

Java Dilinde String Sınıfı

İlk dersimizde String sınıfının önceden tanımlı bir sınıf olduğundan bahsetmiştik. Yani **import** edilmeden kullanılan bir sınıftır.

```
1 String newString=new String("Hello");
2 String message="Hello";
```

Bu iki ifade ile newString ve message isimli iki tane String nesnesi yarattık

```
1 char[] charArray={'H','e','l','l','o'};
2 String message=new String(charArray);
```

message nesnesi yine "Hello" değerine sahip olur.

```
1 String s1="Welcome to Java";
2 String s2=new String("Welcome to Java");
3 String s3="Welcome to Java";
```

s1 ile s3 memory'de aynı yere point ederken, s2 başka yere point eder. Bu nedenle **s1==s3** tür. **s1!=s2** dir

String Sınıfının Önemli Metodları

Öncelikle şunu söyleyelim: Bu kısımda anlatacağ olan metodlar String nesnesi yaratılmadan kullanılamazlar. Yani bu metodlar **static** metod değildir, instance metottur.

equals() metodu

Parametre olarak bir String nesnesi alır.

Not: Çift tırnak içinde yazılı olan değerler bir String nesnesidir. Yani "Hello" bir nesnedir

Bu metod nesnelerin tuttuğu adreslerin eşitliğini kontrol etmez. Nesnelerin tuttuğu adreslerdeki değerlerin eşit olup olmadığını kontrol eder. Eğer eşit ise true; değilse **false** döner.

Nesnelerin tuttuğu adresleri kontrol etmek istersek **==** operatörü yeterli olur.

```
1 String s1="Hello";
2 String s2="Hello";
3 if(s1.equals(s2)){
4     System.out.println("s1=s2");
5 }
```

compareTo() metodu parametre olarak bir String nesnesi alır

equals() metodu ile çalışma mantığı aynıdır. Fakat döndereceği değer, nesneler eşit ise 0 dönderir, değilse **ASCII** olarak hangisi daha önce ise pozitif değer aksi takdirde negatif değer dönderir.

Örnek:

```
1 String s1="Hello";
2 String s2="Hallo";
3 s1.compareTo(s2);
```

4 değeri döner. Çünkü e harfi a harfinden 4 karakter daha ilerdedir.

length() metodu

```
1 String s1="Hello";
2 int uzunluk=s1.length();
```

uzunluk değişkeninin değeri 5 olur.

charAt() metodu

Parametre olarak **int** sayı alır.

```
1 String s1="Hello";
2 char letter=s1.charAt(3);
```

parametre olarak aldığı index değerinde bulunan karakteri dönderir. Bu örneğimizde 3. karakter l olduğu için letter değişkeninin değeri 'l' olur.

concat() metodu

Parametre olarak String nesnesi alır.

Döndereceği değer String nesnesidir.

```
1 String s1="Hello";
2 String s2=" World";
3 String resultString=s1.concat(s2);
```

Bu metod ise iki String nesnesinin değerlerini birleştirir.

resultString nesnesinin değeri "Hello World" olur.