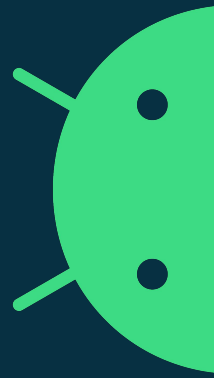
# Hour of Code Kotlin

Iniciación a kotlin



android

# Kotlin

¿Que es?

# ¿Que es kotlin?

- lenguaje tipado y estático
- corre sobre la máquina virtual de Java
- puede ser compilado a código fuente de JavaScript
- desarrollado principalmente por JetBrains


Kotlin

# Usos

- Desarrollo móvil multiplataforma.
- Programación del lado del servidor.
- Ciencia de datos.
- Android.

Kotlin

# Prerequisites

- Tener instalado Intellij IDEA o Kotlin.
- Saber algún lenguaje de programación.
- Orientación a objetos.
- Ganas de aprender.
- Ganas de programar.

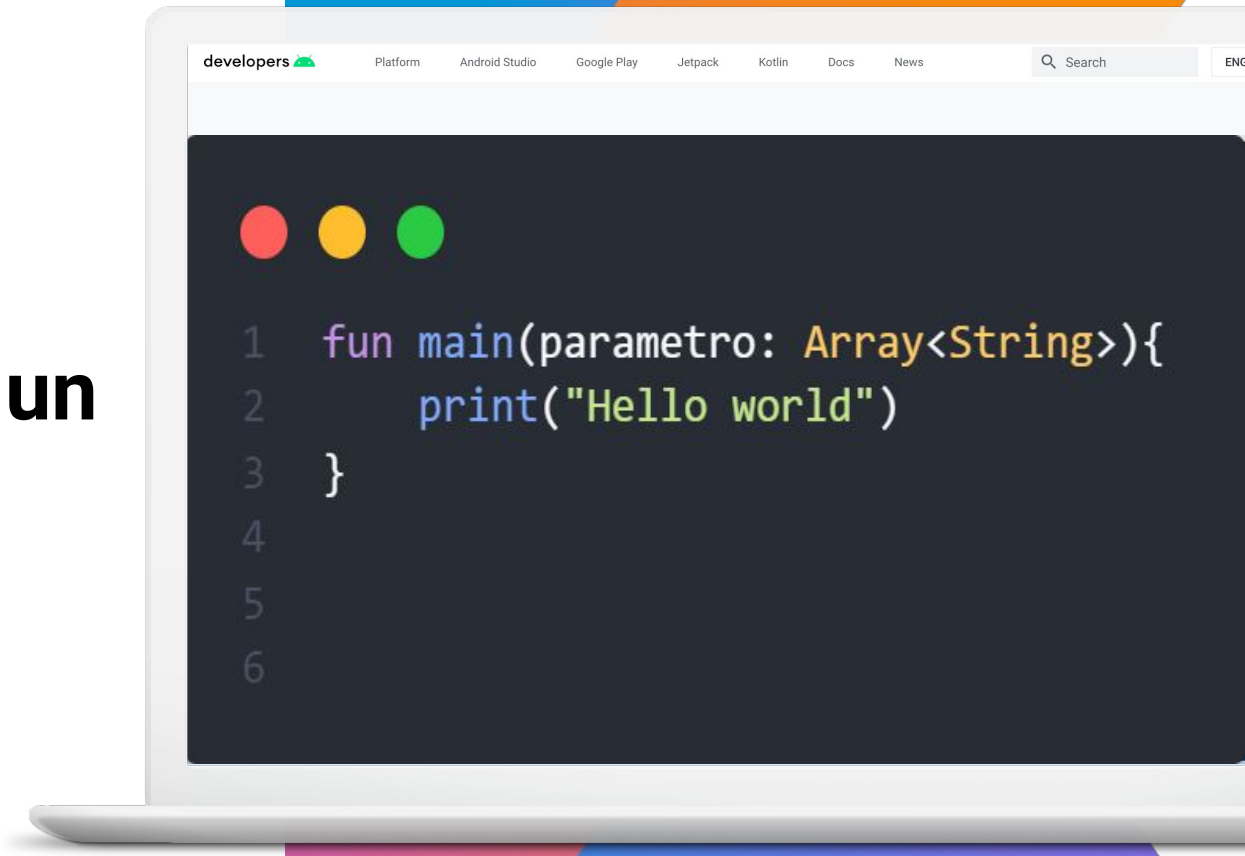
Kotlin

# Empezemos

Kotlin

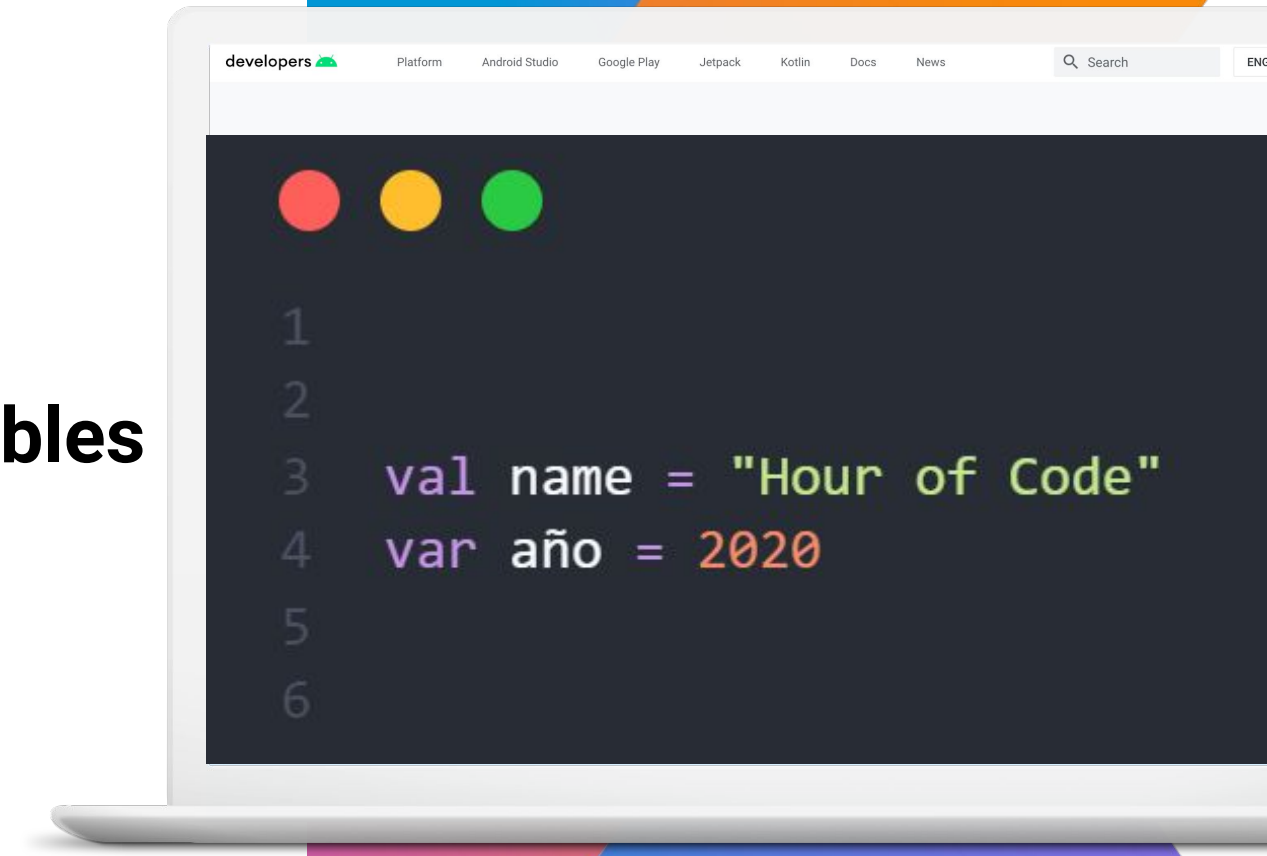
# Who would like to share?

