

BMÜ-425 İnternet Programlama

Dr. Öğr. Üyesi Erdal ÖZBAY

Fırat Üniversitesi Bilgisayar
Mühendisliği Bölümü

Javascript'e Giriş

Javascript Nedir?

- JavaScript, tarayıcı içerisinde kullanılabilecek bir script dilidir.
 - Bir JavaScript genelde doğrudan HTML sayfaları içerisine gömülür.
 - JavaScript is yorumlanan dildir.
 - İsim aslında ECMAScript'ine işaret eder.
 - ECMA: European Computer Manufacturer Association
-

Neden Javascript Öğrenilmeli?

- İstemci tarafında form işlemleri için
 - Daha dinamik bir grafik kullanıcı arayüzü için
 - www.scriptforest.com/javascript_cascading_menu.html
 - html sayfalarının dinamik güncellemesi için- Ajax
 - Ajax'ı kullanabilmek için JavaScript şarttır.
 - **AJAX** (Asynchronous JavaScript And Xml):web sayfalarında **JavaScript** ve**XMLHttpRequest** kullanımı ile etkileşimli uygulamalar yaratmaya yarayan tekniğe verilen addır.
 - Örnek:
 - maps.google.com/maps
 - ny.bestparking.com/
-

Önemli Kavramlar

- **JavaScript**'in düşük seviye sözdizimi **Java'ya benzer**, fakat nesne modeli oldukça farklıdır.
 - Bir **JavaScript**, **GUI olaylarına cevap** verecek şekilde ayarlanabilir.
 - JavaScript, fonksiyonlara birinci sınıf nesne olarak davranır.(Diğer nesnelerin kullanıldığı yerlerde kullanılır).
 - JavaScript , zayıf yazılmış bir dildir.
 - Tarayıcılar Document Object Model (DOM) kullanarak JavaScript ile belge ağacına erişim sağlar.
 - JavaScript kodu sunucudan veri isteyebilir (Dokuman ağacını ve böylece tarayıcının göstereceği sayfa güncellenebilir.)
 - Browser Object Model (BOM) nesne hiyerarşisinin biçimlendirir.
-

Javascript temelleri

- Javascript, web tarayıcıların programlama dili
 - Javascript, web sayfasına özel olan kodu web tarayıcıya eklemek için kullanılır
- Javascript gelişmiş, tamam, çok amaçlı programlama dilidir

Javascript temelleri

- Javascript'in kullanımları
 - Kullanıcı ile iletişim
 - Kullanıcı girişi
 - Kullanıcının girişini kontrol etme
 - Web sayfasının indirildikten sonra güncelleştirme
 - Web sayfası animasyonu

Javascript temelleri

- Javascript'in özellikleri
 - Skript programlama dili (web tarayıcı tarafından web sayfası oluşturulduğunda işletilir)
 - Serbest düzen programlama dili (programın görsel düzeni ve programdaki boşluklar önemli değil)
 - Tipsiz programlama dili (değişken/değerlerinin tipi belirtilmesi gerekmez)
 - Büyük/küçük harf duyarlı (isimler/anahtar kelimeleri büyük/küçük harf duyarlı)

Javascript'ın web sayfasında konumu

- Web sayfasında javascript, `<script>` tag içerisinde her hangi bir yerde bulunabilir
- Ayrıca tagların “onclick” gibi olay işleyicilerinde bulunabilir

Harici Javascript

- Stil sayfalarına benzerdir:
- Script gömülebilir veya harici bir dosya ile gösterilebilir.

```
<html>
```

```
<head>
```

```
<script src="xxx.js" />
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Merhaba JS

```
html>  
<body>  
<h2>  
<script type="text/javascript">  
document.write("Hello World!");  
</script>  
</h2>  
</body>  
</html>
```

Script etiketi, sayfaya
JavaScript eklemek
için kullanılır.

Script içerisindeki
kod sayfa
yüklendiğinde
hemen koşulur.
(Fonksiyon değilse)



Javascript Gelişimi

- Önemli tarayıcılar JavaScript Debugger'a sahiptir: Firefox (Firebug), IE (Visual Studio), hatta Chrome...
 - JavaScript Uygulamaları değişebilir, bu nedenle tüm büyük tarayıcılarda test etmek için yararlı olur.
 - Netbeans iyi bir sözdizim analizi özelliğine sahiptir.
-

Javascript Nesne Notasyonu

Nesne
İsimi

Metod
Çağırması

Metin
Kelimesi

- **document.write("Hello World!");**
- Sözdizimi Java'ya benzerdir.

Örnek

