Introduction au développement Android

Sources: Android

- Udacity “Kotlin Bootcamp for Programmers” (Codelab)
- Udacity “Developing Android Apps with Kotlin”
- Android Fundamentals V2 (slides, V1)

À lire / regarder:

- Kotlin Doc
- Android Doc
- Android Developer Training courses
- Android Jetpack (Videos)
- Advanced Codelabs

Hello Kotlin

Kotlin:

- Peu verbeux
- Moderne
- Java Interop
- Développé par JetBrains
- Kotlin everywhere: Android, Java, Backend, JS, scripts, ...

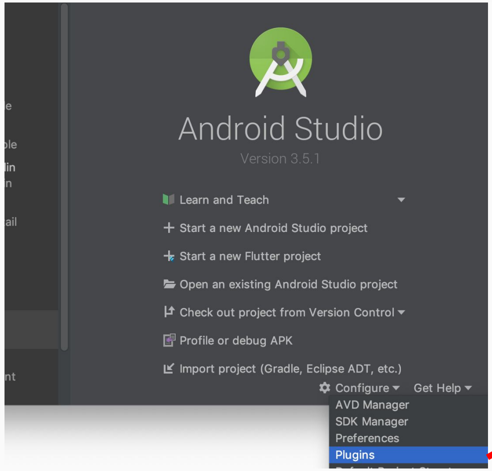
Résumé Java VS Kotlin

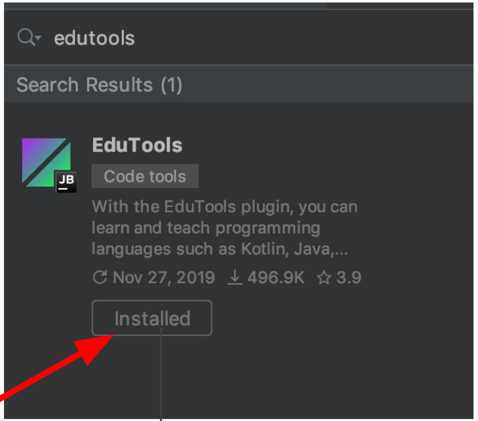
Particularités principales

- Nullables (Interop: `@Nullable`):
  ```
  val variable: Type? = null
  variable?.safe() ?: default() / variable!!.unsafe()
  ```
- Typage statique inféré
- `final` object avec `val`
- `final` class par défaut (`open` sinon)
- `static` -> `companion object {...}`
- Lambdas: `val add: (Int, Int) -> Int = { var1, var2 -> var1 + var2 }`
- `when(variable) { case1 -> {...}... }`

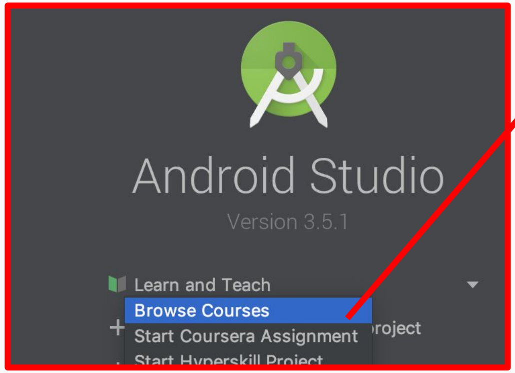
Kotlin Koans

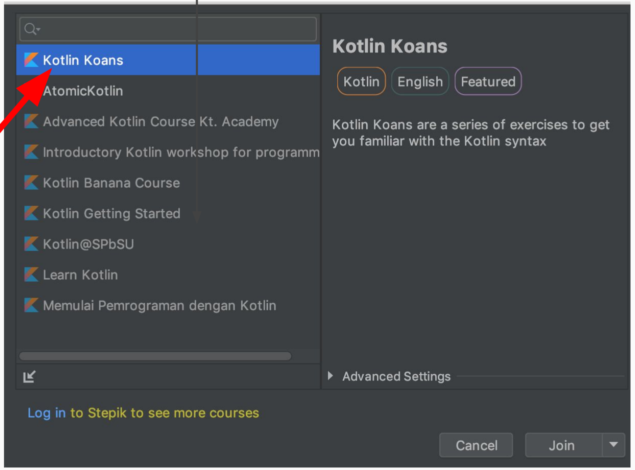
try.kotl.in ou :


Android Studio
Version 3.5.1
Learn and Teach
Start a new Android Studio project
Start a new Flutter project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Configure
Get Help
AVD Manager
SDK Manager
Preferences
Plugins



With the EduTools plugin, you can learn and teach programming languages such as Kotlin, Java,...





