

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.....	Error! Bookmark not defined.
Bài 1)	Tạo ứng dụng đầu tiên	Error! Bookmark not defined.
1.1)	Android Studio và Hello World	Error! Bookmark not defined.
1.2)	Giao diện người dùng tương tác đầu tiên ...	Error! Bookmark not defined.
1.3)	Trình chỉnh sửa bố cục	Error! Bookmark not defined.
1.4)	Văn bản và các chế độ cuộn	Error! Bookmark not defined.
1.5)	Tài nguyên có sẵn.....	Error! Bookmark not defined.
Bài 2)	Activities	Error! Bookmark not defined.
2.1)	Activity và Intent	Error! Bookmark not defined.
2.2)	Vòng đời của Activity và trạng thái	Error! Bookmark not defined.
2.3)	Intent ngầm định.....	Error! Bookmark not defined.
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ.	Error! Bookmark not defined.
3.1)	Trình gỡ lỗi	Error! Bookmark not defined.
3.2)	Kiểm thử đơn vị.....	Error! Bookmark not defined.
3.3)	Thư viện hỗ trợ.....	Error! Bookmark not defined.
CHƯƠNG 2.	Trải nghiệm người dùng	Error! Bookmark not defined.
Bài 1)	Tương tác người dùng	Error! Bookmark not defined.
1.1)	Hình ảnh có thể chọn	Error! Bookmark not defined.
1.2)	Các điều khiển nhập liệu	Error! Bookmark not defined.
1.3)	Menu và bộ chọn	Error! Bookmark not defined.
1.4)	Điều hướng người dùng	Error! Bookmark not defined.
1.5)	RecyclerView	Error! Bookmark not defined.
Bài 2)	Trải nghiệm người dùng thú vị.....	Error! Bookmark not defined.
2.1)	Hình vẽ, định kiểu và chủ đề	Error! Bookmark not defined.
2.2)	Thẻ và màu sắc	Error! Bookmark not defined.

2.3)	Bố cục thích ứng.....	Error! Bookmark not defined.
Bài 3)	Kiểm thử giao diện người dùng.....	Error! Bookmark not defined.
3.1)	Espresso cho việc kiểm tra UI	Error! Bookmark not defined.
CHƯƠNG 3. Làm việc trong nền		Error! Bookmark not defined.
Bài 1)	Các tác vụ nền.....	Error! Bookmark not defined.
1.1)	AsyncTask	Error! Bookmark not defined.
1.2)	AsyncTask và AsyncTaskLoader	Error! Bookmark not defined.
1.3)	Broadcast receivers	Error! Bookmark not defined.
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền	Error! Bookmark not defined.
2.1)	Thông báo	Error! Bookmark not defined.
2.2)	Trình quản lý cảnh báo	Error! Bookmark not defined.
2.3)	JobScheduler.....	Error! Bookmark not defined.
CHƯƠNG 4. Lưu dữ liệu người dùng		Error! Bookmark not defined.
Bài 1)	Tùy chọn và cài đặt.....	Error! Bookmark not defined.
1.1)	Shared preferences.....	Error! Bookmark not defined.
1.2)	Cài đặt ứng dụng.....	Error! Bookmark not defined.
Bài 2)	Lưu trữ dữ liệu với Room	Error! Bookmark not defined.
2.1)	Room, LiveData và ViewModel.....	Error! Bookmark not defined.
2.2)	Room, LiveData và ViewModel.....	Error! Bookmark not defined.

Khóa học Lập trình viên Android Cơ bản (Phiên bản 2)

Cập nhật lần cuối vào Thứ 3 ngày 11 tháng 9 năm 2018

Khóa học này được làm ra bởi đội ngũ đào tạo phát triển Google

Để biết về nội dung của khóa học, bao gồm các liên kết đến tất cả các khái niệm chương , các ứng dụng và bài giảng , hãy xem

[AndroidDeveloperFundamentals\(Version2\).
developer.android.com/courses/adfv2.](https://developer.android.com/courses/adfv2)

Lưu ý: Khóa học này sử dụng nhiều thuật ngữ "codelab" và "practical" thay thế lẫn nhau

Chúng tôi khuyên bạn nên sử dụng phiên bản trực tuyến của khóa học này thay vì bản PDF để đảm bảo bạn đang sử dụng nội dung mới nhất.

Vào developer.android.com/courses/adf-v2.

Chương 1 : Mở đầu

Bản Pdf này chứa ảnh chụp về các bài học trong Bài 1 Mở đầu.

Các bài học trong chương này .

Bài 1: Xây dựng ứng dụng đầu tiên của bạn

1.1: Android Studio and Hello World

1.2A: Giao diện tương tác đầu tiên với người dùng của bạn.

1.2B: Chỉnh sửa bố cục

1.3: Cuộn và xem văn bản

1.4: Tài nguyên có sẵn

Bài 2 : Hoạt động

2.1 : Hoạt động và mong muốn

2.2: Trạng thái và vòng hoạt động

2.3: Intent ẩn

Bài 3 : Kiểm tra , Sửa lỗi , và sử dụng thư viện hỗ trợ

3.1: Sửa lỗi

3.2: Kiểm tra

3.3: Thư viện hỗ trợ

Bài 1.1 : Android Studio và Hello World

Giới thiệu

Trong bài thực hành này, bạn sẽ học 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 trình giả lập và trên thiết bị thực.

