

Web Teknolojileri

Hafta 12

İçerik

- ▶ Javascript

JavaScript

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

JavaScript

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

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

JavaScript

► JavaScript <head> tagı örneği.

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

JavaScript

► JavaScript <body> tagı örneği.

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

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

JavaScript

► JavaScript harici dosya ile kullanımı.

► myScript.js

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

► body içinde çağırma

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

head içinde çağırma

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

JavaScript Görüntü Olanakları

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

JavaScript **window.alert()**

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

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

</body>
</html>
```

JavaScript **document.write()**

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<script>
```

```
document.write(7 + 6);
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript **document.write()**

- ▶ `document.write()` ile tüm içeriği silebilirsiniz.
- ▶ Aşağıdaki kodda butona tıklayınca tüm içerik silinir ve **11** yazar.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>İlk Web Sayfam</h1>
```

```
<p>İlk Paragrafım</p>
```

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

```
</body>
```

```
</html>
```

JavaScript **innerHTML**

► Bir HTML öğesine erişmek için

document.getElementById (id) kullanılır

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>İlk Web Sayfam</h1>
```

```
<p>İlk Paragrafım</p>
```

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

```
<script>
```

```
document.getElementById("demo").innerHTML = 5 + 6;
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript **console.log()**

- ▶ Tarayıcıdaki konsola log yazdırmak için kullanılır.
- ▶ F12 ile tarayıcı konsolunu etkinleştirin ve sekmelerden Console seçiniz.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>İlk Web Sayfam</h1>
```

```
<p>İlk Paragrafım</p>
```

```
<script>
```

```
console.log(7 + 4);
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript sözdizimi

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

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

- ▶ Metinler çift veya tek tırnak içinde yazılır.

"Sakarya Üniversitesi"

'Sakarya Üniversitesi'

JavaScript sözdizimi

- ▶ Değişkenler Html içeriğe aktarılabilir.

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

```
<script>
```

```
var carName = "Volvo";
```

```
document.getElementById("demo").innerHTML = carName;
```

```
</script>
```

JavaScript sözdizimi

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

```
var x = 5;    // Komut çalışır  
// var x = 6;  Komut Çalışmaz
```

```
/*Birden fazla satır  
Yorum bu şekilde yapılır*/
```

- Değişken tanımlarken alt çizgi yada camelcase metodu kullanılabilir.

```
var ad_soyad, kullanıcı_adi;
```

```
var AdSoyad, KullaniciAdi;
```


JavaScript sözdizimi

- ▶ Aritmetik operatörler hesaplama işlemleri için kullanılır.

`(5 + 6) * 10`

- ▶ Değişkenler ile aritmetiksel işlemler yapılabilir.

`x * (5 + 6) * 10`

- ▶ Metinler `+` ile birleştirilebilir.

`"Sakarya" + " " + "Üniversitesi"`

JavaScript Operatörler

Operator	Description
----------	-------------

+	Toplama
---	---------

-	Çıkarma
---	---------

*	Çarpma
---	--------

/	Bölme
---	-------

%	Mod Alma
---	----------

++	Artırma
----	---------

--	Azaltma
----	---------

JavaScript Operatörler

Arttırma

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

Azaltma

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

Çarpma

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

Toplama

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

Çıkarma

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

Bölme

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

JavaScript Atama Operatörler

Operatör	Kullanımı	Anlamı
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$

JavaScript Atama Operatörler

Atama

```
var x = 10;
```

Atama

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

Atama

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

Atama

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

Atama

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

Atama

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

JavaScript Veri Tipleri

Dinamik tipler.

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

Ondalık ve tamsayı tipler.

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

Boolean tipler.

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

JavaScript Diziler

Diziler [] içinde tanımlanır.

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

JavaScript Fonksiyonlar

Bir JavaScript fonksiyonu belirli bir görevi yerine getirmek için tasarlanmış bir kod bloğudur

```
function myFunction(p1, p2) {  
    return p1 * p2;           // Bu fonksiyon p1 ve p2 çarpımını döndürür  
}
```


JavaScript Fonksiyon Syntax

Javascript fonksiyonu function anahtar kelimesi ile tanımlanır.

