



ONDOKUZ MAYIS ÜNİVERSİTESİ  
UZAKTAN EĞİTİM MERKEZİ

İnternet ve Ağ Teknolojileri Önlisans Programı

# JAVASCRIPT

Öğr. Gör. Ömer SEVİNÇ

*Fırsatlar Sunar*



Ünite 7

# Fonksiyonlar

Öğr. Gör. Ömer SEVİNÇ

## İÇİNDEKİLER

<b>7.1. FONKSİYONLAR.....</b>	<b>3</b>
7.1.1. Fonksiyon Tanımlamak.....	3
7.1.2. Fonksiyonlara Veri Gönderme-Alma.....	4
7.1.3. Hazır Fonksiyonlar.....	5



## 7.1. FONKSİYONLAR

Tekrarlama yapılması için gerekli işlem ve komut gruplarına fonksiyon adı verilir. Uygulama içerisinde bir komut ya da komut grubunu tekrar tekrar kullanmak isteyebiliriz. Bu gibi durumlarda aynı satırları defalarca yazmak zorunda kalabiliriz. Tekrar ettiğimiz kodlarda belli bir değeri değiştirmek istediğimizde tüm tekrar eden satırları yeniden düzenlemek gerekecektir.

Kısaca fonksiyonlar tekrar eden kod bloklarını alıp paketlememize yarar. Uygulama içerisinde artık bu paketin ismini kullanarak istediğimiz yerde bu kodları çalıştırabiliriz. Bu sayede daha sonra üstüne yapılacak değişiklikleri ve hata kontrolünü hızlı bir biçimde yapmamıza imkân sağlar.

### 7.1.1. Fonksiyon Tanımlamak

Fonksiyon tanımlamak için aşağıdaki kalıp uygulanır. Fonksiyon tanımlandığında editör penceresinin istenilen yerinde fonksiyon çalıştırılabilir.

```
10 function fonksiyon_ismi()  
11 {  
12     fonksiyon içerisinde çalıştırılacak kodlar  
13 };
```

- Fonksiyonu çalıştırmak için;

```
15 fonksiyon_ismi();
```

şeklinde yazarak istenilen yerde fonksiyon çalıştırılır.

#### Sayfadan girilen ismi 20 defa alt alta yazdıran fonksiyon hazırlamak için;

- Editör programını açarak yeni bir html sayfası oluşturunuz.
- Fonksiyonumuzun ismi yazdir olsun. Prompt komutu ile dışarıdan alınan ifadeyi tutmak

```
11 function yazdir()  
12 {  
13     var isim=prompt("isminizi yazınız :");
```

- İsim değişkenini 20 defa alt alta yazdırmak için for döngüsü kullanıp fonksiyonu kapatalım.

```
17     for(var i=1;i<20;i++)  
18     {  
19         document.write(isim+"<br>");  
20     }  
21 };
```

- Fonksiyonumuz artık hazır. Fonksiyonun ismi ile sayfada istediğimiz yerde fonksiyonu çağırıp kullanabiliriz.

```
23 yazdir();
```

### 7.1.2. Fonksiyonlara Veri Gönderme-Alma

Yazdığımız fonksiyonlara isteğimize göre değer verip işlenmiş verinin sonucunu geri alabiliriz. Böylelikle fonksiyonlarımız anlık olarak verileri işleyip sonucu programa dâhil edebilir.

Fonksiyona parametre göndermek için yapmamız gereken fonksiyon isminden sonra parantez içerisinde gelecek veri için değişken ismi yazmak olacaktır.

**Kullanıcının girdiği yaş bilgisine göre kaç gün yaşadığını hesaplayan script kodlarını yazmak için;**

- Yeni html sayfası oluşturun. Fonksiyonumuzu head etiketleri arasında tanımlayıp body etiketleri arasında kullanacağız.
- Kullanıcıdan isim ve yaş bilgilerini almak için x ve y değişkenlerini kullanacağız.
- Prompt komutunu kullanarak bilgileri değişkenlere atayalım.

```
19 var x = prompt ( "Lütfen isminizi giriniz : ", "" );  
20 var y = prompt ( "lütfen yaşıınızı giriniz.: ", "" );
```

- Fonksiyonumuzu head etiketleri arasında yazmaya başlayalım. Fonksiyonumuz ad ve yas olarak iki parametre alacak.

```
5 <title>Yaş Hesap Uygulaması</title>  
6 <script type="text/javascript">  
7 function ayBul(ad,yas)  
8 {
```

