

Dev Android

Joseph Azar 2022/2023



Qu'allons nous faire?

- 10 séances de TPs, chacune de 2 heures => 20 H
 - Construire différentes applications pour créer un petit portfolio
- 10 séances de TDs, chacune de 1 heures => 10 H
 - Présentations
 - Tutoriels - Codelabs
- Prérequis: JAVA, POO

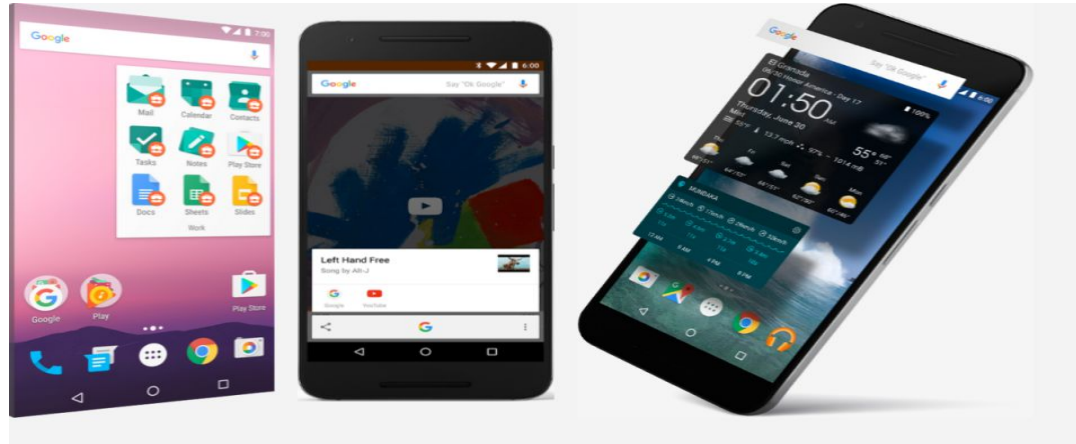


Qu'est-ce qu'Android ?

- Android est un **système d'exploitation** et une **plate-forme de programmation** développés par Google pour les téléphones mobiles et autres appareils mobiles, tels que les tablettes.
- Il peut fonctionner sur de nombreux appareils différents de nombreux fabricants différents.
- Android comprend un kit de développement logiciel (**SDK**) qui vous aide à écrire du code original et à assembler des modules logiciels pour créer des applications pour les utilisateurs d'Android.
- Android fournit également un marché pour distribuer des applications.

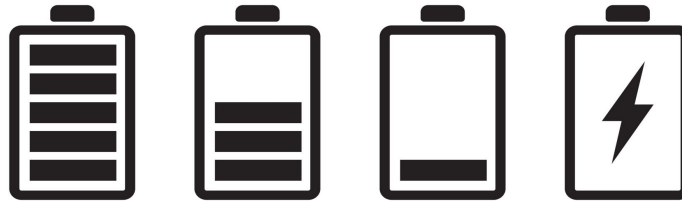
Pourquoi développer des applications pour Android ?

- Les développeurs choisissent de développer pour Android afin d'atteindre la majorité des utilisateurs d'appareils mobiles.
- En tant que plate-forme mobile la plus populaire au monde, Android alimente des centaines de millions d'appareils mobiles dans plus de 190 pays à travers le monde.



Pourquoi développer des applications pour Android ?

- La plate-forme Android, basée sur le noyau **Linux**, est principalement conçue pour les appareils mobiles à écran tactile tels que les téléphones portables et les tablettes.
- Android est conçu pour gérer les processus afin de maintenir la consommation d'énergie au minimum, ce qui permet une utilisation plus longue de la batterie.



En résumé, Android est

- Système d'exploitation mobile basé sur le noyau Linux
- Interface utilisateur pour écrans tactiles
- Utilisé sur plus de 80% de tous les smartphones
- Alimente des appareils tels que des montres, des téléviseurs et des voitures
- Plus de 2 millions d'applications Android sur Google Play Store
- Hautement personnalisable pour les appareils / par les fournisseurs
- Open source
- Gestes tactiles : balayer, taper, pincer
- Clavier virtuel pour les caractères, les chiffres et les emoji
- Prise en charge du Bluetooth, des contrôleurs USB et des périphériques

Android et capteurs

- Les capteurs peuvent découvrir l'action de l'utilisateur et réagir
- Le contenu de l'appareil tourne au besoin
- La marche ajuste la position sur la carte
- L'inclinaison dirige une voiture virtuelle ou contrôle un jouet physique
- Se déplacer trop vite désactive les interactions de jeu



Écosystème Android



Android Software Developer Kit (SDK)

