TP1 : Prise en main

# Objectifs

* Installer l'environnement de développement Android Studio.
* Créer un émulateur (périphérique virtuel) avec une configuration bien déterminée.
* Créer et exécuter l'application Hello World sur les périphériques virtuels et physiques.
* Explorer la structure du projet.

# Configuration requise

* 8 Go de mémoire RAM.
* 8 Go d'espace disque pour tout installer.
* 1280 x 800 comme résolution minimum d'écran
* SE : Microsoft® Windows® 7/8/10 , 64-bits

Mac® OS X® 10.14 ou plus récent

GNOME ou KDE desktop testé dans gLinux basé en Debian.

Chrome OS

## Tache 1 : Installer Android Studio

Android Studio fournit un environnement de développement intégré complet comprenant un éditeur de code avancé et un ensemble de modèles d'application. En outre, il contient des outils de développement, de débogage, de test et de performance qui facilitent le développement d'applications. Vous pouvez tester vos applications avec une large gamme d'émulateurs préconfigurés ou sur votre propre appareil mobile, créez des applications de production et les publier sur le magasin Google Play.

Un SDK, un kit de développement, est un ensemble d'outils permettant de développer pour une cible particulière. Par exemple pour développer pour une console de jeu vidéo, on utilise un SDK spécifique pour développer des applications pour cette console. Le SDK Android est donc un ensemble d'outils que met à disposition Google afin de vous permettre de développer des applications pour Android. Il contient : les librairies Java pour créer des logiciels, les outils de mise en boîte des logiciels, AVD : un émulateur de tablettes pour tester les applications, ADB : un outil de communication avec les vraies tablettes.

1. Accédez au [site des développeurs Android](https://developer.android.com/sdk/index.html) et suivez les instructions pour télécharger et [installer Android Studio](https://developer.android.com/studio/install.html).
2. Acceptez les configurations par défaut pour toutes les étapes et assurez-vous que tous les composants sont sélectionnés pour l'installation.
3. Une fois l'installation terminée, l'assistant d'installation télécharge et installe des composants supplémentaires, notamment le SDK Android.
4. Une fois le téléchargement terminé, Android Studio démarre et vous êtes prêt à créer votre premier projet.

## Tâche 2: Créer l'application Hello World

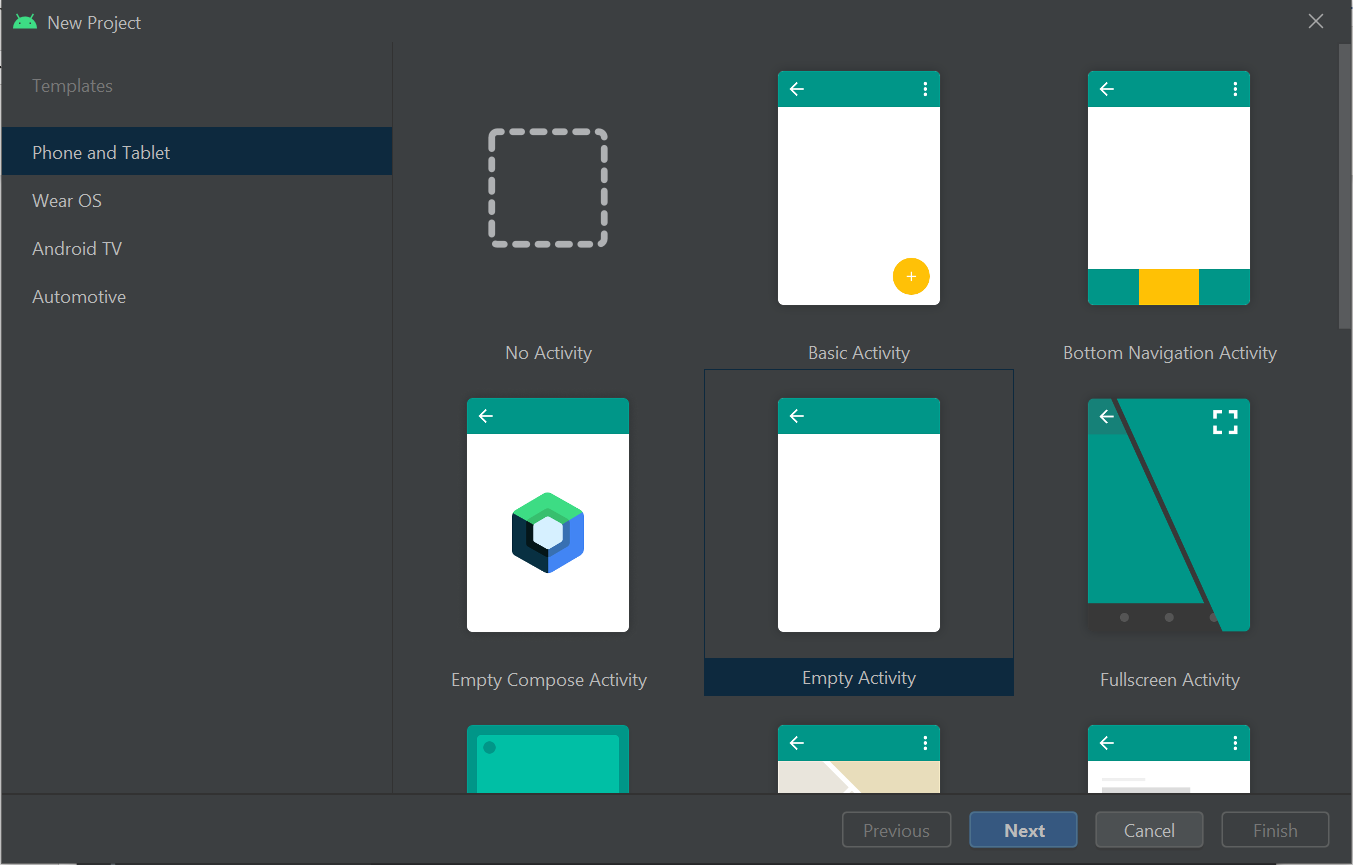
Dans cette tâche, vous allez créer une application qui affiche "Hello World" pour vérifier que le Android Studio est correctement installé et pour apprendre les bases du développement avec Android.

