

Kotlin Flow Control I

if...else Statements

if...else statements have this syntax

```
if(condition){
    //code to run if condition is true
}
else{
    //code to run if condition is false
}
```

The else block is optional but if present the code inside it runs if the expression in the if statement is false e.g

```
val age = 15
if(age > 12){
    println("You are a teenager")
}
else{
    println("You are not a teenager")
}
//You are a teenager
```

if...else statements can also be nested

if...else if...else

When there are more than 2 branches in a decision we can use an if...else if...else statement

```
var nationality = "Ugandan"
if (nationality=="Kenyan"){
    println("Have you ever been to Nairobi?")
}
else if(nationality=="Ugandan"){
    println("Have you ever been to Kampala?")
}
else if(nationality=="Rwandese"){
    println("Have you ever been to Kigali?")
}
else{
    println("Have you ever been to East Africa?")
}
```

The else block in an if...else if...else ladder gets executed if none of the previous conditions are met.

When Expression

The when statement is an elegant replacement for the if...else if else ladder

```
var nationality = "Rwandese"
when(nationality){
    "Ugandan"-> println("Have you ever been to Kampala?")
    "Rwandese"-> println("Have you ever been to Kigali?")
    "Kenyan"-> println("Have you ever been to Nairobi?")
    else-> println("Have you ever been to East Africa?")
}
```

The when statement matches its parameter against each branch until the specified condition is met, otherwise the else block is executed. It is also possible to combine branch conditions separated by commas like so

```
fun selectContainer(litres: Int){
    when(litres){
        1,2,3-> println("Pour the water into a bottle")
        4,5,6,7-> println("Pour the water into a bucket")
        8,9 -> println("Pour the water into a big bucket")
        else-> println("Pour the water into a jerrycan")
    }
}

selectContainer(4) //Pour the water into a bucket
```

For Loop

The for loop is used to iterate over collections like lists, arrays, strings and ranges. The basic syntax is

```
for(item in collection){
    //do something
}
```

e.g

```
var languages = arrayOf("Kotlin", "Python", "Java", "Ruby")
for (language in languages){
    println(language)
}
```

```
for (n in 1..10){  
    println(n)  
}
```

```
var name = "Alexandra"  
for (character in name){  
    println(character)  
}
```