li>liste elemanı 2</li>
    <li class="deneme">liste elemanı 3</li>
    <li>liste elemanı 4</li>
    <li>liste elemanı 5
  </ul>
    <li>alt liste elemanı 1</li>
    <li>alt liste elemanı 2</li>
    <li>alt liste elemanı 3</li>
    <li>alt liste elemanı 4</li>
    <li>alt liste elemanı 5</li>
  </ul>
    <li>liste elemanı 6</li>
    <li>liste elemanı 7</li>
  </ul>
</form>
</body>
```


Javascript Kodu

```
<script type="text/javascript">
    $('li.deneme').nextAll().css('color', 'mediumslateblue');
</script>
```

.prev() ve prevAll()

.next() 'in tam tersidir. Seçilen elemandan önceki elemanı ya da elemanları seçer ve düzenler.

.closest() ve parents()

Closest kelime anlamı olarak *en yakın* anlamına gelmektedir. Closest, jQuery ile seçilen elemana en yakın olan elementi seçmeye ve düzenlemeye yarar. Parents ise seçilen eleman ile ilişkili tüm elemanları seçer ve düzenler.

HTML Kodu

```
<body>
<form id="form1" runat="server">
<ul>
    <li>liste elemanı 1</li>
    <li>liste elemanı 2</li>
    <li>liste elemanı 3</li>
    <li>liste elemanı 4</li>
    <li>liste elemanı 5
</ul>
    <li>alt liste elemanı 1</li>
    <li>alt liste elemanı 2</li>
    <li class="deneme">alt liste elemanı 3</li>
    <li>alt liste elemanı 4</li>
    <li>alt liste elemanı 5</li>
</ul>
    <li>liste elemanı 6</li>
    <li>liste elemanı 7</li>
</ul> </form>
</body>
```

Javascript Kodu

```
<script type="text/javascript">
    $('li.deneme').closest('ul').css('color', 'mediumslateblue');
</script>
```

HTML Kodu

```
<body>
<form id="form1" runat="server">
<ul>
    <li>liste elemanı 1</li>
```

```
<li>liste elemanı 2</li>
<li>liste elemanı 3</li>
<li>liste elemanı 4</li>
<li>liste elemanı 5
<ul>
  <li>alt liste elemanı 1</li>
  <li>alt liste elemanı 2</li>
  <li class="deneme">alt liste elemanı 3</li>
  <li>alt liste elemanı 4</li>
  <li>alt liste elemanı 5</li>
</ul>
  <li>liste elemanı 6</li>
  <li>liste elemanı 7</li>
</ul> </form>
</body>
```

Javascript Kodu

```
<script type="text/javascript">
  $('li.deneme').parents('ul').css('color', 'mediumslateblue');
</script>
```

.find()

Find methodu tam olarak adındanda anlaşılabilirliği gibi bulmak görevini üstlenmiştir. Aşağıda ki örnekte tam olarak find methodu tüm sırasız listeleri seçer, deneme sınıfına sahip elemanı bulur ve ardından rengini değiştirir.

HTML Kodu

```
<body>
<form id="form1" runat="server">
<ul>
  <li>liste elemanı 1</li>
  <li>liste elemanı 2</li>
  <li>liste elemanı 3</li>
  <li>liste elemanı 4</li>
  <li>liste elemanı 5
<ul>
  <li>alt liste elemanı 1</li>
  <li>alt liste elemanı 2</li>
  <li class="deneme">alt liste elemanı 3</li>
  <li>alt liste elemanı 4</li>
  <li>alt liste elemanı 5</li>
</ul>
  <li>liste elemanı 6</li>
```

```
<li>liste elemanı 7</li>
</ul> </form>
</body>
```

Javascript Kodu

```
<script type="text/javascript">
$('ul').find('.deneme').css('color', 'mediumslateblue');
</script>
```

Jquery içerisinde Css

Hazırlanmış olan bir uygulamanın içerisine eğer istersek css koduna müdahale edebiliriz. Aşağıda bunun örneğini görmekteyiz.

HTML Kodu

```
<input type="button" id="btnTikla" value="Beni Tikla" /> <hr />
<ul id="sehirler">
  <li>Istanbul</li>
  <li>Ankara</li>
  <li>Bursa</li>
  <li>Izmir</li>
  <li>Antalya</li>
</ul>
<hr />
<div id="paragraflar">
  <div>
    <p>paragraf 1</p>
  </div>
  <div>
    <p>paragraf 2</p>
  </div>
  <p>paragraf 3</p>
</div>
<hr />
```