### Créer le projet d'application

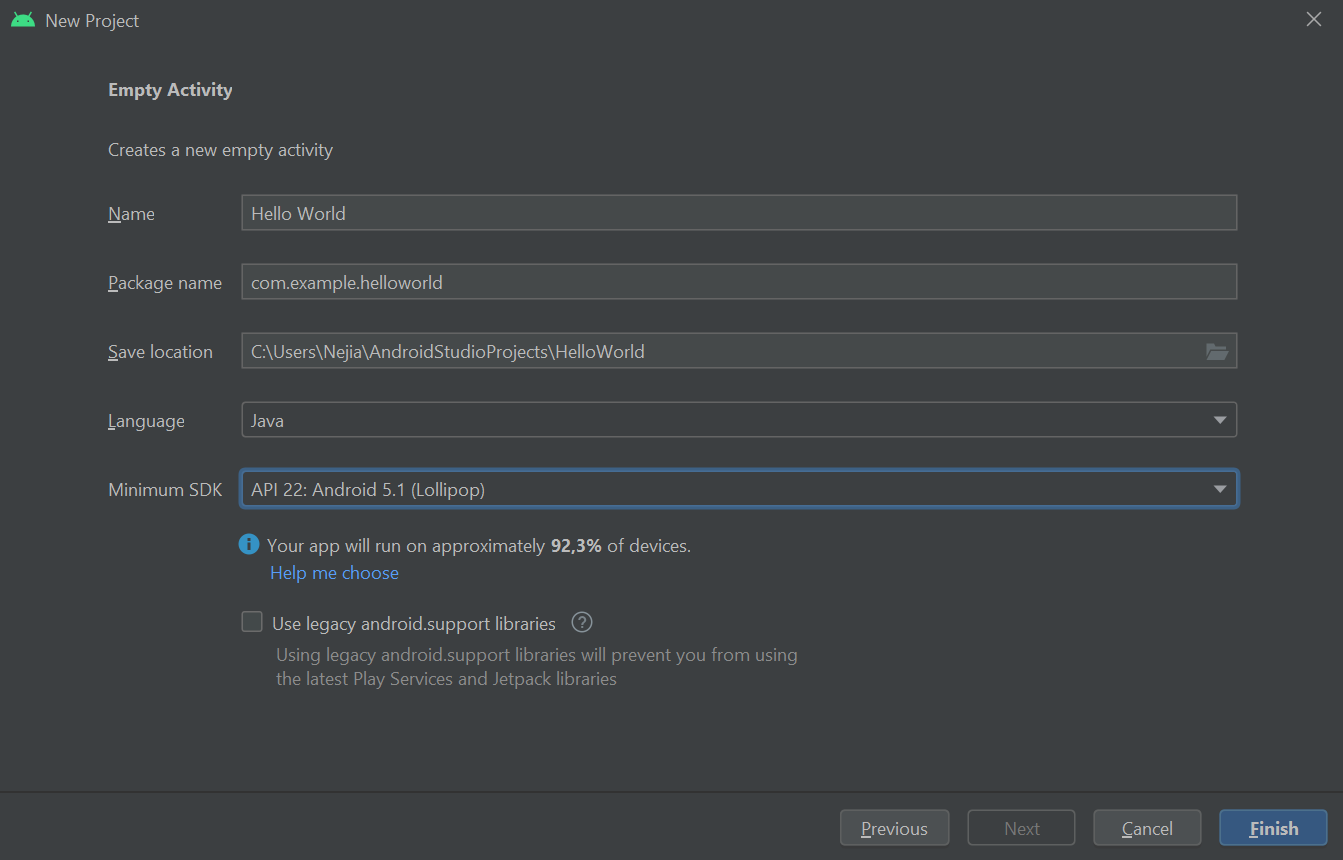
* 1. Dans la fenêtre principale **Welcome to Android Studio**, cliquez sur **Start a new Android Studio** project.
  2. La fenêtre **New Project** apparaît.

Une [Activity](https://developer.android.com/reference/android/app/Activity.html) est une tâche unique et ciblée que l'utilisateur peut faire. C'est un élément crucial de toute application Android. Une **Activity** est généralement associée à une présentation qui définit la manière dont les éléments de l'interface utilisateur apparaissent sur un écran.

Android Studio fournit des modèles d'Activity pour vous aider à démarrer. Pour le projet Hello World, choisissez **Empty Activity** comme indiqué ci-dessous, puis cliquez sur **Next**.



* 1. Dans la fenêtre **New Project,** entrez **Hello World** pour le nom de l'application.

