

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.....	3
Bài 1)	Tạo ứng dụng đầu tiên	3
1.1)	Android Studio và Hello World	3
1.2)	Phần A: giao diện người dùng tương tác đầu tiên của bạn	31
Bài 2)	Activities	33
2.1)	Activity và Intent	33
2.2)	Vòng đời của Activity và trạng thái	33
2.3)	Intent ngầm định.....	33
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ.....	33
3.1)	Trình gỡ lỗi	33
3.2)	Kiểm thử đơn vị.....	33
3.3)	Thư viện hỗ trợ.....	33
CHƯƠNG 2.	TRẢI NGHIỆM NGƯỜI DÙNG.....	34
Bài 1)	Tương tác người dùng.....	34
1.1)	Hình ảnh có thể chọn	34
1.2)	Các điều khiển nhập liệu	34
1.3)	Menu và bộ chọn	34
1.4)	Điều hướng người dùng	34
1.5)	RecyclerView	34
Bài 2)	Trải nghiệm người dùng thú vị.....	34
2.1)	Hình vẽ, định kiểu và chủ đề	34
2.2)	Thẻ và màu sắc	34
2.3)	Bố cục thích ứng.....	34
Bài 3)	Kiểm thử giao diện người dùng.....	34
3.1)	Espresso cho việc kiểm tra UI	34

CHƯƠNG 3.	LÀM VIỆC TRONG NỀN	34
Bài 1)	Các tác vụ nền.....	34
1.1)	AsyncTask	34
1.2)	AsyncTask và AsyncTaskLoader	34
1.3)	Broadcast receivers	34
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	34
2.1)	Thông báo	34
2.2)	Trình quản lý cảnh báo	34
2.3)	JobScheduler.....	34
CHƯƠNG 4.	LƯU DỮ LIỆU NGƯỜI DÙNG	35
Bài 1)	Tùy chọn và cài đặt.....	35
1.1)	Shared preferences.....	35
1.2)	Cài đặt ứng dụng.....	35
Bài 2)	Lưu trữ dữ liệu với Room	35
2.1)	Room, LiveData và ViewModel.....	35
2.2)	Room, LiveData và ViewModel.....	35

3.1) Trình gỡ lỗi

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 cần 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 Ứng dụng

Sau khi bạn cài đặt thành công Android Studio, bạn sẽ tạo từ mẫu 1 dự án mới cho ứng dụng Hello World. Ứng dụng đơn giản này sẽ hiển thị chuỗi “Hello World” trên màn hình của máy ảo Android hoặc thiết bị vật lý hệ điều hành Android.

Nhiệm vụ 1: Tải Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp hoàn chỉnh bao gồm trình soạn thảo mã cao cấp và các mẫu ứng dụng có sẵn. Ngoài ra, nó còn bao gồm các công cụ hỗ trợ việc phát triển, gỡ lỗi, kiểm thử, giúp việc phát triển ứng dụng nhanh hơn, dễ dàng hơn với hiệu suất cao. Bạn có thể kiểm thử ứng dụng với số lượng lớn các máy ảo đã được cấu hình sẵn hoặc thiết bị di động của bạn, phát triển các ứng dụng triển khai và đăng tải lên Cửa hàng Google Play.

Lưu ý: Android Studio luôn trong quá trình phát triển. Thông tin mới nhất về các yêu cầu hệ thống và hướng dẫn cài đặt ở [Android Studio](#).

Android Studio phù hợp cho các máy tính hệ điều hành Windows hoặc Linux, macOS. Phiên bản mới nhất OpenJDK (Bộ phát triển Java) được bao gồm trong Android Studio.

Để cài đặt và khởi chạy Android Studio, kiểm tra [yêu cầu hệ thống](#) để đảm bảo rằng hệ thống của bạn phù hợp. Cách cài đặt tương tự cho mọi nền tảng. Bên dưới là lưu ý nếu có khác biệt.

1. Truy cập trang [Android developers](#) và làm theo chỉ dẫn để tải và [cài đặt Android Studio](#).
2. Chấp nhận các cấu hình mặc định cho mọi bước, đảm bảo rằng các bước đều được chọn cho việc cài đặt.
3. Sau khi cài đặt thành công, Setup Wizard sẽ tải và cài đặt thêm các phần tử các bao gồm Android SDK. Hãy kiên nhẫn, việc này có thể tốn thời gian phụ thuộc vào tốc độ mạng, một vài bước có vẻ không cần thiết.
4. Khi cài đặt thành công, Android Studio sẽ bắt đầu, bạn đã sẵn sàng tạo dự án đầu tiên của mình

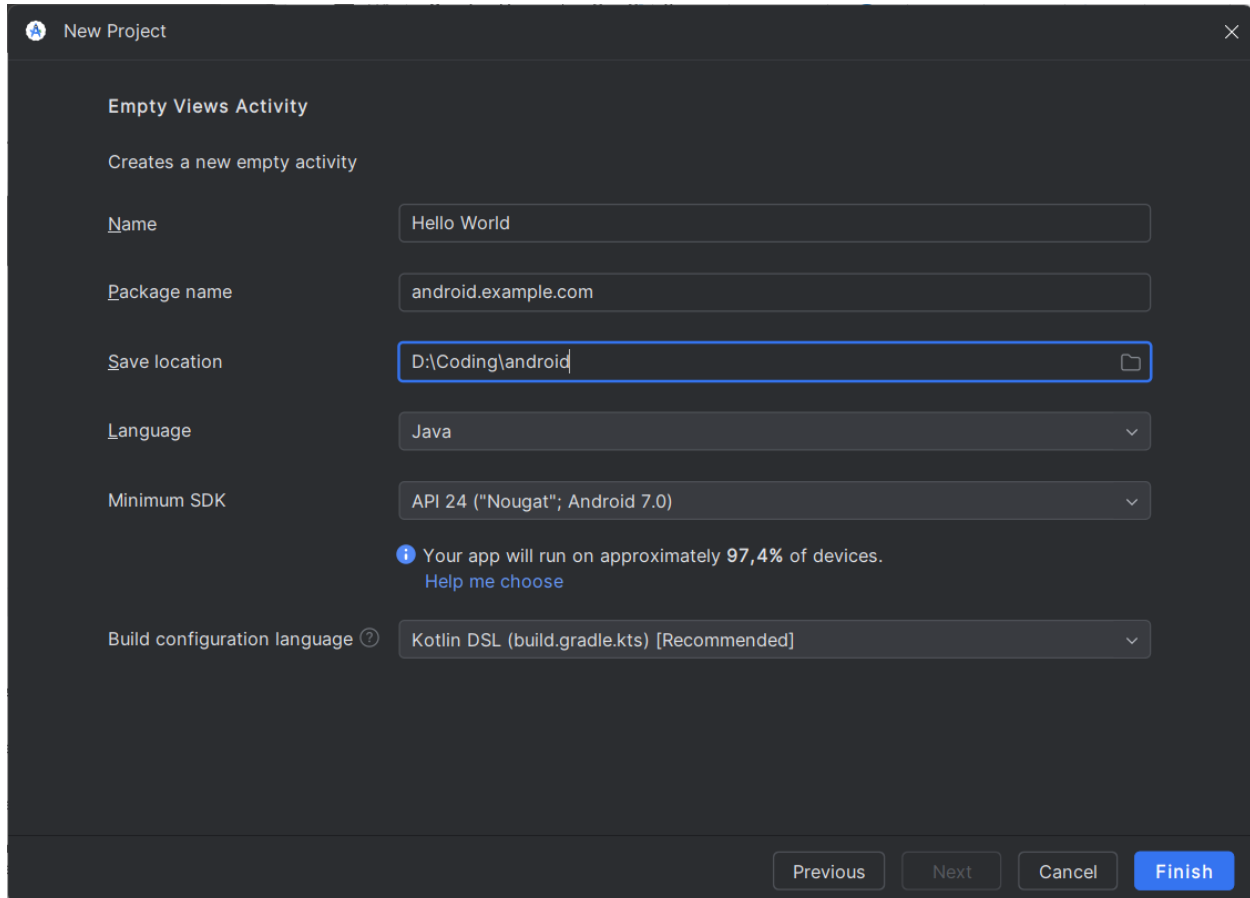
Xử lý sự cố: Nếu bạn gặp lỗi trong quá trình cài đặt, vui lòng truy cập [lưu ý Android Studio](#), hoặc tìm sự trợ giúp từ người hướng dẫn.

Nhiệm vụ 2: Tạo ứng dụng Hello World

Tại nhiệm vụ này, bạn sẽ tạo ứng dụng hiển thị “Hello World” để xác minh rằng Android đã cài đặt hoàn chỉnh, cùng với đó là học lập trình cơ bản với Android Studio.

2.1 Tạo dự án

1. Mở Android Studio nếu chưa mở.
2. Ở màn hình chính **Welcome to Android Studio**, chọn **Start a new Android Studio project**.
3. Tại cửa sổ Create Android Project, gõ **Hello World** cho mục **Application name**.

