

Formation Android



Présenter par Cédric aka “Endy Windaflame”
SAINT-GHISLAIN



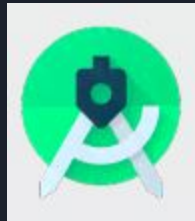
Présentation de Android

“Android est un système d'exploitation mobile open source fondé sur le noyau Linux et développé par un consortium d'entreprises, le Open Handset Alliance, sponsorisé par Google.

Depuis 2012, Android est le système d'exploitation mobile le plus utilisé dans le monde, devant iOS d'Apple.”

Extrait de Wikipédia

Installation de Android Studio



Récupérer l'IDE depuis cette URL : <https://developer.android.com/studio/index.html>



Configuration de l'env de dev

- Ajouter le JDK de Android Studio au PATH de la machine.
java et javac doivent fonctionner dans un invite de commande.
- Si utilisation d'un téléphone pour le cours :
 - Activer le mode développeur du téléphone.
 - Installer le plugin - Google USB Driver. (SDK Manager > SDL Tools)
- Si utilisation d'un émulateur pour le cours :
 - Utiliser Pixel_3a_API_34 dans la création du mobile.



Android - Création d'un projet

Pour une découverte, nous allons utiliser le projet qui se nomme “**Empty Views Activity**”

Utiliser les metadatas suivants, lors de la création du projet :

- Name : HelloWorld
- Package name : fr.solacroup.helloworld
- Language : Java
- Minimum SDK : API 24 (“Nougat”)
- Build configuration language : [Recommended]

DEMO

Android - Structure d'un projet

1/2

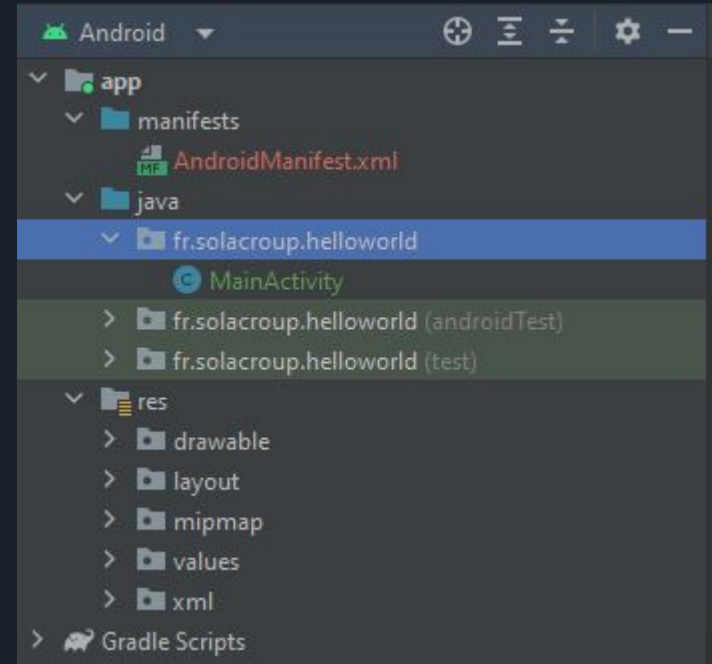
- Le répertoire "Manifests"

Ce repertoire est génération composé d'un seul fichier : **AndroidManifest.xml**. Ce fichier inclus les informations de l'application, les ressources et les activités.

- Le répertoire "Java"

Ce repertoire contient le programme lié au projet, ainsi que les fichiers tests.

