

**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) Giao diện người dùng tương tác đầu tiên	28
1.3) Trình chỉnh sửa bố cục	50
1.4) Văn bản và các chế độ cuộn	50
1.5) Tài nguyên có sẵn	50
Bài 2) Activities.....	50
2.1) Activity và Intent.....	50
2.2) Vòng đời của Activity và trạng thái	50
2.3) Intent ngầm định	50
Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ	50
3.1) Trình gỡ lỗi.....	50
3.2) Kiểm thử đơn vị	50
3.3) Thư viện hỗ trợ	50
CHƯƠNG 2. Trải nghiệm người dùng.....	51
Bài 1) Tương tác người dùng	51
1.1) Hình ảnh có thể chọn.....	51
1.2) Các điều khiển nhập liệu	51
1.3) Menu và bộ chọn	51
1.4) Điều hướng người dùng	51
1.5) RecyclerView	51
Bài 2) Trải nghiệm người dùng thú vị.....	51
2.1) Hình vẽ, định kiểu và chủ đề	51
2.2) Thẻ và màu sắc	51
2.3) Bố cục thích ứng.....	51
Bài 3) Kiểm thử giao diện người dùng	51

3.1) Espresso cho việc kiểm tra UI	51
CHƯƠNG 3. Làm việc trong nền	51
Bài 1) Các tác vụ nền.....	51
1.1) AsyncTask.....	51
1.2) AsyncTask và AsyncTaskLoader	51
1.3) Broadcast receivers	51
Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	51
2.1) Thông báo.....	51
2.2) Trình quản lý cảnh báo	51
2.3) JobScheduler	51
CHƯƠNG 4. Lưu dữ liệu người dùng	52
Bài 1) Tùy chọn và cài đặt.....	52
1.1) Shared preferences	52
1.2) Cài đặt ứng dụng	52
Bài 2) Lưu trữ dữ liệu với Room	52
2.1) Room, LiveData và ViewModel.....	52
2.2) Room, LiveData và ViewModel.....	52
3.1) Trình gowx loi	

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.

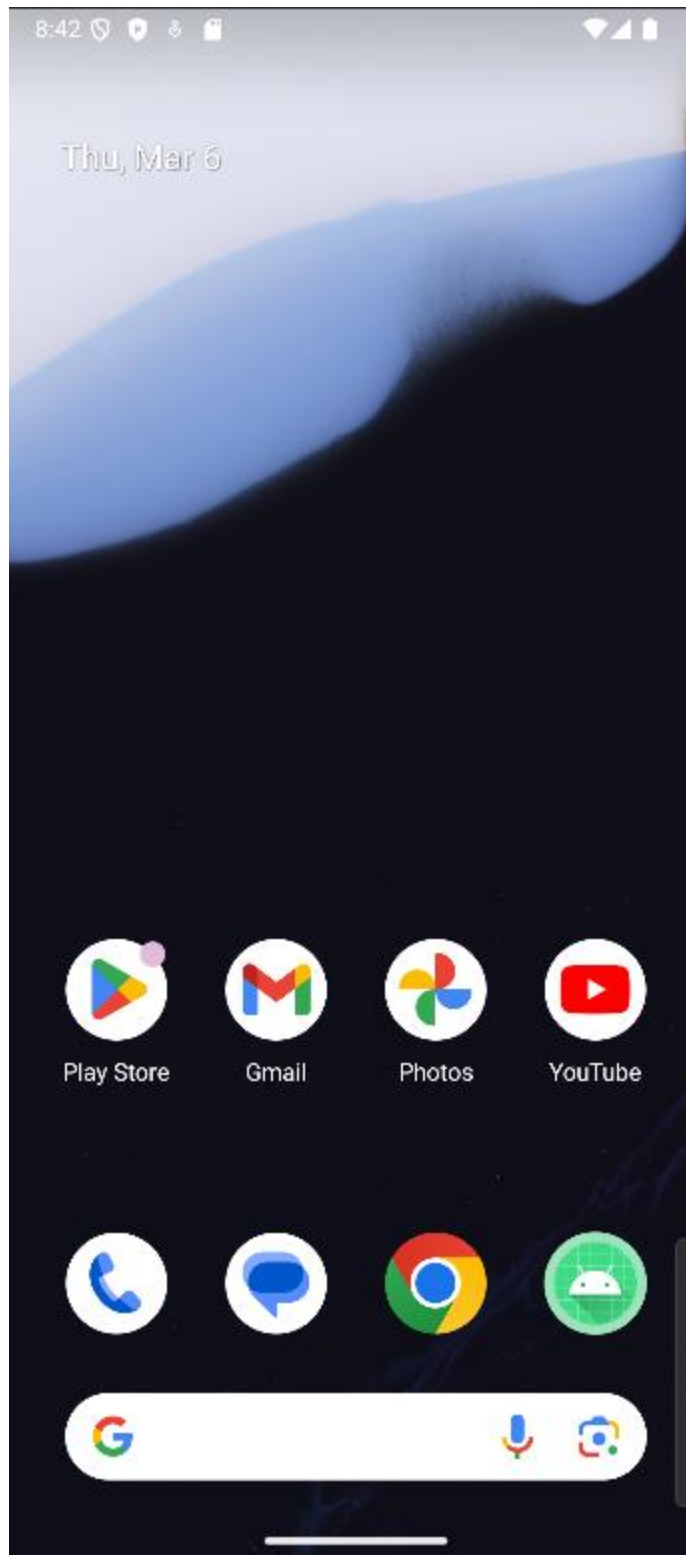
Những gì bạn sẽ làm

- 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, từ một mẫu, một dự án mới là ứng dụng “ Hello word “. Là một ứng dụng đơn giản hiển thị chuỗi “ Hello Word “ trên màn hình của máy ảo Android hoặc thiết bị vật lý.

Đây là những gì sau khi hoàn thành ứng dụng sẽ như thế này :



Task 1 : Cài Đặt Android Studio

Android Studio cung cấp môi trường tích hợp trình chỉnh sửa mã nguồn nâng cao và một bộ ứng dụng mẫu. Ngoài ra, nó còn chứa các công cụ để phát triển, sửa lỗi, kiểm thử, và hiệu suất làm cho nó nhanh hơn và dễ dàng hơn cho việc phát triển các ứng dụng. Bạn có thể kiểm thử ứng dụng của mình với một loạt giả lập được cấu hình sẵn hoặc trên thiết bị di động riêng của bạn, xây dựng các ứng dụng sản xuất, và xuất bản nó trên cửa hàng Google Play.

Android Studio có sẵn cho các máy tính chạy Window hoặc Linux, các máy Mac chạy macOS. OpenJDK mới nhất (Bộ phát triển Java) đi kèm với Android Studio.

Để thiết lập và chạy Android Studio, trước tiên kiểm tra các yêu cầu hệ thống để đảm bảo rằng hệ thống của bạn đáp ứng các yêu cầu. Việc cài đặt tương tự cho tất cả các nền tảng. Bất kỳ sự khác biệt nào được lưu ý bên dưới.

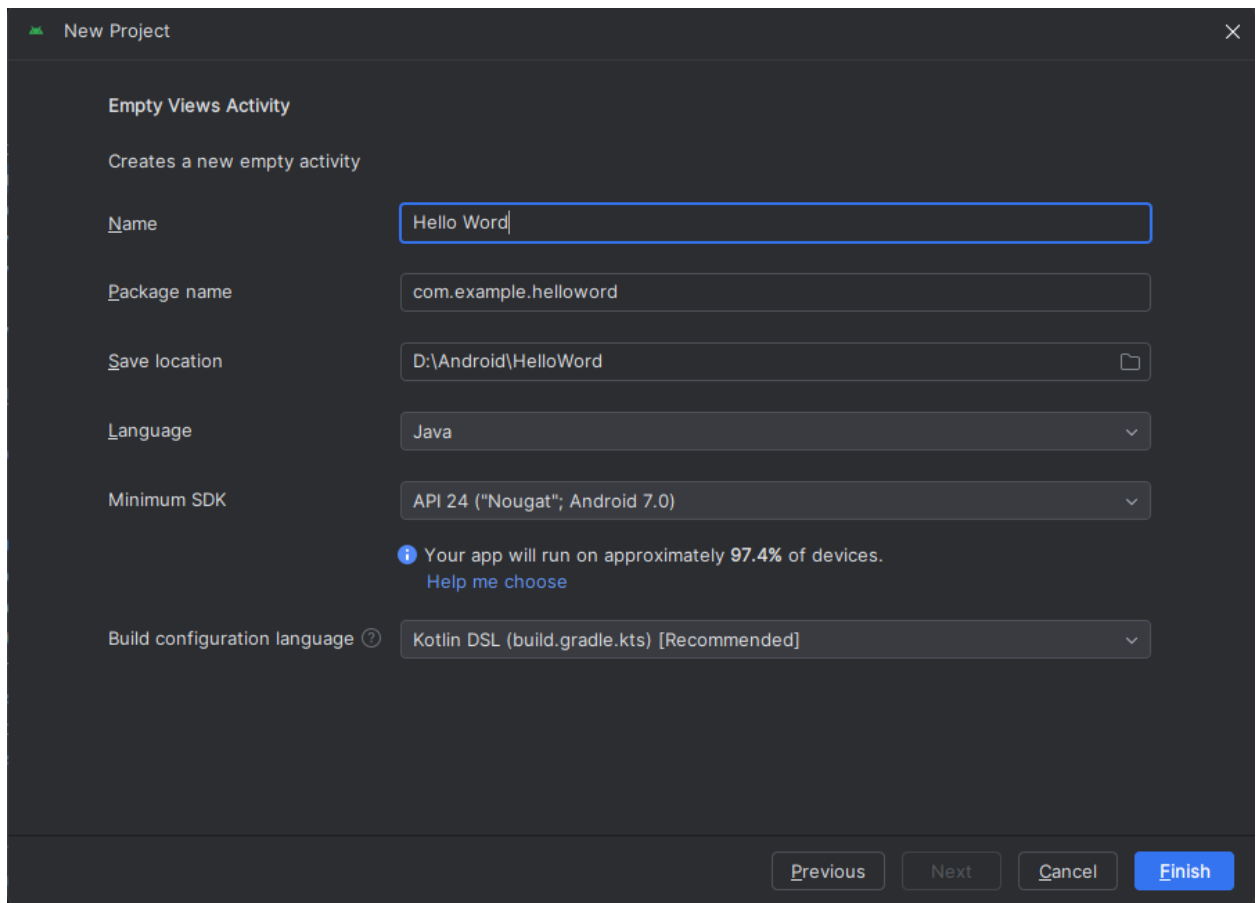
- 1) Chuyển đến trang Web dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.
- 2) Chấp nhận các cấu hình mặc định cho tất cả các bước, và đảm bảo rằng tất cả các thành phần được lựa chọn trong khi tải xuống.
- 3) Sau khi hoàn thành việc cài đặt , trình hướng dẫn thiết lập sẽ tải xuống và cài đặt một số thành phần bổ sung bao gồm SDK Android. Hãy kiên nhẫn, quá trình này có thể mất một khoảng thời gian do tốc độ mạng của bạn, và do một số bước dư thừa.
- 4) Khi hoàn tất việc tải xuống, Android Studio sẽ khởi động, và bạn hãy sẵn sàng tạo cho mình dự án đầu tiên.

Task 2 : Tạo Ứng Dụng Hello Word

Ở nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị ra “ Hello Word “ để xác minh rằng Android Studio được cài đặt một cách chính xác, và tìm hiểu những kiến thức cơ bản của phát triển cùng với Android Studio.

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu nó chưa được mở
2. Trong cửa sổ Welcome to Android Studio chính, nhấp vào Start a new Android Studio project
3. Trong cửa sổ Create Android Project, nhập Hello World cho tên ứng dụng



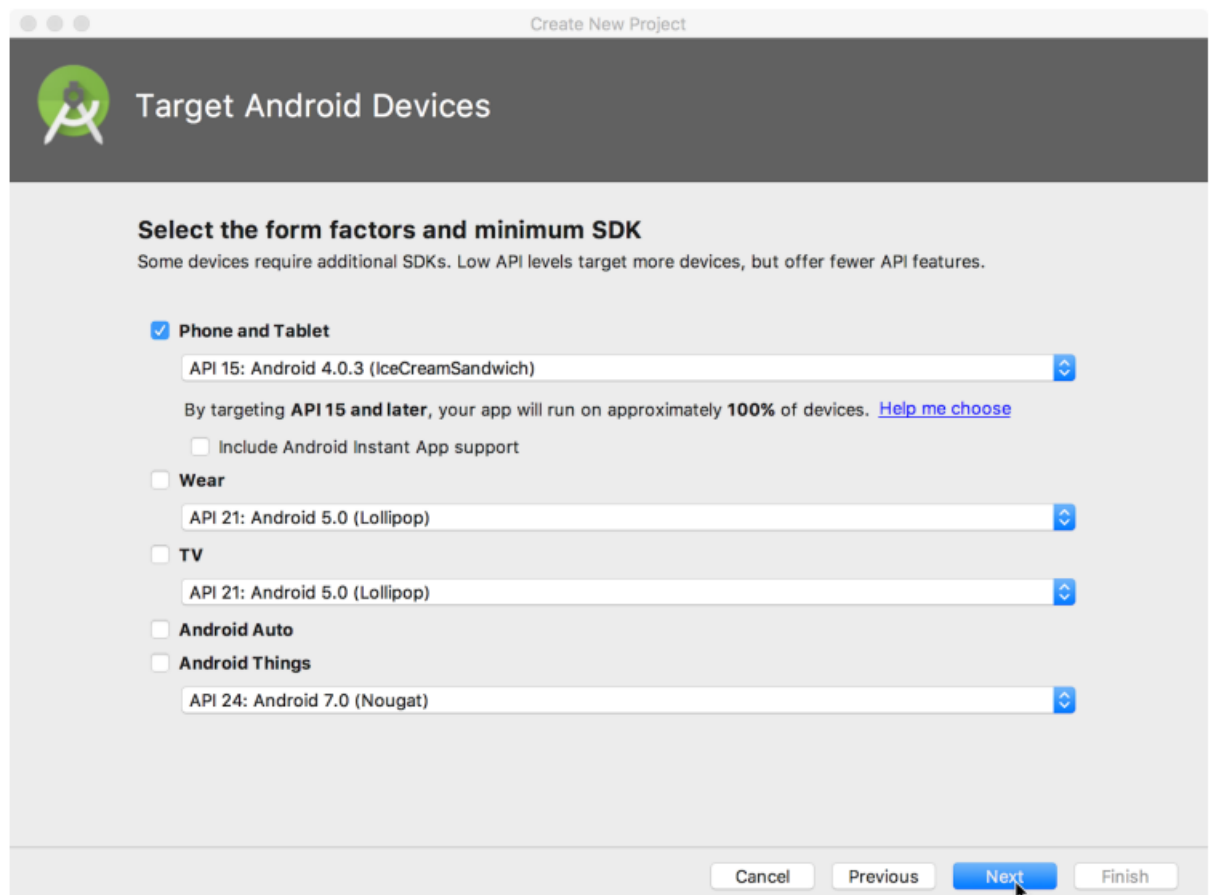
4. Xác minh rằng vị trí mặc định của Dự án là nơi bạn muốn lưu ứng dụng Hello World và các dự án Android Studio khác, hoặc thay đổi nó thành thư mục bạn muốn.

