

## Chapter 3 - Strings

A string is a sequence of characters

A string is instantiated as follows:

```
String name;  
name = new String("Harry");
```

String is a class but can be used like a data type:

[Strings are immutable and cannot be changed]

```
String name = "Harry";  
      Reference      Object
```

Different ways to print in Java

We can use the following ways to print in Java:

1. `System.out.print()` → No newline at the end!
2. `System.out.println()` → Prints a new line at the end
3. `System.out.printf()`
4. `System.out.format()`

`System.out.printf("%c", ch)`

→

%d	for int
%f	for float
%c	for char
%s	for string

### String Methods

String methods operate on Java Strings. They can be used to find length of the string, convert to lowercase, etc.



Some of the commonly used String methods are:

String name = "Harry";  
<sup>0 1 2 3 4</sup>

- 1> name.length() → Returns length of String name.  
(5 in this case)
- 2> name.toLowerCase() → Returns a new String which has all the lowercase characters from the String name.
- 3> name.toUpperCase() → Returns a new String which has all the uppercase characters from the String name.
- 4> name.trim() → Returns a new String after removing all the leading and trailing spaces from the original String.
- 5> name.substring(int start) → Returns a substring from start to the end. substring(3) returns "ry"  
[Note that index starts from 0]
- 6> name.substring(int start, int end) → Returns a substring from start index to the end index. Start index is included and end is excluded
- 7> name.replace(char, char) → Returns a new String after replacing r with p. Happy is returned in this case.



8. `name.startsWith("Ha")` → returns true if name starts with string "Ha". true in this case!  
String
9. `name.endsWith("ry")` → returns true if name ends with string "ry". true in this case.  
String
10. `name.charAt(2)` → returns character at a given index position. r in this case!  
int
11. `name.indexOf(s)` → returns the index of the given string.  
String For ex: `name.indexOf("ar")` returns 1 which is the first occurrence of ar in string "Harry", -1 otherwise
12. `name.indexOf("s", 3)` → returns the index of the given string starting from the index 3 (int). -1 is returned in this case!
13. `name.lastIndexOf("r")` → returns the last index of the given string. 3 in this case!
14. `name.lastIndexOf("r", 2)` → returns the last index of the given string before index 2.
15. `name.equals("Harry")` → returns true if the given string is equal to "Harry" false otherwise [Case Sensitive]



16 `name.equalsIgnoreCase("harry")` → returns true if two strings are equal ignoring the case of characters.

### Escape Sequence Characters

Sequence of characters after backslash '\'  
= Escape sequence characters

Escape sequence characters consist of more than one characters but represents one character when used within the strings.

Examples: `\n`, `\t`, `\'`, `\"`, etc.

newline    Tab    single quote    backslash