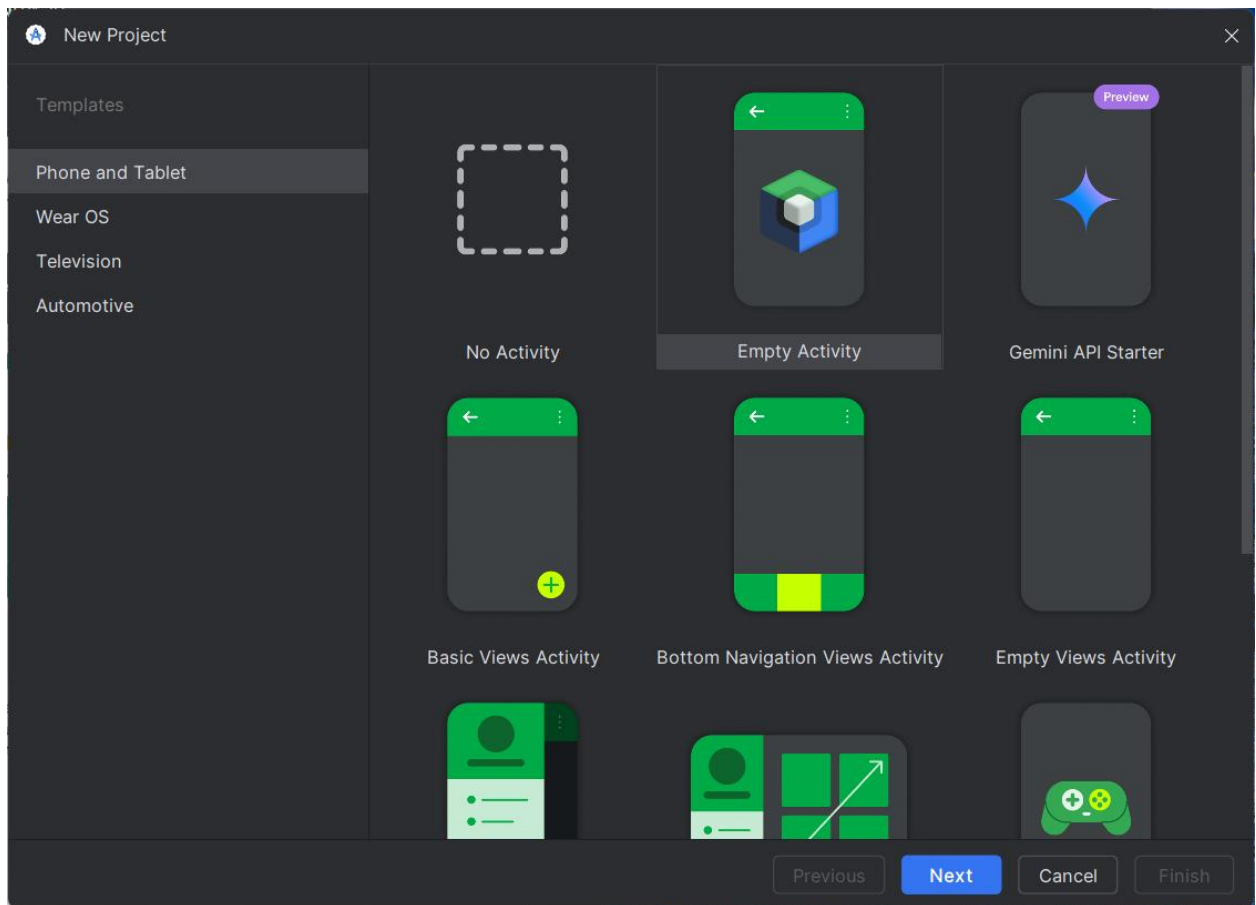


4. Xác minh rằng **Project location** mặc định là nơi bạn muốn lưu giữ ứng dụng Hello World và các dự án Android Studio khác, hoặc hãy đổi sang nơi bạn mong muốn.
5. Chấp nhận mặc định **android.example.com** cho **Company Domain**, hoặc tạo tên miền công ty riêng biệt.

Nếu bạn không có dự định đăng tải ứng dụng, bạn có thể chấp nhận mặc định. Lưu ý rằng việc đổi tên gói cho ứng dụng sau này chỉ là việc bổ sung.

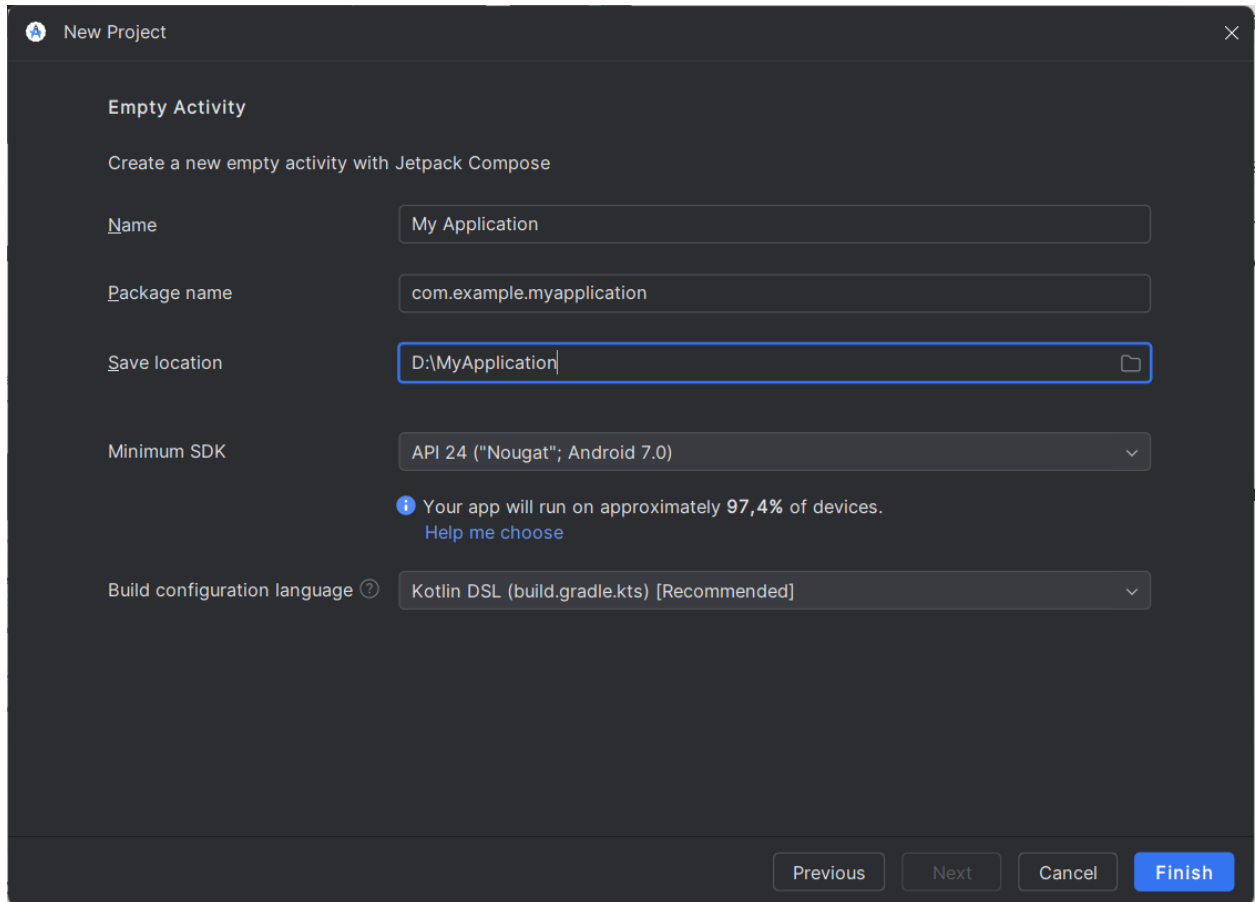
6. Không chọn **Include C++ support** và **Include Kotlin support**, chọn **Next**.
7. Ở **Target Android Devices, Phone and Tablet** nên được chọn. Đảm bảo rằng **API 15: Android 4.0.3 IceCreamSandwich** được cho là SDK nhỏ nhất; nếu không, chọn cửa sổ chọn để lựa chọn.

8. Không chọn **Include Instant App support** và các tùy chọn khác. Sau đó chọn **Next**. Nếu dự án yêu cầu thêm các thành phần với sự lựa chọn SDK, Android Studio sẽ tải chúng tự động.
9. Cửa sổ **Add an Activity** sẽ xuất hiện. [Activity](#) là 1 thứ mà người dùng có thể chú ý. Đó là thành phần thiết yếu của mọi ứng dụng Android. Một Activity thường có 1 bố cục định nghĩa cách các thành phần UI xuất hiện trên màn hình. Android Studio cung cấp các mẫu Activity giúp bạn bắt đầu dự án. Cho dự án Hello World, chọn **Empty Activity** như hình bên dưới, sau đó chọn **Next**.



10. Màn hình **Configure Activity** xuất hiện (Dựa trên template bạn chọn ở bước trước có thể sẽ hiển thị khác nhau). Mặc định, Activity trống của mẫu sẽ mang

tên MainActivity. Bạn có thể đổi nếu muốn, bài học sử dụng tên MainActivity.



11. Đảm bảo rằng mục **Generate layout file** được chọn. Tên bố cục mặc định là activity_main. Bạn có thể đổi nếu muốn, bài học sử dụng tên activity_main.

12. Đảm bảo rằng mục **Backwards Compatibility (App Compat)** được chọn. Việc này sẽ giúp ứng dụng của bạn tương thích với các phiên bản Android đời cũ.

13. Chọn **Finish**.

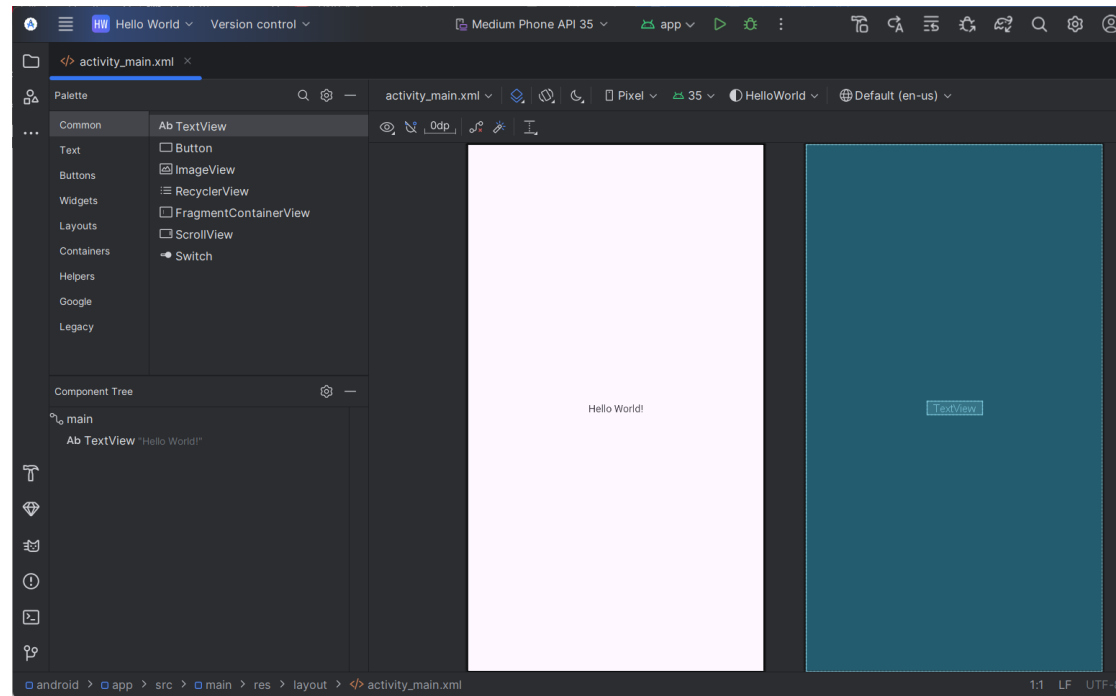
Android Studio tạo thư mục cho dự án của bạn và xây dựng dùng [Gradle](#) (việc này có thể sẽ mất một lúc).

