

# Java Dilinde Constant Değişkenler, Tür Dönüştürme, Nümerik Veri Tipleri, Özel Karakterler

## Constant(Sabit) Değişken Tanımlamak

Constant değişkenlerin değeri değişmez

Constant değişkenler büyük harfle yazılır

**final** keyword alan her değişken bir constant değişken olur

Constant değişkenlerin kullanılmasının nedeni, program boyunca değerinin değişmemesinden dolayıdır. Örneğin Pi sayısını constant tanımlamak mantıklı olacaktır.

```
1 public class ComputeArea{
2     public static void main(String[] args){
3         final double PI=3.14159;
4     }
5 }
```

## Nümerik Veri Tipleri

**byte** 1 byte

**short** 2 byte

**int** 4 byte

**long** 8 byte

**float** 4 byte

**double** 8 byte

12L, 12l denilince 12 sayısı **long** olur.

12F, 12f denilince 12 sayısı **float** olur.

12D, 12d denilince 12 sayısı **double** olur.

**Not:** Onluk tabandaki bir sayıyı sekizlik tabanda ekrana yazdırmak için sayının başına **sıfır(0)** yazmak gerekir:

```
System.out.println(012);
```

Bu statement ekrana 10 yazar. Eğer başa sıfır ve x (0X veya 0x) yazılırsa onaltılık tabana göre yazılacaktır. Yani ekrana 18 yazılırdı.

```
System.out.println(0x12);
```

## Tür Dönüştürme(Casting) İşlemleri

Diyelim ki **double** türünden bir sayımız var. Bu sayıyı bir **int** değişkenine atamak istiyoruz. Bunu nasıl yaparız?

```
1 double d=4.5;
2 int i=(int)d; //Bu işleme casting denir.
```

Casting işlemleri range'si yani değer aralığı(boyutu) büyük olan değişkenlerin değer aralığı küçük olan değişkenlere atandığı zaman yapılır.

```
1 int i=1;
2 double d=i; //Burada casting işlemine gerek yoktur.
```

## Karakter Tipi

**char** keyword ile bir karakter değişkeni tanımlarız.

```
1 char letter='A';//Tek tırnak kullanılır
```

### Özel karakterler

\b backspace

\t tab

\n yeni satır

\f Formfeed

\r yeni satır(Mac OS)

\\ backslash

' single quote

"" double quote

```
1 char letter='A';//Tek tırnak kullanılır
2
3 char ch=(char)65.25;//ch değeri 65 olur. .25 değeri kaybedilir.
4 System.out.println(ch);//Ekrana A yazar. Çünkü A harfinin ASCII değeri 65'tir
5
6 int a='A'; //Burada casting'e gerek yoktur. Çünkü int 4 byte yer kaplar 'A' ise 1 byte
```

Formüllerde karakterler kullanıldığı zaman compiler karakterlerin **ASCII** değerini kullanır:

```
1 int b='A'*24+12;
2 System.out.println(b);//Ekrana 1572 yazar
```