



---

# Kotlin

---

# Concept du langage *Kotlin*

---

</> Kotlin de JetBrains



.....



.....



# Concept du langage *Kotlin*

---

♥ Moderne, Minimaliste et Multiplateforme

📱 Multiplateforme : Bibliothèque partagée

🏢 Serveur : Spring, Ktor

🌐 Web : Kotlin to JS

🤖 Android : 2017

# Concept du langage *Kotlin*

---

## ♡ Couche Multiplateforme

📎 Interopérabilité ★★☆☆

📎 Partage de la logique métier, entre les plateformes, alors que l'UI reste native

- Web
- Mobile
- Bureau

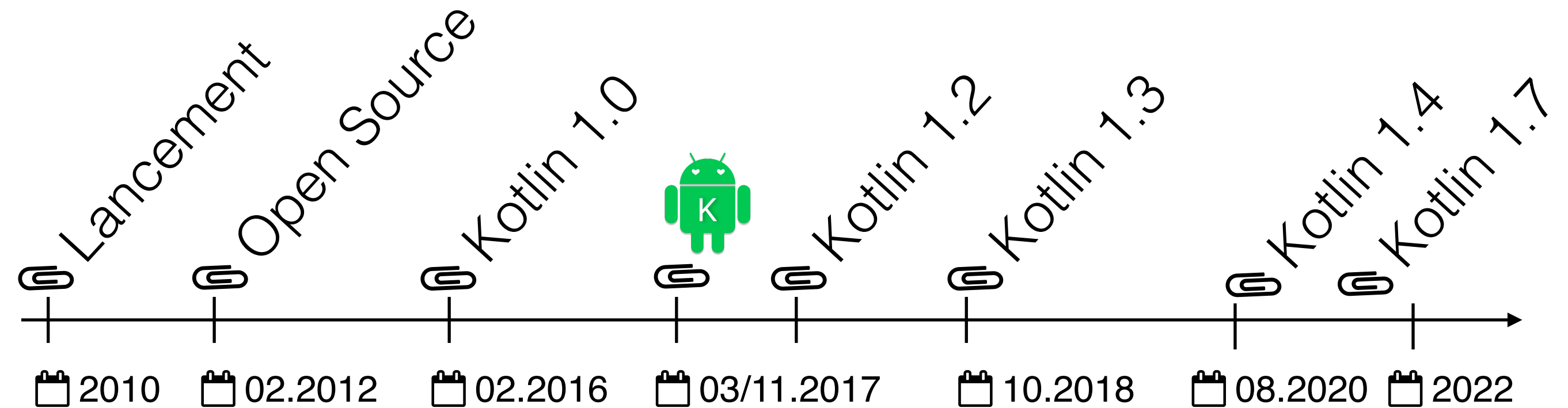
📎 Code Partagé avec application native

- Android
- iOS
- JavaScript, CSS

# Concept du langage *Kotlin*

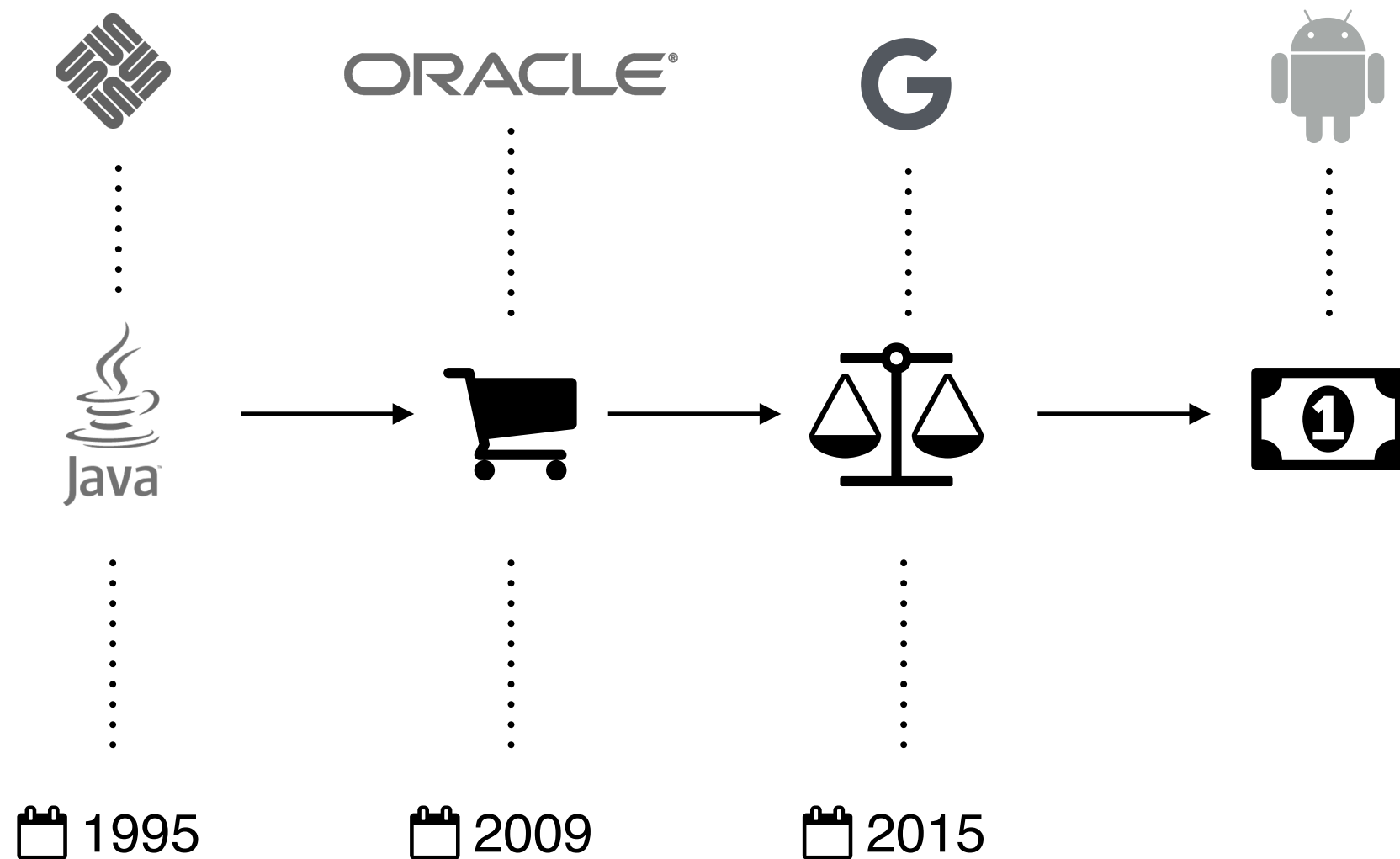
---

## </> Historique



# Concept du langage *Kotlin*

## G Pourquoi Kotlin ?

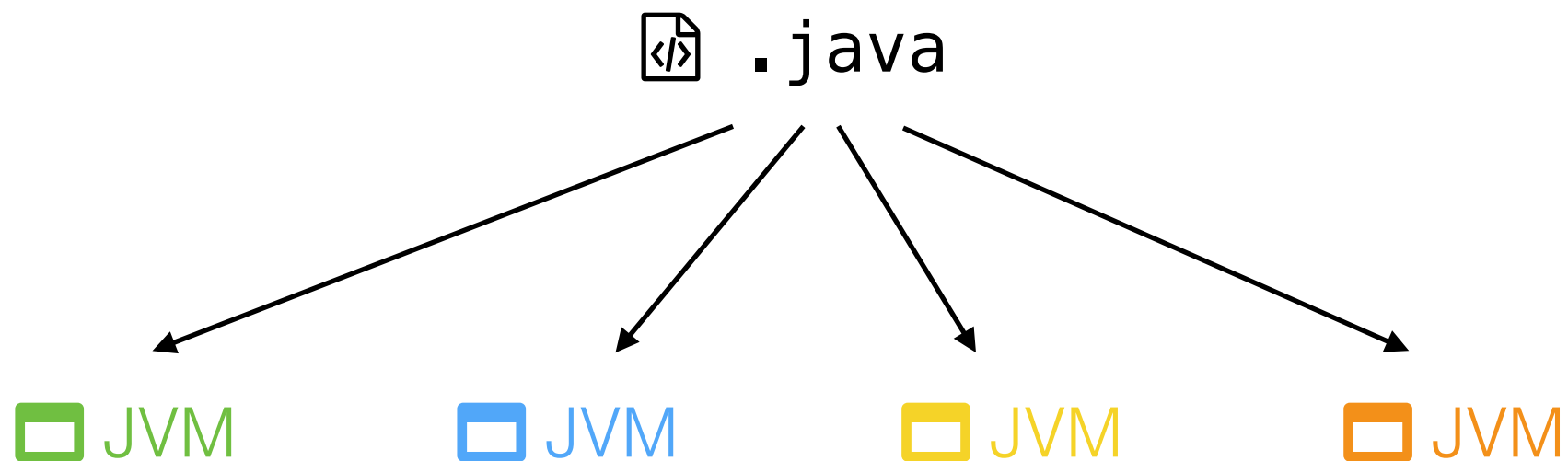


💡 L'histoire de Java

# Concept du langage *Kotlin*

---

