

Nhà phát triển Android

Cơ bản

(Khóa học phiên bản 2)

Cập nhật lần cuối vào thứ Ba, ngày 11 tháng 9 năm 2018

Khóa học này được tạo ra bởi đội ngũ Đào tạo Nhà phát triển của Google.

- Để biết thêm chi tiết về khóa học, bao gồm liên kết đến tất cả các chương khái niệm, ứng dụng và slide, xem

Cơ bản về Nhà phát triển Android (Phiên bản 2).

developer.android.com/courses/adf-v2

Lưu ý: Khóa học này sử dụng các thuật ngữ "codelab" và "thực hành" thay thế cho nhau.

Chúng tôi khuyên bạn nên sử dụng phiên bản trực tuyến của khóa học này thay vì

PDF tĩnh này để đảm bảo bạn đang sử dụng nội dung mới nhất.

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

Khóa học Cơ bản về Phát triển Android (V2) – Bài 1

Bài 1: Bắt đầu

Tài liệu PDF này chứa một bản chụp nhanh một lần về các bài học trong Bài 1: Bắt đầu.

Các bài học trong đơn vị này

Bài học 1: Xây dựng ứng dụng đầu tiên của bạn

1.1: Android Studio và Hello World

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

1.2B: Trình chỉnh sửa bố cục

1.3: Văn bản và các chế độ xem cuộn

1.4: Tài nguyên có sẵn

Bài học 2: Hoạt động

2.1: Hoạt động và ý định

2.2: Vòng đời và trạng thái của hoạt động

2.3: Ý định ngầm

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

3.1: Trình gỡ lỗi

3.2: Kiểm tra đơn vị

3.3: Thư viện hỗ trợ

Bài học 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 một trình giả lập và trên một thiết bị vật lý.

Những gì bạn nên biết trước

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm chung cho các ứng dụng hướng đối tượng sử dụng 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 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 máy Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một cách thay thế để tải các phiên bản mới nhất của Android Studio và Java

về máy tính của bạn.

Khóa học Cơ bản về Phát triển Android (V2) – Bài 1

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 mẫu.
- Cách thêm thông điệp ghi log vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

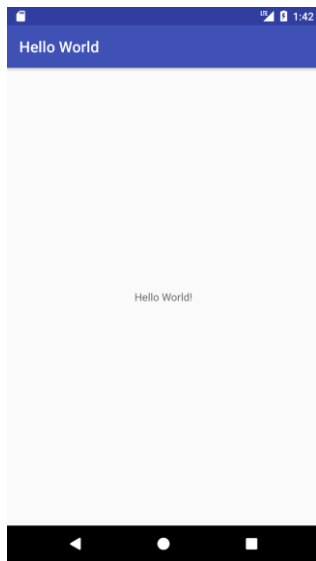
Những gì bạn sẽ làm

- 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 bố cục dự án.
- Tạo và xem các thông điệp ghi log từ ứng dụng của bạn.
- Khám phá tệp AndroidManifest.xml.

Tổng quan về ứng dụng

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

Dưới đây là hình ảnh của ứng dụng hoàn chỉnh:



Nhiệm vụ 1: Cài đặt Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp hoàn chỉnh (IDE) bao gồm một

trình biên tập mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ cho phát triển,

giải quyết lỗi, kiểm tra và hiệu suất giúp việc phát triển ứng dụng nhanh chóng và dễ dàng hơn. Bạn có thể thử nghiệm

các ứng dụng của mình với một loạt các trình giả lập đã được cấu hình sẵn hoặc trên thiết bị di động của riêng bạn, xây dựng

các ứng dụng sản xuất và phát hành trên cửa hàng Google Play.

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

Khóa học Cơ bản về Phát triển Android (V2) – Bài 1

Android Studio có sẵn cho các máy tính chạy Windows hoặc Linux, và cho các máy Mac chạy macOS.

OpenJDK (Bộ công cụ phát triển Java) mới nhất được tích hợp với Android Studio.

Để bắt đầu sử dụng Android Studio, trước tiên hãy kiểm tra yêu cầu hệ thống để đảm bảo rằng hệ thống của bạn đáp ứng được chúng. Việc cài đặt tương tự cho tất cả các nền tảng. Bất kỳ sự khác biệt nào sẽ được ghi chú dưới đây.

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

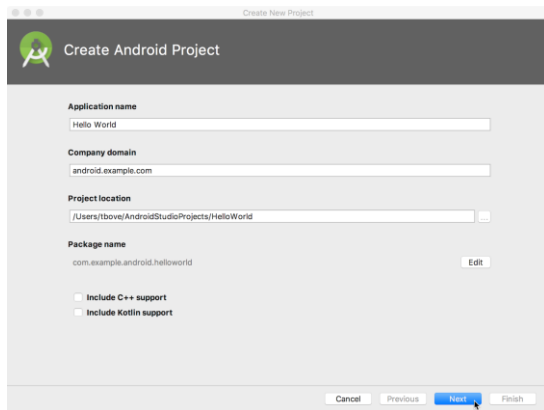
Khắc phục sự cố: Nếu bạn gặp vấn đề với việc cài đặt, hãy kiểm tra ghi chú phát hành của Android Studio, hoặc nhận trợ giúp từ giảng viên của bạn.

Nhiệm vụ 2: Tạo ứng dụng Hello World

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

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu nó chưa được mở.
2. Trong cửa sổ chính Chào mừng đến với Android Studio, nhấp vào Bắt đầu một dự án Android Studio mới.
3. Trong cửa sổ Tạo Dự án Android, nhập Hello World cho Tên ứng dụng. Trang 6.



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

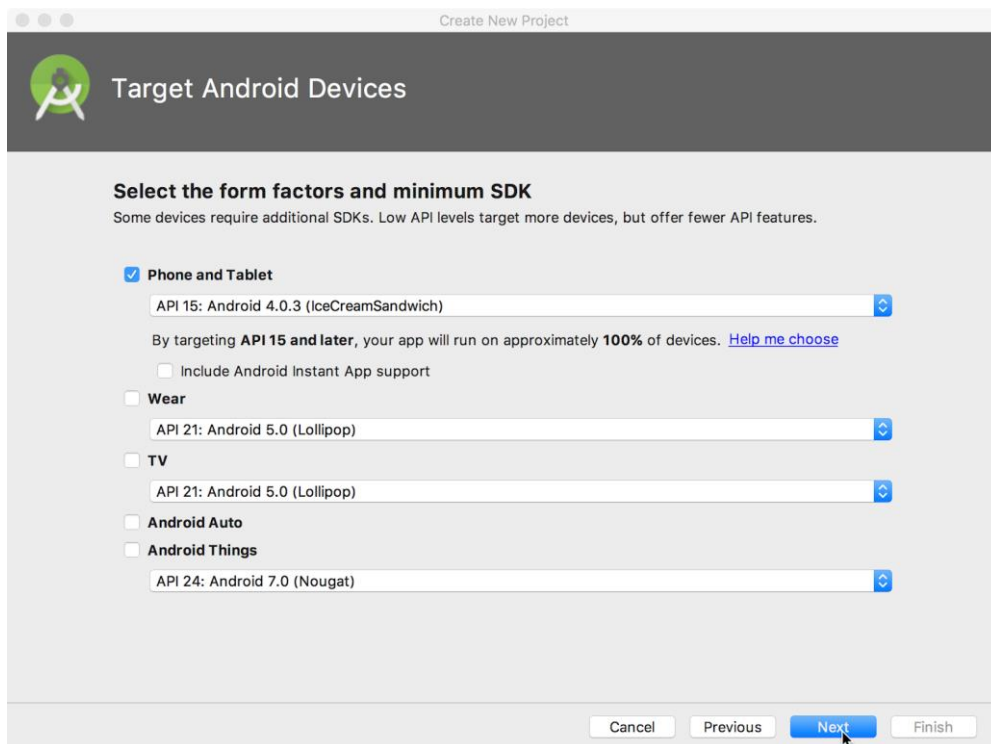
5. Chấp nhận tên miền công ty mặc định android.example.com, hoặc tạo một tên miền công ty độc đáo.

Nếu bạn không có kế hoạch phát hành ứng dụng của mình, bạn có thể chấp nhận mặc định. Hãy lưu ý rằng việc thay đổi tên gói của ứng dụng sau này sẽ tốn thêm công sức.

6. Để không chọn các tùy chọn Bao gồm hỗ trợ C++ và Bao gồm hỗ trợ Kotlin, và nhấp vào Tiếp theo.

7. Trên màn hình Thiết bị Android Mục tiêu, nên chọn Điện thoại và Máy tính bảng. Đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK Tối thiểu; nếu không, hãy sử dụng menu bật lên để đặt nó.

Trang 7.

