

TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



GIÁO TRÌNH
THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI
ĐỘNG

Hà Nội, 2.2025

MỤC LỤC

CHƯƠNG 1.	Làm quen	4
Bài 1)	Tạo ứng dụng đầu tiên	4
1.1)	Android Studio và Hello World	4
1.2)	Giao diện người dùng tương tác đầu tiên	15
1.3)	Trình chỉnh sửa bố cục	15
1.4)	Văn bản và các chế độ cuộn	15
1.5)	Tài nguyên có sẵn	15
Bài 2)	Activities	15
2.1)	Activity và Intent	15
2.2)	Vòng đời của Activity và trạng thái	15
2.3)	Intent ngầm định	15
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ	15
3.1)	Trình gỡ lỗi	15
3.2)	Kiểm thử đơn vị	15
3.3)	Thư viện hỗ trợ	15
CHƯƠNG 2.	Trải nghiệm người dùng	16
Bài 1)	Tương tác người dùng	16
1.1)	Hình ảnh có thể chọn	16
1.2)	Các điều khiển nhập liệu	16
1.3)	Menu và bộ chọn	16
1.4)	Điều hướng người dùng	16
1.5)	RecyclerView	16
Bài 2)	Trải nghiệm người dùng thú vị	16
2.1)	Hình vẽ, định kiểu và chủ đề	16
2.2)	Thẻ và màu sắc	16
2.3)	Bố cục thích ứng	16
Bài 3)	Kiểm thử giao diện người dùng	16

3.1)	Espresso cho việc kiểm tra UI	16
CHƯƠNG 3.	Làm việc trong nền	16
Bài 1)	Các tác vụ nền	16
1.1)	AsyncTask	16
1.2)	AsyncTask và AsyncTaskLoader	16
1.3)	Broadcast receivers	16
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền	16
2.1)	Thông báo	16
2.2)	Trình quản lý cảnh báo	16
2.3)	JobScheduler	16
CHƯƠNG 4.	Lưu dữ liệu người dùng	17
Bài 1)	Tùy chọn và cài đặt	17
1.1)	Shared preferences	17
1.2)	Cài đặt ứng dụng	17
Bài 2)	Lưu trữ dữ liệu với Room	17
2.1)	Room, LiveData và ViewModel	17
2.2)	Room, LiveData và ViewModel	17

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

Giới thiệu

Trong bài thực hành này, bạn sẽ tìm hiểu cách cài đặt Android Studio, môi trường phát triển Android. Bạn cũng sẽ tạo và chạy ứng dụng Android đầu tiên của mình, Hello World, trên một trình giả lập và trên một thiết bị vật lý.

Những gì Bạn nên biết

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng lập trình hướng đối tượng sử dụng một IDE (môi trường phát triển tích hợp) như Android Studio.
- Chứng minh rằng bạn có ít nhất 1-3 năm kinh nghiệm trong lập trình hướng đối tượng, với một phần trong số đó tập trung vào ngôn ngữ lập trình Java. (Các bài thực hành này sẽ không giải thích về lập trình hướng đối tượng hoặc ngôn ngữ Java.

Những gì Bạn sẽ cần:

- Một máy tính chạy Windows hoặc Linux, hoặc một Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một phương pháp thay thế để tải các cài đặt mới nhất của Android Studio và Java lên máy tính của bạn.

Những gì bạn sẽ học

- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.
- Cách thêm thông điệp ghi lại vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