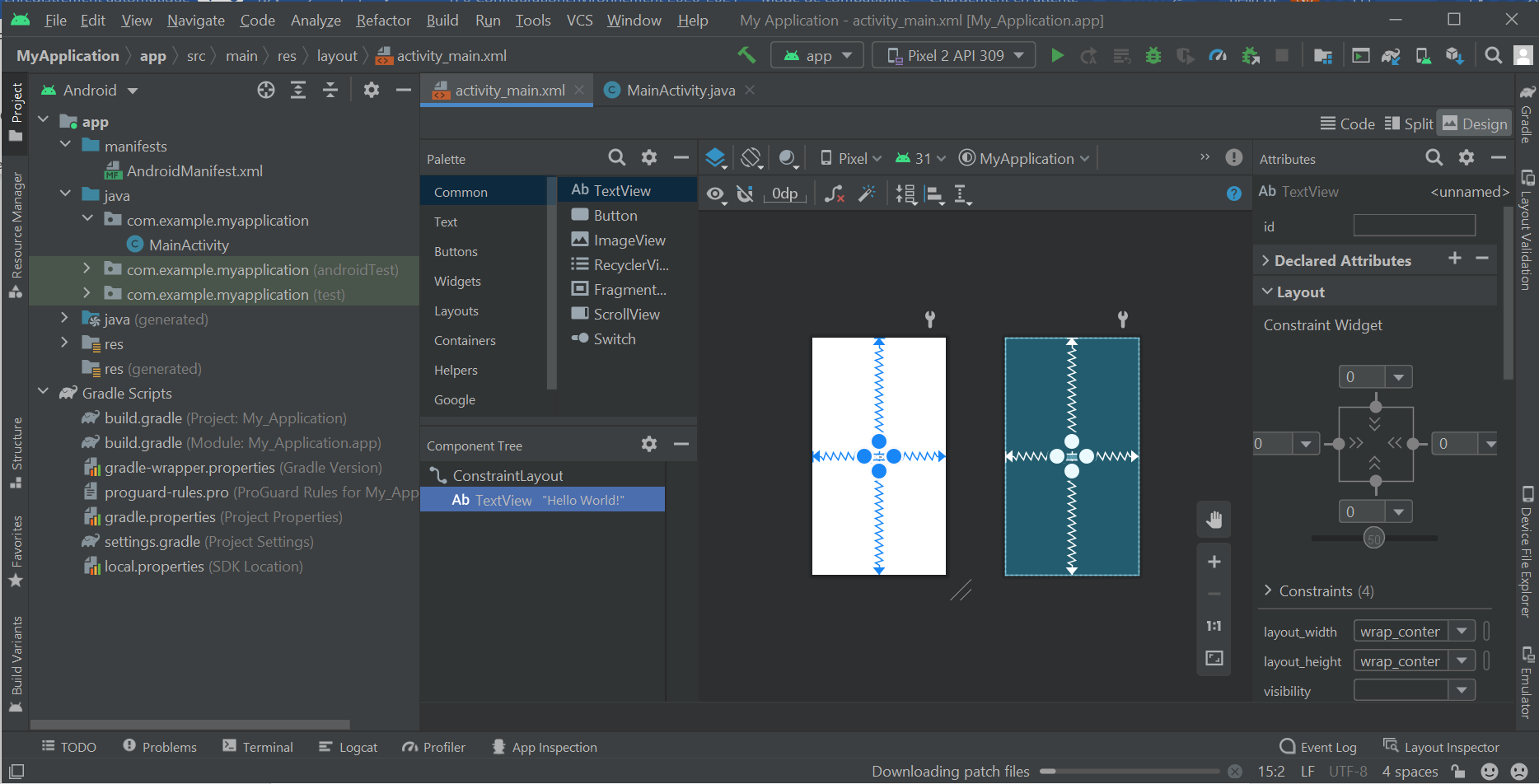


* 1. Acceptez la valeur **com.example.helloworld** par défaut pour **Package name** ou créez un domaine de société unique.
  2. Vérifiez que **Save location**par défaut est l'emplacement où vous souhaitez stocker votre application Hello World et d'autres projets Android Studio.
  3. Choisissez **Java** comme langage de programmation (ou Kotlin).
  4. Par défaut l'API22: Android 5.1 Lollipop est défini comme SDK minimal; Si ce n'est pas le cas, utilisez le menu contextuel pour le définir. Votre code sera compatible avec 92.3% des appareils Android actifs sur le Google Play Store. Cliquez sur **Finish**.

Android Studio crée un dossier pour vos projets et le construit avec [Gradle](https://gradle.org/" \t "_blank) (cela peut prendre quelques instants).

L'éditeur Android Studio apparaît. Suivez ces étapes :

1. Cliquez sur l'onglet **activity\_main.xml** pour afficher l'éditeur de "layout"..



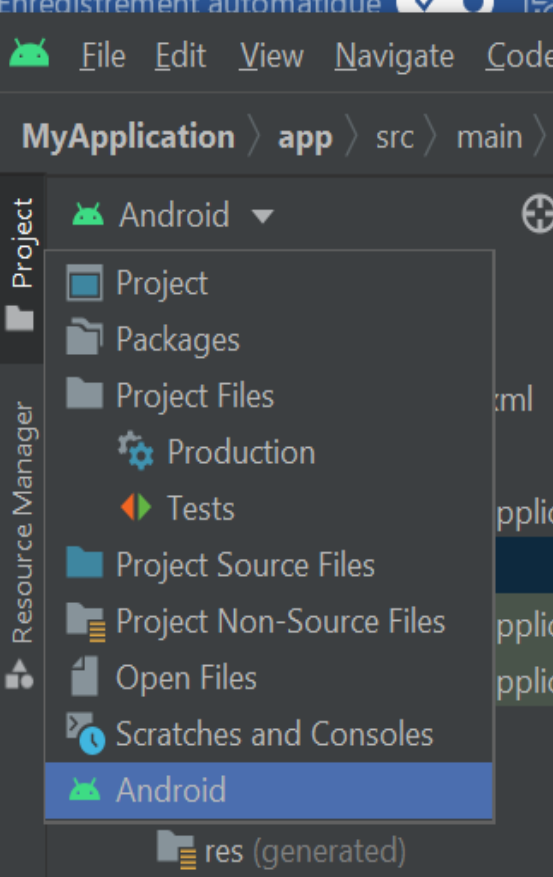
1. Cliquez sur l'onglet **MainActivity.java** pour afficher l'éditeur de code, comme indiqué ci-dessous.