# Calentemos con un “Hola Mundo”



# Tipos de variables

# Dos tipos de variables




Kotlin

# Tipos básicos

- Numbers
- String
- Characters
- Boolean
- Arrays

Kotlin

# Numbers

- Long: 64 Bits
- Int: 32 Bits
- Short: 16 Bits
- Byte: 8 Bits
- Double: 64 Bits
- Float: 32 Bits

Kotlin

```
1
2  val myNumber: Long = 40
3  val MyNumber: Int = 30
4  val sueldo = 1200.55f
5  val titulo = "Sistema de Ventas"
6  var valor1 = 10
7  var valor2 = 100
8  println("La suma de $valor1 + $valor2 es $resultado")
9
10
11
12
13
```

# Conversiones de datos

- toByte(): Byte
- toInt(): Int
- toLong(): Long
- toFloat(): Float
- toChar(): Char
- toDouble(): Double

```kotlin
val b: Byte = 1 // OK, literals are checked statically
val i: Int = b // ERROR
val i: Int = b.toInt() // OK: explicitly widened
print(i)
```

Kotlin

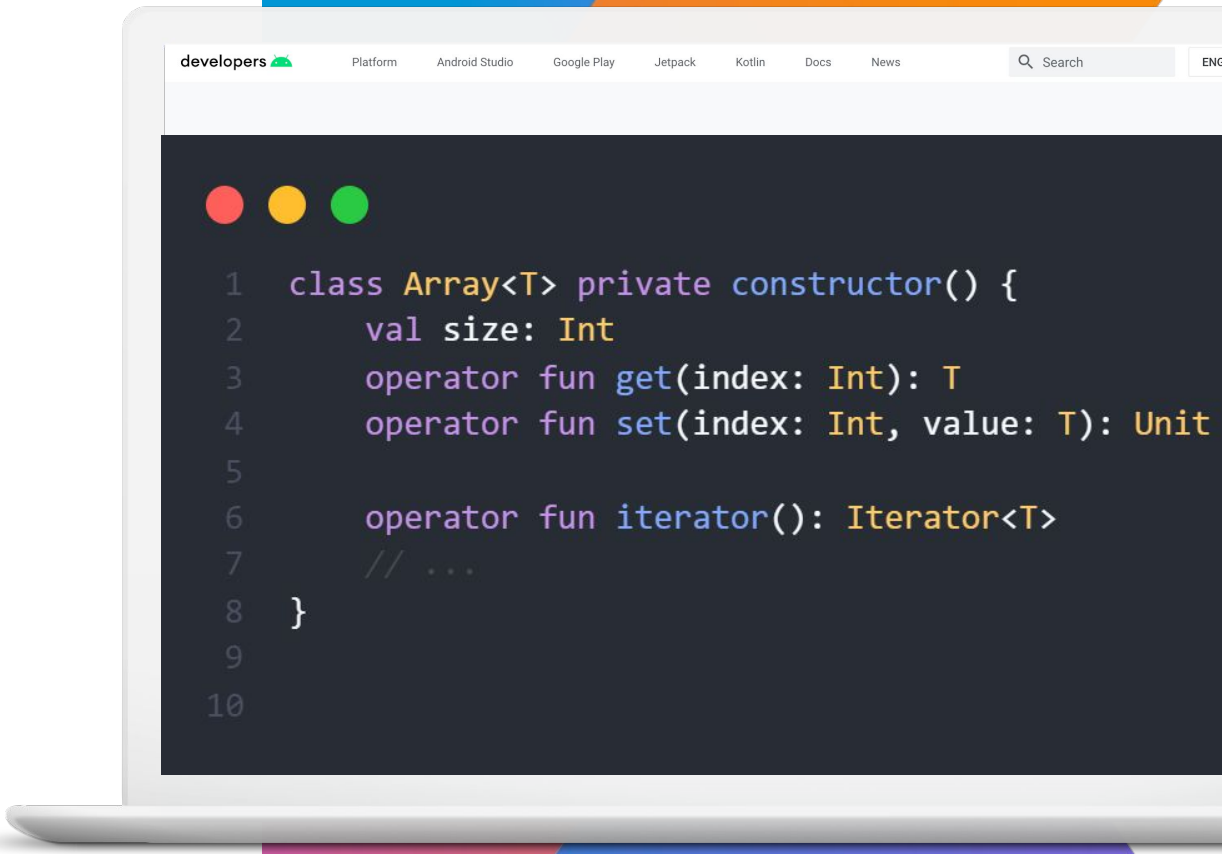
# Bitwise operations

- shl(bits) – signed shift left
- shr(bits) – signed shift right
- ushr(bits) – unsigned shift right
- and(bits) – bitwise and
- or(bits) – bitwise or
- xor(bits) – bitwise xor
- inv() – bitwise inversion

```
1  val b: Byte = 1 // OK, literals are checked statically
2  val i: Int = b // ERROR
3  val i: Int = b.toInt() // OK: explicitly widened
4  print(i)
```

Kotlin

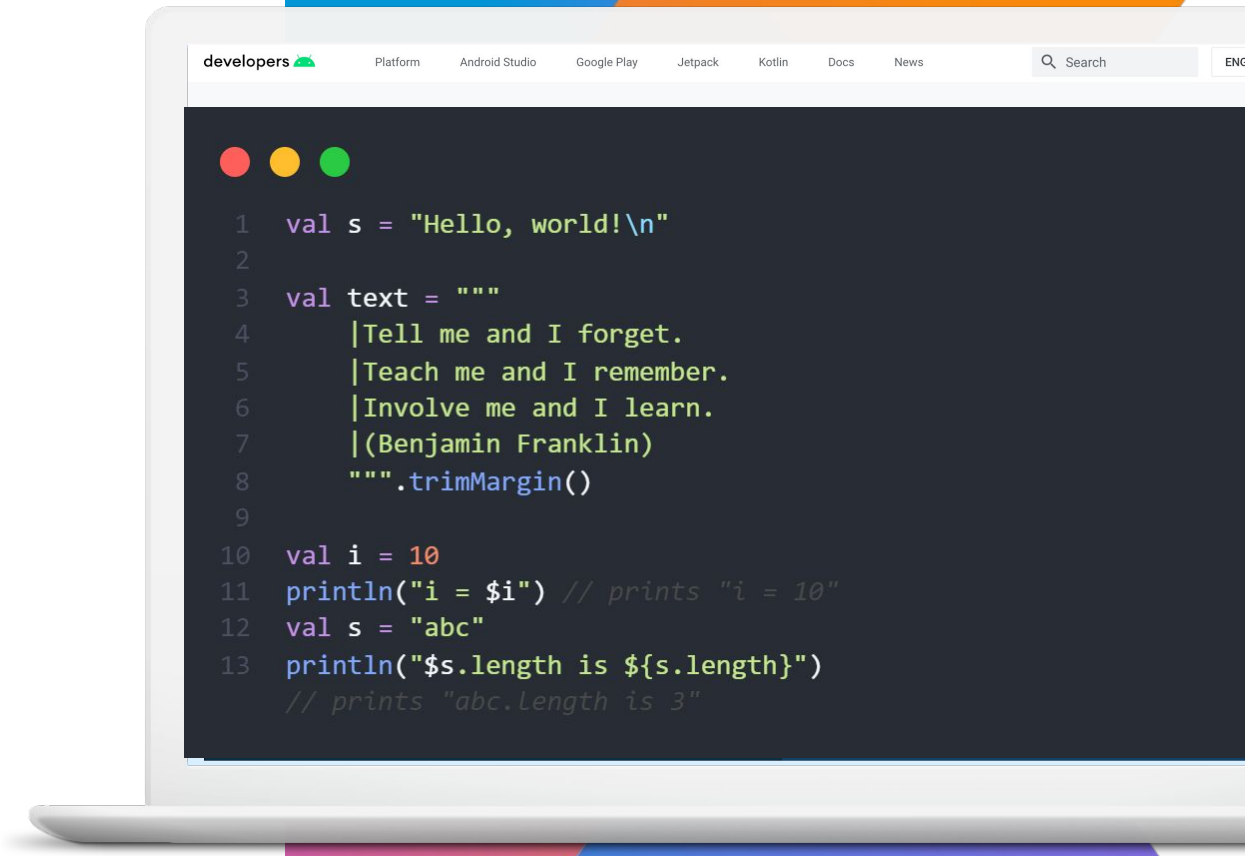
# Arrays




Kotlin

```kotlin
// Crea un Array<String> cno valores ["0", "1", "4", "9", "16"]

val asc = Array(5) { i -> (i * i).toString() }
asc.forEach { println(it) }

val x: IntArray = intArrayOf(1, 2, 3)
x[0] = x[1] + x[2]
// Array de enteros con tamaño 5 con valores [0, 0, 0, 0, 0]

val arr = IntArray(5)

// e.g. Inicializar los valores de un array con una constante


// Array de tamaño 5, con valores [42, 42, 42, 42, 42]
val arr = IntArray(5) { 42 }

// e.g. inicializar los valores con una función lambda
// Array de enteros con tamaño 5 y valores
// [0, 1, 2, 3, 4]
var arr = IntArray(5) { it * 1 }
```

