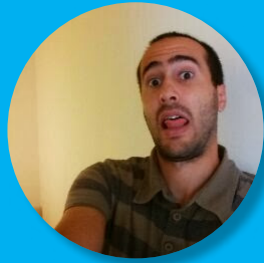


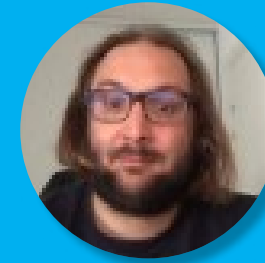
# KOTLIN PAR L'EXEMPLE



Emmanuel Vinas  
Expert Android & Java

 @emmanuelvinas

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



Igor Laborie  
Expert Java & Web

 @ilaborie

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



17h30

17h50

19h30

Présentation,  
Installation

Exercice

...

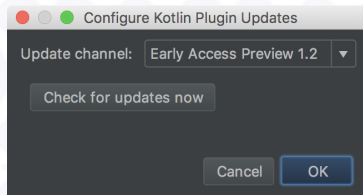
## NAME / CODE

1. Installation de l'IDE

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

2. Configuration du plugin Kotlin

Tools | Kotlin | Configure Kotlin Plugin Updates



Sélectionner Early Access Preview 1.2

Puis cliquer sur Check for updates now

3. Récupérer la présentation

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

4. `git clone TODO`

5. `./gradlew downloadDependencies`

# Pourquoi un nouveau langage ?

#4

- Écrire du code plus sûr
- Faciliter la maintenance
- Écrire plus rapidement
- ...