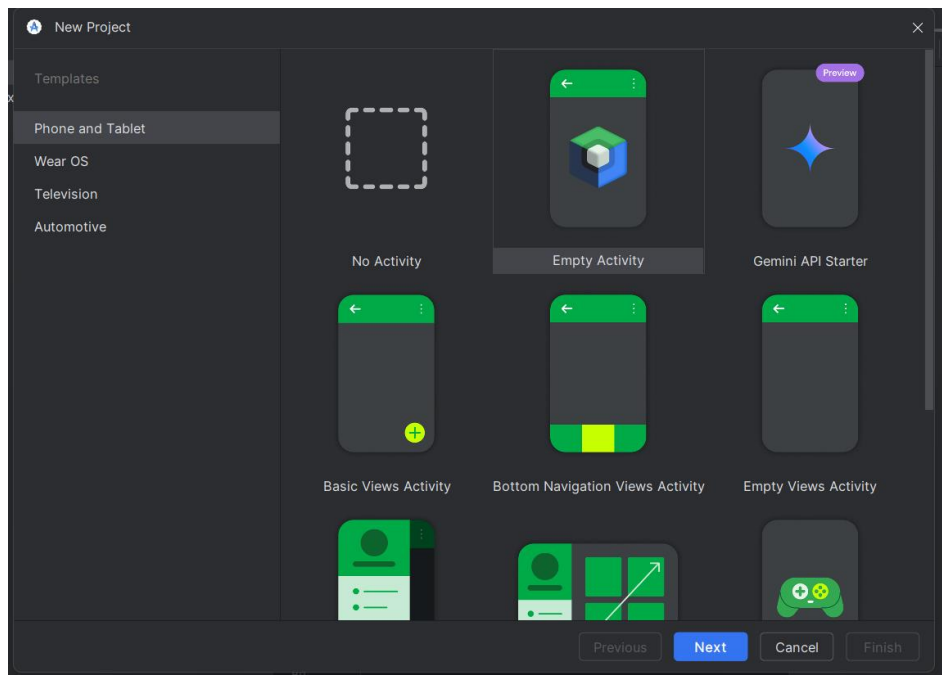


Đây là các cài đặt được sử dụng bởi các ví dụ trong các bài học cho khóa học này. Tính đến thời điểm viết bài này, các cài đặt này làm cho ứng dụng Hello World của bạn tương thích với 97% các thiết bị Android đang hoạt động trên Google Play Store.

8. Bỏ chọn tùy chọn Bao gồm hỗ trợ Ứng dụng Nhanh và tất 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.

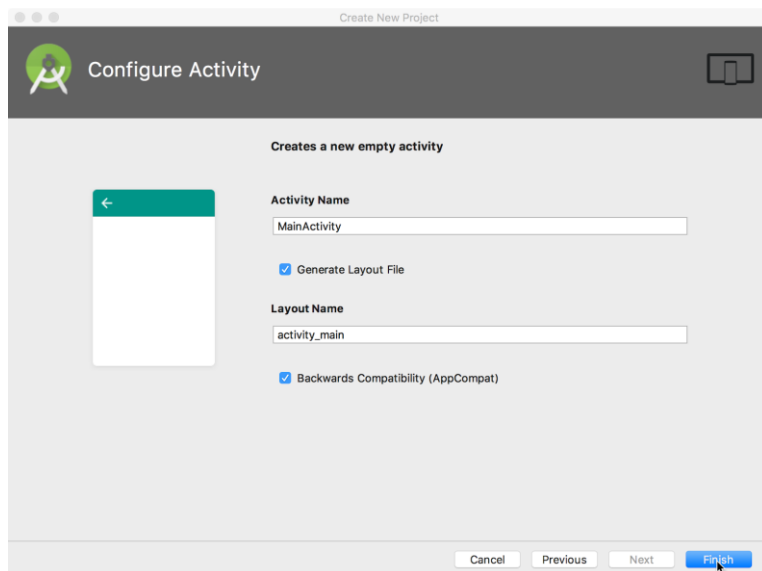
9. Cửa sổ Thêm một Hoạt động xuất hiện. Một Hoạt động là một việc duy nhất, tập trung mà người dùng có thể thực hiện. Đây là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một Hoạt động thường có một bố cục liên quan đến nó, xác định cách các phần tử giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp các mẫu Hoạt động để giúp bạn bắt đầu. Đối với dự án Hello World, chọn Hoạt động Trống như hình dưới đây và nhấp vào Tiếp theo.

Trang 8.



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

Trang 9.



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

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

13. Nhấp vào Hoàn tất (Finish).

Android Studio tạo một thư mục cho các dự án của bạn và xây dựng dự án với Gradle (điều này có thể mất một chút thời gian).

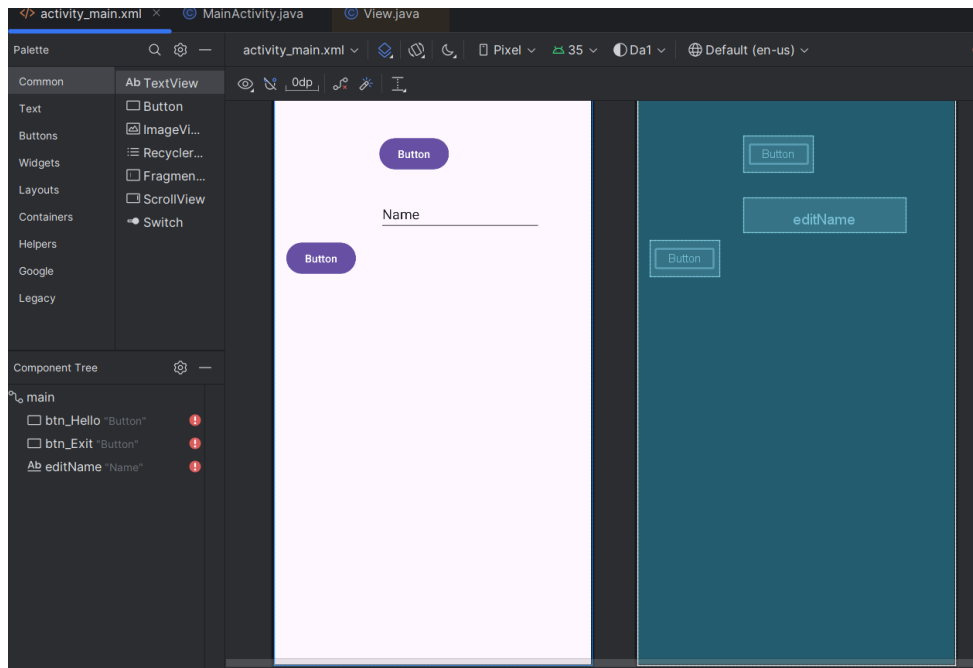
Mẹo: Xem trang cấu hình xây dựng của bạn để biết thông tin chi tiết.

Bạn cũng có thể thấy một thông điệp "Mẹo trong ngày" với các phím tắt và mẹo hữu ích khác. Nhấp vào Đóng (Close) để đóng thông điệp.

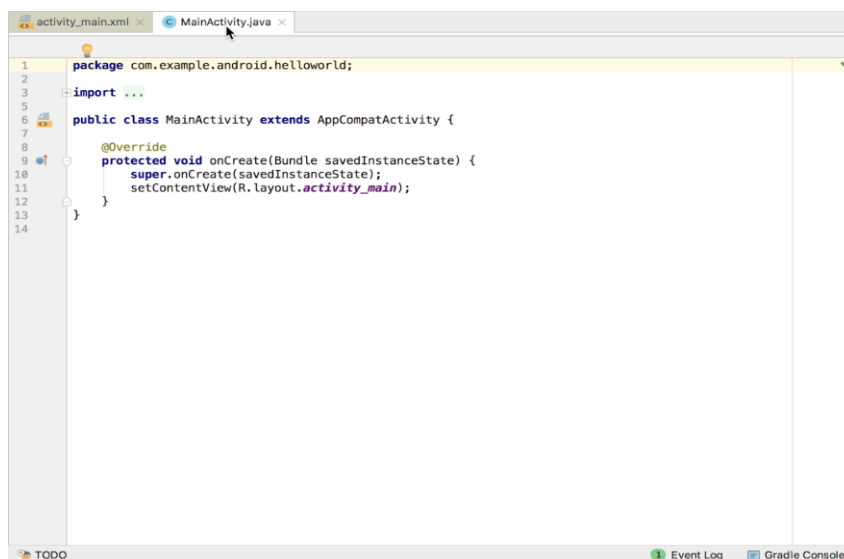
Trình soạn thảo Android Studio xuất hiện. Thực hiện các bước sau:

1. Nhấp vào tab `activity_main.xml` để xem trình soạn thảo bố cục.
2. Nhấp vào tab Thiết kế (Design) của trình soạn thảo bố cục, nếu chưa được chọn, để hiển thị một hình ảnh đồ họa của bố cục như hình dưới đây.

Trang 10.



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



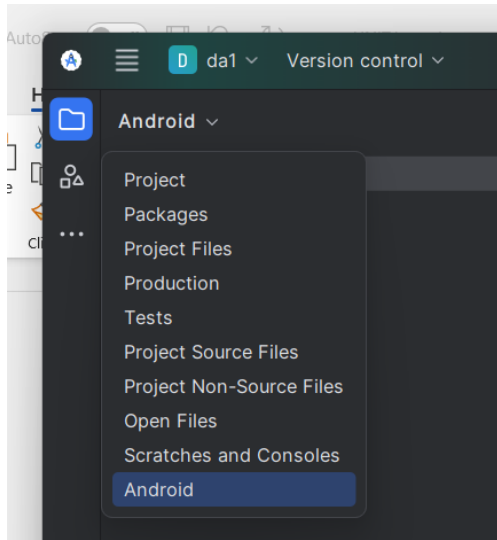
2.2 Khám Phá Dự Án > Bảng Android

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

1. Nếu chưa được chọn, hãy nhấp vào tab Dự Án trong cột tab dọc ở bên trái của cửa sổ Android Studio. Bảng Dự Án sẽ xuất hiện.

2. Để xem dự án theo cấu trúc phân cấp dự án Android tiêu chuẩn, hãy chọn Android từ menu bật lên ở đầu Bảng Dự Án, như hình dưới đây.

Trang 12.



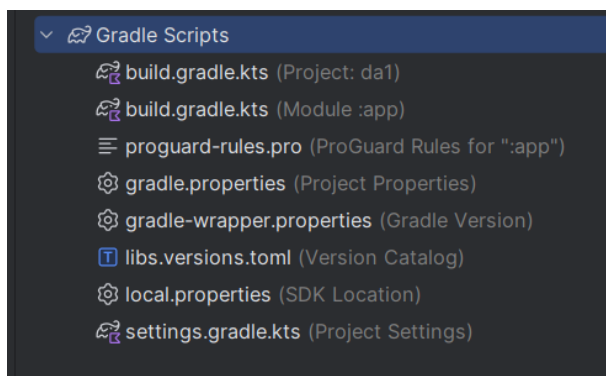
Lưu ý: Chương này và các chương khác đề cập đến bảng Dự án, khi được đặt ở chế độ Android, là bảng Dự án > Android.