Những gì bạn cần biết trước

Bạn nên có khả năng:

Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng hướng đối tượng bằng IDE (môi trường phát triển tích hợp) như Android Studio.

Chứng minh rằng bạn có từ **1-3 năm kinh nghiệm** lập trình hướng đối tượng, trong đó có một phần tập trung vào ngôn ngữ lập trình **Java**. (Các bài thực hành này sẽ không giải thích lập trình hướng đối tượng hoặc ngôn ngữ Java.)

Những gì bạn cần chuẩn bị

Một máy tính chạy **Windows**, **Linux**, hoặc **macOS** (Mac). Xem trang tải xuống Android Studio để biết yêu cầu hệ thống mới nhất.

Kết nối Internet hoặc một phương thức khác để tải Android Studio và Java phiên bản mới nhất về máy tính của bạn.

Tài liệu này được cấp phép theo **Giấy phép Creative Commons Attribution 4.0 International**.

Những gì bạn sẽ học

- Cách cài đặt và sử dụng **Android Studio IDE**.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ mẫu có sẵn.
- Cách thêm **log messages** vào ứng dụng để phục vụ quá trình gỡ lỗi.

Những gì bạn sẽ làm

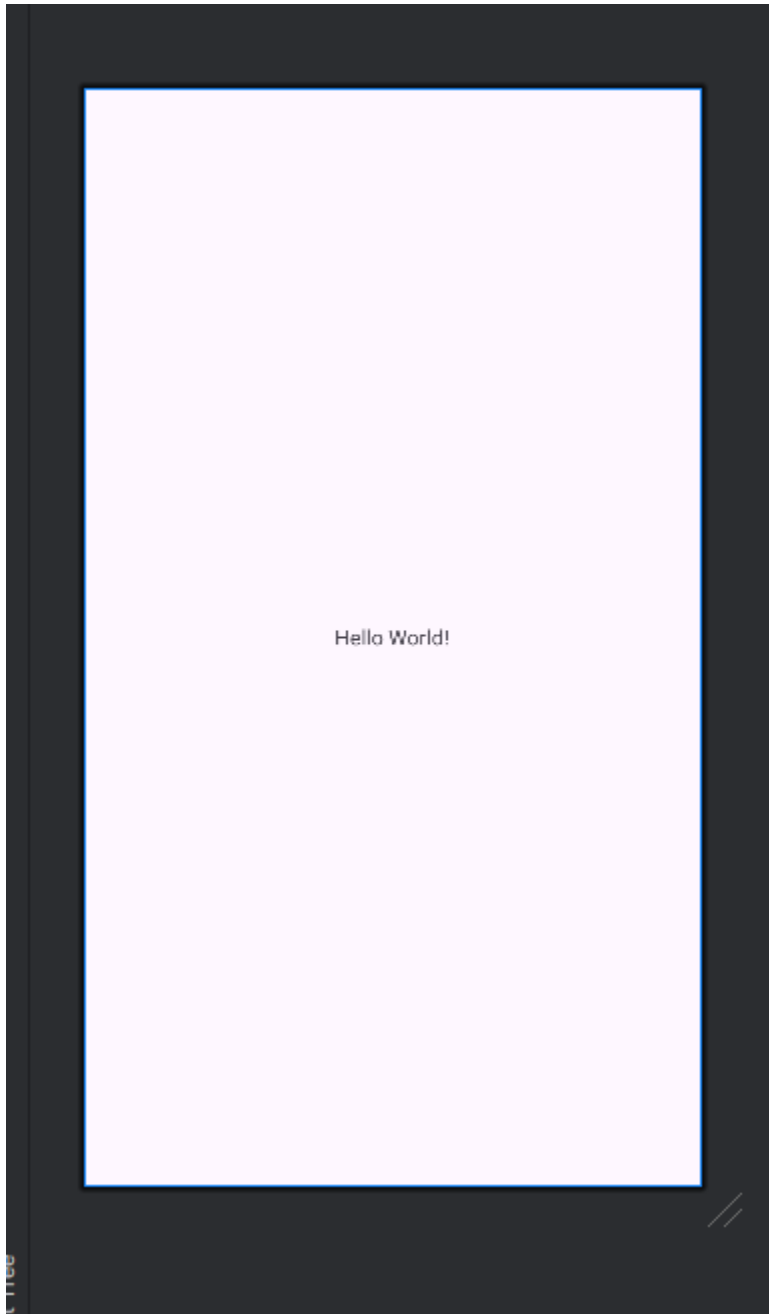
- Cài đặt môi trường phát triển **Android Studio**.
- Tạo một **trình giả lập** (thiết bị ảo) để chạy ứng dụng trên máy tính.
- Tạo và chạy ứng dụng **Hello World** trên cả thiết bị ảo và thiết bị thực.
- Khám phá **cấu trúc dự án** trong Android Studio.
- Tạo và xem **log messages** từ ứng dụng của bạn.
- Tìm hiểu về tệp **AndroidManifest.xml**.

Tổng quan về ứng dụng

Sau khi bạn cài đặt thành công Android Studio, bạn sẽ tạo một dự án mới từ mẫu cho ứng dụng **Hello World**. Ứng dụng đơn giản này sẽ

hiển thị chuỗi **"Hello World"** trên màn hình của thiết bị Android ảo hoặc thực.