DEMO



Android - Structure d'un projet

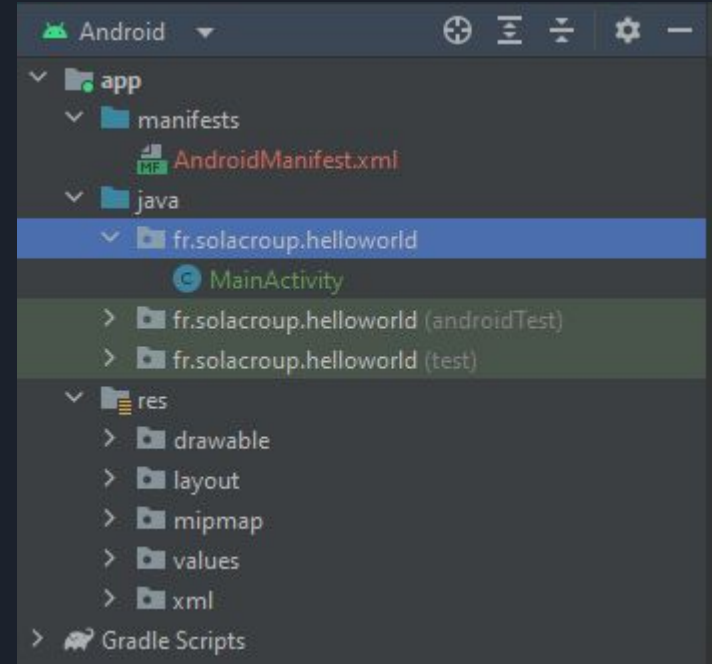
2/2

- Le répertoire Res

Ce répertoire contient toutes les ressources de l'application, et comprend :

- le dossier drawable , contient l'ensemble des cotenus à afficher à l'écran ;
- le dossier layout , contenant l'ensemble des fichiers qui décrivent le rendu visuel des différents écrans de l'application ;
- le dossier mipmap , qui contient principalement l'icône de lancement de l'application ;
- le dossier values , qui contient différents paramétrages et valeurs. Tel que le style et les textes ;
-

DEMO





Projet - Choix du thème

Pour la semaine, un thème pour le projet est à définir.

Une fois celui-ci défini, des ressources pour ce thème seront disponible sur le projet git, dans une branche spécifique à la session de cours.

PROJET PAR DÉFAUT : TODO List



Java - Programmation Orienté Objet

Sous forme de Travaux Pratique, les principes de la programmation objet en Java vont être abordés.

Lien utile : <https://quickref.me/java>

Android - Les fondamentaux

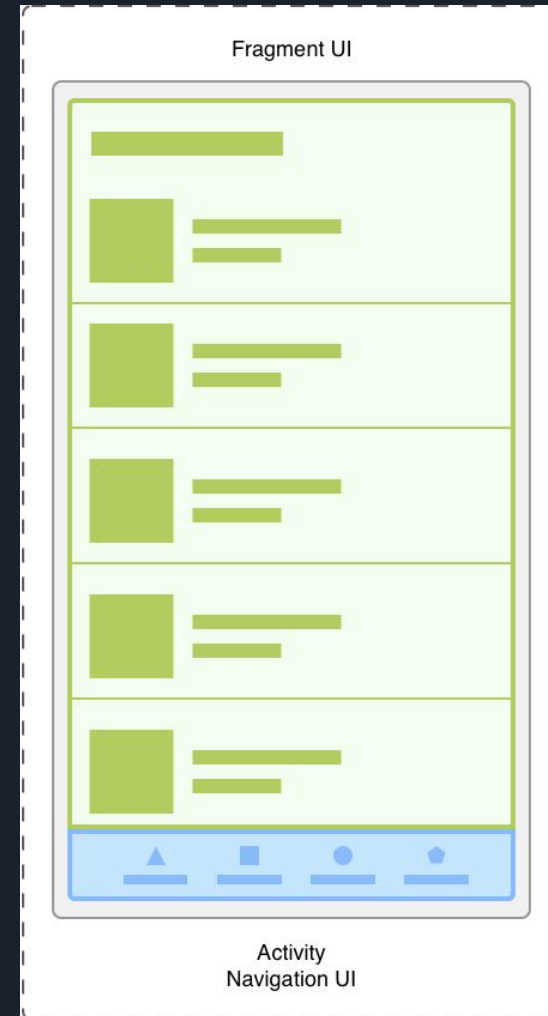
Une application Android est principalement définis par deux briques :