Tip: Xem trang dành cho lập trình viên [tùy chỉnh xây dựng](#) để biết thông tin chi tiết.

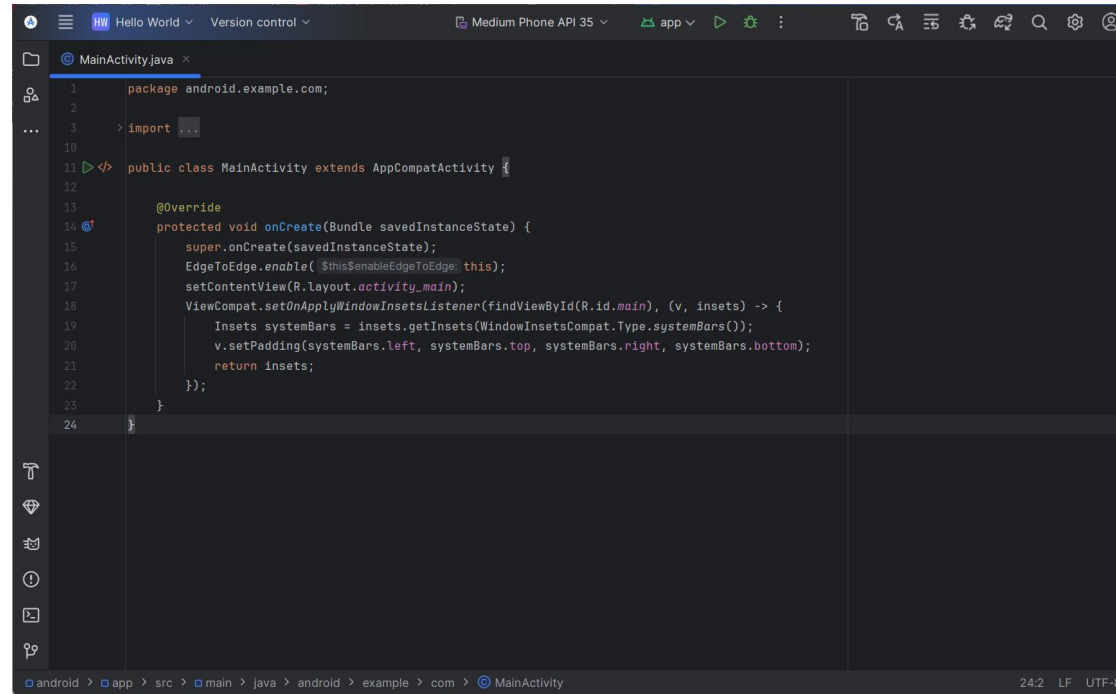
Bạn có thể thấy tin “Tip of the day” cho các phím tắt hoặc mẹo hữu ích. Chọn **Close** để đóng tin.

Trình chỉnh sửa Android Studio xuất hiện. Làm theo các chỉ dẫn:

1. Chọn thẻ **activity_main.xml** để thấy trình chỉnh sửa bố cục.
2. Chọn thẻ **Design (nếu chưa)** trong trình chỉnh sửa bố cục, bản vẽ bố cục được hiển thị như ảnh dưới



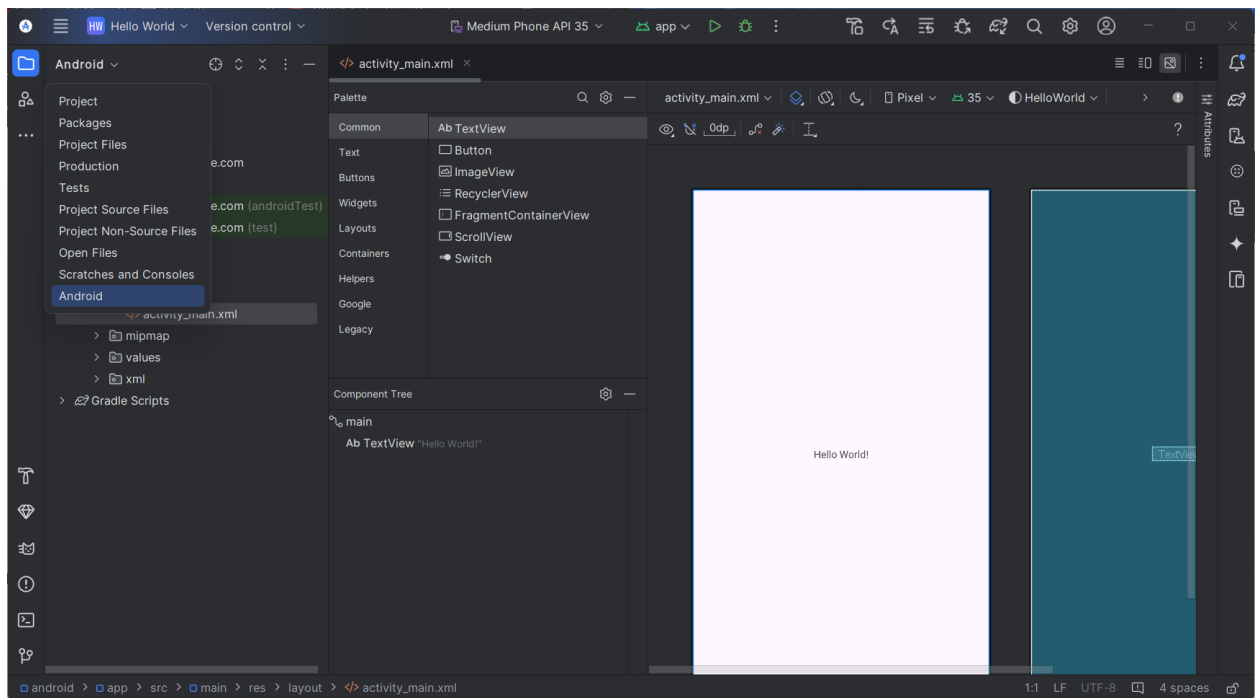
3. Chọn thẻ **MainActivity.java** để thấy trình chỉnh sửa code như bên dưới.



2.2 Khám phá Project > Thẻ Android

Trong ví dụ này, bạn sẽ khám phá cách tổ chức dự án của Android Studio.

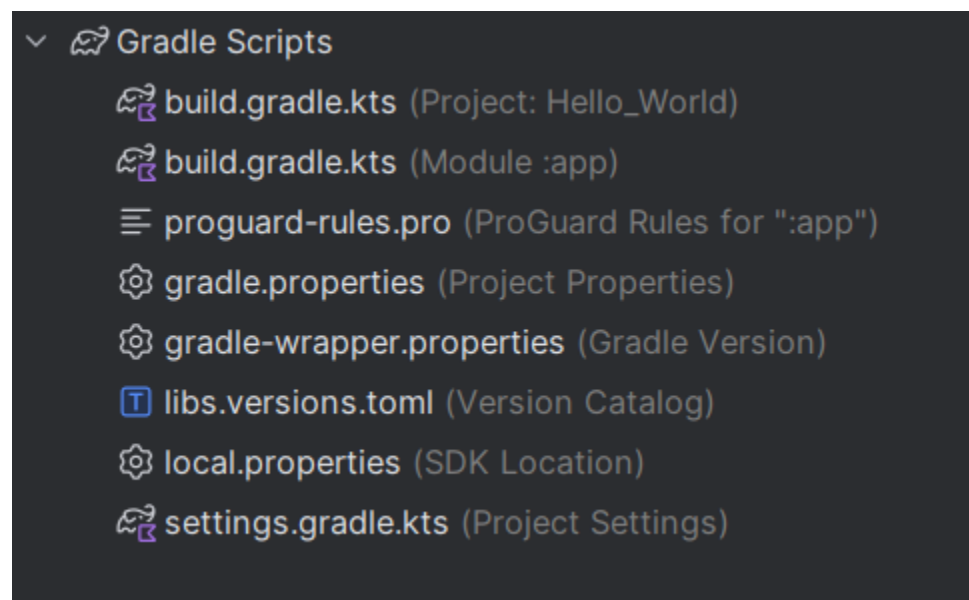
1. Chọn thẻ **Project** (nếu chưa) ở cột thẻ dọc phía bên trái màn hình Android Studio. Thẻ Project xuất hiện.
2. Để hiển thị cấu trúc dự án phân cấp đặt chuẩn của Android, chọn **Android** từ menu bật lên ở phía trên thẻ Project, như hình bên dưới.



2.3 Thư mã Gradle

Hệ thống xây dựng Gradle trong Android Studio giúp dễ dàng trong việc thêm các tệp nhị phân hoặc mô-đun thư viện trong quá trình xây dựng, giống như các phụ thuộc.

Khi bạn tạo dự án lần đầu, thẻ **Project > Android** sẽ xuất hiện cùng với thư mục **Gradle Scripts** được mở rộng như ảnh dưới



Làm theo các bước sau để khám phá hệ thống Gradle:

1. Nếu thư mục **Gradle Scripts** chưa được mở rộng, chọn icon tam giác để mở rộng nó.

Thư mục chứa các tệp cần thiết cho việc xây dựng hệ thống.

