# Google Android

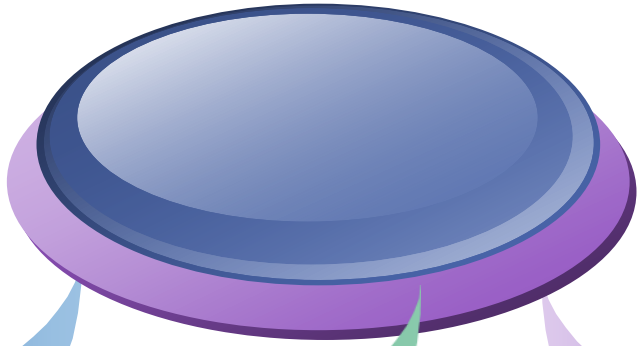
# Nội dung

- Lịch sử phát triển
- Các khái niệm cơ bản
- Đặc tính của Android
- Thành phần giao diện trong Android

# Lịch sử phát triển

- Android là hệ điều hành trên điện thoại di động (và hiện nay là cả trên một số đầu phát HD, HD Player, TV)
- Được phát triển bởi Google và dựa trên nền tảng Linux và những phần mềm trung gian (còn gọi là middleware) để hỗ trợ các ứng dụng mà người sử dụng cần đến.
- Android ban đầu được phát triển bởi Android Inc, một công ty được Google mua lại sau đó,
- Gần đây là Liên minh thiết bị cầm tay mở rộng (Open Handset Alliance).
- Android cho phép các nhà phát triển viết mã được quản lý bằng ngôn ngữ Java, điều khiển thiết bị thông qua các thư viện Java do Google phát triển.

# Lịch sử phát triển



Tháng 7 năm 2005 Google mua lại Android..

Năm 2007 với sự thành lập của hiệp hội các nhà phát triển di động mở với mục đích tạo nên 1 chuẩn mở cho điện thoại di động trong tương lai

Cũng trong năm 2007, nền tảng đầu tiên được ra mắt dựa trên linux kernel 2.6.Google công bố hầu hết các mã nguồn của Android theo bản cấp phép Apache.

# Lịch sử phát triển

- Android hiện nay đang cạnh tranh với một số hệ điều hành dành cho thiết bị di động khác như iOS, …
- Android đã trải qua một số các cập nhật kể từ lần đầu phát hành. Những cập nhật này nhìn chung có nhiệm vụ vá các lỗ hổng và thêm các tính năng mới vào hệ điều hành:
  - Bản 1.0: RC29, phiên bản đầu tiên
  - Bản 1.1: RC30, vá lỗi cho RC29
  - Bản 1.5: (Cupcake) Based on Linux Kernel 2.6.27 : bản cập nhật 1.5 chính thức (Cupcake) được phát hành ngày 30/4/2009

# Các bản cập nhật

- Bản 1.6 và SDK được chính thức phát hành ngày 15/9/2009
- Bản 2.0/2.1: (Eclair) và SDK chính thức được phát hành ngày 26/10/2009.
  - Android 2.0.1 SDK chính thức ra mắt ngày 3/12/2009.
  - Android 2.0.1 SDK chính thức ra mắt ngày 3/12/2009.
- Bản 2.2: (Froyo) dựa trên Linux Kernel 2.6.32: được phát hành ngày 20/5/2010

