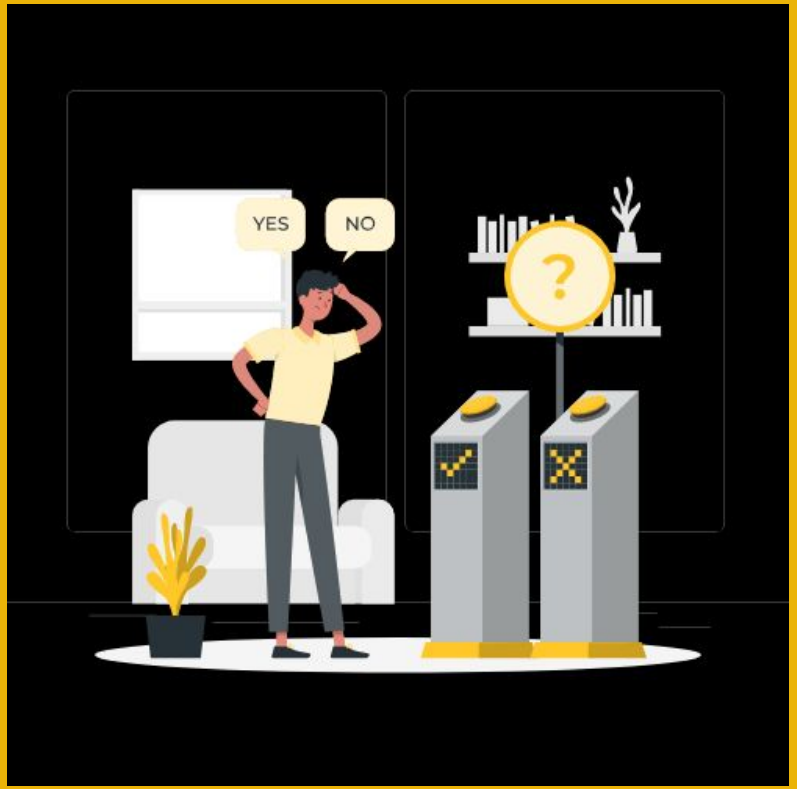
MSIB
magang dan studi independen bersertifikat

INFINITE
LEARNING



Control Flow

How to control the flow
of a program

INFINITE LEARNING

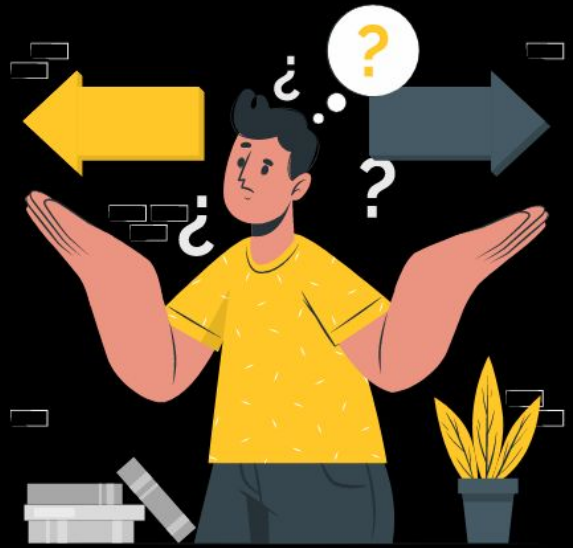
Control Flow

Control flow is a way to control the flow of a program based on conditions when the program is running.

- Expressions & Statements
- When Expressions
- Range
- For Loop & While Loop
- Break & Continue

INFINITE
LEARNING

Expressions & Statements



if…else Expression

INFINITE LEARNING

- In Kotlin, **if** is an expression that can return a value, so the result can be stored as a variable.
- if can be used to test a condition for running a process.
- So let's start with traditional **if...else** statement.

```
val condition = true

if (condition) {
    println("code block to be executed if condition is true")
} else {
    println("code block to be executed if condition is false")
}
```

if…else Expression

INFINITE LEARNING

If in Kotlin can also be used as an expression. Let's try using an if statement that can return a value to a variable.