Những gì bạn sẽ làm

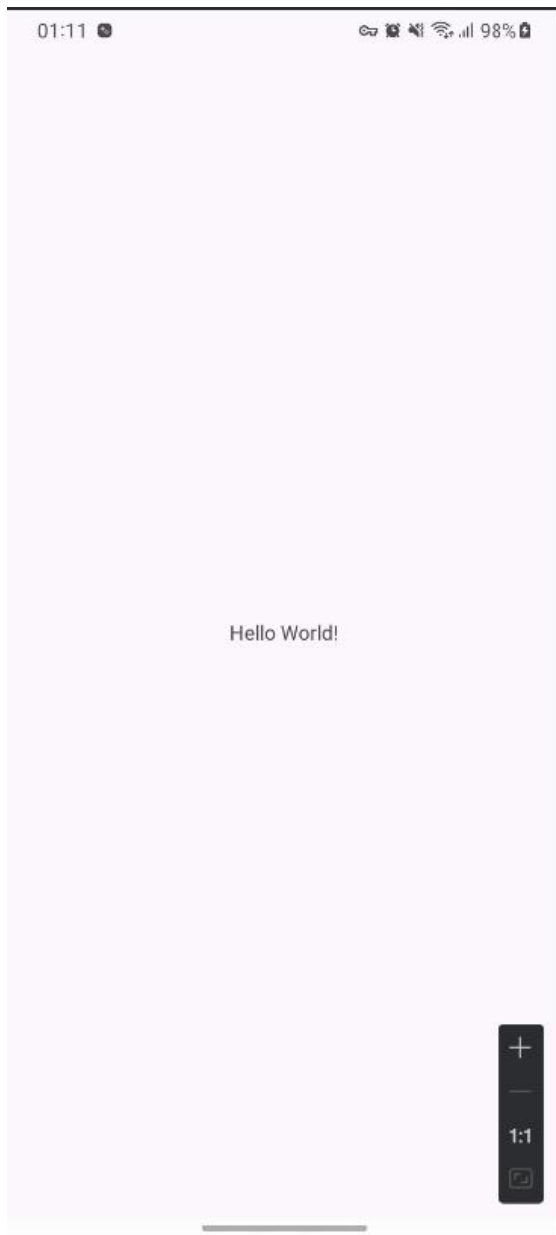
- Cài đặt môi trường phát triển **Android Studio**.
- Tạo một trình giả lập (thiết bị ảo) để chạy ứng dụng của bạn trên máy tính.
- Tạo và chạy ứng dụng **Hello World** trên các thiết bị ảo và vật lý.

- Khám phá cấu trúc dự án.
- Tạo và xem các thông điệp ghi lại từ ứng dụng của bạn.
- Khám phá tệp **AndroidManifest.xml**

Tổng quan về ứng dụng

Sau khi cài đặt thành công Android Studio, bạn sẽ tạo một dự án mới từ mẫu có sẵn cho ứng dụng Hello World. Ứng dụng đơn giản này hiển thị dòng chữ "Hello World" trên màn hình của thiết bị Android (ảo hoặc thật).

Đây là giao diện của ứng dụng khi hoàn thành:



Nhiệm vụ 1: Cài đặt Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp (IDE) hoàn chỉnh bao gồm một trình soạn thảo mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ để phát triển, gỡ lỗi, thử nghiệm và hiệu suất giúp phát triển ứng dụng nhanh hơn và dễ dàng hơn. Bạn có thể thử nghiệm các ứng dụng của mình bằng nhiều trình giả lập được cấu hình sẵn hoặc trên thiết bị di động của riêng bạn, xây dựng các ứng dụng sản xuất và xuất bản trên cửa hàng Google Play.

Lưu ý: Android Studio liên tục được cải tiến. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, hãy xem Android Studio.

Android Studio có sẵn cho các máy tính chạy Windows hoặc Linux, và cho máy Mac chạy macOS. Phiên bản OpenJDK (Java Development Kit) mới nhất được tích hợp sẵn với Android Studio. Để bắt đầu sử dụng Android Studio, trước tiên hãy kiểm tra yêu cầu hệ thống để đảm bảo rằng hệ thống của bạn đáp ứng được chúng. Quy trình cài đặt tương tự cho tất cả các nền tảng. Mọi khác biệt sẽ được ghi chú bên dưới.

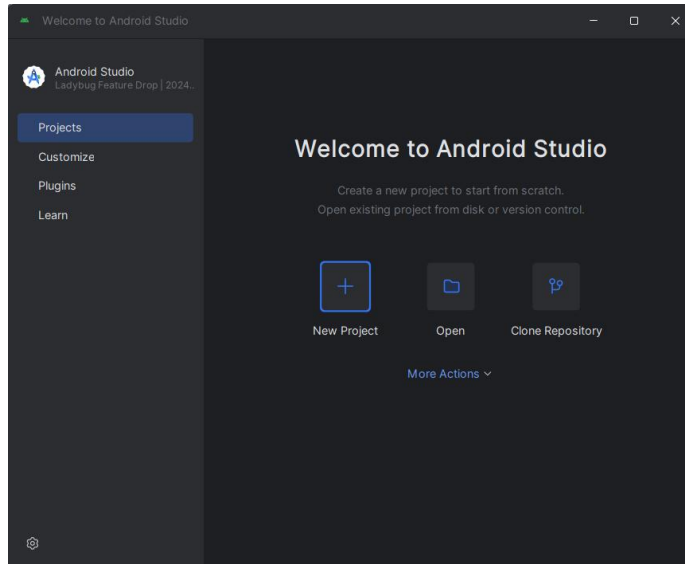
1. Truy cập vào trang web dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.
2. Chấp nhận cấu hình mặc định cho tất cả các bước, và đảm bảo rằng tất cả các thành phần được chọn để cài đặt.
3. Sau khi hoàn tất cài đặt, Trình hướng dẫn Thiết lập sẽ tải xuống và cài đặt một số thành phần bổ sung bao gồm Android SDK. Hãy kiên nhẫn, việc này có thể mất một thời gian tùy thuộc vào tốc độ Internet của bạn, và một số bước có thể có vẻ trùng lặp.
4. Khi việc tải xuống hoàn tất, Android Studio sẽ khởi động, và bạn đã sẵn sàng để tạo dự án đầu tiên của mình.