# Các bản cập nhật

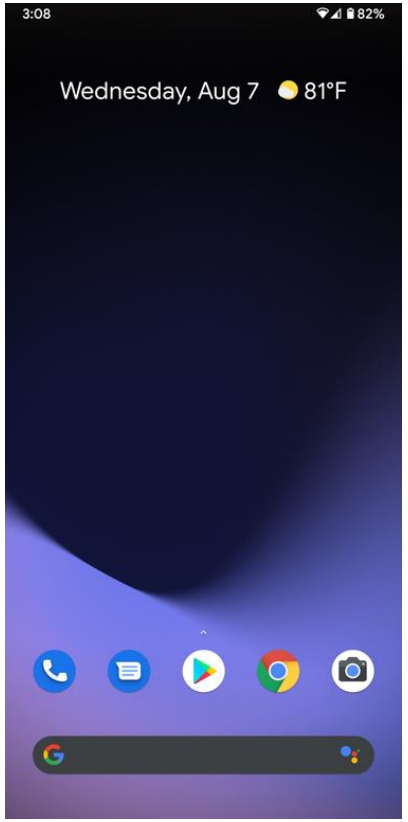
- 1.5 (**C**upcake) Based on Linux Kernel 2.6,27
- 1.6 (**D**onut) Based on Linux Kernel 2.6.29
- 2.0/2.1 (**E**clair) Based on Linux Kernel 2.6.29
- 2.2 (**F**royo) Based on Linux Kernel 2.6.32
- 2.3 (**G**ingerbread) Based on Linux Kernel 2.6.33
- 3.0 (**H**oneycomb) Based on Linux Kernel
- 4.0 (**I**ce cream) Based on Linux Kernel
- 4.1 (**J**elly Bean) 27 June 2012. Based on Linuxkernel 3.0.31, 4.2 - 29 October 2012
- 4.4 (**K**itkat)
- 5.0/5.1 (**L**ollipop)
- 6.0 (**M**arshmallow) 2015
- 7.0/7.1 (**N**ougat) 2016
- 8.0 (**Oreo**) 2017
- 9.0 (Pie) 2018
- 10 2019
- 11 2020

