

Introducció a Kotlin

Fet per TD 7937



Història

- JetBrains comença a 2011 a crear un llenguatge compatible amb Java
- Release Kotlin 1.0 en 2016
- Nom provinent de l'illa Kotlin (Rússia)



Aplicacions típiques

- Desenvolupament d'aplicacions en Android
- Desenvolupament web Back-end , Front-end (K-JS) i Full-Stack
- Data Science

Què té Kotlin d'especial?

- Llenguatge altament expressiu i concís
- Completa interoperabilitat amb JAVA
 - (Compila a bytecode)
- Pot compilar a JavaScript
- Permet asincronia (SUSPEND)
- SafeNulls
- MultiParadigma (Procedural, funcional, OOP)
- Permet funcions d'extensió



Java

```
final int x = // value;
final String xResult;

switch (x) {
    case 0;
    case 11;
        xResult = "0 or 11";
        break;
    case 1;
    case 2;
        //...
    case 10;
        xResult = "from 1 to 10";
        break;
    default;
        if (x < 12 && x > 14) {
            xResult = "not from 12 to 14"
            break;
        }

        if (isOdd(x)) {
            xResult = "is odd";
            break;
        }

        xResult = "otherwise"
}

final int y = //value;
final String yResult;

if(isNegative(y)) {
    yResult = "is zero";
} else if(isOdd(y)) {
    yResult = "is odd";
} else {
    yResult = "otherwise";
}
```

Kotlin

```
val x = //value
val xResult = when (x) {
    0, 11 -> "0 or 11"
    in 1..10 -> "from 1 to 10"
    !in 12..14 -> "not true from 12 to 14"
    else -> if (isOdd(x)) { "is odd" } else { "otherwise" }
}

val y = //value
val yResult = when {
    isNegative(y) -> "is negative"
    isZero(y) -> "is zero"
    isOdd(y) -> "is odd"
    else -> "otherwise"
}
```

```

for (item in collection)
|   print(item)

var n = 100
for (i in 1..n) {
|   println(i)
}

for (i in 6 downTo 0 step 2) {
|   println(i)
}

var x = 500;

while (x > 0) {
|   x--
}

do {
|   println("hola maca")
} while(2 == 1+1)

```

```

fun main(args: Array<String>) {
    listOf(1, 2, 3); mutableListOf("a", "b", "c")

    setOf(1, 2, 3); mutableSetOf("a", "b", "c")

    mapOf(1 to "a", 2 to "b", 3 to "c")
    mutableMapOf("a" to 1, "b" to 2, "c" to 3)

    val aList = listOf(1, 2, 4)

    println(aList.map { elem ->
        elem + 1
    })

    println(aList)

    println(aList.filter { it != 1 })

    fun sum(a: Int, b: Int) = a + b
    println(aList.reduce(::sum))
}

```

```

//Estil funcional , descomposició per casos com a haskell
fun Int.factorial(): Int {
|   return when {
|       this <= 1 -> this
|       else -> this * factorial(this-1)
|   }
}

//Estil imperatiu
fun Int.potencia(n: Int): Int {
|   if (n < 0) throw Exception("red flag amiga")
|   else if(n == 0) return this
|   else{
|       if(n%2 == 0){
|           var r = potencia(this,n/2)
|           return r * r
|       }
|       else{
|           var r = potencia(this, (n-1)/2)
|           return this * r * r
|       }
|   }
}

var x = 2.potencia(30)

```

Book.java

```
1 public class Book {  
2  
3     private String isbn;  
4     private String name;  
5  
6     public Book(String isbn, String name) {  
7         this.isbn = isbn;  
8         this.name = name;  
9     }  
10  
11     public String getIsbn() {  
12         return isbn;  
13     }  
14  
15     public void setIsbn(String isbn) {  
16         this.isbn = isbn;  
17     }  
18  
19     public String getName() {  
20         return name;  
21     }  
22  
23     public void setName(String name) {  
24         this.name = name;  
25     }  
26 }  
27
```