And now let's try for if...else expression:

```
fun main() {

    val timeClose = 8
    val timeNow = 8

    val isClosed = if(timeNow >= timeClose) "Class already closed" else "Class is open"

    println(isClosed)

}
```

if…else Expression

INFINITE LEARNING

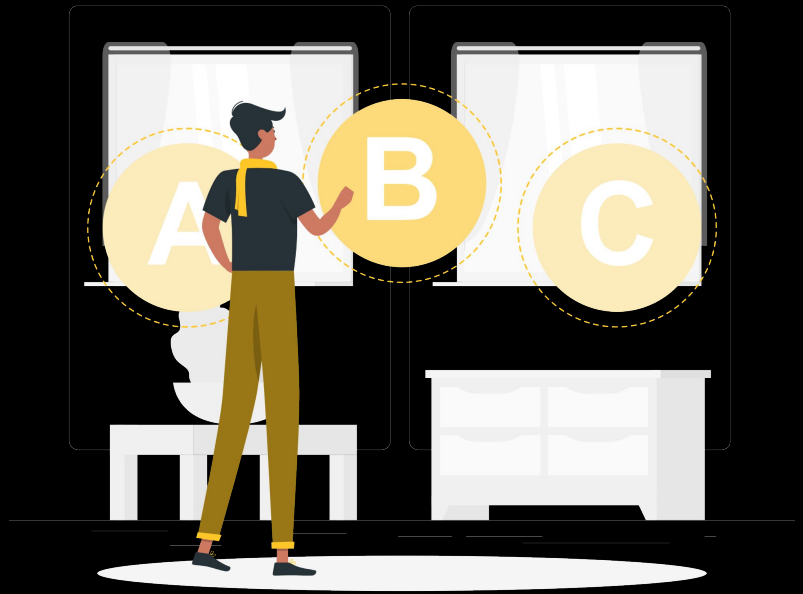
We can use else if condition to specify a new condition if the first condition is false.

Let's code!

```
fun main() {

    val timeOpen = 8
    val timeClose = 12
    val timeNow = 7

    val classStatus = if(timeNow >= timeClose) {
        "Class already closed"
    } else if(timeNow >= timeOpen) {
        "Class is open"
    } else {
        "Class is not open yet"
    }

    println(classStatus)

}
```

INFINITE
LEARNING

When Expression

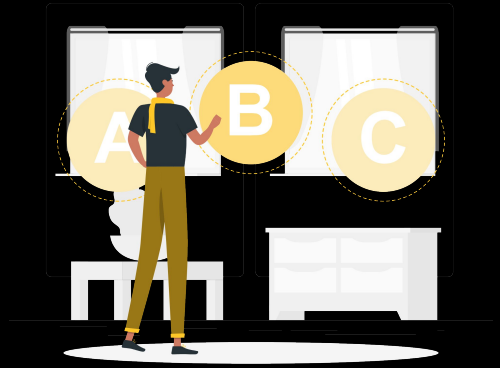


INFINITE LEARNING

When Expression

Though you can use if..else if expression to handle the situation, Kotlin provides when expression to handle the situation in nicer way.

Far easy and more clean.



When Expression

INFINITE LEARNING

- **When** is a mechanism for matches argument against all branches sequentially until some branch condition is satisfied.
- Inside when can also add *else* branch.
- *else* will be evaluated if no one of the conditions are match on the previous branch.
- *else* is mandatory if using a when expression to return a value.

Let's code it!

```
fun main() {

    val day = 1

    val result = when (day) {
      1 -> "Monday"
      2 -> "Tuesday"
      3 -> "Wednesday"
      4 -> "Thursday"
      5 -> "Friday"
      6 -> "Saturday"
      7 -> "Sunday"
      else -> "Invalid day."
    }

    println(result)

}
```

When Expression

INFINITE LEARNING

when can combine multiple conditions into a single condition.

Let’s code it!

```
fun main() {

    val day = 2

    when (day) {
      1, 2, 3, 4, 5 -> println("Weekday")
      else -> println("Weekend")
    }

}
```

When Expression

INFINITE LEARNING

If running two or more lines of code on each branch, put them in curly braces.