5. Chấp nhận tên miền mặc định android.example.com cho Tên miền Công ty, hoặc tạo một tên miền công ty độc đáo.

Nếu bạn không có kế hoạch xuất bản ứng dụng của mình, bạn có thể chấp nhận mặc định. Hãy lưu ý rằng việc thay đổi tên gói (package name) của ứng dụng sau này sẽ đòi hỏi thêm công sức.

6. Không đánh dấu chọn các tùy chọn Hỗ trợ C++ và Hỗ trợ Kotlin, sau đó nhấp vào Tiếp theo.

7. Trên màn hình Thiết bị Android Mục tiêu, mục Điện thoại và Máy tính bảng nên được chọn. Đảm bảo rằng API 24: Android 4.0.3 IceCreamSandwich được đặt làm SDK Tối thiểu; nếu không phải, hãy sử dụng menu thả xuống để thiết lập nó.



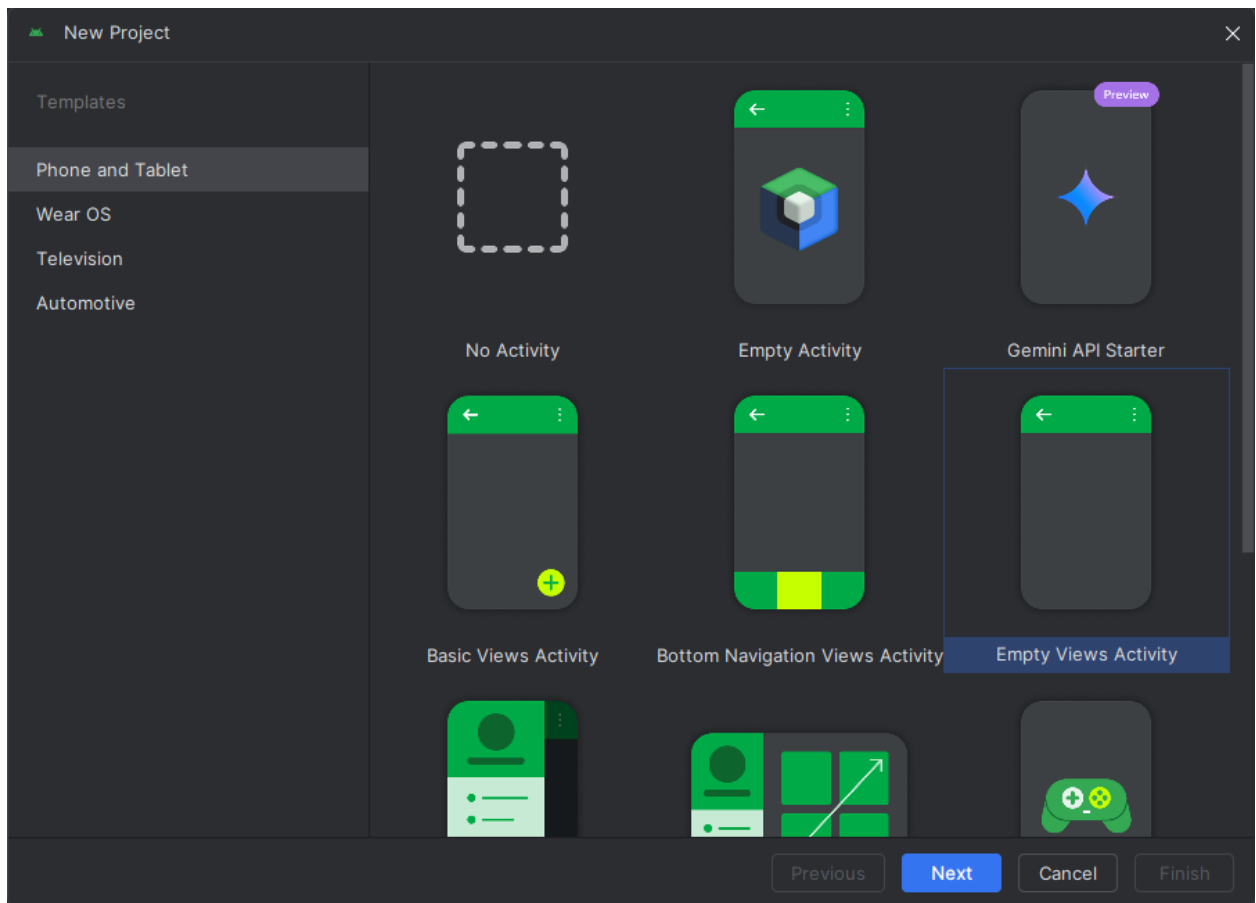
Đây là các cài đặt được sử dụng bởi các ví dụ trong các bài học của khóa học này. Tính đến thời điểm viết tài liệu này, các cài đặt này giúp ứng dụng Hello World của bạn tương thích với 97% thiết bị Android đang hoạt động trên Google Play Store.

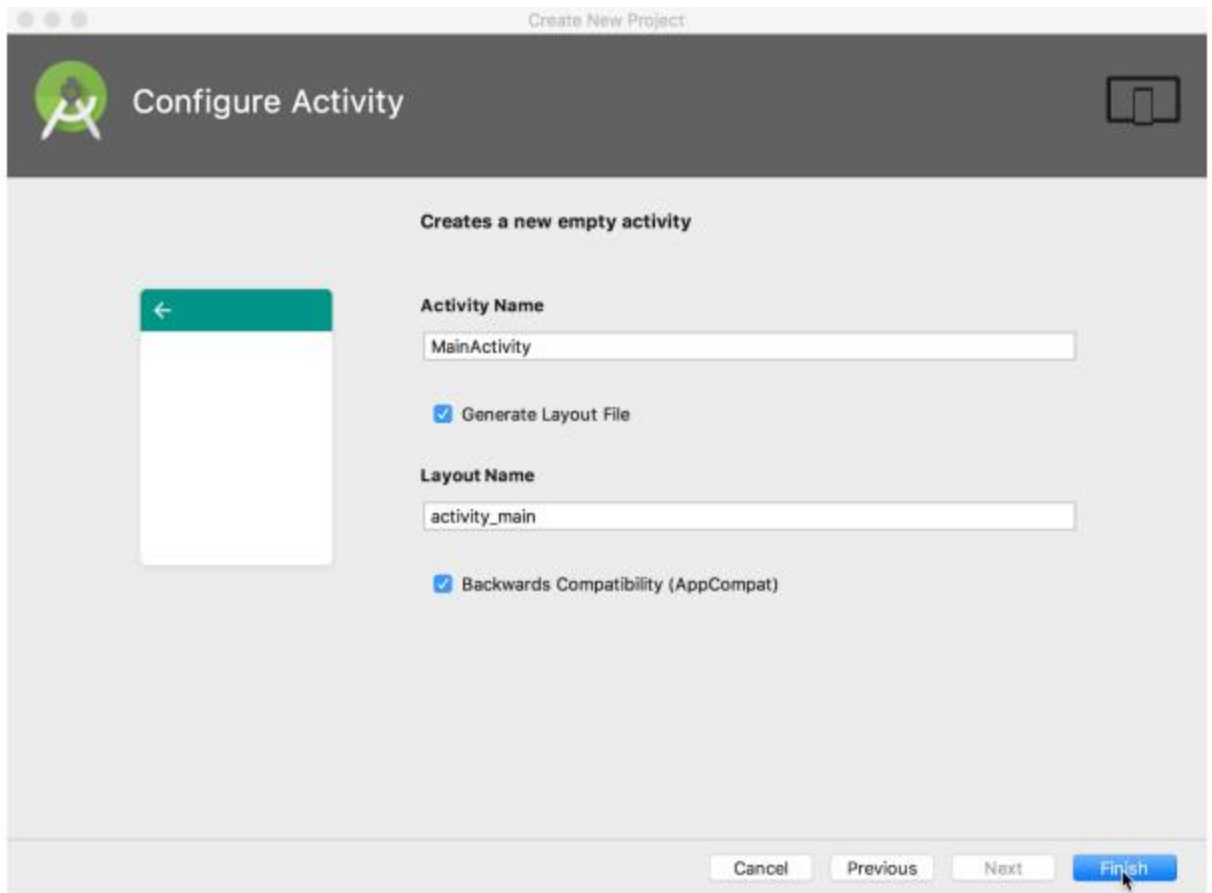
8. Không đánh dấu chọn tùy chọn Hỗ trợ Ứng dụng Tức thì và tắt cả các tùy chọn khác. Sau đó nhấp vào Tiếp theo. Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK mục tiêu đã chọn, Android Studio sẽ tự động cài đặt chúng.

9. Cửa sổ Thêm một Hoạt động xuất hiện. Một Hoạt động (Activity) là một điều đơn lẻ, tập trung mà người dùng có thể thực hiện. Nó là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một Hoạt động thường có một bố cục liên kết với nó, định nghĩa cách các yếu tố giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp các mẫu Hoạt

động để giúp bạn bắt đầu. Đối với dự án Hello World, hãy chọn Hoạt động Trống (Empty Activity) như hình dưới đây, rồi nhấp vào Tiếp theo.

10. Màn hình cấu hình sẽ xuất hiện (màn hình này thay đổi tùy thuộc vào mẫu bạn đã chọn ở bước trước). Theo mặc định, Activity trống được cung cấp bởi mẫu sẽ được đặt tên là MainActivity. Bạn có thể thay đổi tên này nếu muốn, nhưng bài học này sẽ sử dụng tên MainActivity.





11. Hãy đảm bảo rằng tùy chọn "Generate Layout file" (Tạo tệp bố cục) được chọn. Tên bố cục mặc định là `activity_main`. Bạn có thể thay đổi tên này nếu muốn, nhưng bài học này sẽ sử dụng `activity_main`.

12. Hãy đảm bảo rằng tùy chọn "Backwards Compatibility (App Compat)" (Tương thích ngược) được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản trước của Android.

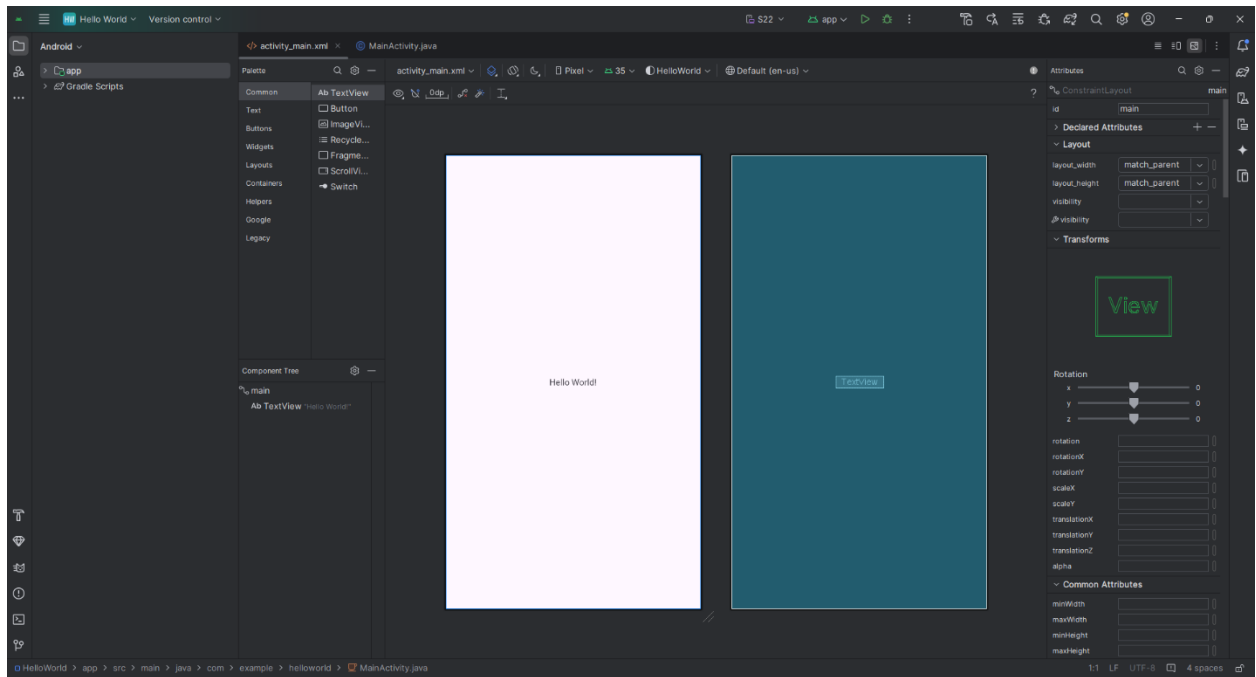
13. Nhấn nút "Finish" (Hoàn tất).

