1. **IDE là gì?**

**IDE**, hay còn gọi là **Môi trường phát triển tích hợp (Integrated Development Environment)** là một ứng dụng phần mềm cung cấp một bộ công cụ toàn diện để phát triển phần mềm. IDE được thiết kế để làm cho quá trình phát triển phần mềm hiệu quả hơn, bằng cách cung cấp một giao diện duy nhất để quản lý tất cả các khía cạnh (aspects).

IDE cung cấp nhiều công cụ và tính năng giúp các lập trình viên có thể viết, test và debug dễ dàng hơn. IDE cũng cải thiện chất lượng code với các công cụ highlight cú pháp, hoàn thành code (code completion) và tái cấu trúc (refactoring). Ngoài ra, IDE cũng thường bao gồm hỗ trợ tích hợp cho các hệ thống kiểm soát nguồn, giúp các lập trình viên dễ dàng cộng tác và quản lý các thay đổi trong dự án.

Hiện nay có hai loại IDE chính, bao gồm **IDE một ngôn ngữ** và **IDE đa ngôn ngữ**.

Ngoài ra, còn có các loại IDE khác như:

+ IDE dựa trên web

+ IDE di động

1. **Các tính năng chính của IDE**

Ba tính năng phổ biến trong hầu hết các IDE là **trình soạn thảo mã nguồn**, **tự động hóa việc xây dựng** và **gỡ lỗi**. Các tính năng bổ sung có thể khác nhau và bao gồm:

- Cải tiến UI của trình soạn thảo mã

- Tính năng kiểm thử tự động

- Hỗ trợ triển khai mã thông qua tích hợp phần bổ trợ

- Hỗ trợ tái cấu trúc mã

- Hỗ trợ đóng gói ứng dụng

**III. IDE dành cho Android**

Tháng 5 năm 2013, Google công bố Android Studio, một môi trường phát triển ứng dụng tích hợp (IDE) dành riêng cho Android, mã nguồn mở, dựa trên IDE **Java IntelliJ** của hãng **JetBrains**(đối thủ với Eclipse và Netbeans, vốn khá quen thuộc với dân lập trình Java).

**Android Studio** chạy trên Windows, Mac và Linux, nhằm thay thế cho **Eclipse Android Development Tool**(ADT) vốn được sử dụng làm IDE chính trong các năm trước đó.

#### Một số tính năng nổi bật:

* Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).
* Chức năng dò và sửa lỗi nhanh, hướng Android.
* Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.
* Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.
* Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.

