

# TBT MATLAB Ders 8

13/12/2022 - Güz Dönemi 12. Hafta

## İçerik Listesi

- [MATLAB Fonksiyonları](#)
  - [Alt fonksiyonlar](#)

## MATLAB Fonksiyonları

MATLAB de yazılan bilgisayar programları fonksiyonlardır. Fonksiyonlar görsel olarak sonuç verebileceği gibi text olarak da sonuç verebilir.

- Her fonksiyon kendine özel bir isme sahip olmalı. Mümkün olduğunca kısa ve amaca uygun olmalı. İsimlendirme kuralları değişkenler ile aynıdır.
- Bir fonksiyon genellikle üzerinde işlem yapacağı bir parametre alır. Giriş parametresi almayan fonksiyonlara örnek `clc()`. Genellikle kullanıcıya bir tane de çıkış parametresi verir.

Bir fonksiyon şöyle tanımlanabilir:

```
function [outputArg] = func1(inputArg)
% FUNC1 Girilen değeri verir
% Girdiğiniz bir değeri aynı şekilde geri verir.
outputArg = inputArg;
end
```

`function` anahtar kelimesi ile bir fonksiyon tanımlayacağımızı belirtiriz. `[]` içine çıkış parametreleri tanımlanır. `=` den sonra fonksiyon ismi verilir. Unutulmaması gereken şudur ki dosya adı ile fonksiyon adı aynı olmalıdır. Yani `func1` in dosyasının adı `func1.m` olmalıdır. Bir dosyada yalnızca 1 fonksiyon olabilir. Daha sonra `()` içerisine giriş parametreleri verilir. İlk 2 yorum satırı kullanıcının `help func1` yazdığında verilecek çıktıdır. Daha sonra işlemler yapılır ve çıkış olarak verilecek değer belirlenir. Ve `end` ile bitirilir.

```
function [cp] = fonk_adi(gp)
function [cp] = fonk_adi(gp1, gp2, ..., gpn)
function [cp1, cp2, ..., cpn] = fonk_adi(gp)
function [cp1, cp2, ..., cpn] = fonk_adi(gp1, gp2, ..., gpn)
function [cp] = fonk_adi()
function fonk_adi(gp)
```

Sırasıyla tek giriş parametrelili ve çıkış parametrelili fonksiyon; tek çıkış, çok giriş parametrelili fonk; çok çıkış, tek giriş parametrelili fonksiyon; çok giriş ve çıkış parametrelili fonk; tek çıkış parametrelili ve giriş parametresiz fonksiyon; çıkış parametresiz fonksiyon.

**Örnek:** İki nokta arası uzaklığı hesaplayan bir MATLAB fonksiyonu yazınız.

```
function sonuc = distance(nokta1, nokta2)
% DISTANCE İki nokta arası uzaklığı bulur
% Dik koordinat sisteminde iki nokta arası uzaklığı hesaplar. Giriş
olarak iki liste alır.
% Kullanım : distance(nokta1, nokta2)    (nokta1 = [x1,y1], nokta2 =
[x2,y2])

x1 = nokta1(1);
x2 = nokta2(1);
y1 = nokta1(2);
y2 = nokta2(2);
sonuc = sqrt((x2-x1)^2 + (y2-y1)^2);
end
```

**Örnek:**  $f(x, y) = x^2y + \sqrt{xy} + \frac{\ln(x+1)}{\log y}$

```
function sonuc = f(x,y)
sonuc = (x^2)*y + sqrt(x*y) + log(x+1)/log10(y);
end
```

Built-in bir fonksiyonla aynı isimde bir fonksiyon yazarsak çalıştırdığımızda çalışacak olan bizim tanımladığımızdır. Önem sırası:

1. Oturum değişkeni
2. Kullanıcı fonksiyonu
3. Built-in fonksiyon

## Alt fonksiyonlar

Ana fonksiyonumuzu parçalara ayırmak için aynı dosya içine yardımcı alt fonksiyonlar oluşturulabilir.

```
function sonuc = anaFonk (x)
sonuc = altFonk(x) + 1;
end

function sonuc = altFonk(a)
```

```
        sonuc = a^3 - 10;  
    end
```

Bu şuna eşittir:

```
function sonuc = fonk (x)  
    sonuc = x^3 - 10 + 1;  
end
```

Bu durum size mantıklı gelmeyebilir ama ileride daha büyük fonksiyonlarda çok yardımcı olur. (Ben bir [never nester](#) olarak bunu çok önemli bulurum.)

Burada unutulmaması gereken şey `altFonk` un CLI da çağırılmamasıdır.

**Örnek:** `isPrime()` kullanmadan bir sayının asal sayı olup olmadığını bulan bir fonksiyon yazınız.

```
function sonuc = asalMi (n)  
    pBolen = bolenler(n);  
    if length(pBolen) == 2  
        sonuc = true;  
    else  
        sonuc = false;  
    end  
end  
  
function liste = bolenler (n)  
    liste = [];  
    for i=1:n  
        if mod(n,i) == 0  
            liste = [liste, i];  
        end  
    end  
end
```

Yazılan bir fonksiyon başka fonksiyonların içinde kullanılabilir.

**Örnek:** n'den küçük asal sayıların listesini bulan fonksiyon.

```
function liste = asallar(n)  
    liste = [];  
    for i=2:n-1  
        if asalMi(i)  
            liste = [liste, i];  
        end  
    end
```

```
end  
end
```

[< Önceki](#) | [Sonraki >](#)