

BLM3580

Sistem Programlama

2021-2022 GÜZ YARIYILI

DR.ÖĞR.ÜYESİ GÖKSEL BIRICIK

JavaScript

JavaScript

Netscape tarafından, önce Mocha, sonra LiveScript.

- Java'ya benzediği için JavaScript adı verildi.
- 1997-1999 ECMA (European Computer Manufacturer Association) tarafından standart haline getirilerek ECMAScript (ECMA-262) olarak adlandırıldı.

JavaScript,

- istenildiği gibi düzenlenebilen (free formatted),
- yorumlanan (interpreted),
- taşınabilir (portable) ve
- donanımdan bağımsız,
- buyuran (imperative) ve yapıli (structured),
- Zayıf tipli (weakly typed)

bir betik (script) dilidir. Hafif sıklettir, HTML içeriği ile etkileşebilir.

ECMAScript Editions

| Ver | Official Name | Description |
|-----|-----------------------|---|
| 1 | ECMAScript 1 (1997) | First Edition. |
| 2 | ECMAScript 2 (1998) | Editorial changes only. |
| 3 | ECMAScript 3 (1999) | Added Regular Expressions. Added try/catch. |
| 4 | ECMAScript 4 | Never released. |
| 5 | ECMAScript 5 (2009) | Added "strict mode". Added JSON support. Added String.trim(). Added Array.isArray(). Added Array Iteration Methods. |
| 5.1 | ECMAScript 5.1 (2011) | Editorial changes. |
| 6 | ECMAScript 2015 | Added let and const. Added default parameter values. Added Array.find(). Added Array.findIndex(). |
| 7 | ECMAScript 2016 | Added exponential operator (**). Added Array.prototype.includes. |
| 8 | ECMAScript 2017 | Added string padding. Added new Object properties. Added Async functions. Added Shared Memory. |
| 9 | ECMAScript 2018 | Added rest / spread properties. Added Asynchronous iteration. Added Promise.finally(). Additions to RegExp. |
| 10 | ECMAScript 2019 | Added Array.prototype.flat, Array.prototype.flatMap, Array.sort is stable. |
| 11 | ECMAScript 2020 | BigInt (integers beyond Number.MAX_SAFE_INTEGER), ?? (null coalescing syntax) |

JavaScript Komutları

JavaScript küçük-büyük harf duyarlıdır.

| | | | | | | | |
|----------|--------|----------|--------|-------|--------|--------|------|
| break | case | continue | delete | do | else | false | for |
| function | if | in | dew | null | return | switch | this |
| true | typeof | var | void | while | with | | |

Komut olarak kullanılmasa da özel anlamlı kelimeler:

| | | | | | |
|--------|-----------|---------|----------|---------|------|
| catch | class | const | debugger | default | enum |
| export | extentsds | Finally | import | super | try |

HTML'de Kullanımı

| Etiket Yapısı | Açıklama | HTML | 3.x | HTML | 4.x | HTML | 5.x |
|--|---|------|-----|------|-----|------|-----|
| <code><script src="URL" type="{ text/ecmascript text/javascript text/jscript text/vbscript text/vbs text/xml}"> </script></code> | | ✓ | | ✓ | | ✗ | |
| <code><script charset="**" defer src="URL" type="media_type" </script></code> | charset: kullanılacak kodlama sistemini belirler defer: sayfa yüklenmesi bitene kadar script'in çalışması bekletilir. src: script'in yükleneceği URL type: MIME tipi olarak "text/javascript" kullanılır. | | | ✓ | | ✓ | |
| <code><script async </script></code> | async özniteliği sadece harici script dosyaları için kullanılır. Bu özniteliğin kullanılması durumunda sayfanın ayrıştırılması (parsing) sırasında script çalıştırılacaktır. Eger bu öznitelik kullanılmamış ise script sayfanın ayrıştırılması bittiğinde çalışmaya başlayacaktır. | ✗ | | ✗ | | ✓ | |

<body> ve <head> içinde JavaScript

JavaScript'ler <body> bloğu içinde kodlandıkları zaman sayfa yüklenirken çalıştırılırlar.