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

Đây là nơi bạn sẽ thấy các lựa chọn cấu hình thường dùng cho các mô-đun xây dựng dự án của bạn. Các dự án Android Studio có duy nhất 1 tệp xây dựng Gradle cấp cao. Hầu hết thời gian, bạn sẽ không cần thay đổi nội dung tệp này, những vẫn cần nên biết ý nghĩa nội dung của nó.

Mặc định, tệp xây dựng cấp cao sử dụng khối buildscript để định nghĩa kho lưu trữ và phụ thuộc của Gradle chung cho mọi mô-đun trong dự án. Khi phụ thuộc của bạn không phải là một thư viện cục bộ hoặc một file tree, Gradle sẽ tìm các tệp trong bất kỳ kho lưu trữ trực tuyến nào được chỉ định trong khối repositories của tệp này. Mặc định, dự án Android Studio mới sẽ định nghĩa Jcenter và Google (bao gồm [kho lưu trữ Google Maven](#)) như vị trí kho lưu trữ:

```
// Top-level build file where you can add configuration options common to all sub-pr
plugins {
    alias(libs.plugins.android.application) apply false
}
```

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

Ngoài tệp build.gradle cho cấp dự án, mỗi mô-đun còn có tệp build.gradle riêng, cho phép cấu hình xây dựng cho từng mô-đun (Ứng dụng HelloWorld chỉ có 1 mô-đun). Cấu hình này cho phép bạn tùy ý lựa chọn đóng gói, như loại biên dịch và đóng gói và phiên bản sản phẩm. Bạn cũng có thể ghi đè cài đặt trong tệp AndroidManifest.xml hoặc tệp cấp cao build.gradle.

Đây là tệp thường xuyên được chỉnh sửa khi bạn thay đổi cấu hình ứng dụng, ví dụ như định nghĩa các phụ thuộc. Bạn có thể định nghĩa thư viện phụ thuộc sử dụng 1 trong các cấu hình phụ thuộc khác nhau. Mỗi cấu hình phụ thuộc cung cấp hướng dẫn Gradle về các sử dụng thư viện. Ví dụ, câu lệnh implementation fileTree(dir: 'libs', include: ['*.jar']) thêm phụ thuộc tệp có “.jar” vào thư viện libs.

Tệp **build.gradle(Module:app)** của ứng dụng HelloWorld:

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "android.example.com"
    compileSdk = 35

    defaultConfig {
        applicationId = "android.example.com"
        minSdk = 27
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

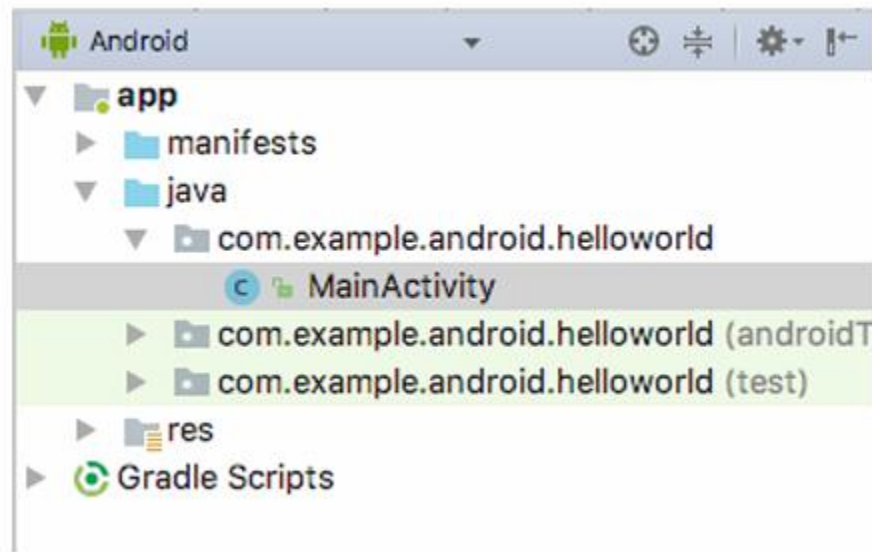
dependencies {
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

4. Chọn icon tam giác để đóng mã **Gradle**.

2.4: Khám phá thư mục app và res

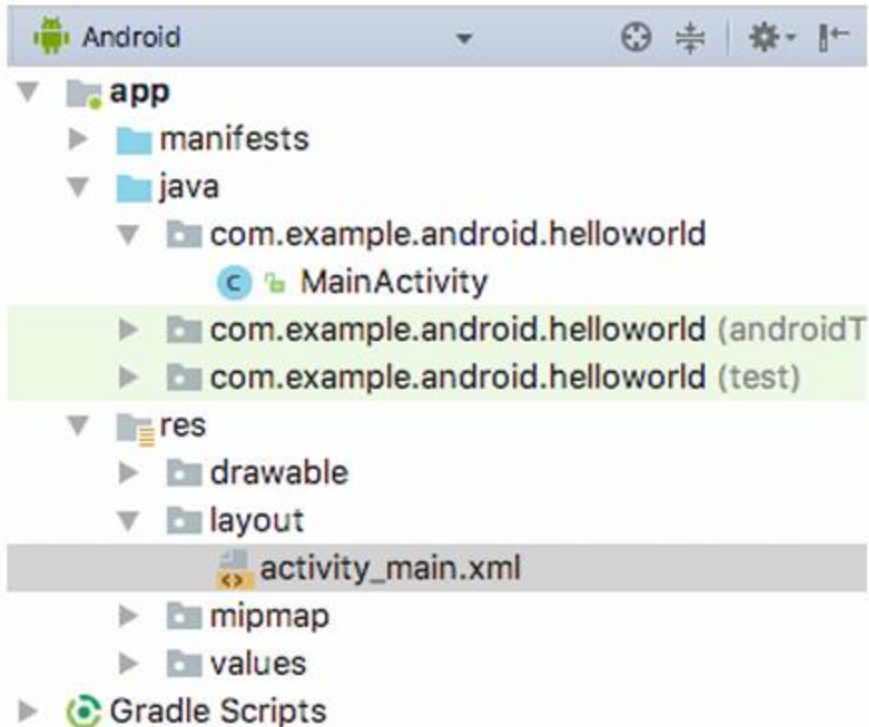
Tất cả mã nguồn và tài nguyên cho ứng dụng nằm trong các thư mục app và res.

1. Mở rộng thư mục app, thư mục java, và thư mục com.example.android.helloworld để thấy tệp MainActivity.java. Nhấp đúp chuột 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 được hiển thị trong hình minh họa 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 khác được sử dụng cho việc kiểm thử và sẽ được mô tả trong bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa MainActivity.java. Tên của Activity (màn hình) đầu tiên mà người dùng thấy, cũng khởi tạo các tài nguyên toàn ứng dụng, thường được gọi là MainActivity (phần mở rộng tệp được bỏ qua trong ngăn Project > Android).

2. Mở rộng thư mục res và thư mục layout, và nhấp đúp chuột vào tệp activity_main.xml để mở nó trong trình chỉnh sửa bố cục.



Thư mục **res** chứa các tài nguyên, như bố cục, chuỗi, và hình ảnh. Một Activity thường được liên kết với một bố cục của các chế độ xem UI được định nghĩa dưới dạng tệp XML. Tệp này thường được đặt tên theo tên Activity của nó.

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 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**, đều 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 sử dụng tính năng.

Để tìm hiểu tổng quan, hãy xem [App Manifest Overview](#).

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

Trong nhiệm vụ này, bạn sẽ sử dụng **Android Virtual Device (AVD) Manager** để tạo một thiết bị ảo (còn được gọi là trình giả lập) mô phỏng cấu hình của một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng.

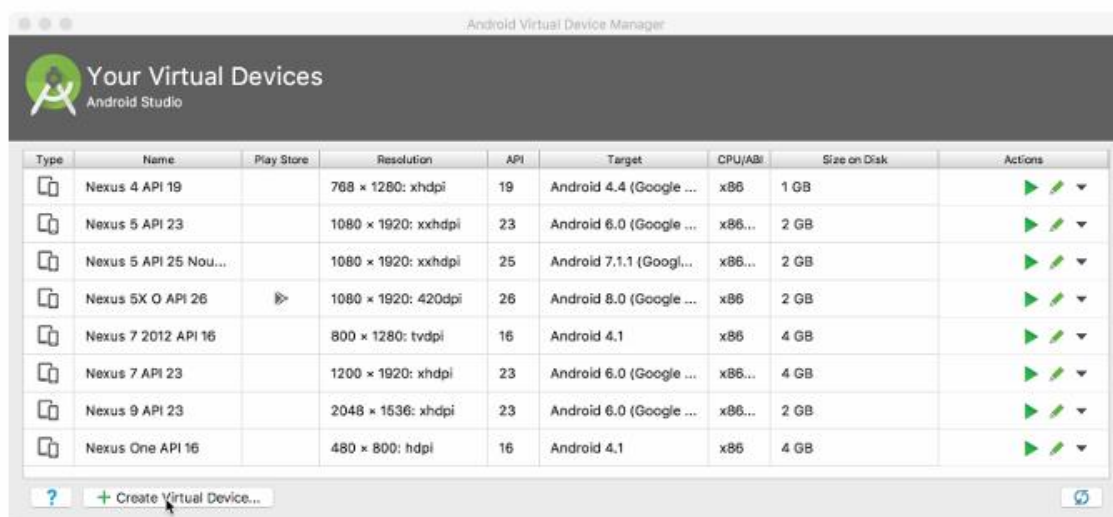
Lưu ý rằng **Android Emulator** có các yêu cầu bổ sung ngoài **yêu cầu hệ thống** cơ bản của Android Studio.

Với AVD Manager, bạn có thể xác định các đặc điểm phần cứng của thiết bị, cấp độ API, dung lượng lưu trữ, giao diện (skin) và các thuộc tính khác, sau đó lưu lại dưới dạng một thiết bị ảo. Việc sử dụng thiết bị ảo giúp bạn kiểm thử ứng dụng trên nhiều cấu hình thiết bị khác nhau (chẳng hạn như máy tính bảng và điện thoại) với các cấp độ API khác nhau mà không cần sử dụng thiết bị thật.

