

JAVASCRIPT

JAVASCRIPT DİLİ

Netscape Navigator 2.0 ile birlikte geliştirilen JavaScript dili istemci taraflı (client-side) script dilidir. Web sayfalarında dinamik içerik sağlamak ya da kullanıcıyla iletişim kurmak için kullanılır. Şu anda kullanılan bütün tarayıcılar tarafından desteklenmektedir.

Javascript HTML kodları arasına yerleştirilen küçük kodlardır. Tek başına .js uzantılı JavaScript kodları da yazılabilir.

JavaScript, Java dilinin küçük bir versiyonu değildir. JavaScript kendi başına bir script dilidir. Genellikle HTML sayfalarında kullanıcı kontrolleri eklemek, formları kontrol etmek, hesaplamalar yapmak gibi işler için kullanılır.

JAVASCRIPT NEREDE KULLANILIR?

Web tasarımında script kullanımının iki farklı yöntemi vardır: istemci taraflı ve sunucu taraflı. HTML gömülü yazılan sayfalara istemci taraflı scriptlerin çoğu JavaScript kullanılarak yazılmaktadır. Bunlar da genellikle kullanıcı bilgilerinin doğru girilip girilmediği ya da kullanıcıya veri girişinde ya da menü kullanımında seçenekler sağlamaktır.

Server Side Programming	Client Side Programming
ASP	HTML
PHP	Javascript
JSP	Vbscript
Perl	CSS (stil)

<SCRIPT> ETİKETİ

Bir tarayıcının anlayacağı temel dil HTML'dir. JavaScript dilinin HTML belgesi içinde yazılabilmesi için <SCRIPT> etiketleri kullanılır.

<HTML>

<SCRIPT>

...

JavaScript kodu

...

</SCRIPT>

</HTML>

Örnek:

<HTML>

<SCRIPT>

document.write ("Merhaba");

</SCRIPT>

</HTML>

HTML BELGESİ İÇİNDE JAVASCRIPT YAZMAK

Bir Web sayfası organizasyonunda JavaScript'leri Web sayfasına eklemek için değişik yollar vardır:

- HEAD bölümünde
- BODY bölümünde
- BODY içinde URL olarak.

<HEAD> İçinde:

Bir HTML belgesinde HEAD bölümü içinde <SCRIPT> etiketi kullanarak istediğiniz JavaScript kodunu HTML belgesine ekleyebilirsiniz. Bu kodlar HTML sayfasının BODY kısmı yüklendikten sonra kullanıma hazır hale gelir.

Örnek:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
function AlertTest( )
{
alert("Dikkat!");
}
// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

<BODY> İçinde

HTML belgesinin BODY kısmında JavaScript kullanmak için iki yol vardır:

Doğrudan çalışacak kodlar ve kullanıcı tarafından çalışacak kodlar.

Doğrudan çalışacak JavaScript kodları <SCRIPT> etiketi kullanılarak HTML belgesinin istenilen yerine yazılır. Bu cümleler tarayıcı tarafından okunduğu anda yerine getirilir:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write("Son okunma tarihi" + document.lastModified)
//-->
</SCRIPT>
```

Kullanılan diğer bir yöntem ise kullanıcı tarafından bir olayın başlatılmasıyla bir JavaScript kodunun çalıştırılmasıdır. Örneğin bir düğmeye tıklamak gibi:

```
<FORM>
```

```
<INPUT TYPE="BUTTON" NAME="TestButton"
VALUE="Tıkla" onClick="AlertTest()">
</FORM>
```

Örnek:

