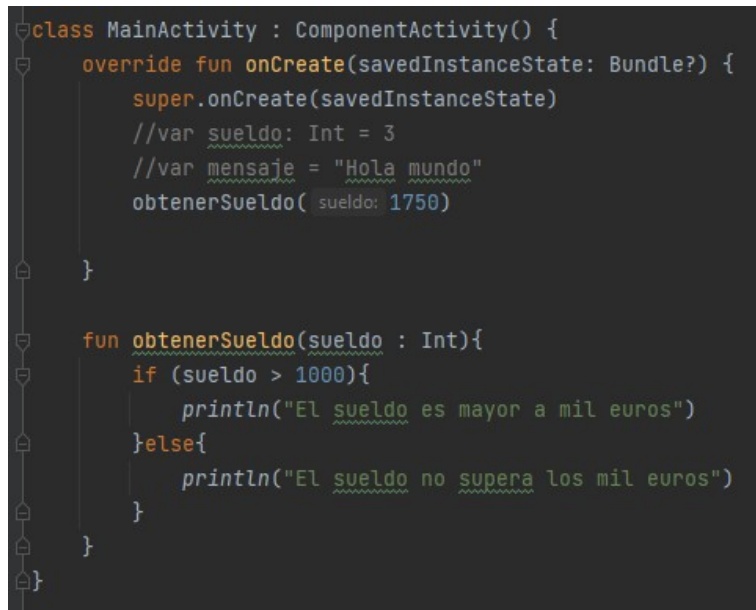


WHEN

La secuencia when es algo así como un switch case:

```
fun main() {  
    val fecha = "05/10/2013"  
    var mes = fecha.subSequence(3,5).toString().toInt()  
  
    //println(mes)  
  
    when(mes){  
        12, 1, 2 -> println("Estamos en invierno")  
        3, 4, 5 -> println("Estamos en primavera")  
        6, 7, 8 -> println("Estamos en verano")  
        9, 10, 11 -> println("Estamos en otoño")  
        else -> println("La fecha es errónea")  
    }  
}
```

FUNCIONES CON PARÁMETROS



```
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        //var sueldo: Int = 3  
        //var mensaje = "Hola mundo"  
        obtenerSueldo( sueldo: 1750)  
    }  
  
    fun obtenerSueldo(sueldo : Int){  
        if (sueldo > 1000){  
            println("El sueldo es mayor a mil euros")  
        }else{  
            println("El sueldo no supera los mil euros")  
        }  
    }  
}
```

FUNCIONES QUE DEVUELVEN UN VALOR

```
fun esPositivo(valor : Int) : Boolean{  
    if (valor >= 0){  
        return true  
    }  
    return false  
}
```

```
fun esPositivo(valor : Int) : Boolean{  
    return valor >= 0  
}
```

ARRAYS

```
var estaciones: Array<String> = arrayOf("Invierno", "Primavera", "Verano", "Otoño")  
  
println(estaciones[2])
```

```
var estaciones: Array<String> = arrayOf("Invierno", "Primavera", "Verano", "Otoño")  
  
for (i in estaciones){  
    println(i)  
}
```

```
var estaciones: Array<String> = arrayOf("Invierno", "Primavera", "Verano", "Otoño")  
  
for (i in 0 ..< (estaciones.size - 1)){  
    println(estaciones[i])  
}
```

```
var estaciones: Array<String> = arrayOf("Invierno", "Primavera", "Verano", "Otoño")  
  
for (i in (0 ..< estaciones.size)){  
    println(estaciones.get(i))  
}
```

SET Y MUTABLESET

```
var clientesVIP : Set<Int> = setOf(123, 265, 324, 605)
var mezclado = setOf(2, 567, "Hola", 'd')
```

Con MutableSet puedes añadir o eliminar elementos, con set no.

Con list y MutableList ocurre lo mismo.

```
I [Juan, Luis, María, Pedro]
I [Juan, Luis, María, Pedro, Jesús]
I [Juan, María, Pedro, Jesús]
```

Maps

```
var mapaPaises: Map<Int, String> = mapOf(  
    1 to "España",  
    2 to "Mexico",  
    3 to "Francia",  
    4 to "Argentina",  
    5 to "Italia"  
)  
  
println(mapaPaises)
```

```
I {1=España, 2=Mexico, 3=Francia, 4=Argentina, 5=Italia}
```

MutableMaps

```
var inversiones = mutableMapOf<String, Float>()  
  
inversiones.put("Coca-Cola", 50f)  
inversiones.put("Amazon", 35f)  
inversiones.put("Bitcoin", 100f)  
  
println(inversiones)
```

```
I {Coca-Cola=50.0, Amazon=35.0, Bitcoin=100.0}
```

```
inversiones.remove(key: "Amazon")  
println(inversiones)
```

```
I {Coca-Cola=50.0, Bitcoin=100.0}
```

EJEMPLO CON BUCLE Y OTRA LISTA

```
var bolsa : List<String> = listOf("Coca-cola", "Amazon", "Bitcoin", "Pfizer", "Repsol", "Enagas", "Ferrovial")
var inversiones = mutableMapOf<String, Float>()

var saldo = 300f
var empresa : String?
var indice : Int = 0
var cantidadInvertir : Float = 90f

while (saldo >= cantidadInvertir){
    empresa = bolsa.elementAtOrNull(indice)
    if (empresa != null){
        inversiones[empresa] = cantidadInvertir // inversiones.put(empresa, cantidadInvertir)
        saldo -= cantidadInvertir
        println("Se ha invertido" + cantidadInvertir + "€ en " + empresa)
        indice++
    }
    else break
}

println(inversiones)
```