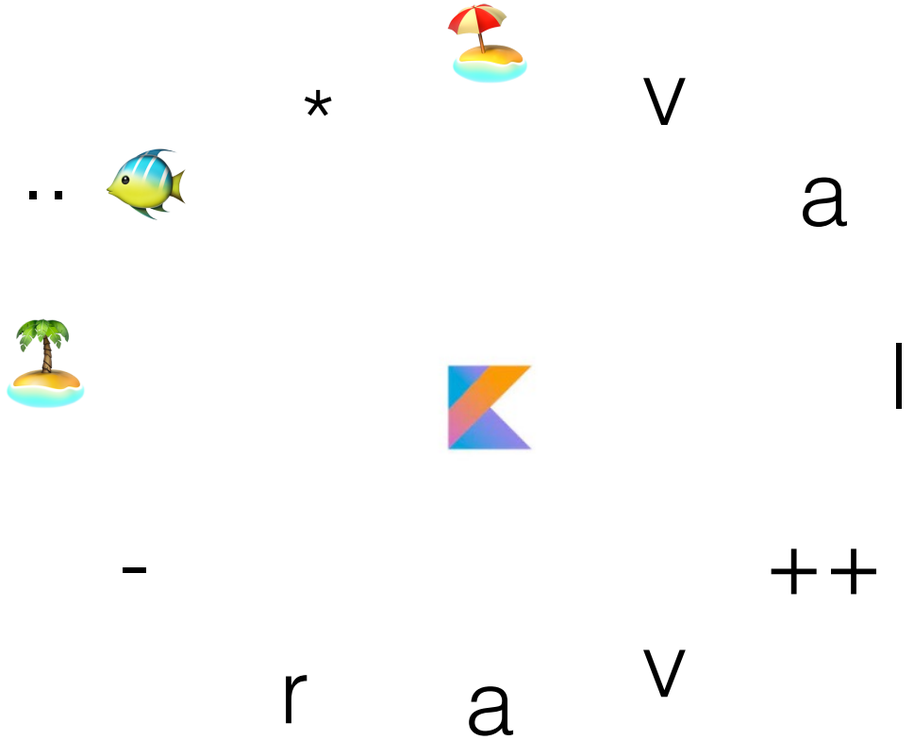

Kotlin

Macha DA COSTA

Kotlin : Variable, Opérateur, Condition

- Déclaration de variable
- Différents types de variable
- Opérateur
- Condition
- Boucle

Variable, Opérateur, Condition

Déclaration de variable

```
var mVariable = 10
val mConstant = 10 // ♥ Préférence pour l'Immuable
```

ℹ var comme variable

Référence [2] : https://antonioleiva.com/variables-kotlin/

Variable, Opérateur, Condition

Différents types de variable

```
String
Boolean
Char
Array
```

📎 Les classiques

Variable, Opérateur, Condition

Exemples de déclaration

```
val mHello: String = "Hello you”
val s = “abc"
val str = "$s length is ${s.length}" // evaluates to “abc length is 3"
```

ℹ interpolation de *String*

Variable, Opérateur, Condition

Type `Unit`

```kotlin
val mObject: Unit = Unit // Objet vide
```

ℹ ≈ `void`

Référence [2] : https://antonioleiva.com/variables-kotlin/

Variable, Opérateur, Condition

Différents types

```
List
MutableList
ArrayList
IntArray
```

📎 Les collections

Référence [Chill] : Tableau Kotlin

Variable, Opérateur, Condition

Différents types : Tableau

```
var mArray = Array(6, { i: Int -> MagicCircle(i, i) })
```

Variable, Opérateur, Condition

Différents types : Tableau

```
var mArray = arrayOf("Jake", "Jill", "Ashley", "Bill")
```

Variable, Opérateur, Condition

Différents types : Liste

```
var mutableArray: MutableList<MagicCircle> = ArrayList()
```

Variable, Opérateur, Condition

Différents types

```
Double (64)
Float (32)
Long (64)
Int (32)
Short (16)
Byte (8)
```

📎 Les nombres

Référence [K] : Kotlin, Basic types

Variable, Opérateur, Condition

Exemple de déclaration

ℹ déclaration type optionnelle

```
val mX = 20 // Conversion implicite en Int
val mZ = 10F // Conversion implicite en Float
val mY: Float = mX.toFloat() // Conversion explicite en Float
```

ℹ conversion de nombres

Référence [2] : https://antonioleiva.com/variables-kotlin/

Variable, Opérateur, Condition

Différents types : correspondances avec Java

`Double`	`double`
`Float`	`float`
`Int`	`int`
`Boolean`	`boolean`
`Double?`	`java.lang.Double`
`Float?`	`java.lang.Float`
`Int?`	`java.lang.Integer`
`String?`	`String`
`Boolean?`	`java.lang.Boolean`
`IntArray`	`int[]`
`Array<Int>`	`Integer[]`

Variable, Opérateur, Condition

Opérateur

- Unaire
- Binaire
- Tableau
- Égalité

Référence [book] : https://antonioleiva.com/kotlin-book/

Variable, Opérateur, Condition

Opérateur

Unary prefix operators

Expression	Translated to
`+a`	`a.unaryPlus()`
`-a`	`a.unaryMinus()`
`!a`	`a.not()`

Unaire

Référence [K] : Kotlin: Introduction by Jetbrains

Variable, Opérateur, Condition

Opérateur

Increments and decrements

Expression	Translated to
`a++`	`a.inc()` + see below
`a--`	`a.dec()` + see below

Unaire

Variable, Opérateur, Condition

Opérateur

Arithmetic operators

Expression	Translated to
`a + b`	`a.plus(b)`
`a - b`	`a.minus(b)`
`a * b`	`a.times(b)`
`a / b`	`a.div(b)`
`a % b`	`a.rem(b)`, `a.mod(b)` (deprecated)
`a..b`	`a.rangeTo(b)`

Binaire

Variable, Opérateur, Condition

Opérateur

Binaire

```
if (a >= 0 && a <= 10)

            ≈

if(a in 0..10 )
```

ℹ inclusif !