```
<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>
```

Örnek

```
<!DOCTYPE html>
<html><head>
<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>
</head>
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try
it</button>
</body>
</html>
```

Örnek

```
<!DOCTYPE html>
<html>
<body><h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try
it</button>
<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First
JavaScript Function";
}
</script>
</body>
</html>
```


Javascript'ın web sayfasında konumu

- `<script>` tagları, web sayfasının herhangi bölümünde bulunabilir (head, body, vb)
- `<script>` tagları, nekadar gerekirse birden çok olabilir ve web sayfasının farklı yerlere yerleştirilebilir
- Çoklu `<script>` tagları, web sayfası indirildiğinde web sayfasının başından sonuna kadar sırayla çalıştırılır

Javascript'in çıktısı

- Javascript web sayfasının elemanlarının içerikleri ve özellikleri değiştirilmesi için kullanılabilir
- Ayrıca javascript web sayfasına direkt olarak çıktı yazdırabilir, `document.write()` fonksiyonu kullanarak

Örnek

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

<h1>My First Web Page</h1>

<p id="demo">My First Paragraph</p>

<script>
elem = document.getElementById("demo");
elem.innerHTML = "My First JavaScript";
</script>

</body>
</html>
```

Örnek

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

<h1>My First Web Page</h1>

<p id="demo">My First Paragraph</p>

<script>
elem = document.getElementById("demo");
elem.innerHTML = "My First JavaScript";
</script>

</body>
</html>
```

Örnek

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

<h1>My First Web Page</h1>

<script>
document.write("<p>My First JavaScript</p>");
</script>

</body>
</html>
```

Notkalı virgül kullanımı

- Javascript serbest düzen programlama dili, yani javascript programının görsel düzeni önemli değil
- Boşluklar programın mantığını göstermek için kullanılabilir

Örnek

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

<h1>Pozitif sayıların toplanması</h1>

<script>
var sum = 0;
var num;
num = prompt("Sayı girin, 0 bitis için");
while (num != 0) {
    if (num>0) {
        sum=sum+parseInt(num);
    }
    num = prompt("Sayı girin, 0 bitis için ");
}
alert("Sum = " + sum);
</script>

</body>
</html>
```

Notkalı virgül kullanımı

- Javascriptte noktalı virgül komutları bitirmek için kullanılır
- Ancak, noktalı virgölün kullanımı da serbest

Örnek

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

<h1>Pozitif sayıların toplanması</h1>

<script>
var sum = 0
var num
num = prompt("Sayi girin, 0 bitis icin")
while (num != 0) {
    if (num>0) {
        sum=sum+parseInt(num)
    }
    num = prompt("Sayi girin, 0 bitis icin ")
}
alert("Sum = " + sum)
</script>

</body>
</html>
```

Javascript Komentler

- Eğer HTML'de aslı kod olarak görülmeyecek bir kısmı belirtmek için `<!-- ... -->` parentez kullanılırsa, `<script>` tagin içinde yani aslı javascript içinde benzer komentler veya yorumlar `//` veya `/* ... */` sembol kullanarak oluşturulur

Örnek

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

<h1>Komentler</h1>

<script>
// başlığa yaz
document.getElementById("myH1").innerHTML="Welcome to my Homepage";
// paragrafa yaz
document.getElementById("myP").innerHTML="This is my first paragraph.";

/*
Başlığa ve
paragrafa yaz
*/
document.getElementById("myH1").innerHTML="Welcome to my Homepage";
document.getElementById("myP").innerHTML="This is my first paragraph.";

var x=5;    // x değişkenini tanımla ve 5 değerini ata
var y=x+2;  // y değişkenini tanımla ve x+2 değerini ata
</script>

</body>
</html>
```

Javascript Komentler

- `//` ve `/* ... */` komentler, javascript bir programında baze kodun parçalarını tarayıcıdan saklamak için de kullanılabilir

Kontrol yapıları: {..} blok

- Herhangi bir programlama dili gibi Javascript'in temel bir kısmı olarak program kontrol yapılarıdır
- Javascript temel blok, döngü ve dallanma veri yapıları tanımlar

Kontrol yapıları: {...} blok

- Javascript blokları, javascript'in birkaç komutu gruplandırmak ve tek komut olarak kullanmak için kullanılır
- Javascript blokları {,} parentez kullanarak belirtilir

Örnek

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

<h1> {,} bloklar</h1>

<script>
function myFunction()
{
document.getElementById("demo").innerHTML="Hello Dolly";
document.getElementById("myDIV").innerHTML="How are you?";
}
</script>

</body>
</html>
```