Đây là giao diện của ứng dụng sau 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 **trình soạn thảo mã nâng cao** và một loạt **mẫu ứng dụng**. Ngoài ra, nó còn tích hợp các công cụ hỗ trợ **phát triển, gỡ lỗi, kiểm thử và tối ưu hiệu suất**, giúp việc phát triển ứng dụng trở nên nhanh hơn và dễ dàng hơn.

Bạn có thể kiểm thử ứng dụng của mình trên nhiều **trình giả lập được cấu hình sẵn** hoặc trên **thiết bị di động thực tế**, xây dựng ứng dụng để phát hành, và đưa ứng dụng lên **Google Play Store**.

📌 **Lưu ý:** *Android Studio liên tục được cập nhật và cải tiến. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, vui lòng truy cập trang web chính thức của Android Studio.*

+Android Studio có sẵn cho các hệ điều hành sau:

- **Windows, Linux** trên máy tính.
- **macOS** trên máy Mac.

Phiên bản mới nhất của **OpenJDK (Java Development Kit)** đã được tích hợp sẵn trong **Android Studio**.

• Các bước cài đặt Android Studio

1. **Truy cập trang web của nhà phát triển Android** và làm theo hướng dẫn để tải xuống và cài đặt **Android Studio**.
2. **Chấp nhận các cấu hình mặc định** trong tất cả các bước và đảm bảo rằng **tất cả các thành phần** cần thiết đều được chọn để cài đặt.
3. **Sau khi cài đặt xong**, trình hướng dẫn cài đặt (**Setup Wizard**) sẽ tự động tải xuống và cài đặt **các thành phần bổ sung**, bao gồm **Android SDK**.
 - Quá trình này có thể mất một khoảng thời gian tùy thuộc vào **tốc độ Internet** của bạn.

- Một số bước có thể trông giống như bị lặp lại, nhưng hãy kiên nhẫn chờ đợi.
4. **Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động, và bạn đã sẵn sàng tạo dự án đầu tiên của mình!**

- **Khắc phục sự cố**

Nếu gặp vấn đề trong quá trình cài đặt, bạn có thể:

Kiểm tra **Ghi chú phát hành (release notes)** của **Android Studio**.

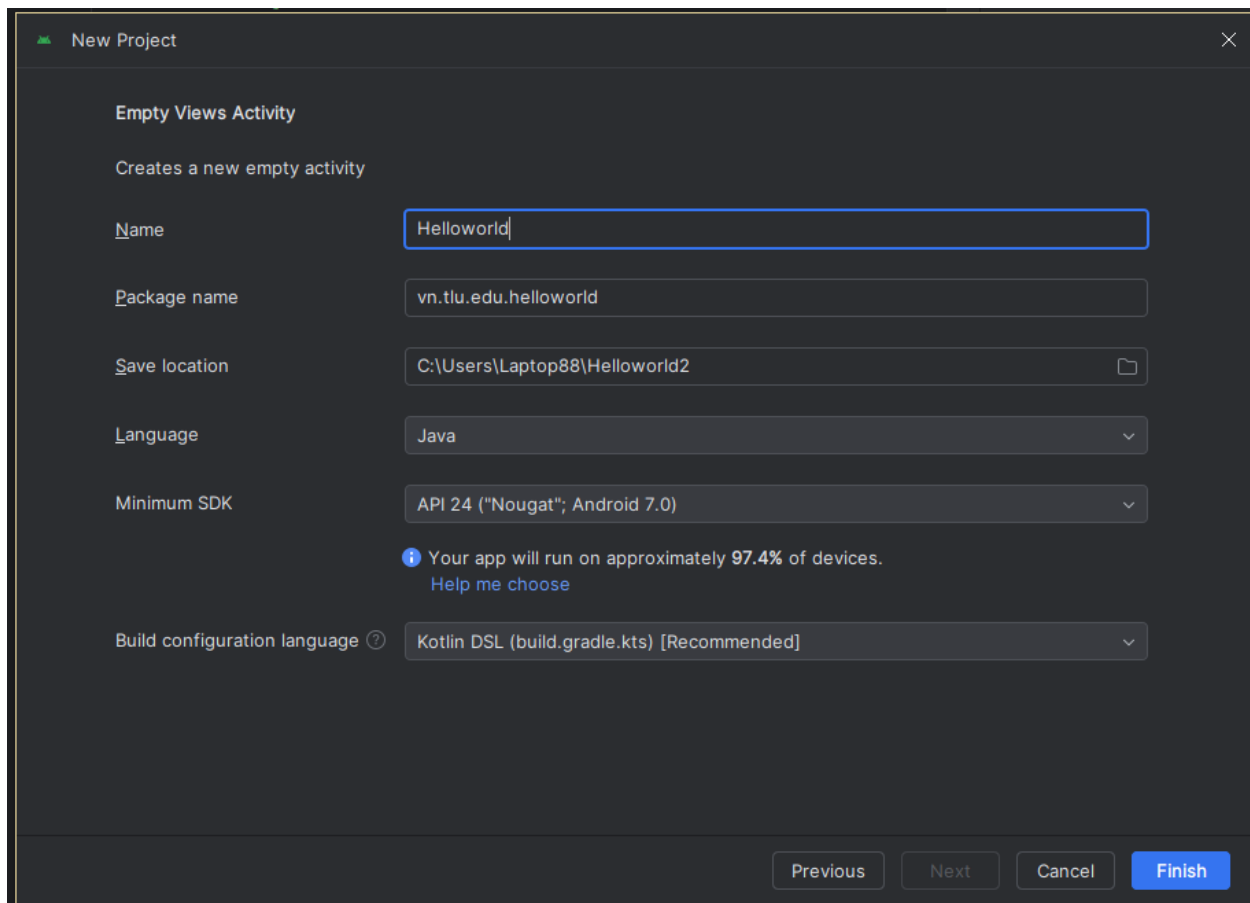
Hoặc Nhờ **giảng viên hoặc cộng đồng hỗ trợ** để được giúp đỡ.