- Fonksiyona parametre olarak girilen yas bilgisini gün sayısına çeviriyoruz.

```
10 var gun = yas * 12 * 30;
```

- Write metodu ile sayfaya yazdırıyoruz ve fonksiyonu kapatıyoruz.

```
12 document.write ("Merhaba "+ad+", Toplamda " + gun+ " gün yaşamışsınız ");  
13 }
```

- Body etiketleri arasında fonksiyonu ismi ile çağırarak çalışmasını sağlıyoruz.

```
26 ayBul (x,y);
```

Yukarıdaki örnekte fonksiyon, kullanıcıdan aldığı bilgileri işleyip ekrana çıktı olarak vermektedir. Fonksiyonun çalışması sonucu bir veri almak istiyorsak return kelimesini fonksiyon içerisinde kullanmak zorundayız.

**Örneğin girilen sayının karesini hesaplayan ve uygulamaya sayısal değer olarak dâhil eden bir fonksiyon yazalım.**

Script etiketleri arasında karesi isminde bir fonksiyon tanımlayalım. Fonksiyonumuz dışarıdan tek parametre alsın.

```
<script type="text/javascript">
function karesi (x)
{
```

- Fonksiyon, dışarıdan aldığı x değerinin kendi içerisinde karesini alma işlemi yapsın.

```
11 var kare=x*x;
```

- Hesapladığı kare değerini fonksiyon dışına veri olarak göndermek için return komutunu kullanalım.

```
13 return kare;
14 }
```

- Fonksiyonumuz hazır artık. Sayfada istediğimiz işlemde fonksiyonumuzu kullanabiliriz.

```
22 alert(karesi(2)); //4
23 alert(karesi(4)); //16
24 alert(karesi(5)); //25
```

### 7.1.3. Hazır Fonksiyonlar

- parseInt()

String ifadenin sayısal kısmını alarak tam sayıya çevirir.

```
14 document.write(parseInt("1980volkan")); //1980
```

İfade ondalıklı ise parseInt fonksiyonu sadece tam olan kısmını alır.

```
16 document.write(parseInt("212.45")); //212
```

String ifade sayısal bir değer ile bağlamıyorsa parseInt fonksiyonunun döndüreceği değer NaN olacaktır.

```
18 document.write(parseInt("volkan1980")); //NaN
```

- parseFloat()

parseInt fonksiyonundan farkı sayısal ifadeleri ondalıklı olarak da alabilir.

```
16 document.write(parseFloat("212.45")); //212.45
```

- String()

İfadeyi string ifade yapar.

```
14 var sayi1=10;
15 var sayi2=15;
16
17 var toplam1=sayi1+sayi2;
18 alert(toplam1); //25
19
20 var toplam2=String(sayi1)+sayi2;
21 alert(toplam2); //1015
```

- Number()

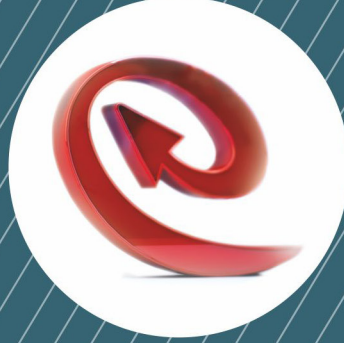
String ifadeyi sayısal değer yapar

```
14 var sayi1="10";
15 var sayi2=15;
16
17 var toplam1=sayi1+sayi2;
18 alert(toplam1); //1015
19
20 var toplam2=Number(sayi1)+sayi2;
21 alert(toplam2); //25
```



Bu Ders Notu Ondokuz Mayıs Üniversitesi, Uzaktan Eğitim Merkezince kullanılmak üzere hazırlanmıştır. Ticari amaçlarla kullanılamaz. Kopyalanması, çoğaltılması ve dağıtılması ilgili birimin yazılı iznine tabidir.





Ondokuz Mayıs Üniversitesi Uzaktan Eğitim Merkezi  
Kurupelit Kampüsü Atakum / SAMSUN



0362. 457 8936 **Fax:** 0362. 457 5806



[irtibat@uzem.omu.edu.tr](mailto:irtibat@uzem.omu.edu.tr)



<http://uzem.omu.edu.tr>