2.3 Khám phá thư mục Gradle Scripts

Hệ thống xây dựng Gradle trong Android Studio giúp bạn dễ dàng bao gồm các nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng của bạn dưới dạng phụ thuộc.

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

Trang 13.



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

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

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

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

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô-đun

tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một tệp Gradle xây dựng cấp cao nhất duy nhất.

Hầu hết thời gian, bạn sẽ không cần phải thực hiện bất kỳ thay đổi nào đối với tệp này, nhưng vẫn

hữu ích để hiểu nội dung của nó.

Theo mặc định, tệp xây dựng cấp cao nhất sử dụng khối buildscript để định nghĩa các kho lưu trữ và phụ thuộc của Gradle mà là chung cho tất cả các mô-đun trong dự án. Khi phụ thuộc của bạn là một thứ gì đó khác ngoài thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong bất kỳ kho trực tuyến nào được chỉ định trong khối repositories của tệp này. Theo mặc định, các dự án Android Studio mới khai báo JCenter và Google (bao gồm cả kho Maven của Google) là các vị trí kho lưu trữ:

```
allprojects {  
  
    repositories {  
  
        google()  
  
        jcenter()  
  
    }  
  
}
```

Trang14.

3. Tìm tệp build.gradle (Module:app).

Ngoài tệp build.gradle cấp dự án, mỗi module đều có tệp build.gradle riêng, cho phép bạn cấu hình các cài đặt xây dựng cho từng module cụ thể (ứng dụng HelloWorld chỉ có một module). Cấu hình các cài đặt xây dựng này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như các loại xây dựng bổ sung và hương vị sản phẩm. Bạn cũng có thể ghi đè các cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp cao nhất.

Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phụ thuộc trong phần phụ thuộc. Bạn có thể khai báo một phụ thuộc thư viện

sử dụng một trong nhiều cấu hình phụ thuộc khác nhau. Mỗi cấu hình phụ thuộc

cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ, câu

lệnh implementation fileTree(dir: 'libs', include: ['*.jar']) thêm một

phụ thuộc cho tất cả các tệp “*.jar” bên trong thư mục libs.

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

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "com.example.android.helloworld"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
```

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
This PDF is a one-time snapshot. See developer.android.com/courses/fundamentals-training/toc-v2 for the latest updates.

Page 15

Android Developer Fundamentals Course (V2) – Unit 1

```
androidTestImplementation 'com.android.support.test:runner:1.0.1'
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```

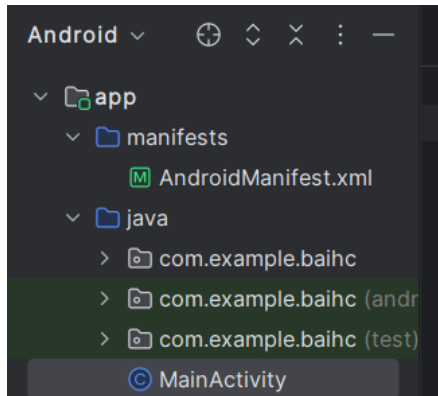
Trang 15.

4. Nhấp vào hình tam giác để đóng Gradle Scripts.

2.4 Khám phá các thư mục app và res

Tất cả mã và tài nguyên cho ứng dụng đều 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 xem tệp MainActivity.java. Nhấp đúp vào tệp để mở nó trong trình chỉnh sửa mã.

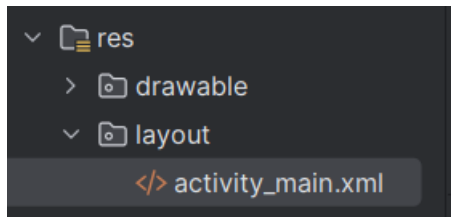


Thư mục java bao gồm các tệp lớp Java trong ba thư mục con, như được hiển thị trong hình trên.

Thư mục com.example.hello.helloworld (hoặc tên miền mà bạn đã chỉ định)

chứa tất cả các tệp cho một gói ứng dụng. Hai thư mục còn lại được sử dụng cho việc 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 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 bị bỏ qua trong Project > Android pane).

2. Mở rộng thư mục res và thư mục layout, sau đó nhấp đúp vào tệp activity_main.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. Một Activity thường

được liên kết với một bố cục của các giao diện người dùng được định nghĩa dưới dạng tệp XML. Tệp này thường được đặt tên theo

Activity của nó.

2.5 Khám phá thư mục manifests

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

Android, mà hệ thống cần có 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 của bạn. Tất cả các thành phần

của một ứng dụng, chẳng hạn như mỗi Activity, phải được khai báo trong tệp XML này. Trong các bài học khác, bạn sẽ

chỉnh sửa tệp này để thêm các tính năng và quyền truy cập tính năng. Để tìm hiểu thêm, xem Tổng quan về Manifest Ứng dụng.

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

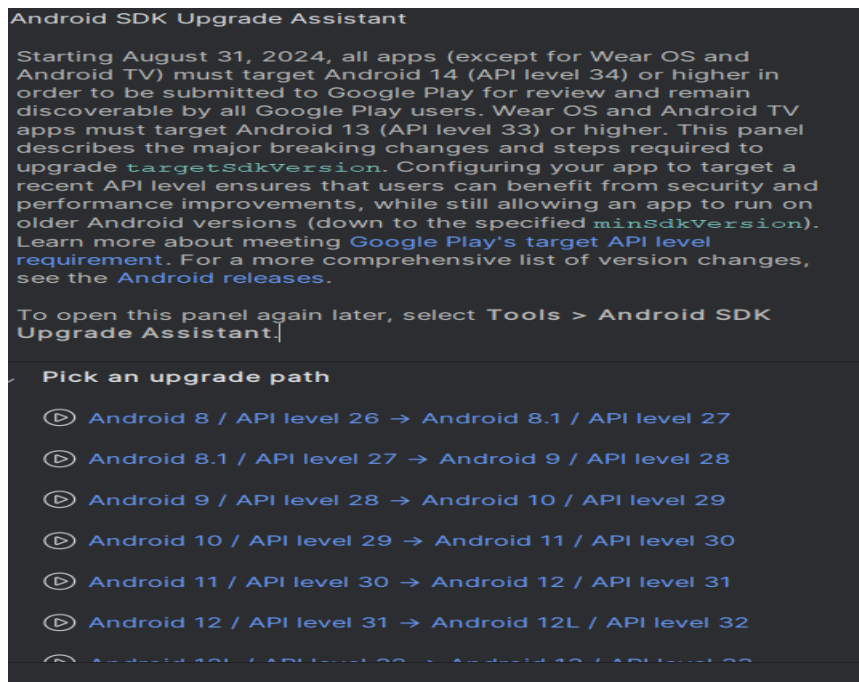
Trong nhiệm vụ này, bạn sẽ sử dụng trình quản lý Thiết bị Ảo Android (AVD) để tạo một thiết bị ảo (còn được gọi là trình giả lập) mô phỏng cấu hình cho một loại thiết bị Android cụ thể, và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng Trình giả lập Android có các yêu cầu bổ sung ngoài các yêu cầu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

Sử dụng Trình quản lý AVD, bạn xác định các đặc điểm phần cứng của một thiết bị, cấp độ API, bộ nhớ, giao diện và các thuộc tính khác và lưu nó dưới dạng một thiết bị ảo. Với các thiết bị ảo, bạn có thể thử nghiệm 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 các thiết bị vật lý.

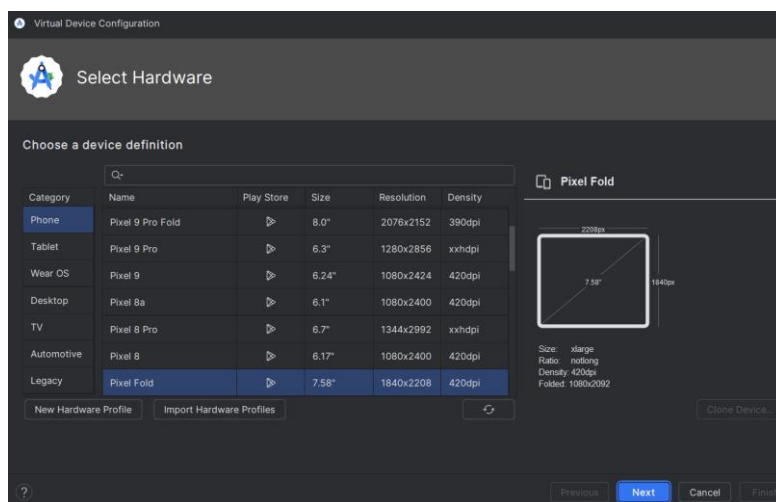
3.1 Tạo một thiết bị ảo Android (AVD)

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

1. Trong Android Studio, chọn Công cụ > Android > Trình quản lý AVD, hoặc nhấp vào biểu tượng Trình quản lý AVD trên thanh công cụ. Màn hình Thiết bị ảo của bạn xuất hiện. Nếu bạn đã tạo thiết bị ảo, màn hình sẽ hiển thị chúng (như hình dưới đây); nếu không, bạn sẽ thấy một danh sách trống.