Let's code it!

```
fun main(args: Array<String>) {
   val day = 2

   when (day) {
     1 -> {
        println("First day of the week")
        println("Monday")
     }
     2 -> {
        println("Second day of the week")
        println("Tuesday")
     }
     3 -> {
        println("Third day of the week")
        println("Wednesday")
     }
     4 -> println("Thursday")
     5 -> println("Friday")
     6 -> println("Saturday")
     7 -> println("Sunday")
     else -> println("Invalid day.")
   }
}
```

INFINITE
LEARNING

Range

Range ..

INFINITE LEARNING

Kotlin lets you easily create ranges of values using the **rangeTo()** function from the kotlin.ranges package and its operator form **...** Usually, rangeTo() is complemented by **in** or **!in** functions.

```
val rangeInt = 1..10
```

Similar to the code below, but cannot use existing functions or properties in Range

```
val rangeInt = intArrayOf(1,2,3,4,5,6,7,8,9,10)
```

Range .., .rangeTo()

INFINITE LEARNING

Let's code it!

Using ..

```
fun main() {

    val rangeInt = 1..10
    println("Step: " + rangeInt.step)
    print(rangeInt.toList())

}
```

Using .rangeTo()

```
fun main() {

    val rangeInt = 1.rangeTo(10)
    println("Step: " + rangeInt.step)
    print(rangeInt.toList())

}
```

Range step

INFINITE LEARNING

- The distance between the two included values is determined by the step.
- By default, step is 1.
- To get the *step*, you can use the step property.

Let's code it!

```
fun main() {

    val rangeInt = 1..10 step 2
    println("Step: " + rangeInt.step)
    print(rangeInt.toList())

}
```

rangeTo() & downTo()

INFINITE LEARNING

To check whether a value is present or not in the range of values, you can use **in** or **!in**.

Let's code it!

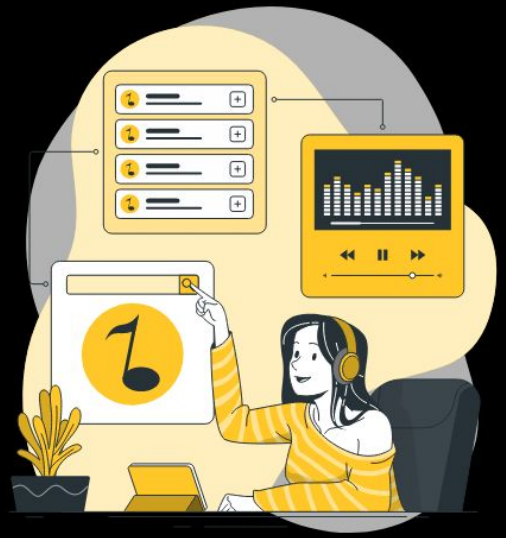
```
fun main() {
    var i = 4
    if (i in 1.rangeTo(10)) { // equivalent of i >= 1 && i <= 10
        println("Value 4 available in Range")
    }
}
```

```
fun main() {
    var i = 20
    if (i !in 10.downTo(1)) {
        println("Value 4 is not available in Range")
    }
}
```

INFINITE LEARNING

For Loop & While Loop

The ***for*** *loop* iterates through anything that provides an iterator.



This is equivalent to the *foreach* loop in languages like C#.

For Loop

INFINITE LEARNING

For loop is the concept of looping on the same block as long as the results of the evaluation conditions are met or are **true**.

For loop can be used on Ranges, Collections, Arrays and anything that provides an iterator

```
for (i in 1..7){
    println("Infinite Learning")
}
```

```
fun main() {
    println("Start")
    println("Infinite Learning")
    println("Infinite Learning")
    println("Infinite Learning")
    println("Infinite Learning")
    println("Infinite Learning")
    println("Infinite Learning")
    println("Infinite Learning")
    println("Finish!")
}
```

For Loop

INFINITE LEARNING

The **for loop** can be performed using a **range expression**

```
fun main() {

    for (i in 1..20){
        println("Value is $i")
    }

}
```

Can also use a for loop to iterate through the **array**

```
fun main() {

    val days = arrayOf("Monday","Tuesday","Wednesday",
                       "Thursday", "Friday",
                       "Saturday", "Sunday")
    for (day in days){
        println(day)
    }

}
```

Let's code it!

