



Strings

Strings are immutable in Java.

Every string method returns a new string, and the original string remains unmodified.

To find the length of a string

```
String name = "Harry";  
  
//int --> name.length()  
System.out.println( name.length() ); //Output: 5
```

Trim a string

```
String name = " _ _ _ Harry _ _ _ ";  
  
//String --> name.trim();  
String trimName = name.trim();  
System.out.println( trimName ); //Output: Harry  
//Remove unwanted spaces at start & end
```

Convert a string to uppercase

```
String name = "Harry";  
  
//String --> name.toUpperCase()  
String upperCaseName = name.toUpperCase();  
System.out.println( upperCaseName ); //Output: HARRY
```

Convert a string to lowercase

```
String name = "Harry";  
  
//String --> name.toLowerCase()
```

```

String lowerCaseName = name.toLowerCase();
System.out.println( lowerCaseName );           //Output: harry
System.out.println( name );                   //Output: Harry
                                                //actual string never changes

```

Substring of a string

```

String name = "Harry";

//String --> name.substring(staringIndex)
System.out.println( name.substring(2) );        //Output: rry

//String --> name.substring(staringIndex, endingIndex)
System.out.println( name.substring(2,4) );        //Output: rr (endIndex not included)

```

Replace a sub-part from a string

```

String name = "Harry";

//String --> name.replace( char, char)
System.out.println( name.replace('r', 'p') );      //Output: Happy

//String --> name.replace( String, String)
System.out.println( name.replace("rry", "mmy") );   //Output: Hammy
System.out.println( name.replace("r", "pp") );        //Output: Happppy

//String --> name.replaceAll( String, String)
System.out.println( name.replaceAll("r", "p") );     //Output: Happy

System.out.println( name.replace('r', "pp") );        //Error

```

Find if a sub-string is present in a string

```

String name = "Harry";
//H a r r y
//0 1 2 3 4

//int --> name.indexOf( String )
System.out.println( name.indexOf("ry") );          //Output: 3

//int --> name.indexOf( char )
System.out.println( name.indexOf('r') );            //Output: 2

//int --> name.indexOf( char, searchstartIndex )
//search is down from left to right
System.out.println( name.indexOf('r',3) );          //Output: 3

//int --> name.lastIndexOf( char )
System.out.println( name.lastIndexOf('r') );        //Output: 3

//int --> name.lastIndexOf( String )
System.out.println( name.lastIndexOf("r") );        //Output: 3

//int --> name.indexOf( char, searchingstartIndex )

```

```
//search is down from right to left  
System.out.println( name.lastIndexOf('r',2) );           //Output: 2  
  
/**If the substring is not present then the output is -1
```

Find a character at a given index from a string

```
String name = "Harry";  
  
//char --> name.charAt(int)  
System.out.println( name.charAt(0) );                  //Output: H  
System.out.println( name.charAt(4) );                  //Output: y  
System.out.println( name.charAt(5) );  
                           //ERROR (Index out of range)
```

Check if two strings are the same or not

```
String name = "Harry";  
  
//Boolean --> str.equals(str1)  
System.out.println( name.equals("Harry") );           //Output: true  
  
//Boolean --> str.equalsIgnoreCase(str2)  
System.out.println(name.equalsIgnoreCase("haRRy"));  //Output: true  
  
//int --> A.compareTo(B)  [Result = A - B]  
System.out.println(name.compareTo("Barry"));          //Output: 6
```

Repeat a single string multiple times

```
String name = "Harry";  
  
//String --> name.repeat(repeatCount);  
System.out.println(name.repeat(4));  
                           //Output: HarryHarryHarryHarry
```

Converting a character array to a string

```
char[] charArray = {'H','a','r','r','y'};  
  
//String --> String.valueOf(charArray)  
String name = String.valueOf(charArray);  
System.out.println(charArray);      //Output: Harry
```

Changing a character to a string

```
char ch = 'a';  
  
//String --> Character.toString( char )  
System.out.println( Character.toString(ch) ); //Output: a
```

Joining a string array to a single string

```
String[] strArray = {"I", "love", "my", "life"};  
  
//String --> String.join(delimiter, strArray)  
String sentence = String.join(" ", strArray);  
System.out.println(sentence);  
//Output: I love my life
```

Concatenate 2 strings

```
String s1 = "Hello ", s2 = "everyone";  
  
System.out.println(s1.concat(s2));  
//Output: Hello everyone
```