



BATCH : B45-46-71
LESSON : **Javascript**
DATE : 29.03.2022
SUBJECT : **Conditional Statements**



 techproeducation.com

 info@techproeducation.com

 +1 (917) 768-7466



ternary

if-else bloğunun tek satırda basit bir şekilde yazılmasıdır

mantıksal_ifade ? doğruysa : yanlışsa

```
let maas = 15000;  
console.log( maas>13000 ? "Normal" : "Az" );
```



ternary

PRACTISE

☐ Emekli
☒ Çalışan
Maaş=13500

- › Şekildeki gibi form oluşturunuz. Çalışan seçildiğinde tam, emekli ise %10 kesinti uygulayarak maaşı gösteriniz. Bunu ternary kullanarak yapınız.



|| ve &&

if bloğunun tek satırda basit bir şekilde yazılmasıdır

mantıksal_ifade **&&** doğruysa

mantıksal_ifade **||** yanlışsa

```
var x = 15  
x == 10 && alert("x is 10");  
x == 10 || alert("x is not 10");
```

Buradaki mantık and ve or un çalışma prensibi ile alakalıdır.

&& operatörü kullanıldığında birinci ifade yanlışsa ikinci ifadeye bakmaz. İlk ifade doğruysa ikinci ifadeye bakar ve gerçekleştirir.

|| operatörü kullanıldığında birinci ifade doğruysa ikinci ifadeye bakmaz. İlk ifade yanlışsa ikinci ifadeye bakar ve gerçekleştirir.



Hesap Makinesi

HOMEWORK

İki textbox, select, button ve label oluřturunuz. Textbox lara girilen sayıları select ten seçilen işlem türüne göre, hesapla butonuna basıldığında işleme tabi tutup sonucu label da gösteren programı yazınız.



JavaScript

- › for
- › while
- › do-while

Basic Loops



Döngüler

Tekrarlanan işlemleri daha az kod yazarak gerçekleştirmek için döngüler kullanılır.

for

while

do-while



for

```
for(var i=0; i<10; i++){  
  .  
  .  
}
```

Döngünün
başlangıç
sayısı

Doğru
olduğu
sürece
döngü tekrar
eder

Artış
miktarı

- › En temel döngü türüdür.
- › Tekrar ettirilecek kodun baştan kaç defa tekrar ettirileceği biliniyorsa bu tür döngüler kullanılabilir.

```
for(let i = 1 ; i<= 10 ; i++){  
  console.log("Merhaba");  
}
```




for

PRACTISE

İki textbox tan alınan sayılar arasında 7 ile bölünebilen sayıların adedini bulan ve sonucu bir h1 içinde gösteren programı yazınız.



for

PRACTISE

 Uygula

- Notebook 1500\$
- Monitor 500\$
- Klavye 10\$
- Mouse 5\$

Bir ul listesinde ürün isimleri ve fiyat bilgileri bulunmaktadır. Textbox a girilen indirim oranı kadar listeye indirim uygulayan programı yazınız.



while

```
while(a<50){
```

```
...
```

```
}
```

Doğru
olduğu
sürece
döngü tekrar
eder

- › Döngünün ne kadar döneceği döngü bloğunun içindeki kodlara bağlı ise bu durumda while döngüleri kullanılabilir.



do-while

```
do{  
  ...  
}  
while(a<50)
```

Doğru
olduğu
süreçe
döngü tekrar
eder

- Do while da kontrol sonda olduğu için döngü içindeki kodlar en az bir kere icra edilir.



do-while

PRACTISE

İki textbox tan alınan sayılar sırayla toplanmaya başlandığında, genel toplamı 2000 i ulaştıran kaç adet sayı olduğunu bulan ve sonucu alert ile gösteren programı yapınız.



JavaScript

Functions

- › Fonksiyon nedir?
- › Nasıl çalışır?
- › Fonksiyon tanımlama
- › İç içe fonksiyon
- › Scope



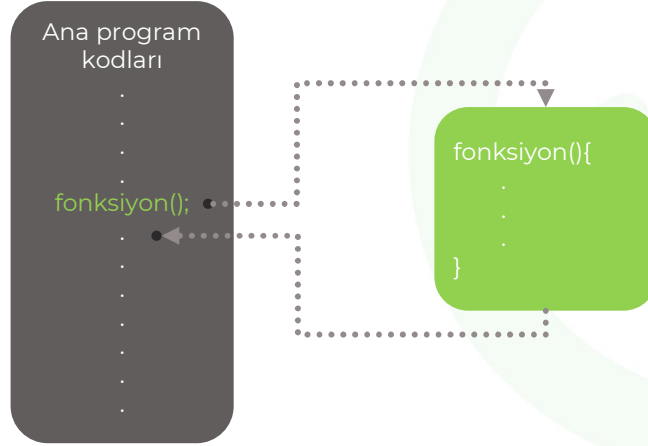
Fonksiyon nedir?

Fonksiyon, programda tekrar tekrar çalıştırılacak alt programlara denir.

- ◉ Kod tekrarlarının önüne geçer
- ◉ Problemler küçük parçalara ayrılarak daha kolay çözülür.
- ◉ Parçalara ayrılan sorun, daha fazla insan tarafından aynı anda çözülür.
- ◉ Hata ayıklama daha kolay olur



Nasıl çalışır



Fonksiyonlar tanımlanarak içlerine gerekli kodlar yazılır. Ana program içinde fonksiyon ismiyle çağrılır. Çağrıldığında program akışı fonksiyona yönelir. Oradaki kodlar çalıştırıldıktan sonra tekrar ana programa dönerek kaldığı yerden devam eder.



Fonksiyon tanımlama

Geleneksel yöntem

```
function fonksiyonIsmi(param1, param2, ...){  
  // işlemler  
  // ...  
  return donenDeger;  
}
```

Fonksiyona dışarıdan
gelen parametreler

Fonksiyonun çağrıldığı
yere döndürdüğü değer

```
var x = fonksiyonIsmi(paramValue1, paramValue2, ...);
```

Fonksiyonun çağırılması

- Bir fonksiyon bir kere tanımlanır ve program içerisinde istenildiği kadar çağrılabilir.
- Fonksiyonun tanımlanması çağrılmadan önce de yapılabilir sonra da yapılabilir.
- Bir Fonksiyon, argüman (parametre) almak zorunda değildir.
- Bir fonksiyon bir değer döndürmek zorunda değildir.



Fonksiyon tanımlama

Yaş hesaplayan bir fonksiyon yazınız.

```
function yasHasapla(){  
  console.log(`Benim yaşıım ${2022 - 1990}`);  
}  
  
yasHasapla();  
yasHasapla();
```

Bu fonksiyon hep aynı yaşı gösteriyor. Farklı doğum tarihlerine göre daha modüler hale getirilebilir mi?



Fonksiyon tanımlama

Yaş hesaplayan bir fonksiyon yazınız.

```
function yasHasapla(dogumTarihi){  
  console.log(`Benim yaşıım ${2022 - dogumTarihi}`);  
}  
  
yasHasapla(1980);  
yasHasapla(2001);
```

Eğer yaş bilgisi ana
programda kullanılmak
istenirse ne yapılabilir?



Fonksiyon tanımlama

Yaş hesaplayan bir fonksiyon yazınız.

```
function yasHasapla(dogumTarihi){  
    return 2022 - dogumTarihi;  
}  
  
var yas = yasHasapla(1980);  
if(yas >= 65){  
    ...  
}
```



Fonksiyon

PRACTISE

Bir sayının asal olup olmadığını bulan fonksiyonu yazınız.