- Outils de développement (débugueur, moniteurs, éditeurs)
- Bibliothèques (cartes, wearables)
- Périphériques virtuels (émulateurs)
- Documentation (developers.android.com)
- Exemple de code
- Integrated Development Environment (IDE) ⇒ Android Studio

La plate-forme Android

**Android is
“many things”**

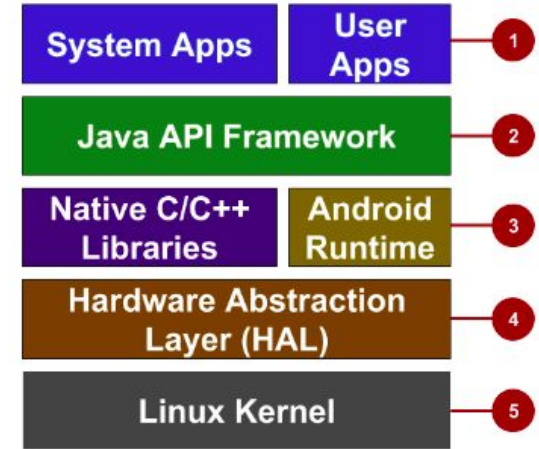
It's a Software Stack



La plate-forme Android

**Android is
“many things”**

It's a Software Stack



La plate-forme Android

Applications : vos applications vivent à ce niveau, ainsi que les applications système de base pour la messagerie électronique, la messagerie SMS, les calendriers, la navigation sur Internet et les contacts.



La plate-forme Android

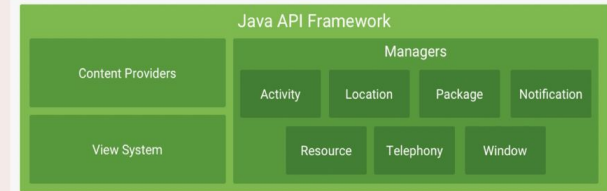
Le Java API Framework est l'endroit où nous, les développeurs, passons la plupart du temps.

Toutes les fonctionnalités de développement Android, telles que:

- les composants d'interface utilisateur
- la gestion des ressources
- la gestion du cycle de vie

sont disponibles via des interfaces de programmation d'application (API). Vous n'avez pas besoin de connaître les détails du fonctionnement des API. Il vous suffit d'apprendre à les utiliser.

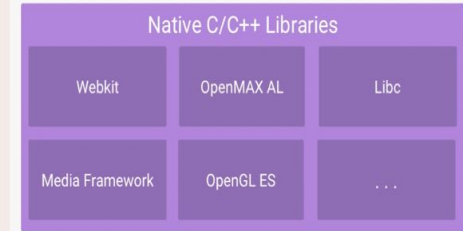
Java API Framework



La plate-forme Android

- Chaque application s'exécute dans son propre processus, avec sa propre instance de l'environnement d'exécution Android.
- Android inclut un ensemble de bibliothèques d'exécution principales qui fournissent la plupart des fonctionnalités du langage de programmation Java.
- De nombreux composants et services du système Android de base sont construits à partir de code natif qui nécessite des bibliothèques natives écrites en C et C++.
- Ces bibliothèques natives sont disponibles pour les applications via le framework d'API Java.

Native C/C++ Libraries



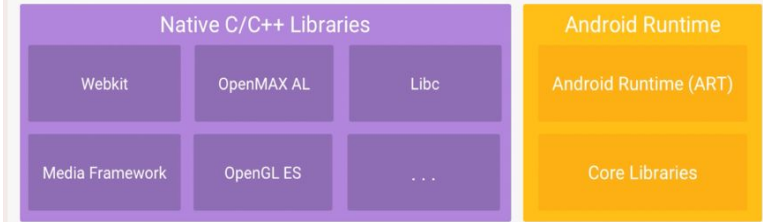
La plate-forme Android

ART ou Android Runtime est un runtime Android qui utilise Ahead Of Time (AOT).

L'ART est ce qui gouverne le système d'exploitation Android.

En utilisant AOT, ce qu'il fait, c'est qu'il convertit ou compile l'ensemble du code de langage de haut niveau en code de niveau machine et au moment de l'installation de l'application et non dynamiquement au fur et à mesure que l'application s'exécute.

