

Web Editörü

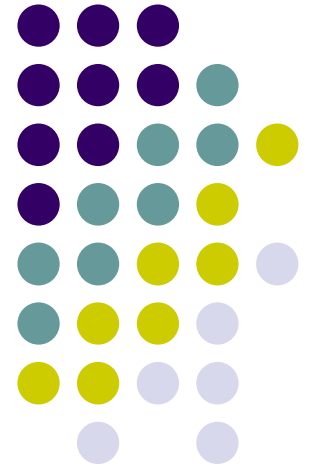
7 - JAVASCRIPT



Öğr.Gör. Serkan KORKMAZ

Harran Üniversitesi
Birecik Meslek Yüksekokulu

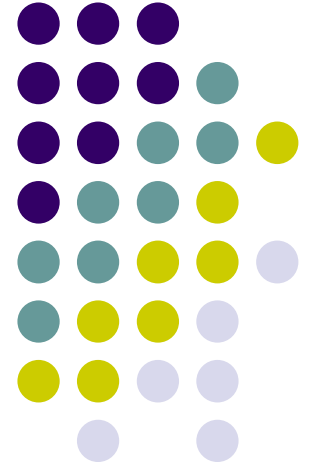
Mart 2022



GİRİŞ

JavaScript web sayfaları için yeni bir script dilidir.

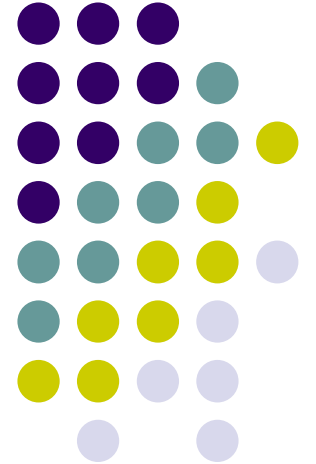
Özelliđi, HTML sayfalarının içine yerleřtirilebilmesidir.



GİRİŞ

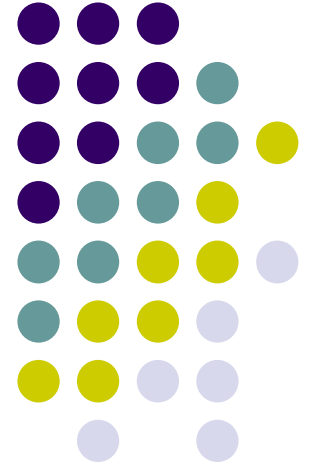
Netscape Navigator 2.0 ile birlikte geliştirilen JavaScript dili **istemci taraflı** (client-side) script dilidir.

Web sayfalarında **dinamik içerik sağlamak** ya da **kullanıcıyla iletişim kurmak** için kullanılır.



GİRİŞ

İlk başta yalnızca Netscape tarayıcıları için tasarlanan JavaScript, daha sonra diğer tarayıcılar tarafından da desteklenen genel bir scripting dili haline gelmiştir.

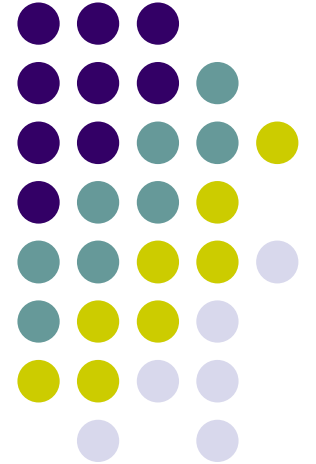


GİRİŞ

İPUCU:

HTML kodları arasına yerleřtirilen küçük kodlardır.

Tek başına **.js** uzantılı JavaScript kodları da yazılabilir.

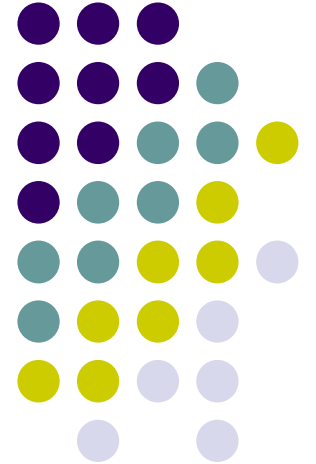


GİRİŞ

JavaScript, Java dilinin bir Light versiyonu değildir.

JavaScript kendi başına bir **script dilidir**.

Genellikle HTML sayfalarında **kullanıcı kontrolleri eklemek**, **formları kontrol etmek**, **hesaplamalar yapmak** gibi işler için kullanılır.



JAVASCRIPT



- Java ile JavaScript oldukça fazla karıştırılmaktadır.
- **Java**, Sun firması tarafından Pascal ve Delphi dillerinden esinlenerek yazılmış **bir programlama dilidir**.
- Sonuçta işletim sistemlerinden bağımsız bir program elde edersiniz.
- Yani exe veya com uzantılı dosya oluşturur.
- Fakat JavaScript bu tür bir programlama dili değildir.
- **Yorumlanması için bir tarayıcıya ihtiyaç duyar**.

JAVASCRIPT



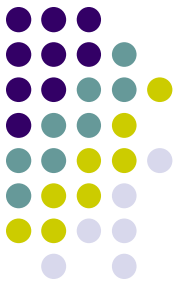
- Bu yüzden script dilidir.
- **Html dosyasını içine gömülüdür.**
- Sonuçta elinizde exe veya com uzantılı bir dosya yoktur.
- Javascript, Netscape firması tarafından C dilinden esinlenilerek yazılmıştır.
- Yazılıma amacı Html'in sahip olmadığı bazı özelliklerin web sayfalarında kullanılmak istenmesidir.
- Ziyaretçi ile etkileşim gibi önemli unsurlarda Html'in eksiklerini tamamlayıcı unsurlara ihtiyaç duyulması sonucunda Netscape firması bu konuya ağırlık vererek JavaScript script dilini internet ortamına kazandırmıştır.

JAVASCRIPT



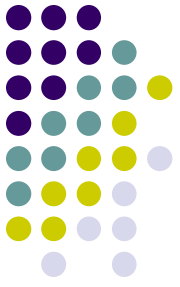
- Netscape ve Internet Explorer tarayıcılarının JavaScript kodundaki anlayışları farklıdır.
- Nedeni ise, Netscape JavaScript dilini hazırladığında Microsoft'un bu dilin özelliklerini veya yazılım tarzını tam anlamıyla Internet Explorer'a eklemeyip kendi yazım kurallarını belirlemesidir.
- Bu yüzden JavaScript kodu yazarken bu **iki tarayıcı özelliklerini de göz önünde bulundurmalıyız.**

NEREDE KULLANILIR?



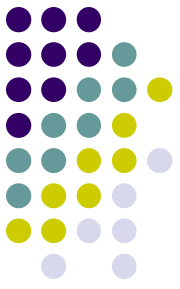
- JavaScript kullanımının iki ana alanı vardır:
 - istemci taraflı script yazmak.
 - sunucu taraflı script yazmak.
- Şu anda HTML sayfalara gömülü yazılan scriptlerin çoğu JavaScript kullanılarak yazılmaktadır.
- Bunlar da genellikle **kullanıcı bilgilerinin doğru girilip girilmediği** ya da **kullanıcıya veri girişinde** ya da **menü kullanımında seçenekler sağlamaktır.**
- Sunucu taraflı JavaScript, aynı PHP ve ASP gibi çalışmaktadır.

Sunucu Tarafli JavaScript



- Sunucu tarafli JavaScript (SSJS);
- Temel JavaScript'e ek olarak nesnelere ve islevler katililarak veritabanlarina erisim, e-mail gonderme ve diger islemlerin yapilmasi saglanir.
- SSJS, veritabanli temelli Web uygulamalarinin yaratilmasini saglar.

JAVASCRIPT



- Javascript kodlarını yazmak için Windows kullanıcıları için Not Defteri yeterlidir.
- JavaScript kodları `<script>` etiketi ile başlar, `</script>` etiketi ile biter.
- `<script>` etiketi, JavaScript'i anlamayan eski sürüm tarayıcıların bu kısmı geçmeleri içindir.