- Sayfanın içeriğinin oluşturulmasını sağlarlar.

JavaScript'ler <head> bloğu içinde yazılmışlar ise onları çağıran bir durum oluştuğunda devreye girerler.

- Bir olay oluşması veya fonksiyonun çağırılması gereklidir.

CSS'deki kullanım gibi her sayfada kullanılması gereken JavaScript'lerin harici bir dosyada tutulması ve HTML sayfalarına dâhil edilmesi gerek programlamayı gerekse sayfalar üzerindeki bakım faaliyetlerini kolaylaştırır.

- <head>

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

- </head>

Veri Tipleri ve Değişkenler

`var ad="Göksel"` `ad="Göksel"`

Değişkenler eğer bir fonksiyonun içinde tanımlanmış ise yerel değişken (local variable) olarak adlandırılır ve sadece o fonksiyon içinden erişilebilir.

Fonksiyonların dışında tanımlanan değişkenlerin ömrü sayfanınki kadardır.

| Veri tipleri | Örnek |
|--------------|---|
| string | <code>var harfKatar1 = 'katar tipinde veri' veya</code> <code>var harfKatar2 = "katar tipinde veri"</code> |
| number | <code>var tamSayı =1234</code> |
| number | <code>var kesirliSayı = 23.50</code> |
| boolean | <code>var dogruMu = false</code> |
| Array | |
| object | |
| BigInt | |

Aritmetik Operatörler

| Operatör | Açıklama | Örnek | Sonuç |
|----------|----------|---------------------------|-------|
| + | Toplama | $x=2 \Rightarrow x+2$ | 4 |
| - | Çıkarma | $x=2 \Rightarrow x-2$ | 0 |
| * | Çarpma | $x=2 \Rightarrow x*2$ | 4 |
| / | Bölme | $x=2 \Rightarrow x/2$ | 1 |
| % | Modülo | $x=2 \Rightarrow x\%2$ | 0 |
| ++ | Artırma | $x=2 \Rightarrow x++$ | 3 |
| -- | Azaltma | $x=2 \Rightarrow x--$ | 1 |
| ** | Üs Alma | $x=2 \Rightarrow x^{**}3$ | 8 |

Atama Operatörleri

| Operatör | Açıklama | Örnek | Sonuç |
|----------|---------------------|------------------------------|---------------|
| = | Atama | $x=2, y=4 \Rightarrow x=y$ | $x=4$ |
| += | Toplayarak Atama | $x=2, y=4 \Rightarrow x+=y$ | $x=x+y = 6$ |
| -= | Çıkartarak Atama | $x=2, y=4 \Rightarrow x-=y$ | $x=x-y = -2$ |
| *= | Çarparak Atama | $x=2, y=4 \Rightarrow x*=y$ | $x=x*y = 8$ |
| /= | Bölerek Atama | $x=2, y=4 \Rightarrow x/=y$ | $x=x/y = 0.5$ |
| %= | Modülo Alarak Atama | $x=2, y=4 \Rightarrow x\%=y$ | $x=x\%y = 2$ |

Karşılaştırma Operatörleri

| Operatör | Açıklama | Örnek | Sonuç |
|----------|-----------------------------|---------------|-------|
| == | Eşit mi? | 2 == 4 | False |
| === | Veri tipi aynı ve eşit mi ? | 123 === "123" | False |
| != | Farklı mı ? | 2 != 4 | True |
| > | Büyük mü? | 2 > 4 | False |
| < | Küçük mü ? | 2 < 4 | True |
| >= | Büyük Eşit mi? | 2 >= 4 | False |
| <= | Küçük Eşit mi ? | 2 <= 4 | True |