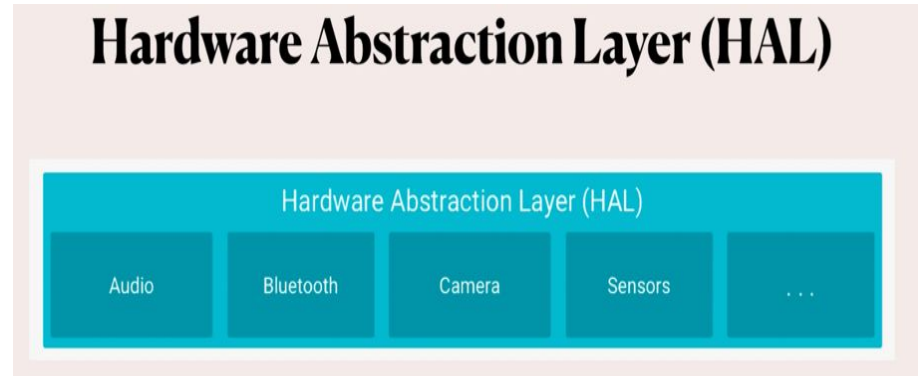
Android Runtime (ART)



La plate-forme Android

Cette couche fournit des interfaces standard qui exposent les capacités matérielles de l'appareil à la structure d'API Java de niveau supérieur.

Le HAL se compose de plusieurs modules de bibliothèque, chacun implémentant une interface pour un type spécifique de composant matériel, tel que la caméra ou le module Bluetooth.



La plate-forme Android

La base de la plate-forme Android est le noyau Linux.

Les couches au-dessus du noyau Linux s'appuient sur le noyau Linux pour le threading, la gestion de la mémoire de bas niveau et d'autres fonctionnalités sous-jacentes.

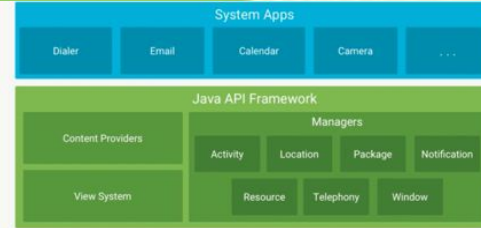
L'utilisation d'un noyau Linux permet à Android de tirer parti des fonctionnalités de sécurité basées sur Linux et permet aux fabricants d'appareils de développer des pilotes matériels pour un noyau bien connu.



La plate-forme Android

Overview

Our stack level as developers

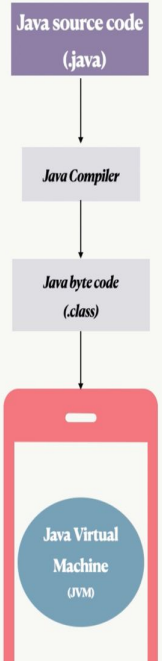


JVM, Dalvik, et ART

Une machine virtuelle Java est une machine virtuelle qui permet à un ordinateur d'exécuter des programmes Java ainsi que des programmes écrits dans d'autres langages qui sont également compilés en bytecode Java.

Java Compilation Process

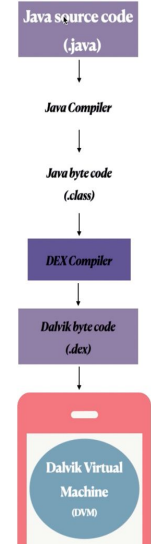
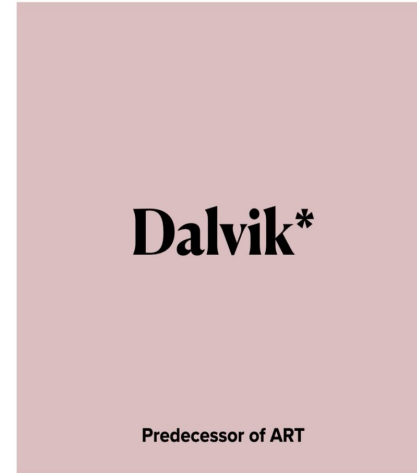
An Overview



JVM, Dalvik, et ART

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour qu'un appareil puisse exécuter efficacement plusieurs machines virtuelles.

La machine virtuelle Dalvik exécute des fichiers au format Dalvik Executable (.dex) qui est optimisé pour une empreinte mémoire minimale.



JVM, Dalvik, et ART



JVM, Dalvik, et ART

Quelles sont les différences entre *ART* et *Dalvik* ?

