

Kotlin Strings

Kotlin strings represent character strings. String literals are represented using double quotes. Kotlin Strings are immutable, any operation that modifies a string creates a new copy instead of changing the existing one.

String Concatenation

Kotlin Strings are concatenated using the + operator.

```
val x = "My name is"
val name = "Mary"
println(x + " " + name)
//output
My name is Mary
```

Kotlin String Indexing

Since a string is a sequence of characters we can obtain individual characters from a string using the index operator []. The first character is index 0 and the second one index 1... etc

```
val text = "Banana"
println(text[0]) //Prints out "B"
println(text[1]) //Prints out "a"
println(text[2]) //Prints out "n"
println(text[3]) //Prints out "a"
println(text[4]) //Prints out "n"
println(text[5]) //Prints out "a"
```

The inbuilt functions `.first()` and `.last()` respectively return the first and last String characters.

Kotlin String Interpolation

String interpolation is substituting a variable with its value in a string. We use the \$ symbol to interpolate strings.

```
val name = "Janet"
val year = "1965"

val sentence = "$name was born in $year"
println(sentence)
//output
"Janet was born in 1965"
```

String Comparison

We use the == operator or `.equals()` method to compare the contents of Strings

```

val message = "PA4782DHUE. Confirmed"
val message2 = "PA4782DHUE. Confirmed"
if(message == message2){
    println(true)
}
//true

message = "duwygryuety"
message2 = "e8273gd27"
if(message.equals(message2)){
    println(true)
}
else{
    println(false)
}
//false

```

String Case

Kotlin has inbuilt functions to manipulate the String case.

```

val x = "african inventor"
println(x.toUpperCase())
println(x.toLowerCase())
println(x.capitalize())
println(x.decapitalize())

//output
AFRICAN INVENTOR
african inventor
African inventor
african inventor

```

Kotlin Empty/ Blank Strings

Empty strings have no characters. Blank strings have whitespace characters. Kotlin has inbuilt functions to check for empty and blank strings.

```
val emptyString = ""
val blankString = " "
println(emptyString.isBlank())
println(emptyString.isEmpty())
println(blankString.isBlank())
println(blankString.isEmpty())

//output
true
true
true
false
```

Trimming white spaces from Strings

Kotlin has inbuilt functions to trim whitespaces from the start/ end or both sides of a string.

```
val name = " Emma Museveni "
println(name.trimStart())
println(name.trimEnd())
println(name.trim())

//output
"Emma Museveni "
" Emma Museveni"
"Emma Museveni"
```

String startsWith()/ endsWith()

Kotlin has inbuilt functions to check the character than a string begins or ends with.

```
val location = "Eldoret"
println(location.startsWith("E"))
print(location.endsWith("f"))

//Output
true
false
```

Kotlin String replace()

We can use the `replace()` function to replace all occurrences of a string with another

```
val text = "Janet Museveni is a Ugandan cabinet minister. " +  
    "She has served the Ugandan people for 5 years"  
println(text.replace("Ugandan", "African"))  
  
//Output  
Janet Museveni is a African cabinet minister. She has served the  
African people for 5 years
```

Kotlin String split()

The `split()` function is used to cut a string into a list of strings based on the delimiter

```
val word = "eagle,falcon,hawk,owl"  
val birds = word.split(",")  
println(birds)  
println(birds[1])  
  
//output  
[eagle, falcon, hawk, owl]  
falcon
```

Kotlin toString()

This function is used to convert an object of another type to a string

```
val weight = 54.5  
print("My weight is "+ weight.toString())  
  
//output  
My weight is 54.5
```