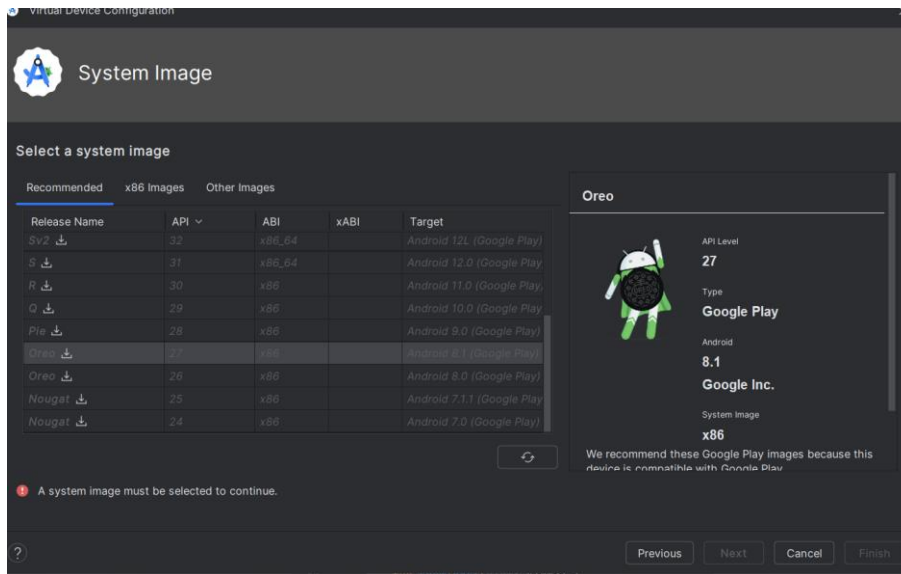


2. Nhấp vào +Tạo Thiết Bị Ảo. Cửa sổ Chọn Phần Cứng xuất hiện hiển thị danh sách các thiết bị phần cứng đã được cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước màn hình chéo (Kích thước), độ phân giải màn hình tính bằng pixel (Độ phân giải) và mật độ pixel (Mật độ).



3. Chọn một thiết bị như Nexus 5x hoặc Pixel XL, sau đó nhấp vào Tiếp theo. Màn hình Hình ảnh Hệ thống sẽ xuất hiện.

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



Có nhiều phiên bản khác có sẵn hơn là những gì được hiển thị trong tab Đề xuất. Hãy xem các tab x86

Hình ảnh và Hình ảnh Khác để xem chúng.

Nếu có liên kết Tải xuống hiển thị bên cạnh một hình ảnh hệ thống mà bạn muốn sử dụng, nó chưa được cài đặt. Nhấp vào

liên kết để bắt đầu tải xuống, và nhấp vào Hoàn tất khi quá trình hoàn tất.

5. Sau khi chọn một hình ảnh hệ thống, nhấp vào Tiếp theo. Cửa sổ Thiết bị Ảo Android (AVD) sẽ xuất hiện.

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

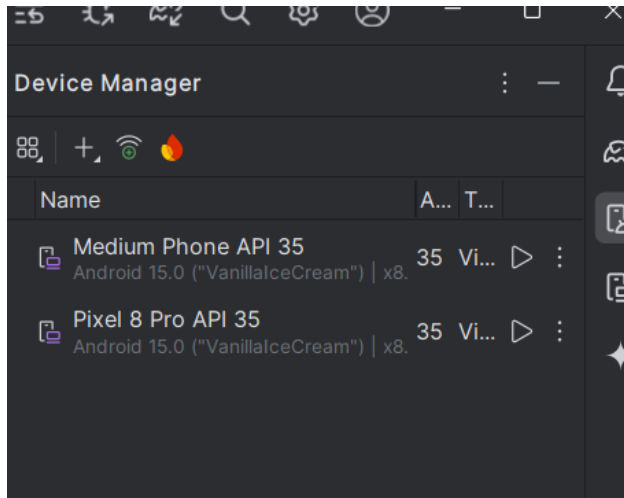
3.2 Chạy ứng dụng trên thiết bị ảo

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

Trang 20.

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, dưới Thiết bị Ảo Có sẵn, chọn thiết bị ảo mà bạn vừa tạo và nhấp OK.

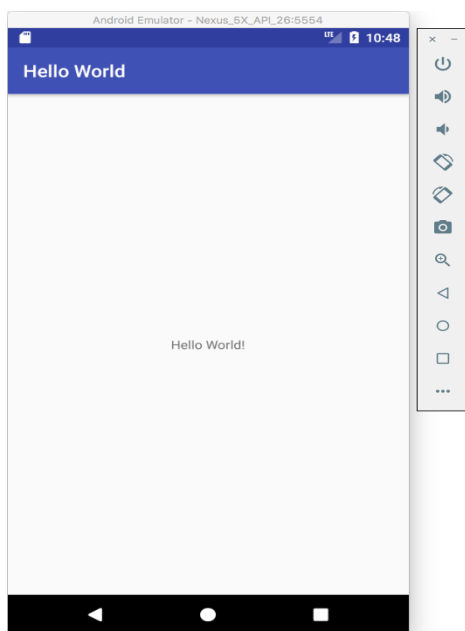


Trình giả lập khởi động và boot giống như một thiết bị vật lý. Tùy thuộc vào tốc độ của máy tính của bạn,

điều này có thể mất một thời gian. Ứng dụng của bạn được xây dựng, và khi trình giả lập sẵn sàng, Android Studio sẽ tải

thư ứng dụng lên trình giả lập và chạy nó.

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



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

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

Trong nhiệm vụ cuối cùng này, bạn sẽ chạy ứng dụng của mình trên một 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 thiết bị vật lý.

Những gì bạn cần:

- Một thiết bị Android như điện thoại hoặc máy tính bảng.
- Một 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 thêm các bước để chạy trên thiết bị phần cứng. Kiểm tra tài liệu Sử dụng Thiết bị Phần cứng. Bạn cũng có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của bạn. Đối với trình điều khiển USB trên Windows, xem Trình điều khiển USB OEM.

4.1 Bật gỡ lỗi USB

Để cho Android Studio giao tiếp với thiết bị của bạn, bạn phải bật Gỡ lỗi USB trên thiết bị Android của mình. Tùy chọn này được kích hoạt trong cài đặt Tùy chọn nhà phát triển của thiết bị.

Trên Android 4.2 và cao hơn, 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, mở Cài đặt, tìm kiếm Thông tin điện thoại, nhấp vào Thông tin 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. Chạm vào Tùy chọn nhà phát triển.

3. Chọn Gỡ lỗi USB.

Trang 23.

4.2 Chạy ứng dụng của bạn trên thiết bị

Bây giờ bạn có thể kết nối thiết bị của mình 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 sẽ mở ra với danh sách các trình giả lập và thiết bị đã kết nối.

3. Chọn thiết bị của bạn và nhấp OK.

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

Khắc phục sự cố

Nếu Android Studio của bạn không nhận diện được thiết bị, hãy thử các bước sau:

1. Rút và cắm lại thiết bị của bạn.

2. Khởi động lại Android Studio.

Nếu máy tính của bạn vẫn không tìm thấy thiết bị hoặc thông báo "không được ủy quyền", hãy làm theo các bước sau:

1. Rút thiết bị ra.

2. Trên thiết bị, mở Tùy chọn nhà phát triển trong ứng dụng Cài đặt.

3. Nhấn vào Thu hồi quyền ủy quyền gỡ lỗi USB.

4. Kết nối lại thiết bị với máy tính của bạn.

5. Khi được nhắc, hãy cấp quyền ủy quyền.

Bạn có thể cần cài đặt driver USB phù hợp cho thiết bị của mình. Xem tài liệu Sử dụng Thiết bị Phần cứng.

Trang 24.

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

Trong nhiệm vụ này, bạn sẽ thay đổi một số thông tin về cấu hình ứng dụng trong tệp `build.gradle(Module:app)` để học cách thực hiện các 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

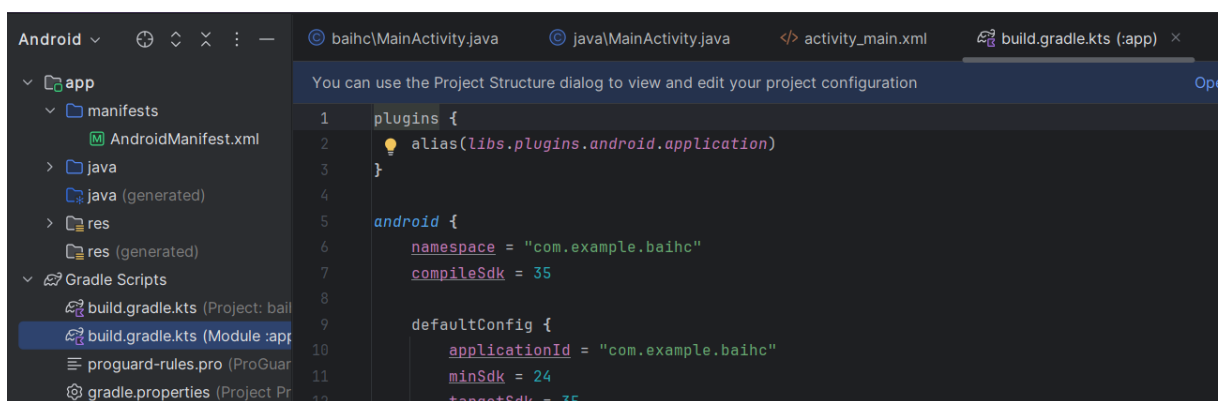
Thực hiện theo các bước sau:

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

Nội dung của tệp sẽ xuất hiện trong trình chỉnh sửa mã.

2. Trong khối `defaultConfig`, thay đổi giá trị của `minSdkVersion` thành 17 như hình dưới đây

(nó ban đầu được đặt là 15).



Trình biên tập mã hiển thị một thanh thông báo ở trên cùng với liên kết đồng bộ ngay bây giờ.

Trang 25.

5.2 Đồng bộ cấu hình Gradle mới

Khi bạn thực hiện thay đổi trong các tệp cấu hình xây dựng của một dự án, Android Studio yêu cầu bạn

đồng bộ các tệp dự án để nó có thể nhập các thay đổi cấu hình xây dựng và thực hiện một số kiểm tra

để đảm bảo rằng cấu hình sẽ không tạo ra lỗi xây dựng.