Nhiệm vụ 2: Tạo ứng dụng Hello World

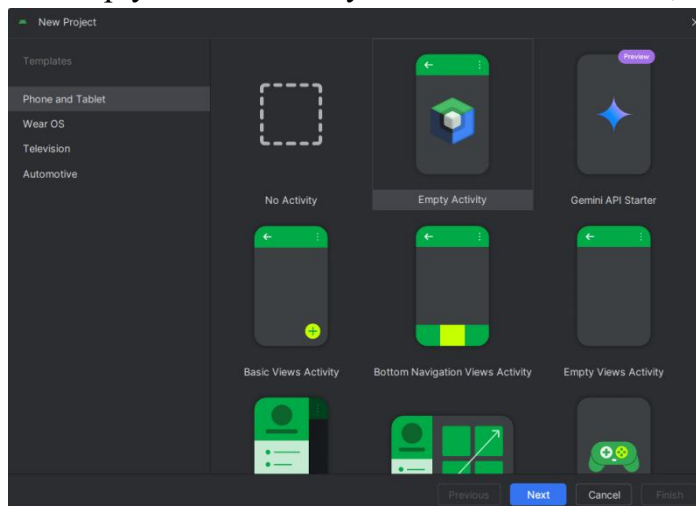
Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị "Hello World" để xác minh rằng Android Studio đã được cài đặt đúng cách, và để học các kiến thức cơ bản về phát triển với Android Studio.

2.1 Tạo dự án ứng dụng

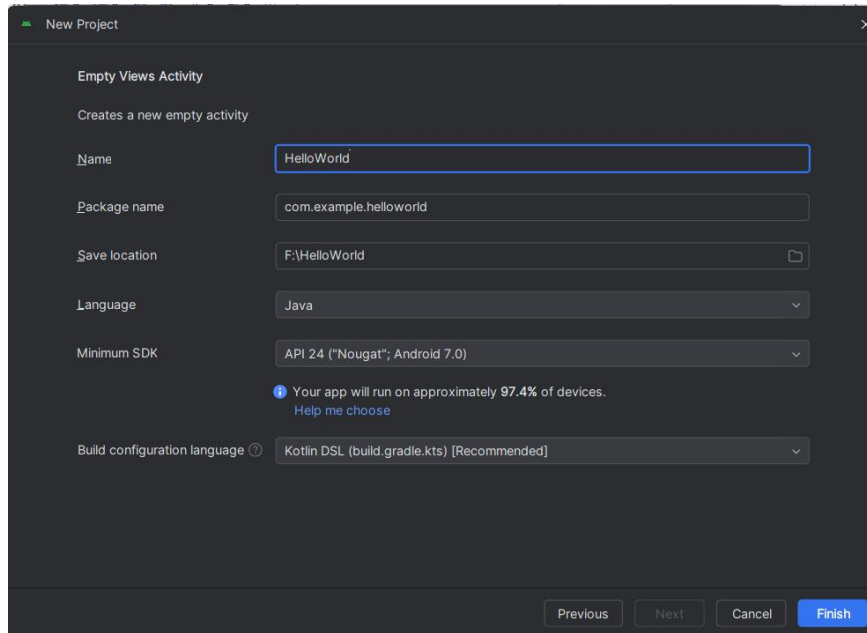
1. Mở Android Studio nếu chưa được mở.
2. Trong cửa sổ chính Welcome to Android Studio, nhấp vào Start a new Android Studio project.



3. Cửa sổ Add an Activity xuất hiện. Activity là một thao tác đơn lẻ, tập trung mà người dùng có thể thực hiện. Đây là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một Activity thường có một layout đi kèm định nghĩa cách các thành phần UI xuất hiện trên màn hình. Android Studio cung cấp các mẫu Activity để giúp bạn bắt đầu. Đối với dự án Hello World, chọn Empty Views Activity như hiển thị bên dưới, và nhấp Next.