### 

### Explorer le volet Projet> Android

Dans cette pratique, vous explorerez comment le projet est organisé dans Android Studio.

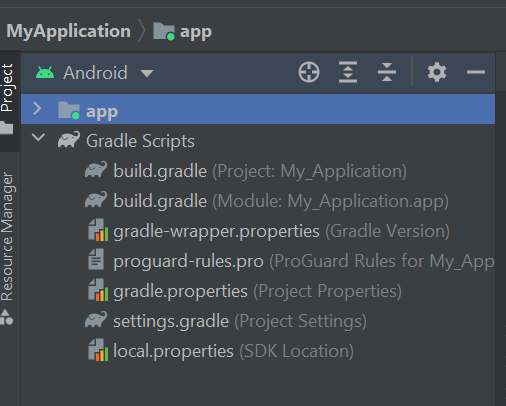
* 1. Si ce n'est déjà fait, cliquez sur l'onglet **Project**dans la colonne d'onglets verticale du côté gauche de la fenêtre d'Android Studio. La fenêtre **Project** apparaît.
  2. Pour afficher le projet dans la hiérarchie de projets Android standard, choisissez **Android** dans le menu contextuel en haut de la sous-fenêtre **Project**, comme indiqué ci-dessous.



### Explorer le dossier Gradle Scripts

Le système de génération **Gradle** dans Android Studio facilite l'inclusion de binaires externes ou d'autres modules de bibliothèque dans votre génération en tant que dépendances.

Lorsque vous créez un projet pour la première fois, le volet **Projet> Android**apparaît avec le dossier **Gradle Scripts** développé comme indiqué ci-dessous.



Suivez ces étapes pour explorer le système Gradle:

* 1. Si le dossier **Gradle Scripts** n'est pas développé, cliquez sur le triangle pour le développer.

Ce dossier contient tous les fichiers nécessaires au système de construction.

* 1. Recherchez le fichier **build.gradle (Project: HelloWorld)**.

Vous y trouverez les options de configuration communes à tous les modules composant votre projet. Chaque projet Android Studio contient un seul fichier de construction Gradle de niveau supérieur. La plupart du temps, vous n'avez pas besoin de modifier ce fichier, mais il est toujours utile de comprendre son contenu.

Par défaut, le fichier de construction de niveau supérieur utilise le bloc buildscript pour définir les référentiels Gradle et les dépendances communes à tous les modules du projet. Lorsque votre dépendance est autre chose qu'une bibliothèque locale ou une arborescence de fichiers, Gradle recherche les fichiers dans les référentiels en ligne spécifiés dans le bloc référentiels de ce fichier.

Par défaut, les nouveaux projets Android Studio déclarent que MavenCentral et Google sont les emplacements du référentiel:

buildscript **{** repositories **{** google()  
 mavenCentral()  
 **}**

* 1. Rechercher le fichier **build.gradle (Module: app)**.

En plus du fichier build.gradle au niveau du projet, chaque module possède son propre fichier build.gradle, qui vous permet de configurer les paramètres de construction de chaque module spécifique (l'application HelloWorld ne comporte qu'un seul module).

Ce fichier est le plus souvent le fichier à modifier lors de la modification de configurations au niveau de l'application, telles que la déclaration de dépendances dans la section dependencies. Vous pouvez déclarer une dépendance de bibliothèque à l'aide de l'une des différentes configurations de dépendance. Chaque configuration de dépendance fournit à Gradle différentes instructions sur l'utilisation de la bibliothèque.

Voici le fichier **build.gradle (Module: app)**de l'application HelloWorld:

plugins **{** id 'com.android.application'  
**}**

android **{** compileSdk 31  
  
 defaultConfig **{** applicationId "com.example.helloworld"  
 minSdk 22  
 targetSdk 31  
 versionCode 1  
 versionName "1.0"  
  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 **}**

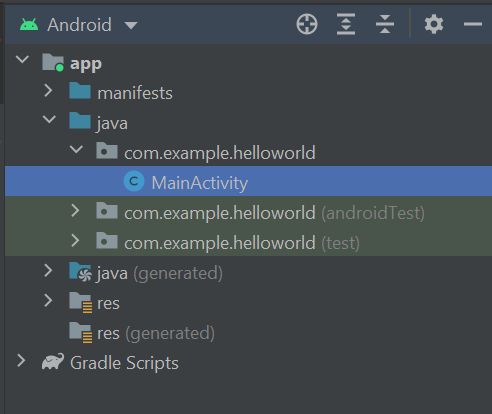
buildTypes **{** release **{** minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 **}  
 }** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}  
}**

dependencies **{** implementation 'androidx.appcompat:appcompat:1.2.0'  
 implementation 'com.google.android.material:material:1.3.0'  
 implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
 testImplementation 'junit:junit:4.+'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
**}**

### Explorer les dossiers "app" et "res"

Tout le code et toutes les ressources de l'application se trouvent dans les dossiers app et res.