Nhiệm vụ 2: Tạo ứng dụng Hello World

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

2.1 Tạo dự án ứng dụng

1. Mở **Android Studio** nếu chưa mở.
2. Trong cửa sổ chính **Welcome to Android Studio**, nhấp vào **Start a new Android Studio project**.
3. Trong cửa sổ **Create Android Project**, nhập **Hello World** vào ô **Application name**.



4. Xác nhận rằng vị trí mặc định của **Project location** là nơi bạn muốn lưu trữ ứng dụng **Hello World** và các dự án khác trong **Android Studio**, hoặc thay đổi sang thư mục bạn muốn.

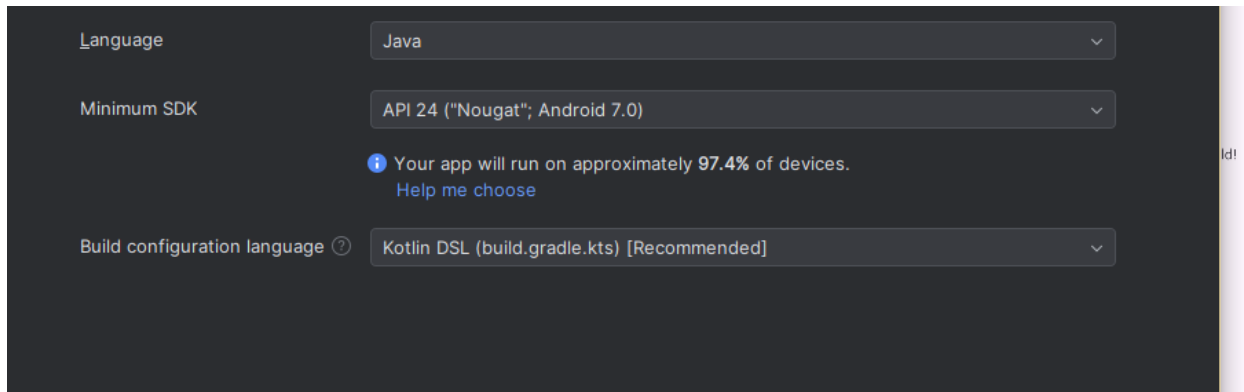
5. Chấp nhận giá trị mặc định **android.example.com** cho **Company Domain**, hoặc tạo một tên miền riêng.
Nếu bạn không có kế hoạch xuất bản ứng dụng, bạn có thể giữ nguyên giá trị mặc định. Lưu ý rằng việc thay đổi **package name** sau này sẽ tốn thêm công sức.

6. Bỏ chọn các tùy chọn **Include C++ support** và **Include Kotlin support**, sau đó nhấp vào **Next**.

7. Trong màn hình **Target Android Devices**, đảm bảo rằng **Phone and Tablet** được chọn.

Đặt API 15: Android 4.0.3 IceCreamSandwich làm Minimum SDK.

Nếu chưa đúng, hãy sử dụng menu thả xuống để chọn.



Đây là các cài đặt được sử dụng trong các ví dụ của bài học trong khóa học này. Tính đến thời điểm viết tài liệu này, các cài đặt này giúp ứng dụng **Hello World** của bạn tương thích với **97%** thiết bị Android đang hoạt động trên **Google Play Store**.