Karşılaştırma Operatörleri Sonuç Tablosu

| x | y | == | === | Object.is | SameValueZero |
|-------------------|-------------------|-------|-------|-----------|---------------|
| undefined | undefined | true | true | true | true |
| null | null | true | true | true | true |
| true | true | true | true | true | true |
| false | false | true | true | true | true |
| 'foo' | 'foo' | true | true | true | true |
| 0 | 0 | true | true | true | true |
| +0 | -0 | true | true | false | true |
| 0 | false | true | false | false | false |
| "" | false | true | false | false | false |
| "" | 0 | true | false | false | false |
| '0' | 0 | true | false | false | false |
| '17' | 17 | true | false | false | false |
| [1, 2] | '1,2' | true | false | false | false |
| new String('foo') | 'foo' | true | false | false | false |
| null | undefined | true | false | false | false |
| null | false | false | false | false | false |
| undefined | false | false | false | false | false |
| { foo: 'bar' } | { foo: 'bar' } | false | false | false | false |
| new String('foo') | new String('foo') | false | false | false | false |
| 0 | null | false | false | false | false |
| 0 | NaN | false | false | false | false |
| 'foo' | NaN | false | false | false | false |
| NaN | NaN | false | false | true | true |

Mantıksal Operatörler

| Operatör | Açıklama | Örnek | Sonuç |
|----------|-------------|---|-------|
| && | And – Ve | $x=2, y=4 \Rightarrow (x<4 \ \&\& \ y>2)$ | True |
| | Or – Veya | $x=2, y=4 \Rightarrow (x<1 \ \ y>2)$ | True |
| ! | Not – Değil | $x=2, y=4 \Rightarrow (x \ ! \ y)$ | True |

Fonksiyonlar

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

```
alert("Alert kutusu açılır!");
```

```
...
```

```
var name = confirm("Tuşa basınız");
```

```
if (name == true) { document.write("Tamam'a bastınız") }
```

```
else { document.write("İptal'e bastınız") }
```

```
...
```

```
var name = prompt("Adınız ?", "");
```

```
if (name != null && name != "") { document.write("Memnun oldum" + name)}
```

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

Fonksiyonlar

```
<body> <script type="text/javascript">
    var fnum, snum, num1, num2, sum;
    fnum = window.prompt("ilk sayıyı veriniz", "0");
    snum = window.prompt("ikinci sayıyı veriniz", "0");
    num1 = parseInt(fnum);
    num2 = parseInt(snum);
    sum = num1 + num2;
    document.writeln("<p>Sayıların toplamı = " + sum + " dır");
</script> </body>
```

Kesirli sayılar için parseFloat kullanılır.

Return

```
<html> <head> <script type="text/javascript" >
    function square(x) { return x*x    }
</script></head>
<body> <script type="text/javascript">
    for (var i=1; i<= 10;++i) {
        document.writeln(i + "sayısının karesi = " + square(i) + "<br>")
    }
</script></body>
</html>
```


If-Else, Switch

```
<script type="text/javascript">
    var d = new Date()
    var time = d.getHours()
    if (time < 10) {
        document.write("Günaydın")
    }
    else {
        document.write("İyi günler")
    }
</script>
```

```
<script type="text/javascript">
    var d=new Date()
    theDay=d.getDay()
    switch (theDay) {
        case 6:    document.write("Cumartesi")
                    break
        case 0:    document.write("Pazar")
                    break
        default:   document.write("İş günü")
    }
</script>
```

Döngüler

while (koşul)

{

.....

}

do{

.....