Variable, Opérateur, Condition

Opérateur

'In' operator

Expression	Translated to
`a in b`	`b.contains(a)`
`a !in b`	`!b.contains(a)`

Binaire

Variable, Opérateur, Condition

Opérateur

Augmented assignments

Expression	Translated to
`a += b`	`a.plusAssign(b)`
`a -= b`	`a.minusAssign(b)`
`a *= b`	`a.timesAssign(b)`
`a /= b`	`a.divAssign(b)`
`a %= b`	`a.modAssign(b)`

Binaire

Variable, Opérateur, Condition

Opérateur

Indexed access operator

Expression	Translated to
`a[i]`	`a.get(i)`
`a[i, j]`	`a.get(i, j)`
`a[i_1, ..., i_n]`	`a.get(i_1, ..., i_n)`
`a[i] = b`	`a.set(i, b)`
`a[i, j] = b`	`a.set(i, j, b)`
`a[i_1, ..., i_n] = b`	`a.set(i_1, ..., i_n, b)`

Tableau

Référence [11] : antonioleiva.com/collections

Variable, Opérateur, Condition

Opérateur

Equality and inequality operators

Expression	Translated to
`a == b`	`a?.equals(b) ?: (b === null)`
`a != b`	`!(a?.equals(b) ?: (b === null))`

Égalité

Variable, Opérateur, Condition

Opérateur

Comparison operators

Expression	Translated to
`a > b`	`a.compareTo(b) > 0`
`a < b`	`a.compareTo(b) < 0`
`a >= b`	`a.compareTo(b) >= 0`
`a <= b`	`a.compareTo(b) <= 0`

📎 Comparaison

Variable, Opérateur, Condition

Opérateur

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    var tv = findViewById(R.id.mainTextView) as TextView
}
```

Conversion

Variable, Opérateur, Condition

Condition

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
        R.id.action_settings -> {
            showAlert()
            return true
        }
        R.id.action_info -> {
            showInfo()
            return true
        }
        else -> return super.onOptionsItemSelected(item)
    }
}
```

 Référence [13] : https://antonioleiva.com/when-expression/

Variable, Opérateur, Condition

Condition

```
when (view) {
    is TextView -> toast(view.text)
    is RecyclerView -> toast("Item count : ${view.adapter.itemCount}")
    is SearchView -> toast("Current query: ${view.query}")
    else -> toast("View type not supported")
}
```

ℹ exclusif !

 Référence [13] : https://antonioleiva.com/when-expression/

Variable, Opérateur, Condition

Condition : *range*

```
val res = when {
    x in 1..10 -> "cheap"
    s.contains("hello") -> "it's a welcome!"
    v is ViewGroup -> "child count: ${v.getChildCount()}"
    else -> ""
}
```

ℹ variable dans un rang

 Référence [13] : https://antonioleiva.com/when-expression/

Variable, Opérateur, Condition

Condition : le classique

```
if (dx < 0)
    cx = 0F
else
    cx = maxX.toFloat()
```

Variable, Opérateur, Condition

Condition : Elvis

```
s2?.length ?: 0
```

Variable, Opérateur, Condition

Boucle

```
val names = arrayOf("Jake", "Jill", "Ashley", "Bill")
for (name in names) {
    toast(name)
}
```


Référence [13] : Tutos android fr : Introduction à Kotlin

Kotlin : Variable, Opérateur, Condition

</> Fonctionnelle :

📎 Déclaration de variable

📎 Type de base

📎 Opérateur `range`

📎 Condition `if` ou `when`

📎 Collection : `list` ou `array`

📎 Boucle

THE EXPERT
AT ANYTHING
WAS ONCE
A BEGINNER.

_ Helen Hayes

Références

Kotlin for Android

- TRY Kotlin
- Kotlin Workshop on Github: Slides and Questions
- https://antonioleiva.com/free-kotlin-android-course/
- ChillCoding.com : Introduction à Kotlin
- ChillCoding.com : Configurer Kotlin dans un projet Android Studio

Library

- ChillCoding.com : Utiliser des bibliothèques graphiques Kotlin dans un projet Android

Fonction d’extension

- Odelia Technologies : Les fonctions d'extension de Kotlin

Kotlin in videos

- Jake Wharton and Kotlin (DEC 2015)
- Tue Dao & Christina Lee on The Road to Kotlintown (KotlinConf 2017)
- Introduction to Kotlin Google I/O '17

Références

Why Kotlin?

- Swift is like Kotlin
- Langage Java
- API Java : Google a enfreint les brevets d'Oracle, selon la Cour Suprême
- Antonioleiva: 12 reasons to strat Kotlin for Android

Kotlin in brief

- Kotlin: pourquoi ce nouveau langage est une bonne nouvelle

Android and Kotlin

- Kotlin - Official Site
- developer.android: Get Started with Kotlin on Android

Type Kotlin

- https://code.tutsplus.com/tutorials/kotlin-from-scratch-variables-basic-types-arrays-type-inference-and-comments--cms-29328
- https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-array/index.html
- http://kotlinlang.org/docs/reference/basic-types.html#arrays

-

try.kotlinlang.org