# Các phiên bản của Android

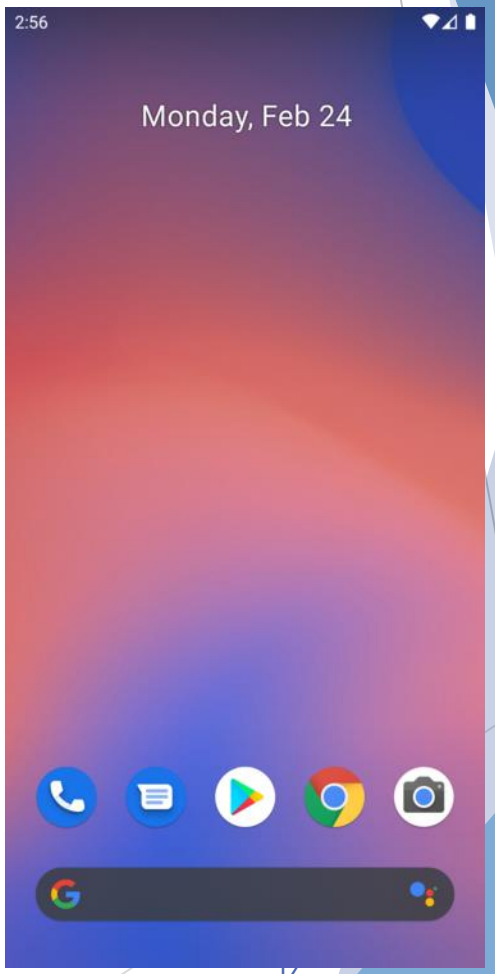
# Các phiên bản của Android

# Các phiên bản của Android

# android 10



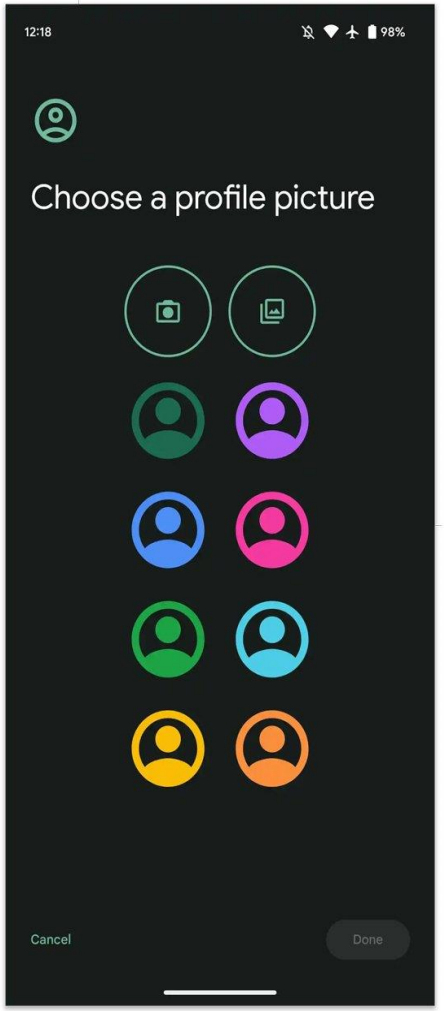
# Android 11


android 12
72°

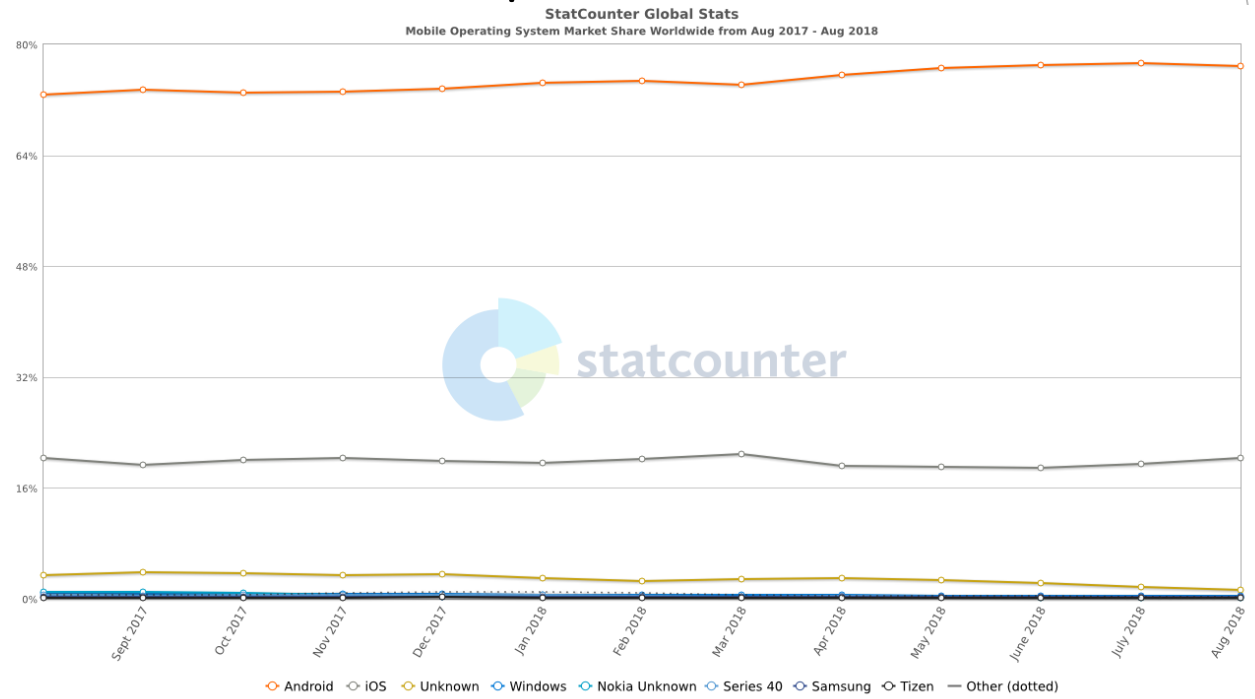
# Android 13 - 2022

# Android 14 – 2023 (Beta)


# Thị phần Android

- So sánh thị phần Android với các hệ điều hành khác




| Android | iOS | Unknown | Windows | Samsung | Series 40 |
|---|---|---|---|---|---|
| 76.88% | 20.38% | 1.23% | 0.41% | 0.28% | 0.23% |