Kontrol yapıları: {...} blok

- Javascript fonksiyonlarda, fonksiyonun vucudunu belirtmek için {...} bloğun kullanımı özellikle zorunludur
- Javascriptte {...} blokları, if/while/for kontrol yapılarında etkilenen işlemin sınırlarını göstermek için de genel olarak kullanılır
- Diğer Java, C, C++ programlama dillerine karşın, Javascriptte {...} blok değişkenlerin kapsamı **etkilemiyor**, yani blokte tanımlanmış veya kullanılmış değişkenler bloğun dışarısına **devam eder**

Örnek

```
{  
komut_1  
komut_2  
...  
komut_n  
}  
  
while (x < 10) {  
x++;  
}  
  
var x = 1;  
{  
var x = 2;  
}  
alert(x); // 2 yazacak  
  
if (saat<12 && saat>6)  
{  
    selam="Günaydın";  
}
```

Kontrol yapıları: if

– Javascript'in dallanma komutu, if komutudur

```
if (koşul)
{
    koşul doğru ise işlem
}
```

```
if (koşul)
{
    koşul doğru ise işlem
}
else
{
    koşul yanlış ise işlem
}
```

Kontrol yapıları: if

– Özel olarak if .. else if .. else dallanma yapısı

```
if (koşul1)
```

```
{
```

```
    koşul1 doğru ise işlem
```

```
}
```

```
else if (koşul2)
```

```
{
```

```
    koşul1 yanlış ve koşul2 doğru ise işlem
```

```
}
```

```
...
```

```
else
```

```
{
```

```
    tüm koşullar yanlış ise işlem
```

```
}
```

Örnek

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

<h1>selamlar</h1>

<script>
var simdi = new Date()
var saat = simdi.getHours()
if (saat<10)
{
    selam="Günaydın"
}
else if (saat<20)
{
    selam="İyi günler"
}
else
{
    selam="İyi akşamlar"
}
alert(selam)
</script>

</body>
</html>
```

Kontrol yapıları: for

- Javascript'in döngü yapısı, for komutudur

for (*baş komutu; dur koşulu; güncelleştirme komutu*)

{

döngü vucudu

}

Örnek

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

<h1> for döngüsü </h1>

<script>
arabalar=["BMW","Volvo","Saab","Ford"];
for (var i=0,l=arabalar.length; i<l; i++)
{
    document.write(arabalar[i] + "<br>");
}
</script>

</body>
</html>
```

Kontrol yapıları: for

- Javascript'in for döngüsünün
 - Baş komutu, döngünün başında çalıştırılır
 - Güncelleştirme komutu, döngünün her geçişin sonrasında çalışır
 - Dur koşulu, döngünün bitişi belirtmek için kullanılır
- Bunlar dışında, for elemanları için her hangi başka kurallar yok

Örnek

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

<h1>for döngüsü</h1>

<script>
arabalar=["BMW","Volvo","Saab","Ford"];
var i=0,len=cars.length;
for (;i<l;)
{
    document.write(arabalar[i] + "<br>");
    i=i++
}
</script>

</body>
</html>
```


Kontrol yapıları: while

- Javascript'in ikinci döngü yapısı, while komutu

```
while (koşul)
```

```
{
```

```
    koşul doğru iken yapılacak işlem
```

```
}
```

Örnek

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

<h1>while döngüsü</h1>

<script>
var sum = 0
var i = 1
var x = ""
while (i<10)
{
    sum = sum + i
    x=x + i + "'e kadar toplam " + sum + " dir <br>";
    i++;
}
document.write(x)
</script>

</body>
</html>
```

Kontrol yapıları: while

- while döngüsünün ikinci formatı

do

{

koşul doğru iken yapılacak işlem

}

while (*koşul*)

Örnek

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

<h1>do..while döngüsü</h1>

<script>
arabalar=["BMW","Volvo","Saab","Ford"];
var i=0,len=cars.length;
do
{
    document.write(arabalar[i] + "<br>");
    i=i++
}
while(i<l)
</script>