```
<div class="divMetin">Burada metin kelimesi gecmektedir</div> <div class="divMetin">Burada o malum kelime
gecmemektedir</div> <div class="divMetin">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur euismod,
  mauris molestie egestas tincidunt, risus diam sodales quam, ac semper diam arcu sit amet urna. Nunc magna sem, convallis
  eget convallis sed, mollis non orci. Donec hendrerit velit id urna fermentum, sit amet adipiscing erat elementum. Mauris tristique
  dignissim orci, in auctor massa mattis sed. Quisque fringilla, lectus non porttitor feugiat, erat libero metin rhoncus nibh, nec
  sagittis ipsum urna sit amet odio. Sed adipiscing enim in quam placerat, ut sodales lacus fermentum. Ut commodo bibendum
  tellus, eu adipiscing nisi commodo a. Ut vitae lacus id leo blandit condimentum. Aenean a iaculis ante. Duis vitae dapibus justo.
  Aliquam vulputate vel nunc eget fermentum. Aliquam erat volutpat. Donec sapien leo, placerat in dolor at, tristique dapibus
  massa. Nunc id eros orci. Curabitur bibendum odio sed lacus posuere pellentesque. </div>
<hr />
<table id="city" cellpadding="2" cellspacing="3" border="1">
```

```
<thead>
  <tr>
    <th>Sehirler</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Istanbul</td>
  </tr>
  <tr>
    <td>Bursa</td>
  </tr>
  <tr>
    <td>Ankara</td>
  </tr>
</tbody>
</table>
```

Javascript Kodu

```
$(document).ready(function ()
{
  $("#btnTikla").click(function () {
    //id'si sehirler olan listenin elemanlarinin yazi renklerini degistirme
    $("#sehirler > li").css("color", "magenta");

    //id'si paragraflar olan div'in elemanlarından p olanının yazi rengini degistirme
    $("#paragraflar > p").css("color", "magenta");

    //idsi sehirler olan listenin elemanlarına çiftlik teklik durumlarına göre css atama
    $("#sehirler li:nth-child(2n+1)")
    $("#sehirler li:nth-child(2n)")
    $("#sehirler li:even").css("color", "red");
    $("#sehirler li:odd").css("color", "green");

    //icerisinde 'metin' kelimesi gecen divlerin stil bilgilerini degistirme
    // $(".divMetin:contains('metin)').css("background-color", "cyan");
    $(".divMetin:contains('metin)').addClass("sonradan");

    //idsi city olan tablonun çift indexli satirlarının borderlerini ve yazi rengini degistirme(addClass ile);
    //border: 3px solid red; renk=> blue
    $("#city tbody tr:even").addClass("tabloicin");
  });
});
```

Jquery Animasyonlar ve Olaylar

Nerelerde Kullanılıyor?

- Sayfamızın bir kısmında veya tamamında görsel değişiklikler yapabilir
- Html nesnelerine kolayca müdahale edebilir
- Animasyonlar yapılabilir
- Programlama dilleri ile yapılan sayfalardan veri çekebilir ve gönderebilir

Javascript Kodu

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta charset="utf-8" />
  <link href="MyStyle.css" rel="stylesheet" />
  <script src="Scripts/jquery-3.1.1.js"></script>
  <script type="text/javascript">
    $(Document).ready(function () {

      $("#rect1").hide(1000).show(1000);
      $("#rect2").slideUp(2000).slideDown(2000);
      $("#rect3").toggle(3000).toggle(3000);
      $("#rect4").slideToggle(4000).slideToggle(4000);
      $("#btn1").click(function () {

        $(".rect").css("background-color", "blue");
      })
      $("#Btn2").hover(function () {

        this.value = "üzerinde";
      })
      $("#Btn2").mouseout(function () {

        this.value = "dışında";
      })
      $("#Btn3").dblclick(function () {

        alert("HTML:" + $("#rect1").html());
        alert("HTML:" + $("#rect1").text());
      })

    })
  </script>
</head>
</html>
```