Kotlin Koans are a series of exercises to get you familiar with the Kotlin syntax

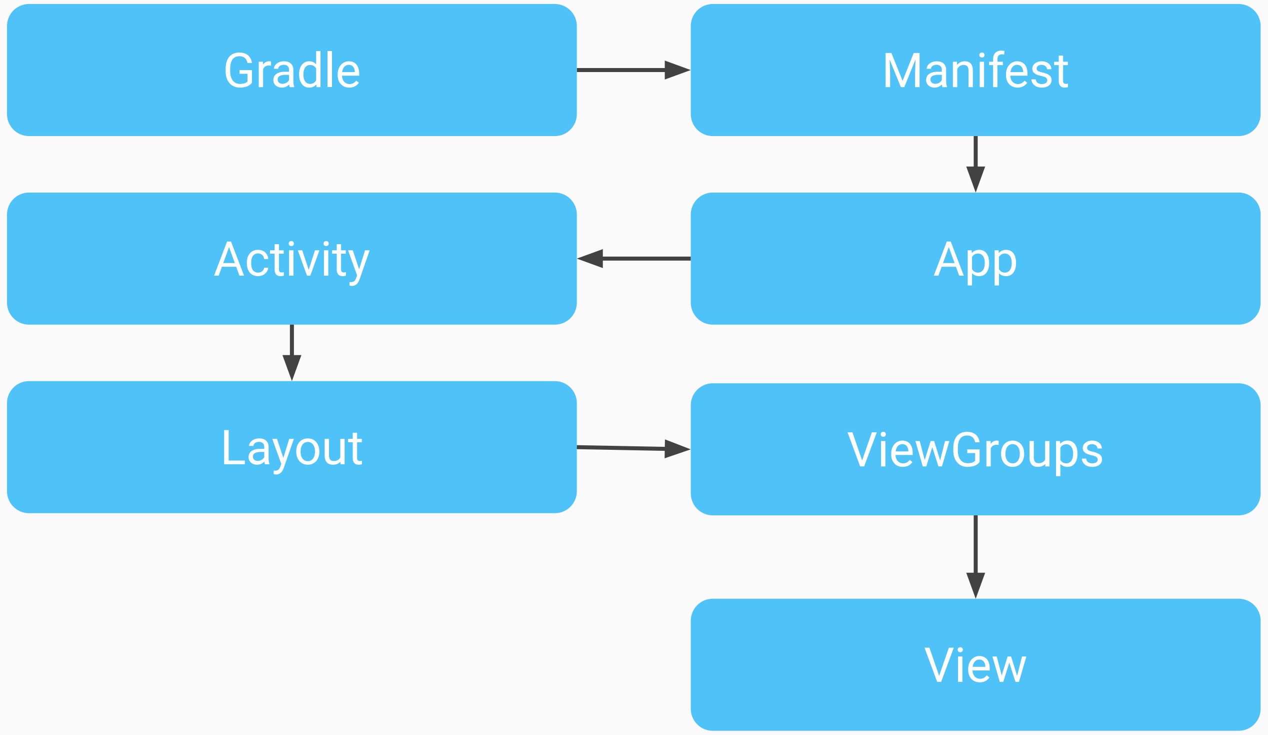
Plus loin avec Kotlin

- Extension functions: `fun String.reverse(): String {...}`
- *Smart casts:* `if (optional != null) { optional.safe() }`
- Delegates `class SomeClass : SomeInterface by SomeImplementation {}`
- Lambda for SAM: `button.setOnCLickListener {...}`
- List & streams:
  `list.filter { … } / stream.asSequence().filter(...)`
- Iterators:
  `for (element in iterator) / iterator.asSequence().filter { … }`
- Specified *returns:* `fun method() { … someLambda { return@method } …`

Hello Android

- Nombreux utilisateurs
- Devices très disparates
- Phone, Tablet, TV, Watch, Auto, Things, Chrome OS
- Versions d’OS anciennes
- Language : Java et Kotlin
- IDE : Android Studio

Éléments d'une app Android



App components

Activity / Fragment
≈ Screen Controller

Service
≈ Headless Process

BroadcastReceiver
≈ Event Listener

ContentProvider
≈ Shared App Data API

Activity / Fragment

Component le plus important.