JAVASCRIPT

- Genellikle yazım tarzı aşağıdaki şekildedir:

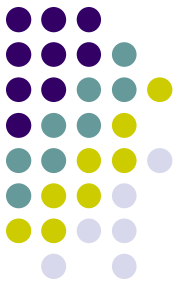
```
<script>
```

```
<!--
```

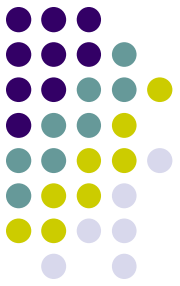
```
JavaScript kodları
```

```
-->
```

```
</script>
```



<SCRIPT> ETİKETİ



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

```
<HTML>
```

```
...
```

```
<SCRIPT>
```

```
...
```

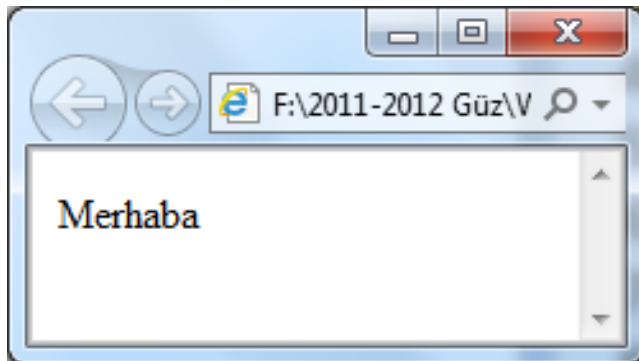
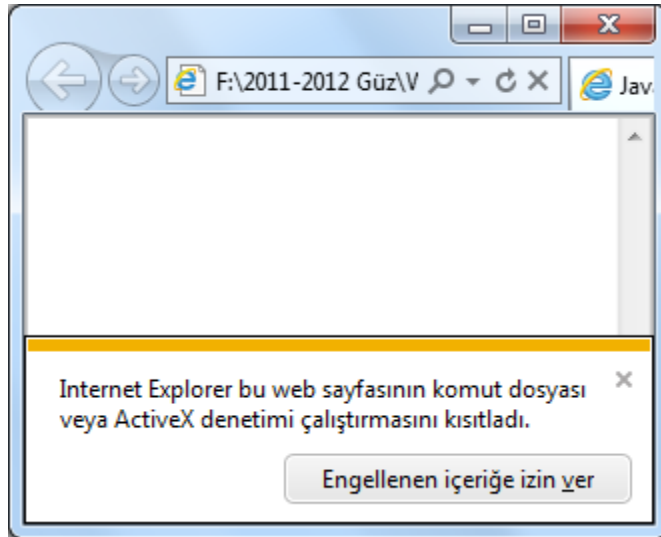
```
JavaScript kodu
```

```
....
```

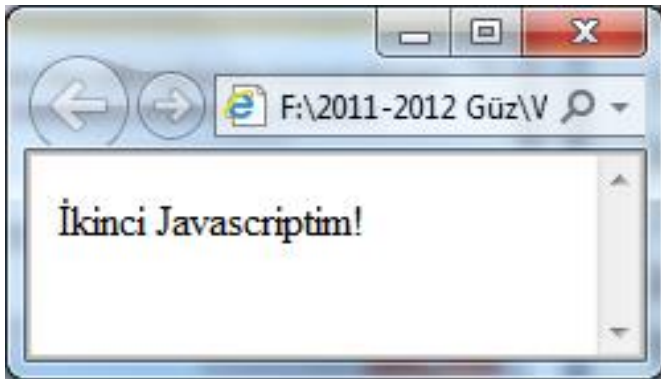
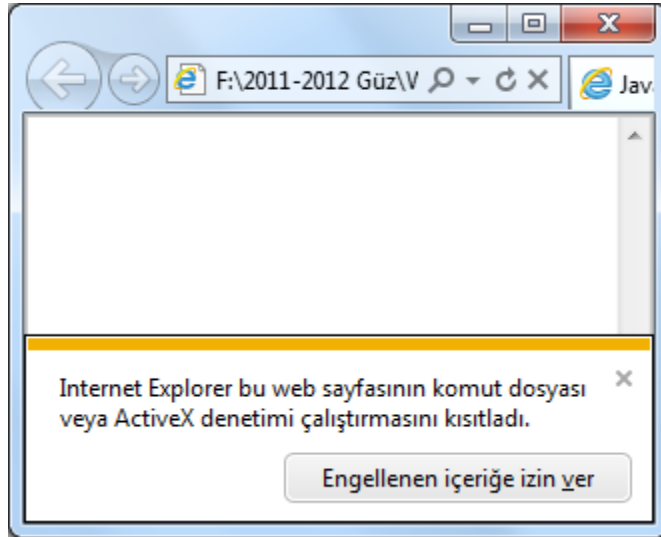
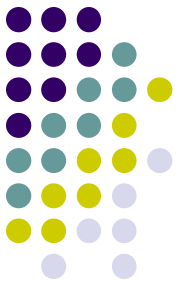
```
</SCRIPT>
```

```
...
```

```
</HTML>
```

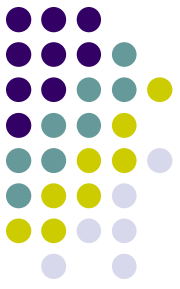


```
file:///F:/2011-2012%20G%FCz/Web%20Tasarı...
Dosya  Düzenle  Biçimlendir
1  <HTML>
2  <head>
3  <title>JavaScript</title>
4  <head>
5  <body>
6    <SCRIPT LANGUAGE="JAVASCRIPT">
7      document.write ("Merhaba");
8    </SCRIPT>
9  </body>
10 </HTML>
```



```
file:///F:/2011-2012%20G%FCz/Web%20Tasarımının%20T...
Dosya  Düzenle  Biçimlendir
1  <html>
2  <head>
3  <title>İkinci javascript</title>
4  </head>
5  <body>
6  <script language="Javascript">
7  document.write("İkinci Javascriptim!")
8  </script>
9  </body>
10 </html>
```

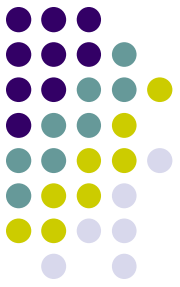

JAVASCRIPT AÇIKLAMA SATIRLARI



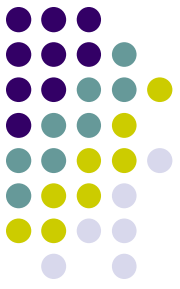
- İyi bir programcı kod satırlarında açıklama yapar.
- Bu satır şu işlemi gerçekleştiriyor gibi açıklayıcı bilgiler yazar kodlarının yanına.
- JavaScript'te bu tür açıklamalar `//` ile başlar ve `//` ile biter.
- Eğer açıklamanız bir satırdan fazla ise `/*` ile başlar `*/` ile biter.

JAVASCRIPT AÇIKLAMA SATIRLARI

```
// bu satır kullanılacak değişkenlerin tanımlanması  
/* açıklama satırı 1  
açıklama satırı 2  
açıklama satırı 3 */
```



JAVASCRIPT



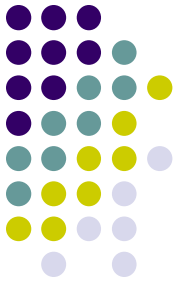
- JavaScript kodları Html kodların arasında yer alır.
- Veya uzantısı js olan dosyalarda saklanarak yine Html içerisinden çağırılır.
- Java Appletleri gibi Html'den ayrı bir unsur değildir.
- Javascript Html'in bir parçasıdır.
- Kullanılacak yere göre Html'in içerisinde kullanılır.
- Fakat genelde <head> </head> etiketleri arasında kullanılır.

JAVASCRIPT



- Javascript kodları bittiğinde elinizde asla kendi başına çalışan uzantısı exe veya com olan bir dosya olmaz.
- Her zaman için tarayıcı tarafından yorumlanması gerekir.
- Yorumlanması demek Javascript kodunun çalışması anlamındadır.
- Nesne ve buna uygulanan olaylar ile ilgili bir takım görevleri vardır. Javascript kullandığı her unsur nesne olarak algılar.
- Siz bu nesnelere tıklamak, üzerine gelmek, üzerinde çıkmak gibi olaylar ile çalıştırırsınız ki bu da Javascript'in ziyaretçi ile etkileşmesi demektir.

HTML BELGESİ İÇİNDE JAVASCRIPT YAZMAK



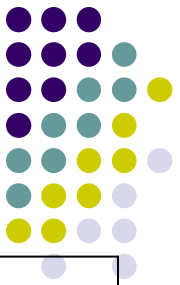
- Bir Web sayfası organizasyonunda JavaScript'leri Web sayfasına eklemek için değişik yollar vardır:
- HEAD bölümünde
- BODY bölümünde
- BODY içinde URL olarak.

<HEAD> İÇİNDE



- 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.