```
<SCRIPT Language="JavaScript">
function Onceki() {
    if (xmldso.recordset.bof){
        alert("Dosya başındasınız.");
    }else{
        xmldso.recordset.movePrevious()
        if (xmldso.recordset.bof){
            xmldso.recordset.moveFirst();
            alert("Bu ilk kayıt.");
        }
    }
}
function Sonraki() {
    if (xmldso.recordset.eof){
        alert("Dosya sonundasınız.")
    }else{
        xmldso.recordset.moveNext()
        if (xmldso.recordset.eof){
            xmldso.recordset.moveLast();
            alert("Bu son kayıt.");
        }
    }
}
</SCRIPT>
```

<BODY> İçinde URL Olarak

JavaScript kodu URL olarak da kullanmak mümkündür:

```
<A HREF="javascript: alert('Son!')">Siteden Çık</A>
```

JAVASCRIPT DİLİNİN ÖZELLİKLERİ

JavaScript dilinin temel gramer yapısına bir bakalım:

İşletme Sırası:

HTML belgesinde JavaScript eklendiğinde, kodun işletimi sırasıyla yapılır. Ancak fonksiyon ve belli bir olaya (ONCLICK gibi) bağlı olan JavaScript kodlarının işletimi o anda yapılır.

Değişkenler:

Bir değeri içeren bellek adlarını ifade ederler.

İşleçler (Operators):

Değerler ya da değişkenler arasında toplama, çıkarma gibi işlemlerin yapılmasını sağlarlar. İşleçler +, *, /, - gibi işaretlerle kullanılır.

İfadeler (Expression):

Değişkenlerin, işleçleri ve deyimlerin bir araya gelerek oluşturdukları yapılara ifade derin. Örneğin: "Ucret = Gün * Yevmiye" formülü bir ifadedir.

Deyimler (Statements):

Deyimler belli bir komut ya da söz dizimi bileşenlerinin grubuna verilen addır. Örneğin bir komut ya da bir IF yapısı deyimini oluşturur:

```
if (toplam>20) {deyimler;} else {deyimler;}
```

Nesneler (Objects)

Kendi değerleri, özellikleri ve işlemleri olan bileşenlere nesne (object) denir. JavaScript dilinde çok sayıda yerleşik nesne vardır.

Fonksiyonlar (Functions)

Bir JavaScript fonksiyonu diğer dillerdeki bir fonksiyon yordam ya da bir alt yordam anlamına gelir. Bir fonksiyon bir dizi işlemi yerine getirir ve bir sonuç döndürür. Ayrıca fonksiyonlar, parametre olarak kabul edilen birçok değer kabul ederler.

```
function adi( ) {  
deyim 1  
deyim 2  
...  
deyim N  
}
```

Case Sensitivity

JavaScript case sensitive (büyük küçük harf duyarlı) bir dildir. Değişkenler ve diğer öğeler küçük ya da büyük yazılabilirler. Ancak A değişkeni ile a değişkeni birbirinden farklıdır.

```
<SCRIPT LANGUAGE="JavaScript">  
<!--  
test= "küçük harf";  
TEST= "BÜYÜK HARF";  
document.writeln("test = "+test+"<BR>");  
document.writeln("TEST = "+TEST+"<BR>");  
/-->  
</SCRIPT>
```

Açıklamalar

Bütün programlama dillerinde olduğu gibi, kod yazarken açıklama yazmak için kullanabileceğimiz bir karakter vardır. JavaScript'te bu <!-- ve //-->. işaretlerinin arasındadır.

<!--bu bir açıklamadır //-->.

JavaScript kodu diğer programlama dillerinde olduğu gibi deyimlerden (statements) oluşur. Bu deyimler atama, değerleri karşılaştırma ve işlemlerin yapılması için komutlar anlamına gelmektedir.

DEĞİŞKENLER VE VERİ TÜRLERİ

Değişkenler verileri saklarlar ve onlara erişimi sağlarlar. Bir değişken bir değeri işaret eder. Örneğin Ucret adlı değişkene bir kişinin ücret bilgisi konur. Değişkenler adlarıyla işaret edilirler. Bir JavaScript değişkeni belli kurallara uymak zorundadır:

Kapsam

Bir değişkene bir değer verildiğinde bu değer ne kadar zaman geçerli olacak? Bu sorunun yanıtı kapsam (scope) kavramını ortaya çıkarır. Bir değişkenin kapsamı global ya da lokal olabilir. Bir global değişkene sayfa üzerinde herhangi bir JavaScript'ten erişilebilir. Lokal değişken ise değişerin atandığı yordam içinde geçerlidir.

Bir değişkene kendisine atama yaparak değer verilir:

```
Ucret= 1000000;
```

Bir fonksiyon yazıyorsanız, yerel bir değişken yaratmak istersiniz. Bu yerel değişkenler “var” sözcüğüyle tanımlanır:

```
function YeniFonksiyon()  
{ var dongu=1;  
  toplam=0;  
  ...deyimler...  
}
```

Yukarıdaki örnekte döngü adlı değişken YeniFonksiyon için yerel, toplam değişkeni ise tüm sayfa için genel (global) olacaktır.

Değişkene bir değer atanır. Bu herhangi bir tür veri olabilir. Bununla birlikte JavaScript, verinin değişik türlerde olmasını bekler. Verinin türüne bağlı olarak değer üzerinde belli işlemler yapılabilir ya da yapılamaz. Örneğin iki string değer üzerinde aritmetik işlem yapılamaz.

Değişken Türleri	
Numbers	Tam ve ondalıklı sayılar içerir. Pozitif, 0 ya da negatif olabilir.
Booleans	True (Doğru) ya da False (Yanlış)
Strings	Karakter bilgi içeren bir değişkendir.
Objects	nesne= new Object();
Null	Değersiz
Undefined	Bir değişken yaratıldıktan sonra içerdiği tanımsız değer.

İşleçler (Operators)

İşleçler, iki ya da daha fazla değer üzerinde işlem yapılmasını sağlar. JavaScript içinde aritmetik ve hesaplama işleçleri olmak üzere iki tür işleç kullanılır:

İşleçler

+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	MOD. Bölmede kalanı verir. (10 % 3 işlemi 1 değerini verir.)
++	Tek artırım. Değer 1 artırılır.
--	Tekli azaltma.
-	Negatifleştirme: İşlenenin negatif değerini döndürür.

Karşılaştırma

Karşılaştırma işleci, iki ya da daha çok değeri birbiriyle karşılaştırarak True ya da False olarak mantıksal bir değer döndürür. İşlenenler sayısal ya da karakter değer olabilir.

Karşılaştırma işleçleri	
==	İşlenen değerler eşitse True döndürür.
!=	İşlenen değerler eşit değilse True döndürür.
>	Soldaki değer sağdakinden büyükse True döndürür.
>=	Soldaki değer büyük ya da eşitse True döndürür.
<	Soldaki değer sağdaki değerden büyükse True döndürür.
<=	Soldaki değer sağdaki değerden küçük ya da eşitse True döndürür.

İkili İşleçler

İkili işleçler birden çok karşılaştırma işlemini tek bir koşul ifadesi olarak birleştirirler. Örneğin VergiDurum =1 AND Ucret > 1000 ise gibi.

İkili işleçler.	
&&	"And" İki ifade de doğruysa True döndürür.
	"Or" İfadelerden birisi doğruysa True döndürür.

!	"Not" Negatifleştirme işlemi doğruysa True döndürür.
---	--

Atama Deyimi

Atama deyimi (=), bir değişkene bir değer atanmasını sağlar. Değişkenlere türlerine ve tanımlamalarına uygun olan herhangi bir değer atanabilir.

= Değerler değişkenlere sağdan atanırlar.

Örnek: toplam=100;

Örnek: toplam=(ucret+vergi+ekkazanc)

+=

(ayrıca -=, *=, /=) Sağdaki değeri soldaki değişkene atar:

Örneğin: toplam+=ekkazanc (ekkazanc değerini toplam değerine ekler ve sonucu toplam değişkenine atar.)

JAVASCRIPT ÖRNEKLERİ

Örnek 1


