



KOTLIN PAR
L'EXEMPLE

Emmanu
/inas

Emmanuel Vinas

Expert Android & Java

 @emmanuelvinas

 emmanuel@monkeypatch.io

Igor
Laborie

Igor Laborie

Expert Java & Web

 @ilaborie

 igor@monkeypatch.io





Roadmap

#1

13h30

13h45

15h30

Présentation,
Installation

Exercice

...



Vous avez déjà une connexion internet correcte.



Sinon utilisez le wifi Monkey / bananaTree.

- Installation de l'IDE

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

- Configuration éventuelle du plugin Kotlin

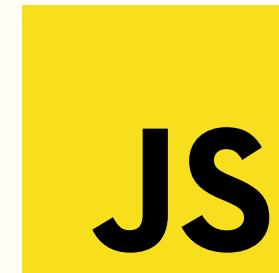
Tools | Kotlin | Configure Kotlin Plugin
Updates

1. `git clone http://github.com/MonkeyPatchIo/KotlinByExample`
2. `git checkout {branch}` avec la branche dans `exo-mobile`,
`exo-server`, `exo-web`
3. `./gradlew clean assemble test` (les tests doivent être en erreurs)
4. (Optionel) Récupérer la présentation :  <http://bit.ly/kotlin-lab>

- Dévoilé en 2011 par  JetBrains
- v1.0 en 2016, v1.1 et v1.2 en 2017
- Expressif et pragmatique
- Sûr: statiquement typé, null-safety
- Inspiré par Java, Scala, C#, Groovy, ...
- Abordable, si on vient de Java
- Cross-platform



JVM et Android



JavaScript



Native avec
LLVM

WATER POURING PROBLEM

Tonneau magique

18



@breizhcamp @emmanuelVinas @ilaborie #Kotlin #



8 / 8



0 / 6

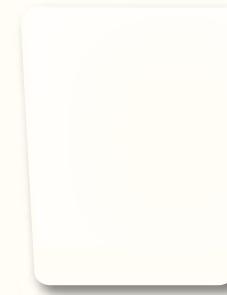


1 / 4

Fill



1 / 4



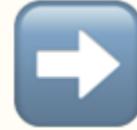
4 / 4

Pour

into

6 / 8

2 / 6



2 / 8

6 / 6

Empty



3 / 4



0 / 4

POUR DÉMARRER

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



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)
    }
}
```



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



Définition et affichage de la
solution sous Android

Navigateur avec
KotlinJS

Définition et affichage de la
solution dans un navigateur

Freestyle

Implémenter sans guide

exo-server

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, ...

exo-mobile

Pré-requis

- Java 8
-  Android Studio 3
- Android SDK 27

Exercices

- Exercice 1: les bases
- Exercice 2: un peu plus loin
- Exercice 3: android

exo-web

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

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](#)