<HEAD> İÇİNDE



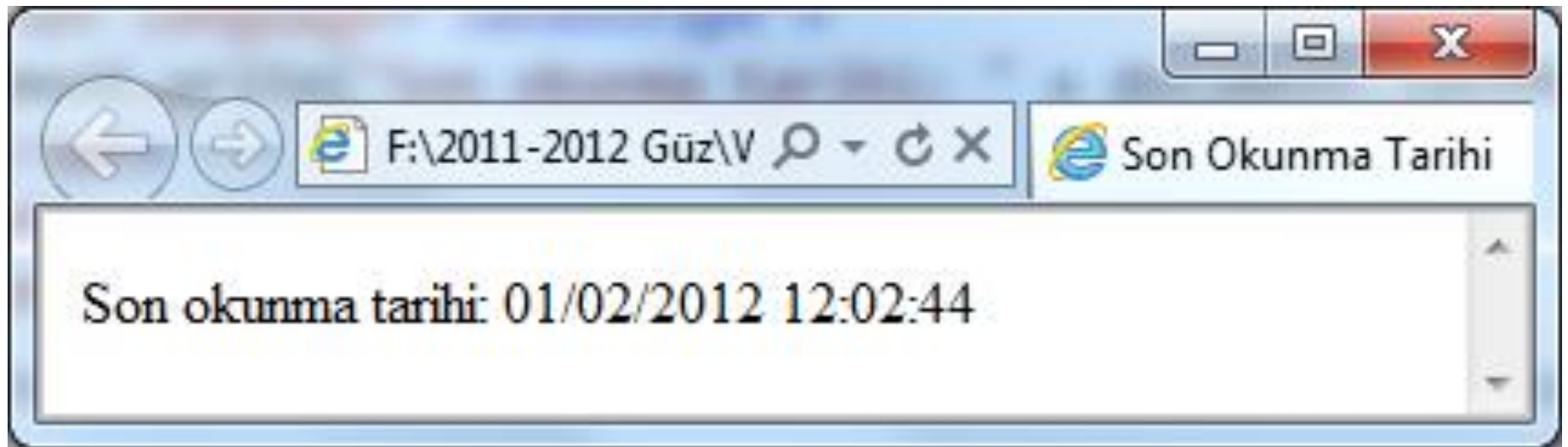
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
function AlertTest( )
{
alert("Dikkat!");
}
-->
</SCRIPT>
</HEAD>
<BODY>
***** buraya yazılacak *****
</BODY>
</HTML>
```

<BODY> İÇİNDE

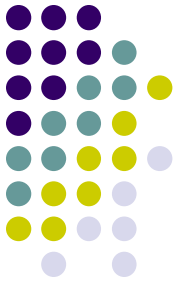


- HTML belgesinin BODY kısmında JavaScript kullanmak için iki yol vardır:
- Doğrudan çalışacak kodlar ve kullanıcı tarafından çalışacak kodlar:
- Doğrudan çalışacak JavaScript kodları <SCRIPT> etiketi kullanılarak HTML belgesinin istenilen yerine yazılır.
- Bu cümleler tarayıcı tarafından okunduğu anda yerine getirilir:


```
file:///F:/2011-2012%20G%FCz/Web%20Tasarımının%20Temelleri/UYGULAMA/12-javascript/js3.ht...
Dosya  Düzenle  Biçimlendir
1 <html>
2 <head>
3 <title>Son Okunma Tarihi</title>
4 </head>
5 <body>
6 <script language="Javascript">
7 document.write("Son okunma tarihi: " + document.lastModified)
8 </script>
9 </body>
10 </html>
```

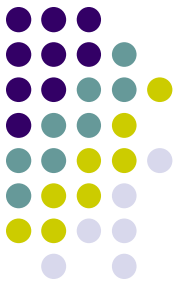


<BODY> İÇİNDE

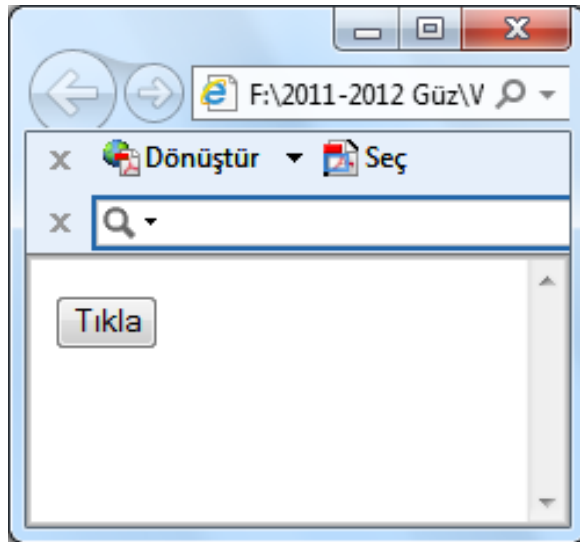


- Daha çok kullanılan diğer bir yöntem ise;
- kullanıcı tarafından bir olayın başlatılmasıyla bir JavaScript kodunun çalıştırılmasıdır.
- Örneğin bir düğmeye tıklamak gibi:

<BODY> İÇİNDE



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



```
file:///F:/2011-2012%20G%FCz/Web%20Tasarımının%20T...
Dosya  Düzenle  Biçimlendir
1 <HTML>
2 <HEAD>
3 <SCRIPT LANGUAGE="JavaScript">
4 <!--
5 function AlertTest( )
6 {
7 alert("Dikkat!");
8 }
9 // -->
10 </SCRIPT>
11 </HEAD>
12 <BODY>
13 <FORM>
14 <INPUT TYPE="BUTTON" NAME="TestButton"
15 VALUE="Tıkla" onClick="AlertTest()">
16 </FORM>
17 </BODY>
18 </HTML>
19
```

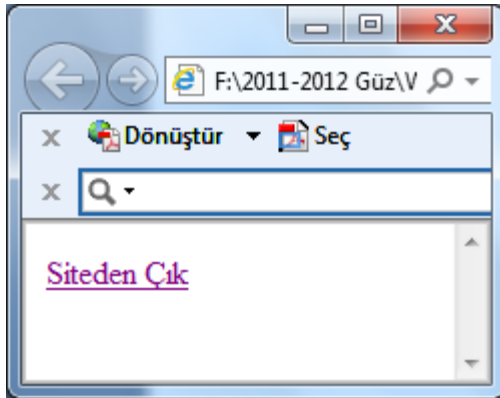
<BODY> İÇİNDE URL OLARAK



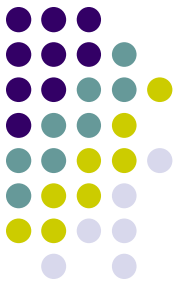
- JavaScript kodu URL olarak da kullanmak mümkündür:
- `Siteden Çık`



```
file:///F:/2011-2012%20G%FCz/Web%20Tasarımının...  
Dosya Düzenle Biçimlendir  
1 <HTML>  
2 <HEAD>  
3 <TITLE>JAVASCRIPT</TITLE>  
4 </HEAD>  
5 <BODY>  
6 <A HREF="javascript: alert('Son!')">  
  Siteden Çık</A>  
7 </BODY>  
8 </HTML>  
9
```

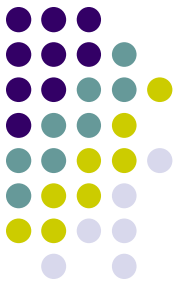


JAVASCRIPT



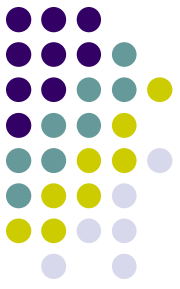
- **İşletme Sırası:**
- HTML belgesinde JavaScript eklendiğinde, kodun işletimi sırasıyla yapılır.
- Ancak fonksiyon ve belli bir olaya (ONCLICK gibi) bağlı olan JavaScript kodlarının işletimi o anda yapılır.
- **Değişkenler:**
- Bir değeri içeren bellek adlarını ifade ederler.
- **Operatörler:**
- Değerler ya da değişkenler arasında toplama, çıkarma gibi işlemlerin yapılmasını sağlarlar.
- Operatörler +, *, /, - gibi işaretlerle kullanılır.

JAVASCRIPT



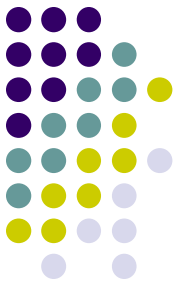
- **İfadeler (Expression):**
- Değişkenlerin, işleçleri ve deyimlerin bir araya gelerek oluşturdukları yapılara ifade denir.
- Örneğin: "Ucret = Gün * Yevmiye" formülü bir ifadedir.
- **Deyimler (Statements):**
- Deyimler belli bir komut ya da söz dizimi bileşenlerinin grubuna verilen addır.
- Örneğin bir komut ya da bir IF yapısı deyimi oluşturur:
- `if (toplam>20) {deyim1;} else {deyim2;}`

JAVASCRIPT



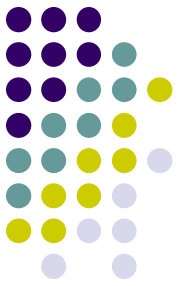
- **Nesneler (Objects):**
- Kendi değerleri, özellikleri ve işlemleri olan bileşenlere nesne (object) denir.
- JavaScript dilinde çok sayıda yerleşik nesne vardır.
- **Fonksiyonlar (Functions):**
- Bir JavaScript fonksiyonu diğer dillerdeki bir fonksiyon yordam ya da bir alt yordam anlamına gelir.
- Bir fonksiyon bir dizi işlemi yerine getirir ve bir sonuç döndürür.
- Ayrıca fonksiyonlar, parametre olarak kabul edilen birçok değer kabul ederler.

JAVASCRIPT



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

JAVASCRIPT

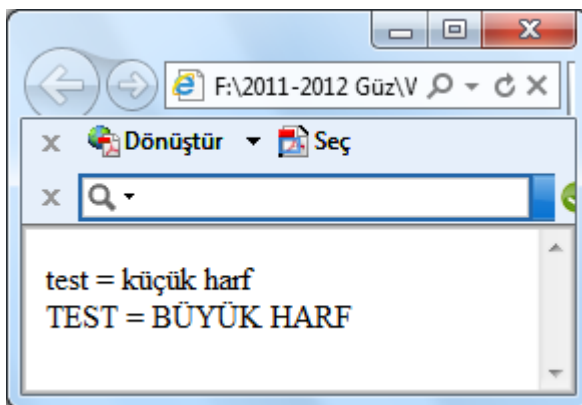


- **Case Sensitivity:**
- JavaScript case sensitive (**büyük küçük harf duyarlı**) bir dildir.
- Değişkenler ve diğer öğeler küçük ya da büyük yazılabilirler.
- Ancak A değişkeni ile a değişkeni birbirinden farklıdır.

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