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

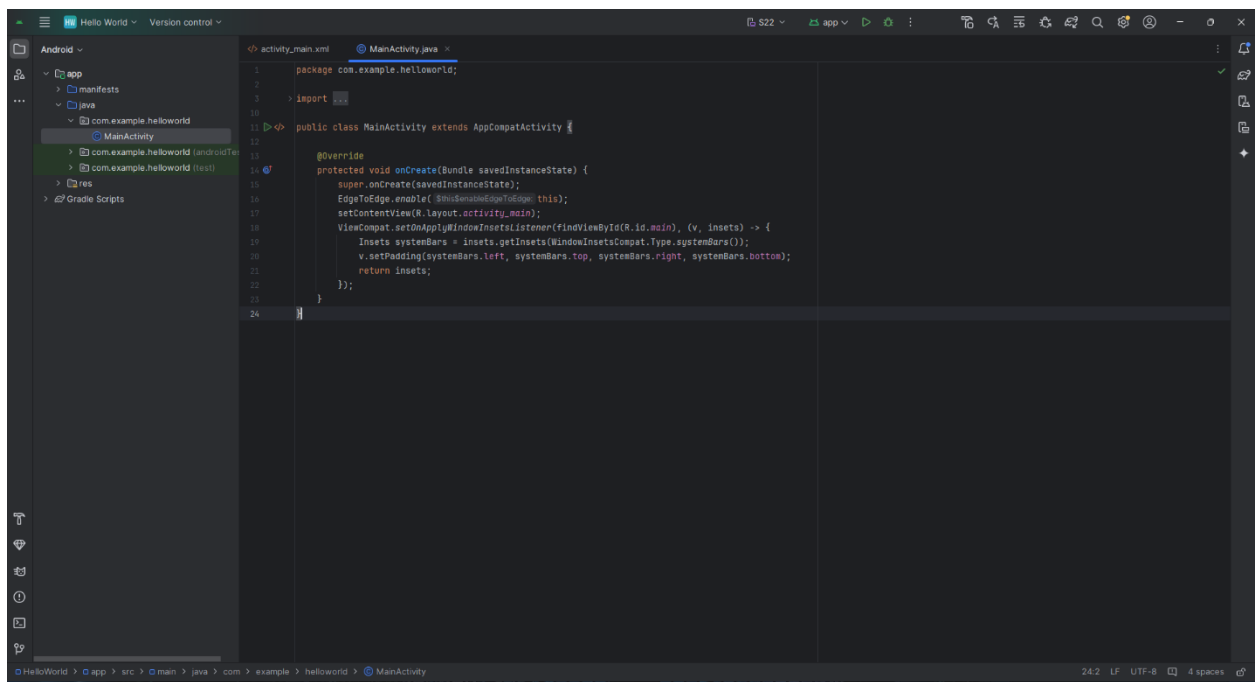
Trình chỉnh sửa Android Studio sẽ xuất hiện. Làm theo các bước sau:

1. Nhấp vào tab **activity_main.xml** để xem trình chỉnh sửa bố cục.

2. Nhấn vào tab "**Design**" (Thiết kế) trong trình chỉnh sửa bố cục, nếu nó chưa được chọn, để hiển thị một bản vẽ đồ họa của bố cục như được minh họa dưới đây.



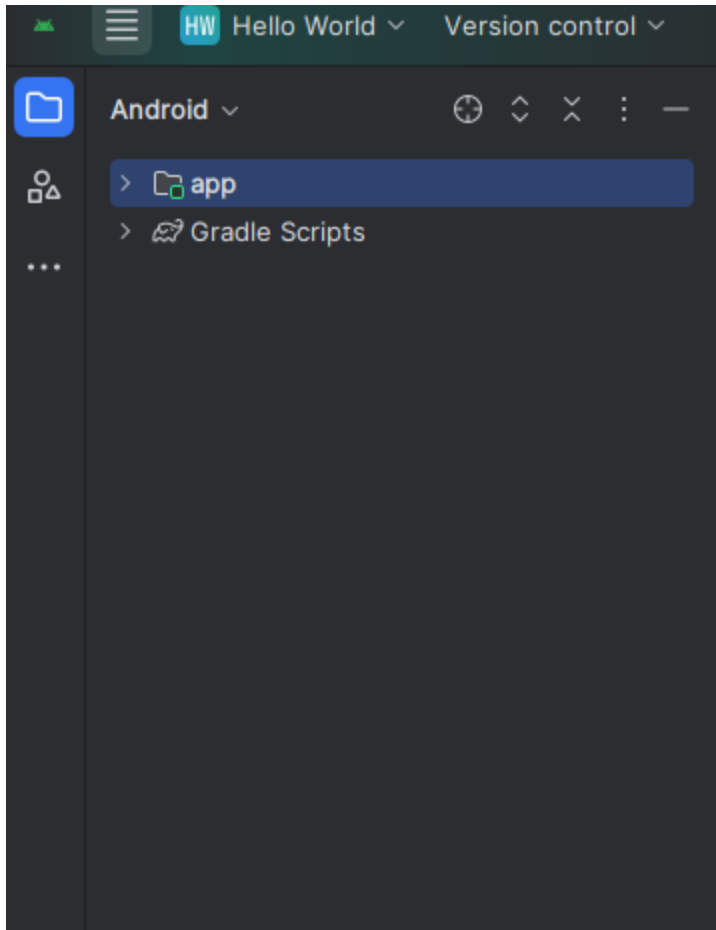
3. Nhấp vào tab **MainActivity.java** để xem trình chỉnh sửa mã như hình dưới đây



2.1 Khám Phá Dự Án > Cửa sổ Android

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

1. Nếu chưa được chọn, hãy nhấp vào tab Project trong cột tab dọc ở phía bên trái của cửa sổ Android Studio. Ngăn dự án xuất hiện
2. Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, hãy chọn Android từ menu bật lên ở đầu ngăn dự án, như được hiển thị bên dưới

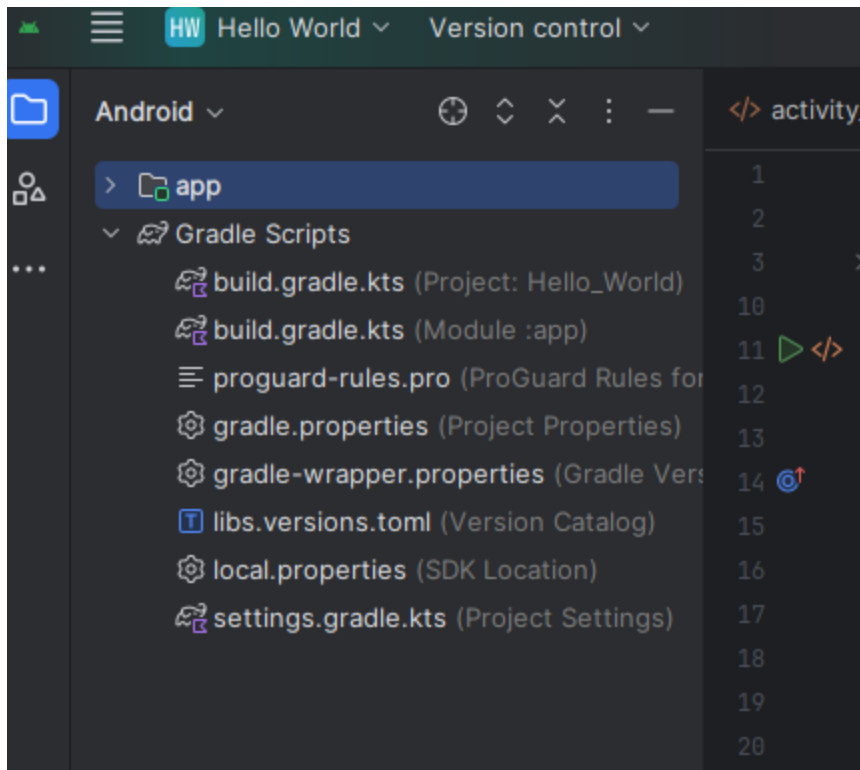


Lưu ý: Chương này và các chương khác đề cập đến khung dự án, khi được đặt thành **Android**, là khung **project > Android**

2.3 Khám Phá Thư Mục Gradle Scripts

Hệ thống xây dựng Gradle trong Android Studio giúp dễ dàng thêm các tệp nhị phân bên ngoài hoặc các module thư viện khác vào bản dựng của mình dưới dạng các phần phụ thuộc.

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



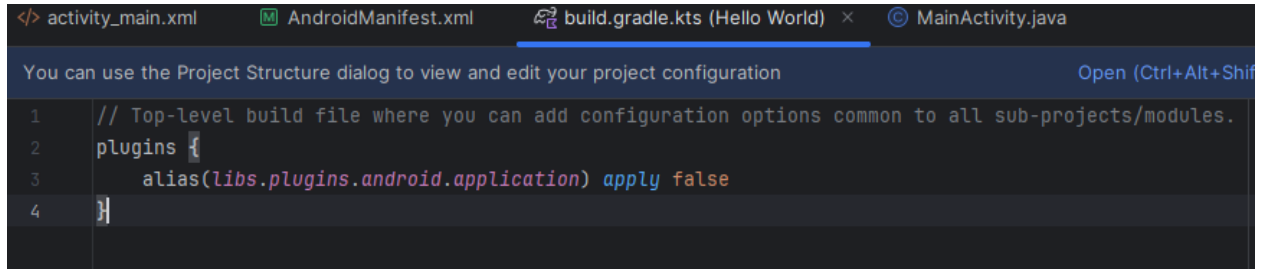
Thực hiện các bước sau để khám phá hệ thống Gradle:

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 dựng.
2. Tìm tệp **build.gradle.kts** (Project: Hello_World)

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung tất cả các module tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một Gradle cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần chỉnh sửa tệp này nhưng việc hiểu nội dung của nó vẫn rất hữu ích.

Theo mặc định, tệp xây dựng cấp cao nhất sử dụng khối buildscript để xác định các kho lưu trữ Gradle vào các phụ thuộc chung cho tất cả các module 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 bất kỳ kho lưu trữ trực tuyến nào được chỉ định trong khối repositories của tệp này. Theo mặc định, các dự án mới

tạo trong Android Studio khai báo JCenter và Google (bao gồm kho lưu trữ Google Maven) là các vị trí kho lưu trữ

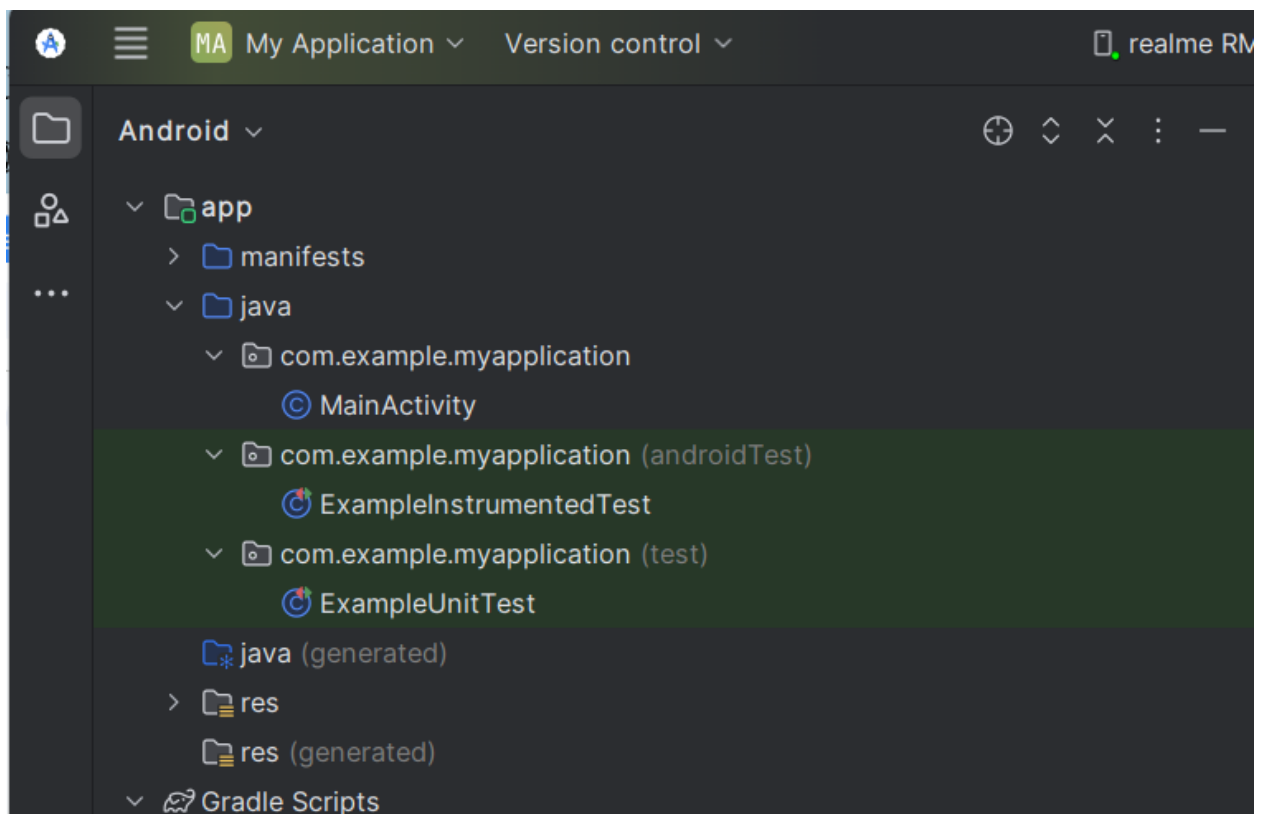


```
<?xml version="1.0" encoding="UTF-8"?>
<!-- You can use the Project Structure dialog to view and edit your project configuration -->
<!-- Open (Ctrl+Alt+Shift) -->
<!-- Top-level build file where you can add configuration options common to all sub-projects/modules. -->
<!-- plugins -->
<!-- alias(libs.plugins.android.application) apply false -->
<!-- } -->
</?xml>
```

