

Web Teknolojileri

JavaScript

1/2

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

Konu & İçerik

➤ JavaScript

JavaScript

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

JavaScript...

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

JavaScript...

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

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

JavaScript...

➤ JavaScript <head> etiketi örneği.

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

JavaScript...

➤ JavaScript <body> etiketi örneği.

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

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

JavaScript...

➤ JavaScript'in harici dosya ile kullanımı.

➤ myScript.js

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

➤ <body> etiketi içerisinde çağırma

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


JavaScript...

➤ JavaScript harici dosya ile kullanımı.

➤ myScript.js

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

➤ <head> etiketi içerisinde çağırma

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

JavaScript...

➤ JavaScript ile HTML sayfasına farklı şekillerde yazdırma işlemi yapılabilir.

- Uyarı penceresi → **window.alert();**
- HTML sayfaya yazdırma → **document.write();**
- HTML tagi içine yazdırma → **innerHTML**
- Tarayıcı konsol içine yazdırma → **console.log()**

JavaScript...

➤ **window.alert()**

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

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

</body>
</html>
```

JavaScript...

➤ document.write()

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

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

</body>
</html>
```

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

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

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

</body>
</html>
```

JavaScript...

➤ innerHTML

➤ document.getElementById (id) – Bir HTML ögesine erişmek için kullanılır.

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

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

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

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

</body>
</html>
```

JavaScript...

➤ **console.Log()**

- Tarayıcıdaki konsola log 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 içeriğine aktarılabilir.

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

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


JavaScript Sözdizimi

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

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

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

- Değişken tanımlarken alt çizgi yada camelcase metodu kullanılabilir.

```
var ad_soyad, 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

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

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

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

```
// Atama  
var x = 10;
```

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

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

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

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

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

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

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

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

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

JavaScript Veri Tipleri

➤ Dinamik Veri Tipleri

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

➤ Ondalık ve Tamsayı Tipler

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

➤ Boolean veri tipleri

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

JavaScript Diziler

➤ Diziler [] içinde tanımlanır.

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

JavaScript Fonksiyonlar

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

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

JavaScript Fonksiyonlar...

➤ Javascript fonksiyonları **return** anahtar kelimesi ile geri değer döndürürler.

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

JavaScript Fonksiyonlar...

➤ ÖRNEK:

➤ fahrenheit değerini derece değerine çeviren fonksiyon örneği

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

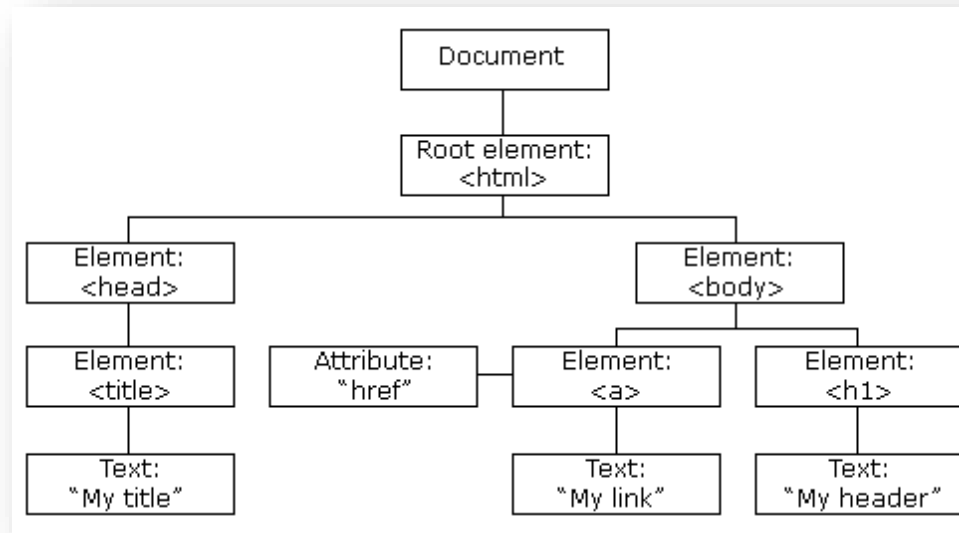
```
// Fonksiyon Kullanımı - Örnek 1  
document.getElementById("demo").innerHTML = toCelsius(77);
```

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

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


JavaScript – HTML DOM

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



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

JavaScript – HTML DOM...

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

JavaScript – HTML DOM...

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

JavaScript – HTML DOM...

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

```
<html>
<body>

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

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

</body>
</html>
```

JavaScript – HTML DOM...

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

```
<html>
<body>

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

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

</body>
</html>
```

JavaScript – HTML DOM...

➤ **getElementById** farklı çeşitleri vardır.

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

JavaScript – HTML DOM...

➤ HTML elemanlarının özelliklerini değiştirebilir.

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

JavaScript – HTML DOM...

➤ Yeni HTML elemanı ekler yada mevcut HTML elemanını silebilir.

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

JavaScript – HTML DOM...

➤ Yeni HTML olayı ekleyebilir.

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

JavaScript – HTML DOM...

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

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

<h1 id="header">Eski Başlık</h1>

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

</body>
</html>
```

JavaScript – HTML DOM...

- HTML etiketlerinin özniteliklerine erişerek değiştirilebilir.

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



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

</body>
</html>
```

JavaScript Olayları

➤ HTML DOM Olayları

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

➤ HTML olaylarına bazı örnekler

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

JavaScript Olayları

➤ **OnClick**

➤ Bir öğeye tıklayınca çalışır

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

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

</body>
</html>
```

JavaScript Olayları...

➤ **OnLoad**

- Kullanıcı web sayfasına girdiği anda sayfa yüklenince çalışır.

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

➤ **OnunLoad**

- Kullanıcı web sayfasından ayrılınca çalışır.

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

JavaScript Olayları...

➤ **OnChange**

- Html ögesinin içeriği değiştirildiğinde çalışır.

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

➤ **OnMouseOver**

- Html ögesine fare ile üzerine gelindiğinde çalışır.

➤ **OnMouseOut**

- Html ögesinden fare ile üzerinden çıkıldığında çalışır

JavaScript – ÖRNEKLER...

➤ OnMouseOver & OnMouseOut

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


JavaScript – ÖRNEKLER...

➤ Form Elemanları

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

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

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

JavaScript – ÖRNEKLER...

➤ Form Doğrulama

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

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

Kaynaklar

➤ <http://www.w3schools.com/>