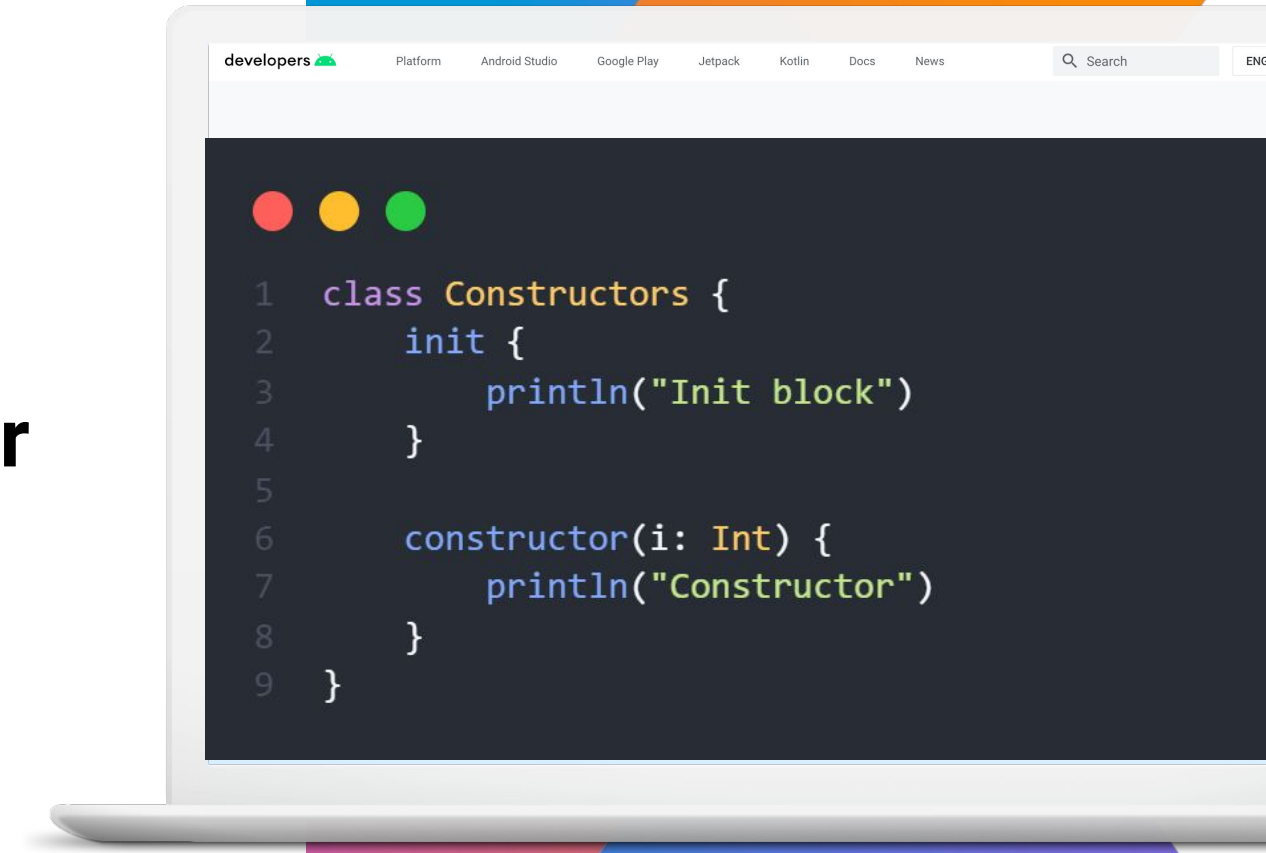
# String




Kotlin

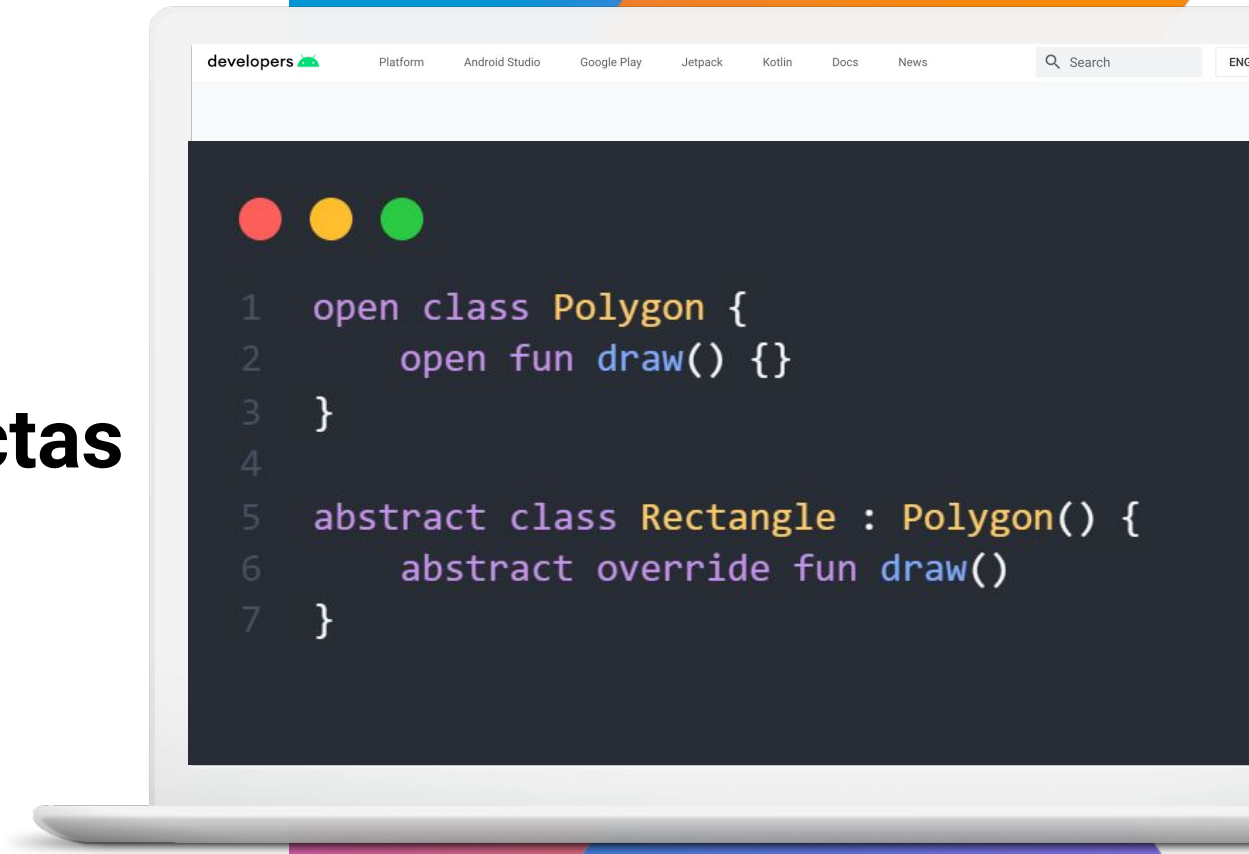
# Clases y objetos

# Constructor




Kotlin

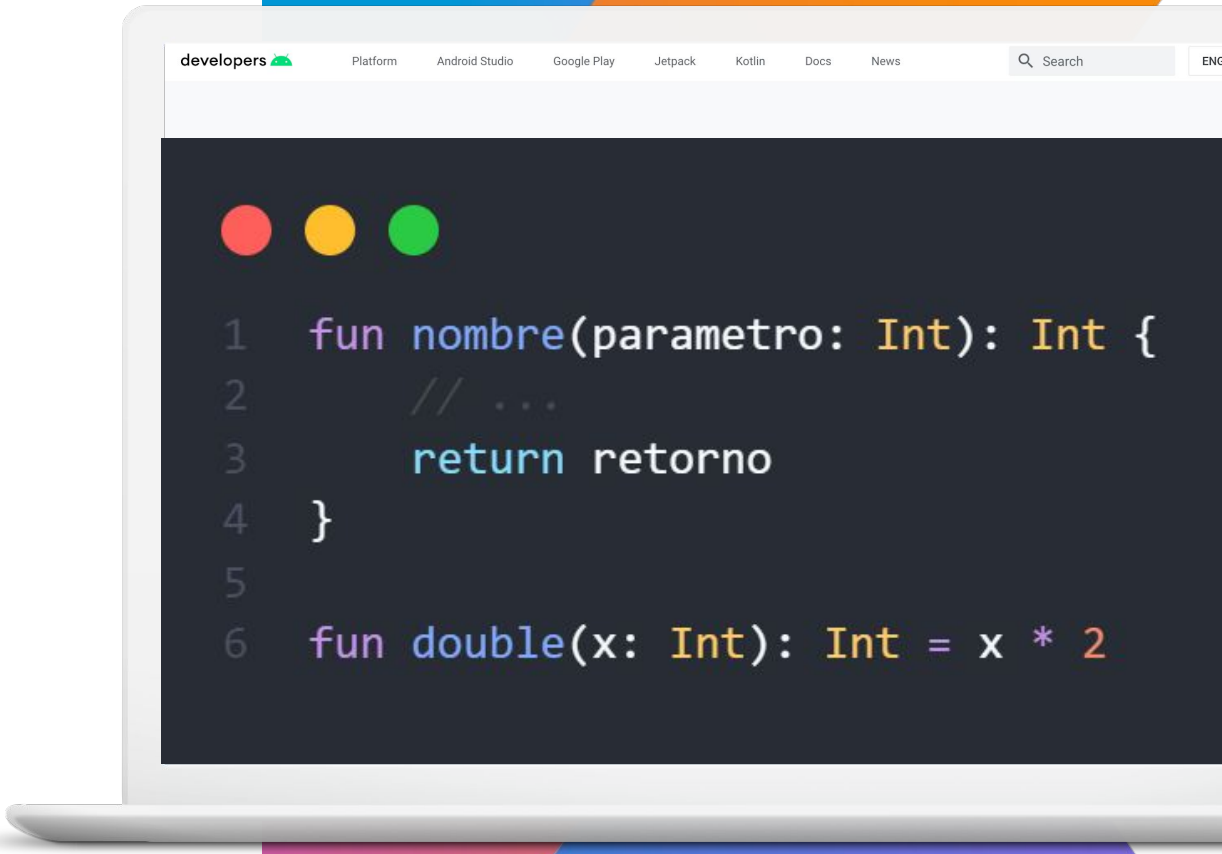
# Clases abstractas




Kotlin

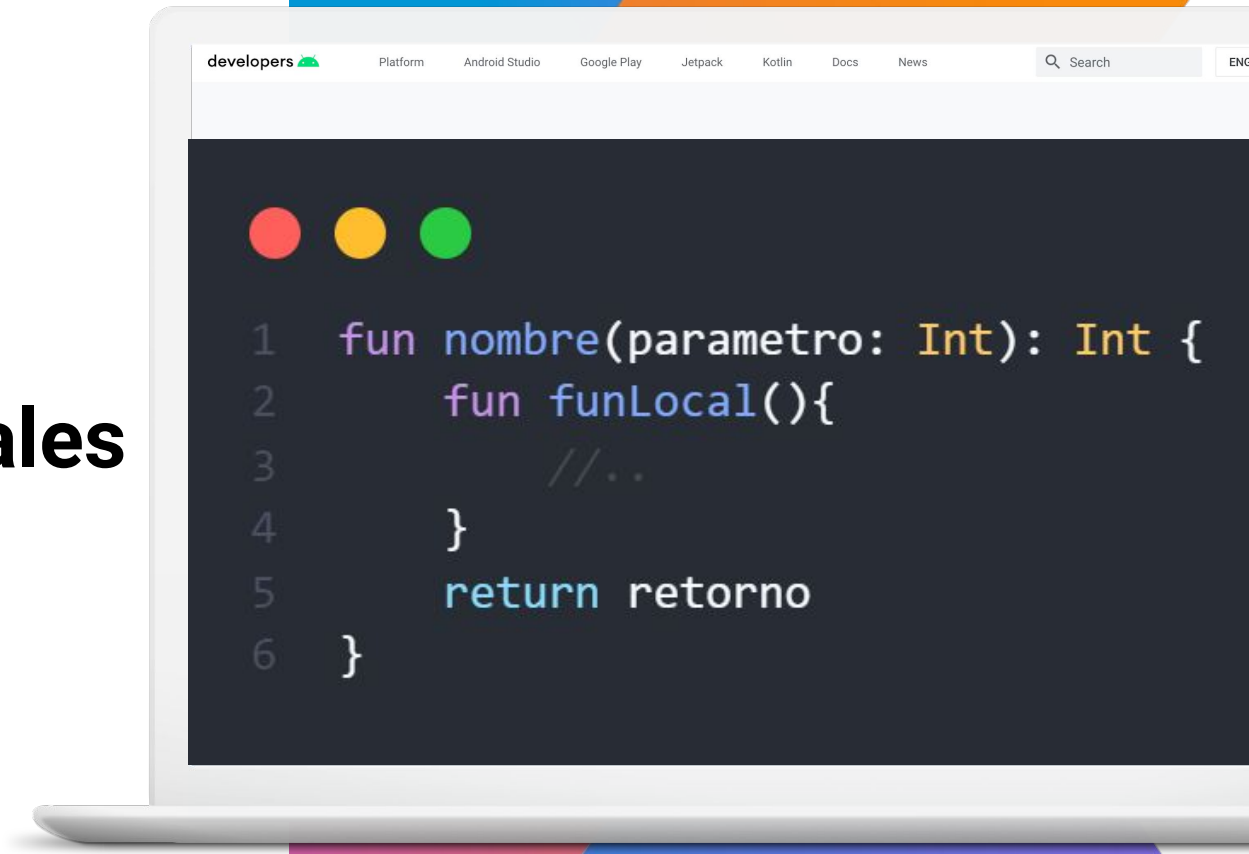
# Función

# Estructura




Kotlin

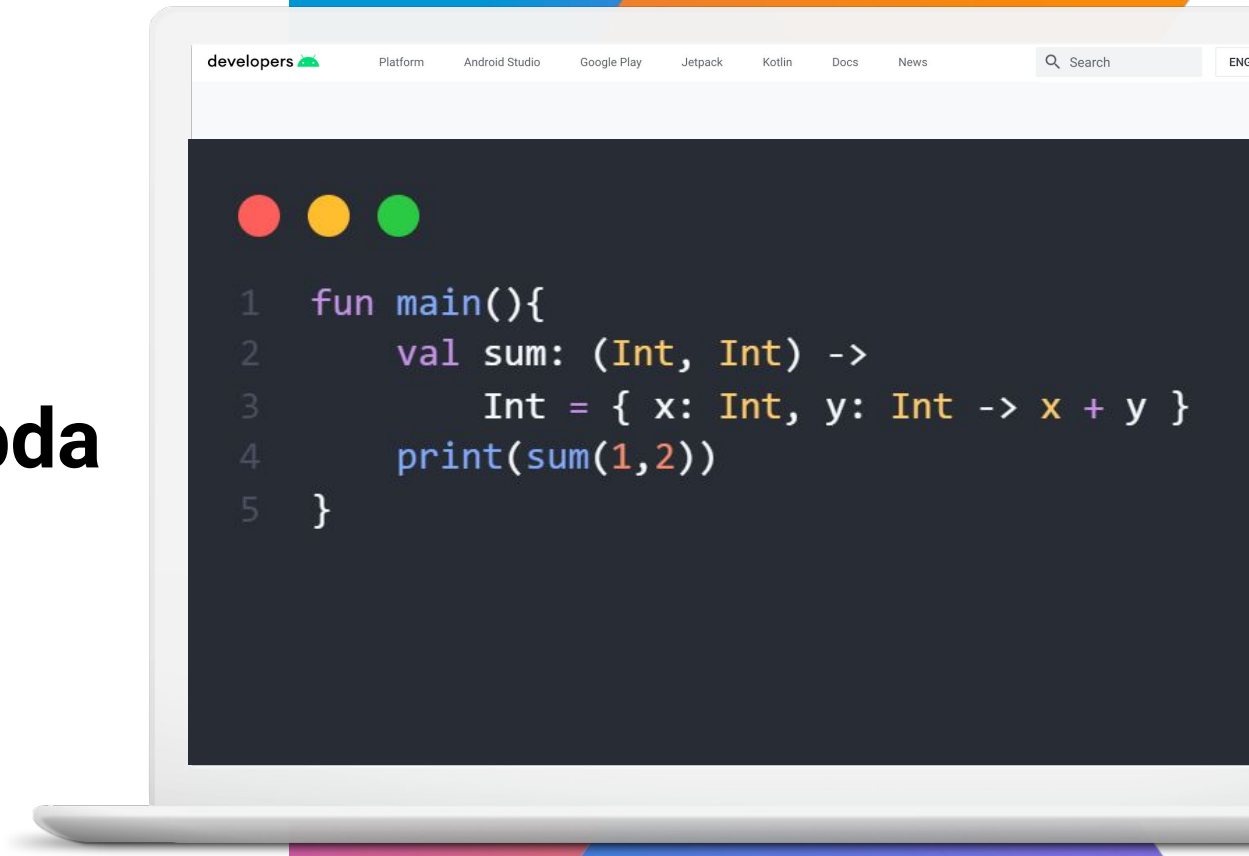