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

(như được hiển thị trong hình trước), hoặc nhấp vào biểu tượng Đồng bộ dự án với tệp Gradle trong

thanh công cụ.

Khi quá trình đồng bộ Gradle hoàn tất, thông báo Xây dựng Gradle đã hoàn thành sẽ xuất hiện ở góc dưới bên trái

của cửa sổ Android Studio.

Để tìm hiểu sâu hơn về Gradle, hãy tham khảo tài liệu Tổng quan về Hệ thống Xây dựng và Cấu hình Xây dựng Gradle.

Nhiệm vụ 6: Thêm các câu lệnh log vào ứng dụng của bạn

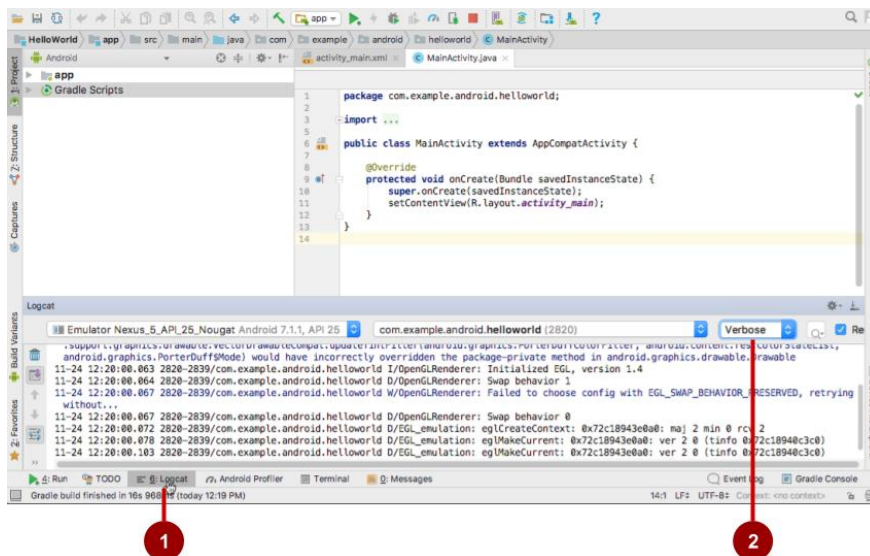
Trong nhiệm vụ này, bạn sẽ thêm các câu lệnh Log vào ứng dụng của mình, hiển thị các thông điệp trong bảng Logcat.

Các thông điệp log là một công cụ gỡ lỗi mạnh mẽ mà bạn có thể sử dụng để kiểm tra các giá trị, các đường dẫn thực thi,

và báo cáo các ngoại lệ.

6.1 Xem bảng Logcat

Để xem bảng Logcat, hãy nhấp vào tab Logcat ở dưới cùng của cửa sổ Android Studio như được hiển thị trong hình dưới đây. Trang 26.



Trong hình trên:

1. Tab Logcat để mở và đóng bảng Logcat, hiển thị thông tin về ứng dụng của bạn khi nó đang chạy. Nếu bạn thêm các câu lệnh Log vào ứng dụng của mình, các thông điệp Log sẽ xuất hiện ở đây.
2. Menu mức Log được đặt thành Verbose (mặc định), hiển thị tất cả các thông điệp Log. Các cài đặt khác bao gồm Debug, Error, Info và Warn.

6.2 Thêm các câu lệnh log vào ứng dụng của bạn

Các câu lệnh log trong mã ứng dụng của bạn hiển thị thông điệp trong bảng Logcat. Ví dụ:

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

Các phần của thông điệp là:

- Log: Lớp Log để gửi các thông điệp log đến bảng Logcat.
- d: Cài đặt mức độ Log Debug để lọc hiển thị thông điệp log trong bảng Logcat. Các mức log khác là e cho Lỗi, w cho Cảnh báo, và i cho Thông tin.
- "MainActivity": Tham số đầu tiên là một thẻ có thể được sử dụng để lọc các thông điệp trong bảng Logcat. Đây thường là tên của Activity từ đó thông điệp xuất phát. Tuy nhiên, bạn có thể đặt tên này thành bất cứ điều gì hữu ích cho bạn trong việc gỡ lỗi.

Theo quy ước, các thẻ log được định nghĩa là hằng số cho Activity:

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

- "Xin chào thế giới": Tham số thứ hai là thông điệp thực tế.

Thực hiện các bước sau:

1. Mở ứng dụng Hello World của bạn trong Android Studio và mở MainActivity.
2. Để tự động thêm các import không mở hồ vào dự án của bạn (chẳng hạn như android.util.Log cần thiết để sử dụng Log), chọn Tập > Cài đặt trong Windows, hoặc Android Studio > Tùy chọn trong macOS.
3. Chọn Trình soạn thảo > Chung > Tự động Nhập. Chọn tất cả các ô kiểm tra và đặt Chèn import khi dán thành Tất cả.
4. Nhấp vào Áp dụng và sau đó nhấp vào OK.
5. Trong phương thức onCreate() của MainActivity, thêm câu lệnh sau:

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

Phương thức onCreate() bây giờ nên trông giống như mã sau:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

Trang 28.

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

6. Nếu bảng Logcat chưa mở, hãy nhấp vào tab Logcat ở dưới cùng của Android Studio để mở nó.

7. Kiểm tra xem tên của mục tiêu và tên gói của ứng dụng có đúng không.

8. Thay đổi mức độ Log trong bảng Logcat thành Debug (hoặc để nguyên là Verbose vì có rất ít thông điệp log).

9. Chạy ứng dụng của bạn.

Thông điệp sau đây sẽ xuất hiện trong bảng Logcat:

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

Thử thách lập trình

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.

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

1. Tạo một dự án mới trong Android Studio.
2. Thay đổi lời chào "Hello World" thành "Chúc mừng sinh nhật" và tên của một người có sinh nhật gần đây.
3. (Tùy chọn) Chụp màn hình ứng dụng hoàn thành của bạn và gửi email cho ai đó mà bạn đã quên sinh nhật.
4. Một cách sử dụng phổ biến của lớp Log là ghi lại các ngoại lệ Java khi chúng xảy ra trong chương trình của bạn. Có một số phương thức 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 một thông điệp Log. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại một ngoại lệ.

Trang 29.

Tóm tắt :

- Để cài đặt Android Studio, hãy truy cập 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 Minimum SDK.
- Để xem cấu trúc Android của ứng dụng trong bảng 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 các thư mục app và res. Thư mục java 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 và quyền cho ứng dụng Android của bạn. Tất cả các thành phần của một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML này.
- Sử dụng trình quản lý Android Virtual Device (AVD) để tạo một thiết bị ảo (còn được gọi là trình giả lập) để chạy ứng dụng của bạn.
- Thêm các câu lệnh Log vào ứng dụng của bạn, hiển thị thông điệp trong bảng Logcat như một công cụ cơ bản để gỡ lỗi.
- Để chạy ứng dụng của bạn trên một thiết bị Android vật lý bằng Android Studio, hãy bật Gỡ lỗi USB trên thiết bị. Mở Cài đặt > Giới thiệu về điện thoại và nhấn vào Số bản dựng bảy lần. Quay lại màn hình trước đó (Cài đặt), và nhấn vào Tùy chọn nhà phát triển. Chọn Gỡ lỗi USB.

Các khái niệm liên quan

Tài liệu về các khái niệm liên quan có trong 1.0: Giới thiệu về Android và 1.1 Ứng dụng Android đầu tiên của bạn.

Trang 30.

Tìm hiểu thêm

Tài liệu Android Studio:

- Trang tải xuống Android Studio
- Ghi chú phát hành Android Studio
- Giới thiệu về Android Studio
- Công cụ dòng lệnh Logcat
- Trình quản lý Thiết bị Ảo Android (AVD)
- Tổng quan về Manifest ứng dụng
- Cấu hình bản dựng của bạn
- Lớp Log
- 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 về 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 log cho các mức độ log khác nhau trong onCreate() của hoạt động chính.
- Tạo một trình giả lập cho một thiết bị, nhắm đến bất kỳ phiên bản Android nào bạn thích, và chạy ứng dụng.
- Sử dụng bộ lọc trong Logcat để tìm các câu lệnh ghi log của bạn và điều chỉnh các mức độ chỉ hiển thị các câu lệnh ghi log debug hoặc lỗi.

Trang 31.

Trả lời các câu hỏi

Câu hỏi 1:

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

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

Câu hỏi 2:

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

- app_name
- xmlns:app
- android:name
- applicationId

Câu hỏi 3

Công cụ nào bạn sử dụng để tạo một trình giả lập mới?

- Android Device Monitor
- AVD Manager
- SDK Manager
- Theme Editor

Câu hỏi 4

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

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

Trang 32.

Bạn sẽ thấy thông báo "Bố cục MainActivity đã hoàn thành" trong bảng Logcat nếu mức độ log được đặt thành một trong các tùy chọn sau? (Gợi ý: có thể có nhiều câu trả lời.)

- Verbose
- Debug
- Info
- Warn
- Error
- Assert

Nộp ứng dụng của bạn để chấm điểm

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

- Một Activity hiển thị "Hello World" trên màn hình.
- Các câu lệnh log trong onCreate() của activity chính.
- Mức độ log trong bảng Logcat chỉ hiển thị các câu lệnh log debug hoặc error.