JVM et interopérabilité



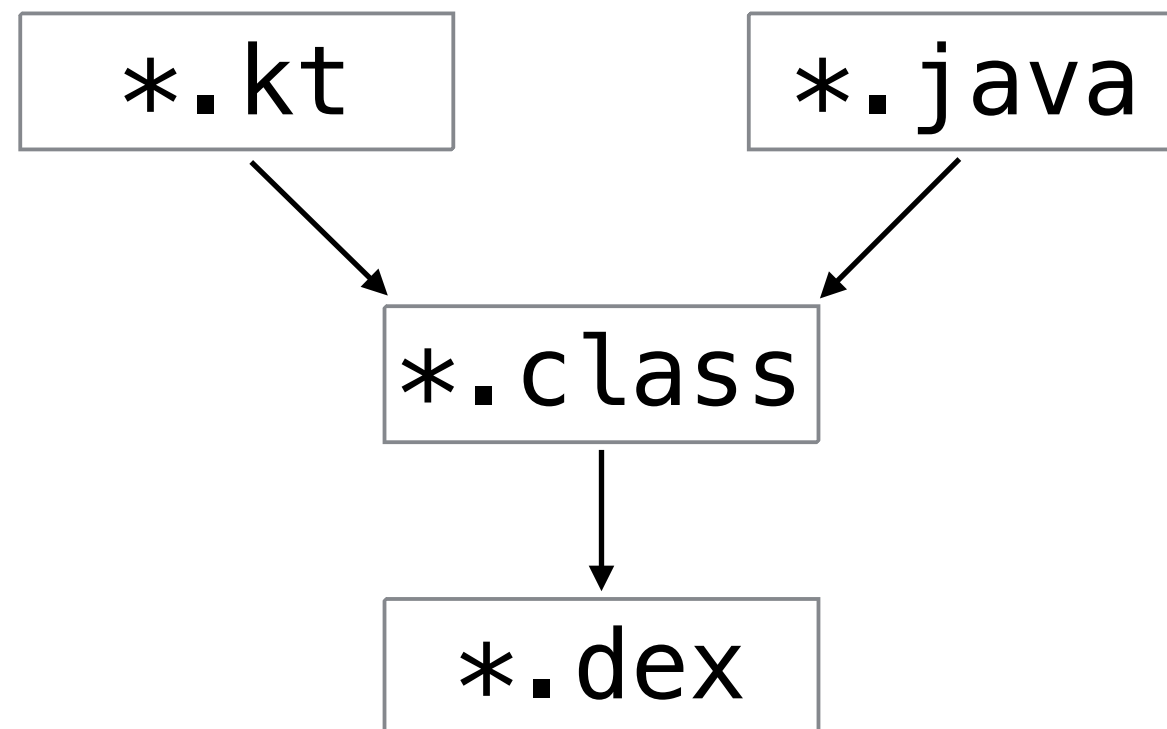
💡 Le concept de Java

# Concept du langage *Kotlin*

---

</> JVM et interopérabilité

📎 Compilé en bytecode Java



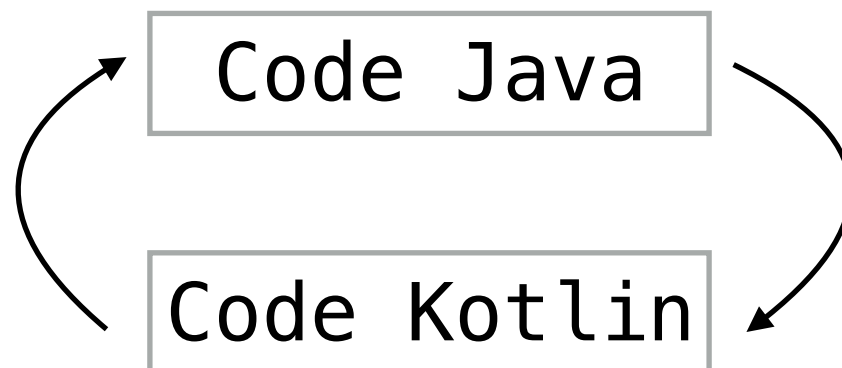


# Concept du langage *Kotlin*

---

♡ JVM et interopérabilité

📎 Cohabitation *Java* & *Kotlin*



# Concept du langage *Kotlin*

---

♥ Environnement de développement Kotlin

📎 Compilateur en ligne

```
/** * You can edit, run, and share this code. * play.kotlinlang.org */  
fun main() { println("Hello, world!!!") }
```

📎 Logiciel *IntelliJ IDEA*

Outils *REPL* (*Read Evaluate Print Loop*)

Tools > Kotlin > Kotlin REPL

# Concept du langage *Kotlin*

---

## </> Kotlin en bref

📎 Langage clair et concis

`;` → 😊

~~getter/setter~~

`Log.i(TAG, "my width is $mWidth")`

*Boilerplate* → *Lambdas*

📎 Affranchissement du `NullPointerException`

