



# KOTLIN PAR L'EXEMPLE



Alexandre Delattre

Expert Android & Java

 @alexandre\_del31

 alexandre@monkeypatch.io

---



Igor Laborie

Expert Java & Web

 @ilaborie

 igor@monkeypatch.io

---





# Roadmap

#1

10h05

10h15

11h45

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-Lite`  
ou `git clone http://github.com/MonkeyPatchIo/KotlinByExample`
2. `git checkout {branch}` avec la branche dans `exo-lite` ou  
`exo-mobile`, `exo-server`, `exo-web`
3. Pour mobile, serveur, ou web : `./gradlew clean assemble test`  
(les tests doivent être en erreurs)

# WATER POURING PROBLEM

# Tonneau magique

#6



@SunnyTech\_MTP @alexandre\_del31 @ilaborie #Kotlin #🍺





8 / 8



0 / 6



1 / 4

# Remplir

#8

Fill



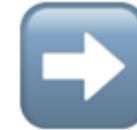
1 / 4



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 Sunny-Tech !")  
}
```



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

Basique

Implémenter le solveur  
sans libs

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

## Pré-requis

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

## Exercice

- Résoudre le problème du transvasement en Kotlin

## 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

-  [Koans](#)

---
-  [Référence](#)

---
-  <https://kotlin.link/>

---
-  [Blog](#)

---
-  [Forum](#)

---
-  [Slack](#)

---
-  [Kotlin Evolution and Enhancement Process](#)