Bài 1.2 Phần A: Giao diện tương tác đầu tiên của bạn

Giao diện người dùng (UI)

Giới thiệu

Giao diện người dùng (UI) xuất hiện trên màn hình của một thiết bị Android bao gồm một hệ thống phân cấp các đối tượng gọi là views — mỗi phần tử trên màn hình đều là một View. Lớp View đại diện cho 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 bấm, hộp kiểm và các trường nhập văn bản. Các lớp con View thường được sử dụng được mô tả qua nhiều bài học bao gồm:

- TextView để hiển thị văn bản.
- EditText để cho phép người dùng nhập và chỉnh sửa văn bản.
- Nút và các phần tử có thể nhấp khác (như RadioButton, CheckBox và Spinner) để cung cấp hành vi tương tác.
- ScrollView và RecyclerView để hiển thị các mục có thể cuộn.
- ImageView để hiển thị hình ảnh.
- ConstraintLayout và LinearLayout để chứa các phần tử View khác và định vị chúng.

Mã Java hiển thị và điều khiển giao diện người dùng nằm trong một lớp mở rộng Activity. Một

Activity thường liên kết với một bố cục của các view UI được định nghĩa dưới dạng tệp XML (eXtended Markup

Language). Tệp XML này thường được đặt tên theo Activity của nó và định nghĩa bố cục của các phần tử View

trên màn hình.

Ví dụ, mã MainActivity trong ứng dụng Hello World hiển thị một bố cục được định nghĩa trong tệp bố cục activity_main.xml, bao gồm một TextView với văn bản "Hello World". Trong các ứng dụng phức tạp hơn, một Activity 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 sẽ tìm hiểu thêm về lớp Activity trong một bài học khác.

Trong bài 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 một ứng dụng sử dụng mẫu Hoạt động Trống. Bạn cũng sẽ học cách sử dụng trình chỉnh sửa bố cục để thiết kế một

bố cục, và cách chỉnh sửa bố cục trong XML. Bạn cần phát triển những kỹ năng này để có thể hoàn thành các bài thực hành khác trong khóa học này.

Những gì bạn nên đã biết

Bạn nên quen thuộc với:

- Cách cài đặt và mở Android Studio.
- Cách tạo ứng dụng HelloWorld.
- Cách chạy ứng dụng HelloWorld. Trang 34.

Những gì bạn sẽ học

- Cách tạo một ứng dụng với hành vi tương tác.
- Cách sử dụng trình chỉnh sửa bố cục để thiết kế một bố cục.
- Cách chỉnh sửa bố cục trong XML.
- Nhiều thuật ngữ mới. Hãy xem từ vựng và danh sách khái niệm để có định nghĩa thân thiện.

Những gì bạn sẽ làm

- Tạo một ứng dụng và thêm hai phần tử Button và một TextView vào bố cục.
- Thao tác với từng phần tử trong ConstraintLayout để ràng buộc chúng với các lề và các phần tử khác.
- Thay đổi thuộc tính của các phần tử giao diện người dùng.
- Chỉnh sửa bố cục của ứng dụng trong XML.
- Trích xuất các chuỗi mã cứng thành tài nguyên chuỗi.
- Triển khai các phương thức xử lý nhấp chuột để hiển thị thông điệp trên màn hình khi người dùng chạm vào từng Button.

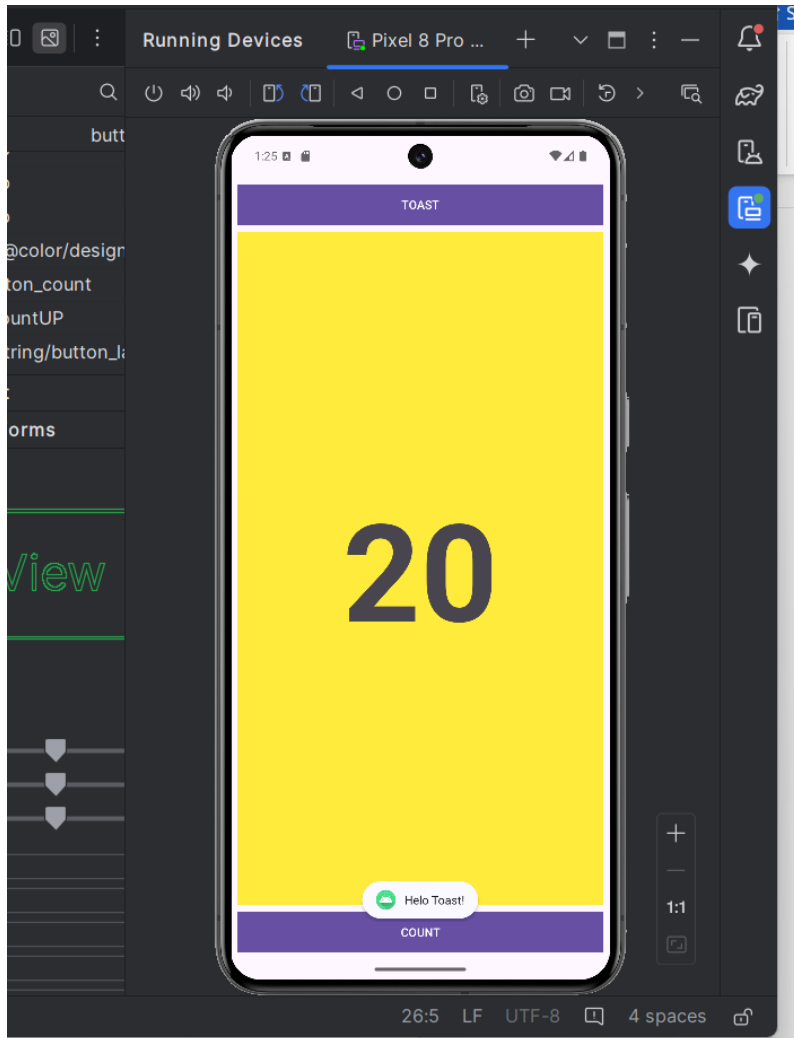
Tổng quan ứng dụng

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 điệp ngắn (một Toast) trên màn hình.

Chạm vào Button thứ hai sẽ tăng một bộ đếm "nhấp" được hiển thị trong TextView, bắt đầu từ số không.

Dưới đây là hình ảnh của ứng dụng hoàn chỉnh:

Trang 35.



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

Trong 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.

1.1 Tạo dự án Android Studio

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

Attribute	Value
-----------	-------

Trang 36.

Application Name	Hello Toast
Company Name	com.example.android (or your own domain)
Phone and Tablet Minimum SDK	API15: Android 4.0.3 IceCreamSandwich
Template	Empty Activity
Generate Layout file box	Selected
Backwards Compatibility box	Selected

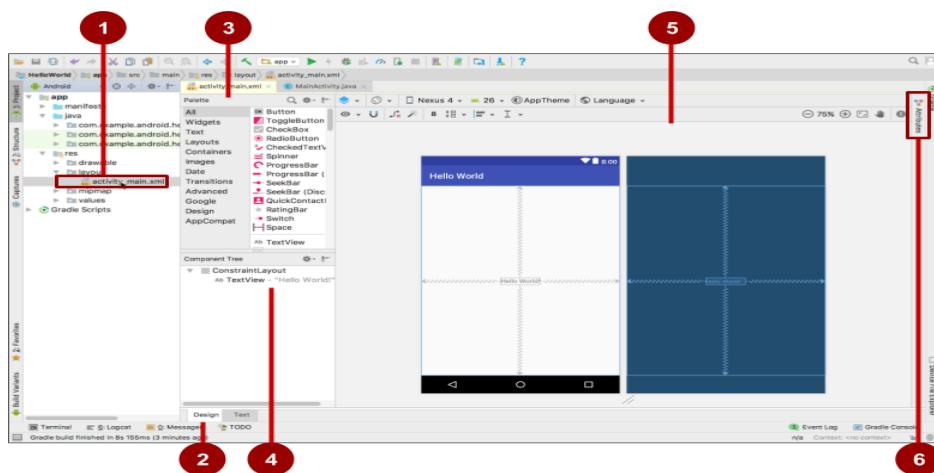
15. Chọn Chạy > Chạy ứng dụng hoặc nhấp vào biểu tượng Chạy trên thanh công cụ để xây dựng và thực thi ứng dụng trên trình giả lập hoặc thiết bị của bạn.

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 giao diện người dùng (UI) của ứng dụng. Nó cho phép bạn kéo các phần tử vào chế độ thiết kế trực quan và chế độ bản đồ, định vị chúng trong bố cục, thêm ràng buộc và thiết lập thuộc tính. Ràng buộc xác định vị trí của một phần tử UI trong bố cục. Một ràng buộc đại diện cho một kết nối hoặc căn chỉnh với một chế độ xem khác, bố cục cha, hoặc một hướng dẫn vô hình.

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

Trang 37.



1. Trong ứng dụng > thư mục res > layout trong Project > bảng Android, nhấp đúp vào tệp

activity_main.xml để mở nó, nếu nó chưa được mở.

2. Nhấp vào tab Design nếu nó chưa được chọn. Bạn sử dụng tab Design để thao tác với các phần tử và bố cục, và tab Text để chỉnh sửa mã XML cho bố cục.

3. Bảng Palettes hiển thị các phần tử UI mà bạn có thể sử dụng trong bố cục của ứng dụng.

4. Bảng Component tree hiển thị cấu trúc cây của các phần tử UI. Các phần tử xem được sắp xếp thành một cấu trúc cây của các bậc cha và con, trong đó một phần tử con kế thừa

các thuộc tính của phần tử cha. Trong hình trên, TextView là một phần tử con của ConstraintLayout.

Bạn sẽ tìm hiểu về các phần tử này sau trong bài học này.

5. Các bảng thiết kế và bản vẽ của trình chỉnh sửa bố cục hiển thị các phần tử UI trong bố cục. Trong

hình trên, bố cục chỉ hiển thị một phần tử: một TextView hiển thị "Hello World".

6. Tab Thuộc tính hiển thị bảng Thuộc tính để thiết lập các thuộc tính cho một phần tử giao diện người dùng.

Mẹo: Xem Xây dựng giao diện người dùng với Trình chỉnh sửa bố cục để biết chi tiết về việc sử dụng trình chỉnh sửa bố cục, và Gặp gỡ Android Studio để xem tài liệu đầy đủ về Android Studio.