Réponse

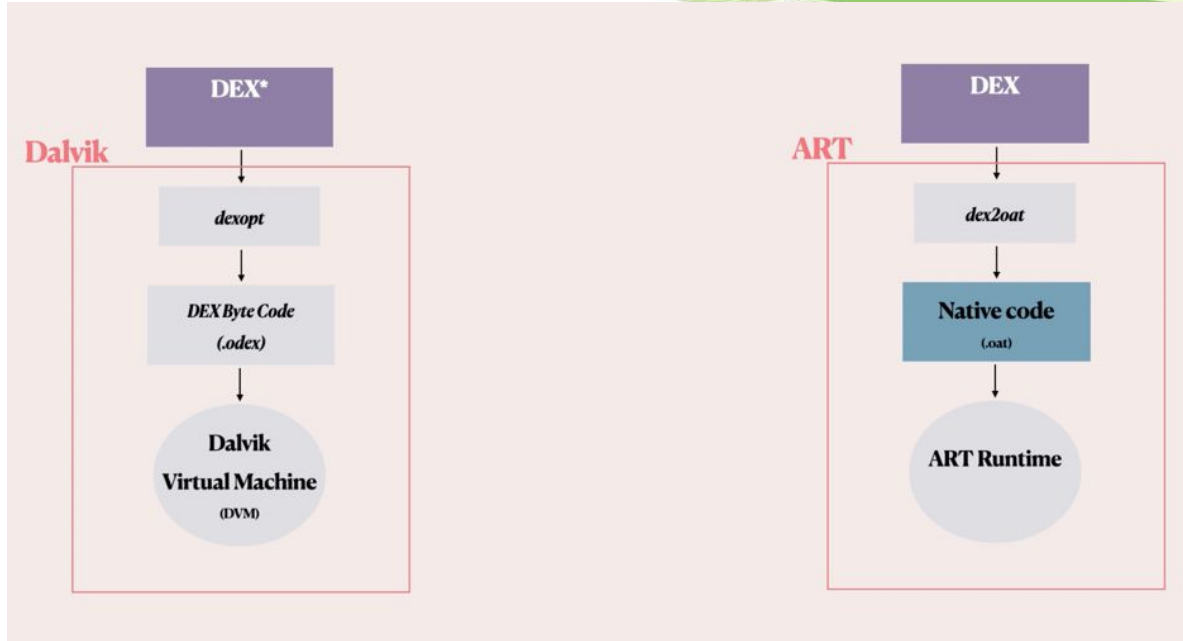
- **Approche** : ART utilise l'approche AOT (Ahead Of Time) et compile l'intégralité du code pendant le temps d'installation, mais Dalvik utilise l'approche JIT (Just In Time) et ne respecte qu'une partie du code lors de l'installation et le reste du code sera compilé dynamiquement .
- **Temps** de démarrage : Comparé à Dalvik, ART prend plus de temps à redémarrer car le cache est construit la première fois. Donc, le démarrage est lent.
- **Espace** : Étant donné que ART utilise l'approche AOT, il a donc besoin de plus d'espace lors de l'installation. Alors que Dalvik utilise l'approche JIT, les téléphones mobiles ayant moins de stockage peuvent utiliser Dalvik.
- **Batterie** : ART augmente considérablement les performances de la batterie grâce à l'approche AOT. Alors que le Dalvik utilise l'approche JIT, cela se traduit par une utilisation accrue de la batterie.
- **Ramassage des ordures** : ART a un meilleur ramassage des ordures que Dalvik.

JIT and AOT

Just In time

Ahead of Time

JVM, Dalvik, et ART

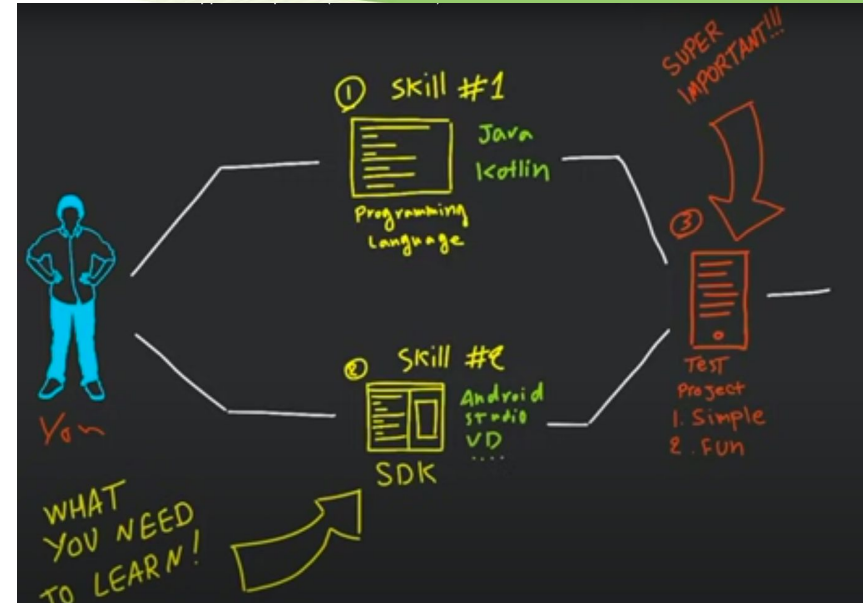


Développement Android



Comment apprendre le développement d'applications Android

- Apprendre une compétence en programmation (JAVA / KOTLIN)
 - XML et JSON
- Améliorez vos compétences dans le SDK Android
- Construisez vos propres projets (créez un portfolio)



Ce qu'il vaut mieux savoir en JAVA

- Fondamentaux du langage (contrôle de flux, etc.)
- Classes et objects
- Methods et data members
- Public, private, and protected
- Static et instance scope
- Exceptions
- Threads
- Collections
- Generics
- File I/O
- Reflection
- Interfaces



https://en.wikibooks.org/wiki/Java_Programming

Vous n'avez pas besoin d'un instructeur pour vous enseigner le SDK Android

developers  Platform Android Studio Google Play Jetpack Kotlin Docs Games

Android Basics in Kotlin

Welcome to Android Basics in Kotlin! In this course, you'll learn the basics of building Android apps with the Kotlin programming language. Along the way, you'll develop a collection of apps to start your journey as an Android developer.