4. Trong cửa sổ Create Android Project, nhập HelloWorld cho tên ứng dụng (Application name).



5. Xác nhận rằng vị trí dự án mặc định (Project location) là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác, hoặc thay đổi nó thành thư mục ưa thích của bạn.
6. Chấp nhận android.example.com mặc định cho Company Domain (Tên miền công ty), hoặc tạo một tên miền công ty riêng. Nếu bạn không có kế hoạch xuất bản ứng dụng của mình, bạn có thể chấp nhận mặc định. Lưu ý rằng việc thay đổi tên gói của ứng dụng sau này sẽ tốn thêm công sức.
7. Không đánh dấu vào các tùy chọn Include C++ support và Include Kotlin support, và nhấp Next.
8. Trên màn hình Target Android Devices, Phone and Tablet (Điện thoại và Máy tính bảng) nên được chọn. Đảm bảo rằng API 24: được đặt làm Minimum SDK; nếu không, hãy sử dụng menu popup để đặt nó.
9. Không đánh dấu vào Include Instant App support và tắt cả các tùy chọn khác. Sau đó nhấp Next. Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK đích đã chọn, Android Studio sẽ tự động cài đặt chúng.
10. Nhấp Finish

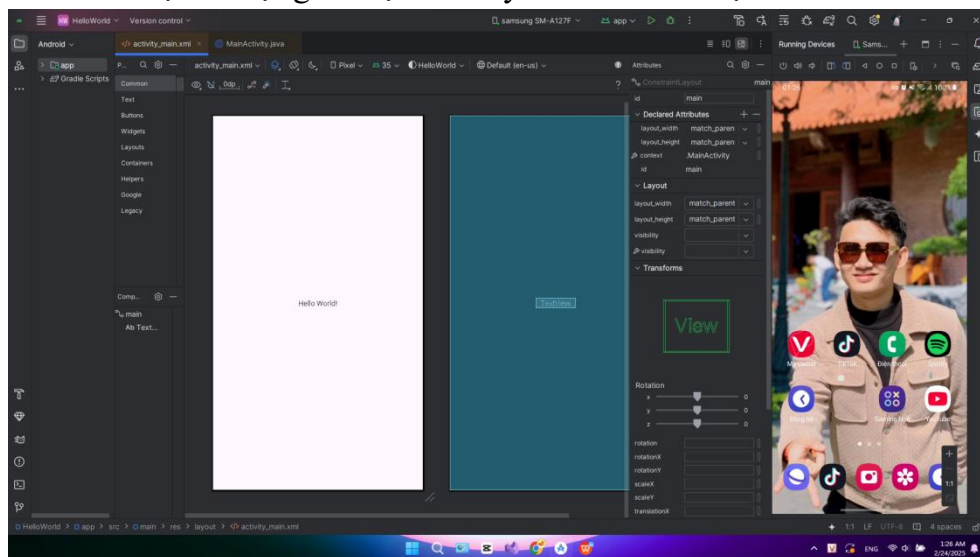
Android Studio tạo một thư mục cho các dự án của bạn, và xây dựng dự án bằng Gradle (điều này có thể mất một vài phút).

Mẹo: Xem trang [Configure your build](#) dành cho nhà phát triển để biết thông tin chi tiết.

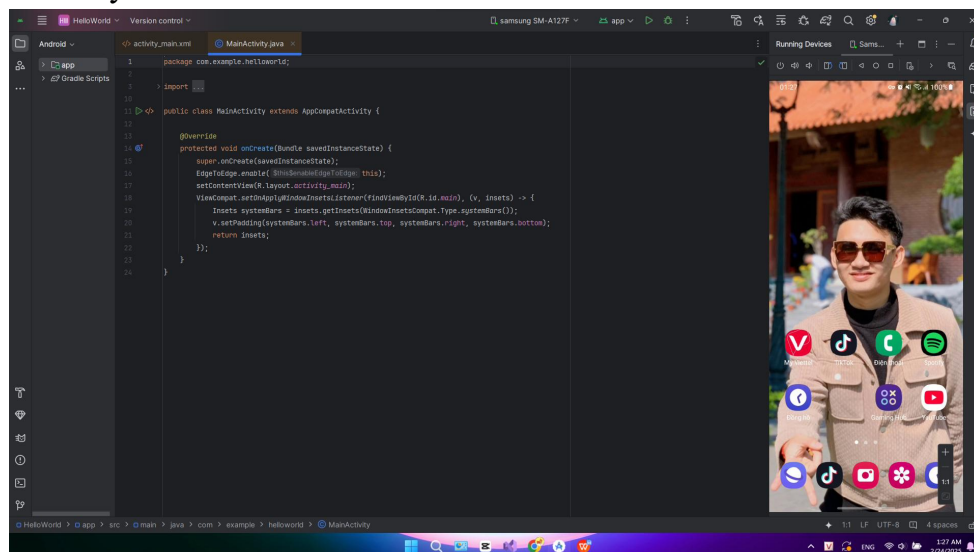
Bạn cũng có thể thấy thông báo "Tip of the day" với các phím tắt và các mẹo hữu ích khác. Nhấp Close để đóng thông báo.