3.1 Tạo máy ảo Android (AVD)

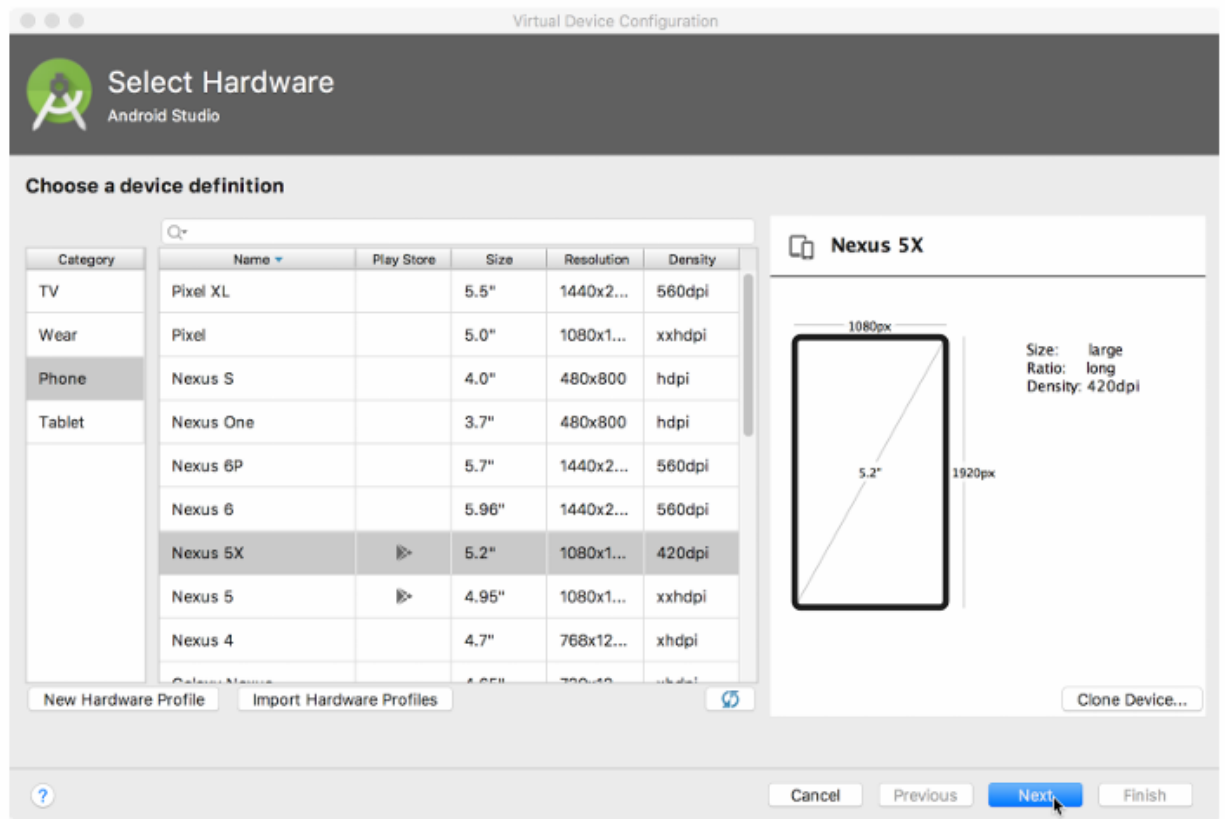
Để chạy trình giả lập trên máy tính của bạn, trước tiên bạn cần tạo một cấu hình mô tả thiết bị ảo.

1. Trong **Android Studio**, chọn **Tools > Android > AVD Manager**, hoặc nhấp vào biểu tượng **AVD Manager** trên thanh công cụ. Màn hình **Your Virtual Devices** sẽ xuất hiện. Nếu bạn đã tạo thiết bị ảo trước đó, danh sách sẽ hiển thị các thiết bị đó (như trong hình bên dưới). Nếu chưa, danh sách sẽ trống.



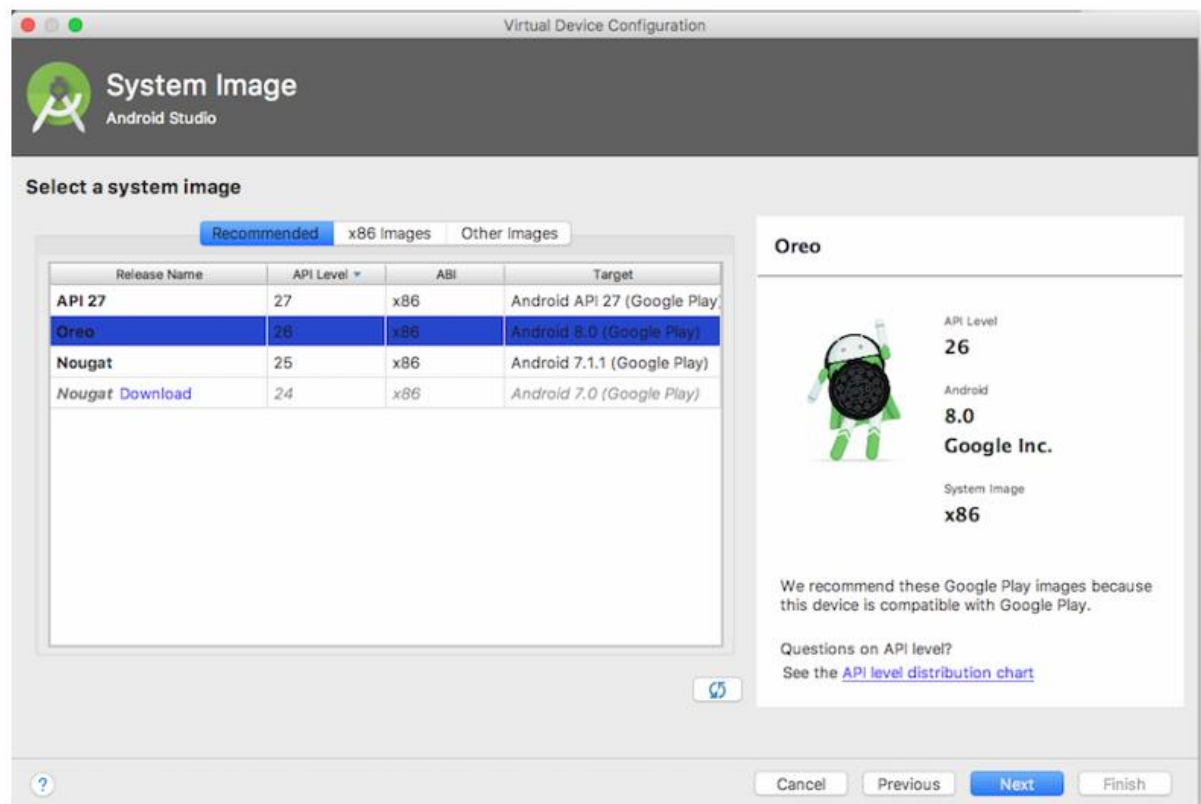
2. Nhấp vào **+Create Virtual Device**. Cửa sổ **Select Hardware** xuất hiện, hiển thị danh sách các thiết bị phần cứng được cấu hình sẵn. Đối với mỗi thiết bị, bảng

cung cấp một cột cho kích thước đường chéo màn hình (**Size**), độ phân giải màn hình tính bằng pixel (**Resolution**) và mật độ điểm ảnh (**Density**).



3. Chọn một thiết bị như **Nexus 5X** hoặc **Pixel XL**, sau đó nhấp vào **Next**. Màn hình **System Image** sẽ xuất hiện.

4. Nhấp vào tab **Recommended** nếu nó chưa được chọn, sau đó chọn phiên bản hệ điều hành Android sẽ chạy trên thiết bị ảo (chẳng hạn như **Oreo**).



Có nhiều phiên bản hơn so với những phiên bản hiển thị trong tab **Recommended**. Hãy xem trong các tab **x86 Images** và **Other Images** để tìm thêm.

Nếu có liên kết **Download** bên cạnh một system image mà bạn muốn sử dụng, điều đó có nghĩa là nó chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống, sau đó nhấp **Finish** khi quá trình tải xuống hoàn tất.

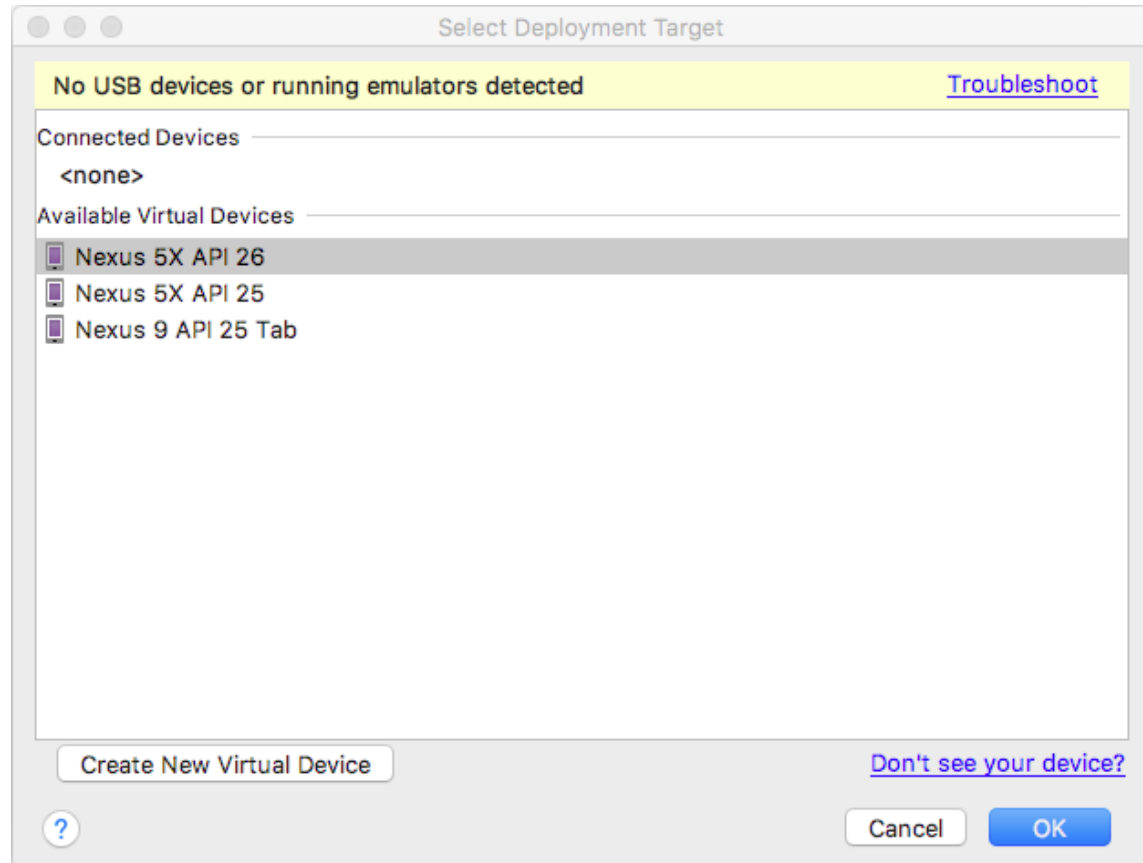
5. Sau khi chọn system image, nhấp vào Next. Cửa sổ Android Virtual Device (AVD) sẽ xuất hiện.

Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào Finish.

3.2 Chạy ứng dụng trên máy ảo

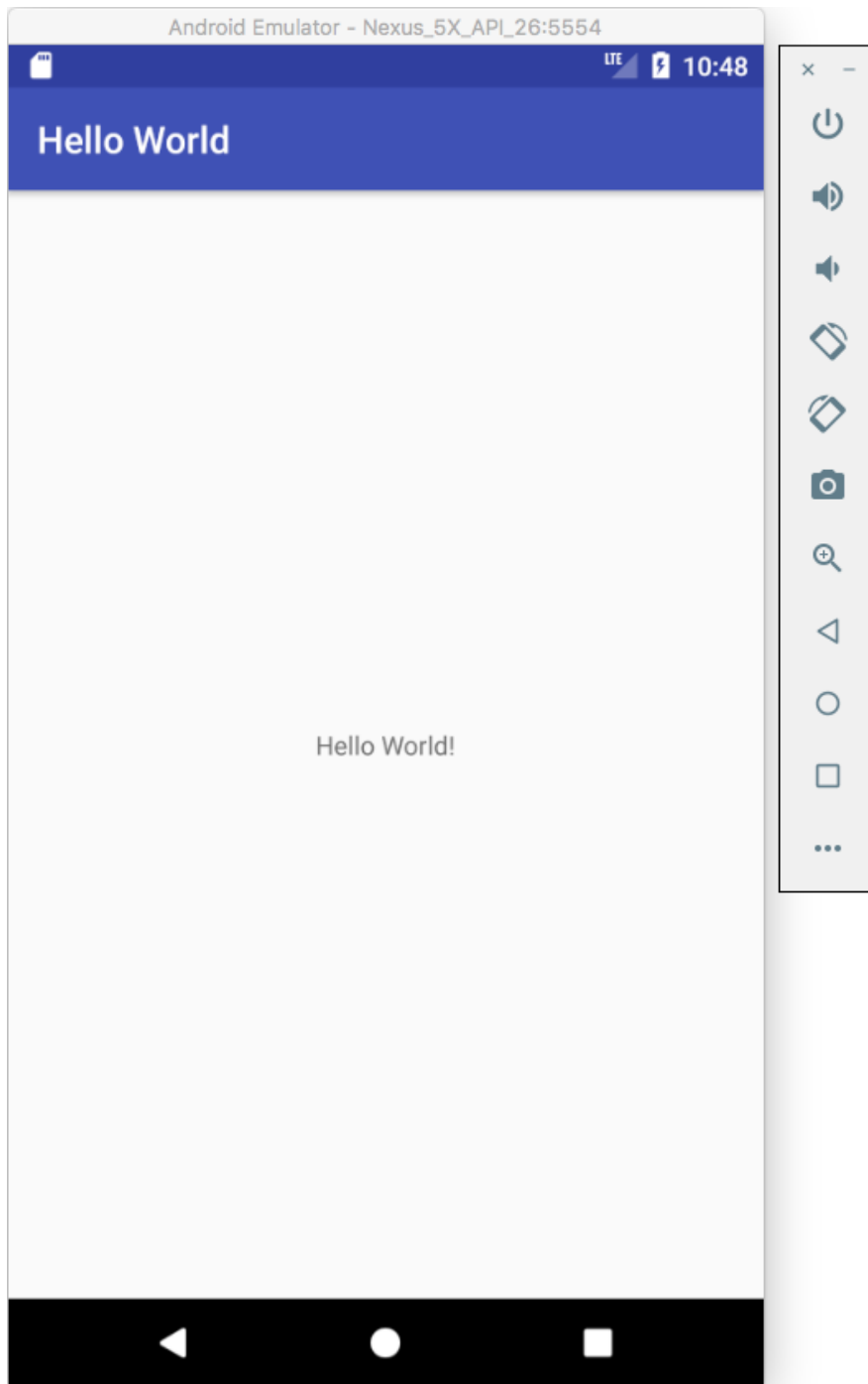
Trong nhiệm vụ này, cuối cùng bạn sẽ chạy ứng dụng Hello World của mình.

1. Trong Android Studio, chọn Chạy > Chạy ứng dụng hoặc nhấp vào biểu tượng Chạy trên thanh công cụ.
2. Trong cửa sổ Chọn mục tiêu triển khai, bên dưới Thiết bị ảo khả dụng, hãy chọn thiết bị ảo mà bạn vừa tạo và nhấp vào OK.



Trình giả lập khởi động và khởi động giống như một thiết bị vật lý. Tùy thuộc vào tốc độ máy tính của bạn, việc này có thể mất một lúc. Ứng dụng của bạn được xây dựng và khi trình giả lập đã sẵn sàng, Android Studio sẽ tải ứng dụng lên trình giả lập và chạy ứng dụng đó.

Bạn sẽ thấy ứng dụng Hello World như hình dưới đây.



Mẹo: Khi thử nghiệm trên thiết bị ảo, bạn nên khởi động thiết bị một lần, ngay từ đầu phiên làm việc. Bạn không nên đóng thiết bị cho đến khi hoàn tất thử nghiệm ứng dụng, để ứng dụng không phải trải qua quá trình khởi động thiết bị một lần nữa. Để đóng thiết bị ảo, hãy nhấp vào nút X ở đầu trình giả lập, chọn Thoát khỏi menu hoặc nhấn Control-Q trong Windows hoặc Command-Q trong macOS.

Nhiệm vụ 4: Sử dụng thiết bị vật lý (Tùy chọn)

Trong nhiệm vụ cuối cùng này, bạn sẽ chạy ứng dụng của mình trên thiết bị di động vật lý như điện thoại hoặc máy tính bảng. Bạn nên luôn kiểm tra ứng dụng của mình trên cả thiết bị ảo và vật lý.

Những gì bạn cần:

- Một thiết bị Android như điện thoại hoặc máy tính bảng.
- Cáp dữ liệu để kết nối thiết bị Android của bạn với máy tính qua cổng USB.
- Nếu bạn đang sử dụng hệ thống Linux hoặc Windows, bạn có thể cần thực hiện các bước bổ sung để chạy trên thiết bị phần cứng. Kiểm tra [Sử dụng thiết bị phần cứng tài liệu](#). Bạn cũng có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Đối với trình điều khiển USB chạy trên Windows, hãy xem [Trình điều khiển USB OEM](#).

4.1 Bật trình gỡ lỗi USB


Để Android Studio giao tiếp với thiết bị của bạn, bạn phải bật USB Debugging trên thiết bị Android của mình. Tính năng này được bật trong cài đặt **tùy chọn Nhà phát triển** của thiết bị.

Trên Android 4.2 trở lên, **màn hình Tùy chọn nhà phát triển** bị ẩn theo mặc định. Để hiển thị tùy chọn nhà phát triển và bật Gỡ lỗi USB:

1. Trên thiết bị của bạn, hãy mở **Cài đặt**, tìm kiếm **Giới thiệu về điện thoại**, nhấp vào Giới thiệu về điện thoại và chạm vào **Số bản dựng** bảy lần.
2. Quay lại màn hình trước đó (**Cài đặt / Hệ thống**). **Tùy chọn nhà phát triển** xuất hiện trong danh sách. Nhấn vào **Tùy chọn nhà phát triển**.
3. Chọn **Gỡ lỗi USB**.

4.2 Chạy ứng dụng của bạn trên thiết bị

Bây giờ bạn có thể kết nối thiết bị và chạy ứng dụng từ Android Studio.

1. Kết nối thiết bị của bạn với máy phát triển bằng cáp USB.
2. Nhấp vào nút **Chạy**  trên thanh công cụ. Cửa sổ **Chọn mục tiêu triển khai** mở ra với danh sách các trình giả lập và thiết bị được kết nối có sẵn.

3. Chọn thiết bị của bạn và nhấp vào **OK**.

Android Studio cài đặt và chạy ứng dụng trên thiết bị của bạn.

Xử lý sự cố

Nếu Android Studio không nhận ra thiết bị của bạn, hãy thử cách sau:

1. Rút phích cắm và cắm lại thiết bị.
2. Khởi động lại Android Studio.

Nếu máy tính của bạn vẫn không tìm thấy thiết bị hoặc tuyên bố thiết bị "không được phép", hãy làm theo các bước sau:

1. Rút phích cắm của thiết bị.
2. Trên thiết bị, mở Tùy chọn nhà phát triển trong ứng dụng Cài đặt.
3. Nhấn vào Thu hồi quyền gỡ lỗi USB.
4. Kết nối lại thiết bị với máy tính.
5. Khi được nhắc, hãy cấp quyền.

Bạn có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Xem [Sử dụng thiết bị phần cứng tài liệu](#).

Nhiệm vụ 5: Thay đổi cấu hình Gradle của ứng dụng

Trong nhiệm vụ này, bạn sẽ thay đổi một số thứ về cấu hình ứng dụng trong tệp `build.gradle(Module:app)` để tìm hiểu cách thực hiện thay đổi và đồng bộ hóa chúng với dự án Android Studio của bạn.