}

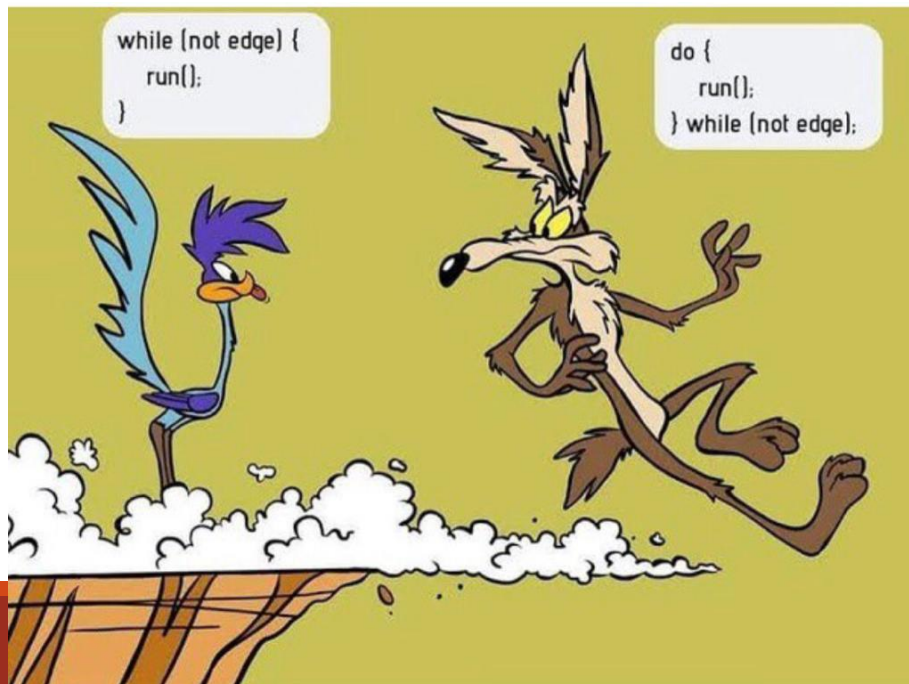
while(koşul)

for(başlangıç;koşul; artım)

{

.....

}



Nesne Kullanımı

| Örnek JavaScript Kod | Ekran Çıktısı |
|---|--------------------------------------|
| <pre><script type="text/javascript"> var eMail = new String("gbiricik@yildiz.edu.tr"); document.write(typeof(eMail)); document.write(" "); document.write(eMail.length); document.write(" "); document.write(eMail.indexOf("@")); var eMailDomain = eMail.substring(eMail.indexOf("@") +1, eMail.length); document.write(" "); document.write(eMailDomain); </script></pre> | <pre>object 22 8 yildiz.edu.tr</pre> |

Diziler (Array)

| Kullanım | Açıklama |
|---|---|
| <code>var dizi = new Array (12);</code> | 12 elemanlı dizi tanımı |
| <code>var dizi = [10,20,30,40,50];</code> | 5 elemanlı ilk değerleri belli dizi |
| <code>var dizi = new Array (10,20,30,40,50);</code> | 5 elemanlı ilk değerleri belli dizi |
| <code>var dizi = [10,20,,40,50];</code> | 5 elemanlı dizi. 4 elemanın ilk değerleri belli |

```
<script type="text/JavaScript">
    function start() {
        var dizi = [1,2,3,4,5];
        var toplam1 = 0, toplam2 = 0
        for (var i= 0; i < dizi.length; i++)      toplam1 += dizi[i];
        document.writeln("İlk toplam = " + toplam1);
        for (var eleman in dizi)      toplam2 += dizi[eleman];
        document.writeln("İkinci toplamı = " + toplam2);
    }
</script>
```

Dizi Sıralama

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

```
    var bizimArkadaslar = new Array( "Can", "Lale", "Ayşe", "Pelin", "Burak" );
```

```
    bizimArkadaslar.sort( );
```

```
    for (i in bizimArkadaslar){
```

```
        document.write(bizimArkadaslar[i] + "<br>" );
```

```
    }
```

```
</script>
```