2.4 Khám phá ứng dụng và thư mục res

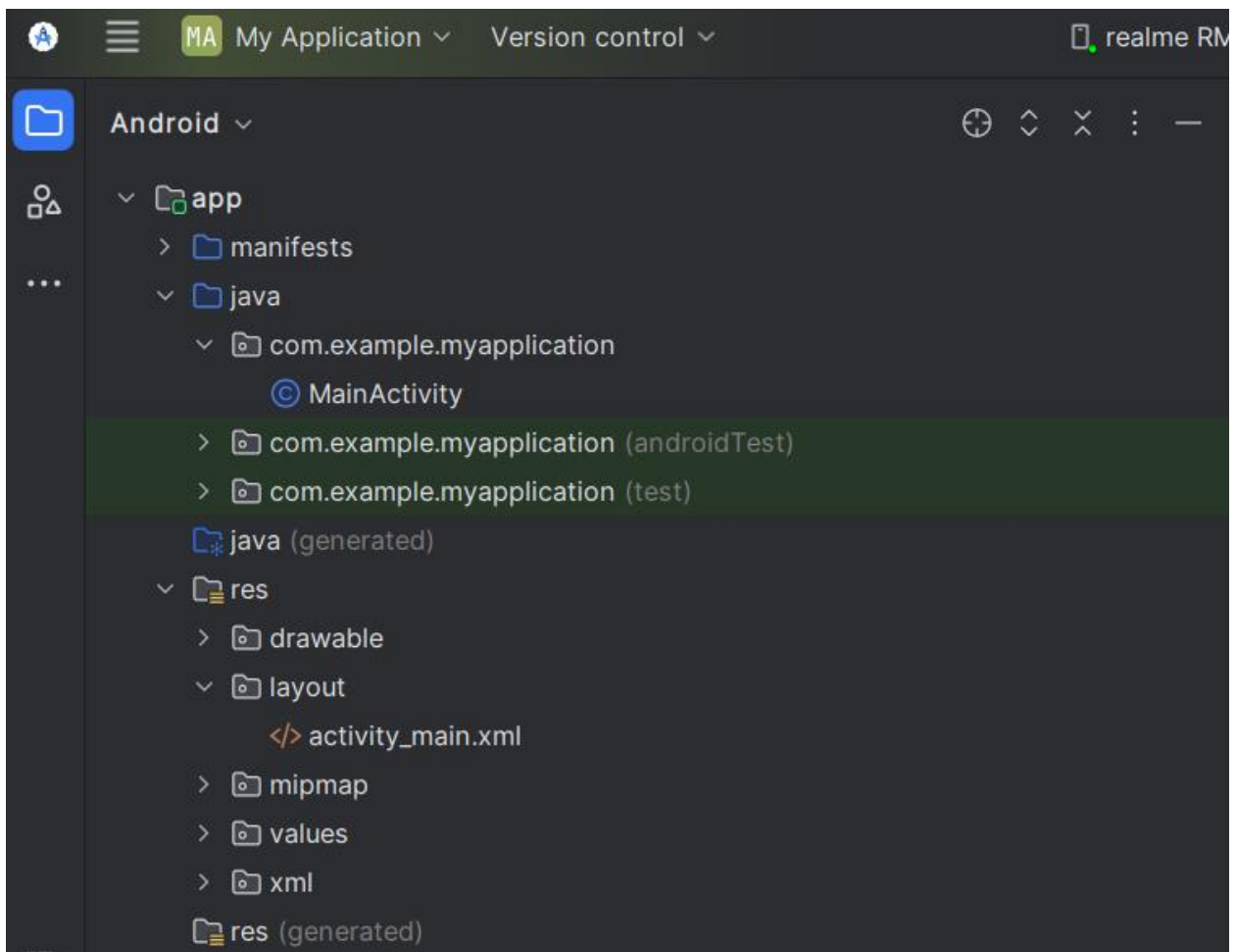
Tất cả mã và tài nguyên cho ứng dụng nằm trong thư mục app và res .

Mở rộng thư mục **app**, thư mục **java** và thư mục **com.example.android.helloworld** để xem tệp java **MainActivity**. Nhấp đúp vào tệp sẽ mở tệp đó trong trình soạn thảo mã.



Thư mục **java** bao gồm các tệp lớp Java trong ba thư mục con, như trong 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 tra và được mô tả trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa MainActivity.java. Tên của màn hình A (màn hình) đầu tiên mà người dùng nhìn thấy, cũng khởi tạo tài nguyên trên toàn ứng dụng, thường được gọi là **MainActivity** (phần mở rộng tệp bị bỏ qua trong ngăn **Project > Android**).

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



Thư mục **res** chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh. Activity thường được liên kết với bố cục của các chế độ xem giao diện người dùng được định nghĩa dưới dạng tệp XML. Tệp này thường được đặt tên theo Hoạt động của nó.

2.5 Khám phá thư mục tệp kê khai

Thư mục manifests chứa các tệp cung cấp thông tin cần thiết về ứng dụng của bạn cho

Hệ thống Android, hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng.

Mở rộng thư mục **manifests**.

Mở tệp **AndroidManifest.xml**.

Tệp AndroidManifest.xml mô tả tất cả các thành phần của ứng dụng Android. Tất cả các thành phần cho một ứng dụng, chẳng hạn như mỗi thành phần A , phải được khai báo trong tệp XML này. Trong các bài học khóa học khác, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để biết phần giới thiệu, hãy xem Tổng [quan về bản kê khai pp](#).

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

Trong tác vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo (AVD) để tạo một thiết bị ảo (còn được gọi là trình mô phỏng) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Xin lưu ý rằng Trình mô phỏng Android có các yêu cầu riêng ngoài các yêu cầu hệ thống cơ bản cho Android Studio.

Khi sử dụng Trình quản lý AVD, bạn xác định các đặc điểm phần cứng của thiết bị, cấp độ API, bộ nhớ, giao diện và các thuộc tính khác và lưu thiết bị đó dưới dạng thiết bị ảo. Với thiết bị ảo, bạn có thể kiểm tra ứng dụng trên các 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 phải sử dụng thiết bị thực.

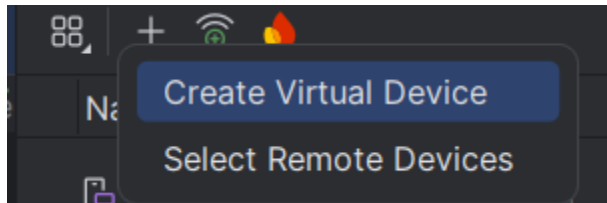
3.1 Tạo thiết bị ảo Android (AVD)

Để chạy trình mô phỏng trên máy tính, bạn phải tạo một cấu hình mô tả thiết bị ảo.

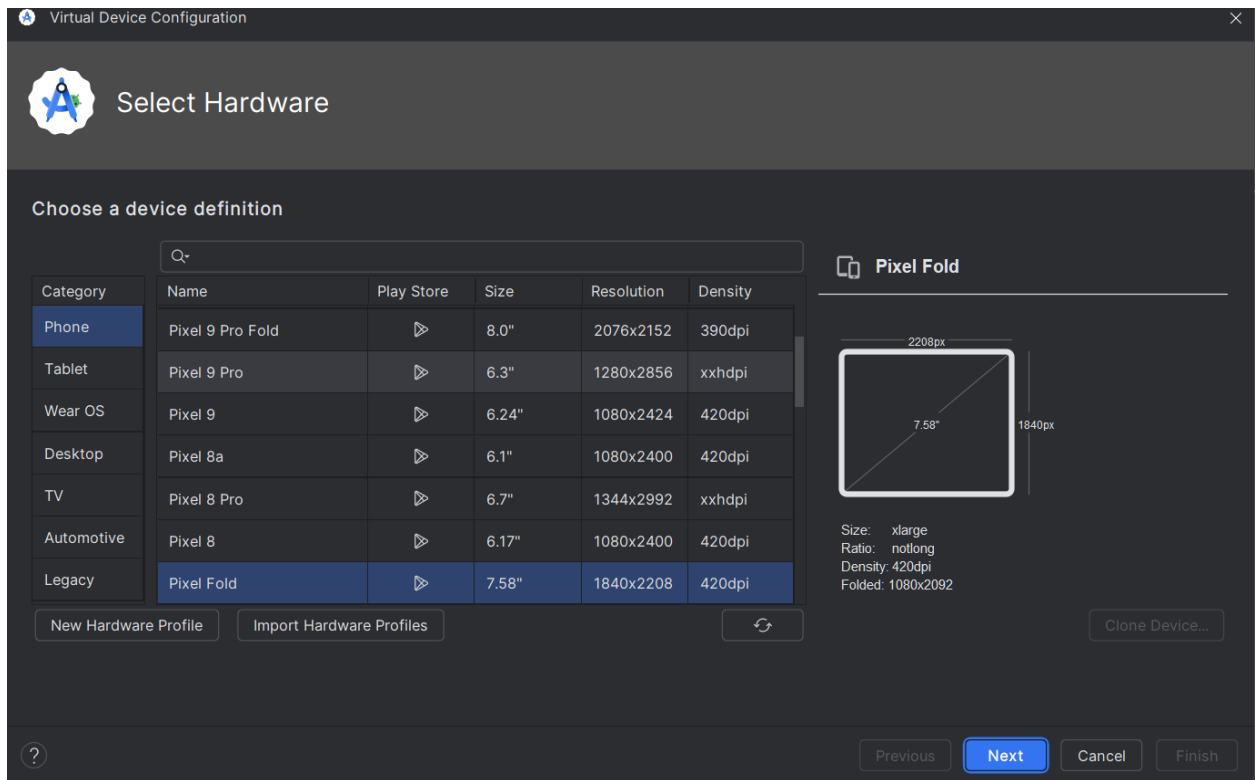
Trong Android Studio, chọn **Tools > Trình quản lý Android > AVD** hoặc nhấp vào biểu tượng Trình quản lý AVD



trên thanh công cụ. Thiết bị ảo của chúng tôi xuất hiện. Nếu bạn đã tạo thiết bị ảo, màn hình hiển thị chúng (như trong hình bên dưới); nếu không, bạn sẽ thấy một danh sách trống.

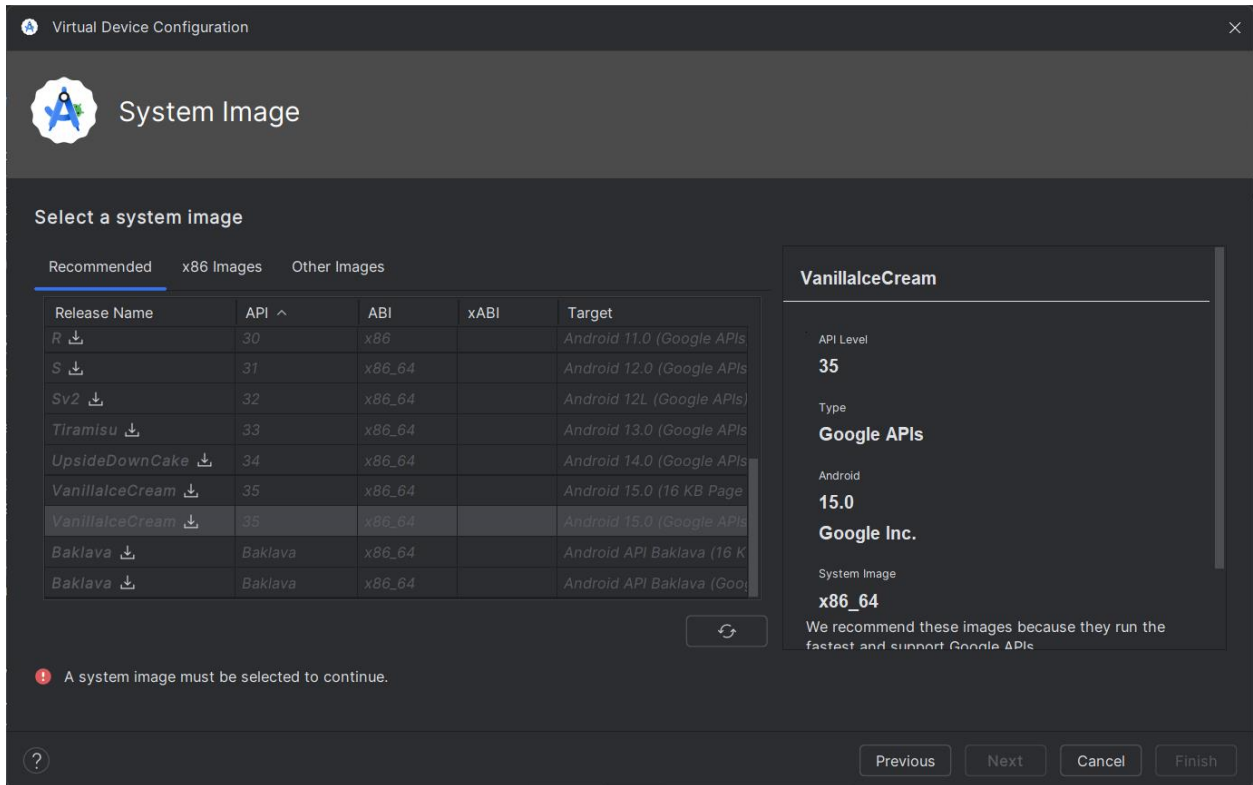


2. Nhấp vào **+ Tạo thiết bị ảo**. Cửa sổ **Select Hardware** xuất hiện hiển thị danh sách các thiết bị phần cứng được định cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo (**Size**), độ phân giải màn hình tính bằng pixel (**Resolution**) và mật độ pixel (**Density**).