* 1. Développez le dossier **app**, le dossier **java** et le dossier  **com.example.android.helloworld** pour afficher le fichier java **MainActivity**.



Le nom de la première Activity (écran) visible par l'utilisateur, qui initialise également les ressources de l'application, est généralement appelé **MainActivity**.

* 1. Développez les dossiers **res** et **layout**, puis double-cliquez sur le fichier  **activity\_main.xml**pour l'ouvrir dans l'éditeur de "layout".

Une image contenant texte, capture d’écran, moniteur, noir

Description générée automatiquement

Le dossier **res** contient des ressources, telles que des dispositions, des chaînes et des images. Une activité est généralement associée à une disposition de vues d'interface utilisateur définies sous la forme d'un fichier XML. Ce fichier porte généralement le nom de son activité.

### Explorer le dossier "manifestes"

Le dossier manifests contient des fichiers qui fournissent des informations essentielles sur votre application au système Android, que le système doit posséder avant de pouvoir exécuter le code de l'application.

* 1. Développez le dossier **manifests**.
  2. Ouvrez le fichier **AndroidManifest.xml**.

Le fichier AndroidManifest.xml décrit tous les composants de votre application Android. Tous les composants d'une application, tels que chaque activité, doivent être déclarés dans ce fichier XML. Dans d'autres leçons, vous modifierez ce fichier pour ajouter des fonctionnalités et des autorisations de fonctionnalités.

## Tâche 3: Utiliser un périphérique virtuel (émulateur)

Dans cette tâche, vous utiliserez le Android Virtual Device (AVD) manager pour créer un périphérique virtuel (également appelé émulateur) simulant la configuration d'un type particulier de périphérique Android, puis utiliser ce périphérique virtuel pour exécuter l'application.

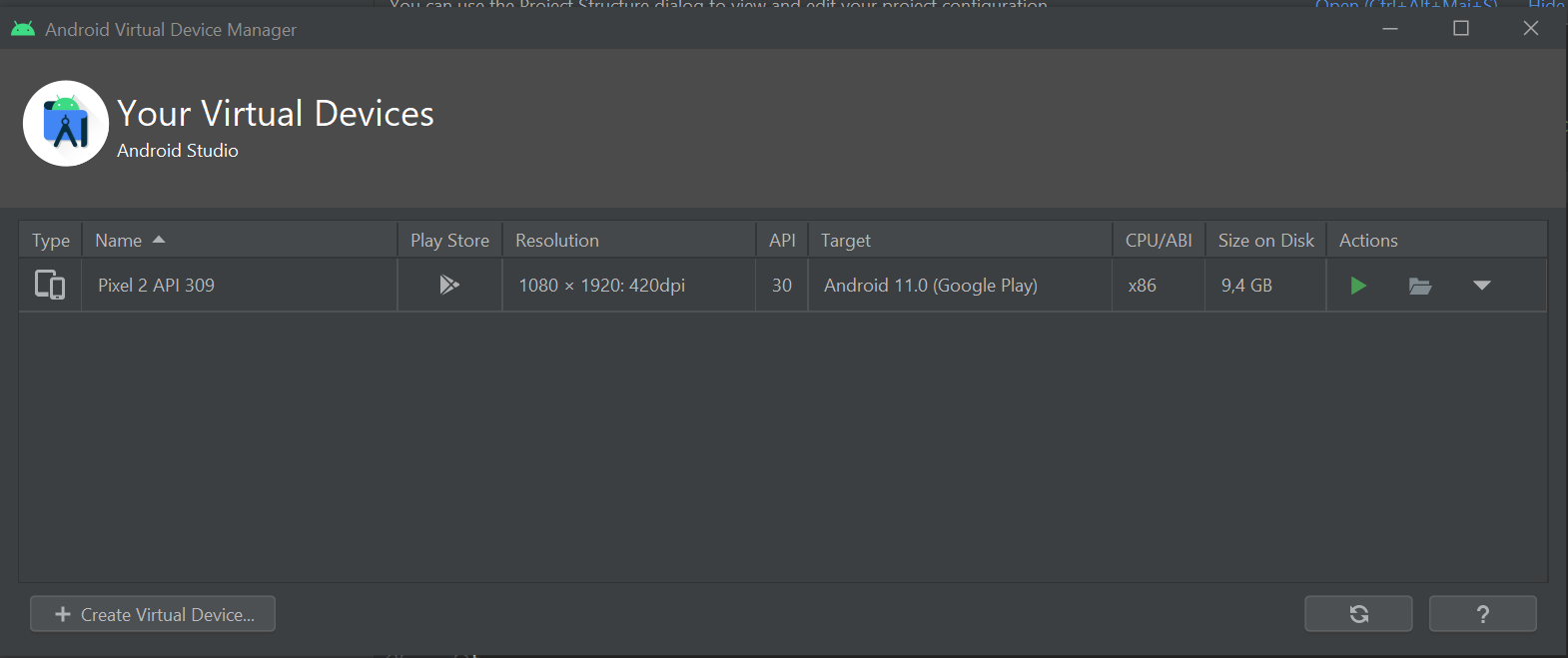
À l'aide du AVD Manager, vous définissez les caractéristiques matérielles d'un périphérique, son niveau d'API, son stockage, son apparence et d'autres propriétés, puis enregistrez-le en tant que périphérique virtuel.