```
file:///F:/2011-2012%20G%FCz/Web%20Tasarımının%20Tem...
Dosya  Düzenle  Biçimlendir
1  <HTML>
2  <HEAD>
3  <TITLE>JAVASCRIPT</TITLE>
4  </HEAD>
5  <BODY>
6  <SCRIPT LANGUAGE="JavaScript">
7  <!--
8  test= "küçük harf";
9  TEST= "BÜYÜK HARF";
10 document.writeln("test = "+test+"<BR>");
11 document.writeln("TEST = "+TEST+"<BR>");
12 //-->
13 </SCRIPT>
14 </BODY>
15 </HTML>
16
```

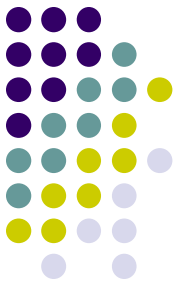


JAVASCRIPT

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

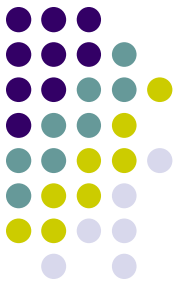


DEĞİŞKENLER



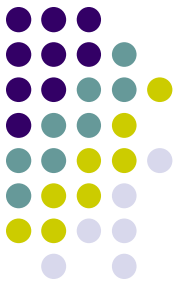
- Değişkenler verileri saklarlar ve onlara erişimi sağlarlar.
- Bir değişken bir değeri işaret eder.
- Örneğin Ucret adlı değişkene bir kişinin ücret bilgisi konur.
- Değişkenler adlarıyla işaret edilirler.
- Bir JavaScript değişkeni belli kurallara uymak zorundadır:

DEĞİŞKENLER



- Değişkenler Javascript'te ve diğer programlama dillerinde olduğu gibi bilgi depolamak bu bilgiyi kullanmak amacıyla kullanılırlar.
- Değişkenler "**var**" komutu ile oluşturulurlar.
- Karakter olarak kullanıldıklarında işlem yapılamazlar.
- Nümerik olarak kullanıldıklarında ancak işlem yapabilirler.
- Kullanımına bir örnek verelim.
- *var sayi;*
- *var sayi1=10;*
- *var yazi1="10";*

DEĞİŞKENLER



- Burada birinci satırdaki "sayi" değişkeni script kodunun herhangi bir yerinde kullanılmak üzere oluşturulmuştur.
- İkinci satırda "sayi1" adındaki değişkenin değeri hemen o satırda = ifadesinden sonra verilmiştir.
- Böyle değişken tanımı da yapılabilir.
- Üçüncü satırda ise değişkenin karakter ifadesi olarak kullanımını görüyoruz.
- Burada önemli olan karakter değişkenlerin alıntı " " ifadesinin arasında kullanılmasıdır.
- Her değişkenden sonra ; işareti konulmalıdır.
- Tarayıcı, bir başka komut satırına geçtiğini bu yol ile anlar.

DEĞİŞKENLER



- Şimdi değişkenlerle ilgili matematik işlemlerinin nasıl olacağını görelim.
- Bunu daha iyi açıklayabilmek için örnekler üzerinde çalışalım.

var sayi1=10;

var sayi2=20;

var sayi3=sayi1+sayi2;

- Birinci ve ikinci satırlarda değişkenler oluşturduk.
- Üçüncü satırdaki ise sayi3 değişkeni ile diğer iki değişkeni topladık.
- Burada önemli olan işlem yapmak istediğimizde değişken değerinin alıntı " " işaretlerinin arasına konmamasıdır.

DEĞİŞKENLER



- Şimdi de değişkenleri karakter olarak tanımladığımızda neler olduğuna bakalım.

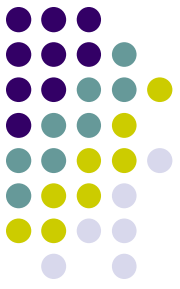
```
var sayi1="10";
```

```
var sayi2="20";
```

```
var sayi3=sayi1+sayi2;
```

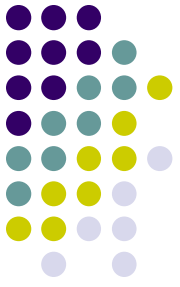
- Bir önceki örnekten farklı olarak değişken değerlerinin alıntı işaretleri içerisinde yazdık.
- Eğer sayi3 adlı değişkeni tarayıcıda bastırırsak göreceğimiz ifade 1020 ifadesidir.
- Yani tarayıcı karakter olarak tanımladığımız değişkenleri ardada ekledi.
- Burada unutulmaması gereken şey bunun sadece + işleminde geçerli olmasıdır. Diğer işlem türlerinde bu tür bir sonuç alınamaz.

DEĞİŞKENLER



- Değişkenlere ad verirken uymamız gereken kurallar;
- Değişken isimleri harf veya _ karakteri ile başlayabilir.
- Rakam kullanmak istersek 2. karakterden sonra kullanabiliriz.
- Yani değişkenin ilk karakteri rakam olamaz.
- Değişken isimlerine örnekler;
- *var url="webteknikleri"; doğru*
- *var _rakam=12; doğru*
- *var a1=123; doğru*
- *var 3uzler="üçüzler"; yanlış*

DEĞİŞKENLER



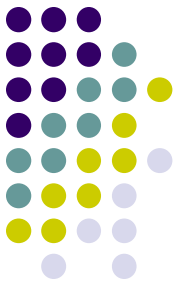
- Değişken tanımlarken bir veya birden fazla boşluk bırakmak tanımlama açısından herhangi bir sorun teşkil etmez.
- Değişken adı verirken kullandığımız harflerin büyük veya küçük olması bazı tarayıcılarda fark etmezken çoğu tarayıcıda farklı bir değişken anlamındadır.
- Yani;
- *var say=1;*
- *var Say=1;*
- Birçok tarayıcıda farklı değişkenler olarak algılanır.

Değişkenlerin İşlem Operatörleri İle Kullanımı



- Değişkenlere işlem yaptıracak operatörleri ve özelliklerini inceleyelim.
- Operatörleri birkaç kısma ayırarak inceleyelim;
- Aritmetik operatörler
- Karşılaştırma operatörleri
- Mantıksal operatörler
- Özel operatörler

Aritmetiksel Operatörler

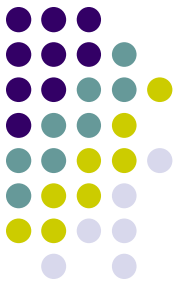


- Her zaman kullandığımız bu operatörler + , - , * , / , % 'dir.
- Değişkenlerin çeşitli aritmetik operatörlerle kullanımına bir örnek verelim.