# Funciones locales




Kotlin

# Funciones Lambda

# Ejemplo Lambda




Kotlin

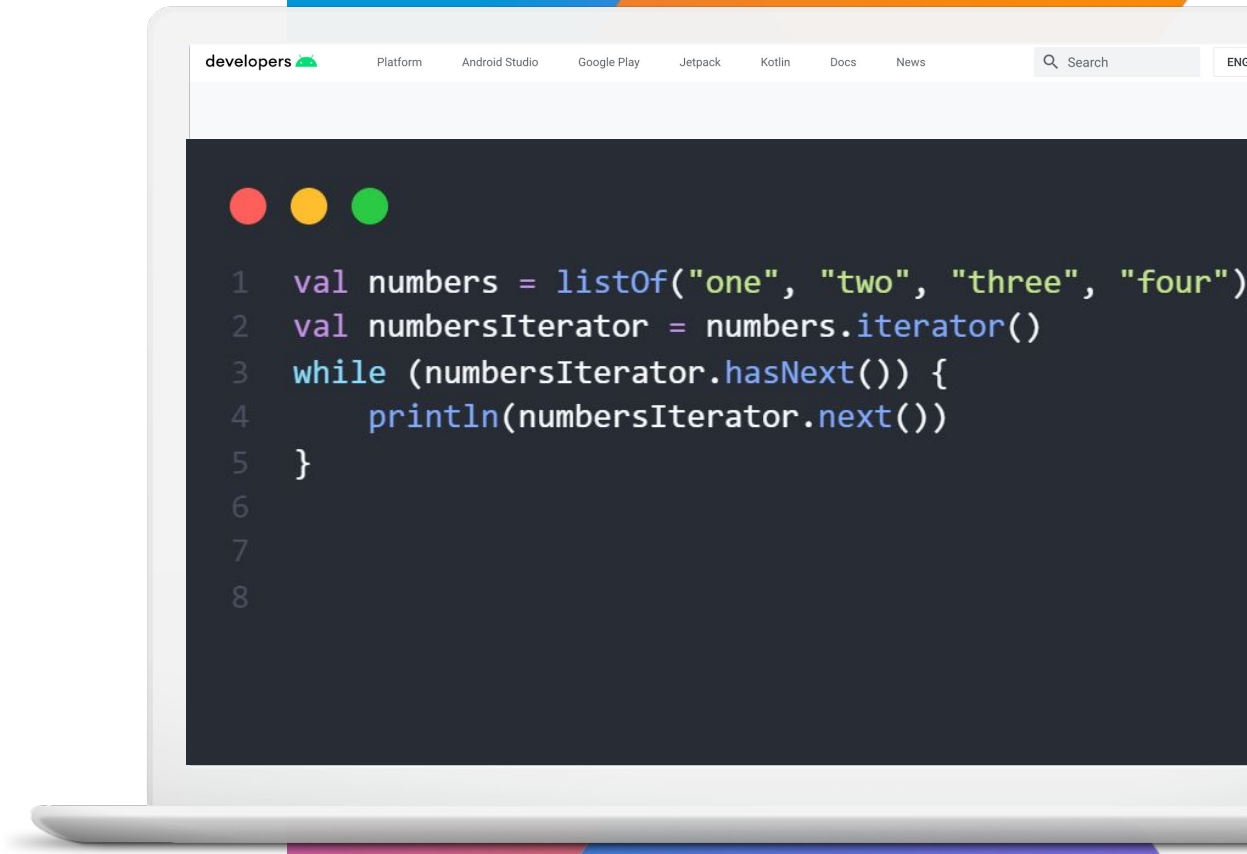
# Funciones inLine

Kotlin

```kotlin
fun retornarSuperficie(lado: Int) = lado * lado

fun main() {
    print("Ingrese el valor del lado del cuafrado:")
    val la = readLine()!!.toInt()
    println("La superficie del cuadrado es ${