Trang 38.

Nhiệm vụ 2: Thêm các phần tử View trong trình chỉnh sửa bố cục

Trong nhiệm vụ này, bạn sẽ tạo bố cục giao diện người dùng cho ứng dụng HelloToast trong trình chỉnh sửa bố cục bằng cách sử dụng các tính năng của ConstraintLayout. Bạn có thể tạo các ràng buộc thủ công, như sẽ được trình bày sau, hoặc tự động bằng cách sử dụng công cụ Autoconnect.

2.1 Kiểm tra các ràng buộc của phần tử

Thực hiện các bước sau:

1. Mở `activity_main.xml` từ `Project > Android` nếu nó chưa được mở. Nếu tab `Thiết kế` chưa được chọn, hãy nhấp vào nó.

Nếu không có bản thiết kế, hãy nhấp vào nút `Chọn Bề mặt Thiết kế` trong thanh công cụ và chọn `Thiết kế + Bản thiết kế`.

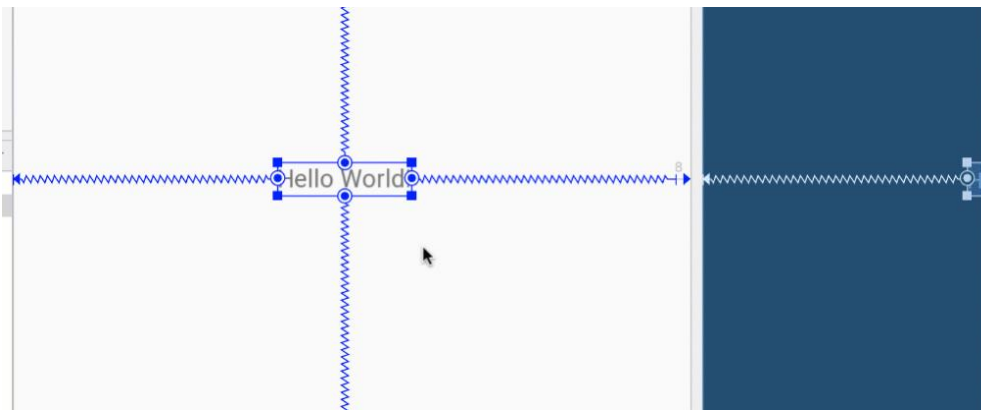
2. Công cụ `Autoconnect` cũng nằm trong thanh công cụ. Nó được bật theo mặc định. Đối với bước này, hãy đảm bảo rằng công cụ không bị vô hiệu hóa.

3. Nhấn nút phóng to để phóng to các khung thiết kế và bản vẽ để có cái nhìn cận cảnh.

4. Chọn `TextView` trong khung `Component Tree`. `TextView "Hello World"` được làm nổi bật trong các khung thiết kế và bản vẽ và các ràng buộc cho phần tử đó được hiển thị.

5. Tham khảo hình ảnh động bên dưới cho bước này. Nhấn vào tay cầm hình tròn ở bên phải của `TextView` để xóa ràng buộc ngang liên kết view với bên phải của bố cục. `TextView` nhảy sang bên trái vì nó không còn bị ràng buộc ở bên phải. Để thêm lại ràng buộc ngang, nhấn vào cùng một tay cầm và kéo một đường thẳng sang bên phải của bố cục.

Trang 39.



Trong các bản thiết kế hoặc bảng thiết kế, các tay cầm sau xuất hiện trên phần tử `TextView`:

- Tay cầm ràng buộc: Để tạo một ràng buộc như được hiển thị trong hình động ở trên, hãy nhấp vào một tay cầm ràng buộc, được hiển thị dưới dạng một hình tròn ở bên cạnh một phần tử. Sau đó, kéo tay cầm đến một tay cầm ràng buộc khác, hoặc đến ranh giới của cha. Một đường zigzag đại diện cho ràng buộc.

- Tay cầm thay đổi kích thước: Để thay đổi kích thước của phần tử, hãy kéo các tay cầm thay đổi kích thước hình vuông. Tay cầm sẽ chuyển thành một góc nghiêng trong khi bạn đang kéo nó.

Trang 40.

2.2 Thêm một nút vào bố cục

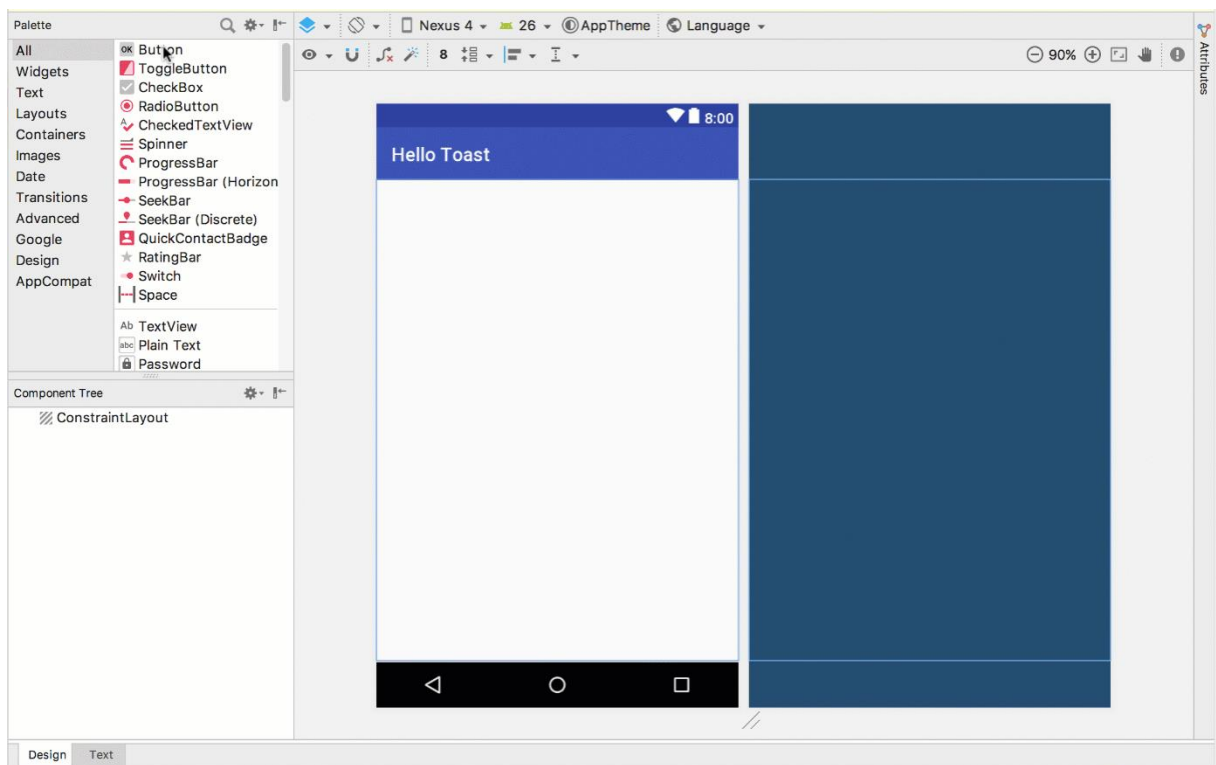
Khi được kích hoạt, công cụ Autoconnect tự động tạo hai hoặc nhiều ràng buộc cho một phần tử giao diện người dùng với bố cục cha. Sau khi bạn kéo phần tử vào bố cục, nó sẽ tạo ràng buộc dựa trên vị trí của phần tử đó.

Thực hiện các bước sau để thêm một nút:

1. Bắt đầu với một trang trắng. Phần tử TextView không cần thiết, vì vậy trong khi nó vẫn được chọn,

hãy nhấn phím Delete hoặc chọn Chỉnh sửa > Xóa. Bây giờ bạn đã có một bố cục hoàn toàn trống.

2. Kéo một nút từ bảng Palette vào bất kỳ vị trí nào trong bố cục. Nếu bạn thả nút ở khu vực giữa trên của bố cục, các ràng buộc có thể tự động xuất hiện. Nếu không, bạn có thể kéo các ràng buộc đến phía trên, bên trái và bên phải của bố cục như được hiển thị trong hình động bên dưới.



2.3 Thêm một nút thứ hai vào bố cục

1. Kéo một nút khác từ bảng Palette vào giữa bố cục như hình minh họa bên dưới. Autoconnect có thể cung cấp các ràng buộc ngang cho bạn (nếu không, bạn có thể tự kéo chúng).
2. Kéo một ràng buộc dọc đến đáy của bố cục (tham khảo hình bên dưới).



Bạn có thể xóa ràng buộc khỏi một phần tử bằng cách chọn phần tử và di chuột qua nó để hiển thị nút Xóa Ràng Buộc. Nhấp vào nút này để xóa tất cả ràng buộc trên phần tử đã chọn. Để xóa một ràng buộc cụ thể, nhấp vào tay cầm cụ thể thiết lập ràng buộc đó. Để xóa tất cả ràng buộc trong toàn bộ bố cục, nhấp vào công cụ Xóa Tất Cả Ràng Buộc trên thanh công cụ. Công cụ này rất hữu ích nếu bạn muốn làm lại tất cả các ràng buộc trong bố cục của mình.

Nhiệm vụ 3: Thay đổi thuộc tính của phần tử giao diện

Bảng thuộc tính cung cấp quyền truy cập vào tất cả các thuộc tính XML mà bạn có thể gán cho một phần tử giao diện. Bạn có thể tìm thấy các thuộc tính (được gọi là thuộc tính) chung cho tất cả các chế độ xem trong tài liệu lớp View.