</body>
</html>
```

Kontrol yapıları: break

- Javascript özel “break” komutuna sahiptir
- “break” komutu bir döngüyü zorunlu olarak bitirmek için kullanılabilir

Örnek

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

<h1>for döngüsü</h1>

<script>
for (i=0;i<10;i++)
{
    x=x + "Satır sayısı " + i + " dir<br>";
    if (i==3) break;
}
</script>

</body>
</html>
```

Kontrol yapıları: break

- Baze programlarda “break” komutunun bir isimli {,} bloğun zorunlu olarak dışarısına çıkmak için kullanımı görülebilir

Örnek

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

<h1>isimli break</h1>

<script>
arabalar=["BMW", "Volvo", "Saab", "Ford"];
liste:
{
document.write(arabalar[0] + "<br>");
document.write(arabalar[1] + "<br>");
break liste;
document.write(arabalar[2] + "<br>");
document.write(arabalar[3] + "<br>");
}
</script>

</body>
</html>
```


Javascript değişkenleri

- Javascript serbest tipli programlama dili, yani değişkenlerin kullanımı çok serbesttir
 - Değişkenlerin tipinin tanımlanması gerekmez
 - Değişkenlerin kullanılmasından önce tanımlanması gerekmez
 - Değişkenler herhangi tipten veriler içerebilir

Javascript değişkenleri

Syntax

var strname = "x"

veya

strname = "x "

Fonksiyon içerisinde tanımlanan değişkenler yereldirler.

Fonksiyon dışında tanımlanan değişkenler pencere (window) nesnesinin değişkenleridir.

(Sayfada her yerde erişilebilirler)

Javascript değişkenleri

- Değişken “var” komutu olarak isteğe bağlı tanımlanabilir ve “=” komutu kullanarak bir değeri atanabilir

```
var x = 2
```

```
var a=0, y = x + 2
```

```
z = x + y
```

```
t = 0xFFFF // javascript 16bit sayıları  
anlar
```

Javascript değişkenleri

- Ancak tüm sayısal javascript değişkenleri 64bit float bir sayıdır
- Tamsayılar, dolayısıyla, 15 basamağa kadar içerebilir
- Sayısal değişkenler ek olarak “infinity” ve “nan” değerleri içerebilir; !javascriptte infinity bir sayıdır!, ancak $10/0 == \text{infinity}$
- Nan – sayı olmayan bir değerdir

Javascript değişkenleri

- Değişkenlerin isimleri
 - Bir harf veya \$ veya _ sembol ile başlanmalı
 - Küçük/büyük harf duyarlı
- Javascript ek olarak, değişkenleri için “undefined” ve “null” değeri tanımlar
 - “var” komutu kullanarak tanımlanmış ama değeri atanmamış değişkenler default olarak “undefined” değeri atanır
 - “null” değeri de değişkeninin boş olduğu anlamına gelir

Popup Kutuları

- Uyarı kutusu (Alert box) : Kullanıcı ilerlemek için “OK” demek zorundadır.
- Doğrulama kutusu (Confirm box): Kullanıcı ilerlemek için “OK” veya “CANCEL” demek zorundadır.
- İstem kutusu (Propt box): Kullanıcı ilerlemek için bir değer girer, sonra “OK” veya “CANCEL” der.

alert(“Email must be filled out”);
confirm(“sometext”);
prompt(“sometext”, “defaultvalue”);

Window nesnesinin metodudur!!



Javascript operatörleri

- Javascript genel işlemlere sahiptir
 - `=`, `+`, `-`, `*`, `/`, `%` (mod), `++` (arttırma), `--` (azaltma)
 - `=`, `+=`, `-=`, `*=`, `/=`, `%=` (`a+=b` demek `a=a+b`)
 - `==` (değer eşitliği), `===` (değer ve tip eşitliği)
 - `!=` (değer eşitliği), `!==` (değer ve tip eşitliği)
 - `>`, `<`, `>=`, `<=`
 - `&&` (AND), `||` (OR), `!` (NOT)

Javascript operatörleri

- Javascriptte “+” operatörü bir string ile kullanıldığında stringlerin birleştirilme anlamına gelir

Javascript operatörleri

- `txt1="İyi";`
`txt2="günler";`
`txt3=txt1+txt2;`
Sonucu: "İyigünler"

Javascript operatörleri

- `txt1="İyi ";`
`txt2="günler";`
`txt3=txt1+txt2;`
Sonucu: "İyi günler"

Javascript operatörleri

- `txt1="İyi";`
`txt2="günler";`
`txt3=txt1+" "+txt2;`
Sonucu: "İyi günler"

Javascript operatörleri

- `x=5+5;`
`y="5"+5;`
`z="Merhaba"+5;`

Sonuçları:

10

55

Merhaba5

- (string ve sayı toplandığında sayı bir string olarak stringe ekleniyor)

Javascript operatörleri

- Javascriptte “?” koşul komutu da var

değişken=(koşul) ? doğru-ise-değer : yanlış-ise-değer

Örnek

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

<h1>erişim kontrolü</h1>

<script>
yas=prompt("yaşınızı girdirin")
erisim=(yas<=18)?"Yasaktır":"Uygundur";
alert(erisim)
</script>

</body>
</html>
```