retornarSuperficie(la)}")
}

```

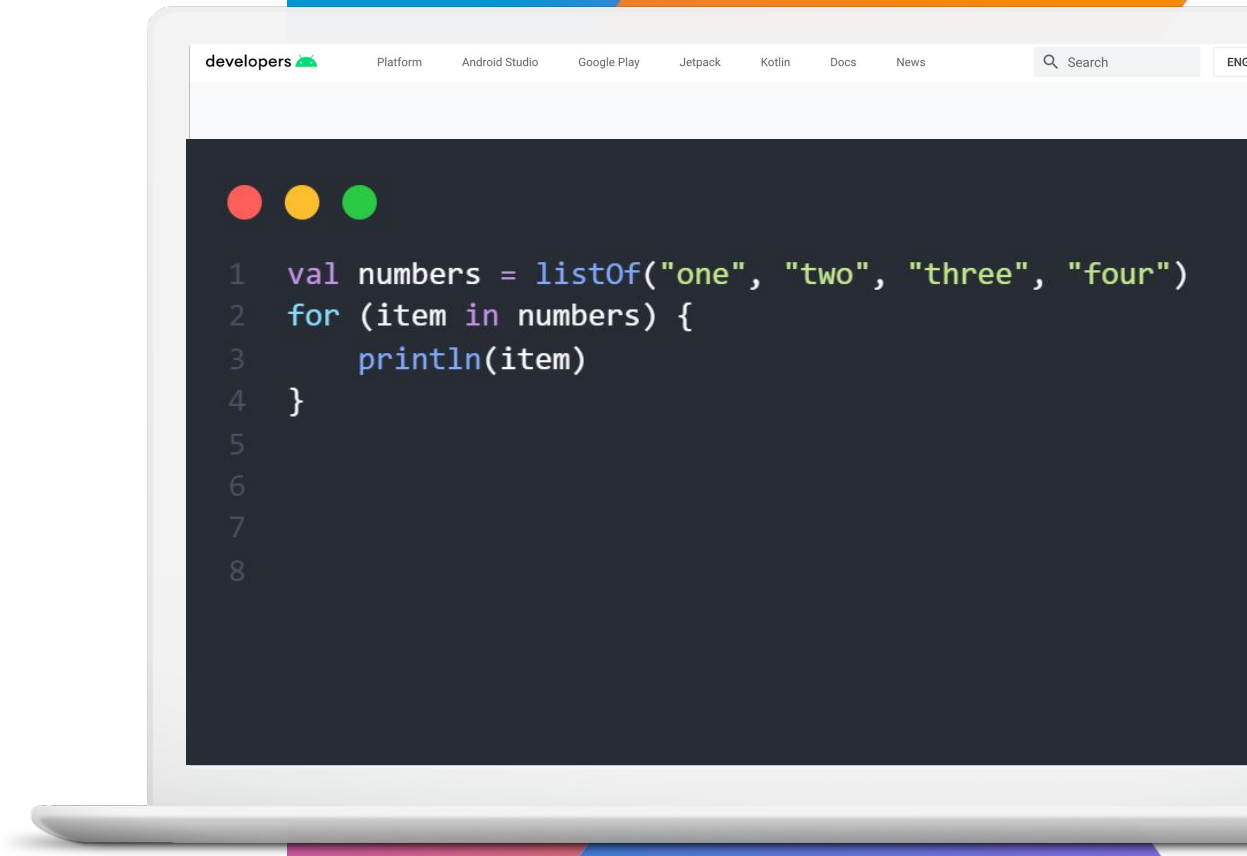
# Iterators

# while



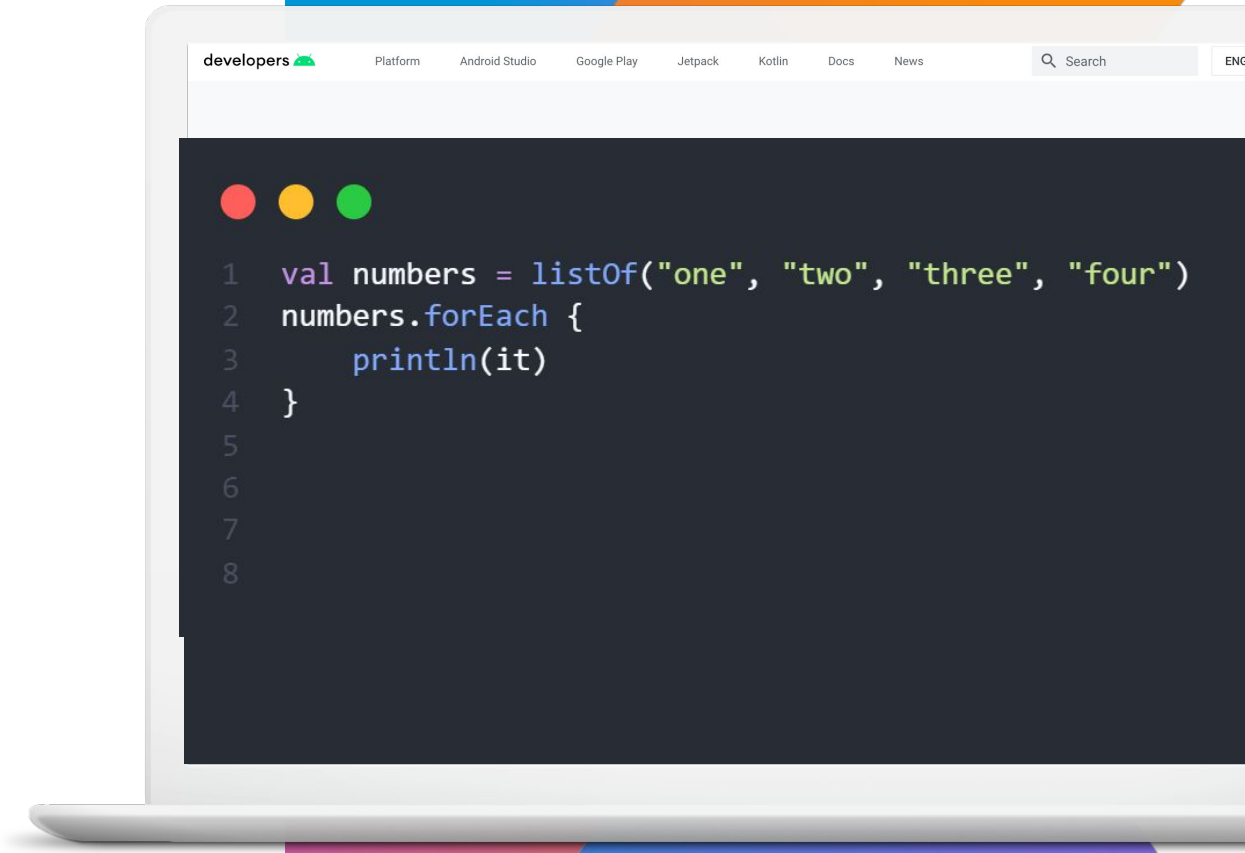

Kotlin

# for


developers
Platform
Android Studio
Google Play
Jetpack
Kotlin
Docs
News
Search
ENG
1 val numbers = listOf("one", "two", "three", "four")
2 for (item in numbers) {
3 println(item)
4 }
5
6
7
8