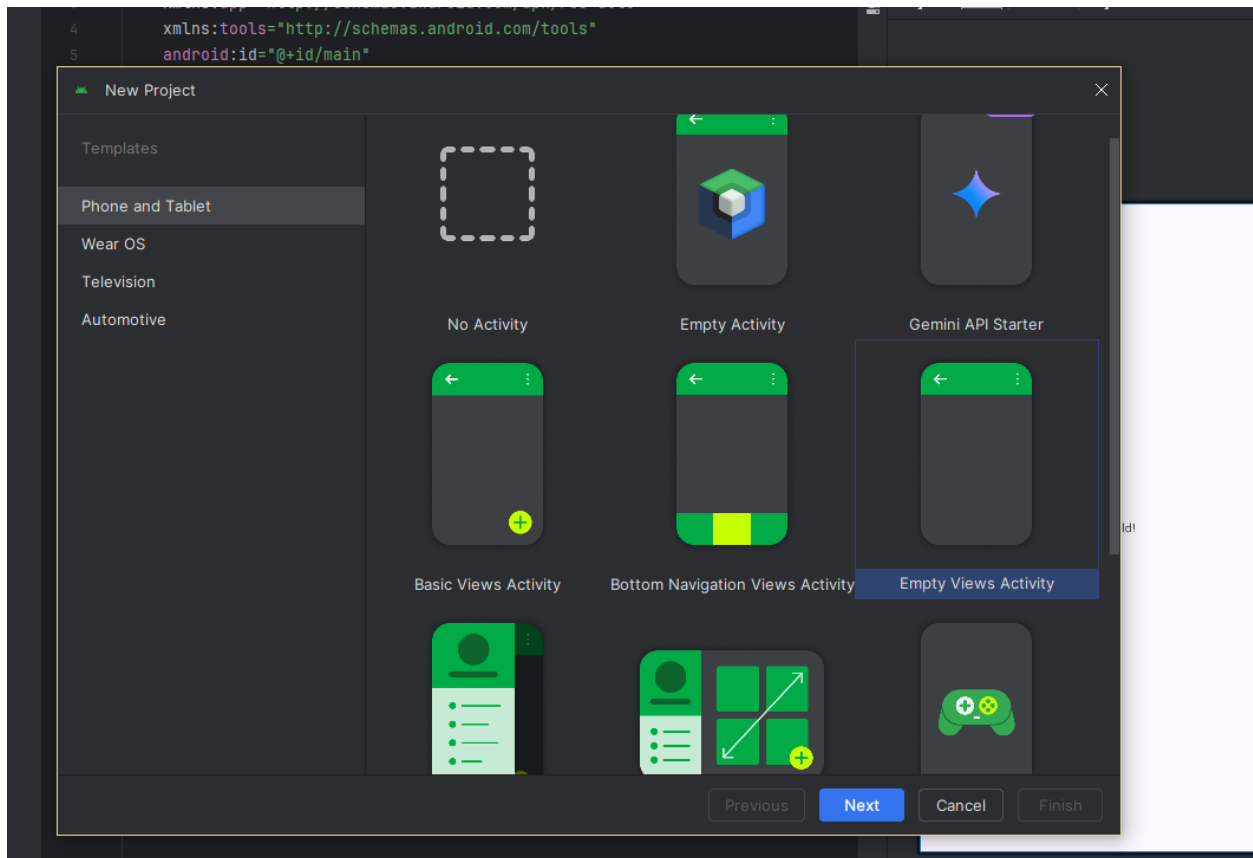
8. Bỏ chọn tùy chọn **Include Instant App support** và tắt cả các tùy chọn khác, sau đó nhấp vào **Next**.

Nếu dự án của bạn yêu cầu các thành phần bổ sung cho **target SDK** đã chọn, **Android Studio** sẽ tự động cài đặt chúng.

9. Cửa sổ **Add an Activity** sẽ xuất hiện. **Activity** là một thành phần quan trọng trong bất kỳ ứng dụng Android nào, đại diện cho một tác vụ cụ thể mà người dùng có thể thực hiện.

Một **Activity** thường có một **layout** đi kèm để xác định cách các phần tử giao diện (**UI elements**) xuất hiện trên màn hình. **Android Studio** cung cấp các mẫu **Activity** để giúp bạn bắt đầu nhanh chóng.

Đối với dự án **Hello World**, hãy chọn **Empty Activity** như hình bên dưới, sau đó nhấp vào **Next**.



10. Màn hình **Configure Activity** sẽ xuất hiện (giao diện này có thể khác nhau tùy thuộc vào mẫu **Activity** bạn đã chọn ở bước trước).

Theo mặc định, **Activity** trống do mẫu cung cấp sẽ có tên là **MainActivity**. Bạn có thể thay đổi tên nếu muốn, nhưng bài học này sẽ sử dụng **MainActivity**.

11. Đảm bảo rằng tùy chọn **Generate Layout file** được chọn. Theo mặc định, tên **layout** sẽ là **activity_main**. Bạn có thể thay đổi nếu muốn, nhưng bài học này sẽ sử dụng **activity_main**.

