



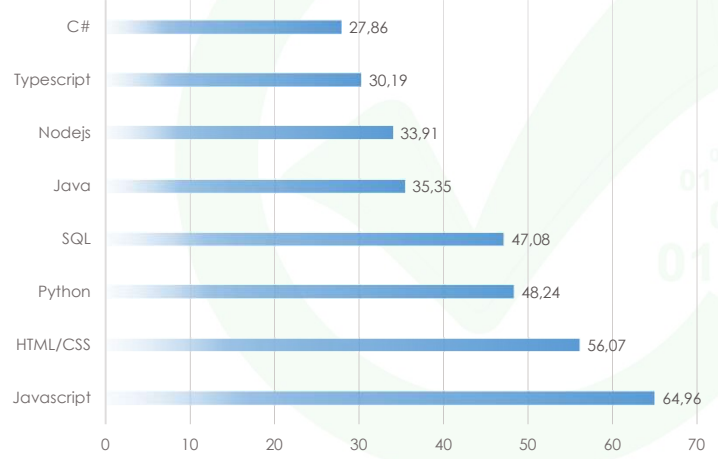
# IT Fundamentals

## Javascript Basics

- › Javascript
- › Variables & Constants
- › Data Types
- › Conditional Statements
- › Loops
- › Functions



## Javascript



<https://www.statista.com>

Web in en popüler programlama dillerinden biridir.

Web sitelerinin neredeyse tamamı Javascript kullanmaktadır.

Önceleri client side için tercih edilse de şimdi nodejs kütüphanesi sayesinde backend programlamada da tercih edilmektedir.

<https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>



## Aritmetik operatörler

Operatör	Sembol	Örnek
Toplama	+	45+6
Çıkarma	-	35-7
Çarpma	*	25*2
Bölme	/	16/5
Üs Alma	^	5^3
Mod Alma	%	12%5



## Karşılaştırma operatörleri

Operatör	Sembol	Örnek
Eşittir	== veya ===	Not==75
Eşit değildir	!= veya !==	Not != 45
Büyüktür	>	Yas > 45
Küçüktür	<	Yas < 18
Büyük veya eşit	>=	Ort >=80
Küçük veya eşit	<=	Ort<=90



Karşılaştırma işlemlerinin sonucu **TRUE** veya **FALSE** olur



## Mantıksal operatörler

Operatör	Sembol	Örnek
And	&&	ort > 50 && ort < 70
Or		yas < 16    yas > 70
Not	!	! (yas > 45)

A	B	A && B	A    B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

A	!A
1	0
0	1



## Mantıksal Operatörler

not1	50
not2	60
not3	70

- T** not1 > 30 **&&** not1 < 70
- F** not2 >= 60 **&&** not3 != 70
- T** not3 < 100 **||** not2 > 80
- F** not1 + not2 < not3 **||** not1 < 50
- T** not1 < 30 **||** not2 > 50 **&&** not3 == 70



## İşlem önceliği

Aritmetik işlemler yapılırken bilgisayar operatör önceliğine göre işlem yapar.



Öncelik seviyesi aynıysa soldaki ifadeyi önce yapar.

**()** Parantez

**^** Üs alma

**\*** Çarpma

**/** Bölme

**+** Toplama

**-** Çıkarma



## İşlem önceliği

$$8/2*(2+2)$$

16

$$30 - 3^2 / 3 + 10$$

37

$$16 / 2 * 3 - 2^{(4 / 2)}$$

20

$$(14 * 2 / 7)^2 / 4 + 5$$

9





## Case Types



camelCase



snake\_case



kebab-case



Train-Case



PascalCase



MACRO\_CASE



## Javascript syntax



- › Case sensitive bir dildir.
- › Satır sonlarına genellikle noktalı virgül (;) konulur.
- › Identifier lar(variables, constants, functions) harf, \_ veya \$ ile başlayabilir.
- › Identifier lar **camel case** ile yazılır.

Case sensitive: Büyük küçük harf duyarlı.



## Console

Javascript ile tarayıcıların console bölümüne yazdırmak için **console** object kullanılır.

```
console.log("bla bla bla");
```



## Alert, Confirm, Prompt

### Alert

Bilgilendirme yapmak için kullanılır  
`alert("Silme yetkiniz yok");`

Silme yetkiniz yok.

Tamam

### Confirm

Soru sormak için kullanılır.  
`confirm("Silme istediğinizden emin misiniz?");`

Silme istediğinizden emin misiniz?

Tamam

İptal

### Prompt

Bilgi almak için kullanılır.  
`prompt("Adınızı giriniz");`

Adınızı giriniz

Tamam

İptal



## Variables (Değişkenler), Constants (Sabitler)

CANADA	CAD	0.9512	0.8883
CHINA	CNY	7.3169	6.0910
EURO	EUR	0.6644	0.6100
JAPAN	JPY	109.00	102.00
SINGAPORE	SGD	1.3712	1.2630
HONG KONG	HKD	0.0043	0.0072
NEW ZEALAND	NZD	1.1646	1.0675
NEW ZEALAND	NZD	1.2536	1.2818
NEW ZEALAND	NZD	1.2536	1.2818
HONG KONG	HKD	0.0043	0.0072

► Programlama dillerinde ihtiyaç olduğu an ulaşılacak veri tutuculara **değişken** ya da **sabit** adı verilir.