- **Activité**

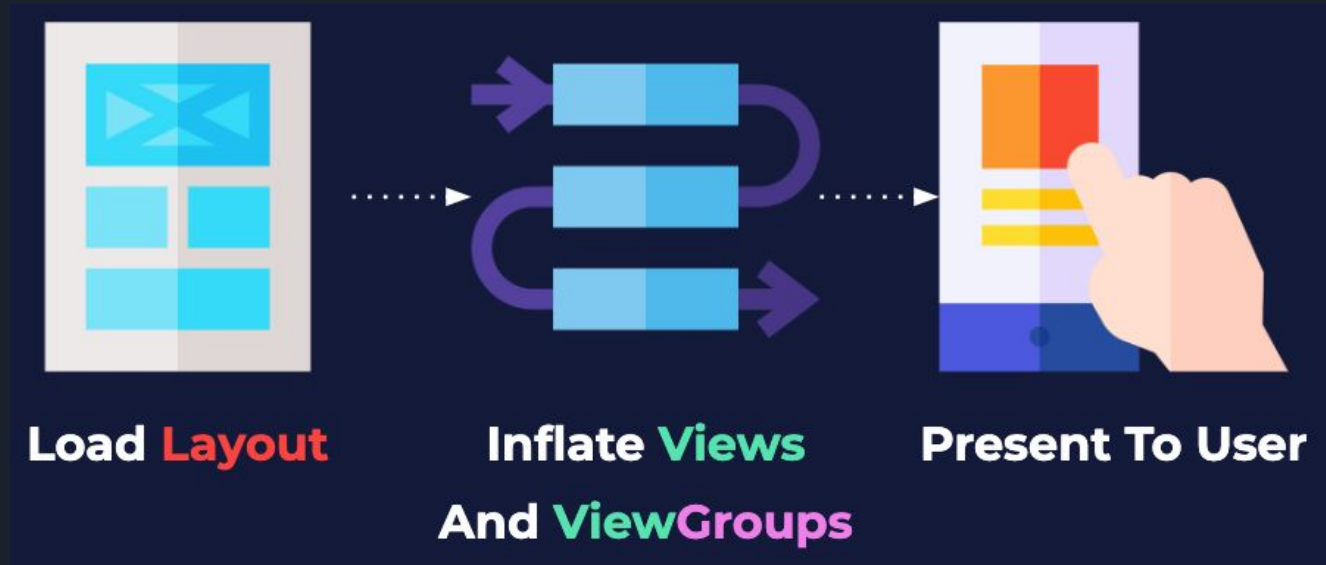
Une activité (ou Activity, en anglais) est le point d'entrée de n'importe quelle application Android. C'est elle qui va contenir l'ensemble des éléments graphiques du type texte, bouton, titre, etc. Elle intercepte aussi les interactions faites à l'écran par l'utilisateur.

- **Fragment**

Un fragment contient également des éléments visuels. Il correspond à une portion d'un écran, voire à un écran complet. Tout comme une activité, il est également capable d'intercepter les interactions de l'utilisateur. Un fragment ne peut exister qu'au sein d'une activité.

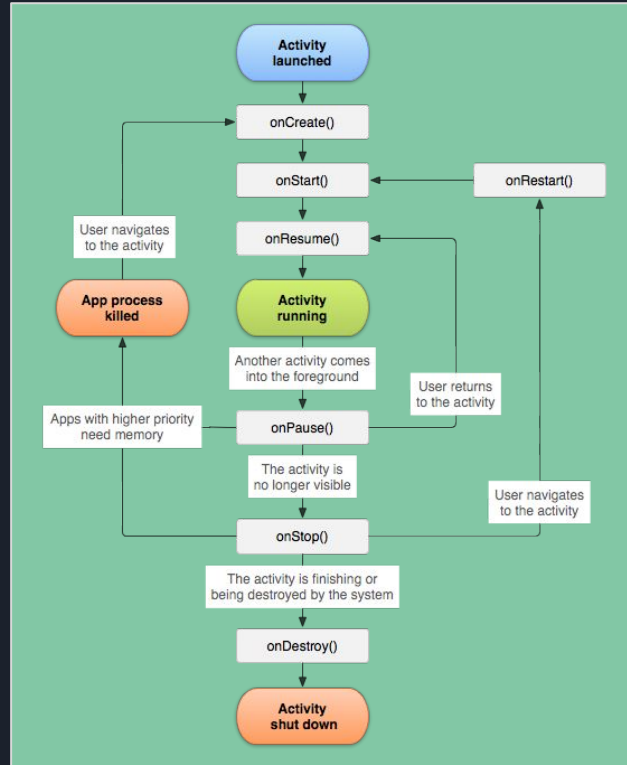


Android - Qu'est-ce qu'une Activity ?