<https://developer.android.com/courses/android-basics-kotlin/course>

Unit 1: Kotlin basics

Take your first steps programming in Kotlin, add images and text to your Android apps, and learn how to use classes, objects, and conditionals to create an interactive app for your users.

EARNABLE BADGES



[Explore](#)

Unit 2: Layouts

Build two different apps, and improve the user interface of your app by learning about layouts, Material Design guidelines, and best practices for UI development.

EARNABLE BADGES



[Explore](#)

Unit 3: Navigation

Enhance your users' ability to navigate across, into and back out from the various screens within your app for a consistent and predictable user experience.

EARNABLE BADGES



Unit 4: Connect to the internet

Write coroutines for complex code, and learn about HTTP and REST to get data from the internet. Then, use the Coil library to display images in your app.

EARNABLE BADGES



Vous n'avez pas besoin d'un instructeur pour vous enseigner le SDK Android



Choose a topic

Select all

- ☐ Ads
- ☐ AI & Machine Learning
- ☐ Analytics
- [More](#)

Choose a product

Select all

- ☐ AdMob
- ☐ Analytics
- ☒ Android
- [More](#)

Choose an event

Select all

- ☐ EuroPython
- ☐ Google I/O 2018
- ☐ Google I/O 2019
- [More](#)

Android

Create your first Tile in Wear OS

47 minutes Updated September 7, 2022

In this codelab, you'll learn how to create your own Tile for Wear OS. You'll use Tiles Material Components, a library that makes it easier to build UI that conforms to Material design guidelines, and preview your tiles in Android Studio as you develop.

Start

Practice: Kotlin Fundamentals

1 hour 40 minutes Updated September 1, 2022

Apply the basic concepts of the Kotlin programming language to solve the given problems.

Start

Preferences DataStore

Updated September 1, 2022

In this codelab, you learn how to use a data storage solution called Jetpack DataStore to store a key-value pair in your application.

Start

Create a birthday message in Kotlin

Updated August 31, 2022

In this codelab, you will create a birthday message program in the Kotlin language.

Start

Jetpack Compose basics

1 hour 5 minutes Updated August 30, 2022

In this codelab, you'll learn the basics of Compose.

Start

Download and install Android Studio

Updated August 26, 2022

In this codelab, you will download and install the Android Studio development tool.

Start

Create a Birthday Card app

Updated August 26, 2022

In this codelab, you will create an Android app that displays text, using the Layout Editor in Android Studio.

Start

Write your first program in Kotlin

Updated August 26, 2022

In this codelab, you will write your first program in the Kotlin language.

Start

Write your first program in Kotlin

1. Before you begin

2. Run your first program in Kotlin

3. Modify your program

4. Extend your program

5. Solution code

6. Summary

7. Learn more

8. Practice on your own

4. Extend your program

Print more than one message

Great job! You printed one line of text using the `println()` function. However, you can put as many lines of instructions inside a function as you want or need to get a task accomplished.

- Copy the line `println("Happy Birthday!")` and paste it two more times below it. Make sure your pasted lines are inside the curly braces of the `main` function.
- Change one text to be printed to someone's name, say "Jhansi".
- Change the other text to be printed to "You are 25".

Your code should look like the code below.

```
fun main() {  
    println("Happy Birthday!")  
    println("Jhansi")  
    println("You are 25!")  
}
```

What would you expect this code to do when it runs?

- Run your program to see what it does.
- Go to the output pane, and you should see 3 lines printed in the console window, as shown below.

Happy Birthday!
Jhansi
You are 25!

Nice work!

Dealing with errors

Making mistakes while programming is normal, and most tools will give you feedback to help you fix mistakes. In this step, create a mistake to see what happens.