Parametre Aktarımı

```
<script language="JavaScript">
    function start() {
        var dizi = [1,2,3,4,5];
        document.writeln("Dizinin ilk hali");
        for (var i= 0; i < dizi.length; i++)
            document.writeln("dizi[" + i + "]= " + dizi[i]);
        document.writeln("Dizinin işlem sonrası hali");
        carp(dizi);
        for (var i= 0; i < dizi.length; i++)
            document.writeln("dizi[" + i + "]= " + dizi[i]);
    }
    function carp (bilgi) {
        for ( var i in bilgi)
            bilgi[i] *=2;
    }
</script>
```

DHTML

W3C standardı değildir. Olay tabanlı çalışır.

| Olay | Açıklama |
|-------------|---|
| OnClick | Fare ile tıklanması |
| OnLoad | Sayfanın yüklenmesi |
| OnError | Sayfada hata oluşması |
| OnMouseOver | Farenin bir elemanın sınırları içinde olması |
| OnMouseOut | Farenin bir elemanın sınırları dışına hareket ettirilmesi |
| OnChange | Bir değişiklik olduğu zaman |
| OnSubmit | Submit butonuna tıklandığı zaman |
| OnMouseMove | Farenin bir elemanın üzerine hareket ettirilmesi |
| OnReset | Formun temizlenmek için reset butonuna basılması |
| OnFocus | Bir eleman üzerine odaklanıldığında |
| OnBlur | Bir eleman üzerindeki odaklanma işleminin sonlanması |

DHTML Örnekleri

```
html>
<head>
  <script type="text/JavaScript">
var seconds = 0;
function timer() {
  window.setInterval ("updatetime()",1000);
}
function updatetime() {
  seconds++;
  soFar.innerHTML = seconds;
}
</script>
</head>
<body OnLoad="timer()">
  Geçen Zaman: <a ID="soFar">0</a>
</body>
</html>
```

```
<html>
<head>
<script type="text/javascript">
function cursorPos() {
  document.getElementById('xxx').innerHTML=event.clientX
  document.getElementById('yyy').innerHTML=event.clientY
}
</script>
</head>
<body onmousemove="cursorPos()">
  cursor X: <span id="xxx"> </span> <br/>
  cursor Y: <span id="yyy"> </span>
</body>
</html>
```


AJAX - Asynchronous JavaScript & XML

Javascript, XML, HTML ve CSS'nin bir arada kullanılması ile oluşur.

Yeni nesil bir programlama dili olmayıp, hızlı, güzel web sayfaları oluşturmak için mevcut araçların farklı bir şekilde kullanılmasıdır.

Her submit'te sayfanın tekrar işlenmesi ve yanıt üretilmesi zahmetli ve maliyetlidir.

XMLHttpRequest nesnesi sayesinde head bloğundaki JavaScript kodları, sayfanın yeniden yüklenmesine gerek kalmadan işlem gerçekleştirebilir.

```
<html>
  <head>
    <script>
      function loadXMLDoc(){
        var xmlhttp;
        if (window.XMLHttpRequest) {
          // IE7+, Firefox, Chrome, Opera, Safari, için XMLHttpRequest() kullanılır.
          xmlhttp=new XMLHttpRequest();
        }
        else
        {
          // IE5 ve IE6 XMLHttpRequest() yerine ActiveXObject() kullanılır.
          xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange=function() {
          if (xmlhttp.readyState==4 && xmlhttp.status==200) {
            document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
          }
        }
        xmlhttp.open("GET","ajax_info.txt",true);
        xmlhttp.send();
      }
    </script>
  </head>
  <body>
    <div id="myDiv">
      <h2>Bu yazı AJAX kullanılarak değiştirilecektir.</h2>
    </div>
    <button type="button" onclick="loadXMLDoc()">Değişiklik yapmak için tıkla</button>
  </body>
</html>
```

open(method,url,async)

GET ve POST Farkı

GET metodu daha hızlı ve basittir, aktarılan parametre değişmediği sürece daha önce aldığı ve cache bellekte bulunan veriyi kullanır.