```
var i=10;  
var j=11;  
var k=12;  
var m,n;  
m=i*j+k; // m=k+i*j;  
n=i*(j+k);
```

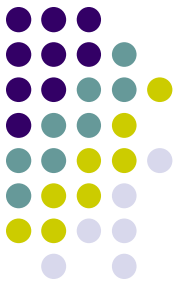
- İlk üç satırda değişkenlerimizi hem tanımlayıp hem de değer atadık.
- Dördüncü satırda ise m ve n değişkenlerini tanımlayıp değer atamadık.
- Diğer satırlarda ise m ve n değişkenlerinden istenen işlemleri tanımladık.
- Buna göre son iki satırın sonuçları farklıdır.
- Çünkü parantezlerin işlem önceliği vardır.

Aritmetiksel Operatörler



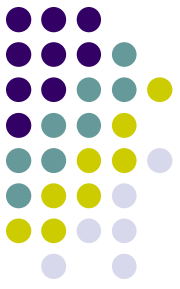
- Beşinci satırın cevabı $(10*11)+12 = 122$ şeklinde olacaktır.
- Son satırda ise sonuç $10*(11+12) = 230$ olacaktır.
- Diğer bir işlem operatörü olan % 'nin yaptığı işlemi şu şekilde anlatabiliriz.
- % operatörü % işaretinin solundaki değişkeni sağındaki değişkene böler ve kalanı verir.
- Örnek olarak;
- *var a=100; var b=9; var c=100%9;*
- Burada c değişkeninin değeri 100/9'un kalanı 1'dir.
- Yani c değişkeninin değeri 1 olacaktır.

Aritmetiksel Operatörler



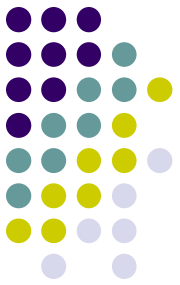
- Diğer $-$ (eksi) ve $/$ (bölme) operatörlerinin işlemleri kendilerine atanan çıkartma ve bölme işlemidir.
- Bu operatörlerin kısa kullanımı içinde Javascript bize kolaylık sağlar.
- Bu operatörleri sıralamak istersek;
- $+=$ $-=$ $*=$ $/=$ $\%=$ $++$ $--$
- $x+=y$ $x=x+y$
- $x-=y$ $x=x-y$
- $x*=y$ $x=x*y$
- $x/=y$ $x=x/y$
- $x\%=y$ $x=x\%y$
- $x++$ $x=x+1$
- $x--$ $x=x-1$

Aritmetiksel Operatörler

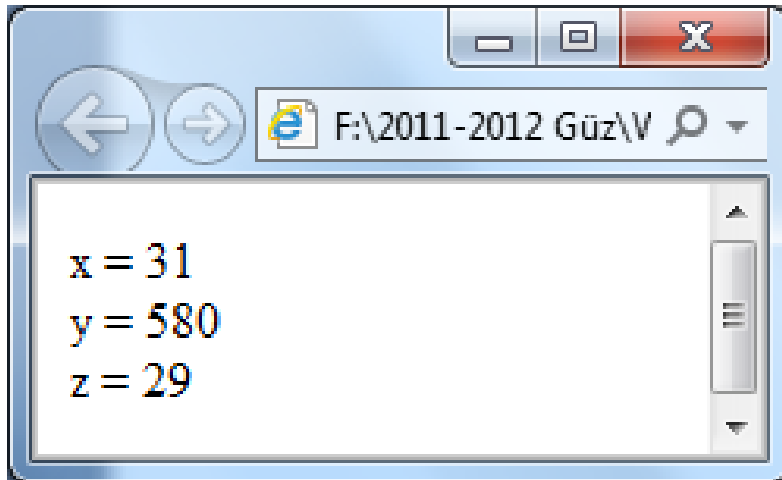


- Bu operatörlerin nasıl işlem yaptığını bir örnekte görelim.
- `var x,y,z;`
- `x=10;`
- `y=20;`
- `z=30;`
- `x++;`
- `x+=y;`
- `z--;`
- `y*=z;`
- Şimdi her zamanki gibi işlem satırlarının cevaplarını birlikte bulalım.
- `x++` satırı `x=x+1` işleminin yapılmasını söyler.
- Buna göre `x` değişkeni 11 değerini alır.
- İkinci satıra geldiğimizde `x+=y` satırı `x=x+y` işleminin yapılmasını söyler.
- Bir önceki satırdaki `x`'in değeri 11 idi. Böylelikle yeni `x`'in değeri $11+20=31$ olacaktır.
- Diğer satırda `z--` işlemi sonucunda `z`'nin değeri 29 olacaktır.
- Son satırda ise `y*=z` işlemi ile `y` değişkeni $20*29= 580$ değerini alacaktır.

Aritmetiksel Operatörler



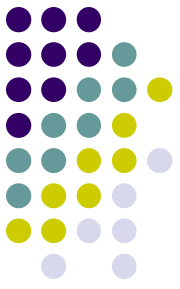
X	Y	Z	İŞLEM
10	20	30	x=10; y=20; z=30;
11	20	30	x++;
31	20	30	x+=y;
31	20	29	z--;
31	580	29	y*=z;



A screenshot of a text editor window showing the source code of the JavaScript script. The file path is file:///F:/2011-2012%20G%FCz/Web%20Tasarımı... The menu bar includes Dosya, Düzenle, and Biçimlendir. The code is as follows:

```
1 <HTML>  
2 <HEAD>  
3 <TITLE>JAVASCRIPT</TITLE>  
4 </HEAD>  
5 <BODY>  
6 <SCRIPT LANGUAGE="JavaScript">  
7 <!--  
8 var x,y,z;  
9 x=10;  
10 y=20;  
11 z=30;  
12 x++;  
13 x+=y;  
14 z--;  
15 y*= z;  
16 document.writeln("x = "+x+"<BR>");  
17 document.writeln("y = "+y+"<BR>");  
18 document.writeln("z = "+z+"<BR>");  
19 //-->  
20 </SCRIPT>  
21 </BODY>  
22 </HTML>  
23
```

Karşılaştırma Operatörleri



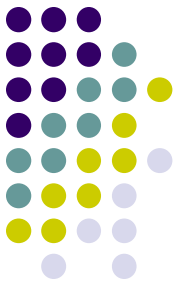
- Değişkenlerin birbirleri ile karşılaştırılmak istendiğinde kullanılan operatörlerdir.
- Bu operatörler ise;
- `==` operatörü iki değişkenin birbirine eşitliğini kontrol eder.
- `!=` operatörü iki değişkenin birbirine eşit olmadığı durumlarda kullanılır.
- `<` operatörü bilindiği üzere küçüktür operatörüdür. Soldaki değişkenin sağdakinde küçüklüğünü kontrol eder.
- `<=` soldaki değişkenin sağdaki değişkene küçük eşitliğini kontrol eder.
- `>` soldaki değişkenin sağdaki değişkene göre büyük olup olmadığını kontrol eder.
- `>=` soldaki değişkenin sağdaki değişkene büyük eşitliğini kontrol eder.

Mantıksal Operatörler



- İki değişkene bağlı karşılaştırmaların yapılmak istendiği durumlarda kullanılır.
- Operatörler `&&` , `||` , `!` operatörleridir.
- `&&` And (ve) operatörü iki değişkenin de değeri doğru olması istendiğinde kullanılır.
- `||` Or (veya) operatörü iki değişkenden en az birinin doğru olması durumu istediğinde kullanılır.
- `!` Not (değil) operatörü değişkenin değeri doğru ise yanlış , yanlış ise doğru olması istendiği durumlarda kullanılır.

Özel Karşılaştırma Operatörleri



- Bu operatör iki değişken arasında karşılaştırma yapmanın en sade ve kısa yoludur.
- Operatörün kullanım biçimi :
- `değişken_1 [karşılaştırma operatörü] değişken_2 ? değişken_3 : değişken_4`
- Bunu bir örnekle açıklayalım.
- Değişkenleri `var` ile tanımladığımızı farz ediyorum.
- Buna göre ;
- `a < b ? c : d`
- Yukarıdaki satırda yapılması istenen işlem;
- `a` değişkeninin `b` değişkeninden küçük olup olmadığı karşılaştırılıyor.
- Buna göre cevap doğruysa işlemin sonucu `c` değişkeninin değeri değilse `d` değişkeni oluyor.

Ekranaya Çıktı Ve Klavyeden Bilgi Giriş



- Html üzerinden klavye aracılığı ile ziyaretçiden bilgi almayı ve herhangi bir değişken vb. türde yazıların html'e nasıl yazdırılacağını göreceğiz.
- PROMPT ()

PROMPT



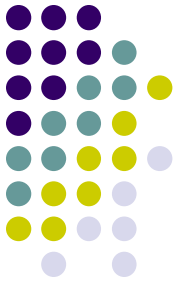
- PROMPT ()
- Hemen başlayalım.
- Ziyaretçiden bilgi alma iki tür JavaScript komutuyla gerçekleşir.
- Birisi Prompt yani bizim burada bahsedeceğimiz komut.
- Diğeri ise Form yoluyla bilgi alınması.
- Form yoluyla alınan bilgiler formun Html üzerinde yer alması yüzünden Prompt komutu ile birbirinden ayrılır.
- Prompt komutu ile Html sayfasından hariç bir pencere açılır.
- Alınmak istenen bilgi ziyaretçiye bu yol ile sorulur ve hemen altındaki boşluk yardımıyla cevap alınır.

PROMPT



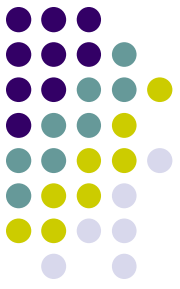
- *prompt* ("Sorulan soru" , "Cevap örneği");
- Bu komutun yorumlanması şu şekildedir.
- Html üzerinde Html'den bağımsız bir pencere aç (bu prompt komutu ile yapılır).
- İlk çift tırnak içerisinde olan kelime veya kelime grubu, pencerenin üst kısmında ve değiştirilemeyen kısımdır.
- Burada soru veya pencerenin niçin açıldığı ile ilgili bir açıklama verilir.
- İkinci çift tırnakta ise doldurulacak olan cevap satırının içinde seçili bir haldedir.
- Bu ise genel olarak cevap örneği olarak ziyaretçiye sunulur.
- Kullanılması zorunlu değildir.

PROMPT



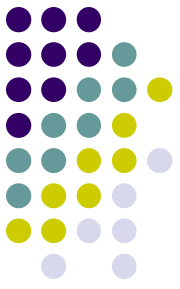
- prompt ("Tarayıcınızın türünü giriniz ?" ,"lütfen cevabı ie veya nn olarak veriniz");
- Şimdi kullanıcıdan nasıl bilgi alınacağını gördük fakat bu bilgiyi nasıl kullanabiliriz ?
- Bu sorunun cevabı prompt komutunu var ile bir değişkene atmak suretiyle kullanabiliriz olacaktır.
- Yani ;
- var tara=prompt ("Tarayıcınızın türünü giriniz ?" ,"lütfen cevabı ie veya nn olarak veriniz");
- Biz bu satır ile ne yapmış olduk?
- Ziyaretçiden prompt komutu ile tarayıcısı sorduk ve bunu var değişken tanımlama komutuyla tara değişkenine atadık.
- Ziyaretçiden aldığımız bu bilgi sonucunda tara değişkeni ya ie yada nn değerini alacaktır.

WRITE()



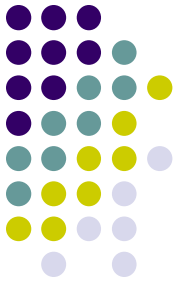
- Html dosyasına yazı yazdırma komutu ise **write** dır.
- Bu kodun yazım kurallarını inceleyecek olursak ;
- `document.write ("görüntülenmek istenenler" , değişken_ismi);`
- Kodu inceleyelim:

WRITE()



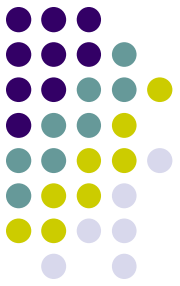
- `document.write ("görüntülenmek istenenler" , değişken_ismi);`
- Kodu inceleyelim:
- Javascript html üzerinde yazı yazmak istediğinde write komutunu tek başına kullanamaz.
- Bunun için document fonksiyoneli (yardımcısı manasında) ile birlikte kullanılır.
- `document.write` komutundan sonra parantez açılır.
- Daha sonra yazılmak istenen sıraya göre değişken ismi veya görüntülenmek istenenler yazılır.
- Değişkenler çift tırnak içerisinde yazılmazlar.
- Sadece görüntülenmek istenenler çift tırnak içerisinde yazılır.

WRITE()



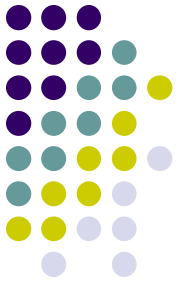
- Şimdi prompt komutu ile write komutunu birleştirerek bir kod hazırlayalım.
- Bu kodumuzda prompt aracılığıyla ziyaretçiye adını sorup ad değişkenine atıyoruz.
- Daha sonra bu değişkeni write komutu yardımıyla ziyaretçiye Merhaba diyoruz.
- Şimdi kodlara geçelim.

PROMPT & WRITE