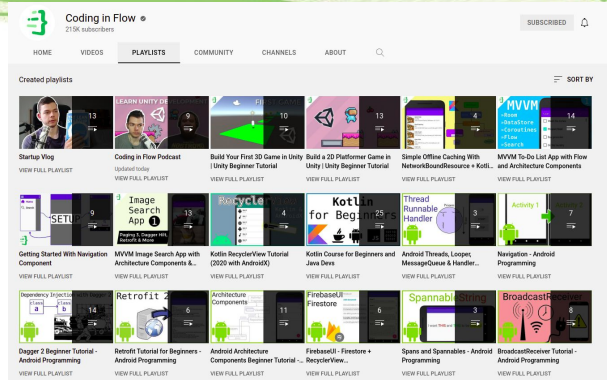
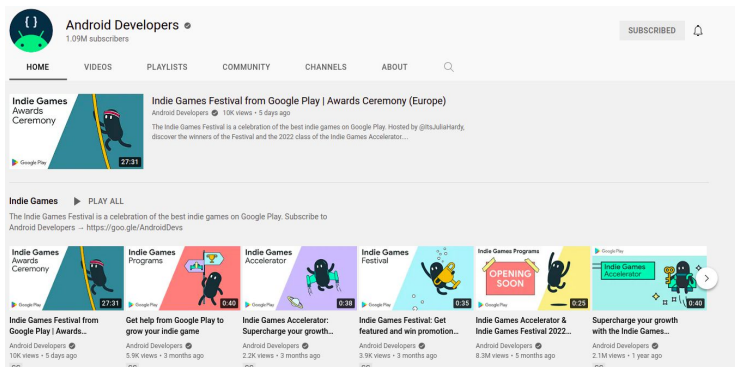
Back