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

13. Nhấp vào **Finish**.

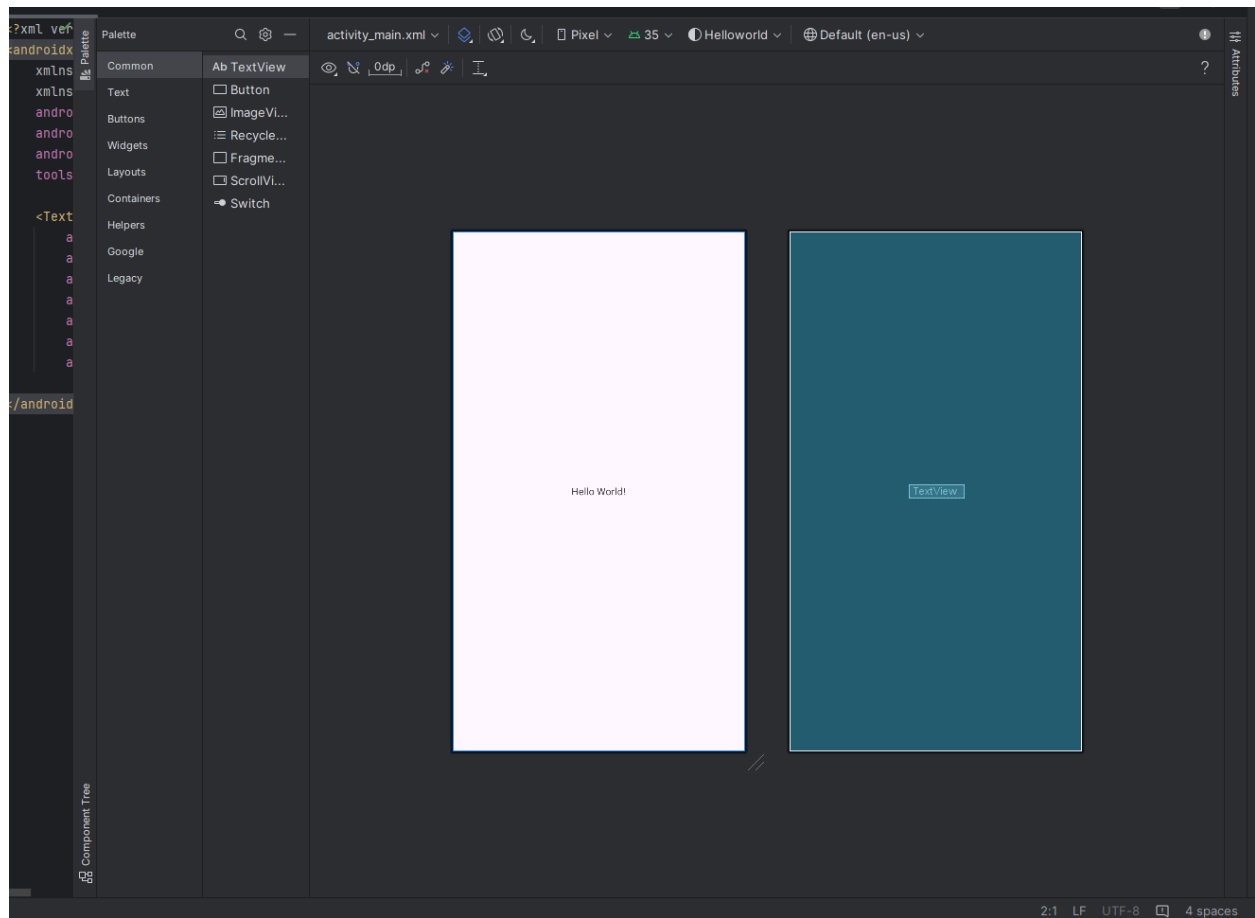
Android Studio sẽ tạo một thư mục cho các dự án của bạn và tiến hành **build** dự án bằng **Gradle** (quá trình này có thể mất vài phút).

Mẹo: Xem trang **Configure your build** dành cho nhà phát triển để biết thông tin chi tiết.

Bạn cũng có thể thấy cửa sổ "**Tip of the Day**" hiển thị các phím tắt và mẹo hữu ích khác. Nhấp vào **Close** để đóng thông báo này.

Sau đó, trình chỉnh sửa **Android Studio** sẽ xuất hiện. Hãy thực hiện các bước sau:

1. Nhấp vào tab **activity_main.xml** để mở trình chỉnh sửa **layout**.
2. Nhấp vào tab **Design** trong trình chỉnh sửa **layout** (nếu chưa được chọn) để hiển thị bản xem trước đồ họa của giao diện, như hình dưới đây.



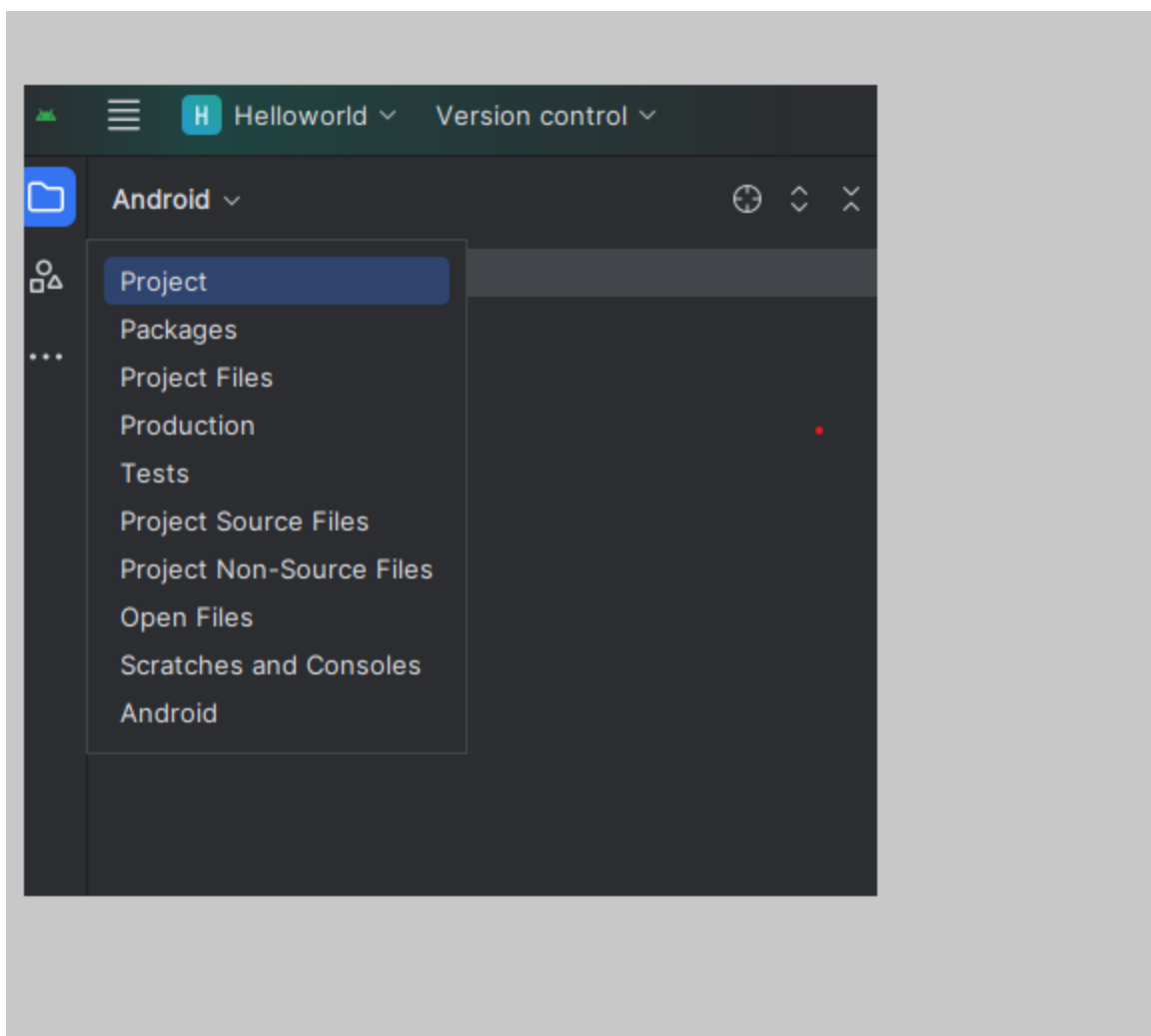
3. Nhấp vào tab MainActivity.java để xem mã soạn thảo

```
</> activity_main.xml    MainActivity.java x
1  package vn.tlu.edu.helloworld;
2
3  > import ...
10
11  </> public class MainActivity extends AppCompatActivity {
12
13      @Override
14      protected void onCreate(Bundle savedInstanceState) {
15          super.onCreate(savedInstanceState);
16          EdgeToEdge.enable(this);
17          setContentView(R.layout.activity_main);
18          ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
19              Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
20              v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
21              return insets;
22          });
23      }
24  }
```