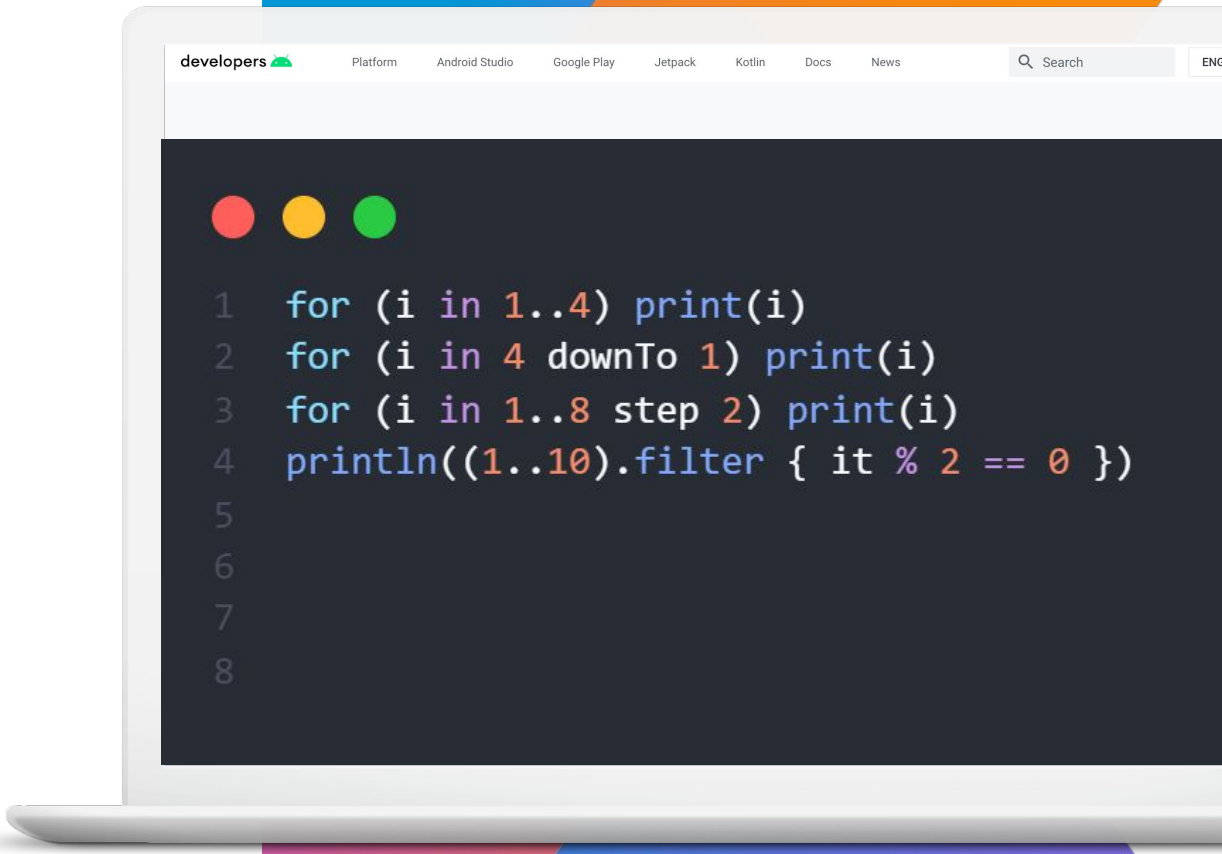
Kotlin

# foreach



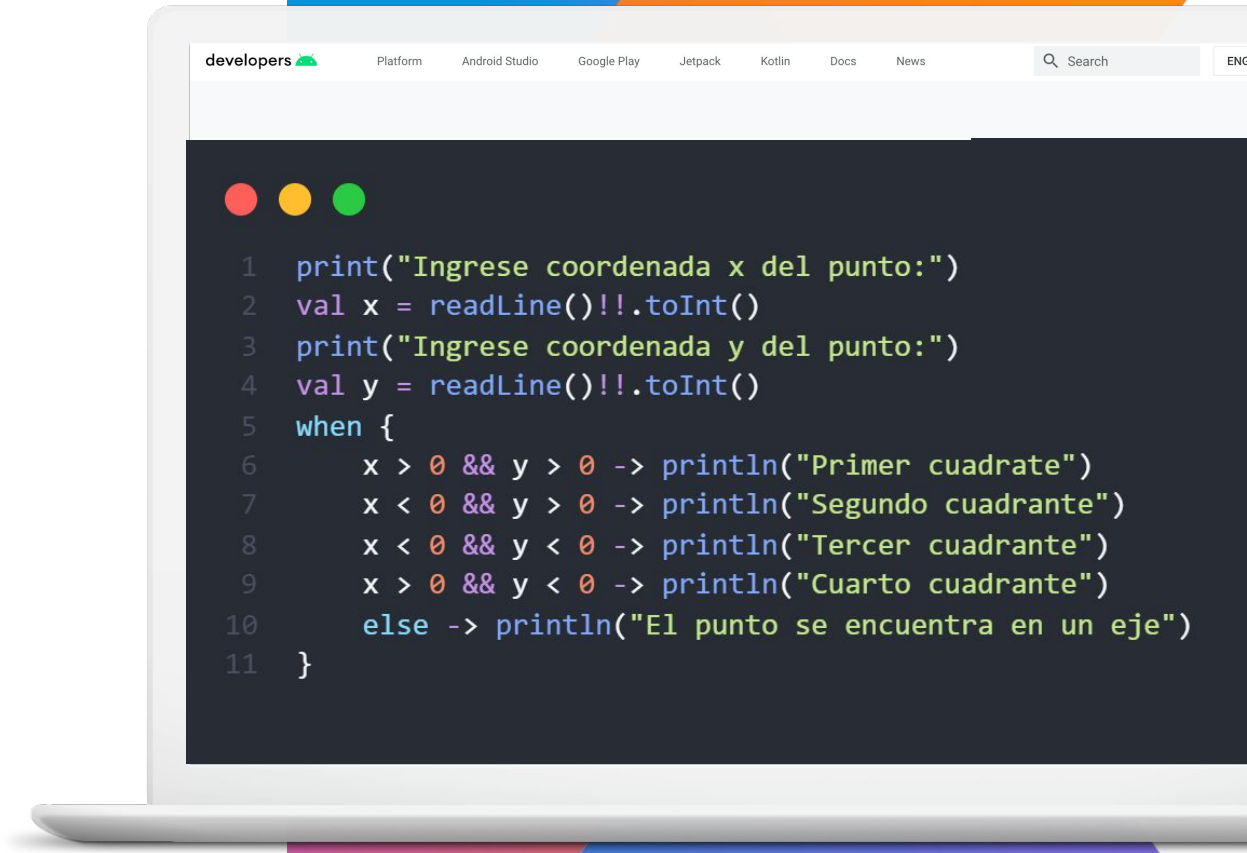

Kotlin

# Range




Kotlin

# When

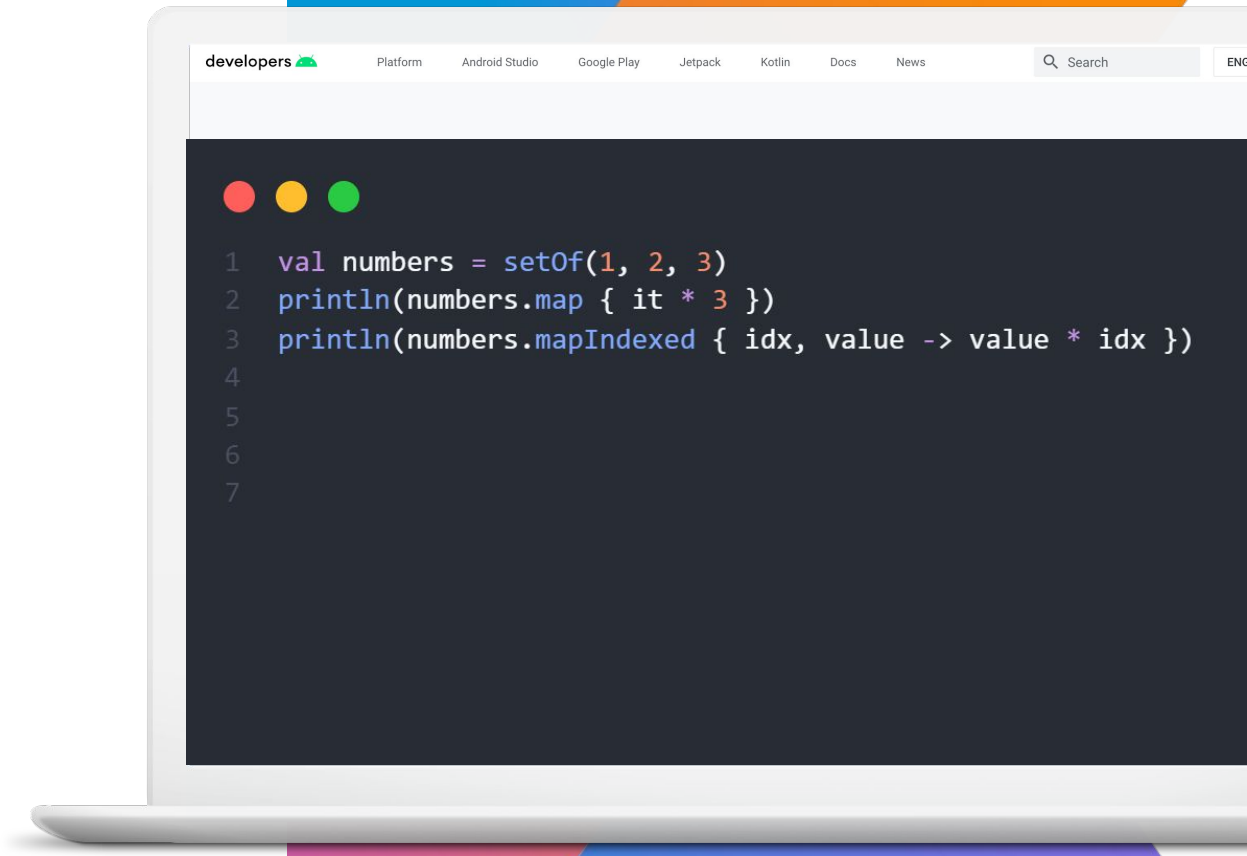

developers
Platform
Android Studio
Google Play
Jetpack
Kotlin
Docs
News
Search
ENG
print("Ingrese coordenada x del punto:")
val x = readLine()!!.toInt()
print("Ingrese coordenada y del punto:")
val y = readLine()!!.toInt()
when {
x > 0 && y > 0 -> println("Primer cuadrate")
x < 0 && y > 0 -> println("Segundo cuadrante")
x < 0 && y < 0 -> println("Tercer cuadrante")
x > 0 && y < 0 -> println("Cuarto cuadrante")
else -> println("El punto se encuentra en un eje")
}

Kotlin

```
print("Ingrese un valor entero entre 1 y 5:")
    val valor = readLine()!!.toInt()
    when (valor) {
        1 -> print("uno")
        2 -> print("dos")
        3 -> print("tres")
        4 -> print("cuatro")
        5 -> print("cinco")
        else -> print("valor fuera de rango")
    }
```