Rôle: Fait le lien entre le Layout et la logique de l’app

Attention: Éviter la tendance à mettre toute l’app dans Activity

Fragment ≈ SubActivity

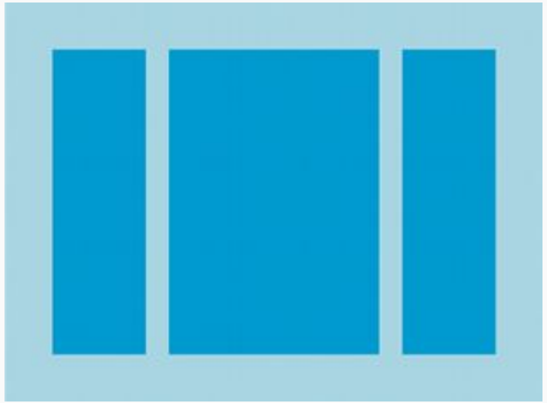
Layouts

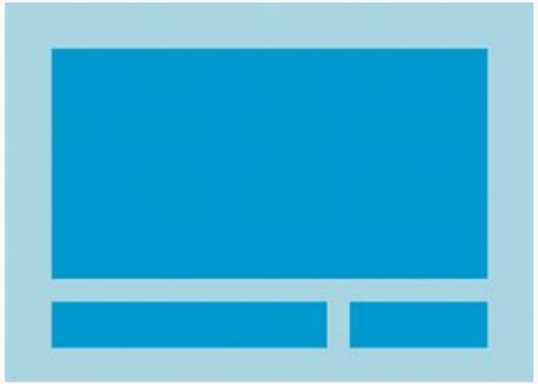
Fichier **XML** décrivant un écran (ou une partie).

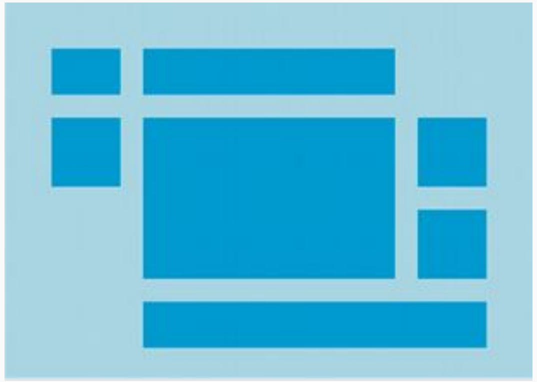
ViewGroup: View contenant d'autres Views, avec diverses règles d'affichage: LinearLayout, RelativeLayout, ConstraintLayout, Stack, ...

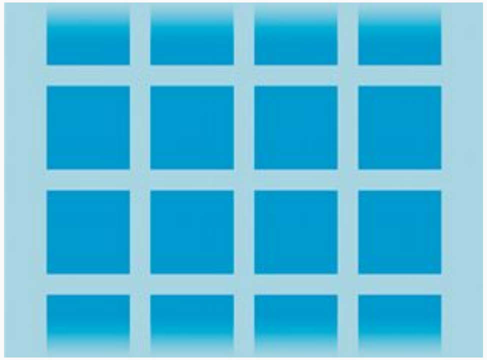
View: Élément graphique de l'interface: Text, Image, Button

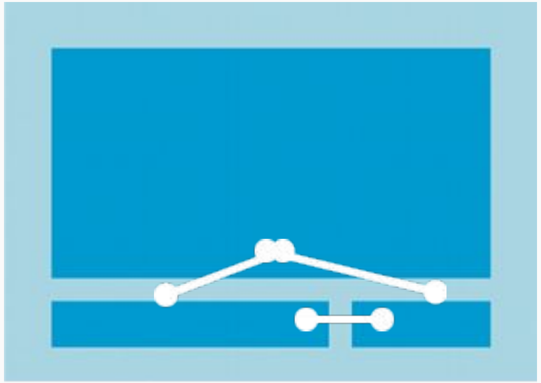
Layouts - ViewGroup


LinearLayout


RelativeLayout


TableLayout


GridLayout


ConstraintLayout

Views

Élément **XML** décrivant une vue.