3. Chọn một thiết bị như **Nexus 5x** hoặc **Pixel XL**, và nhấp vào **Next**. **System Image** screen xuất hiện.

4. Nhấp vào tab **Đề xuất** nếu nó chưa được chọn và chọn phiên bản hệ thống Android để chạy trên thiết bị ảo (chẳng hạn như **Oreo**).



Có nhiều phiên bản có sẵn hơn so với hiển thị trong tab **R khuyến nghị**. Nhìn vào các tab **x86 Images** và **Other Images** để xem chúng.

Nếu liên kết **Download** hiển thị bên cạnh hình ảnh hệ thống bạn muốn sử dụng, nó vẫn chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào **Finish** khi hoàn tất.

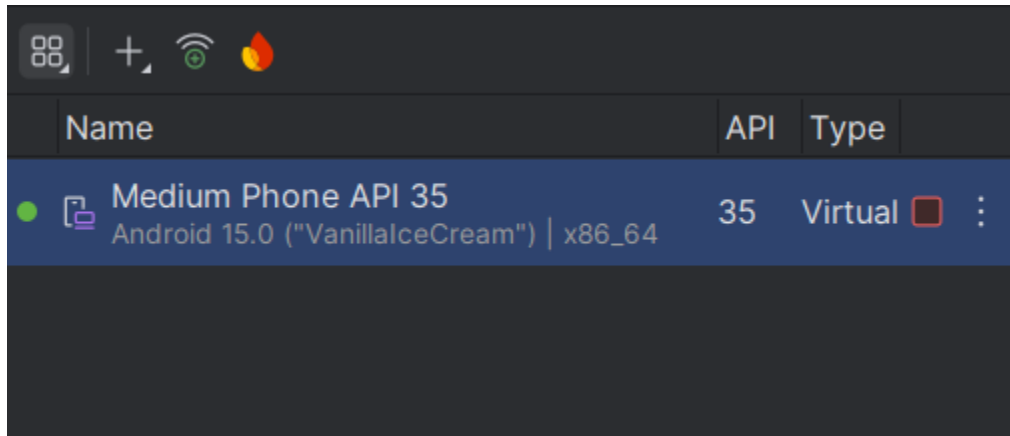
5. Sau khi chọn hình ảnh hệ thống, nhấp vào **Next**. Cửa sổ **Thiết bị ảo Android (AVD)** xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào **Finish**.

1. 3.2 Chạy ứng dụng trên thiết bị ảo

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

1. Trong Android Studio, chọn **Run > Chạy ứng dụng** hoặc nhấp vào **R** biểu tượng Tôi trên thanh công cụ.

2 Cửa sổ Chọn mục tiêu triển khai, bên dưới Thiết bị ảo có sẵn, 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.

8:35



Hello World!

Mẹo : Khi thử nghiệm trên thiết bị ảo, bạn nên khởi động thiết bị một lần, ngay khi bắt đầu phiên làm việc. Bạn không nên đóng thiết bị cho đến khi hoàn tất việc 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** tại

trên cùng của trình giả lập, chọn **Q uit** từ menu hoặc nhấn **C ontrol -Q** trong Windows hoặc **Command-Q** trong macOS.

Task 4: (Optional) Use a physical device (Tùy chọn) Sử dụng thiết bị vật lý

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 [U Tài liệu về Thiết bị phần cứng](#). 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 [O Trình điều khiển USB EM](#)

2. 4.1 Bật gỡ lỗi USB


Để cho phép 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 **D phát triển cài đặt tùy chọn** của thiết bị của bạn.

Trên Android 4.2 trở lên, **D Màn hình tùy chọn 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 **S cài đặt**, tìm kiếm Giới thiệu **P mài giữa**, nhấp vào **A trận đấu P mài giữa**, và chạm vào **Số bản dựng** bảy lần.
2. Trở về màn hình trước đó (**S cài đặt / Hệ thống**) . **D tùy chọn phát triển** xuất hiện trong danh sách. Nhấn **D tùy chọn phát triển** .
3. Chọn **bạn Gỡ lỗi SB** .

3. **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 **R** nút  **un** trong thanh công cụ. **S** Cửa sổ **Deployment Target** 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 **O K**.

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

4. **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 **D** Tùy chọn **eveloper trong ứng dụng Cài đặt** .
3. Nhấn Thu hồi **U** Quyền **gỡ lỗi SB** .
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 [U hát Thiết bị phần cứng tài liệu](#) .

5. **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 `build.gradle` (Mô-đun: ứng dụng) tệp để tìm hiểu cách thực hiện thay đổi và đồng bộ hóa chúng với dự án Android Studio của bạn.

6. **5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng**

Thực hiện theo các bước sau:

1. Mở rộng **G** thư mục **radle Scripts** nếu nó chưa được mở và nhấp đúp vào tệp **build.gradle (Module:app)** .

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

2. Trong `defaultConfig` khối , thay đổi giá trị của `minSdkVersion` Phiên bản trong SDK đến 17 như được hiển thị bên dưới (ban đầu nó được đặt thành 15) .


```
activity_main.xml MainActivity.java build.gradle.kts (:app) ×
You can use the Project Structure dialog to view and edit your project configurati... Open (Ctrl+Alt+Shift+S) Hide notificat

1  plugins {
2      alias(libs.plugins.android.application)
3  }
4
5  android {
6      namespace = "com.example.myapplication"
7      compileSdk = 35
8
9      defaultConfig {
10         applicationId = "com.example.myapplication"
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("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 }
```

1. 5.2 Đồng bộ cấu hình Gradle mới

Khi bạn thực hiện thay đổi đối với các tệp cấu hình xây dựng trong một dự án, Android Studio yêu cầu bạn *S 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 **Sync Bây giờ** trên thanh thông báo xuất hiện khi thực hiện thay đổi

(như thể hiện trong hình trước đó), hoặc nhấp vào **S** Biểu tượng Dự án  **ync với Tập Gradle** trên thanh công cụ.

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

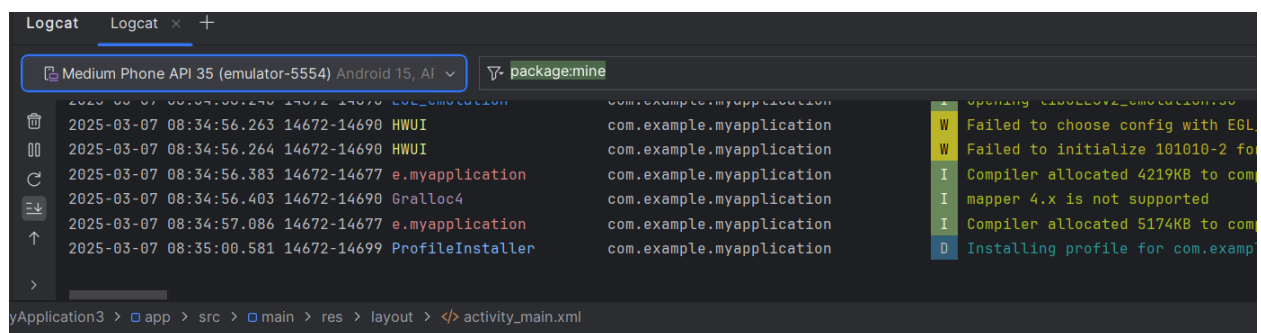
Để có cái nhìn sâu hơn về Gradle, hãy xem [B Tổng quan hệ thống build](#) và [C cấu hình tài liệu Gradle Builds](#).

Task 6: Add log statements to your app

Trong nhiệm vụ này, bạn sẽ thêm **Log** các câu lệnh cho ứng dụng của bạn, hiển thị các thông báo trong **Logcat** pane. Nhặt ký tin nhắn là 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.

1. Giao diện người dùng tương tác đầu tiên **6.1 Xem ngăn Logcat**

Để xem **Logcat**, nhấp vào **Logcat** ở cuối cửa sổ Android Studio như thể hiện trong hình bên dưới.



Trong hình trên:

Chữ **Logcat** tab để mở và đóng **Logcat** pane, hiển thị thông tin về ứng dụng của bạn khi nó đang chạy. Nếu bạn thêm **Log** các câu lệnh cho ứng dụng của bạn, **Log** tin nhắn xuất hiện ở đây.

Chữ **Log** menu cấp độ được đặt thành **Verbose** (mặc định), hiển thị tất cả **Log** tin nhắn. Các thiết lập khác bao gồm **Debug**, **Error**, **Thông tin**, và **Warn**.

6.2 Thêm câu lệnh nhật ký vào ứng dụng của bạn

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

```
Log.d (" MainActivity ", "Xin chào thế giới");
```

Các phần của tin nhắn bao gồm:

- **Nhật ký** : Chữ **L**og lớp để gửi tin nhắn nhật ký đến ngăn Logcat.
- **d** : Chữ **D** **lỗi** Log thiết lập mức để lọc hiển thị thông báo nhật ký trong ngăn Logcat. Các mức nhật ký khác là **e** cho **E** **lỗi** , chúng tôi cho **W** **arn** , và tôi vì **t**ôi **thông tin** .
- **" Hoạt động chính "** : Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong ngăn Logcat. Đây thường là tên của A hoạt động mà thông điệp bắt nguồn từ đó. 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 A hoạt động :

```
private static final String LOG_TAG =  
MainActivity.class.getSimpleName();
```

Mở ứng dụng Hello World của bạn trong Android studio và mở MainActivity.

Để tự động thêm các mục nhập rõ ràng vào dự án của bạn (chẳng hạn như android.util.Log cần thiết để sử dụng Log) , chọn **File > Settings** n Windows, hoặc **Android Studio > Preferences** trong macOS.

Chọn **Editor > General >Auto Import**. Chọn tất cả các hộp kiểm và đặt **I** **nhập khẩu** **n**sert **ô**i **n** **dán** vào **All** .

Nhấp vào **apply** và sau đó nhấp vào **O K**.

Trong onCreate() phương pháp của MainActivity, thêm câu

lệnh sau: Log.d("MainActivity", "Hello World");

The onCreate()