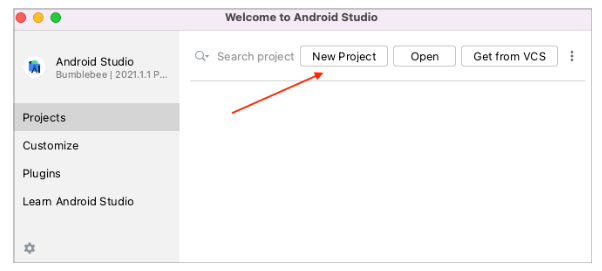
**IV.Tạo, chạy và debug cho Android:**

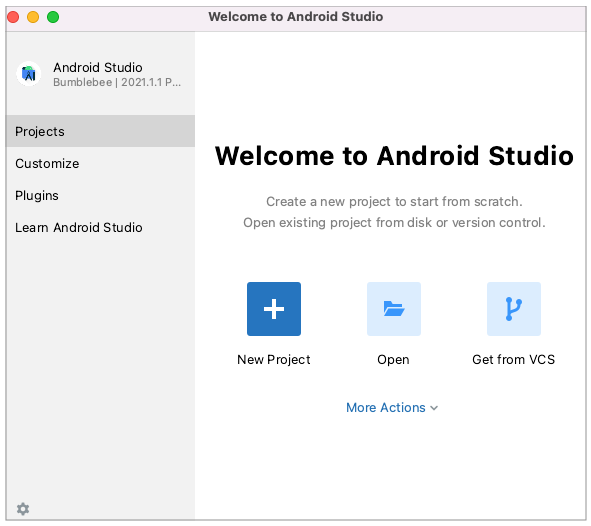
**+ Tạo:**

1. Khởi động Android Studio (nếu chưa mở) bằng cách nhấp vào biểu tượng Android Studio:

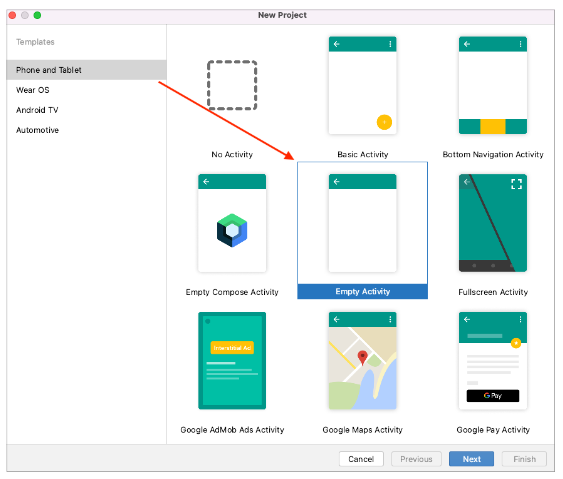


1. Cửa sổ **Chào mừng bạn đến với Android Studio** sẽ mở ra. Nhấp vào **New Project** (Dự án mới).





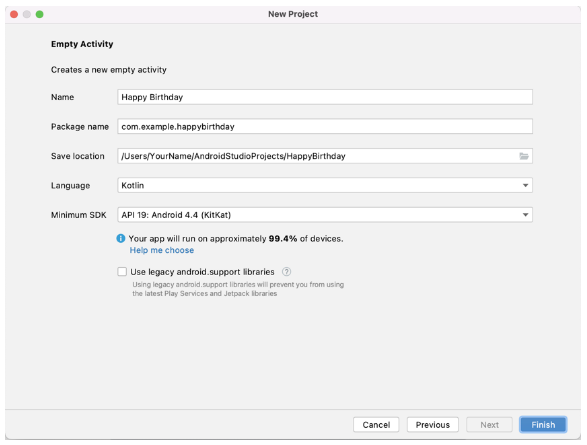
1. Cửa sổ **Dự án mới** mở ra với một danh sách các mẫu do Android Studio cung cấp.



1. Hãy nhấp vào thẻ ở trên cùng và duyệt qua các mẫu để biết những việc bạn có thể làm! Có nhiều mẫu dành cho nhiều loại thiết bị (chẳng hạn như điện thoại, máy tính bảng và đồng hồ) và nhiều loại ứng dụng (ứng dụng có màn hình cuộn được, ứng dụng có bản đồ và ứng dụng dễ thao tác).
2. Ở trên cùng bên trái cửa sổ, nhấp vào thẻ **Phone and Tablet** (Điện thoại và máy tính bảng).
3. Ở trên cùng của dòng, nhấp vào mẫu **Empty Activity** (Hoạt động trống) để chọn mẫu đó làm mẫu cho dự án.

Mẫu **Empty Activity** (Hoạt động trống) là mẫu đơn giản nhất có thể dùng để tạo ứng dụng. Nó có một màn hình và hiển thị thông điệp đơn giản "Hello World!" (Xin chào mọi người)

1. Ở cuối cửa sổ, nhấp vào **Next** (Tiếp tục). Hộp thoại **New Project** (Dự án mới) sẽ mở ra.