Next

<https://codelabs.developers.google.com/?product=android>

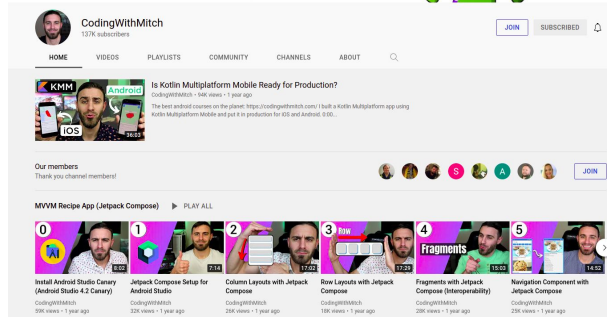


Vous n'avez pas besoin d'un instructeur pour vous enseigner le SDK Android

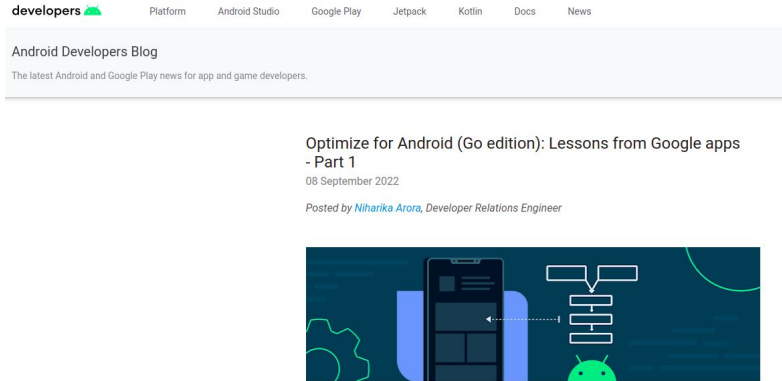


Become an Android developer

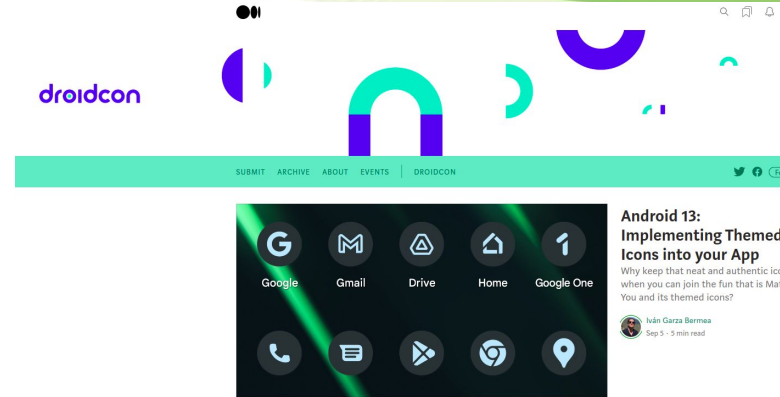
codingwithmitch.com



Vous n'avez pas besoin d'un instructeur pour vous enseigner le SDK Android

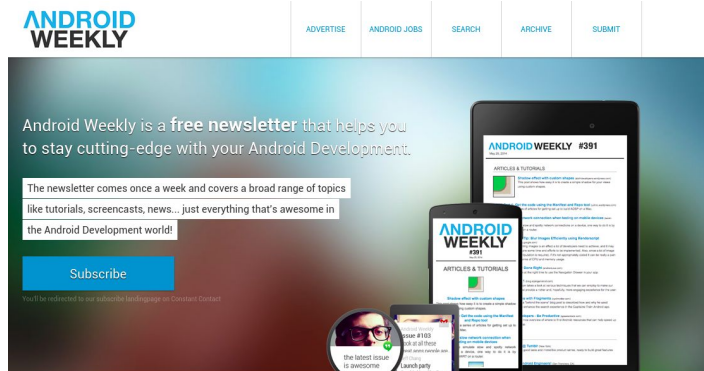


<https://android-developers.googleblog.com/>



<https://proandroiddev.com/>

Vous n'avez pas besoin d'un instructeur pour vous enseigner le SDK Android



<https://androidweekly.net/>

ANDROID DEVELOPMENT TUTORIALS - ANDROID, ACTIVITY, INTENT, ADT, SERVICES, BROADCASTRECEIVER

ANDROID DEVELOPMENT STARTER TUTORIALS

- Introduction to Android development with Android Studio
- Android Development with Android Studio
- Android Development with Kotlin

ANDROID FUNDAMENTAL TUTORIALS

- Android Intents
- Android application and activity life cycle
- The Android permissions model
- Using lists and grids in Android with RecyclerView
- Using Databinding in Android applications
- Using lists in Android (ListView)
- Android Logging

USING FRAGMENTS AND THE TOOLBAR

- Multi-pane development in Android with Fragments
- Using the Android Toolbar (ActionBar)
- Using Swipe-to-refresh in Android applications

ANDROID FILE AND NETWORK ACCESS AND ASYNCHRONOUS PROCESSING

- Android SQLite database and content provider
- Android XML Processing with the XmlPullParser
- Android Persistence with preferences and files
- Android Networking
- Using Retrofit for REST based backends
- Using RxJava in Android
- Android background processing with Handlers, AsyncTask and Loaders
- JSON in Android

<https://www.vogella.com/tutorials/android.html>

Quand dois-je me considérer comme un développeur Android ?

- Fondamentaux d'Android

- Comprendre l'architecture du système Android
- Être capable de décrire les éléments de base d'une application Android
- Savoir créer et exécuter une application Android
- Afficher des messages simples dans une popup à l'aide d'un Toast ou d'un Snackbar
- Pouvoir afficher un message en dehors de l'interface utilisateur de votre application à l'aide des notifications
- Comprendre comment localiser une application
- Être capable de planifier une tâche en arrière-plan à l'aide de WorkManager

- Créer une interface utilisateur

- Dispositions
- Menus
- Afficher des textes et des images à l'écran
- Animations

Quand dois-je me considérer comme un développeur Android ?

- Gestion de données

- Stocker localement les données
- Base de données SQLite
- Télécharger des données depuis Internet
- Requêtes HTTP
- Firebase

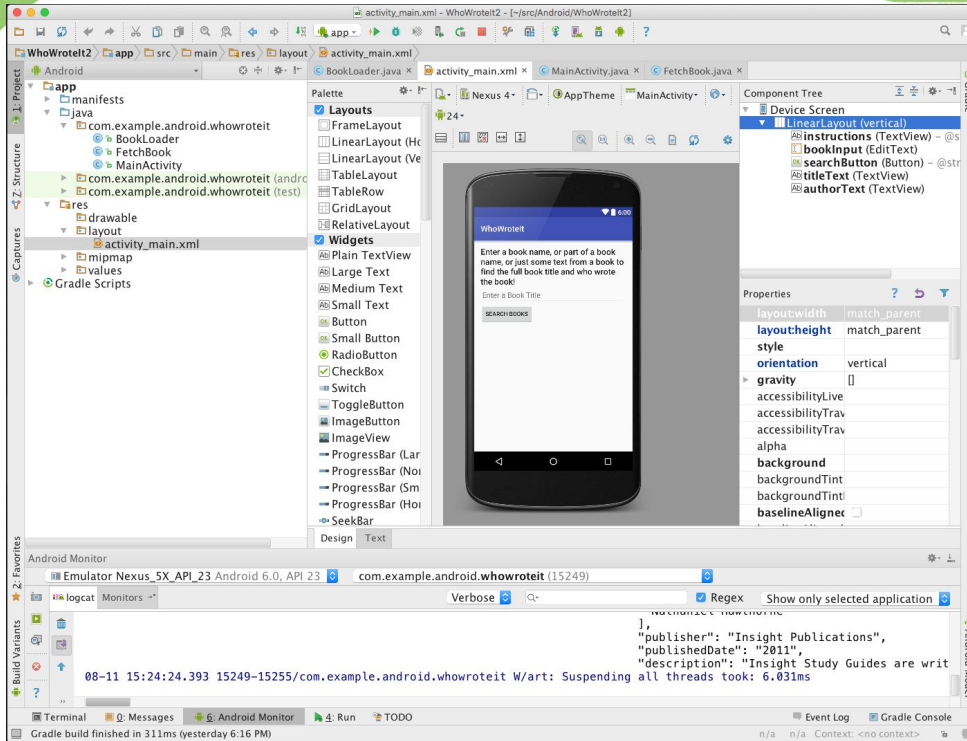
- Débogage

- Comprendre les techniques de débogage de base disponibles dans Android Studio
- Savoir comment déboguer et résoudre les problèmes liés au comportement fonctionnel et à l'ergonomie d'une application