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d("MainActivity", "Hello World");
}
```

Nếu ngăn Logcat chưa mở, hãy nhấp vào **Logcat** ở cuối Android Studio để mở nó.

Kiểm tra xem tên mục tiêu và tên gói của ứng dụng đã chính xác chưa.

Thay đổi Log mức độ trong **Logcat** pane đến **Debug** (hoặc để nguyên như **Verbose** vì có rất ít tin nhắn nhật ký).

Chạy ứng dụng của bạn.

Thông báo sau sẽ xuất hiện trong ngăn Logcat

11-24 14:06:59.001 4696-4696/? D/ Hoạt động chính : Xin chào thế giới

Coding challenge

Lưu ý: Tất cả các thử thách lập trình đều là tùy chọn và không phải là điều kiện tiên quyết cho các bài học sau.

Thách thức: 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 và gửi qua email cho người mà bạn quên mất ngày sinh nhật.
4. Một cách sử dụng phổ biến của Log lớp là để ghi nhật ký J phát hiện các ngoại lệ 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 các phương pháp bạn có thể sử dụng để

bao gồm một ngoại lệ với `Log` tin nhắn . Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại ngoại lệ.

Summary

- Để cài đặt Android Studio, hãy đi tới [Android Studio](#) và làm theo hướng dẫn để tải xuống và cài đặt.
- Khi tạo một ứng dụng mới, hãy đảm bảo rằng **API 15: Android 4.0.3 IceCreamSandwich** được đặt làm SDK tối thiểu.
- Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy nhấp vào **P** tab **dự án** trong cột tab dọc, sau đó chọn **Android** trong menu bật lên ở trên cùng.
- Chỉnh sửa `build.gradle` (Mô-đun: ứng dụng) tệp 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 `src/main` thư mục. Các `java` Thư mục `ava` bao gồm các hoạt động, bài kiểm tra và các thành phần khác trong mã nguồn Java. `res` là thư mục chứa các tài nguyên như bố cục, chuỗi và hình ảnh.
- Chỉnh sửa `AndroidManifest.xml` tệp để thêm các tính năng, thành phần 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 \(AVD\) Android](#) để tạo thiết bị ảo (còn gọi là trình giả lập) để chạy ứng dụng của bạn.
- Thêm `Log` 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 Gỡ lỗi USB trên thiết bị. Mở **Settings > About phone** và gõ **Build number** bảy lần. Quay lại màn hình trước đó (**Settings**) và chạm vào **Developer** . Chọn **USB Debugging**

Related concepts

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

Learn more

Tài liệu Android Studio:

- [Trang tải xuống Android Studio](#)
- [Ghi chú phát hành Android Studio](#)
- [Làm quen với Android Studio](#)
- [Công cụ dòng lệnh Logcat](#)
- [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 ảo Khác:](#)
- [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](#)

Homework

Build and run an app (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 lọc trong **Logcat** để tìm các câu lệnh nhật ký của bạn và điều chỉnh các cấp độ để chỉ hiển thị các câu lệnh ghi nhật ký gỡ lỗi hoặc lỗi.

7. **Trả lời những câu hỏi này**

8. **Câu hỏi 1**

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

- MainActivity.java
- AndroidManifest.xml
- activity_main.xml
- build.gradle

9. **Câu hỏi 2**

Tên của chuỗi tài nguyên chỉ định tên ứng dụng là gì?

- App_name
- xmlns:app
- android:name
- applicationId

10. **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
- Trình biên tập chủ đề

11. **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 câu lệnh " Bố cục MainActivity đã hoàn tất" trong **L ogcat** pane nếu menu Mức nhật ký được đặt thành tùy chọn nào sau đây? (Gợi ý: trả lời nhiều câu hỏi cũng được.)

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

12. Gửi ứng dụng của bạn để chấm điểm

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

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

Giao diện người dùng tương tác đầu tiên

Bài 2) 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 *views* — mọi thành phần của màn hình đều là chữ [V ôi](#). Chữ V tôi lớp 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. V thường được sử dụng tôi các lớp con được mô tả trong nhiều bài học bao gồm:

- [Xem văn bản](#) để hiển thị văn bản.
- [Sửa văn bản](#) cho phép người dùng nhập và chỉnh sửa văn bản.
- [Cái nút](#) và các thành phần có thể nhấp khác (như [R Nút audio](#), [C Hộp quai vật](#), và [S người ghim](#)) để cung cấp hành vi tương tác.
- [CuộnXem](#) và [R ecyclerXem](#) để hiển thị các mục có thể cuộn.

- [Hình ảnhXem](#) để hiển thị hình ảnh.
- [Bố cục ràng buộc](#) và [L trong tai](#) để chứa V khác tôi các yếu tố và định vị chúng.

Mã Java hiển thị và điều khiển UI được chứa trong một lớp mở rộng [A hoạt động](#). MỘT

Hoạt động thường được liên kết với bố cục của chế độ xem UI được định nghĩa là tệp XML (Ngôn ngữ đánh dấu mở rộng). Tệp XML này thường được đặt tên theo A của nó hoạt động và xác định bố cục của V các thành phần trên màn hình.

Ví dụ, MainActivity mã trong ứng dụng Hello World hiển thị bố cục được xác định trong `activity_main.xml` tệp tin bố trí, bao gồm một TextView mở rộng với nội dung "Xin chào thế giới".

Trong các ứng dụng phức tạp hơn, A hoạt động có thể thực hiện các hành động để phản hồi các lần chạm của người dùng, vẽ nội dung đồ họa hoặc yêu cầu dữ liệu từ cơ sở dữ liệu hoặc internet. Bạn tìm hiểu thêm về Hoạt động lớp học ở 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.

2.1) 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.

2.2) Những gì bạn sẽ học được

- Cách tạo ứng dụng có hành vi 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 [V từ vựng và khái niệm vocabulary từ điển](#) f hoặc những định nghĩa thân thiện.

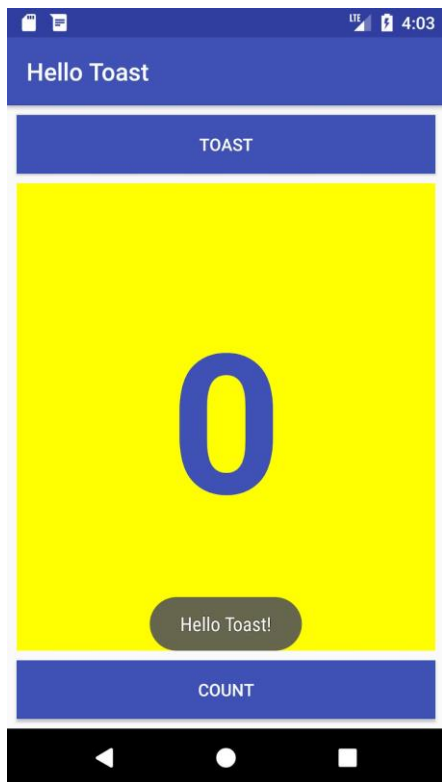
2.3) Bạn sẽ làm gì

- Tạo một ứng dụng và thêm hai B nút các yếu tố và một T Xem mở rộng vào bố cục.
- Thao tác từng phần tử trong [C Bố cục hạn chế](#) 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 B nút .

App overview

Ứng dụng HelloToast bao gồm hai phần tử Button và một TextView. Khi người dùng chạm vào Button đầu tiên, nó sẽ hiển thị một thông báo ngắn (một Toast) trên màn hình. Chạm vào Button thứ hai sẽ tăng một

bộ đếm "nhấp chuột" được hiển thị trong TextView, bắt đầu từ số không



Bài 3) Nhiệm vụ 1: Tạo và khám phá một dự án mới


Trong bài thực hành này, bạn thiết kế và triển khai một dự án cho ứng dụng HelloToast. Một liên kết đến mã giải pháp được cung cấp ở cuối.