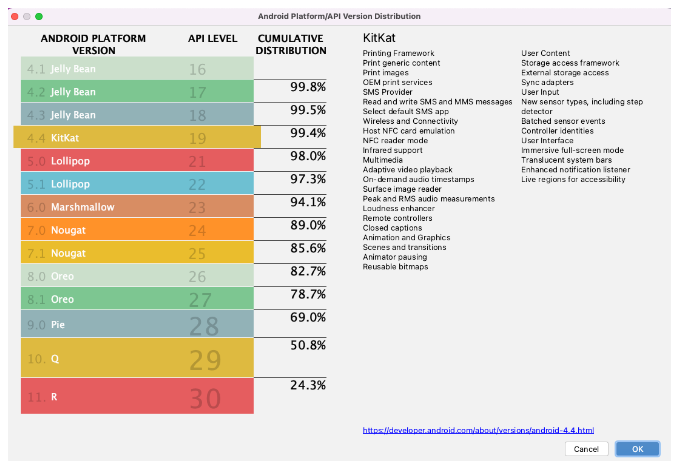


Định cấu hình dự án như sau:

1. **Name** (Tên) là tên của ứng dụng. Trong hộp bên dưới **Name** (Tên), nhập Happy Birthday làm tên cho dự án.
2. **Package name** (Tên gói) là tên mà hệ thống Android sử dụng để nhận dạng duy nhất ứng dụng của bạn. Thông thường, tên mặc định là tên tổ chức của bạn, theo sau là tên ứng dụng, tất cả đều viết bằng chữ thường (trong trường hợp này là "com.example.happybirthday").
3. **Save location** (Vị trí lưu) là vị trí bạn đã lưu tất cả các tệp liên quan đến dự án. Ghi lại vị trí của tệp trên máy tính để bạn có thể tìm thấy các tệp của mình. Hiện tại, bạn có thể giữ nguyên vị trí lưu.
4. **Language** (Ngôn ngữ) xác định ngôn ngữ lập trình bạn muốn sử dụng cho dự án. Hãy đảm bảo **Language** (Ngôn ngữ) là Kotlin.
5. **Minimum SDK** (SDK tối thiểu) chỉ phiên bản Android tối thiểu mà ứng dụng có thể chạy trên đó. API 19: Android 4.4 (KitKat)Chọn từ danh sách thả xuống.

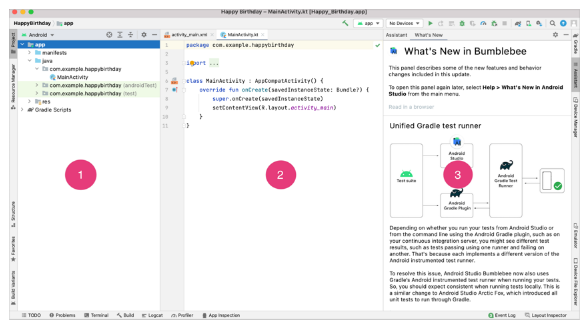
**Lưu ý:** Có nhiều phiên bản hệ điều hành Android, mỗi phiên bản được đặt tên theo thứ tự bảng chữ cái khi được phát hành.

1. Bên dưới **Minimum SDK** (SDK tối thiểu), hãy chú ý đến ghi chú thông tin về số lượng thiết bị mà ứng dụng có thể chạy ở cấp API đã chọn. Để biết thêm chi tiết, hãy nhấp vào liên kết **Help me choose** (Giúp tôi chọn) để hiển thị danh sách các phiên bản Android như hiển thị bên dưới. Sau đó, hãy quay lại cửa sổ **New Project** (Dự án mới).



1. Trong cửa sổ **New Project** (Dự án mới), nhớ bỏ đánh dấu chọn hộp **Use legacy android.support libraries** (Sử dụng thư viện android.support cũ). Nhấp vào dấu chấm hỏi nếu muốn biết thêm về điều này.
2. Nhấp vào **Finish** (Hoàn tất).

Android Studio sẽ mở dự án và tất cả các tệp trong dự án.



Khi mở Android Studio lần đầu tiên, bạn sẽ thấy ba cửa sổ:

(1) Cửa sổ **Project** (Dự án) hiện các tệp và thư mục trong dự án.