Fonksiyon isimleri harfler, sayılar, altçizgi içerebilir.

Fonksiyonlar birden fazla parametre alabilir.

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

JavaScript Fonksiyon Geri Değeri

Javascript fonksiyonları return anahtar kelimesi ile geri değer döndürürler.

```
var x = myFunction(4, 3);  
/* Fonksiyon çağırılır, return değeri x değişkenine atanır */  
  
function myFunction(a, b) {  
    return a * b;  
// Fonksiyon a ve b değerinin çarpımını geri gönderir  
}
```

JavaScript Fonksiyon Örneği

Aşağıdaki fonksiyon fahrenheit derece değerini celsius değer çevirir.

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

Fonksiyonu Kullanım Örnekleri

```
document.getElementById("demo").innerHTML = toCelsius(77);
```

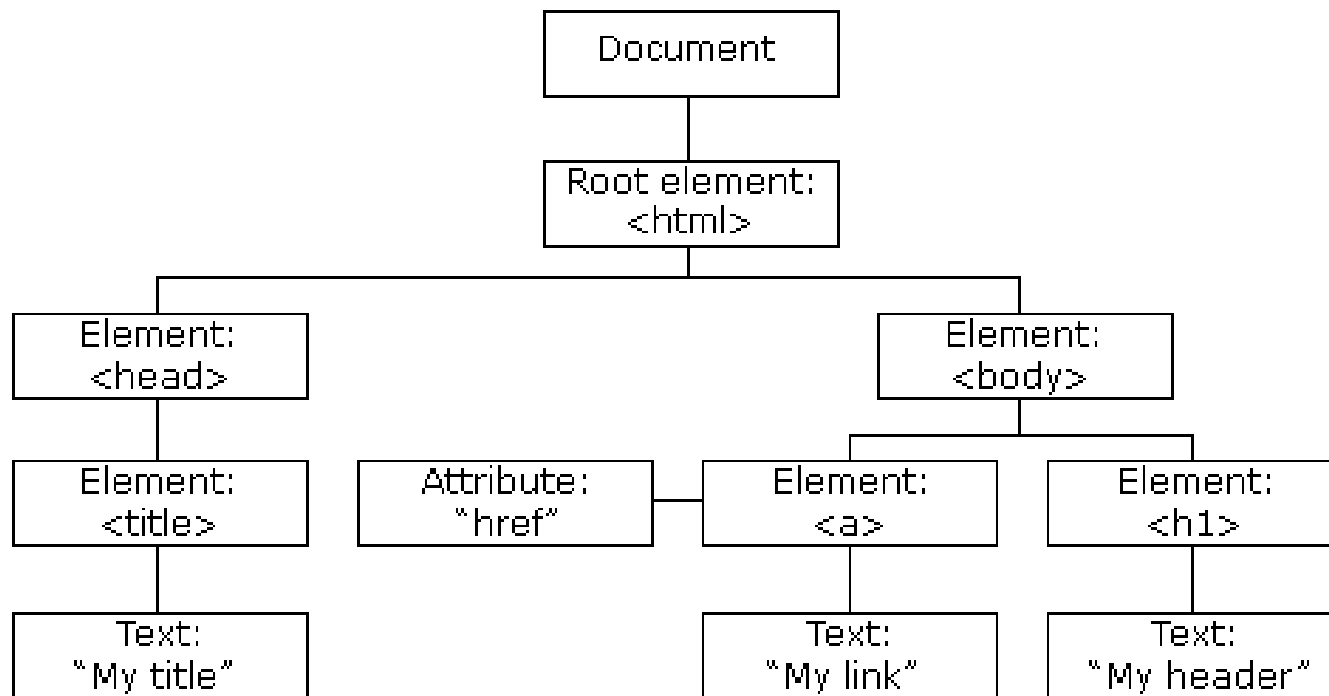
```
var text = "Sıcaklık :" + toCelsius(77) + " Celsius";
```

```
/* Farklı bir kullanımı */  
var x = toCelsius(32);  
var text = "Sıcaklık :" + x + " Celsius";
```

JavaScript Html DOM

HTML DOM ile JavaScript erişebilir ve bir HTML belgesinin tüm unsurları değiştirebilir.