📎 Programmation orientée objet et fonctionnelle

# Concept du langage *Kotlin*

---

♥ Android et Kotlin

## 📎 Conversion automatique

`.java : Ctrl C → .kt : Ctrl V`

`'Code' > 'Convert Java File to Kotlin File'`  
`(Command + Option + Shift + K)`

# Conclusion

---

- 📎 Parce que *Kotlin* ?!?
- 📎 POO et Fonctionnelle
- 📎 JVM et interopérabilité
- 📎 Kotlin play ou IntelliJ REPL
- 📎 Structure d'une app. *Kotlin*
- 📎 Convention similaire au *Java*

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 Kotlin town \(KotlinConf 2017\)](#)
- [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](#)
- [Antoniroleiva: 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](http://try.kotlinlang.org)

The screenshot shows a web browser window with the URL `try.kotlinlang.org`. The page features the Kotlin logo and navigation links for `LEARN`, `COMMUNITY`, and `TRY ONLINE`. Below the navigation bar, there are social media icons for Facebook, Google+, GitHub, and JetBrains. A toolbar contains buttons for `Shortcuts`, `Convert from Java`, and `Fullscreen`.

The main content area is titled `Kotlin Koans > Introduction > Hello, world! > Task.kt`. On the left, a sidebar lists the koans, with `Task.kt` selected. The main panel displays the task instructions for `Simple Functions`, which ask the user to implement a `start` function that returns the string `"OK"`. The instructions mention that the `TODO()` function is used for tasks and that the user's job is to replace it with meaningful code.

At the bottom of the main panel, there is a code editor showing the following Kotlin code:

```
1 fun start() = "OK"
```

Below the code editor, there are buttons for `Check`, `Revert`, and `Show answer`.