

Kotlin BBL



Emmanuel Vinas
Expert Android & Java

 @emmanuelvinas

 emmanuel@monkeypatch.io



Igor Laborie
Expert Web & Java

 @ilaborie

 igor@monkeypatch.io



I. Water Pouring Problem

II. Live Coding

III. Android

IV. Conclusion

Pourquoi un nouveau langage ?

#3

- Écrire du code plus sûr
- Faciliter la maintenance
- Écrire plus rapidement
- Résoudre de nouveaux problèmes
- ...

- Éviter les NPE, statiquement typé
- Abordable, si on vient de Java
- Expressif et pragmatique
- Inspiré par Java, Scala, C#, Groovy, ...
- Cross-platform



JVM et Android



JavaScript



Native avec
LLVM

WATER POURING PROBLEM

Théière magique

#7

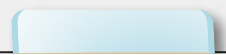


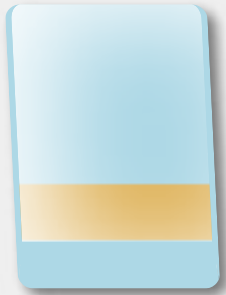


8 / 8

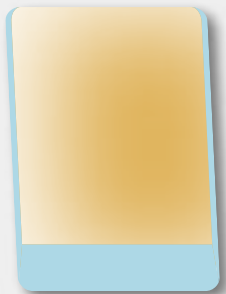


0 / 6





1 / 4



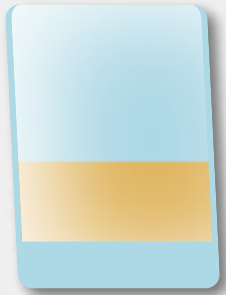
4 / 4



3 / 4



0 / 4



2 / 6



LIVE CODING



Utilisez `Alt + Shift + (Cmd|Ctrl) + K` pour convertir une classe Java en Kotlin Ou copiez du code Java dans un fichier Kotlin










En écrivant du Kotlin vous aurez plein de fun ! Le `typealias` nécessite Kotlin 1.1.



Avec les `sealed` et les `data class` on peut faire des *Abstract Data Class* Le `sealed` nécessite Kotlin 1.1.

ANDROID

CONCLUSION

-  [Koans](#)
-  [Référence](#)
-  <https://kotlin.link/>
-  [Blog](#)
-  [Forum](#)
-  [Slack](#)
-  [Kotlin Evolution and Enhancement Process](#)