For Loop - withIndex()

INFINITE LEARNING

Access the index for each element in Ranges using the **withIndex()** function

Let's code it!

```
fun main() {

    val days = arrayOf("Monday","Tuesday","Wednesday",
                       "Thursday", "Friday",
                       "Saturday", "Sunday")
    for ((index, value) in days.withIndex()) {
        println("value $value with index $index")
    }

}
```

For Loop - forEach

INFINITE LEARNING

In addition to using the *for* keyword, for make the loop process can use the **forEach** extension.

Let's code it!

```
fun main() {

    val days = arrayOf("Monday","Tuesday","Wednesday",
                       "Thursday", "Friday",
                       "Saturday", "Sunday")
    days.forEach { value ->
        println("Day is $value")
    }

}
```

For Loop - forEachIndexed

INFINITE LEARNING

In the previous code *forEach* is a lambda expression that has only one argument which is a **single value** enclosed in ranges.

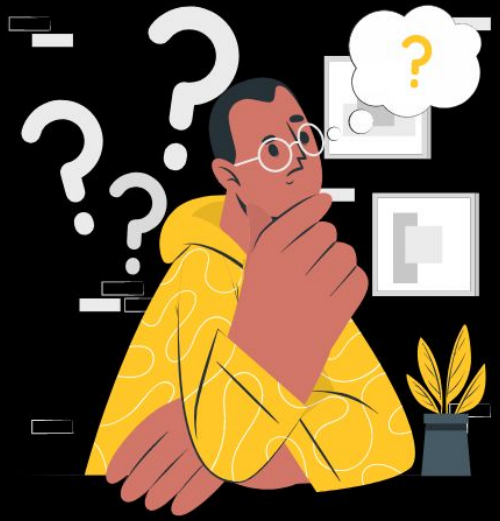
If you want to get the index of each included value you can use the **forEachIndexed** extension.

```
fun main() {

    val days = arrayOf("Monday","Tuesday","Wednesday",
                       "Thursday", "Friday",
                       "Saturday", "Sunday")
    days.forEachIndexed {  index, value ->
        println("Day $value with index: $index")
    }

}
```

Let’s code it!

INFINITE LEARNING

Any question?



While & Do While

INFINITE LEARNING

- *While* loop is very flexible
- To make a **loop** with *while*, use the keyword **while**
- While checks the **condition**, if the **condition** evaluates to **true**, it will execute the while **block**

keyword **while**

while **condition**

```
fun main() {
    while (condition) {
        // body of the loop
    }
}
```

while loop **body**

While & Do While

INFINITE LEARNING

The code block is **repeated** until the while condition evaluates to **false**

Let's code it!

```
fun main() {
    var i = 0

    while (i < 10) {
        println("Loop to : $i")
        i++
    }
}
```

While & Do While

INFINITE LEARNING

The block code will be executed **first**, then the condition **check** will be carried out at the **end**

Let's code it!

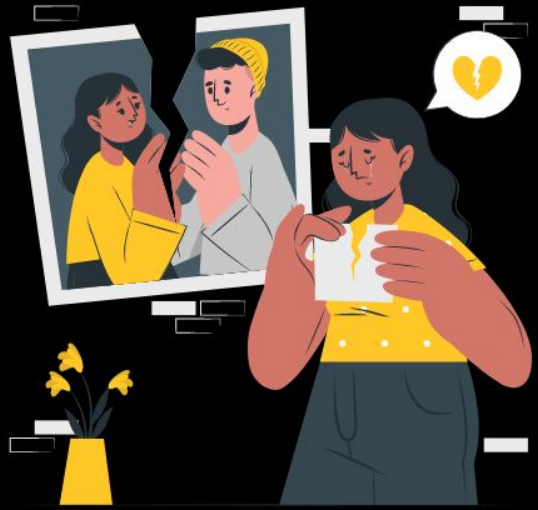
```
fun main() {
    var i = 0
    do {
        println("$i Kampus Merdeka")
        i++
    } while (i <=8)
}
```