</script>

HTML Kodu

```
<body>
  <div id="header">
    <h1>Hello JQuery</h1>
    <input id="btn1" type="button" value="button1" />
    <input id="Btn2" type="button" value="button2" />
    <input id="Btn3" type="button" value="button3" />
    <input id="Btn4" type="button" value="button4" />
  </div>
  <div id="Panels">
    <div id="rect1" class="rect">Rect1</div>
    <div id="rect2" class="rect">Rect2</div>
    <div id="rect3" class="rect">Rect3</div>
    <div id="rect4" class="rect">Rect4</div>
  </div>
  <div class="clear"></div>
</body>
</html>
```

CSS Kodu

```
.rect
{
    background-color:#ff6a00;
    color:black;
    width:100px;
    height:100px;
    text-align:center;
    margin:10px;
    float:left;
}

#header
{
    height:150px;
    padding-top:20px;
    padding-left:400px;
    background-color:#ff6a00;
}

#Panels
{
    background-color:#808080;
    padding-left:300px;
    height:150px;
}

.clear
{
    clear:both;
}
```

BOOTSTRAP

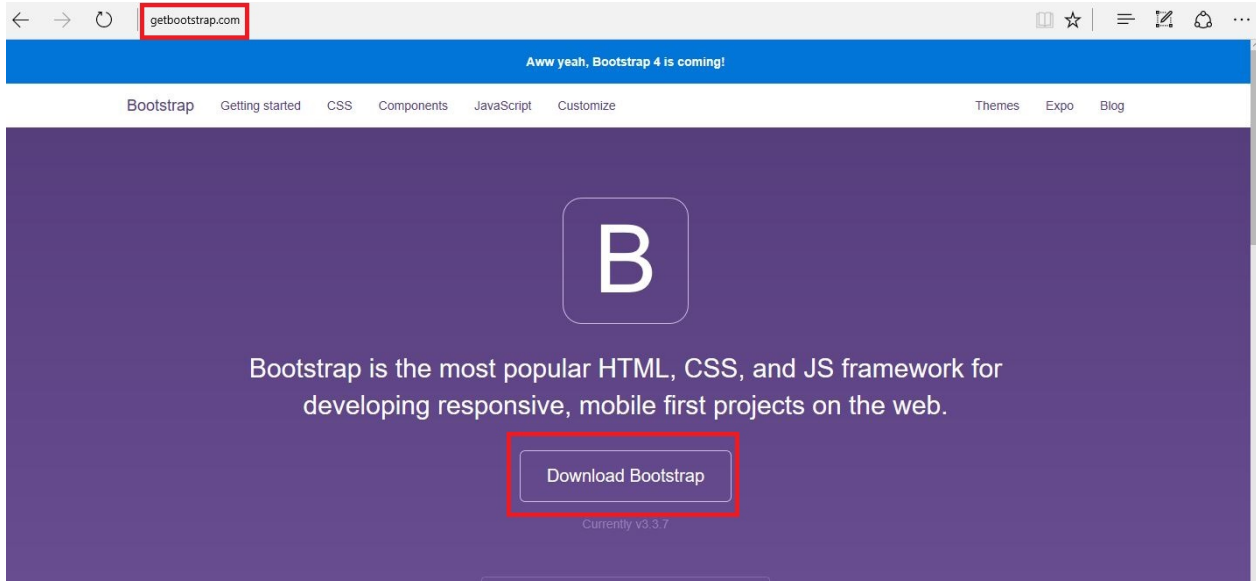
Bootstrap, twitter tarafından geliştirilen, açık kaynak kodlu, ücretsiz bir CSS framework'tür. Responsive davranır yani telefonlar, tabletler, dizüstü ve masaüstü bilgisayarlar için farklı ve cihazların çözünürlüklerine uygun görsel açıdan herhangi bir arayüz ögesi kaybetmeden temalar/tasarımlar geliştirebileceğimiz ortamdır. Bootstrap'in en büyük özelliği ise bir web uygulaması için arayüz geliştirirken aklımıza gelebilecek (form öğeleri, etiketler, tablolar, uyarı ve bilgi metinleri, navigasyon bar, sayfalandırma modülü, açılan menüler, farklı özellikte butonlar) bir çok tasarım ögesini içerisinde barındırır. Kısacası her şeyi hazır olan kodlarla yeni bir web sayfası ortaya koymak çok basit ve pratiktir. Geliştirme esnasında ihtiyaç duyacağımız tüm .js, .css dosyaları daha önceden sistemin içerisine implemente edilmiştir. Bize tek kalan bunları çağırmak ve doğru yerde kullanmaktır ayrıca bootstrap konusunda ki en güzel kısım ise tüm tarayıcılar tarafından desteklenmesi ve tarayıcı değişikliği gerçekleştiğinde davranış değişikliğinin olmamasıdır.