3.1) 1.1 Tạo dự án Android Studio

14. Khởi động Android Studio và tạo một dự án mới với các tham số sau:

Thuộc tính	Giá trị
Tên ứng dụng	Xin chào bánh mì nướng
Tên công ty	com.example.android (hoặc tên miền của riêng bạn)
SDK tối thiểu cho điện thoại và máy tính bảng	API15: Android 4.0.3 IceCreamSandwich

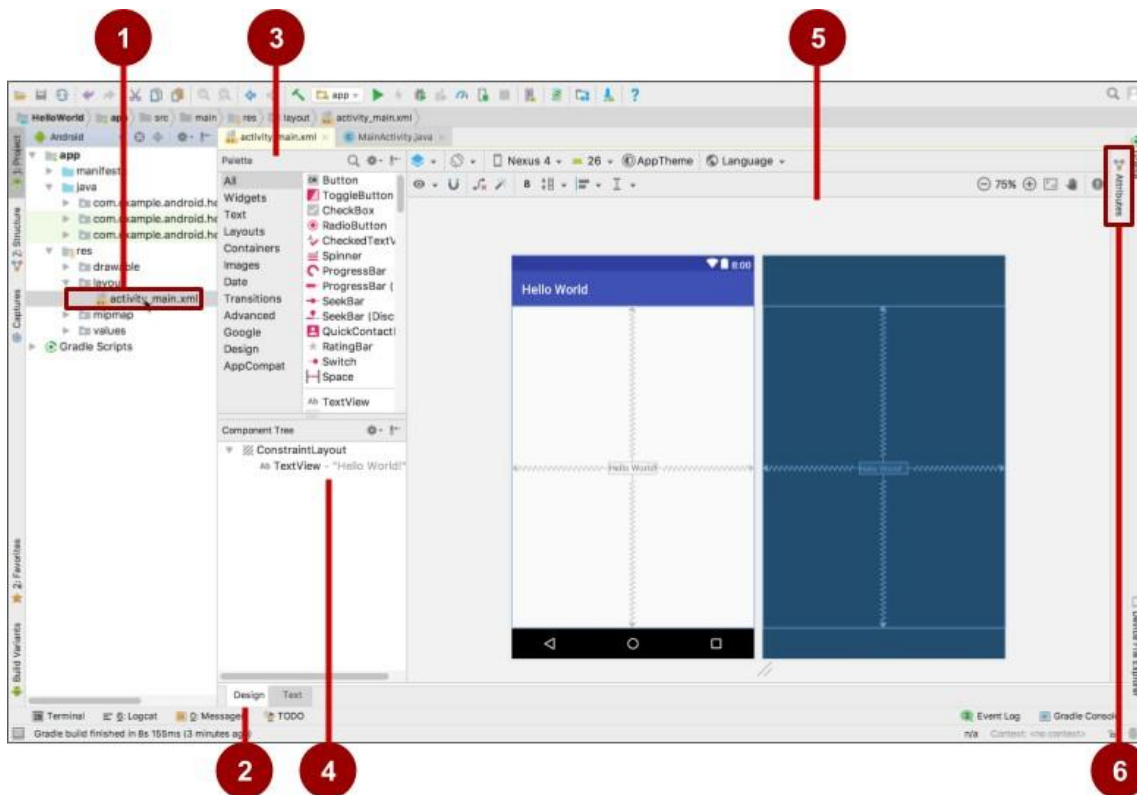
Bản mẫu	Hoạt động trống
Tạo hộp tập tin Bố cục	Đã chọn
Hộp tương thích ngược	Đã chọn

15. Chọn **R un > Chạy ứng dụng** hoặc nhấp vào **R biểu tượng un**  trên thanh công cụ để xây dựng và thực thi ứng dụng trên trình giả lập hoặc thiết bị của bạn.

3.2) 1.2 Khám phá trình chỉnh sửa bố cục

Android Studio cung cấp trình chỉnh sửa bố cục để nhanh chóng xây dựng bố cục của các thành phần giao diện người dùng (UI) của ứng dụng. Nó cho phép bạn kéo các thành phần vào chế độ xem thiết kế trực quan và bản thiết kế, định vị chúng trong bố cục, thêm ràng buộc và đặt thuộc tính. *Constraints* xác định vị trí của một thành phần UI trong bố cục. Ràng buộc thể hiện sự kết nối hoặc căn chỉnh với chế độ xem khác, bố cục cha hoặc hướng dẫn vô hình.

Khám phá trình chỉnh sửa bố cục và tham khảo hình bên dưới khi bạn làm theo các bước được đánh số:



1. Trong một **pp** > **res** > thư mục layout trong **P roject** > **Android** pane, nhấp đúp vào tệp **activity_main.xml** để mở tệp đó nếu tệp đó chưa được mở.
2. Nhấp vào **D** tab **esign** nếu nó chưa được chọn. Bạn sử dụng **D** tab **esign** để thao tác các thành phần và bố cục, và **T** tab **mở rộng** để chỉnh sửa mã XML cho bố cục.
3. Chữ **P** Bảng điều khiển **allettes** hiển thị các thành phần UI mà bạn có thể sử dụng trong bố cục ứng dụng của mình.
4. Chữ **C** Bảng **cây thành phần** hiển thị hệ thống phân cấp chế độ xem của các thành phần UI. V tôi các phần tử được tổ chức thành một hệ thống phân cấp cây gồm cha mẹ và con cái, trong đó một con kế thừa các thuộc tính của cha mẹ nó. Trong hình trên, T Xem mở rộng là con của C Bố cục hạn chế. Bạn sẽ tìm hiểu về những yếu tố này ở phần sau của bài học này.
5. **thị** các thành phần UI trong bố cục. Trong hình trên, bố cục chỉ hiển thị một thành phần: TextView hiển thị "Hello World". 6. A tab **thuộc tính** hiển thị **A** **ngăn thuộc tính** để thiết lập thuộc tính cho phần tử UI.

Mẹo : Xem [B xây dựng giao diện người dùng với Trình chỉnh sửa bố cục](#) để biết chi tiết về cách sử dụng trình chỉnh sửa bố cục và [M Ăn Android Studio](#) để có tài liệu đầy đủ về Android Studio.

Bài 4) Nhiệm vụ 2: Thêm các thành phần View vào trình chỉnh sửa bố cục

Trong tác vụ này, bạn tạo bố cục UI cho ứng dụng HelloToast trong trình chỉnh sửa bố cục bằng cách sử dụng

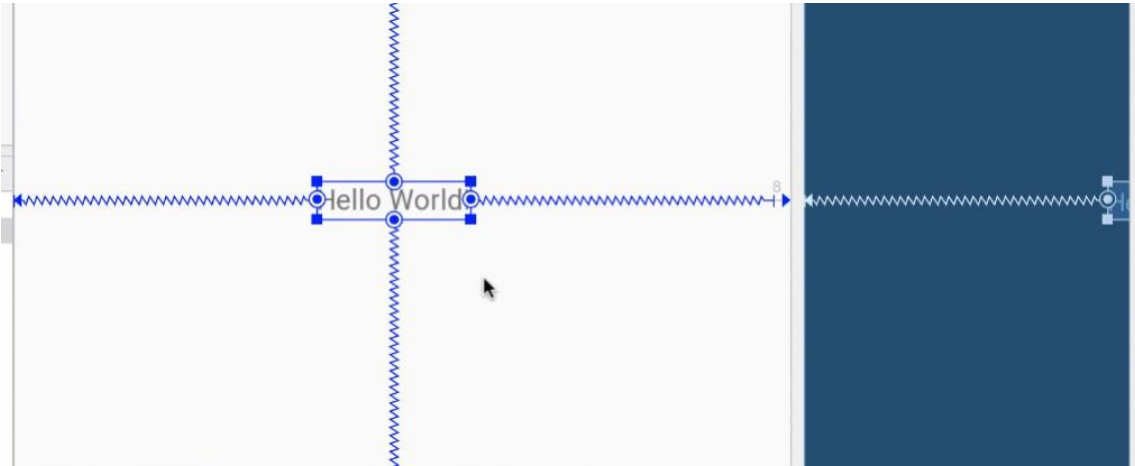
[Bố cục ràng buộc](#) tính năng. Bạn có thể tạo các ràng buộc theo cách thủ công, như được hiển thị sau, hoặc tự động bằng cách sử dụng **A kết nối** uto dụng cụ.

4.1) 2.1 Kiểm tra các ràng buộc phần tử

Thực hiện theo các bước sau:

1. Mở một `ctivity_main.xml` từ **P roject** > ngán Android nếu nó chưa được mở. Nếu **Thiết kế** chưa được chọn, hãy nhấp vào tab đó.
Nếu không có bản thiết kế, hãy nhấp vào **S** chọn nút  **Design Surface** trên thanh công cụ và chọn **D thiết kế điện tử + Bản thiết kế** .
2. Chữ **A** Công cụ **U utoconnect** cũng nằm trong thanh công cụ. Nó được bật theo mặc định. Đối với bước này, hãy đảm bảo rằng công cụ không bị tắt.
3. Nhấp vào  90%  nút phóng to để phóng to bản thiết kế và khung bản thiết kế để xem cận cảnh.
4. Chọn **T extView** trong ngán Cây thành phần. "Hello World" T Xem mở rộng được đánh dấu trong khung thiết kế và bản thiết kế và các ràng buộc cho phần tử có thể nhìn thấy được.
5. Tham khảo hình ảnh động bên dưới cho bước này. Nhấp vào tay cầm tròn ở bên phải chữ T Xem mở rộng để xóa ràng buộc theo chiều ngang liên kết chế độ xem với phía bên phải của bố cục. T Xem mở rộng nhảy sang bên trái vì nó không còn bị giới hạn ở bên phải nữa.

Để thêm lại giới hạn ngang, hãy nhấp vào cùng một tay cầm và kéo một đường sang bên phải của bố cục.



Trong bản thiết kế hoặc khung thiết kế, các tay cầm sau xuất hiện trên T Xem mở rộng yếu tố:

- **Xử lý ràng buộc** : Để tạo ràng buộc như trong hình động ở trên, hãy nhấp vào một tay cầm ràng buộc, được hiển thị dưới dạng một vòng tròn ở bên cạnh một phần tử. Sau đó kéo tay cầm đến một tay cầm ràng buộc khác hoặc đến ranh giới cha. Một đường ngoằn ngoèo biểu thị ràng buộc.



- **Thay đổi kích thước tay cầm** : Để thay đổi kích thước phần tử, hãy kéo các tay cầm thay đổi kích thước hình vuông. Tay cầm sẽ thay đổi thành góc nghiêng khi bạn kéo nó.

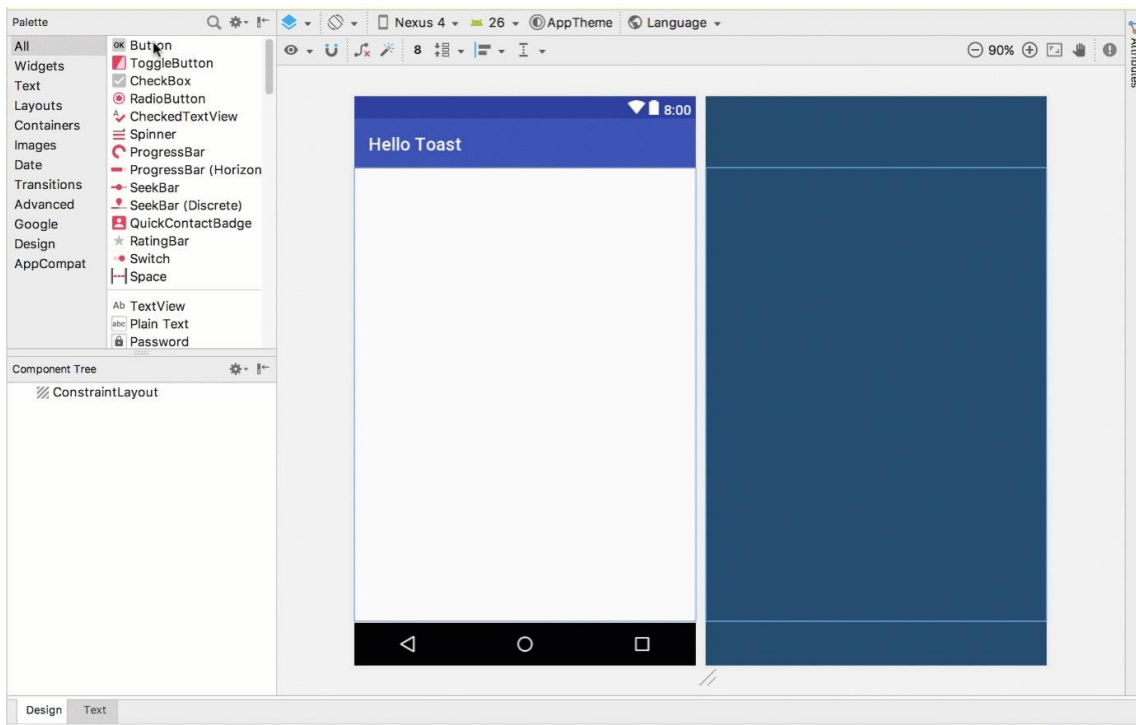


4.2) 2.2 Thêm nút vào bố cục

Khi được bật, **A** Công cụ **utoconnect** tự động tạo hai hoặc nhiều ràng buộc cho một thành phần UI vào bố cục cha. Sau khi bạn kéo thành phần vào bố cục, nó sẽ tạo ra các ràng buộc dựa trên vị trí của thành phần.

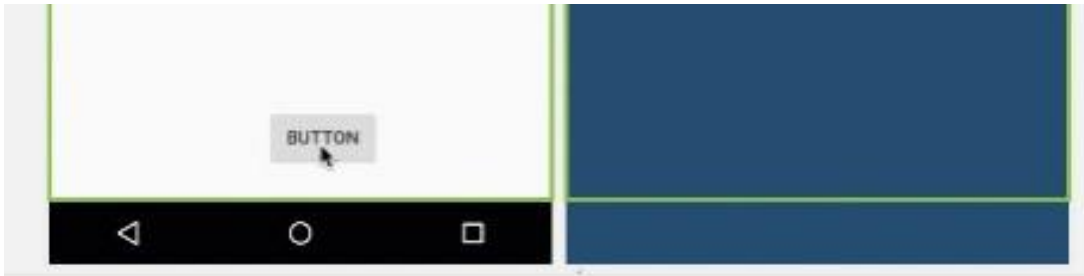
Thực hiện theo các bước sau để thêm B Nút :

1. Bắt đầu với một bảng sạch. T Xem mở rộng phần tử không cần thiết, vì vậy khi nó vẫn được chọn, hãy nhấn **D xóa** phím hoặc chọn **E dit > Xóa** . Bây giờ bạn có một bố cục hoàn toàn trống.
2. Kéo một **B nút** từ **P alette** pane đến bất kỳ vị trí nào trong bố cục. Nếu bạn thả B nút ở vùng giữa trên cùng của bố cục, các ràng buộc có thể tự động xuất hiện. Nếu không, bạn có thể kéo các ràng buộc lên trên cùng, bên trái và bên phải của bố cục như minh họa trong hình động bên dưới.




4.3) 2.3 Thêm Nút thứ hai vào bố cục

1. Kéo thêm một **B** nữa nút từ **P alette** pane vào giữa bố cục như trong hình động bên dưới. Autoconnect có thể cung cấp các ràng buộc theo chiều ngang cho bạn (nếu không, bạn có thể tự kéo chúng).
2. Kéo ràng buộc theo chiều dọc xuống dưới cùng của bố cục (tham khảo hình bên dưới).



Bạn có thể xóa các ràng buộc khỏi một phần tử bằng cách chọn phần tử đó và di con trỏ của bạn

trên đó để hiển thị nút Xóa ràng buộc . Nhấp vào nút này để xóa // ràng buộc trên phần tử đã chọn. Để xóa một ràng buộc duy nhất, hãy nhấp vào tay cầm cụ thể đặt ràng buộc đó.

Để xóa tất cả các ràng buộc trong toàn bộ bố cục, hãy nhấp vào **C** công cụ **All Constraints** trên thanh công cụ. Công cụ này hữu ích nếu bạn muốn làm lại tất cả các ràng buộc trong bố cục của mình.