Trình soạn thảo Android Studio xuất hiện. Thực hiện theo các bước sau:

1. Nhấp vào tab **activity_main.xml** để xem trình soạn thảo layout.
2. Nhấp vào tab Design của trình soạn thảo layout, nếu chưa được chọn, để hiển thị bản dựng đồ họa của layout như hiển thị bên dưới.



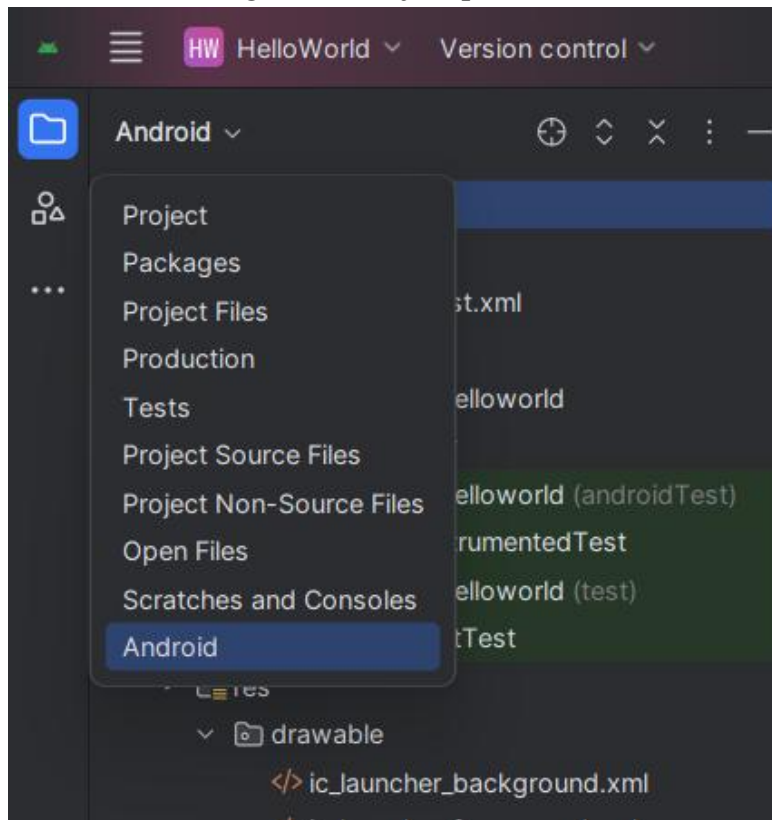
3. Nhấp vào tab **MainActivity.java** để xem trình soạn thảo mã như hình dưới đây



2.2 Tạo dự án ứng dụng

Trong bài thực hành này, bạn sẽ khám phá cách tổ chức dự án trong Android Studio.

1. Nếu chưa được chọn, hãy nhấp vào **Project** trong cột tab dọc ở phía bên trái của sổ Android Studio. Ngăn Dự án sẽ xuất hiện.
2. Để xem dự án theo cấu trúc chuẩn của Android, hãy chọn **Android** từ menu thả xuống ở đầu Project pane, như minh họa bên dưới.

