



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA:

Sentencias de Iteración, Control y Funciones en Kotlin y Swift

PRESENTADO POR:

Valenzuela Esparza Javier Ivan

GRUPO:

9B

MATERIA:

Desarrollo de aplicaciones Inteligentes

PROFESOR:

Ray Brunett Parra Galaviz

FECHA:

02/10/2024

Sentencias de Iteración, Control y Funciones en Kotlin y Swift

Sentencias de Iteración.

1. **for loop:**

Se utiliza para recorrer rangos, arrays o listas.

```
for (i in 1..5) {  
    println(i)  
}
```

2. **while loop**

Ejecuta el bloque de código mientras una condición sea verdadera.

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

3. **do-while loop:**

Ejecuta el bloque de código al menos una vez y luego evalúa la condición.

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

Swift:

1. for loop:

Permite iterar sobre rangos, listas o diccionarios.

```
for i in 1...5 {  
  
    print(i)  
  
}
```

2. while loop:

Ejecuta el código mientras la condición sea verdadera.

```
var i = 0  
  
while i < 5 {  
  
    print(i)  
  
    i += 1  
  
}
```

3. repeat-while loop:

Similar al do-while de Kotlin, se ejecuta al menos una vez.

```
var i = 0  
  
repeat {  
  
    print(i)  
  
    i += 1  
  
} while i < 5
```

Sentencias de Control

Kotlin

1. if/else:

Sentencia condicional.

```
val a = 5
```

```
if (a > 3) {
```

```
    println("Mayor que 3")
```

```
} else {
```

```
    println("Menor o igual a 3")
```

```
}
```

2. when:

Es equivalente al switch en otros lenguajes.

```
val x = 2
```

```
when (x) {
```

```
    1 -> println("Es 1")
```

```
    2 -> println("Es 2")
```

```
    else -> println("No es ni 1 ni 2")
```

```
}
```

3. try/catch:

Manejo de excepciones.

```
try {  
    val result = 10 / 0  
} catch (e: ArithmeticException) {  
    println("Error: División por cero")  
}
```

Swift:

1. if/else:

Sentencia condicional.

```
let a = 5
```

```
if a > 3 {
```

```
    print("Mayor que 3")
```

```
} else {
```

```
    print("Menor o igual a 3")
```

```
}
```

2. switch:

Sentencia condicional múltiple.

```
let x = 2
```

```
switch x {
```

```
case 1:
```

```
    print("Es 1")
```

```
case 2:
```

```
    print("Es 2")
```

```
default:
```

```
    print("No es ni 1 ni 2")
```

```
}
```

3. **do/catch:**

Manejo de excepciones.

```
do {  
    try someFunctionThatThrows()  
} catch {  
    print("Error capturado")  
}
```

Funciones.

Kotlin:

Las funciones se declaran utilizando la palabra clave fun.

```
fun sumar(a: Int, b: Int): Int {  
    return a + b  
}
```

```
val resultado = sumar(5, 3)
```

```
println(resultado) // Output: 8
```

Swift:

Las funciones se declaran usando func.

```
func sumar(a: Int, b: Int) -> Int {  
    return a + b  
}
```

```
let resultado = sumar(a: 5, b: 3)
```

```
print(resultado) // Output: 8
```

Fuentes:

Cómo crear y usar funciones en Kotlin | Android Developers. (2024). Android Developers.

<https://developer.android.com/codelabs/basic-android-kotlin-compose-functions?hl=es-419#0>

<https://www.facebook.com/grokkeepcoding>. (2022, June). *Sentencia de control switch en Swift.*

KeepCoding Bootcamps. <https://keepcoding.io/blog/sentencia-de-control-switch-en-swift/>