## Variables (Değişkenler), Constants (Sabitler)

```
var sayi;  
var Sayi;  
var genelOrtalama;  
const KDV = 1.18;
```

- › Javascript te değişken tanımlamak için **var** veya **let** keyword leri kullanılır.
- › Sabit tanımlamak için ise **const** keyword u kullanılır.
- › Sabitler tanımlanırken değerleri de verilmelidir.

### Değişken oluştururken dikkat edilmesi gerekenler:

- Değişken isimleri **case sensitive** yani büyük küçük harf duyarlıdır. Örneğin; **sayi** ve **Sayi** değişkenleri birbirlerinden farklıdır.
- Değişken isimleri içinde tutacağı bilgiyi anımsatacak şekilde isimlendirilirler. Örneğin bir kişinin yaşı saklanacaksa «var yas» şeklinde tanımlanabilir.
- Değişken isimlendirmek için kullanılan farklı standartlar vardır: Camel Case, Pascal Case, Snake Case, Kebap Case farklı standartlar kullanılmaktadır. Javascript te genellikle camel case tercih edilirken,

class oluşturulurken pascal case kullanılır.

- camelCaseBuSekildeYazilir
  - PascalCaseBuSekildeYazilir
  - snake\_case\_bu\_sekilde\_yazilir
  - kebab-case-bu-sekilde-yazilir
- 
- Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir. Ancak sayılar başa gelmez. Örneğin **sayi1** doğru bir isimlendirmeyken **1sayi** doğru bir isimlendirme değildir.
  - Değişken isimlendirilirken alt tire ( \_ ve \$ ) kullanılabilir. Ancak boşluk ve diğer özel karakterler ( ?, %, !, ., + vb. ) kullanılmaz. Örneğin **ev adresi** ya da **kimlik%no** gibi değişken isimleri kurallara aykırı olduğundan hataya neden olacaktır.
  - Değişken isimlendirilirken özel kullanım için ayrılmış olan **if, for, true vb.** ifadeler hata vermemesine rağmen özellikle kodların daha anlaşılır olması amacıyla kullanılmamalıdır.
  - İngiliz alfabesinde bulunmayan karakterler (**ç,ğ,ı,ö,ş,ü**) kullanılamaz.



## Variables (Değişkenler), Constants (Sabitler)

PRACTISE

✓ders

✗1not

✓gen\_ort

✓\$maas

✗ilk sayi

✗son\_şekil

✗gun%

✗harf-bir

✓dGunu





## Setting variables

```
var yas;  
yas = 13;  
yas = 21;  
console.log(yas);
```

- › Değişkenlerin değerini değiştirmek için eşittir (=) ifadesi kullanılır.
- › Eşittir in sağındaki değer, soldaki değişkene atanır.



## Setting variables

```
var yas;  
yas = 13;  
yas = yas + 5;  
console.log(yas);  
yas = yas - 3;  
yas = yas / 2;
```

- › Bir değişkenin değeri kendi değerine bağlı olarak değiştirilebilir.



## Setting constants

```
var tutar = 1500;  
const kdv = 1.18;  
tutar = 2000;  
kdv = 1.30;
```

Değişkenlerin program içinde değerleri değiştirilebilir, sabitlerin değiştirilemez.

Sabitlerin değerleri ilk tanımlandıklarında belirlenir.



## Data types (Veri tipleri)

### String

```
var adi="Ali";
```

### Number

```
var yas=18;
```

### Boolean

```
var emekli=true;
```

**! typeof**

Javascript te 6 veri tipi bulunmaktadır. Bunlardan en temel olanları String, Number ve Boolean dır. Ancak Javascript te diğer dillerden farklı olarak değişkenler/sabitler tanımlanırken veri türü (data type) belirtilmez. Tanımlama yönteminden Javascript değişkenin/sabitin içinde nasıl bir veri saklayacağını anlar.

**typeof** ifadesi ile bir değişkenin değeri öğrenilebilir

```
var x = 15;  
console.log(typeof(x));
```



## Javascript basics

PRACTISE

Girilen ismi  
merhaba ile  
karşılaman  
programı  
yazınız.

```
const isim = prompt("Adınızı giriniz:");  
console.log("Merhaba " + isim);
```



## Javascript basics

PRACTISE

Girilen iki sayının ortalamasını hesaplayan ve sonucu gösteren programı yazınız

```
const sayi1 = prompt("1. sayıyı giriniz:");  
const sayi2 = prompt("2. sayıyı giriniz:");  
var ortalama = (Number(sayi1) + Number(sayi2))/2;  
  
console.log(ortalama);
```