

## 2. Kotlin Básico

Alberto Luengo



# Un vistazo a lo que viene

KOTLIN

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        ...  
        fab.setOnClickListener { view ->  
            Snackbar.make(view, "Hello $name", Snackbar.LENGTH_LONG).show()  
        }  
    }  
}
```

Nullable and NonNull  
types help reduce  
NullPointerExceptions

Use lambdas for concise  
event handling code

Use template expressions  
in strings to avoid concatenation

Semicolons are optional

# Declarar variables en Kotlin

En kotlin tenemos **2** tipos de variables, val y var, y se declaran de la siguiente manera.

```
var mensaje = "Este es un texto"  
val contador = 195
```

Diferencias entre var y val

**val** (immutable): el valor de la variable declarada mediante la palabra clave **val** no se puede cambiar una vez que se asigna un valor.

**var** (mutable): el valor de la variable declarada mediante la palabra clave **var** se puede cambiar más adelante.

En el ejemplo, la variable “mensaje”, más adelante lo podemos cambiar a cualquier valor, en cambio, “contador” siempre se va a mantener 195.

```
mensaje = "¿que tal?"  
contador = 11
```

Val cannot be reassigned

# Comentarios

Los comentarios en Kotlin/Java son de la siguiente manera

```
// Aqui escribo un mensaje

/**
 * Aqui escribo un mensaje
 */

/*
    Aqui escribo un mensaje
*/
```

# Tipos de datos

Siempre que asignes un valor a una variable, será un dato, la mayoría serán de los siguientes tipos. Estos son los más básico, pero hay muchos más ([aquí están](#)).

| Tipo de dato | Descripción           | Ejemplo de Valor  |
|--------------|-----------------------|---|
| String       | Cualquier texto       | "Hola Mundo 2024"   |
| Int          | Numérico 32 Bits      | Cualquier numero entre -2.147.483.648 y 2.147.483.647                                     |
| Boolean      | Verdadero o Falso     | true o false  |
| Float        | Numérico 7 decimales  | 123.5f  |
| Double       | Numérico 15 decimales | 123.5, 123.5e10   |
| Array        | Conjunto de datos     | Es un tipo de dato estructurado que permite almacenar datos del mismo tipo y relacionados |

Se utilizan de la siguiente manera, aunque en ocasiones, el dato que pones es común,, kotlin te lo identifica automáticamente.

```
var miVariable = "Hola mundo"
var miTexto: String = "Hola mundo"
```



# Operadores aritméticos

Te permiten expresar operaciones aritméticas entre dos operandos.

| Operador | Operación      | Expresión        | Función Equivalente     |
|----------|----------------|------------------|-------------------------|
| +        | Suma           | <code>a+b</code> | <code>a.plus(b)</code>  |
| -        | Resta          | <code>a-b</code> | <code>a.minus(b)</code> |
| *        | Multiplicación | <code>a*b</code> | <code>a.times(b)</code> |
| /        | División       | <code>a/b</code> | <code>a.div(b)</code>   |
| %        | Residuo        | <code>a%b</code> | <code>a.rem(b)</code>   |

# Operadores de asignación compuesta

Estos operadores son la combinación entre el operador de asignación y los operadores aritméticos, con el fin de usar como operando la variable de resultado.

| Operador        | Expresión simplificada | Expresión Completa | Función Equivalente           |
|-----------------|------------------------|--------------------|-------------------------------|
| <code>+=</code> | <code>a+=b</code>      | <code>a=a+b</code> | <code>a.plusAssign(b)</code>  |
| <code>-=</code> | <code>a-=b</code>      | <code>a=a-b</code> | <code>a.minusAssign(b)</code> |
| <code>*=</code> | <code>a*=b</code>      | <code>a=a*b</code> | <code>a.timesAssign(b)</code> |
| <code>/=</code> | <code>a/=b</code>      | <code>a=a/b</code> | <code>a.divAssign(b)</code>   |
| <code>%=</code> | <code>a%=b</code>      | <code>a=a%b</code> | <code>a.remAssign(b)</code>   |

# Operadores de incremento y decremento

El operador de incremento, representado por dos signos de suma (++), incrementa en la unidad al operando.

```
var num = 0  
num++
```

Análogamente, el operador de decremento, doble signo menos (--), disminuye en la unidad al operando.

```
num--
```



# Operadores relacionales

Los operadores relacionales te permiten verificar enunciados de igualdad y desigualdad entre dos valores.

El tipo de dato resultante de la expresión es Boolean, indicando la veracidad del enunciado expresado.

| Operador           | Enunciado                              | Expresión            | Función Equivalente                          |
|--------------------|--|----------------------|--|
| <code>==</code>    | <i>a</i> es igual a <i>b</i>           | <code>a==b</code>    | <code>a?.equals(b) ?: (b === null)</code>    |
| <code>!=</code>    | <i>a</i> es diferente de <i>b</i>      | <code>a!=b</code>    | <code>!(a?.equals(b) ?: (b === null))</code> |
| <code>&lt;</code>  | <i>a</i> es menor que <i>b</i>         | <code>a&lt;b</code>  | <code>a.compareTo(b)&lt;0</code>             |
| <code>&gt;</code>  | <i>a</i> es mayor que <i>b</i>         | <code>a&gt;b</code>  | <code>a.compareTo(b)&gt;0</code>             |
| <code>&lt;=</code> | <i>a</i> es menor ó igual que <i>b</i> | <code>a&lt;=b</code> | <code>a.compareTo(b) &lt;=0</code>           |
| <code>&gt;=</code> | <i>a</i> es mayor o igual que <i>b</i> | <code>a&gt;=b</code> | <code>a.compareTo(b)&gt;= 0</code>           |

# Operadores lógicos

Los operadores lógicos te permiten crear expresiones de lógica proposicional como lo son conjunción, disyunción y negación.

| Operador                | Descripción   | Expresión                 |
|-------------------------|---|---------------------------|
| <code>&amp;&amp;</code> | Conjunción (and): el resultado es <code>true</code> si <i>a</i> y <i>b</i> son <code>true</code>  | <code>a&amp;&amp;b</code> |
| <code>  </code>         | Disyunción (or): el resultado es <code>true</code> si <i>a</i> o <i>b</i> son <code>true</code>   | <code>a  b</code>         |
| <code>!</code>          | Negación (not): el resultado es <code>false</code> si <i>a</i> es <code>true</code> , o viceversa | <code>!a</code>           |

# Ejercicios

# Ejercicios

1. Haz las operaciones **suma**, **resta**, **multiplicacion** y **division** con los numeros 23 y 7 guardando cada resultado en una variable diferente.  
Imprime el resultado por consola
2. Compara si de los numeros anteriores el primero es mayor que el segundo.  
Imprime el resultado por consola
3. Compara si de los numeros anteriores el primero es menor que el segundo.  
Imprime el resultado por consola
4. Realiza el **ejercicio 1** pero con los numeros 12.6 y 3.4
5. Comprueba si el resultado de la resta del ejercicio **1** y de la suma del ejercicio **4** son diferentes.  
Imprimir el resultado por consola.



# Gracias.