<https://developer.android.com/reference/android/app/Activity>

Android - Cycle de vie d'une activité



<https://developer.android.com/reference/android/app/Activity#ActivityLifecycle>



Android - Qu'est-ce qu'un Layout ?

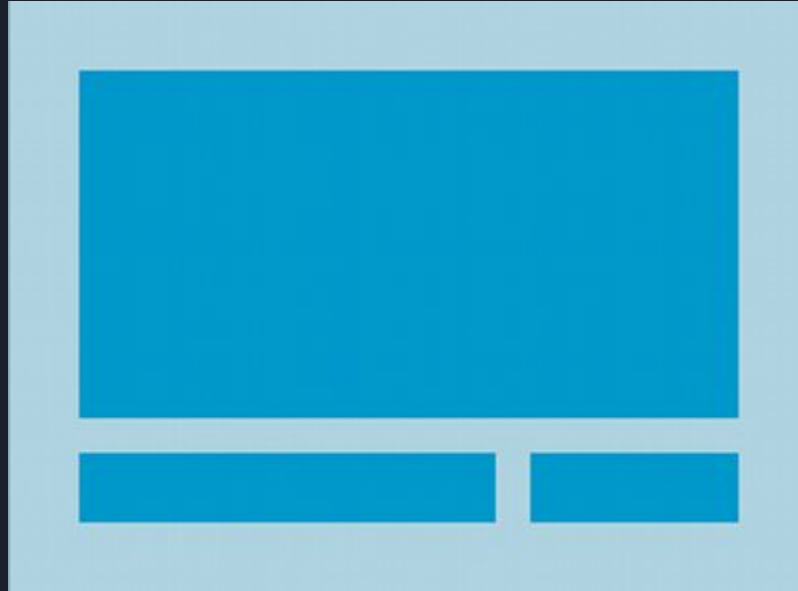
Un layout définit la structure d'une interface utilisateur dans votre application, par exemple dans une activité. Tous les éléments de la présentation sont construits à l'aide d'une hiérarchie d'objets View et ViewGroup.

Une vue dessine généralement un élément que l'utilisateur peut voir et avec lequel il peut interagir.

<https://developer.android.com/develop/ui/views/layout/declaring-layout>



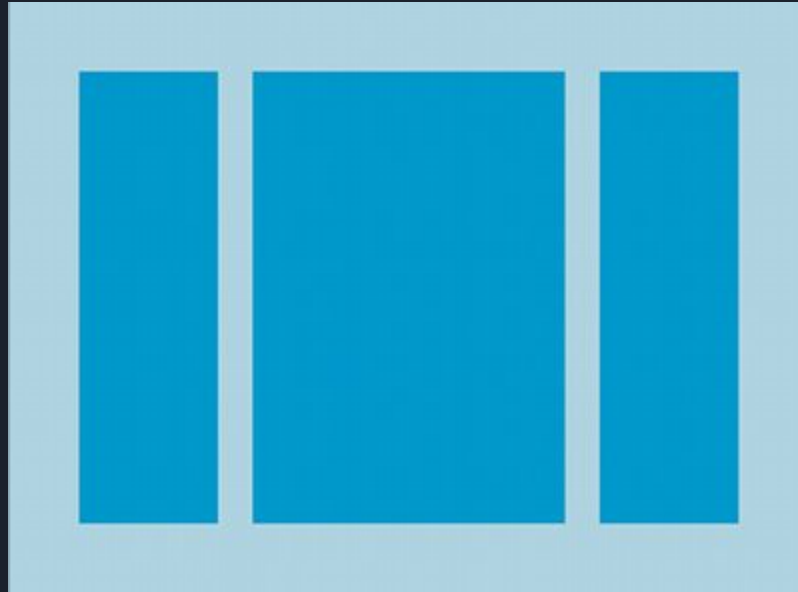
Android - Relative Layout



<https://developer.android.com/develop/ui/views/layout/relative>



Android - Linear Layout



<https://developer.android.com/develop/ui/views/layout/linear?hl=fr>



Android - Consulter les logs

Pour consulter les logs dans Android Studio,
Il faut accéder à l'outil Logcat.