Book.kt ▶ ...

```
1 class Book(var isbn: String, var name: String)  
2
```

Ús de estructures de dades

```
val map = mapOf(1 to "one", 2 to "two")
for ((k, v) in map) {                                // Traverse a map or a list of pairs
    println("$k -> $v")
}

fun obtainKnowledge() = Pair("The Answer", 42)        // Single-expression functions

val (description, answer) = obtainKnowledge()        // Destructure into a pair of two variables
println("$description: $answer")
```


Tipatge

- Molta Inferència de tipus (declaracions amb VAR i VAL)

- Tipatge estàtic i FORT

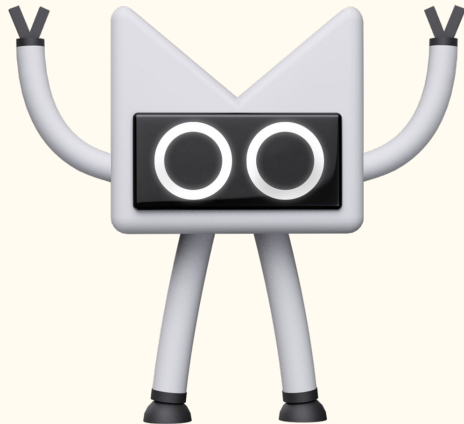
- Safe Nulls i Nullability

 - Safe-Null Assignment (?.)

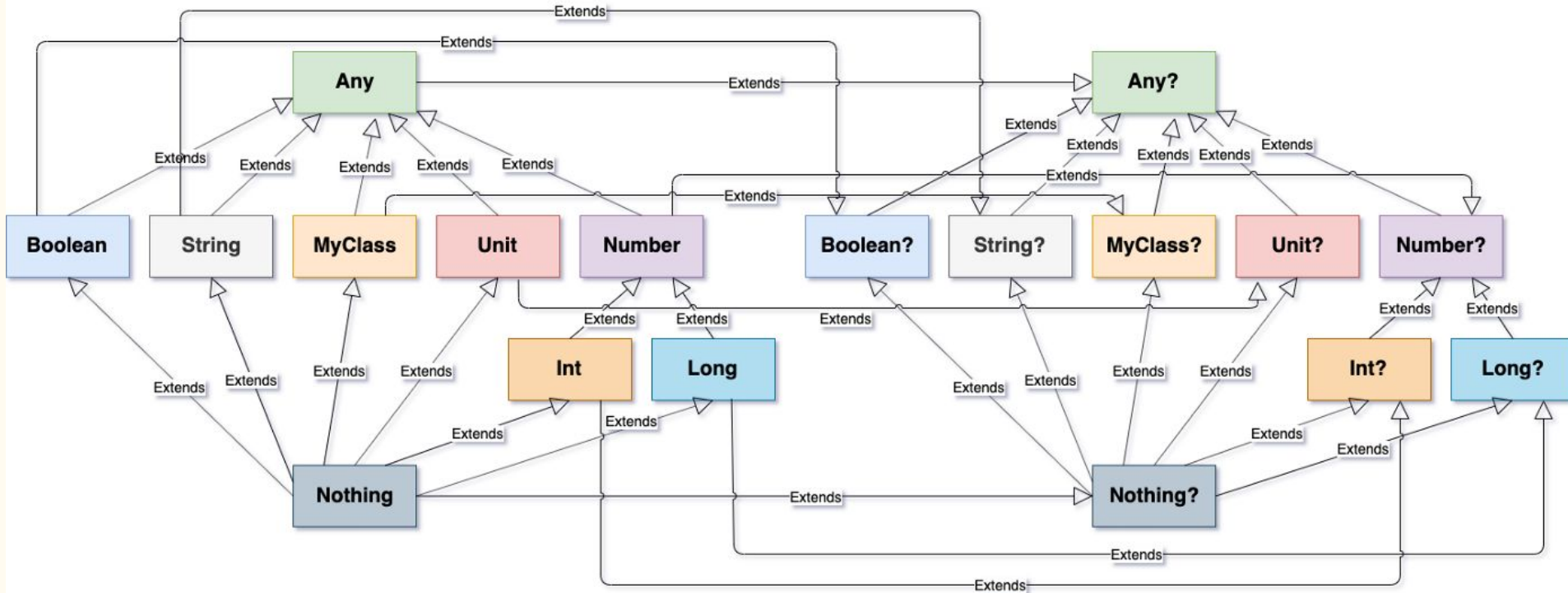
 - Redefine Value if Null (?:)