```
<script language = "javascript">
    alert ( "Sayfama Hoş Geldiniz" ) ;
</ script>
```

Örnek 2:

```
<script language = " javascript">
    var isim = prompt ( "Lütfen Adınızı Giriniz", "" ) ;
    alert ( "Sayfama Hoş Geldin : "+isim ) ;
< / script>
```

Örnek 3 :

```
<script language = "javascript ">
    var isim=prompt ( "Lütfen İsminizi Giriniz", "" );
    var kitap=prompt ( "Sayın: " +isim+ " Kaç Kitap İstiyorsunuz", "Rakam ile");
    var cd=prompt("Sayın: " +isim+ "Kaç Cd İstiyorsunuz", "Rakam ile");
    var tutar=(kitap*35000000) + ( cd*1250000);
    alert ( "Sayın " +isim+ " Toplam Borcunuz : " +tutar+ " TL'dir")
</ script>
```

 sayfaya yazması için alert yerine document.write yazılır.

Örnek : 4

```
<script language=javascript>
    window.open ( "js3.htm" ) ;
</script>
```

Sayfa yüklenince farklı bir sayfa açılacaktır.

Örnek : 5

```
<script language = "javascript">
```

```
        window.open ("http://iibf.erciyes.edu.tr", "", "fullscreen");
    </ script>
```

Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi'nin giriş sayfası tam ekran olarak açılacaktır.

Örnek 6:

```
<script language = "javascript">
function kapat ( )
{
    window.open ("script6.htm")
}
</ script >
<body onload = 'kapat ()'>
```

Sayfa kapatılınca tekrar açılacaktır.

Örnek 7: Bu örnek < body> içine yazılmalı

```
<form name = form1>
    <select name = gun>
        <script language = "java script">
            for( i = ; i <=100 ; i ++ )
            {
                document.write ("<option>"+i+"</option>");
            }
        </script>
    </select>
</form>
```

Örnek 8: < head > içerisine

```
< script language = " javascript">
    window . status = " Bu site Ali AK tarafından tasarlanmıştır";
</script>
```



Durum çubuğunda bu site Ali AK tarafından tasarlanmıştır, yazar

KONTROL YAPILARI (IF)

```
If( şart ) {
    ...
} else {
    ...
}
```

Örnek 9:

```
<script language = " javascript">
    var isim = prompt ( " Lütfen adınızı giriniz ", "");
    var yaş =prompt ("sayın : "+ isim + "lütfen yaşınızı giriniz "; "rakam ile");
    if (yaş<=18)
    {
        alert("Yaşınız 18'den küçük " + isim );
    } else {
        alert ("Yaşınız 18'den büyük.");
    }
}
```


Örnek 10:

```
<script language = " javascript">
    var tarih =new date ( ) ;
        gun = tarih . getDate ( ) ;
        gun = tarih . getDay ( ) ;
        ay = tarih .getMont( ) ;
        yıl = tarih . getYear ( ) ;
    alert ( Bugün : " + gun + " / " + (ay+1) + " / " + yıl )
```



mesaj olarak sistem tarihine bağlı bugün 11/01/2003 yazar alert yerine

```
Windows . status =" bugün;" +gun+"/" + (ay+1) + "/" + yıl)
```



Yazılırsa durum çubuğunda çıkar

```
Gunler = new Arsay ( ) ;
Gunler [ 0] = "Pazar";
Gunler [1] = "Pazartesi";
...
Gunler [6] = " Cumartesi";
Aylar = new Arsay ( ) ;
Aylar [0] = " Ocak"
...
Aylar [12]=" Aralık
```

```
Takvim =gun + " " + aylar [ay] + " " + yıl + " , " + gunler [gun2]
Document.write (takvim)
```