(2) Cửa sổ **Editing** (Chỉnh sửa) là để chỉnh sửa mã.

(3) Cửa sổ **What's New** (Tính năng mới) hiển thị tin tức và các mẹo hữu ích.

Ở góc dưới bên phải của Android Studio, bạn sẽ thấy một thông báo hoặc thanh tiến trình cho biết ứng dụng Android Studio có vẫn đang thiết lập dự án hay không. Ví dụ:

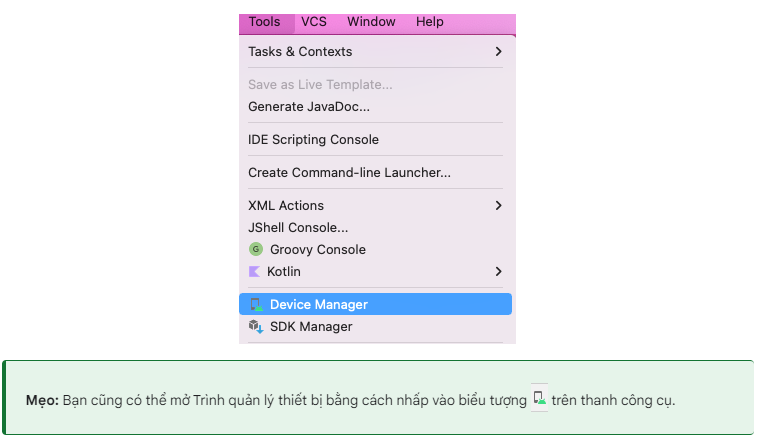


1. Hãy đợi cho đến khi Android Studio hoàn tất quá trình thiết lập dự án. Thông báo ở góc dưới bên trái, chẳng hạn như thông báo hiển thị bên dưới, sẽ cho bạn biết khi nào dự án hoàn tất.



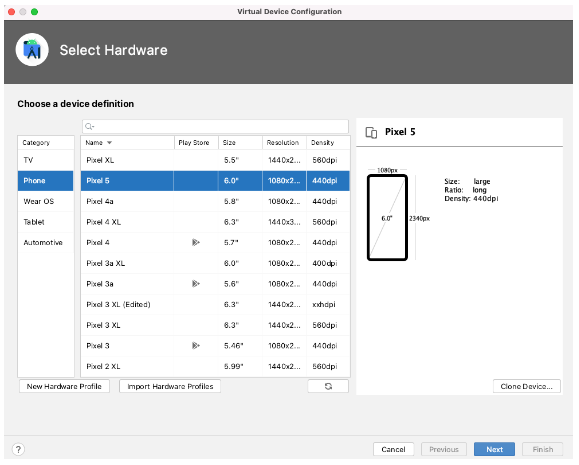
**+ Chạy:**

1. Từ thanh trình đơn Android Studio, chọn **Tools > Device Manage** (Công cụ > Trình quản lý thiết bị).



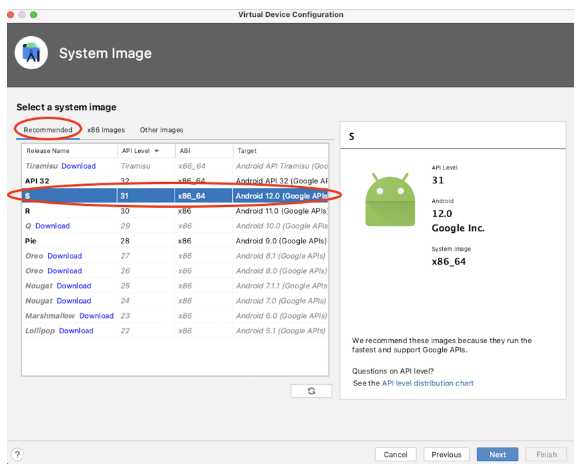
1. Nhấp vào **Tạo thiết bị** (Create Device).

Cửa sổ **Select Hardware** (Chọn phần cứng) xuất hiện.