Bài 5) Nhiệm vụ 3: Thay đổi thuộc tính của phần tử UI

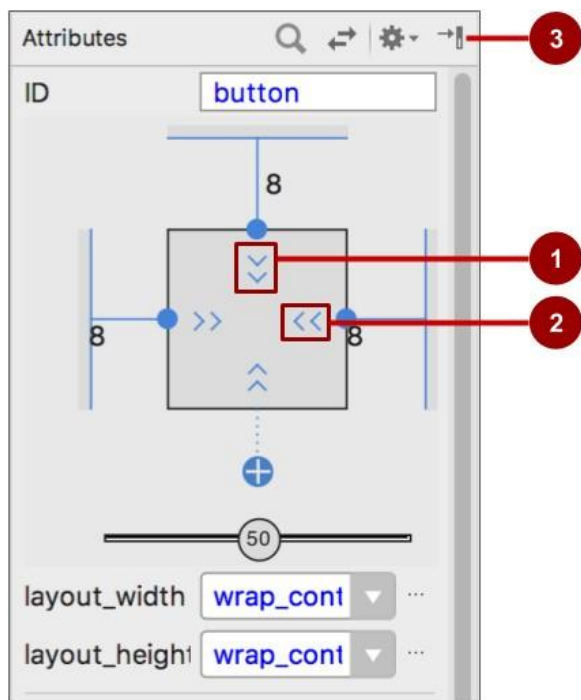
Chữ **A** **ngăn thuộc** tính cung cấp quyền truy cập vào tất cả các thuộc tính XML mà bạn có thể gán cho một phần tử UI. Bạn có thể tìm thấy các thuộc tính (được gọi là *p chức năng*) chung cho tất cả các quan điểm trong [V tài liệu lớp iew](#) .

Trong nhiệm vụ này, bạn nhập các giá trị mới và thay đổi các giá trị cho B quan trọng nút các thuộc tính, áp dụng cho hầu hết V tôi các loại.

5.1) 3.1 Thay đổi kích thước nút

Trình chỉnh sửa bố cục cung cấp các nút điều chỉnh kích thước ở cả bốn góc của chữ V tôi vì vậy bạn có thể thay đổi kích thước View nhanh chóng. Bạn có thể kéo các tay cầm ở mỗi góc của chữ V tôi để thay đổi kích thước, nhưng làm như vậy sẽ mã hóa cứng các kích thước chiều rộng và chiều cao. Tránh mã hóa cứng các kích thước cho hầu hết V tôi các thành phần, vì các kích thước được mã hóa cứng không thể thích ứng với các nội dung và kích thước màn hình khác nhau.

Thay vào đó, hãy sử dụng **A ngăn thuộc tính** ở bên phải của trình chỉnh sửa bố cục để chọn chế độ định cỡ không sử dụng kích thước được mã hóa cứng. **A ngăn thuộc tính** bao gồm một bảng điều khiển kích thước hình vuông được gọi là *thanh tra chế độ xem* ở trên cùng. Các ký hiệu bên trong hình vuông biểu thị các thiết lập chiều cao và chiều rộng như sau:

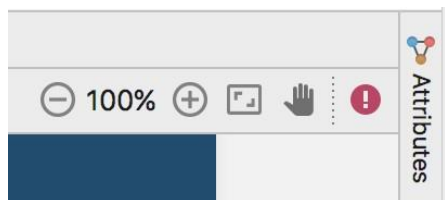


Trong hình trên:

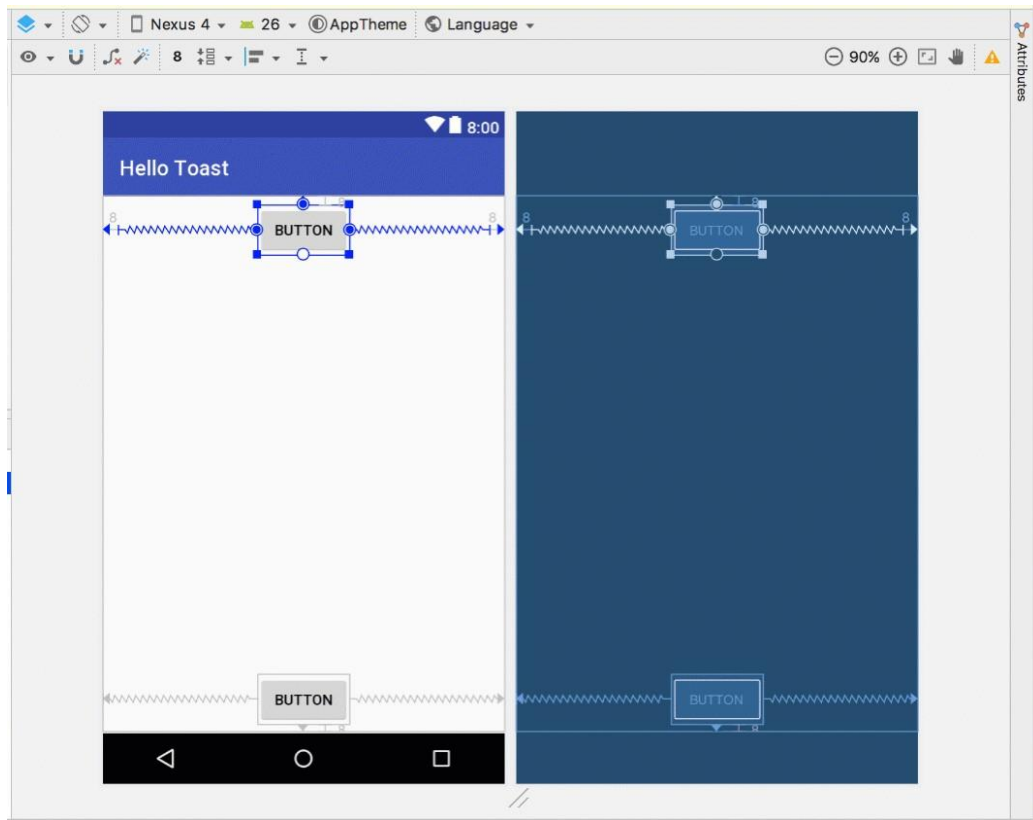
1. **Kiểm soát chiều cao** . Kiểm soát này chỉ định 1 chiều cao thuộc tính và xuất hiện trong hai phân đoạn ở phía trên và phía dưới của hình vuông. Các góc cho biết rằng điều khiển này được đặt thành w nội dung rap , có nghĩa là V tôi sẽ mở rộng theo chiều dọc khi cần để vừa với nội dung của nó. "8" biểu thị lề chuẩn được đặt thành 8dp.
2. **Kiểm soát chiều rộng** . Kiểm soát này chỉ định 1 chiều rộng ngoài và xuất hiện ở hai phân đoạn bên trái và bên phải của hình vuông. Các góc cho biết rằng điều khiển này được đặt thành w nội dung rap , có nghĩa là V tôi sẽ mở rộng theo chiều ngang khi cần thiết để vừa với nội dung của nó, lên đến biên độ 8dp.
3. **thuộc tính** . Nhấp để đóng ngăn.

Thực hiện theo các bước sau:

1. Chọn B trên cùng nút trong **C Bảng cây thành phần** .
2. Nhấp vào **A Tab thuộc tính** ở phía bên phải của cửa sổ trình chỉnh sửa bố cục.

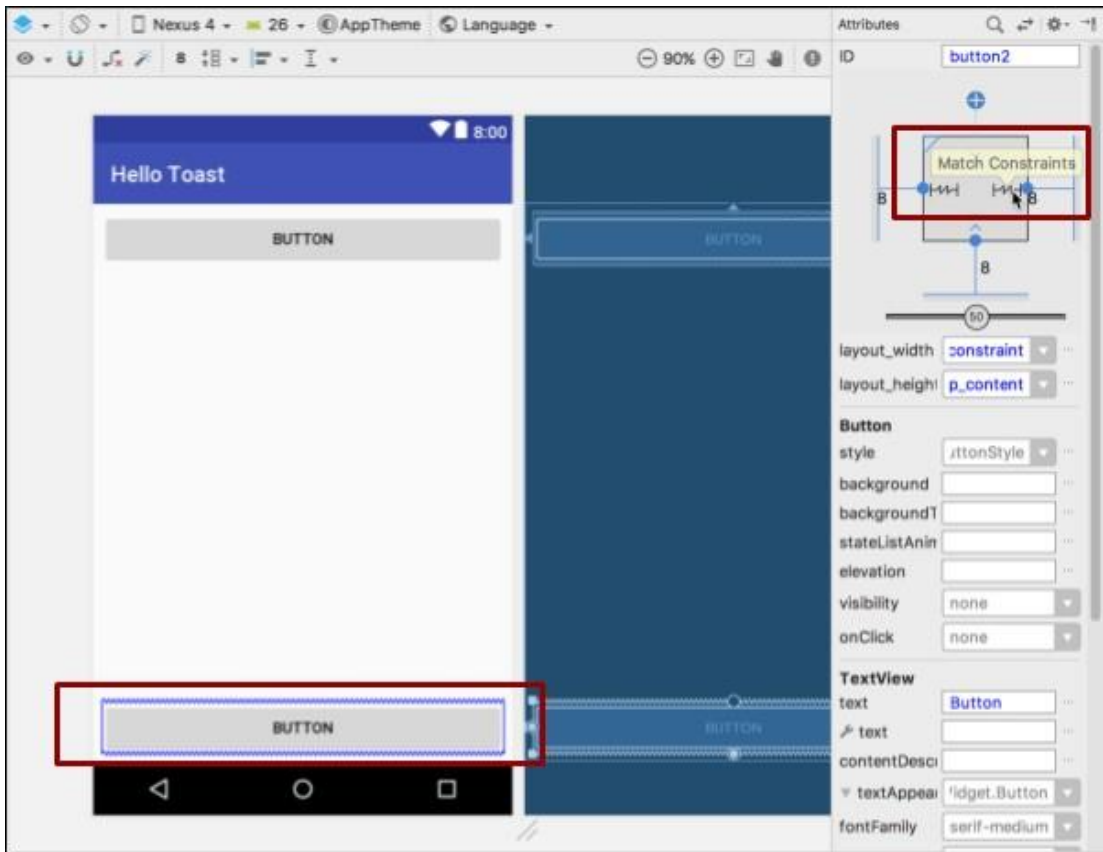


3. Nhấp vào điều khiển chiều rộng hai lần—lần nhấp đầu tiên sẽ thay đổi thành **F được cố định** bằng các đường thẳng và lần nhấp thứ hai sẽ thay đổi thành **M Các ràng buộc** với lò xo cuộn, như thể hiện trong hình ảnh động bên dưới.



Kết quả của việc thay đổi điều khiển chiều rộng, 1 chiều rộng ngoài thuộc tính trong **A ngăn thuộc tính** hiển thị giá trị m ràng buộc atch và B nút phần tử kéo dài theo chiều ngang để lấp đầy khoảng trống giữa bên trái và bên phải của bố cục.

4. B thứ hai nút , và thực hiện những thay đổi tương tự cho 1 chiều rộng ngoài như trong bước trước, như thể hiện trong hình bên dưới.



Như đã trình bày ở các bước trước, 1 chiều rộng ngoài và 1 chiều cao thuộc tính trong **A ngắ n thuộ c** tính thay đổi khi bạn thay đổi các điều khiển chiều cao và chiều rộng trong thanh tra. Các thuộc tính này có thể lấy một trong ba giá trị cho bố cục, đó là `ConstraintLayout` :

- Họ ràng buộc `match` thiết lập mở rộng V tới phần tử để lấp đầy phần tử cha của nó theo chiều rộng hoặc chiều cao—lên đến một lề, nếu có. Phần tử cha trong trường hợp này là `ConstraintLayout` hạn chế . Bạn tìm hiểu thêm về `ConstraintLayout` trong nhiệm vụ tiếp theo.
- Các `wrap_content` thiết lập thu nhỏ V tới kích thước của phần tử sao cho nó vừa đủ lớn để bao bọc nội dung của nó. Nếu không có nội dung, V tới phần tử trở nên vô hình.
- Để chỉ định kích thước cố định phù hợp với kích thước màn hình của thiết bị, hãy sử dụng số lượng [pixel không phụ thuộc vào mật độ cố định](#) (`dp` đơn vị). Ví dụ, `16dp` có nghĩa là 16 pixel không phụ thuộc vào mật độ.

Mẹo : Nếu bạn thay đổi 1 chiều rộng ngoài thuộc tính sử dụng menu bật lên của nó, 1 chiều rộng ngoài thuộc tính được đặt thành số không vì không có kích thước được đặt. Thiết lập này giống như m ràng buộc atch — chế độ xem có thể mở rộng tối đa để đáp ứng các ràng buộc và thiết lập lề.

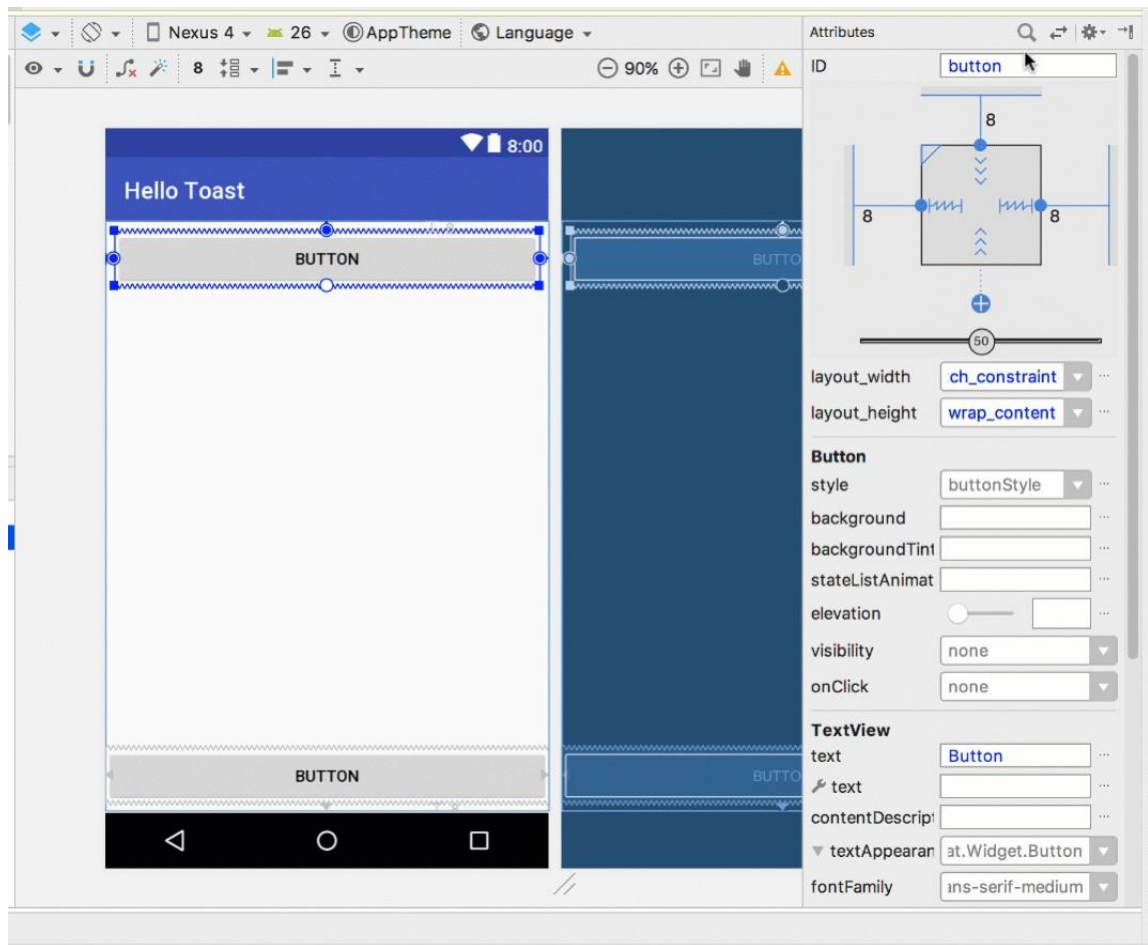
5.2) 3.2 Thay đổi thuộc tính của Button

Để xác định mỗi V tôi duy nhất trong một A hoạt động bố trí, mỗi V tôi hoặc V tôi lớp con (như Button) cần một ID duy nhất. Và để có thể sử dụng, B nút các yếu tố cần có văn bản. V tôi các thành phần cũng có thể có nền có thể là màu sắc hoặc hình ảnh.

Chữ **A ngăn thuộc** tính cung cấp quyền truy cập vào tất cả các thuộc tính bạn có thể gán cho V tôi phần tử. Bạn có thể nhập giá trị cho mỗi thuộc tính, chẳng hạn như `android:id` , `b bối cảnh` , `t màu mở rộng` , và `t thuộc tính mở rộng` .

Hình ảnh hoạt hình sau đây minh họa cách thực hiện các bước này:

1. Sau khi chọn B đầu tiên nút , chỉnh sửa tôi D cánh đồng ở đầu **A thuộc tính** pane cho **button_toast** cho `android:id` thuộc tính được sử dụng để xác định phần tử trong bố cục.
2. Đặt `b bối cảnh` thuộc tính cho **@ màu sắc/màu sắcChính** . (Khi bạn nhập **@ c** , các lựa chọn xuất hiện để lựa chọn dễ dàng.)
3. Đặt `t màu mở rộng` thuộc tính cho **@ android:màu/trắng** .
4. Chỉnh sửa `t mở rộng` thuộc tính cho **T bột mì** .



5. B thứ hai nút , sử dụng **button_count** là ID, **Đếm** cho t mở rộng thuộc tính và màu nền và màu văn bản giống như các bước trước.

Cái c màu sắc chính là màu chính của chủ đề, một trong những màu cơ bản của chủ đề được xác định trước trong `colors.xml` tệp tài nguyên. Nó được sử dụng cho thanh ứng dụng. Sử dụng màu cơ bản cho các thành phần UI khác sẽ tạo ra một UI thống nhất. Bạn sẽ tìm hiểu thêm về chủ đề ứng dụng và Material Design trong bài học khác.

5.3)

5.4) Trình chỉnh sửa bố cục

5.5) Văn bản và các chế độ cuộn

5.6) Tài nguyên có sẵn

Bài 6) Activities

6.1) Activity và Intent

6.2) Vòng đời của Activity và trạng tháiHello WordIntent ngầm định

Bài 7) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ

7.1) Trình gỡ lỗi

7.2) Kiểm thử đơn vị

7.3) 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