Ancak veritabanında bir güncelleme yapılacak veya bir dosya yüklenecek ise, sunucuya yollanacak veri büyük ve özellikle kullanıcı tarafından girilen veri (ne olduğu bilinmeyen karakterlerden oluşabilen) sunucuya aktarılacak ise mutlaka POST metodu kullanılmalıdır.

| | |
|------|---|
| GET | <pre>xmlhttp.open("GET","info.php?adi=Goksel&sadi=Biricik",true); xmlhttp.send();</pre> |
| POST | <pre>xmlhttp.open("POST","info.php",true); xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded"); xmlhttp.send("adi=Goksel&sadi=Biricik");</pre> |

open(method,url,async) URL, async

Sunucu tarafındaki dosyayı belirler.

Bu dosya XML, TXT uzantılı bir metin dosyası olabileceği gibi PHP, JSP vb. bir sunucu taraflı kod dosyası da olabilir.

Async, true (asynchronous) veya false (synchronous) değerlerini alabilir. AJAX, asynchronous (an uyumsuz) olduğu için değerin true seçilmesi gereklidir.

An uyumsuz çalışma nedeni ile yapılan isteğin cevabının sunucudan ne zaman döneceği de tam olarak bilinemez.

- Bunun için onreadystatechange olayının (event) oluşması ile tetiklenecek olan bir fonksiyonun kullanılması gereklidir
- readyState değeri 4 ise bağlantının tamamlandığı, verinin gönderip/alındığı anlaşılır.
- status değeri 200 ise bağlantının olması gerektiği şekilde sonlandığı anlaşılır. (http 200 OK)

jQuery

AJAX'ı düz JavaScript ile kullanmak zordur.

Browserlar arası farkları ele almamız gerekir.

jQuery, oluşturulan web sitelerinin ve/veya web uygulamalarının farklı web gezgini yazılımları tarafından doğru/düzgün biçimde görselleştirilmesine (Cross-browser) imkân veren, istemci tarafındaki işleyişi kolaylaştıran ve açık kaynak kodlu (MIT ve GNU lisansı ile dağıtımı yapılan) bir JavaScript kütüphanesidir.

jQuery den yararlanabilmek için jquery.com adresinden uygun betik dosyasının indirilmeli ve `<head><script src="jquery-1.11.0.min.js"></script></head>` gibi bir tanım ile dahil edilmelidir.

jQuery Sözdizimi

| Selector | Açıklama |
|-------------------------|---|
| \$(this) | Geçerli elemanı seçer |
| \$("h1") | Tüm <h1> elemanlarını seçer |
| \$("p:first") | İlk <p> elemanını seçer |
| \$("p:last") | Son <p> elemanını seçer |
| \$("p:first-child") | Ailesinin ilk üyesi olan <p> elemanını seçer |
| \$("p:first-of-type") | Ailesinin ilk üyesi olan tüm <p> elemanlarını seçer |
| \$("p:last-child") | Ailesinin son üyesi olan <p> elemanını seçer |
| \$("p:last-of-type") | Ailesinin son üyesi olan tüm <p> elemanlarını seçer |
| \$("p.intro") | class="intro" olan tüm <p> elemanlarını seçer |
| \$("li:nth-child(1)") | Liste içindeki ilk elemanını seçer |
| \$("li:nth-of-type(2)") | Listeler içindeki ikinci elemanlarını seçer |
| \$("tr:even") | Tablonun çift numaralı <tr> elemanlarını seçer |
| \$("tr:odd") | Tablonun tek numaralı <tr> elemanlarını seçer |
| \$("div > p") | <div> elemanının çocuğu durumdaki tüm <p> elemanlarını seçer |
| \$("div p") | <div> elemanının doğrudan altında yer alan tüm <p> elemanlarını seçer |
| \$("ul li:eq(0)") | elemanı altındaki 0. elemanını seçer |
| \$("ul li:lt(2)") | elemanı altındaki indisi 2'den küçük elemanlarını seçer |
| \$("ul ~ table") | elemanı ile eş düzeyde (kardeş) <table> elemanlarını seçer |