Layout: layout_weight, layout_width / layout_height -> match_parent / wrap_content

Dimens: density independent pixel (dp)

Visibility: VISIBLE, INVISIBLE, GONE

Lien avec le code : *android:id="@+id/my_id"*

Kotlin sur Android

- Tous les avantages de Kotlin
- Conversion avec Android Studio
- Android KTX
- Synthetics (~~ButterKnife~~)
- Lambdas: setOnClickListener
- Data class = POJOs (+ Moshi)

[Codelab: Taking advantage of Kotlin]

Kotlin sur Android

Pas vraiment de désavantages car équivalent à Java et interop possible

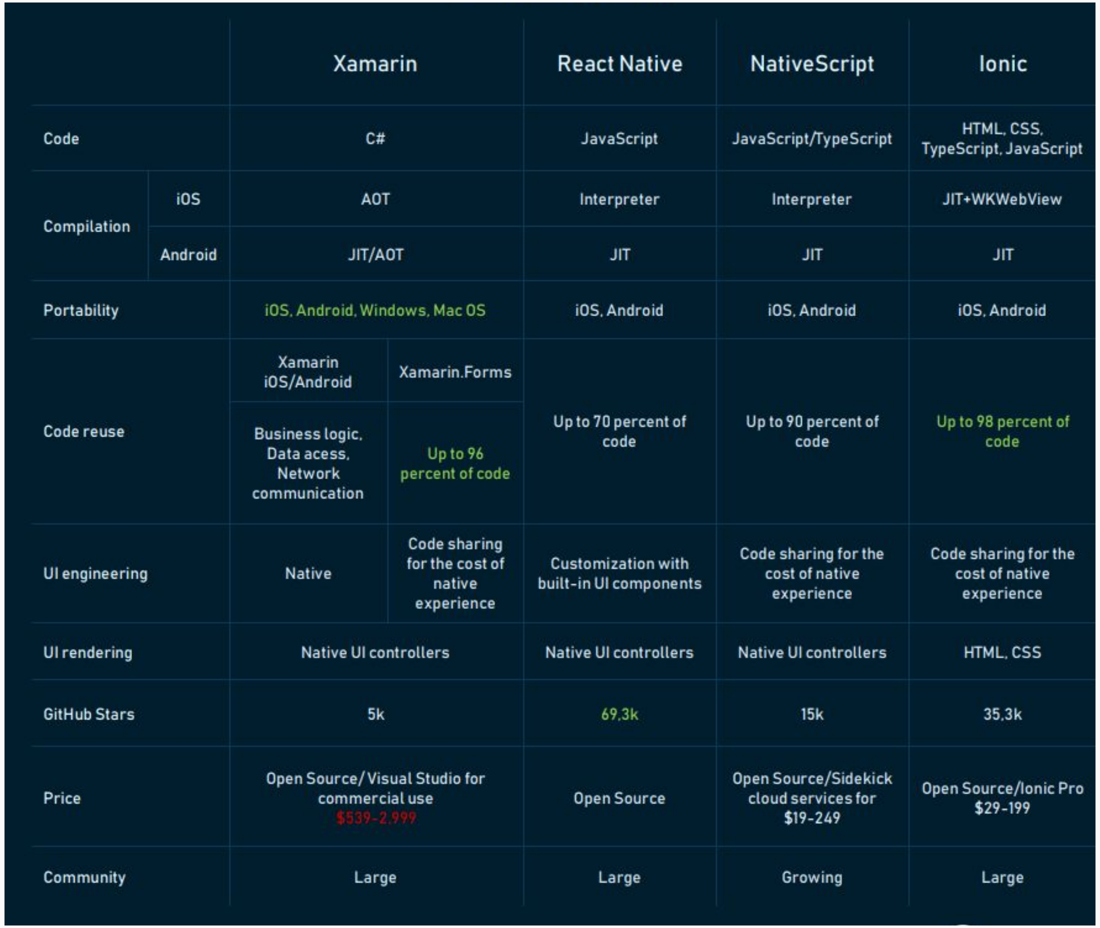
Mais:
⚠️ La compilation est plus lente si on utilise les 2

⚠️ Attention à ne pas être dépassés par les features de Kotlin, il vaut mieux rester simple et clair qu'économiser la moindre ligne de code