2. Khám phá ngăn Project > Android

Trong phần thực hành này, bạn sẽ tìm hiểu cách tổ chức dự án trong **Android Studio**.

1. Nếu chưa được chọn, hãy nhấp vào tab **Project** trong cột tab dọc bên trái của cửa sổ **Android Studio**. Khi đó, ngăn **Project** sẽ xuất hiện.
2. Để xem dự án theo cấu trúc thư mục chuẩn của **Android**, chọn **Android** từ menu thả xuống ở đầu ngăn **Project**, như hình bên dưới.

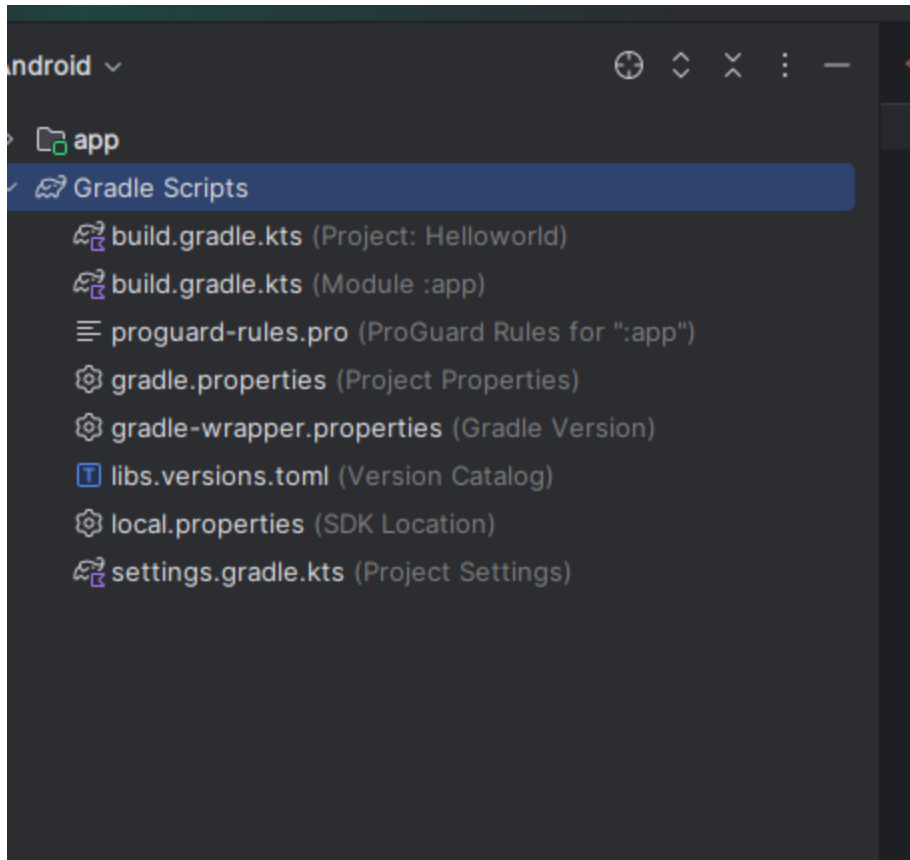


Lưu ý: Chương này và các chương khác đề cập đến bảng **Project** khi được đặt ở chế độ **Android** với tên gọi **Project > Android pane**.