 - Not Null Assertion (!!)

- String?, Int?... Poden tenir NULL com a valor



Kotlin Type System Hierarchy





```
val x: Int? = null
```

```
// This will give run time exception - NPE
```

```
val y = x.toDouble()
```

```
// Explicit Null Check in Kotlin
```

```
// Secure Access Operation
```

```
val y = x?.toDouble()
```

```
val l = b?.length ?: -1
```

```
val l: Int = if (b != null) b.length else -1
```

```
val l = b!!.length
```

Comparació amb altres llenguatges



Kotlin

Vs.

Java

Comparació amb altres llenguatges



VS



Avantatges	Inconvenients
Codi minimalista	Comunitat petita i Documentació escassa
Fàcil d'aprendre (Si coneixes bé Java)	Compilació lenta
Multiplataforma	Difícil per nous programadors
Interoperable amb Java i retrocompatible	Costa inicialment acostumar-se al sistema de no anul·labilitat i els operadors...
Dona eines per evitar errors i NPE	
Flexibilitat de paradigmes de programació	

Opinió Personal



8/10



Bibliografía i Webgrafía

-Salas, I. (s. f.). *Kotlin*. programandointentandolo.

<https://programandointentandolo.com/kotlin>

-González, R. (5d. C., marzo 21). *Qué es Kotlin?* Crehana.

<https://www.crehana.com/blog/transformacion-digital/que-es-kotlin/>

-colaboradores de Wikipedia. (2022, 2 agosto). *Kotlin (lenguaje de programación)*. Wikipedia, la enciclopedia libre.

[https://es.wikipedia.org/wiki/Kotlin_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Kotlin_(lenguaje_de_programaci%C3%B3n))

-*Kotlin Programming Language*. (s. f.). Kotlin.

<https://kotlinlang.org/>

-Team, D. (2021, 13 agosto). *Kotlin vs Java: Which one's better for android app development in 2021?* Devathon.

<https://devathon.com/blog/kotlin-vs-java-android-app-development/>

-Akhin, M. (s. f.). *Kotlin language specification*.

<https://kotlinlang.org/spec/introduction.html>

Engineerhoon. (2020, 14 junio). *Tutorial #13 | How does Suspend function work internally? | Engineer*. YouTube.

<https://www.youtube.com/watch?v=4YyVxNLFaBQ>

Bibliografía i Webgrafía

MoureDev by Brais Moure. (s. f.). YouTube.

<https://www.youtube.com/@mouredev>

Concurrency and Coroutines. (2022, 22 septiembre).

<https://kotlinlang.org/docs/multiplatform-mobile-concurrency-and-coroutines.html>.

Wikipedia contributors. (2022, 11 noviembre). *Callback (computer programming)*. Wikipedia.

[https://en.wikipedia.org/wiki/Callback_\(computer_programming\)](https://en.wikipedia.org/wiki/Callback_(computer_programming))

Vivo, M. (2021, 13 diciembre). *The suspend modifier — under the hood - Android Developers*. Medium.

<https://medium.com/androiddevelopers/the-suspend-modifier-under-the-hood-b7ce46af624f>