### Créer un périphérique virtuel Android (AVD)

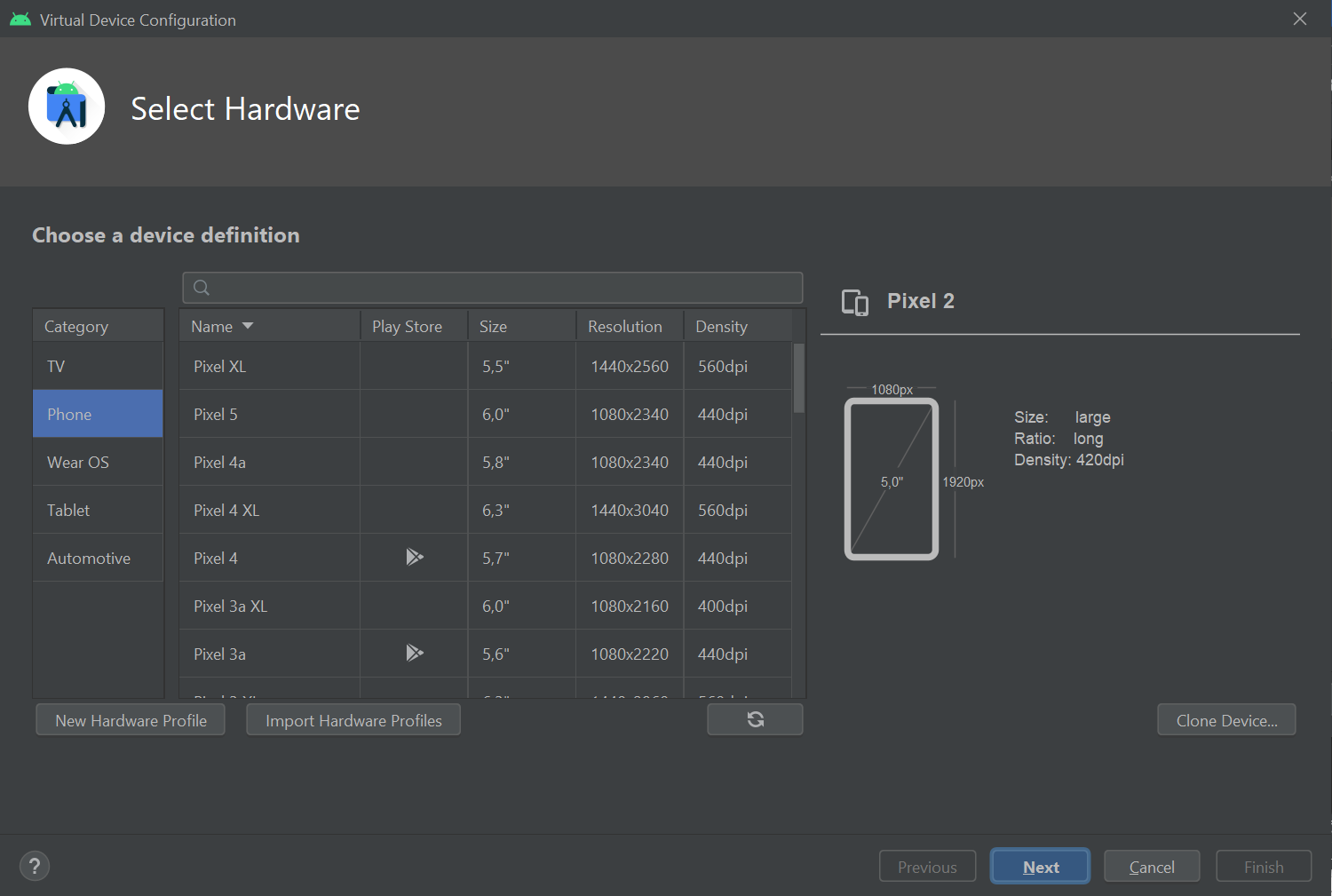
Pour exécuter un émulateur sur votre ordinateur, vous devez créer une configuration décrivant le périphérique virtuel.

* 1. Dans Android Studio, sélectionnez **Tools > Android > AVD Manager** ou cliquez sur l'icône