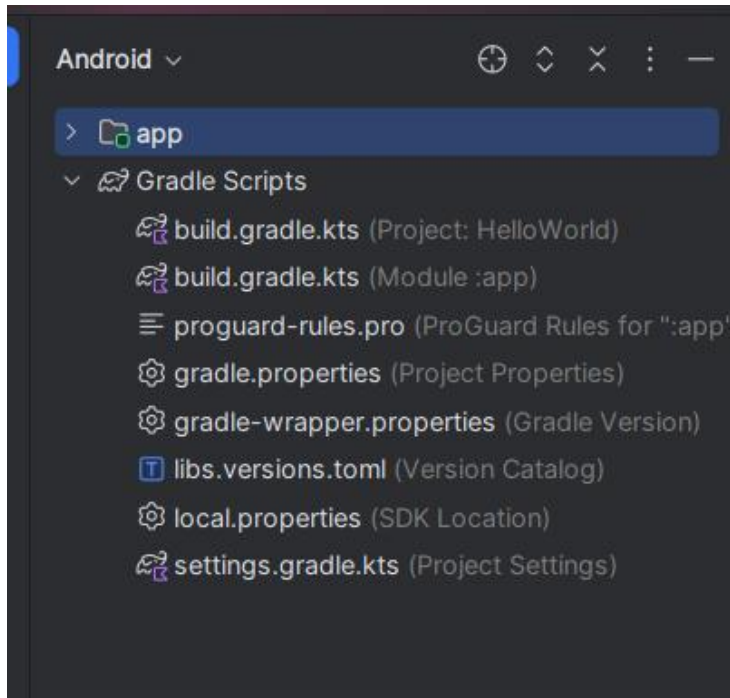


Lưu ý: Chương này và các chương khác đề cập đến Project pane khi được đặt thành Android là Project > Android pane.

2.3 Tìm hiểu về thư mục Gradle Scripts

Hệ thống xây dựng Gradle trong Android Studio giúp bạn dễ dàng thêm các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng của mình dưới dạng dependencies.

Khi bạn tạo một dự án ứng dụng lần đầu tiên, Project > Android pane sẽ xuất hiện với thư mục Gradle Scripts được mở rộng như minh họa bên dưới.



Nếu thư mục Gradle Scripts chưa được mở rộng, hãy nhấp vào biểu tượng tam giác để mở rộng nó.

Làm theo các bước sau để tìm hiểu hệ thống Gradle:

1. Nếu thư mục Gradle Scripts chưa được mở rộng, hãy nhấp vào biểu tượng tam giác để mở rộng nó.

Thư mục này chứa tất cả các tệp cần thiết cho hệ thống build.

2. Tìm tệp **build.gradle (Project: HelloWorld)**.

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô-đun trong dự án của bạn. Mỗi dự án Android Studio đều có một tệp Gradle build cấp cao duy nhất. Hầu hết thời gian, bạn sẽ không cần thay đổi tệp này, nhưng việc hiểu nội dung của nó vẫn rất hữu ích.

Theo mặc định, tệp build cấp cao nhất sử dụng khối `buildscript` để xác định các kho lưu trữ Gradle và các phụ thuộc chung cho tất cả các mô-đun trong dự án. Khi phụ thuộc của bạn không phải là thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong các kho lưu trữ trực tuyến được chỉ định trong khối `repositories` của tệp này. Mặc định, các dự án mới trong Android Studio khai báo JCenter và Google (bao gồm cả kho Maven của Google) làm vị trí kho lưu trữ.

3. Tìm tệp **build.gradle**(Module: app).

Ngoài tệp **build.gradle** ở cấp dự án, mỗi mô-đun cũng có tệp **build.gradle** riêng, giúp bạn cấu hình các thiết lập xây dựng cho từng mô-đun cụ thể (trong ứng dụng HelloWorld, chỉ có một mô-đun).

Tệp này thường là tệp cần chỉnh sửa khi thay đổi các cấu hình cấp ứng dụng, chẳng hạn như khai báo các thư viện phụ thuộc trong phần **dependencies**. Bạn có thể khai báo một thư viện phụ thuộc bằng một trong nhiều cấu hình phụ thuộc khác nhau. Mỗi cấu hình phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ, câu lệnh **implementation** **fileTree(dir: 'libs', include: ['*.jar'])** thêm tất cả các tệp “**.jar**” bên trong thư mục **libs** làm thư viện phụ thuộc.

Đây là tệp **build.gradle** (Module: app) cho ứng dụng **HelloWorld**:

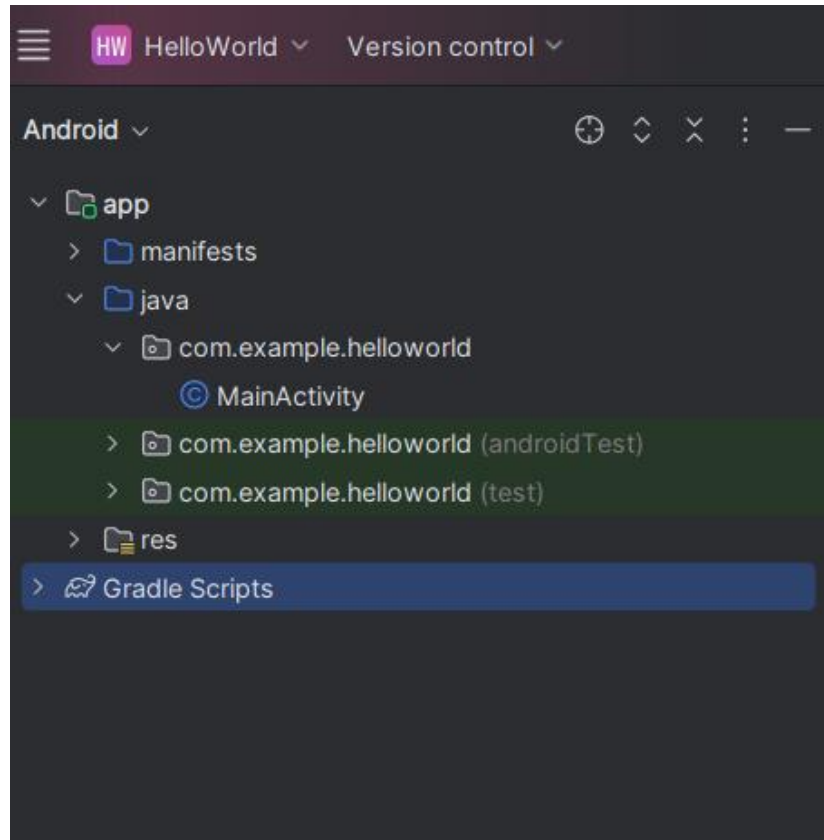
```
1  plugins {
2      alias(libs.plugins.android.application)
3  }
4
5  android {
6      namespace = "com.example.helloworld"
7      compileSdk = 35
8
9      defaultConfig {
10         applicationId = "com.example.helloworld"
11         minSdk = 24
12         targetSdk = 35
13         versionCode = 1
14         versionName = "1.0"
15
16         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes {
20         release {
21             isMinifyEnabled = false
22             proguardFiles(
23                 getDefaultProguardFile( name: "proguard-android-optimize.txt"),
24                 "proguard-rules.pro"
25             )
26         }
27     }
28     compileOptions {
29         sourceCompatibility = JavaVersion.VERSION_11
30         targetCompatibility = JavaVersion.VERSION_11
31     }
32 }
33
34 dependencies {
35     implementation(libs.appcompat)
36     implementation(libs.material)
37     implementation(libs.activity)
38     implementation(libs.constraintlayout)
39     testImplementation(libs.junit)
40     androidTestImplementation(libs.ext.junit)
41     androidTestImplementation(libs.espresso.core)
42 }
43 }
```

4. Nhấp vào biểu tượng tam giác để đóng **Gradle Scripts**.

2.4 Tìm hiểu các thư mục app và res.

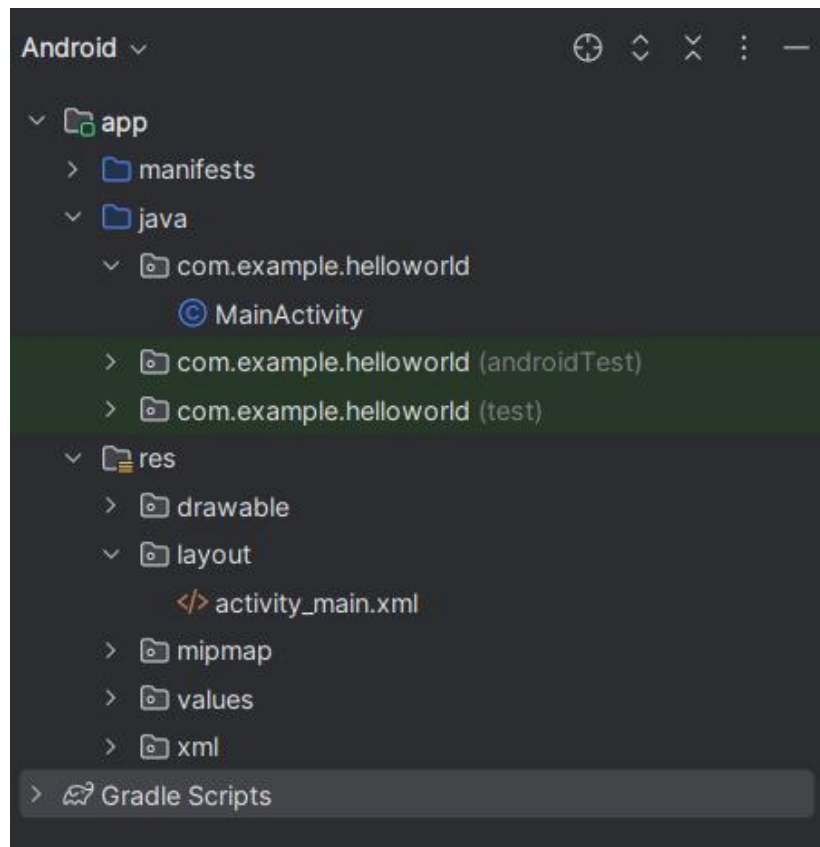
Tất cả mã nguồn và tài nguyên cho ứng dụng đều nằm trong thư mục app và res.