jQuery Sözdizimi

| Selector | Açıklama |
|---------------------------------------|---|
| <code>\$("ul ~ table")</code> | elemanı ile eş düzeyde (kardeş) <table> elemanlarını seçer |
| <code>\$("div:has(p)")</code> | İçinde <p> elemanı bulunan <div> elemanlarını seçer |
| <code>\$(":contains(abc)")</code> | İçinde abc katarının geçtiği tüm elemanları seçer |
| <code>\$(":header:not(h1)")</code> | <h1> haricindeki <h> elemanlarını seçer |
| <code>\$(":checkbox")</code> | type="checkbox" olan tüm elemanları seçer |
| <code>\$(":submit")</code> | type="submit" olan tüm elemanları seçer |
| <code>\$(":radio")</code> | type="radio" olan tüm elemanları seçer |
| <code>\$(":checked")</code> | İşaretli (checked) olan tüm form elemanlarını seçer |
| <code>\$("[id]")</code> | id özniteliği olan tüm elemanları seçer |
| <code>\$("[id =ab]")</code> | id özniteliği olan ve değeri ab veya ab- ile başlayan elemanları seçer |
| <code>\$("[id\$=xyz]")</code> | id özniteliği olan ve değeri xyz ile biten elemanları seçer |
| <code>\$("[id^=A]")</code> | id özniteliği olan ve değeri A harfi ile başlayan elemanları seçer |
| <code>\$("[id=adres]")</code> | id özniteliği olan ve değeri adres olan elemanları seçer |
| <code>\$("[id*=s]")</code> | id özniteliği olan ve değerinde s geçen elemanları seçer |
| <code>\$("[abc~=xyz]")</code> | abc isimli özniteliği olup değeri xyz olan elemanları seçer |
| <code>\$("p:dil(tr)")</code> | dil isimli özniteliği tr olan <p> elemanlarını seçer |
| <code>\$("a[target='_blank']")</code> | Target isimli özniteliğinin değeri _blank olan tüm <a> elemanlarını seçer |

jQuery Örnekleri

| Kullanım | Açıklama |
|---------------------------------|---|
| <code>\$(this).hide()</code> | Geçerli elemanı gizler |
| <code>\$("p").hide()</code> | Tüm <code><p></code> elamanlarını gizler |
| <code>\$(".test").hide()</code> | <code>class="test"</code> olarak tanımlanmış her elemanı gizler |
| <code>\$("#test").hide()</code> | <code>id="test"</code> olarak tanımlanmış her elemanı gizler |

```
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
<script>
$(function){
    // jQuery metodları
});
</script>
</head>
```


| index.html | style.css | app.js |
|--|---|--|
| <pre> <!DOCTYPE html> <html> <head> <meta charset='utf-8' /> <title>Basit jQuery Kullanım Örneği</title> <script src="jquery-1.11.0.min.js"></script> <script src="app.js"></script> <link rel="stylesheet" type="text/css" href="style.css" /> </head> <body> <div id="box">kutu</div> </body> </html> </pre> | <pre> #box { width: 100px; height: 100px; text-align: center; background: #f00; font-size: 14px; } </pre> | <pre> \$(function() { var box = \$("#box"); box.fadeOut("slow", function() { alert("kutu kayıp oldu."); }); }); </pre> |

jQuery Olay Tipleri

| Fare | Klavye | Form | Belge/Pencere |
|------------|-----------|--------|---------------|
| click | keypresed | submit | load |
| dbclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave | | blur | unload |

```
<!DOCTYPE html>
<html>
  <head>    <script src="jquery-1.11.0.min.js" />
             <script>
               $(function(){$("#p1").mouseenter( function(){alert("Fareyi yazının üzerine getirdiniz."); });
               });
             </script></head>
  <body>    <p id="p1">Fareyi bu yazı üzerine getirin.</p>                </body>
</html>
```

Gelecek Ders

jQuery Detayları