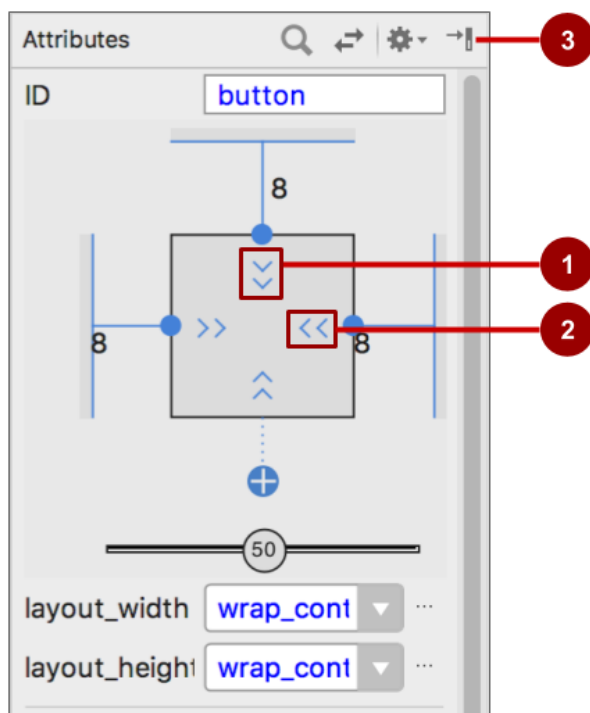
Trong nhiệm vụ này, bạn sẽ nhập các giá trị mới và thay đổi các giá trị cho các thuộc tính quan trọng của Button, áp dụng cho hầu hết các loại chế độ xem.

Trang 42.

3.1 Thay đổi kích thước nút

Trình chỉnh sửa bố cục cung cấp các tay cầm thay đổi kích thước ở bốn góc của một View để bạn có thể thay đổi kích thước View một cách nhanh chóng. Bạn có thể kéo các tay cầm ở mỗi góc của View để thay đổi kích thước, nhưng làm như vậy sẽ cố định các kích thước chiều rộng và chiều cao. Tránh cố định kích thước cho hầu hết các phần tử View, vì các kích thước cố định không thể thích ứng với nội dung và kích thước màn hình khác nhau.

Thay vào đó, hãy sử dụng bảng Thuộc tính ở bên phải của trình chỉnh sửa bố cục để chọn chế độ kích thước không sử dụng kích thước cố định. Bảng Thuộc tính bao gồm một bảng kích thước hình vuông gọi là trình kiểm tra View ở phía trên. Các ký hiệu bên trong hình vuông đại diện cho các cài đặt chiều cao và chiều rộng như sau:



Trong hình trên:

1. Điều khiển chiều cao. Điều khiển này xác định thuộc tính `layout_height` và xuất hiện ở hai phân đoạn ở phía trên và phía dưới của hình vuông. Các góc chỉ ra rằng điều khiển này được cài đặt thành `wrap_content`, có nghĩa là View sẽ mở rộng theo chiều dọc khi cần thiết để phù hợp với nội dung của nó. Số "8" chỉ ra một khoảng cách tiêu chuẩn được đặt là 8dp.

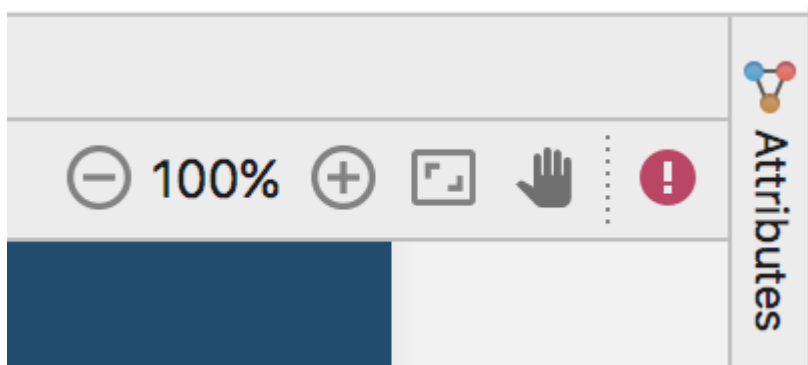
2. Điều khiển chiều rộng. Điều khiển này xác định `layout_width` và xuất hiện ở hai phân đoạn ở bên trái và bên phải của hình vuông. Các góc chỉ ra rằng điều khiển này được cài đặt thành `wrap_content`, điều này có nghĩa là View sẽ mở rộng theo chiều ngang khi cần thiết để phù hợp với nội dung của nó, tối đa là 8dp.

3. Nút đóng bảng thuộc tính. Nhấp để đóng bảng.

Thực hiện các bước sau:

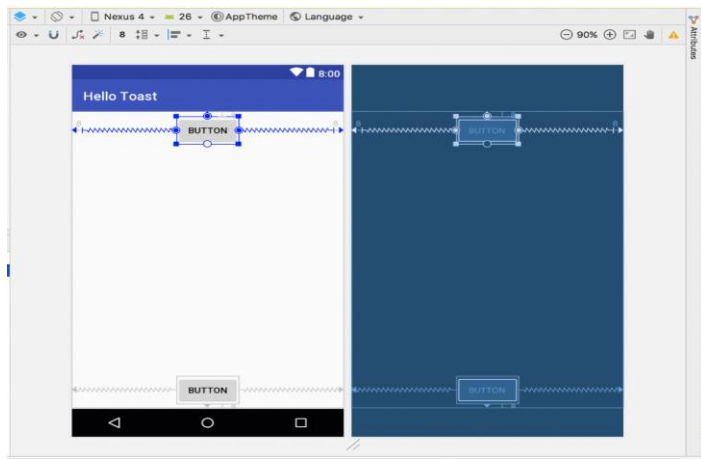
1. Chọn nút trên cùng trong bảng Cây thành phần.

2. Nhấp vào tab Thuộc tính ở bên phải của cửa sổ trình chỉnh sửa bố cục. Trang 43.



3. Nhấp vào điều khiển độ rộng hai lần - lần nhấp đầu tiên sẽ chuyển nó thành Cố định với các đường thẳng, và lần nhấp thứ hai sẽ chuyển nó thành Phù hợp với Ràng buộc với các cuộn lò xo, như được hiển thị trong hình hoạt hình bên dưới.

Trang 44.

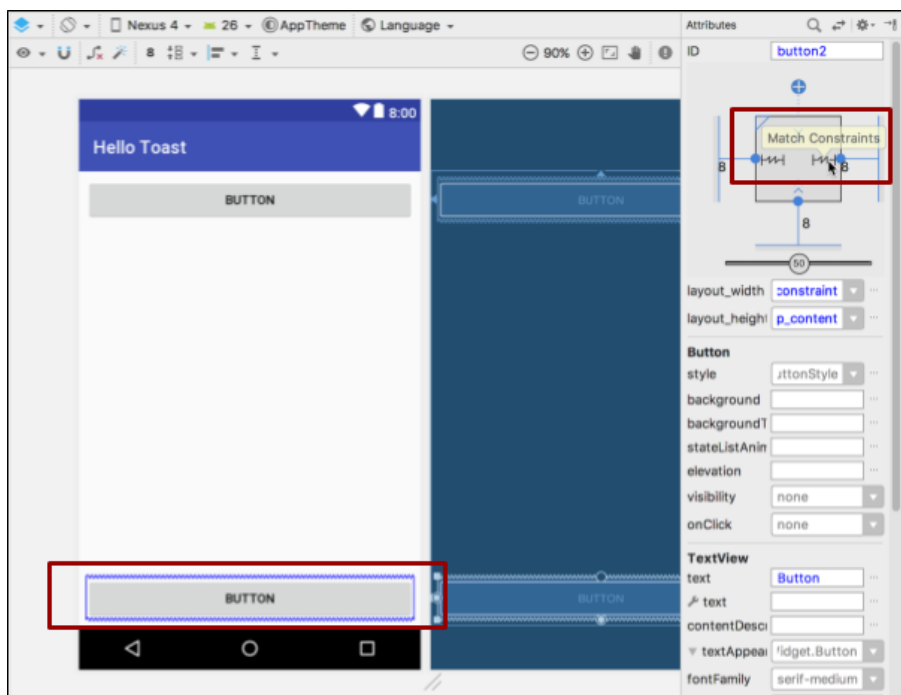


Kết quả của việc thay đổi điều khiển chiều rộng, thuộc tính `layout_width` trong bảng Thuộc tính

hiển thị giá trị `match_constraint` và phần tử Button kéo dài theo chiều ngang để lấp đầy không gian giữa hai bên trái và phải của bố cục.

4. Chọn Button thứ hai và thực hiện các thay đổi tương tự cho `layout_width` như trong bước trước, như được hiển thị trong hình dưới đây.

Trang 45.



Như đã trình bày trong các bước trước, các thuộc tính `layout_width` và `layout_height` trong bảng thuộc tính thay đổi khi bạn thay đổi các điều khiển chiều cao và chiều rộng

trong trình kiểm tra. Các thuộc tính này có thể nhận một trong ba giá trị cho layout, đó là `ConstraintLayout`:

- Cài đặt `match_constraint` mở rộng phần tử View để lấp đầy cha của nó theo chiều rộng hoặc chiều cao - cho đến một khoảng cách, nếu có. Cha trong trường hợp này là `ConstraintLayout`. Bạn sẽ tìm hiểu thêm về `ConstraintLayout` trong nhiệm vụ tiếp theo.
- Cài đặt `wrap_content` thu nhỏ kích thước của phần tử View sao cho vừa đủ để bao bọc nội dung của nó. Nếu không có nội dung, phần tử View sẽ trở nên vô hình.
- Để chỉ định kích thước cố định điều chỉnh theo kích thước màn hình của thiết bị, hãy sử dụng một số lượng pixel độc lập với mật độ cố định (đơn vị dp). Ví dụ, 16dp có nghĩa là 16 pixel độc lập với mật độ.

Trang 46.