2.3 Khám phá thư mục Gradle Scripts

Hệ thống **Gradle build** trong **Android Studio** giúp bạn dễ dàng thêm các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào dự án dưới dạng **dependencies**.

Khi bạn tạo một dự án ứng dụng lần đầu tiên, ngăn **Project > Android** sẽ hiển thị với thư mục **Gradle Scripts** được mở rộng như hình dưới đây.



Thực hiện các bước sau để khám phá 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)**.

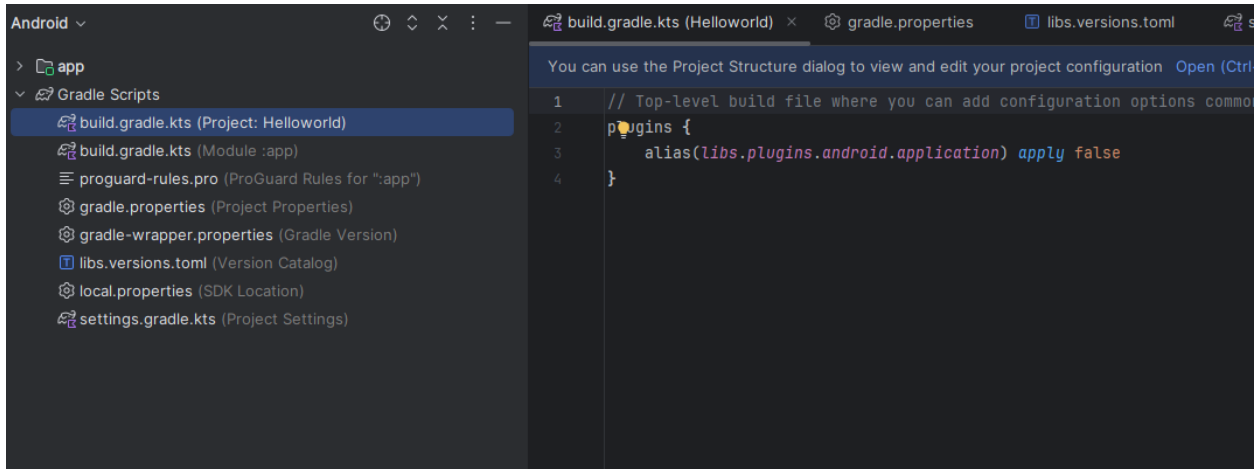
Đây là nơi chứa các tùy chọn cấu hình chung cho tất cả các mô-đun trong dự án của bạn.

Mọi dự án trong **Android Studio** đều có một tệp **Gradle build** cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần thay đổi tệp này, nhưng vẫn nên hiểu nội dung của nó.

Mặc định, tệp **build.gradle** cấp cao nhất sử dụng khối **buildscript** để định nghĩa các **repository** và **dependencies** chung cho tất cả các mô-đun trong dự án.

Khi **dependency** của bạn không phải là thư viện cục bộ hoặc thư mục tệp, **Gradle** sẽ tìm các tệp trong các **repository trực tuyến** được khai báo trong khối **repositories** của tệp này.

Mặc định, các dự án mới trong **Android Studio** sẽ khai báo **JCenter** và **Google** (bao gồm **Google Maven repository**) làm vị trí **repository**, như sau:



3. Tìm tệp build.gradle (Module: app)

Ngoài tệp **build.gradle** cấp dự án, mỗi mô-đun trong dự án cũng có một tệp **build.gradle** riêng. Điều này cho phép bạn cấu hình các cài đặt **build** cho từng mô-đun cụ thể (trong ứng dụng **HelloWorld**, chỉ có một mô-đun).

Cấu hình các cài đặt **build** giúp bạn tùy chỉnh cách đóng gói ứng dụng, chẳng hạn như:

Thêm **build types** (kiểu build)

Thêm **product flavors** (phiên bản sản phẩm)

Bạn cũng có thể **ghi** đề các cài đặt trong tệp **AndroidManifest.xml** hoặc **build.gradle** cấp dự án.

Đây là tệp quan trọng nhất cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như:

Khai báo **dependencies** trong phần **dependencies**.

Bạn có thể khai báo thư viện phụ thuộc theo nhiều cách khác nhau.

Mỗi cách khai báo sẽ cung cấp cho **Gradle** hướng dẫn khác nhau về cách sử dụng thư viện đó. Ví dụ Câu lệnh trên thêm tất cả các tệp **“.jar”** bên trong thư mục **libs** vào dự án như một dependency.

Dưới đây là tệp **build.gradle (Module: app)** của ứng dụng **HelloWorld**:

```
You can use the Project Structure dialog to view and edit your project configuration  Open (Ctrl+Alt+Shift+S)  Hide notification

5  android {
6      namespace = "vn.tlu.edu.helloworld"
7      compileSdk = 35
8
9      defaultConfig {
10         applicationId = "vn.tlu.edu.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
36     implementation(libs.appcompat)
37     implementation(libs.material)
38     implementation(libs.activity)
39     implementation(libs.constraintlayout)
40     testImplementation(libs.junit)
41     androidTestImplementation(libs.ext.junit)
42     androidTestImplementation(libs.espresso.core)
```

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