Nesneler HTML DOM Ağacı



JavaScript Html DOM

DOM nedir?

DOM W3C (World Wide Web Consortium) standardıdır.

DOM belgeleri erişmek için bir standart tanımlar:

HTML Dom aşağıdakileri yapabilir.

1. HTML elemanları **nesne** olarak tanımlar
2. Tüm Html elemanların **özelliklerini** kullanır.
3. Tüm Html öğelerin **yöntemlerine** erişebilir.
4. Tüm Html elemanlarının **olaylarını** tanımlar.

JavaScript Html DOM

HTML DOM ile aşağıdaki işlevler yapılabilir.

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

JavaScript Html DOM

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

```
<html>
```

```
<body>
```

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

```
<script>
```

```
document.getElementById("demo").innerHTML = "Merhaba Arkadaşlar";
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript Html DOM

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

```
<html>
```

```
<body>
```

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

```
<script>
```

```
document.getElementById("demo").innerHTML = "Merhaba Arkadaşlar";
```

```
</script>
```

```
</body>
```

```
</html>
```


JavaScript Html DOM

getElementById farklı çeşitleri vardır.

Metot

Açıklama

`document.getElementById(id)`

id ile elemana erişim

`document.getElementsByTagName(name)`

Etiket ismi ile elemana erişim

`document.getElementsByClassName(name)`

Class ismi ile elemana erişim

JavaScript Html DOM

HTML elemanlarının özelliklerini değiştirebilir.

Metot

Açıklama

`element.innerHTML = yeni html içeriği`

Elemanın Html içeriği değiştirilir

`element.ozellik = yeni değer`

Elemanların özelliklerine değer atama

`element.setAttribute(ozellik,deger)`

Elemanların özelliklerine değer atama

`element.style.property = yeni stil`

Elemanlara stil tanımlama

JavaScript Html DOM

HTML elemanı ekler yada silebilir.

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

HTML olayı ekleyebilir.

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

JavaScript Html DOM

- ▶ HTML taglarına erişerek değiştirilebilir.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1 id="header">Eski Başlık</h1>
```

```
<script>
```

```
var element = document.getElementById("header");
```

```
element.innerHTML = "Yeni Başlık";
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript Html DOM

- ▶ HTML tagların özelliklerine erişerek değiştirilebilir.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```

```

```
<script>
```

```
document.getElementById("myImage").src = "landscape.jpg";
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript Olaylar

HTML Dom Olayları

Her bir HTML elemanına olay eklenebilir.

Olaylara tepki verilebilir.

HTML olayları örnekler:

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

Olaylar OnClick

Bir öğeye tıklayınca çalışır.

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

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

</body>
</html>
```

Olaylar OnLoad, OnUnload

Kullanıcı web sayfasına girdiği anda sayfa yüklenince **onLoad** olayı çalışır. Kullanıcı sayfadan ayrılınca **onUnload** olayı çalışır.

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

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


Olaylar OnChange

Html ögesinin içeriği değiştirildiğinde çalışır.

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

OnMouseOver

Html ögesine fare ile üzerine gelindiğinde çalışır.

OnMouseOut

Html ögesinden fare ile üzerinden çıkıldığında çalışır.

OnMouseOver ve OnMouseOut Örnek

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

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

</body>
</html>
```

JavaScript Form Elemanlar

Aşağıdaki javascript kodu frm1 isimli formdaki tüm elemanların içindeki değerleri demo Id değere sahip p tagına basar.

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

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

JavaScript Form Doğrulama

Form kontrolü yapılabilir.

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

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