# Création d'une application "Hello World"

## 1. Introduction à Android Studio

**Android Studio** est l'environnement de développement officiel (*IDE - Integrated Development Environment*) recommandé par Google pour créer des applications Android. Basé sur IntelliJ IDEA, il fournit tous les outils nécessaires pour concevoir, développer, tester, déboguer et publier des applications mobiles modernes.

Grâce à Android Studio, le développeur peut créer rapidement une interface graphique, écrire du code en Java ou Kotlin, exécuter l'application sur un émulateur ou un téléphone réel, analyser les performances et générer l'application finale (APK ou AAB) destinée au Google Play Store.

> **Remarque importante :** Avant de commencer le développement Android, il est indispensable d'installer Android Studio ainsi que le SDK Android. Il est également conseillé d'utiliser la version la plus récente afin de bénéficier des dernières fonctionnalités et des correctifs de sécurité.

## 2. Création d'un nouveau projet Android

La première étape consiste à créer un nouveau projet Android.

1. Ouvrir **Android Studio**.
2. Cliquer sur **New Project**.
3. Sélectionner le modèle **Empty Activity**.
4. Cliquer sur **Next**.
5. Renseigner les informations du projet :
   - **Name :** HelloWorld
   - **Package Name :** com.example.helloworld
   - **Language :** Java
   - **Minimum SDK :** API 21 ou supérieure
6. Cliquer sur **Finish**.

Après quelques instants, Android Studio génère automatiquement la structure complète du projet ainsi que tous les fichiers nécessaires au développement.

apcpedagogie.com

## 3. Structure d'un projet Android

Un projet Android est composé de plusieurs dossiers et fichiers ayant chacun un rôle précis.

- **java/ :** contient les classes Java qui définissent le comportement de l'application.
- **res/layout :** contient les fichiers XML décrivant les interfaces graphiques.
- **res/drawable :** contient les images, icônes et ressources graphiques.
- **res/values :** contient les chaînes de caractères, couleurs, dimensions et styles.
- **AndroidManifest.xml :** décrit l'application, ses activités, ses permissions et sa configuration générale.
- **Gradle Scripts :** permettent de gérer les dépendances et la compilation du projet.

> **À retenir :** Les fichiers Java contiennent la logique de l'application tandis que les fichiers XML définissent son apparence graphique. Cette séparation facilite la maintenance et l'évolution des applications Android.

## 4. Création de l'application "Hello World"

Lorsqu'un projet est créé avec le modèle **Empty Activity**, Android Studio génère automatiquement une activité principale appelée **MainActivity** ainsi qu'une interface graphique.

Le fichier Java correspondant est le suivant :

```
package com.example.helloworld;

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);
    }
}
```

**Explication :**

- `AppCompatActivity` représente une activité Android.

- `onCreate()` est appelée automatiquement lors du lancement de l'activité.
- `setContentView()` affiche l'interface graphique définie dans le fichier XML.

## 5. Interface graphique (activity_main.xml)

Le fichier XML définit les composants visibles à l'écran.

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

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World !"
        android:textSize="28sp"
        android:textStyle="bold"/>

</LinearLayout>
```

Dans cette interface :

- **LinearLayout** organise les composants verticalement.
- **TextView** affiche un texte à l'utilisateur.
- **layout_width** et **layout_height** définissent la taille du composant.
- **gravity="center"** centre le contenu au milieu de l'écran.
- **textSize** définit la taille du texte.

## 6. Exécution de l'application

Pour lancer l'application :

1. Créer un émulateur Android (AVD) ou connecter un smartphone avec le mode Développeur activé.
2. Cliquer sur le bouton **Run ▶**.
3. Sélectionner l'appareil cible.
4. Attendre la compilation du projet.

apcpedagogie.com

Une fois l'exécution terminée, l'écran affiche simplement le message :

```
Hello World !
```

Cette première application confirme que l'environnement Android Studio est correctement configuré et prêt pour le développement.

## 7. Analyse de l'application

Bien que très simple, cette application permet de comprendre plusieurs notions essentielles :

- La création d'un projet Android.
- La structure générale d'une application Android.
- Le rôle d'une activité (**Activity**).
- Le fonctionnement d'un fichier XML.
- L'affichage d'un composant graphique (**TextView**).
- La compilation et l'exécution sur un appareil Android.

> **Conclusion :** L'application *Hello World* constitue la première étape dans l'apprentissage du développement Android. Elle permet de découvrir l'environnement Android Studio, la structure d'un projet, le cycle de vie d'une activité et les bases de la création d'interfaces graphiques avant d'aborder des applications plus complexes.

*Rédigé et structuré par :* **Med Khalil Kribi**

Source : https://apcpedagogie.com/creation-application-hello-world-android · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com