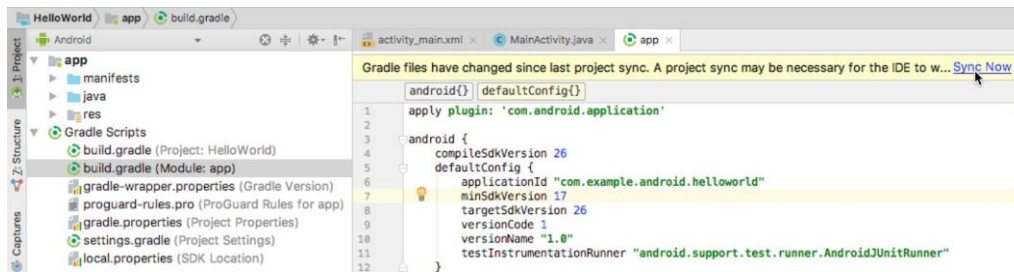
5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng

Thực hiện theo các bước sau:

1. Mở rộng thư mục Gradle Scripts nếu nó chưa được mở và nhấp đúp vào build.gradle(Module:app) tài liệu.

Nội dung của tệp sẽ xuất hiện trong trình soạn thảo mã.

2. Trong khối defaultConfig, hãy thay đổi giá trị của minSdkVersion thành 17 như hiển thị bên dưới (ban đầu giá trị này được đặt thành 15).



Trình chỉnh sửa mã hiển thị thanh thông báo ở trên cùng với liên kết Đồng bộ hóa ngay.

5.2 Đồng bộ cấu hình Gradle mới

Khi bạn thực hiện thay đổi đối với tệp cấu hình bản dựng trong một dự án, Android Studio yêu cầu bạn đồng bộ hóa các tệp dự án để có thể nhập các thay đổi cấu hình bản dựng và chạy một số kiểm tra để đảm bảo cấu hình sẽ không tạo ra lỗi bản dựng.

Để đồng bộ các tệp dự án, hãy nhấp vào **Đồng bộ ngay** trên thanh thông báo xuất hiện khi thực hiện thay đổi

(như được hiển thị trong hình trước đó) hoặc nhấp vào biểu tượng **Đồng bộ**

hóa dự án với biểu tượng Gradle  trên thanh công cụ.

Khi quá trình đồng bộ hóa Gradle hoàn tất, thông báo Gradle build finished sẽ xuất hiện ở góc dưới bên trái của cửa sổ Android Studio.

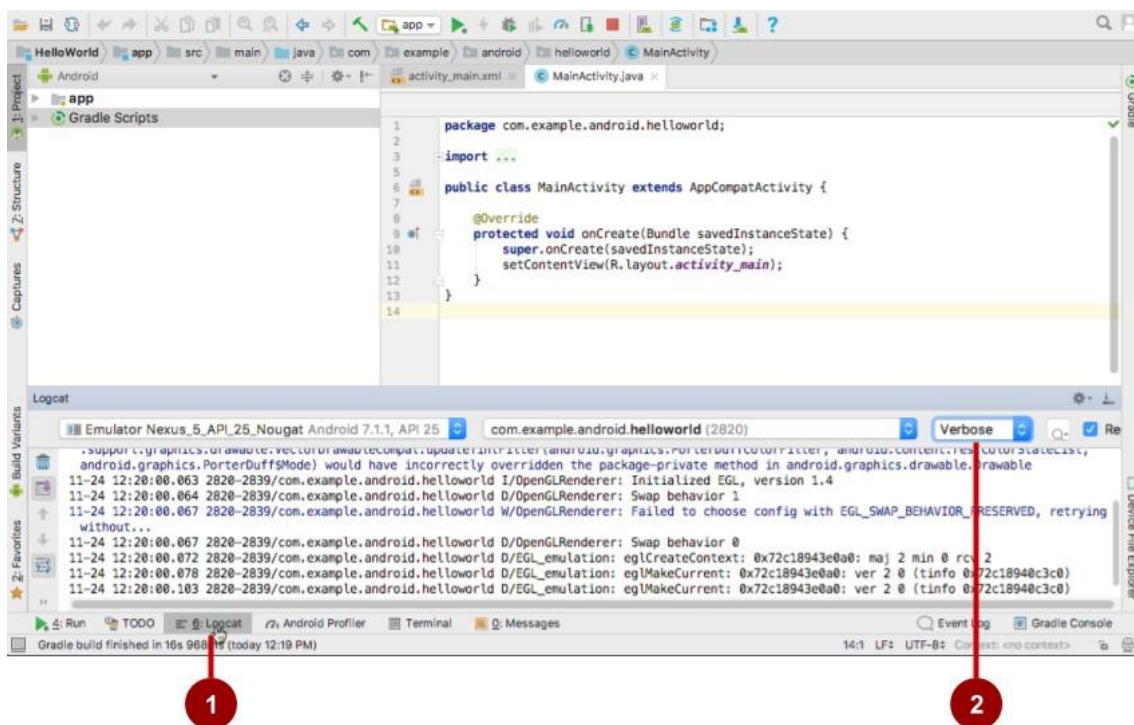
Để có cái nhìn sâu hơn về Gradle, hãy xem [Tổng quan về hệ thống xây dựng](#) Và [Cấu hình bản dựng Gradle](#) tài liệu.

Nhiệm vụ 6: Thêm câu lệnh nhật ký vào ứng dụng của bạn

Trong nhiệm vụ này, bạn sẽ thêm **Nhật ký** câu lệnh cho ứng dụng của bạn, hiển thị thông báo trong ngăn **Logcat**. Thông báo nhật ký là một công cụ gỡ lỗi mạnh mẽ mà bạn có thể sử dụng để kiểm tra giá trị, đường dẫn thực thi và báo cáo ngoại lệ.

6.1 Xem ngăn Logcat

Để xem ngăn Logcat, hãy nhấp vào tab Logcat ở cuối cửa sổ Android Studio như minh họa trong hình bên dưới.



Trong hình trên:

1. Tab **Logcat** để mở và đóng ngăn **Logcat**, hiển thị thông tin về ứng dụng của bạn khi nó đang chạy. Nếu bạn thêm câu lệnh Nhật ký vào ứng dụng của mình, thông báo Nhật ký sẽ xuất hiện ở đây.
2. Menu cấp độ Nhật ký được đặt thành Chi tiết (mặc định), hiển thị tất cả các thông báo Nhật ký. Cài đặt khác bao gồm **Gỡ lỗi**, **Lỗi**, **Thông tin** và **cảnh báo**.

6.2 Thêm các câu lệnh nhật ký vào ứng dụng của bạn

Các câu lệnh nhật ký trong mã ứng dụng của bạn hiển thị thông báo trong ngăn Logcat.
Ví dụ:

```
Log.d("MainActivity", "Hello World");
```

Các phần của tin nhắn bao gồm:

- Nhật ký: **Nhật ký** lớp để gửi tin nhắn nhật ký đến ngăn Logcat.
- d: Các **Gỡ lỗi** Thiết lập mức nhật ký để lọc hiển thị thông báo nhật ký trong ngăn Logcat. Các mức nhật ký khác được e cho **Lỗi**, w là **Cảnh báo** và i là **Thông tin**.
- "MainActivity": Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong Bảng Logcat. Đây thường là tên của Hoạt động mà từ đó tin nhắn có nguồn gốc. Tuy nhiên, bạn có thể biến điều này thành bất cứ điều gì hữu ích cho bạn để gỡ lỗi.

Theo quy ước, thẻ nhật ký được định nghĩa là hằng số cho Hoạt động:
`private static final String LOG_TAG = MainActivity.class.getSimpleName();`

- "Hello world": Đối số thứ hai là thông điệp thực tế.

Thực hiện theo các bước sau:

- 6 Mở ứng dụng Hello World trong Android studio và mở MainActivity.
- 7 Để tự động thêm các lệnh nhập rõ ràng vào dự án của bạn (chẳng hạn như `android.util.Log` là bắt buộc để sử dụng `Log`), hãy chọn **File** > **Settings** trong Windows hoặc **Android Studio** > **Preferences** trong macOS.
- 8 Chọn **Editor** > **General** > **Auto Import**. Chọn tất cả các hộp kiểm và **Insert imports on paste** thành **All**.
- 9 Nhấp vào **Áp dụng** rồi nhấp vào **OK**.
- 10 Trong phương thức `onCreate()` của MainActivity, hãy thêm câu lệnh sau:

```
Log.d("MainActivity", "Hello World");
```

Phương thức `onCreate()` bây giờ sẽ trông giống như đoạn mã sau:

```
@Override
```

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

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

```
setContentView(R.layout.activity_main);  
  
Log.d("MainActivity", "Hello World");  
  
}
```

1. Nếu ngăn **Logcat** chưa mở, hãy nhấp vào tab **Logcat** ở cuối Android Studio để mở.
2. Kiểm tra xem tên mục tiêu và tên gói của ứng dụng đã chính xác chưa.
3. Thay đổi mức Nhật ký trong ngăn **Logcat** thành **Gỡ lỗi** (hoặc giữ nguyên mức **Chi tiết** vì có rất ít thông báo nhật ký).
4. Chạy ứng dụng của bạn.

Thông báo sau sẽ xuất hiện trong ngăn Logcat:

```
11-24 14:06:59.001 4696-4696/? D/MainActivity: Hello World
```

Thử thách mã hóa

Ghi chú: Mọi thử thách lập trình đều là tùy chọn và không phải là điều kiện tiên quyết cho các bài học sau.

Thử thách: Bây giờ bạn đã thiết lập và quen thuộc với quy trình phát triển cơ bản, hãy thực hiện như sau:

1. Tạo một dự án mới trong Android Studio.
2. Đổi lời chào "Hello World" thành "Chúc mừng sinh nhật " và tên của một người nào đó có ngày sinh nhật gần đây.
3. (Tùy chọn) Chụp ảnh màn hình ứng dụng đã hoàn thành của bạn và gửi qua email cho người có ngày sinh nhật bạn quên rồi.
4. Một cách sử dụng phổ biến của [Nhật ký lớp](#) học là để ghi nhật ký [Ngoại lệ Java](#) khi chúng xảy ra trong chương trình của bạn. Có một số phương pháp hữu ích, chẳng hạn như [log.e\(\)](#), mà bạn có thể sử dụng cho mục đích này. Khám phá phương pháp bạn có thể sử dụng để bao gồm ngoại lệ với thông báo Nhật ký. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại ngoại lệ.