Bootstrap Nasıl Kurulur ?

Herhangi bir proje içerisinde Bootstrap kullanmak istediğimiz noktada iki farklı şekilde kullanabiliriz. Bu yollar neler şimdi onlara bir gözatalım.

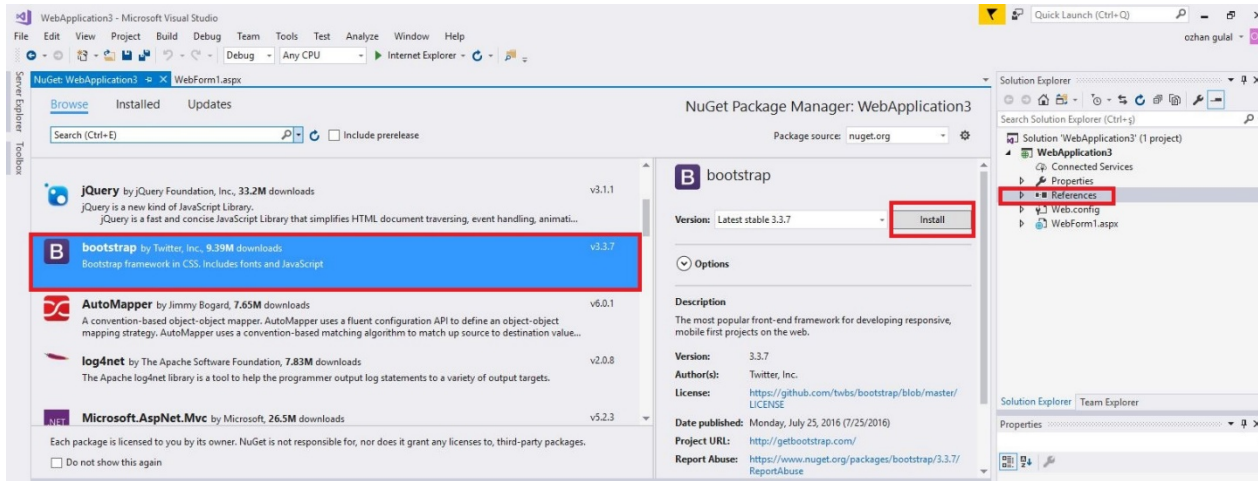
1.Yol;

www.getbootstrap.com adresine gidip resimde ki gibi download butonuna tıkladıktan sonra inen dosyayı projemize içersine alıp kullanabiliriz.



2.Yol;

2.Yol ise yine jqueryi uygulamamızın içersine kurduğumuz gibi manage nuget packages den faydalanarak kurmak aynı yolu tekrar izliyoruz yani projemizi başlattıktan sonra solution explorer üzerinde sağ tık yapıyoruz references sekmesinden manage nuget packagesi seçiyoruz ve açılan pencerede bootstrap i kurup uygulamamızın içersine almış oluyoruz.



Bootstrap nasıl kullanılır?

Bootstrapı kullanabilmemiz için en başta sayfamız html5 olmalı ve bootstrap için ana css dosyası ve jquery scriptinide sayfamıza eklemelisiniz. Sayfamızda kullanacağımız her bir öge için getbootstrap.com

sitesi referansımız olacak.

[Bootstrap](#)[Getting started](#)[CSS](#)[Components](#)[JavaScript](#)[Customize](#)

Siteye girdikten sonra yukarıda ki üst menuden getting started sekmesine tıklayıp ihtiyacımız olan dosyaları download ediyoruz, projemize implemente ettikten sonra ise CSS, Components, JavaScript alanlarına girip ihtiyacımız olan kodları buralardan kopyala yapıştır ile alacağız

Bootstrap Containers

Bootstrap site içeriğinizi doğru bir şekilde göstermek için kendi mevcut grid sistemini kullanır. Grid sistemini kullanmak için ise 2 farklı tipte Container e ihtiyaç duyarız:

.container: responsive için statik genişliktedir.