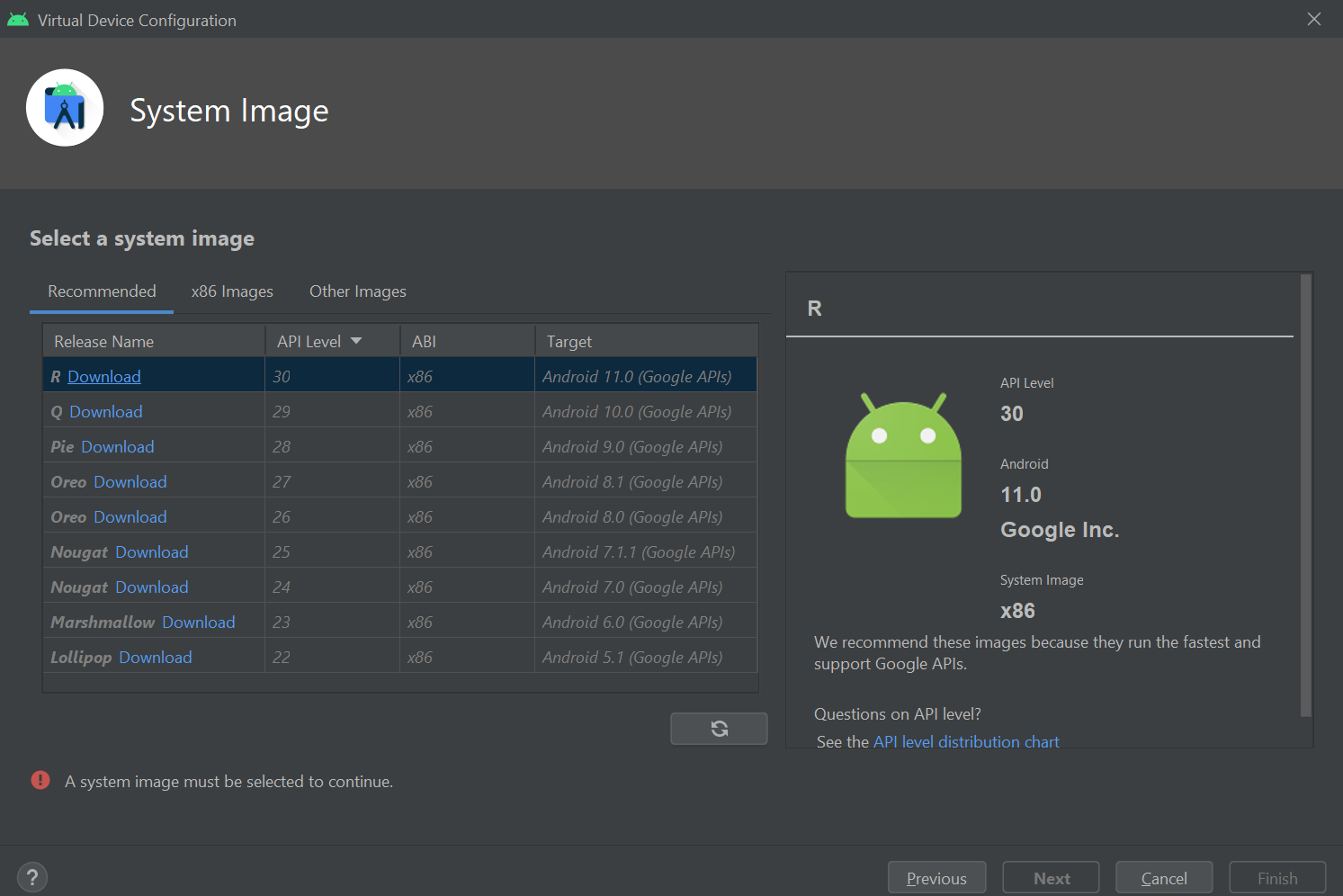
L'écran **Your Virtual Devices** apparaît. Si vous avez déjà créé des périphériques virtuels, l'écran les affiche.



* 1. Cliquez sur le bouton **+Create Virtual Device**. La fenêtre **Select Hardware** apparaît avec une liste de périphériques matériels pré-configurés. Pour chaque périphérique, le tableau fournit une colonne pour sa taille d'affichage diagonale (**Size**), sa résolution d'écran en pixels (**Resolution**) et sa densité de pixels (**Density**).



* 1. Choisissez un appareil tel que **Pixel 5** ou **Pixel XL**, puis cliquez sur **Next**. L'écran Image système apparaît.
  2. Choisissez la version du système Android à exécuter sur le périphérique virtuel (telle que **Oreo**).