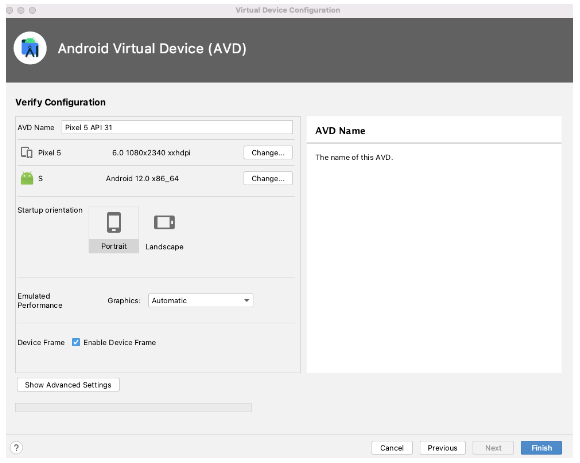
1. Chọn danh mục **Phone** (Điện thoại).
2. Chọn điện thoại (ví dụ: Pixel 5), sau đó nhấp vào **Next** (Tiếp theo). Bạn có thể chọn bất kỳ điện thoại nào.

Cửa sổ **System Image** (Hình ảnh hệ thống) sẽ xuất hiện. Từ đây, bạn chọn phiên bản hệ thống Android để chạy trên thiết bị ảo của mình. Điều này cho phép bạn kiểm thử ứng dụng trên các phiên bản khác nhau của hệ thống Android.



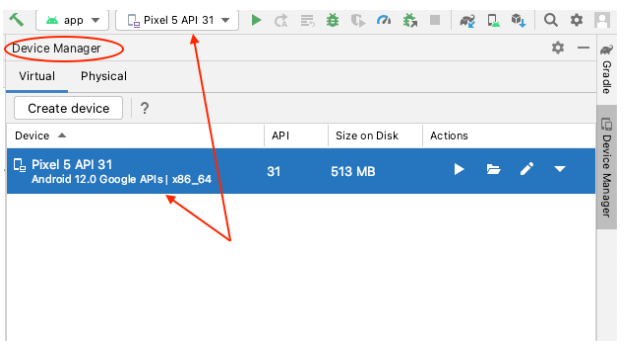
1. Từ thẻ **Recommended** (Đề xuất), chọn S làm phiên bản hệ thống Android để chạy trên thiết bị ảo. Nhấp vào **Next** (Tiếp theo).

Cửa sổ **Android Virtual Device (AVD)** (Thiết bị Android ảo (AVD)) sẽ hiển thị, từ đó bạn có thể chọn thêm thông tin chi tiết về cấu hình cho thiết bị.



1. Trong trường **AVD Name** (Tên AVD), nhập tên cho Thiết bị ảo Android của bạn. Giữ nguyên phần còn lại.
2. Nhấp vào **Finish** (Hoàn tất).

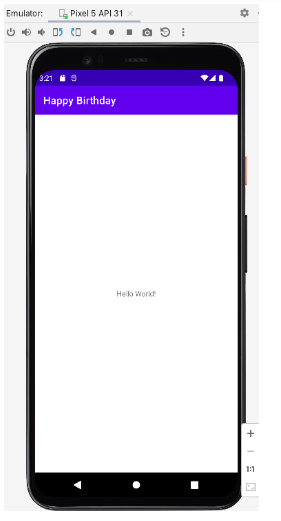
Thiết bị ảo mới sẽ hiển thị trong cửa sổ **Device Manager** (Trình quản lý thiết bị) và sẵn sàng để sử dụng.