Cet outil permet de visualiser tous les
messages imprimés par les différentes
applications présentes sur l'appareil.

```
Log.v(String tag, String message)
// Logs a verbose message.
Log.d(String tag, String message)
// Logs a debug message.
Log.i(String tag, String message)
// Logs a information message.
Log.w(String tag, String message)
// Logs a warning message.
Log.e(String tag, String message)
// Logs a error message.
```

<https://developer.android.com/studio/command-line/logcat?hl=fr>



Projet - Création du patron

1/2

```
package fr.solacroup.todolist ;
```

```
import androidx.appcompat.app.AppCompatActivity ;
```

```
import android.os.Bundle ;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState) ;
```

```
        setContentView(R.layout. activity_main) ;
```

```
    }
```

```
}
```



Projet - Création du patron

2/2

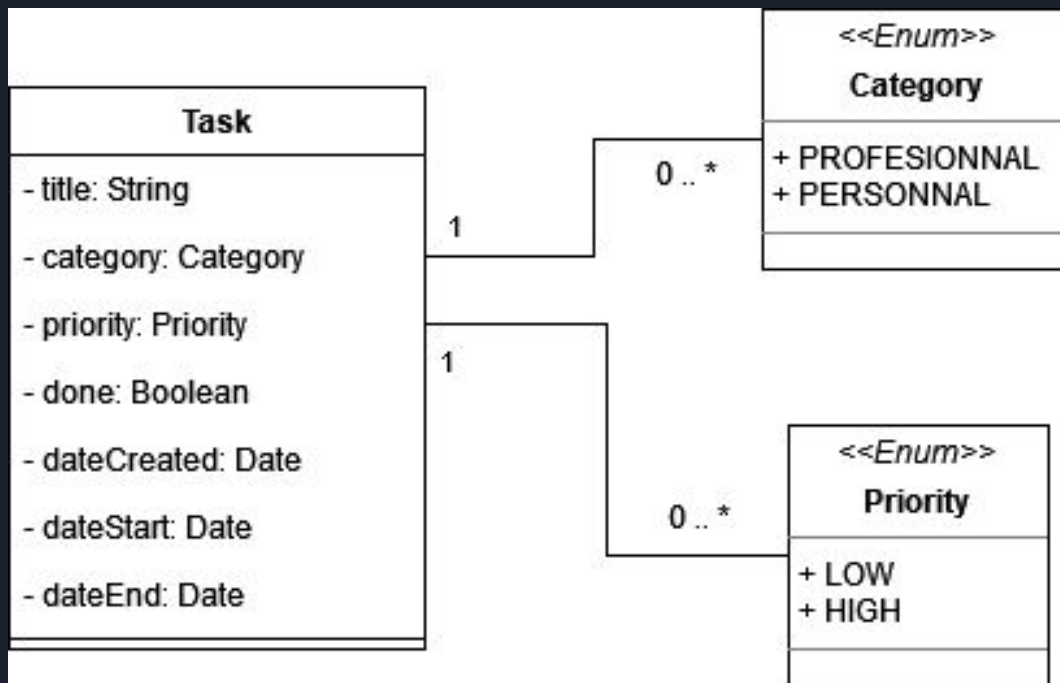
```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.aziflaj.todolist.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```

Projet - Création d'un modèle





Android - Transactions de fragment

- La transactions de fragment permet deux choses :
 - Naviguer entre différents Fragments au travers d'un `FragmentManager` et à l'aide du `FragmentManager`.
 - Partager des données à l'aide de `Bundle`.

<https://developer.android.com/guide/fragments/transactions?hl=fr>



Android - Création d'une pop-up

Il est possible d'utiliser une pop-up pour afficher une boîte de dialog au dessus de l'affichage courant. Plusieurs composants sont disponible pour effectuer cette action :

- Dialog : Une boîte de dialogue permettant d'afficher un message et d'effectuer une interaction avec l'utilisateur.