Örnek

```
<h1>Dizi işlemleri</h1>
```

```
<script>
```

```
var arabalarim = new Array();  
arabalarim[0] = "Saab";  
arabalarim[1] = "Volvo";  
arabalarim[2] = "BMW";
```

```
var arabalarim=new Array("Saab","Volvo","BMW");
```

```
var arabalarim=["Saab","Volvo","BMW"];
```

```
dizim[0]=Date.now;  
dizim[1]=fonksiyonum;  
dizim[2]=arabalarim;
```

```
var x=arabalarim.length           // dizinin uzunluğu  
var y=arabalarim.indexOf("Volvo") // dizide arama  
</script>
```

Örnek

```
<script>
var dizi = new Array()
for(i=0;i<25;i++)
{
    dizi[i]=Math.random();
}

var max = -infinity
for(i=0;i<dizi.length;i++)
{
    if(dizi[i]>max)
    {
        max=dizi[i];
    }
}
</script>
```


Örnek

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

<h1>ortalama</h1>

<script>
var dizi = [1,3,4,2,5,6,2,3,4], sum = 0
for(i=0;i<dizi.length;i++)
{
    sum+=dizi[i]
}
alert('ortalama ' + sum/dizi.length)
</script>

</body>
</html>
```

Javascript fonksiyonları

- Javascript fonksiyonları, web uygulamaların önemli bir kısmı olarak web sayfalarında çeşitli işlemler uygulamak için kullanılır
- Javascript fonksiyonları genellikle özel .js bir dosyada olup web sayfasına `<script src="myscript.js"></script>` tagını kullanarak bağlanır
- Öyle fonksiyonlar web sayfasındaki javascriptlerde ve sayfanın HTML elemanlarının olay işleyicilerinde çok sık kullanılır

Javascript fonksiyonları

- Javascript fonksiyonun formatı

```
function fonksiyonun-ismi()  
{  
    fonksiyonun kodu  
}
```

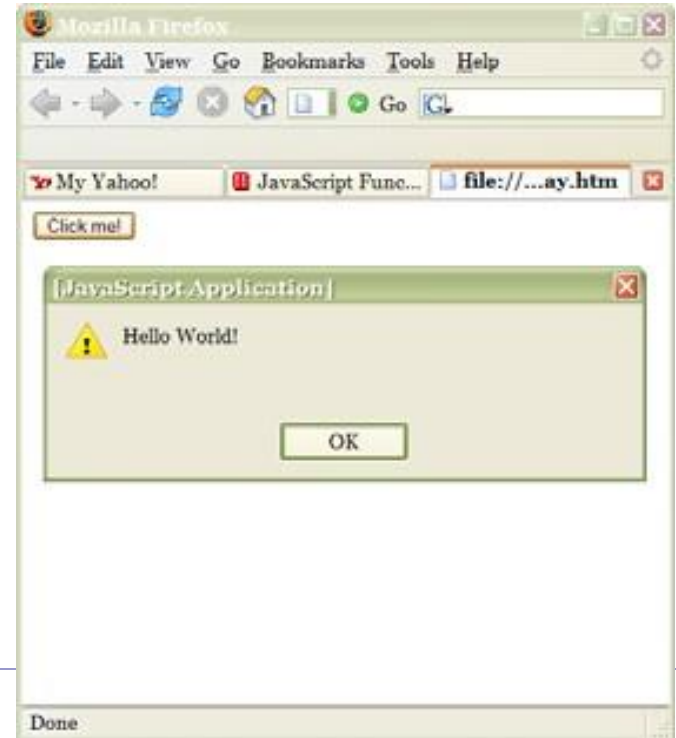
Javascript fonksiyonları

- Javascript parametrelili fonksiyonun formatı