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



JVM et Android



JavaScript



Native avec  
LLVM

# WATER POURING PROBLEM



# Théière magique

#S



#DevoxxMa @ilaborie @EmmanuelVinas #kotlin

DEVOXX<sup>TM</sup>  
MOROCCO



Fill



1 / 4



4 / 4

Empty



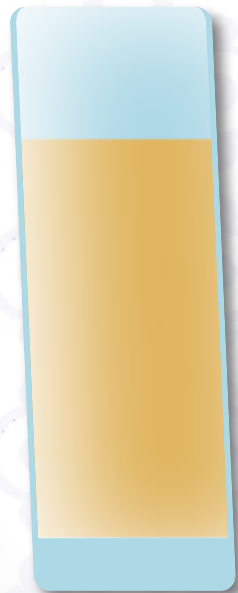
3 / 4



4 / 4



Pour



6 / 8

into

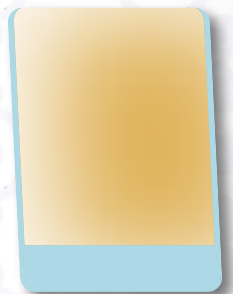


2 / 6



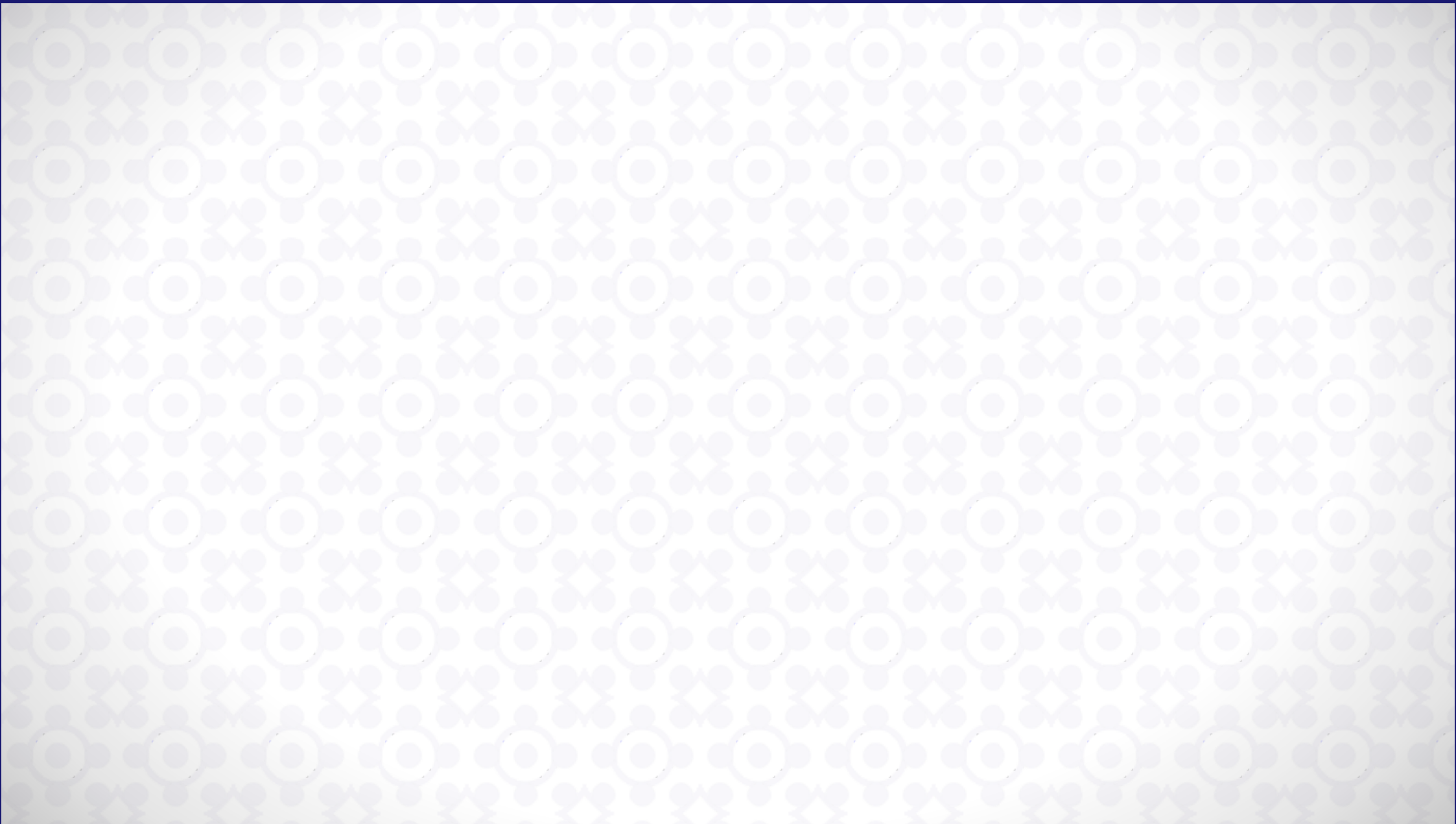
2 / 8

,



6 / 6





# POUR DÉMARRER

```
fun main(args: Array<String>) {  
    println("Hello Devoxx Morocco !")  
}
```

## Tips

Utilisez Alt + Shift + (Cmd|Ctrl) + K pour convertir une classe Java en Kotlin

Ou copiez du code Java dans un fichier Kotlin

```
data class Glass(val capacity: Int, val current: Int = 0) {  
    init {  
        require(capacity > 0) {  
            "Capacity: $capacity should be > 0"  
        }  
        require(current in 0..capacity) {  
            "Current: $current should be into [0, $capacity]"  
        }  
    }  
}  
  
typealias State = List<Glass>
```

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

```
sealed class Move

data class Empty(val index: Int) : Move()

data class Fill(val index: Int) : Move()

data class Pour(val from: Int, val to: Int) : Move() {
    init {
        require(from != to)
    }
}
```

## Tips

Avec les `sealed` et les `data class` on peut faire des *Abstract Data Class*

Le `sealed` nécessite Kotlin 1.1.



# EXERCICES



Serveur avec  
SpringBoot 2

---

Implémentation du solveur  
côté serveur



Android

---

Définition et affichage de la  
solution sous Android

Navigateur avec  
KotlinJS

---

Définition et affichage de la  
solution dans un  
navigateur

Freestyle

---

Implémenter sans guide

## Pré-requis

- Java 8
- ??? Android Studio 3
- ??? Android SDK ?

## Exercices

???

## Pré-requis

- Java 8
- Un IDE qui marche bien avec Kotlin comme  IntelliJ IDEA CE

## Exercices

- Exercice 1: les bases
- Exercice 2: un peu plus loin
- Exercice 3: (SpringBoot) injection, routes, reactor, ...

## Variantes


- ➡ SparkJava
- ➡ Ktor
- ...



## Pré-requis

- Java 8
- NodeJS LTS ou stable (8+)
- Un IDE qui marche bien avec Kotlin comme  IntelliJ IDEA CE
- Un navigateur moderne

## Exercices

- Exercice 1: les bases
- Exercice 2: construire du HTML avec le DSL  kotlinx-html
- Exercice 3: créer des actions
- Exercice 4: implémenter la résolution du problème

## Variantes

- Utiliser  RxJS
- Utiliser un autre gestionnaire d'état










## Pré-requis

- Java 6+
- Un IDE qui marche bien avec Kotlin comme  IntelliJ IDEA CE

## Exercice

- Résoudre le problème du transvasement en Kotlin

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