- Toasts : Sert à indiquer un message court, une indication ne nécessitant pas d'interaction utilisateur.

<https://developer.android.com/develop/ui/views/components/dialogs>



Android - Affichage d'une liste

Dans un projet Android, une liste représente un ensemble d'éléments s'affichants les uns à la suite des autres.

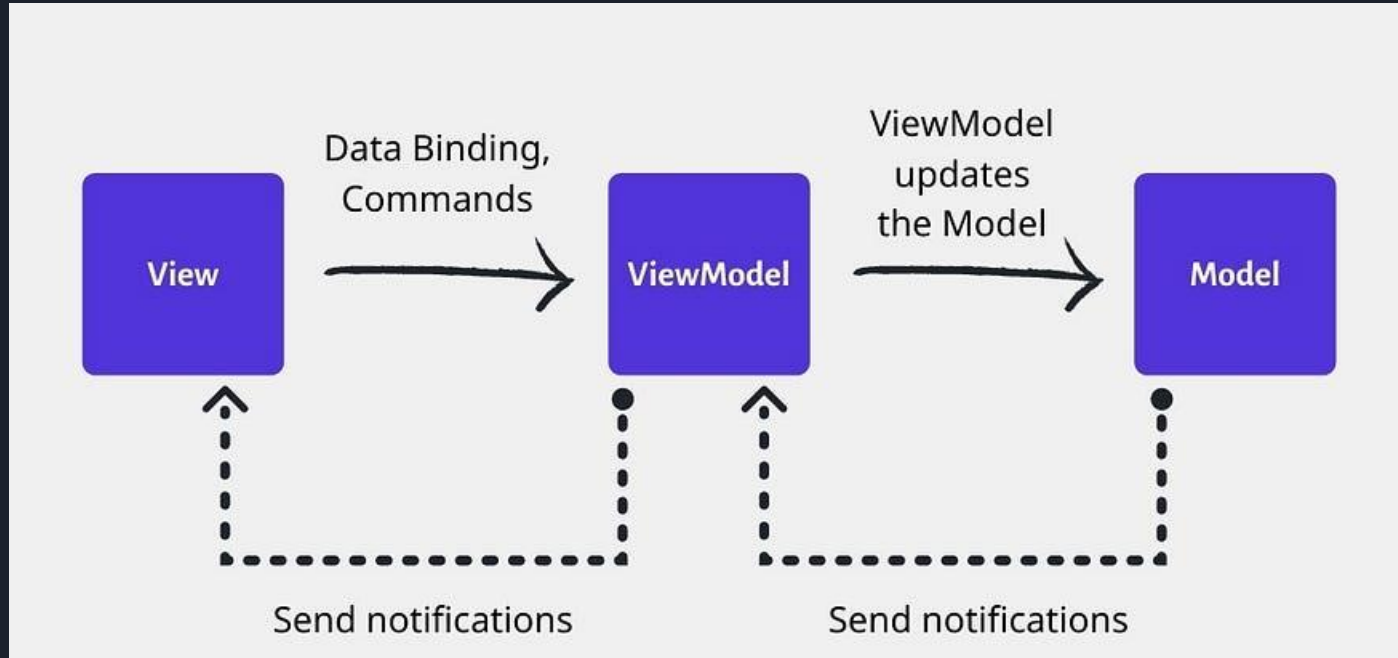
Pour insérer des données dans une liste, android propose d'utiliser les classes Adapter qui permet de lier des données à une liste :

- ArrayAdapter permet de remplir une liste à partir d'un tableau.
- SimpleCursorAdapter permet de remplir une liste à partir d'une base de donnée.

Pour aller plus loin sur cette utilisation, il est recommandé d'utiliser un RecyclerView.

<https://developer.android.com/reference/android/widget/ArrayAdapter>

Architecture Modèle-Vue-VueModel (MVVM)



<https://developer.android.com/codelabs/android-room-with-a-view>



Android - Stocker les données dans l'application

Pour le stockage des données de petites tailles au format d'une collection de clés/valeurs, vous pouvez utiliser les API SharedPreferences

<https://developer.android.com/training/data-storage/shared-preferences?hl=fr>

Pour le stockage de données persistantes avec des entités, vous pouvez utiliser Room.

<https://developer.android.com/training/data-storage/room/defining-data?hl=fr>



Android - Et plus encore

- JetPack
<https://developer.android.com/jetpack?hl=fr>
- Intent
<https://developer.android.com/guide/components/intents-filters>
- Construire un UI responsive avec ConstraintLayout.
<https://developer.android.com/develop/ui/views/layout/constraint-layout>
- Construire une liste dynamique avec RecyclerView.
<https://developer.android.com/develop/ui/views/layout/recyclerview>