```
function fonksiyonun-ismi(par1,par2)
{
    fonksiyonun kodu
}
```

Fonksiyon Örneği

```
<html> <head>  
<script type="text/javascript"> function displaymessage() {  
alert("Hello World!") } </script> </head>  
<body> <form>  
<input type="button" value="Click me!" onclick="displaymessage()" >  
</form> </body> </html>
```



Örnek

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

<h1>ortalama</h1>

<button onclick="myFunction('Harry Potter','Sihirbaz')">Try it</button>
<button onclick="myFunction('Bob','İnşaatçı')">Try it</button>

<script>
function myFunction(isim,is)
{
alert("Hoş geldin " + isim + ", " + is);
}
</script>

</body>
</html>
```

Javascript fonksiyonları

- Bir değeri hesaplayıp geri gönderen fonksiyon şu şekilde tanımlanabilir

```
function fonksiyonun-ismi(par1,par2)
{
    fonksiyonun kodu
    return hesaplama-sonucu
}
```

Örnek

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

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

```
<script>
```

```
function myFunction(a,b)
```

```
{
```

```
return a*b;
```

```
}
```

```
document.getElementById("demo").innerHTML=myFunction(4,3);
```

```
</script>
```

```
</body>
```

```
</html>
```


Javascript fonksiyonları

- Anonim fonksiyonlar, belirli yerlerde ayrı isimli fonksiyonu tanımlamadan fonksiyonu kullanımına imkan sağlar
- Anonim fonksiyon şu şekilde tanımlanır

```
function(par1,par2)
{
    fonksiyonun kodu
}
```

Örnek

```
<!DOCTYPE html>
<html>
<body onload='setInterval(function()
{
    var obj = document.getElementById("demo")
    if(obj.style.color=="red")
    {
        obj.style.color="blue";
    }
    else
    {
        obj.style.color="red";
    }
},1000) '>

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

</body>
</html>
```

Örnek

```
<!DOCTYPE html>
<html>
<body onload='setInterval("renk()",1000) '>
<p id="demo">MERHABA</p>

<script>
renk=function()
{
    var obj = document.getElementById("demo")
    if(obj.style.color=="red")
    {
        obj.style.color="blue";
    }
    else
    {
        obj.style.color="red";
    }
}
</script>

</body>
</html>
```

Javascript fonksiyonları

- Değişkenlerin kapsamı
 - Bir fonksiyonda tanımlanmış/kullanılmış değişkenin kapsamı bu fonksiyonun vucuduna eşit, yani fonksiyonlarda kullanılan değişkenler “Lokal”
 - Aynı isimli ancak farklı değerleri içeren değişkenlerin farklı fonksiyonlarda kullanılmasına imkan var
 - Herhangi bir fonksiyonun dışında tanımlanmış/kullanılmış değişkenler “Global” ve tüm fonksiyonlarda erişilebilir

Javascript fonksiyonları

- Değişkenlerin kapsamı
 - Değişken “var” komutu kullanarak veya değeri direkt olarak atayarak kullanıldığında oluşturulur
 - Lokal değişkenler ilgili fonksiyon bittiğinde silinir
 - Global değişkenler web sayfasın kapandığında silinir

Javascript kullanıcı girişi

- Kullanıcının girişi “input” taglarının “value” alanı olarak elde edilebilir, ve javascript’te belirli işlemler için kullanılabilir

Örnek

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

<input type="text" id="num1" value="2"/><br/>
<input type="text" id="num2" value="4"/><br/>
<button onclick="myFunction()">Click me</button>

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

<script>
function myFunction()
{
    var num1Element = document.getElementById("num1")
    var num2Element = document.getElementById("num2")
    var res = num1Element.value * num2Element.value
    document.getElementById("demo").innerHTML = res
}
</script>

</body>
</html>
```

Olaylar

- Olaylar (events) JavaScript tarafından belirlenen aksiyonlardır.
- Web sayfasındaki elemanlar, JavaScript fonksiyonlarını koşabilecek olaylara (events) sahiptir.
- Örnekler:
 - Fara tıklanması
 - Imge yüklenmesi
 - Fare ile üzerinde gezinme
 - Form sunumu