- Testing

- Bien comprendre les principes fondamentaux des tests
- Être capable d'écrire des tests JUnit locaux utiles

Android Studio



- EDI Android officiel
- Développer, exécuter, déboguer, tester et emballer des applications
- Moniteurs et outils de performance
- Périphériques virtuels
- Vues du projet
- Éditeur de mise en page visuelle

Google Play Store

Publiez des applications via Google Play Store :

- Boutique d'applications officielle pour Android
- Service de distribution numérique exploité par Google



Versions Android

Google fournit des mises à niveau incrémentielles majeures du système d'exploitation Android en utilisant des noms sur le thème de la confiserie.



Cupcake
1.5



Donut
1.6



Eclair
2.0/2.1



Froyo
2.2



Gingerbread
2.3



Honeycomb
3.0/3.1



Ice Cream Sandwich
4.0



Jelly Bean
4.1/4.2/4.3



KitKat
4.4



Lollipop
5.0



Marshmallow
6.0



Nougat
7.0



Oreo
8.0

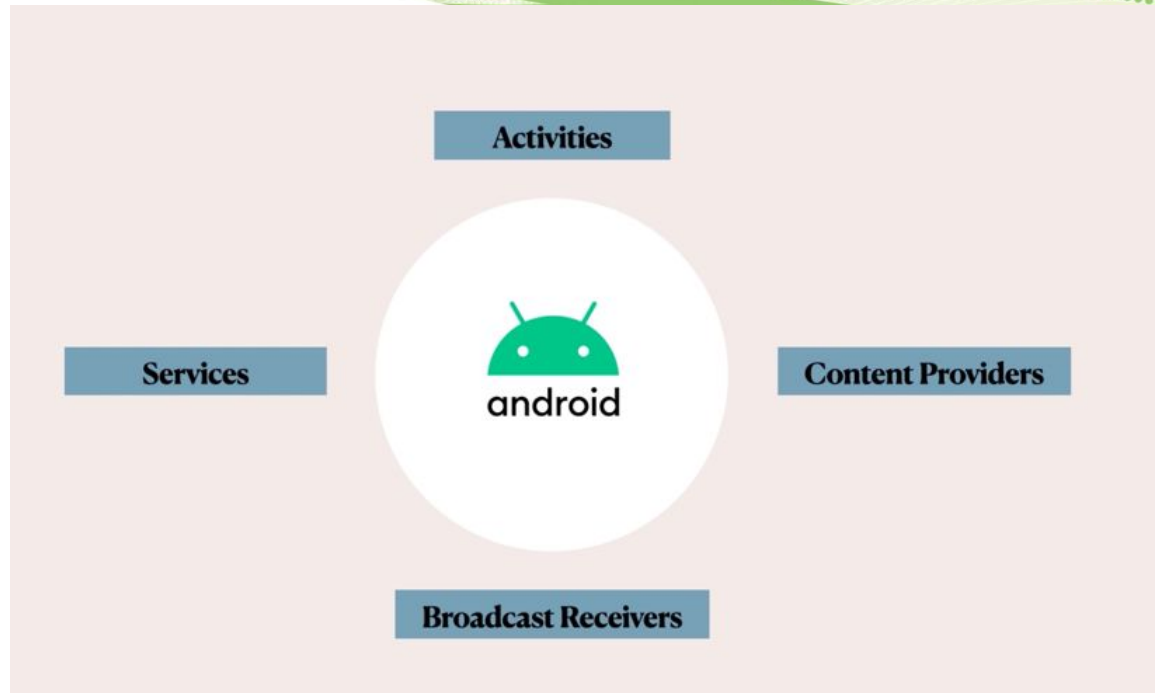


Pie
9.0



android

Composants/blocs de construction Android



Défis du développement Android

- Plusieurs tailles et résolutions d'écran
- Performance : rendez vos applications réactives et fluides
- Sécurité : protégez le code source et les données des utilisateurs
- Compatibilité : fonctionne bien sur les anciennes versions de plate-forme
- Marketing : comprendre le marché et vos utilisateurs

Installer/Configurer Android Studio