Chạy ứng dụng trên thiết bị ảo

1. Nếu chưa sẵn sàng, hãy dùng Android Studio làm nền trước.
2. Trong Android Studio, trên thanh công cụ, tìm trình đơn thả xuống gồm các thiết bị ảo (giống ví dụ bên dưới), sau đó chọn thiết bị ảo bạn đã tạo từ danh sách thả xuống.  (Bạn cũng có thể nhấp vào **Run > Select Device..** (Chạy > Chọn thiết bị..) rồi chọn thiết bị ảo từ các thiết bị có trong cửa sổ bật lên.)
3. Trong Android Studio, chọn **Run > Run app** (Chạy > Chạy ứng dụng) hoặc nhấp vào biểu tượng **Run** (Run) trên thanh công cụ. Thiết bị ảo sẽ bắt đầu chạy và khởi động giống như thiết bị thực. Tuỳ vào tốc độ máy tính, quá trình này có thể mất một chút thời gian.

Khi ứng dụng sẵn sàng, nó sẽ mở trên thiết bị ảo như được hiển thị dưới đây.



Tuyệt vời! Thiết bị ảo của bạn hiện đã hoạt động. Lưu ý hiện tại tiêu đề là "Happy Birthday" và "Hello World!" đang hiển thị trên màn hình.

**+ Debug:**

Debug cơ bản với Android Studio: Để có thể thao tác debug, trước tiên, bạn cần phải có kiến thức về nền tảng JAVA thì mới hiểu được flow code và từ đó mới có thể đặt Log để debug chính xác được.

* Để bắt đầu debug, các bạn nhấn vào nút Debug  trên thanh công cụ (khi trỏ vào sẽ có chữ “Debug ‘app’”).

Lúc này Android Studio sẽ build ứng dụng ra file APK, ký (sign) file APK bằng key debug, và cài đặt lên thiết bị của bạn. Cuối cùng, cửa sổ Debug sẽ được mở ra:

Sử dụng Log: Log giúp chúng ta hiểu được thứ tự khởi chạy của code bằng cách thu nhặt dữ liệu của hệ thống trong khi chúng ta chạy app, và cho chúng ta biết được ứng dụng lỗi ở điểm nào.

Bước 1: Đặt Log trong code

Bước 2: Đọc Log

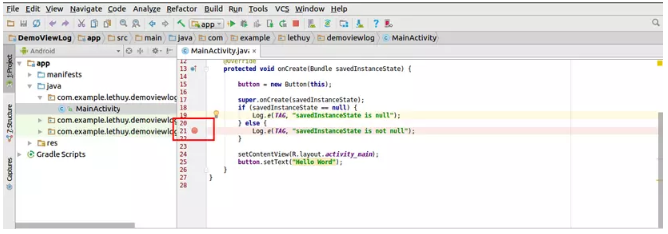
Và bây giờ, chạy app (chạy - Run hoặc Debug đều được, vì chúng ta chỉ cần xem Log thôi mà):



**Làm việc với Breakpoint**

* Android Studio có cung cấp cơ chế đặt điểm dừng (Break Point) để khởi chạy một thao tác gỡ lỗi ứng dụng nào đó. Trong đó phổ biến nhất là đặt breakpoint để dừng ứng dụng tạm thời khi một dòng code nào đó được gọi tới.
* Sau khi app được dừng (pause), các bạn có thể kiểm tra thông số các biến, kiểm tra biểu thức, hoặc chạy tiếp code từng dòng một để tìm ra điểm lỗi trong code.
* Để đặt Breakpoint trong code, các bạn làm như sau:

Tìm dòng code muốn đặt Breakpoint, click chuột trái vào khoảng trống ở cột trái trước dòng code.



Khi code chạy đến đúng điểm dừng đã đặt, Android Studio sẽ dừng ứng dụng lại (màn hình sẽ bị dừng lại – đóng băng, không thể thao tác được gì thêm).

* Để nhảy đến dòng code tiếp theo (mà không vào đầu phương thức), nhấn nút Step Over Kteam hoặc nút F6 trên bàn phím.
* Để nhảy vào dòng đầu tiên trong một phương thức, nhấn Step Into hoặc phím F5.
* Để chạy code lại như bình thường, nhấn nút Resume Program hoặc phím F8.