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
<div class="container">
  ....
</div>
</body>
</html>
```

.container-fluid: ekranınızda son kullanıcı tarafından görünen her yeri kapsar.

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
<div class="container-fluid">
  ....
</div>
</body>
</html>
```

Bootstrap Media Tag

Bootstrap, sayfamız içerisinde kullanılacak olan media nesnelerini hizalayabilmemiz için kolay bir yol sunar. Genelde ise blog sayfaları ya da kısa metinler ve bu metinler içerisinde resim ya da video yerleştirmek istediğimizde kullanılabilir.

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
<!-- sol tarafa hizali -->
<div class="media">
  <div class="media-left">
    
  </div>
  <div class="media-body">
    <h4 class="media-heading">John Doe</h4>
    <p>Lorem ipsum...</p>
  </div>
</div>

<!-- sag tarafa hizali -->
<div class="media">
  <div class="media-body">
    <h4 class="media-heading">John Doe</h4>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
  </div>
  <div class="media-right">
    
  </div>
</div></body>
</html>
```

Bootstrap Grid(Izgara Sistemi)

Bootstrap grid sistemi mevcut sayfamızı 12 sayfaya kadar bölmeye olanak tanır. 12 alanın tamamını kullanmak istemiyorsak düzenli olarak grüplayarak eş parçalar ya da tek bir bütün halinde kullanabiliriz.

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 Sınıfları

Bootstrap ayrıca web uygulamamızın tam anlamıyla kullanışlı olabilmesi için grid açısından 4 farklı class'a sahip

- Xs: Cep telefonları için
- Sm: Tablet bilgisayarlar için
- Md: desktop için
- Lg: büyük ölçekli desktoplar için

Aşağıda ise en temel hali ile gridin kullanımını görüyoruz:

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
  <div class="row">
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
</div></body>
</html>
```

Bu örneğimizde ise sayfaya 3 eşit alana bölen bir grid örneği görüyoruz.

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
  <div class="row">
    <div class="col-sm-4">.col-sm-4</div>
    <div class="col-sm-4">.col-sm-4</div>
    <div class="col-sm-4">.col-sm-4</div>
  </div></body>
</html>
```

.col-sm-4	.col-sm-4	.col-sm-4
-----------	-----------	-----------

Bootstrap içerisinde Input ve Button Kullanımları

Bootstrap altında aklınıza gelebilecek tüm kontrolleri responsive yani cihazlarda çözünürlük problemi olmadan kullanabilirsiniz aşağıda ki örnekte ise button ve text üzerinde ki kullanılış şeklini görmekteyiz.

Öğrenci Ekleme Formu

Ad giriniz
Soyad giriniz
Yaş giriniz
Sınıf giriniz

Ekle

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>YeniOlustur</title>
  <link href="/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
  <div class="container">
    <div class="row">
      <h2>Öğrenci Ekleme Formu</h2>
    </div>
    <div class="col-lg-6 col-sm-12">
      <form role="form" action="/Ogrenci/Ekle" method="post">
        <input name="Ad" class="form-control" placeholder="Ad giriniz" />
        <input name="Soyad" class="form-control" placeholder="Soyad giriniz" />
        <input name="Yas" class="form-control" placeholder="Yaş giriniz" />
        <input name="Sinif" class="form-control" placeholder="Sınıf giriniz" />
        <div class="text-right">
          <button type="submit" class="btn btn-info">Ekle</button>
        </div>
      </form>
    </div>
  </div>
</body>
</html>
```

Öğrenci Listesi

[Yeni öğrenci ekle](#)

İsim	Soyisim	Yaş	Sınıf		
Ahmet	Demir	18	Yazılım	Sil	Düzenle
Gokhan	Karacaer	25	Yazılım	Sil	Düzenle
Seydi	Işın	28	Yazılım	Sil	Düzenle

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
  <div class="container" style="margin-top:40px;">
    <div>
      <h2>Öğrenci Listesi</h2>
    </div>
    <div class="row">
      <a class="btn btn-success" href="/Ogrenci/YeniOlustur">Yeni öğrenci ekle</a>
    </div>
    <table class="table table-hover">
      <thead>
        <tr>
          <th>İsim</th>
          <th>Soyisim</th>
          <th>Yaş</th>
          <th>Sınıf</th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        @foreach (var item in Model)
        {
          <tr>
            <td>@item.Ad</td>
            <td>@item.Soyad</td>
            <td>@item.Yas</td>
            <td>@item.Sinif</td>
            <td>
              <a class="btn btn-danger" href="/Ogrenci/Sil/@item.ID">Sil</a>
              <a class="btn btn-warning" href="/Ogrenci/Duzenle/@item.ID">Düzenle</a>
            </td>
          </tr>
        }
      </tbody>
    </table>
  </div>
```