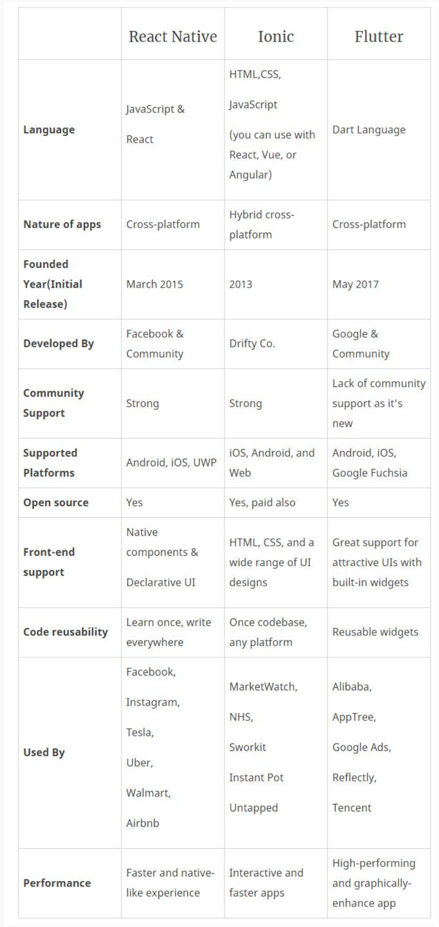
↳ "Favor readability over minimizing lines of code. It's easy to go overboard with Kotlin syntactic sugar." -- Documentation Android

Cross-Plateform

- Permet de coder une seule fois
- On perd souvent les possibilités spécifiques ou récentes des OS (“PGCD”)
- On perd parfois aussi en performances
- Programmation à base **Components** à la **React**
- Apple et Google s’en inspirent: SwiftUI, Jetpack Compose, Flutter



		Xamarin		React Native	NativeScript	Ionic
Code		C#		JavaScript	JavaScript/TypeScript	HTML, CSS, TypeScript, JavaScript
Compilation	iOS	AOT		Interpreter	Interpreter	JIT+WKWebView
	Android	JIT/AOT		JIT	JIT	JIT
Portability		iOS, Android, Windows, Mac OS		iOS, Android	iOS, Android	iOS, Android
Code reuse		Xamarin iOS/Android	Xamarin.Forms	Up to 70 percent of code	Up to 90 percent of code	Up to 98 percent of code
		Business logic, Data acess, Network communication	Up to 96 percent of code			
UI engineering		Native	Code sharing for the cost of native experience	Customization with built-in UI components	Code sharing for the cost of native experience	Code sharing for the cost of native experience
UI rendering		Native UI controllers		Native UI controllers	Native UI controllers	HTML, CSS
GitHub Stars		5k		69.3k	15k	35.3k
Price		Open Source/ Visual Studio for commercial use $539-2.999		Open Source	Open Source/Sidekick cloud services for $19-249	Open Source/Ionic Pro $29-199
Community		Large		Large	Growing	Large



	React Native	Ionic	Flutter
Language	JavaScript & React	HTML,CSS, JavaScript (you can use with React, Vue, or Angular)	Dart Language
Nature of apps	Cross-platform	Hybrid cross-platform	Cross-platform
Founded Year(Initial Release)	March 2015	2013	May 2017
Developed By	Facebook & Community	Drifty Co.	Google & Community
Community Support	Strong	Strong	Lack of community support as it's new
Supported Platforms	Android, iOS, UWP	iOS, Android, and Web	Android, iOS, Google Fuchsia
Open source	Yes	Yes, paid also	Yes
Front-end support	Native components & Declarative UI	HTML, CSS, and a wide range of UI designs	Great support for attractive UIs with built-in widgets
Code reusability	Learn once, write everywhere	Once codebase, any platform	Reusable widgets
Used By	Facebook, Instagram, Tesla, Uber, Walmart, Airbnb	MarketWatch, NHS, Sworkit, Instant Pot, Untapped	Alibaba, AppTree, Google Ads, Reflectly, Tencent
Performance	Faster and native-like experience	Interactive and faster apps	High-performing and graphically-enhance app

Cross-platform: Comparaison

TD - Introduction à Android

Google Codelabs: Android Kotlin Fundamentals (À partir de **02.1**)

Ajouter les dépendances suivantes dans app/build.gradle d’un projet vide et builder (pour éviter les problèmes de réseau au TD suivant)

```
dependencies {
  // ...
  implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.2'
  implementation 'com.squareup.retrofit2:retrofit:2.6.1'
  implementation 'com.squareup.moshi:moshi:1.9.1'
  implementation 'androidx.recyclerview:recyclerview:1.0.0'
  implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.2'
   // ...
}
```