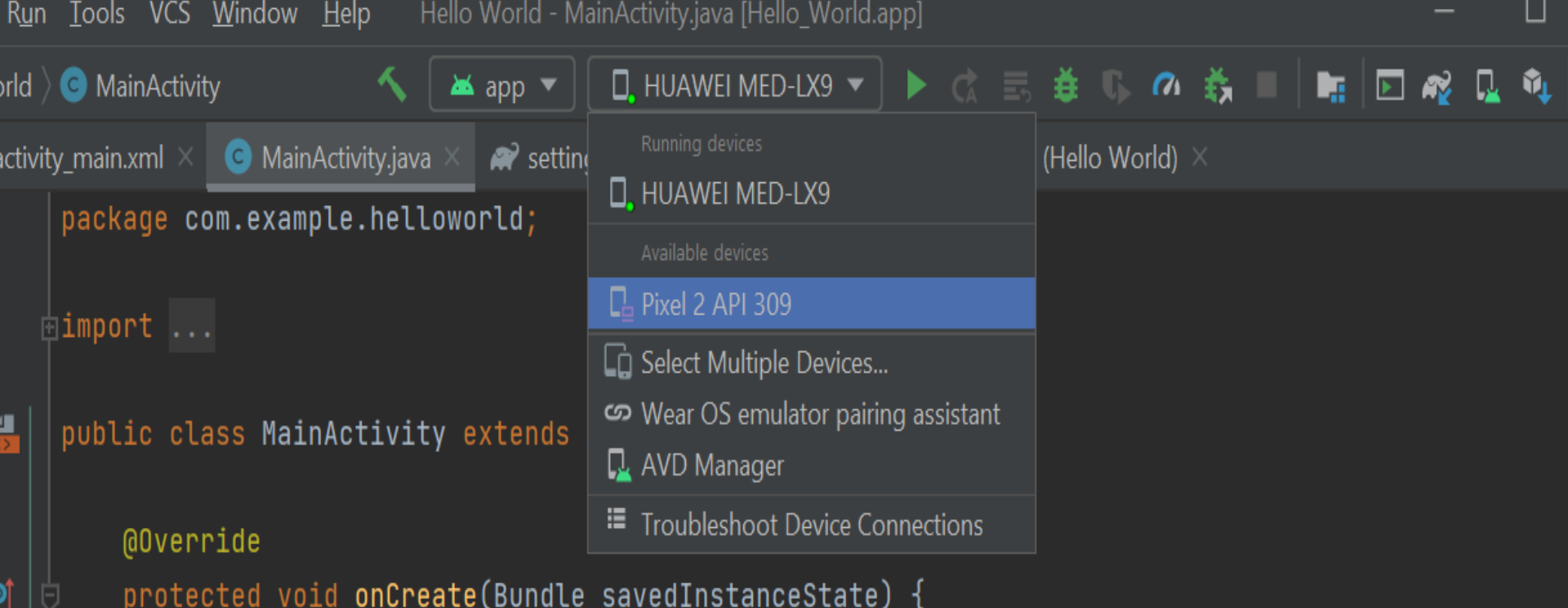


Si un lien **Download** est visible à côté de l'image système que vous souhaitez utiliser, celle-ci n'est pas encore installée. Cliquez sur le lien pour lancer le téléchargement, puis cliquez sur **Finish** lorsque vous avez terminé.

### Exécuter l'application sur le périphérique virtuel

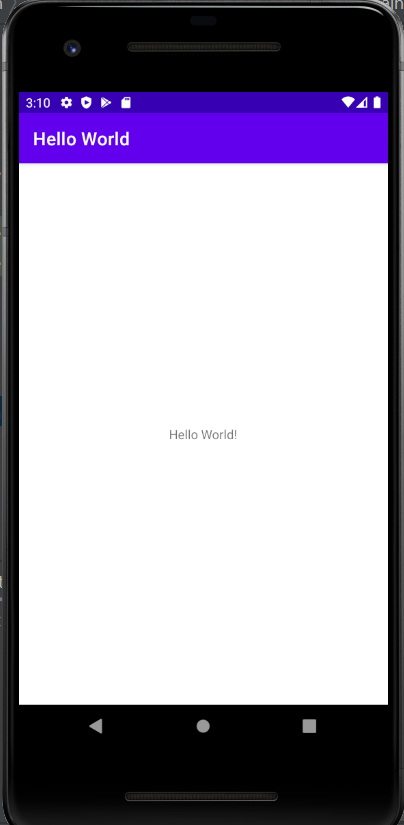
Dans cette tâche, vous allez enfin lancer votre application Hello World.

* 1. Dans Android Studio, choisissez **Run > Run app** ou cliquez sur l'icône **Run** dans la barre d'outils.
  2. Ou choisissez directement, dans la liste, le périphérique virtuel que vous venez de créer, puis cliquez sur **OK**.



L'émulateur démarre comme un périphérique physique. Votre application est créée et une fois que l'émulateur est prêt, Android Studio la télécharge sur l'émulateur et l'exécute.

Vous devriez voir l'application Hello World comme indiqué dans la figure suivante.



## Tâche 4: utiliser un périphérique physique

Dans cette dernière tâche, vous exécuterez votre application sur un périphérique mobile physique tel qu'un téléphone ou une tablette. Vous devez toujours tester vos applications sur des périphériques virtuels et physiques.

Qu'est-ce que vous avez besoin:

* Un appareil Android tel qu'un téléphone ou une tablette.
* Un câble de données pour connecter votre appareil Android à votre ordinateur via le port USB.
* Si vous utilisez un système Linux ou Windows, vous devrez peut-être exécuter des étapes supplémentaires pour pouvoir s'exécuter sur un périphérique matériel. Consultez [la documentation Utilisation de périphériques matériels](http://developer.android.com/tools/device.html). Vous devrez peut-être également installer le pilote USB approprié pour votre appareil. Pour les pilotes USB Windows, voir [OEM USB Drivers](http://developer.android.com/tools/extras/oem-usb.html).

### Activer le débogage USB

Pour permettre à Android Studio de communiquer avec votre appareil, vous devez activer le débogage USB sur votre appareil Android. Ceci est activé dans les paramètres **Developer options** de votre appareil.

Sur Android 4.2 et supérieur, l'écran **Developer options** est masqué par défaut. Pour afficher les options du développeur et activer le **USB Debugging**:

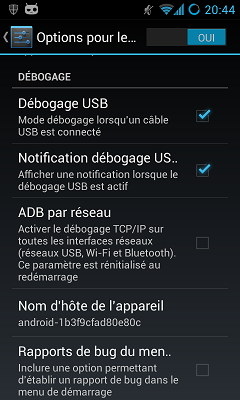
* 1. Sur votre appareil, ouvrez **Settings**, recherchez **About phone**, cliquez sur **About phone**, puis appuyez sur **Build number** sept fois.

* 1. Retournez à l'écran précédent (**Settings / System**). **Developer options** apparaît dans la liste. Appuyez sur **Developer options**.



* 1. Choisissez **USB Debugging**.



### Exécuter votre application sur un appareil

Vous pouvez maintenant connecter votre appareil et exécuter l'application à partir d'Android Studio.

* 1. Connectez votre appareil à votre ordinateur de développement avec un câble USB.
  2. Cliquez sur le bouton **Run** dans la barre d'outils.
  3. Ou sélectionnez votre appareil dans la liste des émulateurs disponibles et des périphériques connectés et cliquez sur **OK**.

Android Studio installe et exécute l'application sur votre appareil.

Vous devrez peut-être installer le pilote USB approprié pour votre périphérique. Consultez [la documentation Utilisation de périphériques matériels](http://developer.android.com/tools/device.html).