Bản tóm tắt

- Để cài đặt Android Studio, hãy truy cập [Studio Android](#) và làm theo hướng dẫn để tải xuống và cài đặt.
- Khi tạo ứng dụng mới, hãy đảm bảo rằng API 15:Android 4.0.3 IceCreamSandwich được đặt thành SDK tối thiểu.
- Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy nhấp vào tab Dự án trong cột tab dọc, sau đó chọn Android trong menu bật lên ở trên cùng.
- Chỉnh sửa tệp build.gradle(Module:app) khi bạn cần thêm thư viện mới vào dự án hoặc thay đổi phiên bản thư viện.
- Tất cả mã và tài nguyên cho ứng dụng đều nằm trong thư mục app và res. Javathư mục bao gồm các hoạt động, bài kiểm tra và các thành phần khác trong mã nguồn Java. Thư mục res chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh.
- Chỉnh sửa tệp AndroidManifest.xml để thêm các thành phần tính năng và quyền vào ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML này.
- Sử dụng [Trình quản lý thiết bị ảo Android \(AVD\)](#) để tạo một thiết bị ảo (còn gọi là trình giả lập) để chạy ứng dụng của bạn.
- Thêm vào [Nhật ký](#) các câu lệnh cho ứng dụng của bạn, hiển thị thông báo trong ngăn Logcat như một công cụ cơ bản để gỡ lỗi.
- Để chạy ứng dụng của bạn trên thiết bị Android vật lý bằng Android Studio, hãy bật USB Debugging trên thiết bị. Mở Cài đặt > Giới thiệu về điện thoại và chạm vào Build number bảy lần. Quay lại màn hình trước đó (Cài đặt) và chạm vào Developer options. Chọn USB Debugging.

Các khái niệm liên quan

Tài liệu khái niệm liên quan có trong [1.0: Giới thiệu về Android](#) Và [1.1 Ứng dụng Android đầu tiên của bạn](#)

Tìm hiểu thêm

Tài liệu Android Studio:

- [Trang tải xuống Android Studio](#)
- [Ghi chú phát hành Android Studio](#)
- [Làm quen với Android Studio](#)
- [Dòng lệnh Logcat cũng vậy](#)
- [Trình quản lý thiết bị ảo Android \(AVD\)](#)
- [Tổng quan về App Manifest](#)
- [Cấu hình bản dựng của bạn](#)
- [Nhật ký lớp học](#)
- [Tạo và quản lý thiết bị](#)

- [Làm thế nào để cài đặt Java?](#)
- [Cài đặt phần mềm JDK và thiết lập JAVA_HOME](#)
- [Trang web Gradle](#)
- [Cú pháp Apache Groovy](#)
- [Trang Wikipedia Gradle](#)

Bài tập về nhà

Xây dựng và chạy một ứng dụng

- Tạo một dự án Android mới từ Mẫu trống.
- Thêm các câu lệnh ghi nhật ký cho nhiều cấp độ nhật ký khác nhau trong onCreate() trong hoạt động chính.
- Tạo trình giả lập cho thiết bị, nhắm tới bất kỳ phiên bản Android nào bạn thích và chạy ứng dụng.
- Sử dụng tính năng lọc trong Logcat để tìm các câu lệnh nhật ký và điều chỉnh các cấp độ để chỉ hiển thị các câu lệnh ghi nhật ký gỡ lỗi hoặc lỗi.

Trả lời những câu hỏi

Câu hỏi 1

Tên của tệp bố cục cho hoạt động chính là gì?

- MainActivity.java
- AndroidManifest.xml
- hoạt động_main.xml
- xây dựng.gradle

Câu hỏi 2

- Tên của chuỗi tài nguyên chỉ định tên ứng dụng là gì?
- tên_ứng_dụng
- xmlns:ứng_dụng
- android:tên
- ứng_dụngId

Câu hỏi 3

Bạn sử dụng công cụ nào để tạo trình giả lập mới?

- Giám sát thiết bị Android
- Quản lý AVD
- Trình quản lý SDK
- Biên tập chủ đề

Câu hỏi 4

Giả sử ứng dụng của bạn bao gồm câu lệnh ghi nhật ký này:

```
Log.i("MainActivity", "MainActivity layout is complete");
```

Bạn thấy thông báo "Bố cục MainActivity đã hoàn tất" trong ngăn Logcat nếu menu Cấp độ nhật ký được đặt thành tùy chọn nào sau đây? (Gợi ý: có thể trả lời nhiều câu hỏi.)

- Dài dòng
- Gỡ lỗi
- Thông tin
- Cảnh báo
- Lỗi
- khẳng định

Gửi ứng dụng của bạn để chấm điểm

Kiểm tra để đảm bảo ứng dụng có những điều sau:

- Một hoạt động hiển thị "Hello World" trên màn hình.
- Ghi lại các câu lệnh trong onCreate() trong hoạt động chính.
- Mức nhật ký trong ngăn Logcat chỉ hiển thị các câu lệnh ghi nhật ký gỡ lỗi hoặc lỗi.

1.2) Phần A: giao diện người dùng tương tác đầu tiên của bạn

Giới thiệu

Giao diện người dùng (UI) xuất hiện trên màn hình của thiết bị Android bao gồm một hệ thống phân cấp các đối tượng được gọi là chế độ xem — mọi thành phần của màn hình là một [View](#). Lớp View biểu thị khối xây dựng cơ bản cho tất cả các thành phần UI và là lớp cơ sở cho các lớp cung cấp các thành phần UI tương tác như nút, hộp kiểm và trường nhập văn bản. Các lớp con View thường được sử dụng được mô tả trong nhiều bài học bao gồm:

- [TextView](#) để hiển thị văn bản.
- [EditText](#) cho phép người dùng nhập và chỉnh sửa văn bản.
- [Button](#) và các yếu tố có thể nhấp khác (chẳng hạn như [RadioButton](#), [CheckBox](#), và [Spinner](#)) để cung cấp hành vi tương tác.
- [ScrollView](#) và Chế độ [RecyclerView](#) để hiển thị các mục có thể cuộn.
- [ImageView](#) để hiển thị hình ảnh.
- [ConstraintLayout](#) và [LinearLayout](#) để chứa các thành phần View khác và định vị chúng.

Mã Java hiển thị và điều khiển UI được chứa trong một lớp mở rộng [Activity](#). Một Activity thường được liên kết với một bố cục của các chế độ xem UI được định nghĩa là một tệp XML (eXtended Markup Language). Tệp XML này thường được đặt tên theo Activity của nó và định nghĩa bố cục của các thành phần View trên màn hình.

Ví dụ, mã MainActivity trong ứng dụng Hello World hiển thị một bố cục được xác định trong Tệp bố cục activity_main.xml, bao gồm một TextView có nội dung "Hello World".

Trong các ứng dụng phức tạp hơn, một Activity có thể triển khai các hành động để phản hồi thao tác chạm của người dùng, vẽ nội dung đồ họa hoặc yêu cầu dữ liệu từ cơ sở dữ liệu hoặc internet. Bạn sẽ tìm hiểu thêm về lớp Activity trong bài học khác.

Trong phần thực hành này, bạn sẽ học cách tạo ứng dụng tương tác đầu tiên của mình—một ứng dụng cho phép người dùng tương tác. Bạn sẽ tạo ứng dụng bằng mẫu Empty Activity. Bạn cũng sẽ học cách sử dụng trình chỉnh sửa bố cục để thiết kế bố cục và cách chỉnh sửa bố cục trong XML. Bạn cần phát triển các kỹ năng này để có thể hoàn thành các phần thực hành khác trong khóa học này.

Những điều bạn nên biết

Bạn nên biết về:

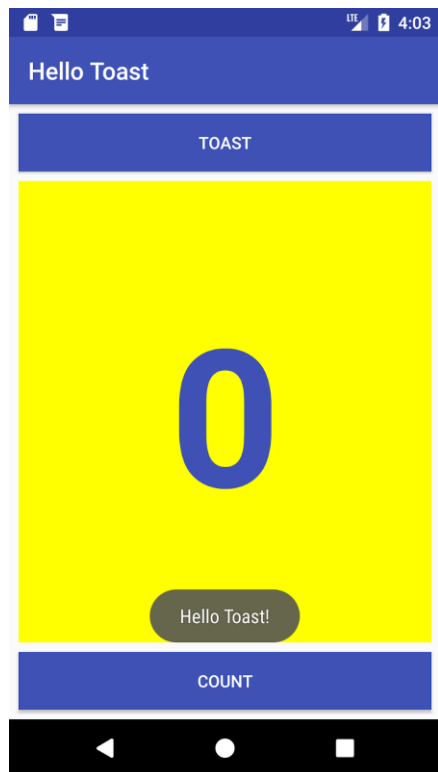
- Cách cài đặt và mở Android Studio.
- Cách tạo ứng dụng HelloWorld.
- Cách chạy ứng dụng HelloWorld.
-

Những gì bạn sẽ học được

- Cách tạo ứng dụng có tính tương tác.
- Cách sử dụng trình chỉnh sửa bố cục để thiết kế bố cục.
- Cách chỉnh sửa bố cục trong XML.
- Rất nhiều thuật ngữ mới. Kiểm tra Từ vựng và khái niệm thuật ngữ để có định nghĩa thân thiện.

Bạn sẽ làm gì

- Tạo một ứng dụng và thêm hai phần tử Button và một TextView vào bố cục.
- Thao tác từng phần tử trong Bố cục ràng buộc để giới hạn chúng ở phần lề và các yếu tố khác.
- Thay đổi thuộc tính của phần tử UI.
- Chỉnh sửa bố cục của ứng dụng trong XML.
- Trích xuất các chuỗi được mã hóa cứng thành các tài nguyên chuỗi.
- Triển khai các phương thức xử lý nhấp chuột để hiển thị thông báo trên màn hình khi người dùng chạm vào từng Nút



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