```
<html>
<head>
<title>write örneği</title>
</head>
<body>
<script language="JavaScript">
<!-- -
//Kodları eski tarayıcılardan gizliyoruz
var isim = prompt ("İsminizi Giriniz " , "Küçük harf veya büyük harf
kullanabilirsiniz" );
document.write ("Merhaba " , isim , " !" );
// Saklamayı bitir
-->
</script>
</body>
</html>
```

PROMPT & WRITE



Explorer Kullanıcı Komut İstemi

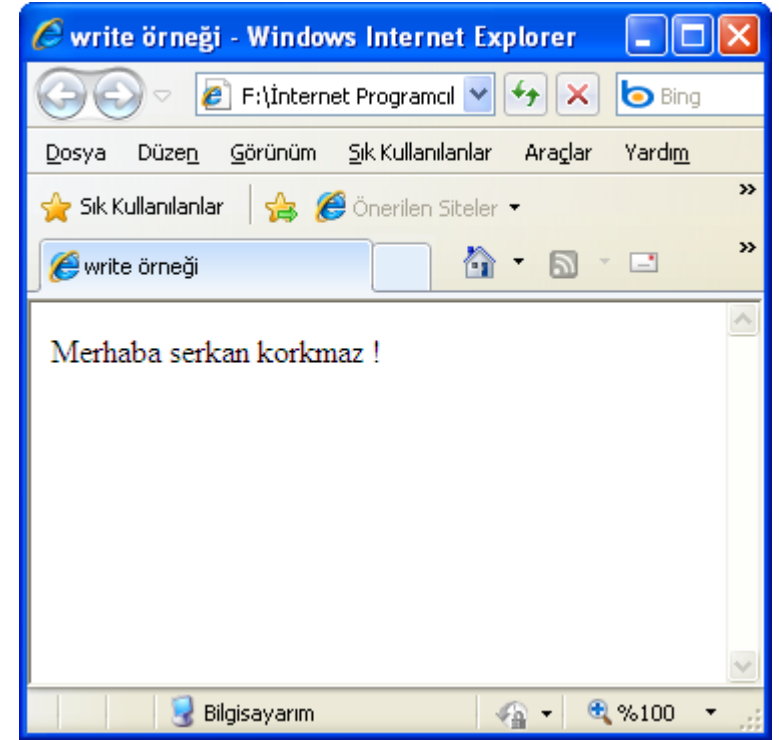
Kod Komut İstemi:

İsminizi Giriniz

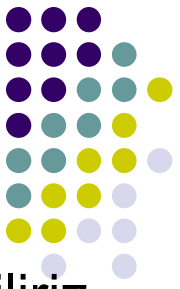
Explorer Kullanıcı Komut İstemi

Kod Komut İstemi:

İsminizi Giriniz

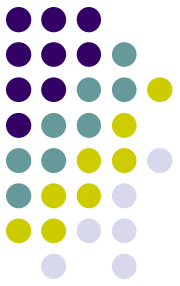


WRITE()



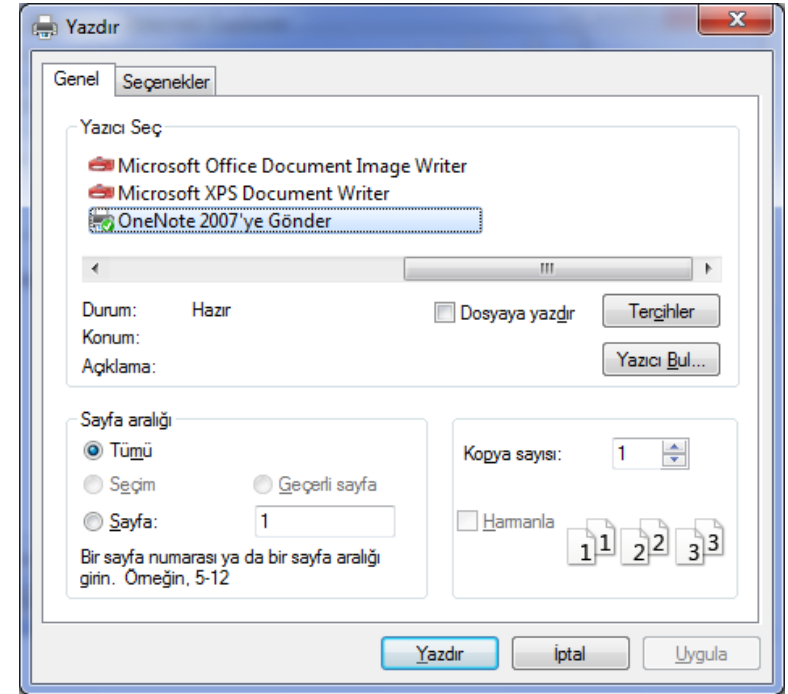
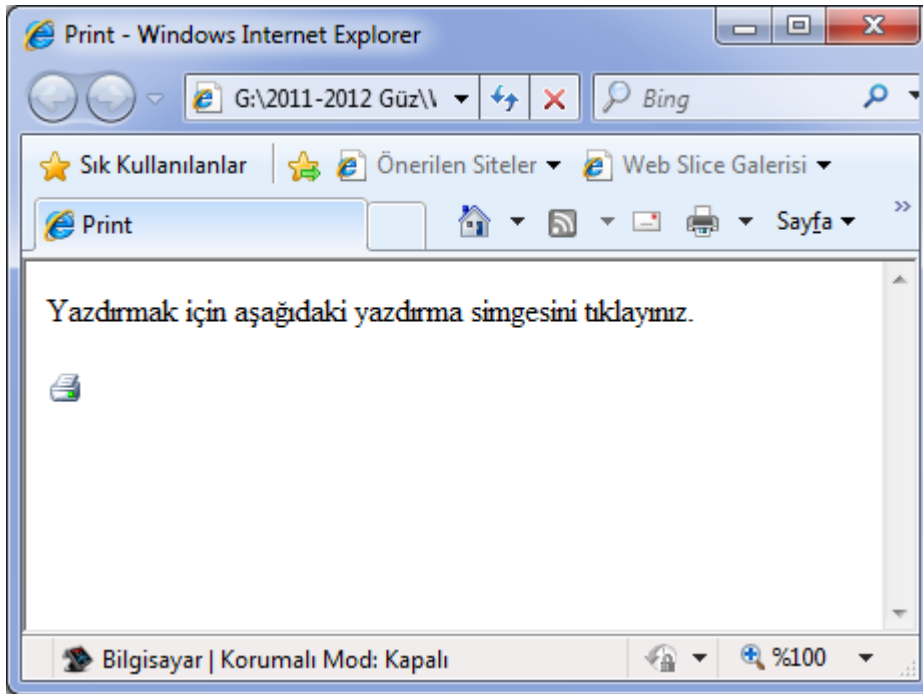
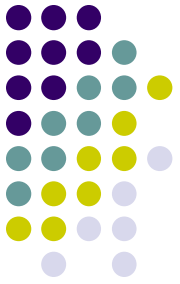
- Biz aslında Html'de font kurallarını kullanarak yazı da yazdırabiliriz.
- Hiçbir kural uygulamadığınızda tarayıcının standart (default) değerleri kullanılır.
- Örneğin, Merhaba dedikten sonra alınan ismin bir alt satırda görüntülenmesini istiyorsak bunu Javascript'e şu şekilde yaptırabiliriz
- (
 türünde Html etiketlerinin tümünü Javascript'e yaptırabilirsiniz).
- `document.write ("Merhaba" , "
" , isim)`

PRINT()

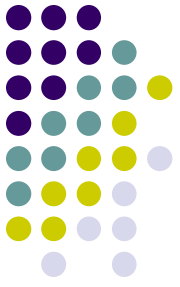


1. `<html>`
2. `<head>`
3. `<title>Print</title>`
4. `</head>`
5. `<body>`
6. `<p>Yazdırmak için aşağıdaki yazdırma simgesini tıklayınız.</p>`
7. ``
8. ``
9. ``
10. `</body>`
11. `</html>`

PRINT()

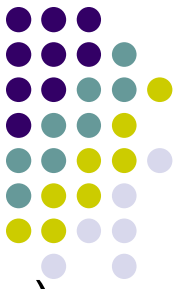


KOŞUL VE DÖNGÜLER



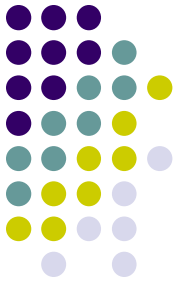
- Bilgisayarı bilgisayar yapan konular bunlardır.
- Çünkü hiçbir bilgisayar kendi kriterleriyle yorum yapamaz.
- Bizim koşullar ve döngüler ile verdiğimiz, önceden belirlenmiş kıstasları göz önünde bulundurarak gerekli seçimleri yapar.
- **Koşul Yapıları**
- Javascript'in en önemli özelliklerinden biri koşul yapılarıdır.
- Aslında bu konu sadece Javascript'in değil bilgisayarın da en önemli konusudur.
- Şimdi konunun inceliklerine bir göz atalım.

IF (EĞER)



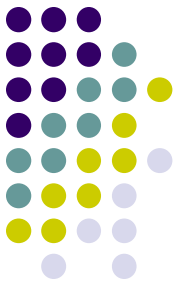
- Javascript'te çoğu dilde olduğu gibi koşul yapısının kodu If (eğer) komutudur.
- Yazılım şekli ise şu şekildedir.
- *If (a==b)*
- *//koşul doğru ise ilk satır işleme konulur*
- *// koşul doğru değilse ilk satırın altındaki komut satırı işleme konulur.*
- Şimdi kodumuzu biraz inceleyelim :

IF (EĞER)



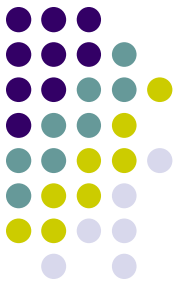
- Koşul komutu yani if ile işleme başlıyoruz.
- Daha sonra karşılaştırılacak değişkenler veya başka nesnelere parantez içerisinde sorgulanıyor.
- Dikkat ederseniz çift eşittir kullandık. Çünkü tek eşittir işareti değer atama işlemidir.
- Çift değişken ile koşul yapısı sağlanır.
- Eğer koşul doğruysa hemen altındaki satır işleme konulur.
- Eğer koşul yanlış ise ikinci satır işleme konulur.
- Yok ben koşul doğru ise 2 ve daha çok işlem yaptırmak istiyorsanız bunun cevabı yapılması istenen işlemlerin { } arasında yer almasıdır.
- Yani :

IF (EĞER)



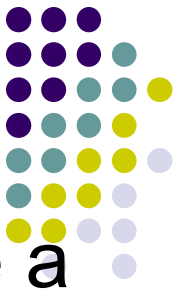
- *If (a==b)*
- *{*
- *// 1.işlem*
- *// 2. İşlem*
- *...*
- *...*
- *}*

IF .. ELSE (EĞER ... DEĞİLSE)



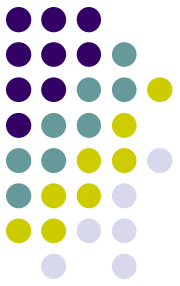
- Bu bölümde ise If koşul ifademize Else komutunu ekleyerek koşul yapısı örneklerle açıklanacaktır.
- *If (a==b)*
- *{*
- *// şunları şunları yap*
- *}*
- *else*
- *{*
- *//değilse şunları yap*
- *}*

IF (EĞER)



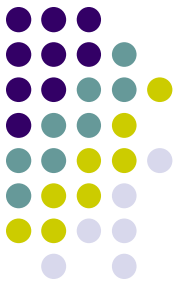
- Yani örnekten de anlaşıldığı gibi if koşulu ile a ile b nin eşitliği karşılaştırılıyor.
- Eğer doğruysa hemen altındaki kısım işleme konuluyor.
- Else ile yok değilse altındaki kısmı işleme koyuyoruz.

IF .. ELSE (EĞER ... DEĞİLSE)



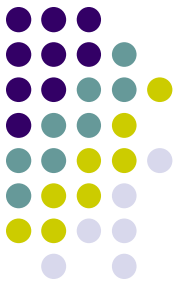
- *if (a==b)*
- *{*
- *//şunları yap*
- *}*
- *if (a==c)*
- *{*
- *//şunları yap*
- *}*
- *else*
- *{*
- *//şunları yap*
- *}*

IF (EĞER)



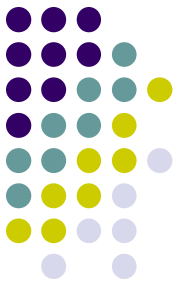
- Şimdi bu kod Javascript'e:
- a değişkeni b değişkenine eşitse normal olarak alt satırı işleme koymasını, eğer bu karşılaştırma yanlış ise altındaki işlemleri geçerek a'nın c'ye eşitliğini kontrol etmesini ve bu da değilse (else) alt satırdaki işlemleri devreye koymasını bildirir.
- Else yapısı genel olarak bir karşılaştırma sonucunda cevap yanlış ise diğer bütün durumlarda şu işi yap manasında kullanılır.

IF .. ELSE (EĞER ... DEĞİLSE)



- `<html>`
- `<head>`
- `<title>Koşul yapıları </title>`
- `</head>`
- `<body>`
- `<script language="JavaScript">`
- `<!-- //eski sürüm tarayıcılardan kodumuzu saklayalım`
- `var gun = prompt ("Bugün günlerden ne ?" , "lütfen küçük harf kullanınız");`
- `if (gun=="pazar")`
- `{`
- `document.write ("Bugün günlerden " , gun , " olduğuna göre hafta sonundayız"`
- `,"
")`
- `document.write ("" , "İyi tatiller.." , "")`
- `}`

IF .. ELSE (EĞER ... DEĞİLSE)



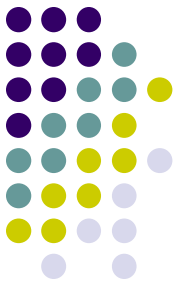
- else
- {
- document.write ("Bugün günlerden pazar olmadığına göre tatil gününde değiliz!" , "
")
- document.write ("İyi çalışmalar..")
- }
- //saklamayı bitir-->
- </script>
- </body>
- </html>

Döngü Yapısı



- Javascript'te diğer programlama dillerinde olduğu gibi istediğiniz işlemi 2 veya daha fazla kez yaptırmak için belli program kodları mevcuttur.
- Bu diğer dillere çok benzer olan for döngü komutudur.
- Bu komutun yaptığı işlem, istenilen fonksiyon veya fonksiyon parçalarını istenilen değerde tekrar etmektir.

FOR DÖNGÜSÜ



- *for (değişken_başlangıç_değerler1 ,
değişken_başlangıç_değeri2 ; döngü*
- *sayısı ; değişecek_değişken_adi_ve_türü*
)
- *{ yapılması istenen işlemler }*

FOR DÖNGÜSÜ



- Burada parantezler içerisinde verilen `değişken_başlangıç_değerler` kısmı ve `değişecek_değişken_adi_ve_türü` kısmını yazmanız gerekmez.
- Döngü içerisinde kullanılan değişken daha sonrada istenilen şekilde arttırılabilir veya azaltılabilir.
- Yapı gözünüzü korkutmasın hemen bir örnekle daha iyi anlayalım.

FOR DÖNGÜSÜ

- *for (a=0 , b=0 ; c<=3 ; c++)*
- *{ yapılması istenen işlemler }*
- *for ifadesi için kısa yazılım :*
- *var a,b=0;*
- *for (;c<=3;c++)*
- *{ yapılması istenen işlemler }*

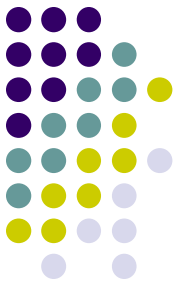


FOR DÖNGÜSÜ



- Şimdi bunu tam bir örnekle daha da pekiştirelim.
- Varsalım ki elimizde bir çarpım tablosu yapmak istiyoruz.
- Buna göre 5 sayısı için 1'den 10'a kadar sayıları bir tablo içerisinde vereceğiz.
- Şimdi bu durumda for döngüsüz 10 adet tablo yazmamız gerekecekti fakat biz for döngüsü ile işlemi 1 satıra indirgeyeceğiz.

FOR DÖNGÜSÜ