```
<input type="button"  
value="Click me!"  
onclick="displaymessage()"  
>
```

Bu özellik, üzerine
bilgi
yükleyebileceğinin
özellikte değildir.

Tipik Olay İşleyicileri

- **onclick** – bir eleman üzerinde gezinirken tıklanarak işaret edilmesi
 - **onmouseover** – İşaretçi bir eleman üzerine geldiğinde.
 - **onchange** – Bir kontrol giriş odağını kaybettiğinde, değeri değiştiğinde.
 - **onblur** – İşaretçi cihaz veya sekme gezintisinde bir eleman odaklandığında.
 - **onsubmit** – Bir form elemanının sunma (submit) düğmesi tıklandığında.
-

Javascript animasyonu

- Javascript kullanarak web sayfanın animasyonu, sayfanın farklı elemanlarının “style” özellikleri javascript programında değiştirilerek sağlanabilir
- Javascript'te bir fonksiyonun belirli zaman aralığı ile tekrarlamak için **setInterval(“komut”, zaman aralık)** kullanılır; zaman aralık milisaniye kullanarak belirtilir

Örnek

```
<!DOCTYPE html>
<html>
<body onload='setInterval("myFunction()",1000) '>

<h1>Kar</h1>



<script>
function myFunction()
{
    document.getElementById("demo").style.top=
        (parseInt(document.getElementById("demo").style.top)+1)+"%"
}
</script>

</body>
</html>
```

Javascript animasyonu

- Öyle animasyonları durdurmak için `clearInterval(par)` fonksiyonu kullanılmalı

Örnek

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

<h1>Kar 1</h1>



<script>
var myTimer=setInterval("myFunction()",1000)

function myFunction()
{
    var obj = document.getElementById("demo");
    var top=parseInt(obj.style.top)+1;
    obj.style.top=top+"%";

    if(top>90) clearInterval(myTimer);
}
</script>

</body>
</html>
```

Örnek

```
<!DOCTYPE html>
<html>
<body>
<script>
function resimGuncellestir()
{
    element = document.getElementById('myimage');
    if (element.src.match("bulbon"))
    {
        element.src = "pic_bulboff.gif";
    }
    else
    {
        element.src = "pic_bulbon.gif";
    }
}
</script>



<p>Tıklama lambayı açılıp kapatacak</p>

</body>
</html>
```

Örnek

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

<p id="demo">
JavaScript kullanarak elemanın stili değiştirilmesi
</p>

<script>
function myFunction()
{
    x = document.getElementById("demo"); // id="demo" isimli eleman al
    x.style.fontSize = "25px";           // font boyutunu değiştir
    x.style.color = "#ff0000";           // rengini değiştir
}
</script>

<button type="button" onclick="myFunction()">Click Me!</button>

</body>
</html>
```

Örnek

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

<p>Değişkeni oluşturur ve değerini "demo" paragrafında gösterir</p>

<button onclick="myFunction()">Try it</button>

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

<script>
function myFunction()
{
    var carname="Volvo";
    document.getElementById("demo").innerHTML=carname;
}
</script>

</body>
</html>
```


Örnek

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

<p id="demo">Bir dizi oluşturur ve sonucu göster</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
    var fruits = ["Banana", "Orange", "Apple", "Mango"];
    var x=document.getElementById("demo");
    x.innerHTML=fruits.toString();
}
</script>

</body>
</html>
```

Egzersizler - 1

- “Neon” – temel javascript kullanarak, zamanla rengi değiştiren tarayıcı penceresinde neon işareti yapın

NOT: metin içeren elemanın `style.color` özelliğini javascriptte değiştirin, ayrıca animasyonu oluşturmak için `setInterval` fonksiyonunu kullanmanız lazım

Egzersizler - 2

- “Popup penceresi” – temel javascript kullanarak, bir linki tıklandığında açılan ve başka bir linki tıklandığında kapanan bir kutu oluşturun

NOT: kutuyu oluşturan elemanın `style.display` özelliğini kullanın ve ilgili fonksiyonu `a` tagın `onclick` olay işleyicisine bağlayın

Egzersizler - 3

- “Zıplayan top” – tarayıcı penceresinde zıplayan bir topu oluşturmak için javascript yazın

NOT: top içeren img tagın style.top özelliğini zamanla güncelleştirin, ayrıca setInterval fonksiyonu da kullanılmalı