1. Mở rộng thư mục app, thư mục java, và thư mục com.example.android.helloworld để xem tệp MainActivity.java. Nhấp đúp vào tệp để mở nó trong trình chỉnh sửa mã.



Thư mục **java** bao gồm các tệp lớp Java trong ba thư mục con, như hình trên. Thư mục **com.example.hello.helloworld** (hoặc tên miền bạn đã chỉ định) chứa tất cả các tệp cho một gói ứng dụng. Hai thư mục còn lại được sử dụng để kiểm thử và sẽ được mô tả trong một bài học khác. Trong ứng dụng Hello World, chỉ có một gói duy nhất và nó chứa MainActivity.java. Tên của Activity đầu tiên (màn hình đầu tiên mà người dùng thấy), cũng như khởi tạo các tài nguyên toàn cục của ứng dụng, theo thông lệ được gọi là **MainActivity** (phần mở rộng tệp bị ẩn trong **Project > Android** pane).

2. Mở rộng thư mục res, sau đó mở thư mục **layout**, và nhấp đúp vào tệp **activity_main.xml** để mở nó trong Layout Editor.



Thư mục **res** chứa các tài nguyên như bố cục (layout), chuỗi (strings) và hình ảnh (images). Một **Activity** thường được liên kết với một bố cục giao diện người dùng (UI) được định nghĩa trong một tệp XML. Tệp này thường được đặt tên theo **Activity** tương ứng.

2.5 Khám phá thư mục manifests.

Thư mục **manifests** chứa các tệp cung cấp thông tin quan trọng về ứng dụng của bạn cho hệ thống Android. Hệ thống cần những thông tin này trước khi có thể chạy bất kỳ đoạn mã nào của ứng dụng.

1. Mở rộng thư mục **manifests**.
2. Mở tệp **AndroidManifest.xml**.

Tệp **AndroidManifest.xml** mô tả tất cả các thành phần của ứng dụng Android. Mọi thành phần của ứng dụng, chẳng hạn như từng **Activity**, phải được khai báo trong tệp XML này. Trong các bài học khác của khóa học, bạn sẽ chỉnh sửa tệp này để thêm tính năng và quyền truy cập. Để tìm hiểu tổng quan, hãy xem **App Manifest Overview**.

Nhiệm vụ 3: Sử dụng thiết bị ảo (trình giả lập)

1.2) Giao diện người dùng tương tác đầu tiên

1.3) Trình chỉnh sửa bố cục

1.4) Văn bản và các chế độ cuộn

1.5) Tài nguyên có sẵn

Bài 2) Activities

2.1) Activity và Intent

2.2) Vòng đời của Activity và trạng thái

2.3) Intent ngầm định

Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ

3.1) Trình gỡ lỗi

3.2) Kiểm thử đơn vị

3.3) Thư viện hỗ trợ

CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG

Bài 1) Tương tác người dùng

- 1.1) Hình ảnh có thể chọn**
- 1.2) Các điều khiển nhập liệu**
- 1.3) Menu và bộ chọn**
- 1.4) Điều hướng người dùng**
- 1.5) RecyclerView**

Bài 2) Trải nghiệm người dùng thú vị

- 2.1) Hình vẽ, định kiểu và chủ đề**
- 2.2) Thẻ và màu sắc**
- 2.3) Bố cục thích ứng**

Bài 3) Kiểm thử giao diện người dùng

- 3.1) Espresso cho việc kiểm tra UI**

CHƯƠNG 3. LÀM VIỆC TRONG NỀN

Bài 1) Các tác vụ nền

- 1.1) AsyncTask**
- 1.2) AsyncTask và AsyncTaskLoader**
- 1.3) Broadcast receivers**

Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền

- 2.1) Thông báo**
- 2.2) Trình quản lý cảnh báo**
- 2.3) JobScheduler**

CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG

Bài 1) Tùy chọn và cài đặt

1.1) Shared preferences

1.2) Cài đặt ứng dụng

Bài 2) Lưu trữ dữ liệu với Room

2.1) Room, LiveData và ViewModel

2.2) Room, LiveData và ViewModel