```
<html>
<head>
<title>for döngüsü</title>
</head>
<body>
<script language="JavaScript">
<!-- //eski sürüm tarayıcılardan kodumuzu saklayalım
var cevap=0;
for ( sayi=1 ; sayi<=10 ;sayi++)
{
var cevap = 5 * sayi ;
document.write( "5 * " , sayi , " =" , cevap ,"<br>")
}
//saklamayı bitir-->
</script>
</body>
</html>
```

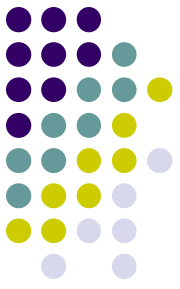
FOR DÖNGÜSÜ



```
file:///E:/2013-2014%20G%FCz/Web%20Tasarımının%20Temelleri/UYGULAMA/12-javascript/%E7arp...
Dosya Düzenle Biçimlendir
1 <html>
2 <head>
3 <title>for döngüsü</title>
4 </head>
5 <body>
6 <script language="JavaScript">
7 <!-- //eski sürüm tarayıcılardan
   kodumuzu saklayalım
8 var cevap=0;
9 for ( sayi=1 ; sayi<=10 ;sayi++)
10 {
11 var cevap = 5 * sayi ;
12 document.write( "5 * " , sayi , " =" ,
   cevap , "<br>")
13 }
14 //saklamayı bitir-->
15 </script>
16 </body>
17 </html>
18
```

```
for döngüsü - Windo...
E:\2013-2014 Güz\V
Sık Kullanılanlar Önerilen Siteler
for döngüsü
5 * 1 =5
5 * 2 =10
5 * 3 =15
5 * 4 =20
5 * 5 =25
5 * 6 =30
5 * 7 =35
5 * 8 =40
5 * 9 =45
5 * 10 =50
Bilgisayar | Kor %175
```

FOR DÖNGÜSÜ



- Burada gördüğünüz gibi işlem tek bir satıra indirildi.
- Şimdi de for döngüsünün yapmak istediğimiz işlemlerde yetersiz kaldığı durumlarda kullanabileceğimiz yapıları görelim.

ŞARTLI DÖNGÜ YAPISI WHILE



- Javascript kodu yazarken -programda bir önceki örnekte olduğu gibi- sayaç değişkeninin her değeri için istediğiniz işlemi yapmasını istemeyebilirsiniz.
- Bunun için while komutunu kullanırsınız ki bu Javascript'e "İstediğim işi şu şart sağlanıyorsa yap !" demiş olursunuz.
- While döngüsünde for döngüsünden farklı olarak döngü içerisindeki değişkenlerin tanımlanması gerekir.
- Şimdi yazım kurallarına bir göz atalım.

WHILE DÖNGÜSÜ



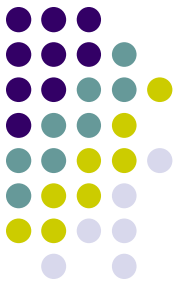
- Javascript kodu yazarken -programda bir önceki örnekte olduğu gibi- sayaç değişkeninin her değeri için istediğiniz işlemi yapmasını istemeyebilirsiniz.
- Bunun için while komutunu kullanırsınız ki bu Javascript'e "İstediğim işi şu şart sağlanıyorsa yap !" demiş olursunuz.
- While döngüsünde for döngüsünden farklı olarak döngü içerisindeki değişkenlerin tanımlanması gerekir.
- Şimdi yazım kurallarına bir göz atalım.

WHILE DÖNGÜSÜ



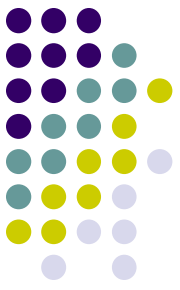
- *while (döngü şartı)*
- *{ şart doğruysa yapılacak işlemler}*
- *şart doğru değilse yapılacak işlemler*

WHILE DÖNGÜSÜ



```
file:///E:/2013-2014%20G%FCz/Web%20Tasarımının%20Temelleri/UYGULAMA/12-javascript/while.html - Özgün Kay...
Dosya Düzenle Biçimlendir
1 <html>
2 <head>
3 <title>while döngüsü</title>
4 </head>
5 <body bgcolor="#FF9933">
6 <div align="center"><font
   color="#FF0000">WHILE</font><font
   color="#FF0000" face="Verdana, Geneva, sans-
   serif"> Döngüsü</font></div>
7 <script language="JavaScript">
8 <!--
9 var c=0 , toplam=0 ;
10 while ( c<10 )
11 {
12     c++;
13     toplam+=c;
14     document.write( "c = " , c , "<br>" )
15 }
16     document.write( "toplam = " , toplam , "<br>" )
17 -->
18 </script>
19 </body>
20 </html>
21
```


WHILE DÖNGÜSÜ



```
while döngüsü - Windows Internet Explorer  
E:\2013-2014 Güz\V  
Bing  
Sık Kullanılanlar | Önerilen Siteler  
while ...  
WHILE DÖNGÜSÜ  
c = 1  
c = 2  
c = 3  
c = 4  
c = 5  
c = 6  
c = 7  
c = 8  
c = 9  
c = 10  
toplam = 55  
Bilgisayar | Korumalı Mod: Kapalı | %175
```