```
</body>
</html>
```

Bootstrap içerisinde İkon,Navigasyon ve Uyarılar

İkonlar

Bootstrapda ciddi anlamda web uygulamamızın daha şık ve kullanışlı gözükmesini sağlayacak iconlar mevcut. Aşağıda en basit haliyle bunların kullanımlarını ve çıktılarını görmekteyiz.

HTML Kodu

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Liste</title>
  <link href="/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
  <div class="container" style="margin-top:40px;">
    <div>
      <h2>Öğrenci Listesi</h2>

      <p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
      <p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
      <p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
    </div>
  </body>
</html>
```



glyphicon glyphicon-envelope



glyphicon glyphicon-search



glyphicon glyphicon-print

Navigation Bar

Bootstrap bizlere ciddi anlamda kullanışlı ve her anlamda esnek bir de navigation bar sunmakta kullanımı ise:

WebSiteName Home Page 1 Page 2 Page 3

HTML Kodu

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Page 1</a></li>
      <li><a href="#">Page 2</a></li>
      <li><a href="#">Page 3</a></li>
    </ul>
  </div>
</nav>
```

Uyarılar

Son kullanıcılar nötr,pozitif,negatif ya da bilgi verilmek istendiğinde kullanılan durumlardır. İçerisinde bulunan duruma göre renk açısından değişiklik gösterebilir.

Success! Başarılı ya da olumlu bir durum geliştiğinde kullanılan uyarı şekli.

Info! Son kullanıcıya bilgi verilmek istendiğinde kullanılan uyarı şekli.

Warning! Son kullanıcı uyarılmak istendiğinde kullanılan uyarı şekli.

Danger! Tehlikeli ya da olumsuz bir durum geliştiğinde kullanılan uyarı şekli.

HTML Kodu

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="WebApplication3.WebForm2"%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <div class="alert alert-success">
                <strong>Success!</strong> Başarılı ya da olumlu bir durum geliştiğinde kullanılan uyarı şekli.
            </div>

            <div class="alert alert-info">
                <strong>Info!</strong> Son kullanıcıya bilgi verilmek istendiğinde kullanılan uyarı şekli.
            </div>

            <div class="alert alert-warning">
                <strong>Warning!</strong> Son kullanıcı uyarılmak istendiğinde kullanılan uyarı şekli.
            </div>

            <div class="alert alert-danger">
                <strong>Danger!</strong> Tehlikeli ya da olumsuz bir durum geliştiğinde kullanılan kullanılan uyarı şekli.
            </div>
        </div>
    </form>
</body>
</html>
```

jQuery ile Bootstrap Beraber Kullanımı

Bir web uygulaması geliştirirken tüm ihtiyaçlarımızı giderebileceğimiz 2 farklı frameworkü yani jquery ve bootstrapi nasıl sentezleriz ona bir göz atalım şimdide. Aşağıda ki örnekte jquery ve bootstrapden fayadalanılarak bir dropdownlist hazırladık.

HTML Kodu

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
  <h2>Dropdowns</h2>
  <p>jQuery den faydalanarak dropdownlist kullanımı:</p>
  <div class="dropdown">
    <button class="btn btn-default dropdown-toggle" type="button" id="menu1" data-toggle="dropdown">Tutorials
    <span class="caret"></span></button>
    <ul class="dropdown-menu" role="menu" aria-labelledby="menu1">
      <li role="presentation" class="dropdown-header">Dropdown basligi 1</li>
      <li role="presentation"><a role="menuitem" tabindex="-1" href="#">HTML</a></li>
      <li role="presentation"><a role="menuitem" tabindex="-1" href="#">CSS</a></li>
      <li role="presentation"><a role="menuitem" tabindex="-1" href="#">JavaScript</a></li>
      <li role="presentation" class="divider"></li>
      <li role="presentation" class="dropdown-header">Dropdown basligi 2</li>
      <li role="presentation"><a role="menuitem" tabindex="-1" href="#">About Us</a></li>
    </ul>
  </div>
</div>

</body>
</html>
```

Dropdowns

Jquery den faydalanarak dropdownlist kullanımı:

Tutorials ▼

Dropdown basligi 1

HTML

CSS

JavaScript

Dropdown basligi 2

About Us