



UNIVERSIDAD
DE COLIMA

Kotlin

Integrantes

Diego de Jesús Villaseñor Villa

Jose Enrique Ruvalcaba Segura

Mariana Guadalupe Fajardo Toscano

Angel Isaac Bejarano Flores

Jonathan Solis Ceballos




Variables

Kotlin distingue dos tipos distintos de variables: las fijas, que solo se pueden leer, se introducen con `val`; las otras variables, cuyo valor puede alterarse más adelante, se introducen con `var`.

Es importante aclarar que Kotlin utiliza un tipado estático.

```
val name = "Clara Oswald"  
var age = 22
```

Al contrario del nombre, que es inalterable, la edad se puede ajustar, por ejemplo, en una función.





Constantes

Una constante no puede cambiar durante la ejecución del programa (lo mismo hacen los valores), pero aparte de ello, su asignación debe ser constante.

Para declararla se usa `const val`

Por cierto, una constante solo puede ser declarada dentro de un objeto o en la parte superior de una clase.

```
object constantes{  
    const val PI = 3.1416  
}  
println(constantes.PI)
```





Constantes

La diferencia de la constante con val es que su asignación debe ser constante. Lo siguiente es incorrecto:

```
object constantes{  
    const val PI = obtenerPI()  
    fun obtenerPI() = 3.1416  
}  
println(constantes.PI)
```

En este caso, para obtener la constante, estamos llamando a una función. Y aunque esa función devuelve un solo valor, la misma podría devolver otros valores, es decir, nada asegura que esa función siempre devolverá el mismo valor.





Tipos básicos

Kotlin trabaja con distintos tipos de variables y clases. Cada tipo constituye un objeto, y en este punto Kotlin se diferencia un poco de Java. Aquí todos los tipos ya son realmente objetos.



Números enteros

- Long: 64 Bit
- Int: 32 Bit
- Short: 16 Bit
- Byte: 8 Bit

```
val myNumber: Long = 40  
val myLong= myInt.toLong()
```

Números de punto flotante

- Double: 64 Bit
- Float: 32 Bit

```
val myDouble = 45.78  
val myFloat = 34.43F
```





Conversiones

Cada tipo numérico soporta las siguientes conversiones:

`toByte() : Byte`

`toShort() : Short`

`toInt() : Int`

`toLong() : Long`

`toFloat() : Float`

`toDouble() : Double`

`toChar() : Char`



Tipos básicos

Booleano

- Expresa un valor que puede ser verdadero (**true**) o falso (**false**)

```
val myTrueBoolean = true
val myFalseBoolean = false
```

Char

```
val character = 'x'
```

String

- `val myString = "Esto es un string de una sola línea."`
- `val myLongString = """Esto es un string, de más de una línea."""`

Interpolación

- `val author = "Sandra" -> val myString = "El autor de este texto es $author"`
- `val edad= 40 -> val otroMensaje = "You are ${if (edad > 60) "old" else "young"}" // You are young`



Matrices (arrays)

Dos maneras de crearlos

- **arrayOf()**

`val myArray1= arrayOf(0,1,2,"Hola",4,false)` - array mixto

`val myArray3 = arrayOf<Int>(4, 5, 7, 3)` - array de tipo Int

`val myArray4 = intArrayOf(4, 5, 7, 3)` - array de tipo int

- **constructor Array()** - longitud y función lambda

`val numbersArray = Array(5, { i -> i * 2 })` - array de tipo int
0, 2, 4, 6, y 8



Propiedades

size - El equivalente a **length**

```
var weekDays = arrayOf("Lunes", "Martes", "Miércoles", "Jueves",  
"Viernes", "Sábado", "Domingo");  
println(weekDays.size) //7
```

Métodos

set() - `weekDays.set(0, "Lunes")`

get() - `weekDays.get(0) //Lunes`

`weekDays[0] = "Lunes"`

`weekDays[0] //Lunes`





Otros métodos

indices()

lastIndex()

count()

distinct()

drop()

filter()

find()

first()

forEach()

indexOf()

isEmpty()

map()

none()

random()

reduce()

reverse()

slice()

sort()

subtract()

sum()

take()





Recorrer arrays

for in

```
for (posicion in weekdays.indices) {  
    println(weekdays.get(posicion))  
}
```

```
for ((posicion, valor) in weekdays.withIndex()) {  
    println("La posición $posicion contiene el valor  
$valor")  
}
```

```
for (weekday in weekdays) {  
    println(weekday)  
} //Devuelve los valores
```





Recorrer arrays

```
for in
  for (i in 0..weekDays.size-1)
  {
    println(weekDays[i])
  }
```

forEach

```
weekDays.forEach({ index -> println(index) })
```

