String and String Templates in Kotlin

In this tutorial, you will learn string and string templates along with examples. Don't forget to check out important properties and functions of string.

String in Kotlin

String is a sequence of characters enclosed in double quotation marks or triple quotes. Similar to Java, Strings are immutable in Kotlin. You can declare the string by following the below syntax-

```
val variablename = "character"
OR
var variablename = "character"
```

Example

```
val str = "Hello, World"
```

Also, if you want you can specify the type `String` while declaring a variable. And initialize the variable in another statement.

```
val str: String
...
str = "Hello Kotlin"
```

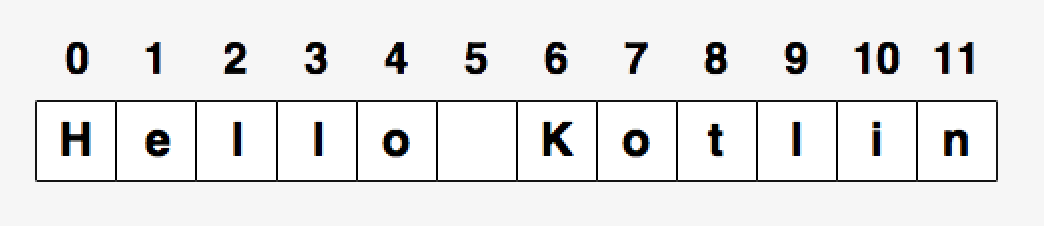
> **Note-** When you create a string using double quotation marks, then you must escape special characters otherwise you will get a compile-time error.
> When you create a string using triple quotes, then it becomes a **raw string**. In raw string, no escaping is required and you can write all character.

How to access individual characters of a string?

With the help of index access operator([]), you can access individual characters of a string. Remember one thing, string follows zero based indexing which means indexing starts from 0 not 1. Let's see how we can do this by looking at the below example-

```
val str = "Hello Kotlin"
val s = str[6]
```

In the above example, variable `s` contains `K` which seventh character of the string `str`. Complete indexing of string `str` is given below for better understanding-



0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		K	o	t	l	i	n

How to iterate through String in Kotlin?

If you are looking for an easy way to loop through elements of a string then use for loop.

```
fun main(args: Array){
   val str = "Hello, World!"

   for(element in str){
      println(element)
   }
}
```

Output

```
H
e
l
l
o
,

W
o
r
l
d
!
```

Escape Sequences in Kotlin

Following are the escape sequences supported by Kotlin-

Escape Sequences	Description
\n	Used to insert newline.
\r	Used to insert carriage return.
\t	Used to insert tab.

Escape Sequences	Description
\b	Used to insert backspace.
\"	Used to insert double quote.
\'	Used to insert single quote.
\\	Used to insert backslash.
\$	Used to insert dollar.

Example of Raw String

If a string contains newlines without writing \n and arbitrary string, then it is called raw string. It is placed in triple quotes(""").

```
fun main(arr: Array<String>){
        val str = """
        i=0
        while(i<=5){
                println(i)
        }
        """
        println(str)
}
```

On running the above program, you will get the below output-

```
i=0
while(i<=5){
   println(i)
}
```

String Templates in Kotlin

Kotlin introduced a new concept which is called string templates using which you can embed a variable or expression inside a string without string concatenation.

Syntax of string templates

```
$variablename
or
${expression}
```

Example of string templates

```
fun main(arr: Array<String>){
   val lang = "Kotlin"
   val str2 = "Hello " + lang //Old way of doing string concatenation
   val str1 = "Hello $lang" //Kotlin way of doing string concatenation
   println(str1) //it will print Hello Kotlin
   println(str1) //it will also print Hello Kotlin
}
```

On running the above code, you will get the below output-

```
Hello Kotlin
Hello Kotlin
```

As you can see above, string template is a way to make the string concatenation more simple and readable. You can even use an expression with string template.

```
fun main(arr: Array<String>){
   val lang = "Kotlin"
   val str = "The word Kotlin has ${lang.length} characters."
   println(str)
}
```

On running the above code, you will get the below output-

```
The word Kotlin has 6 characters.
```

Important Properties and Functions of String

Strings are objects of String class. And because of this, string has properties and methods that are explained below-

Properties/Functions	Description with Examples

length	It is a property and it returns the number of characters present in the string.
	`val str = "Hello Kotlin"`
	`println(str.length)`

Properties/Functions	Description with Examples

get(index)	It is a function that return takes Int as an argument and returns the character at the specified index.
	`val str = "Hello Kotlin"`
	`println(str.get(4)) //it will print o`

subSequence(startIndex, endIndex)	It is a function that takes two argument- startIndex and endIndex. It returns the subsequence of string between startIndex and endIndex, but excluding character at the endIndex.
	`val str = "Hello Kotlin"`
	`println(str.subSequence(0,4)) //it will print Hell`

compareTo(string)	It is a function that takes another string as an argument. It compares the two string and returns 0 if both are equal.
	`val str1 = "Hello Kotlin"`
	`val str2 = "Hello Kotlin"`
	`println(str1.compareTo(str2))`