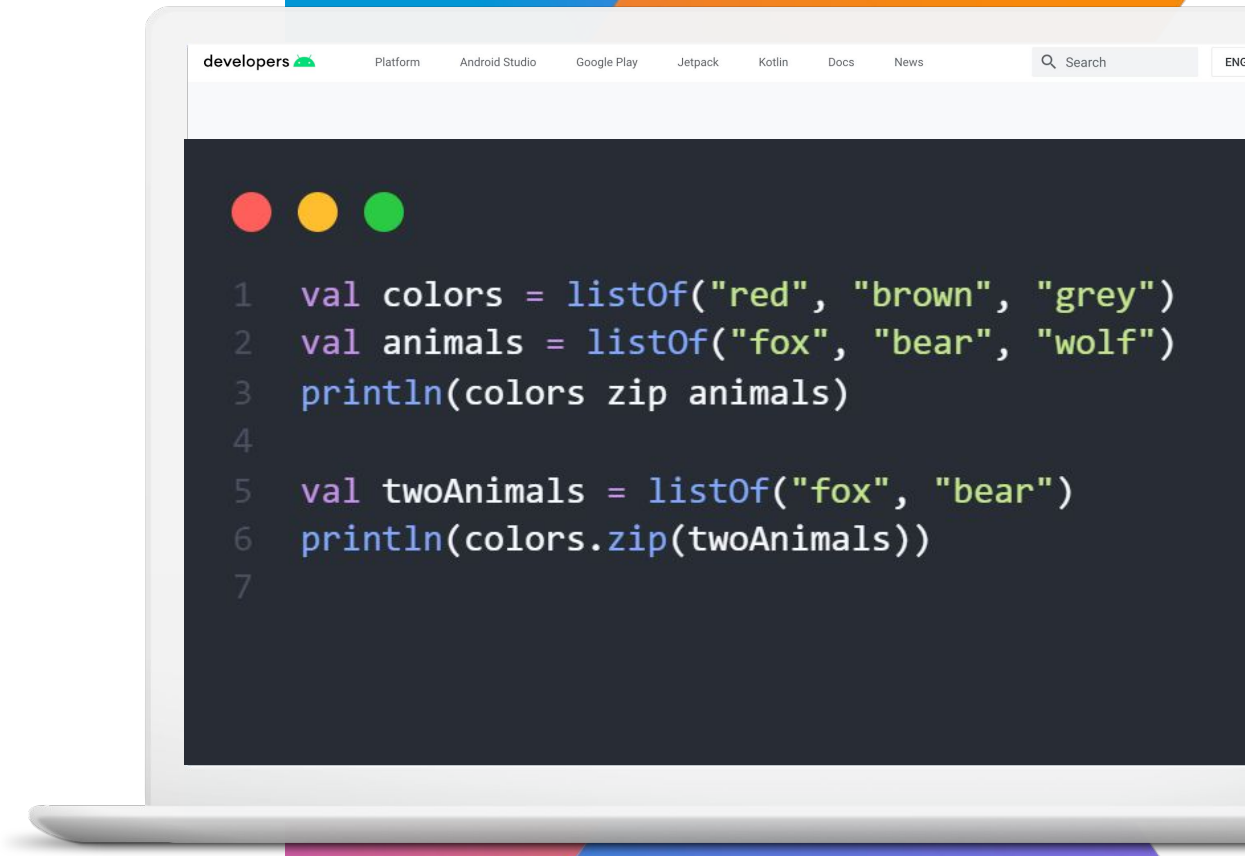
# Collection Transformation

# Mapping



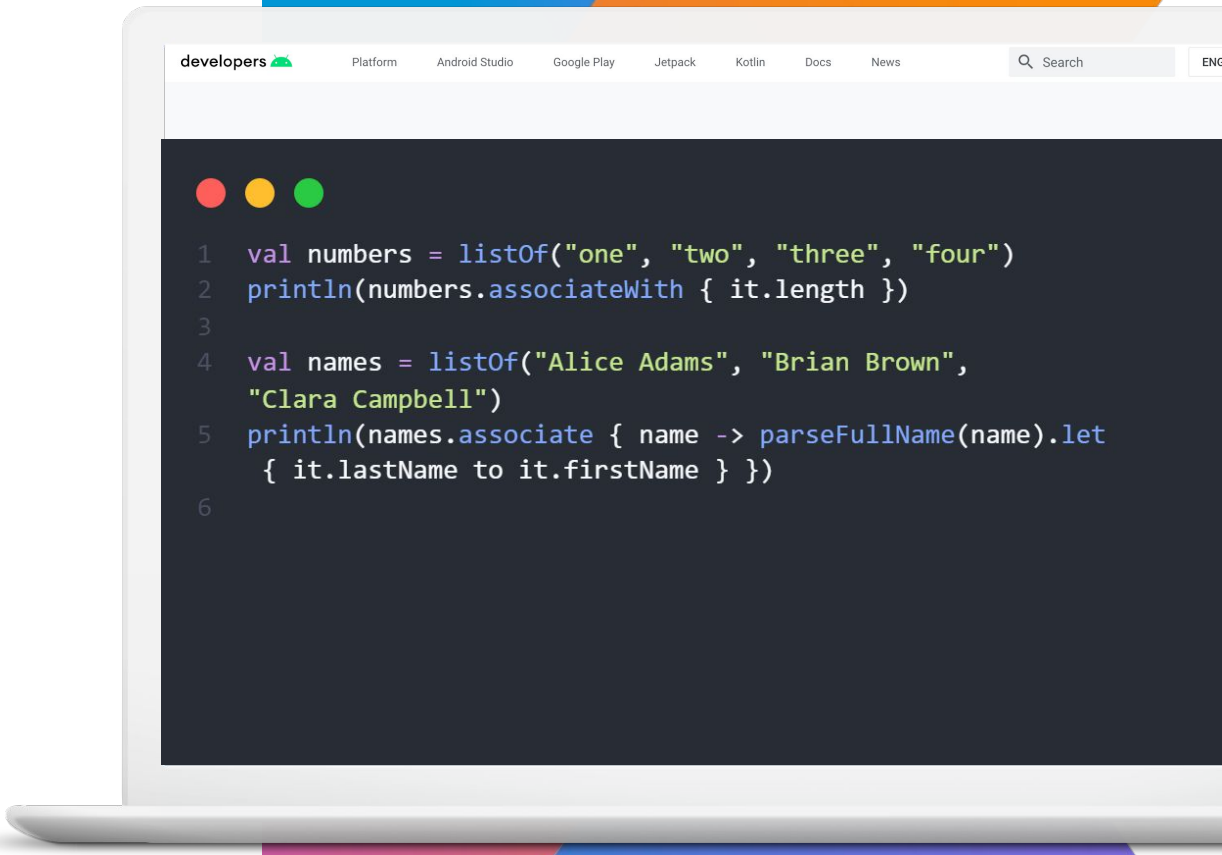

Kotlin

# Zipping



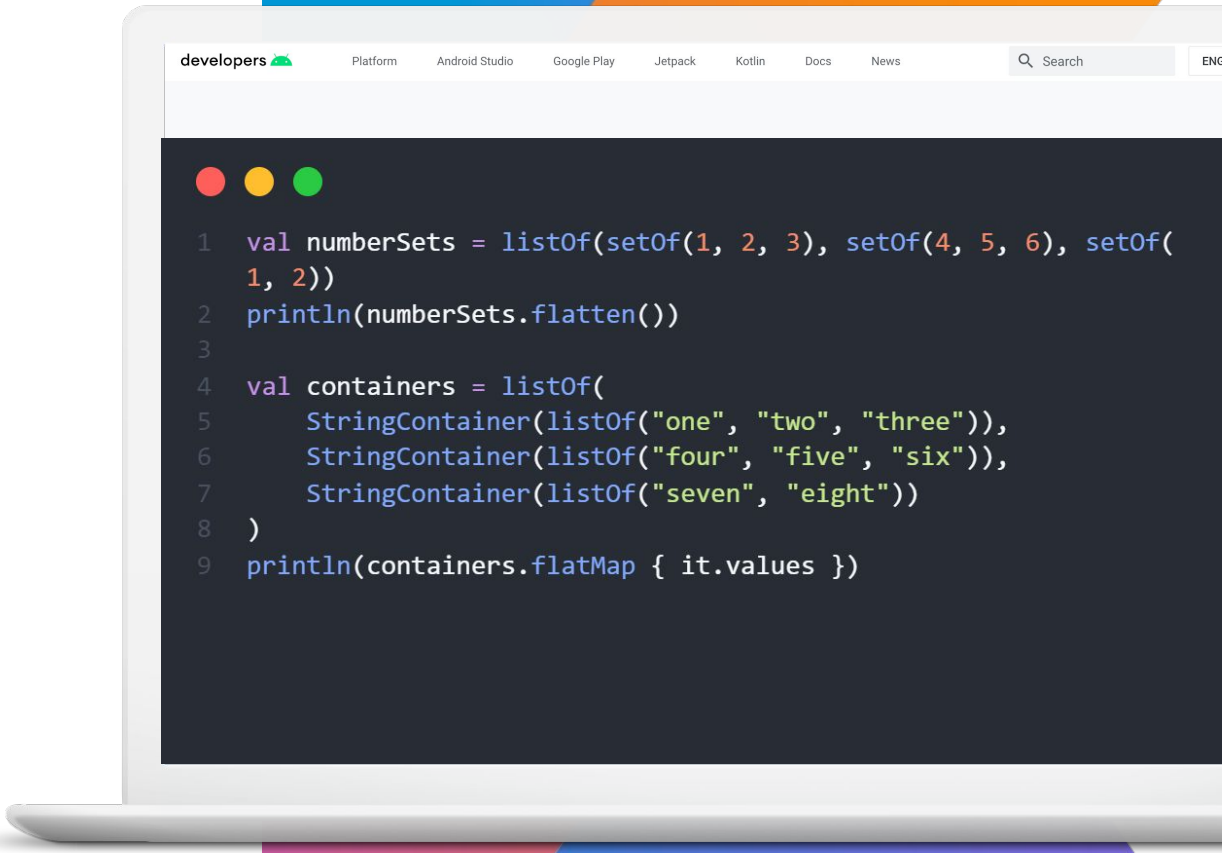

Kotlin

# Association



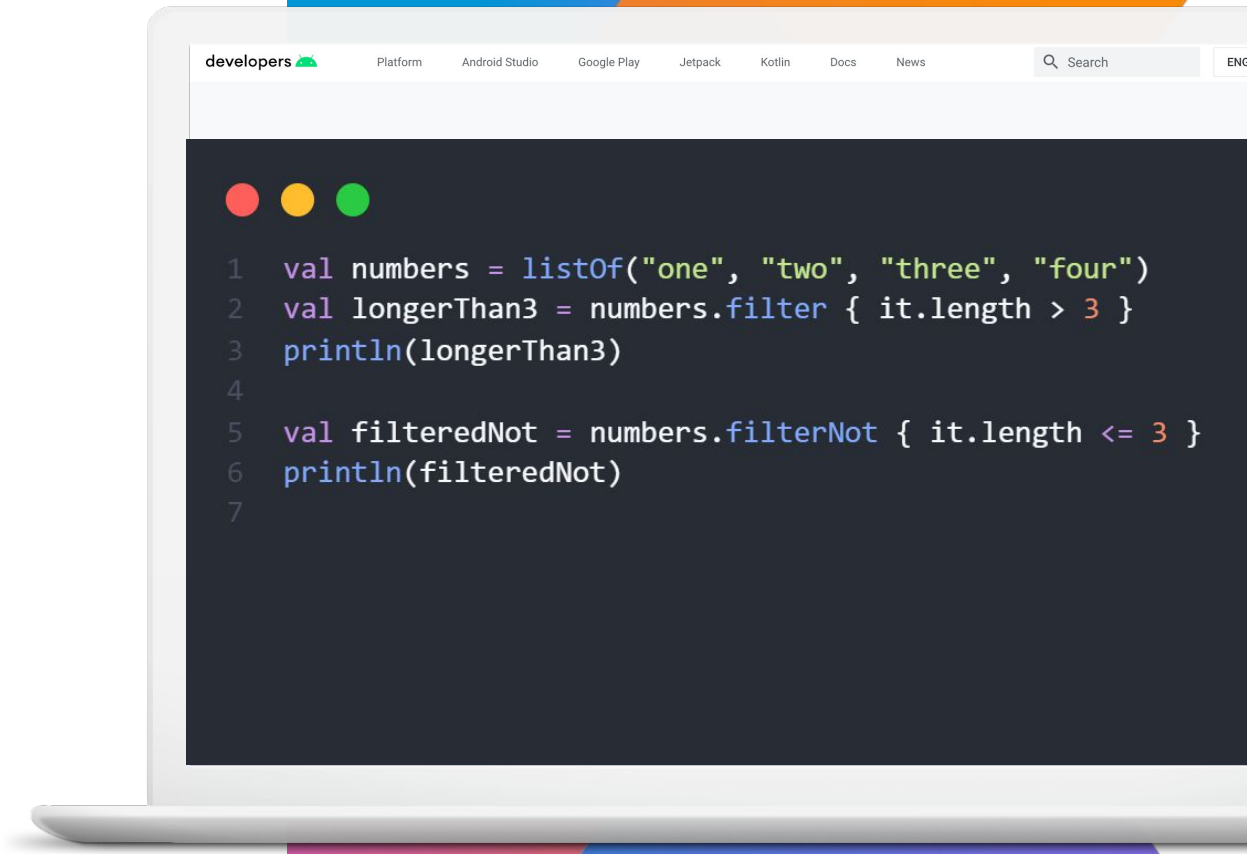

Kotlin

# Flattening




Kotlin

# Filtering




Kotlin

# Muchas gracias por asistir

Si queréis contactar con nosotros:

Angel Moreno
github.com/angelmorenocalvo
@__angelmoreno

Gonzalo Calvo
github.com/GonzaCS
@Gonza_cs175

Kotlin