

# Kotlin BBL



Emmanuel Vinas

Expert Android & Java

[@emmanuelvinas](https://twitter.com/emmanuelvinas)

[emmanuel@monkeypatch.io](mailto:emmanuel@monkeypatch.io)



Igor Laborie

Expert Web & Java

[@ilaborie](https://twitter.com/ilaborie)

[igor@monkeypatch.io](mailto:igor@monkeypatch.io)

I. Water Pouring Problem

II. Live Coding

III. Android

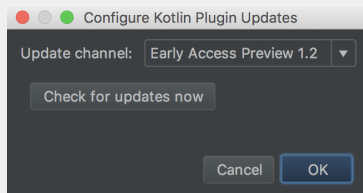
IV. Conclusion

- Installation de l'IDE

 [Android Studio 3.0](#),  [Download IntelliJ IDEA](#)

- Configuration du plugin Kotlin

Tools | Kotlin | Configure Kotlin Plugin Updates



Sélectionner **Early Access Preview 1.2**

Puis cliquer sur **Check for updates now**

- Récupérer la présentation

 <https://ilaborie.github.io/slides/kotlinHandsOn/devoxx-ma.pdf>

- `git clone  http://github.com/MonkeyPatchIo/KotlinByExample`
- `git checkout exo-server` OU `git checkout exo-web` OU  
`git checkout exo-mobile`
- `./gradlew clean assemble test`

# Pourquoi un nouveau langage ?

#5

- 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**

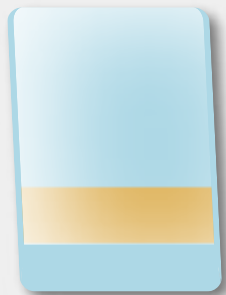
# Théière magique

#9

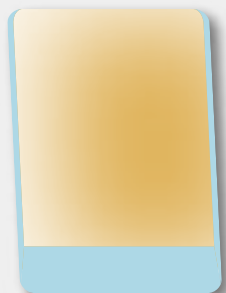


```
<div class="glass-container">  
  <div class="glass"  
    style="height: 6rem;background-image: linear  
    <span>0 / 6</span>  
</div>
```

```
<div class="glass-container">  
  <div class="glass"  
    style="height: 4rem;background-image: linear  
    <span>1 / 4</span>
```



1 / 4



4 / 4



3 / 4



0 / 4

into

```
<span class="ope">➡</span>
```

```
<div class="glass-container">
```

```
  <div class="glass"
```

```
    style="height: 8rem;background-image: linear
```



**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