Run: MainKt

```
1 Kampus Merdeka
2 Kampus Merdeka
3 Kampus Merdeka
4 Kampus Merdeka
5 Kampus Merdeka
6 Kampus Merdeka
7 Kampus Merdeka
8 Kampus Merdeka

Process finished with exit code 0
```

INFINITE LEARNING

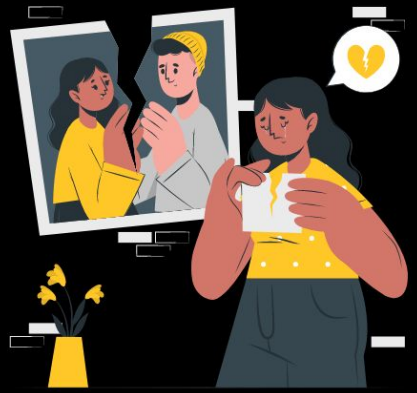
Break & Continue



Break

INFINITE LEARNING

- When doing a loop and want to skip or stop the loop, for example if the resulting value is null? can use **Break** and **Continue**.
- **Break** is used to **stop** the iteration process.
- To **stop** a loop with the keyword **Break.**



Break

INFINITE LEARNING

Let's code it!

```
fun main() {
    var i = 0
    while (true) {
        println("Break $i")
        i++
        if (i > 500) {
            break
        }
    }
}
```

```
fun main() {

    val c = 2
    for (b in 1..10) {

      println("Sayang...")

      if(b == c){

        println("Putus!")

        break
      }
    }

}
```

Continue

INFINITE LEARNING

- **Continue** is used to **stop** the loop that is running, and **immediately continue** to the next loop
- To skip a loop with the keyword Continue

```kotlin
fun main() {
    val listInt = listOf(1, 2, null, 4, 5, null, 7)

    for (i in listInt) {
        if (i == null) continue
        print(i)
    }
}
```

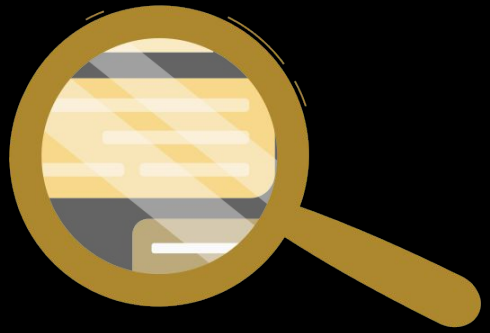
```kotlin
fun main() {
    println("Print odd numbers")
    for (i in 1 ≤ .. ≤ 20) {
        if (i % 2 == 0) {
            continue
        }
        println("$i is an odd number")
    }
}
```

GUIDING RESOURCE

INFINITE LEARNING

Guiding Resources:

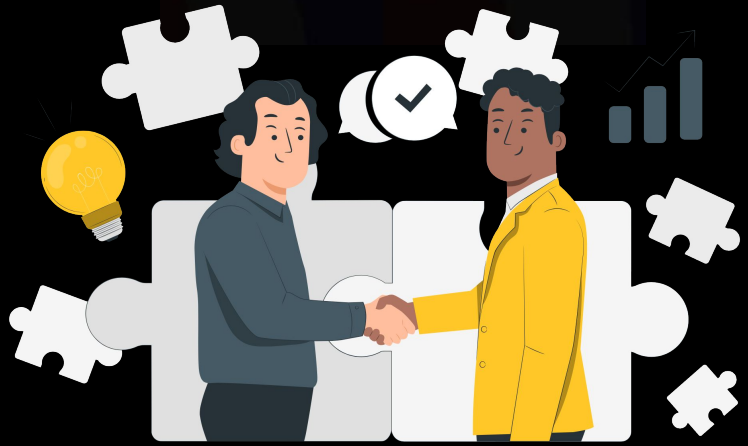
1. https://kotlinlang.org/docs/control-flow.html
2. https://kotlinlang.org/docs/ranges.html#type-checks-and-automatic-casts
3. https://www.tutorialspoint.com/kotlin/kotlin_control_flow.htm



Design Asset:

https://storyset.com

INFINITE
LEARNING



Cheers

@infinitelearning_id

Infinite Learning Indonesia

infinitelearning_id