Mobile Operating System Market Share Worldwide - August 2018

# Khả năng phát triển

Nhiều lựa chọn và Giá rẻ => Phổ biến cho người dùng tại Việt Nam

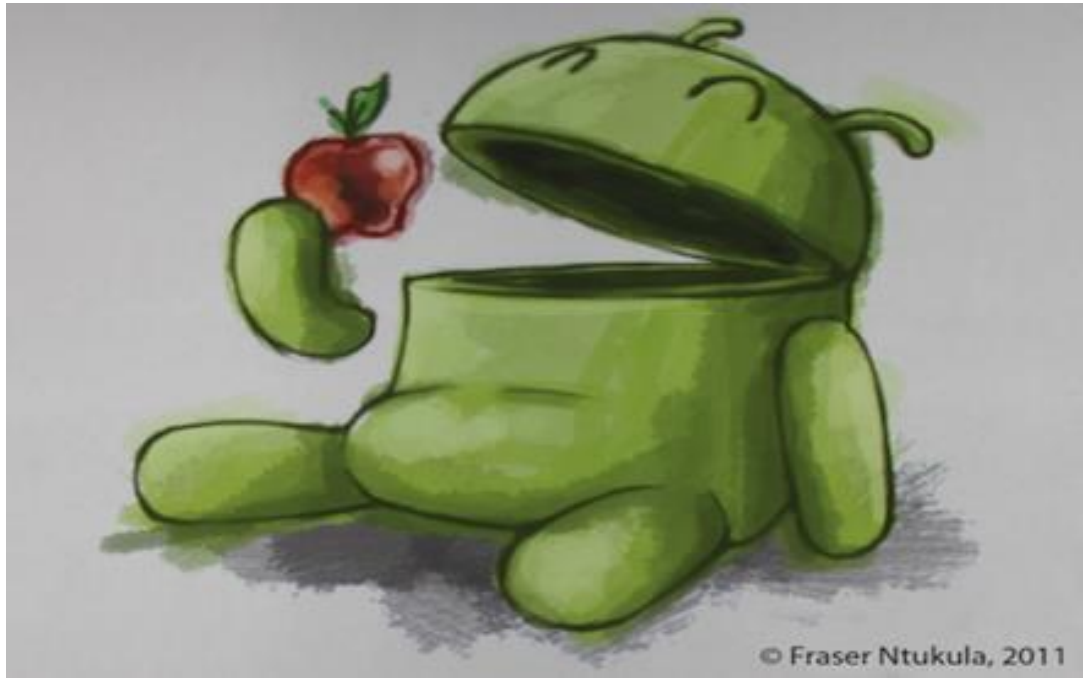
Đa dạng về thiết bị di động chạy Android như: Smartphone, Tablet, …

Được các “Ông lớn” “hậu thuẫn”: Google, HTC , Dell ,…….

Cơ hội dành cho các lập trình viên thiết bị di động Việt Nam

# Đặc tính của Android

- Tính Năng Mở của hệ điều hành Android: Android được xây dựng  để cho phép các nhà phát triển để tạo ra các ứng dụng di động hấp dẫn tận dụng tất cả tài nguyên một chiếc điện thoại đã cung cấp.



# Đặc tính của Android

- Tất cả các ứng dụng có thể được tạo ra cho Android: Android không phân biệt giữa các ứng dụng lõi của điện thoại và các ứng dụng của bên thứ ba. Các nhà phát triển có thể sử dụng miễn phí bộ Kit Android Software Development để xây dựng các ứng dụng của mình.

# Đặc tính của Android

- Với Android tốc độ nhanh & phát triển ứng dụng dễ dàng: Android cung cấp truy cập đến một loạt các thư viện công cụ hữu ích và có thể được sử dụng để xây dựng các ứng dụng phong phú.
- Phát triển Android ROM: Rất nhiều nhà phát triển hệ điều hành Android đã vào cuộc và các ROM cho Android độc đáo được ra đời với nhiều tính năng nổi trội được tích hợp và đầy sáng tạo.
- Android hỗ trợ:
  - 3G
  - Wifi
  - Màn hình cảm ứng đa điểm

# Đặc tính của Android

- Android hỗ trợ:
  - Trình duyệt dựa trên webkit
  - Tin nhắn (SMS) theo luồng
  - Định dạng MPEG-4, H.264, MP3, AAC
  - Bộ tăng tốc đồ họa 3D
  - Removable  storage
  - Widgets
  - Media sync
  - Microsoft support
  - …..

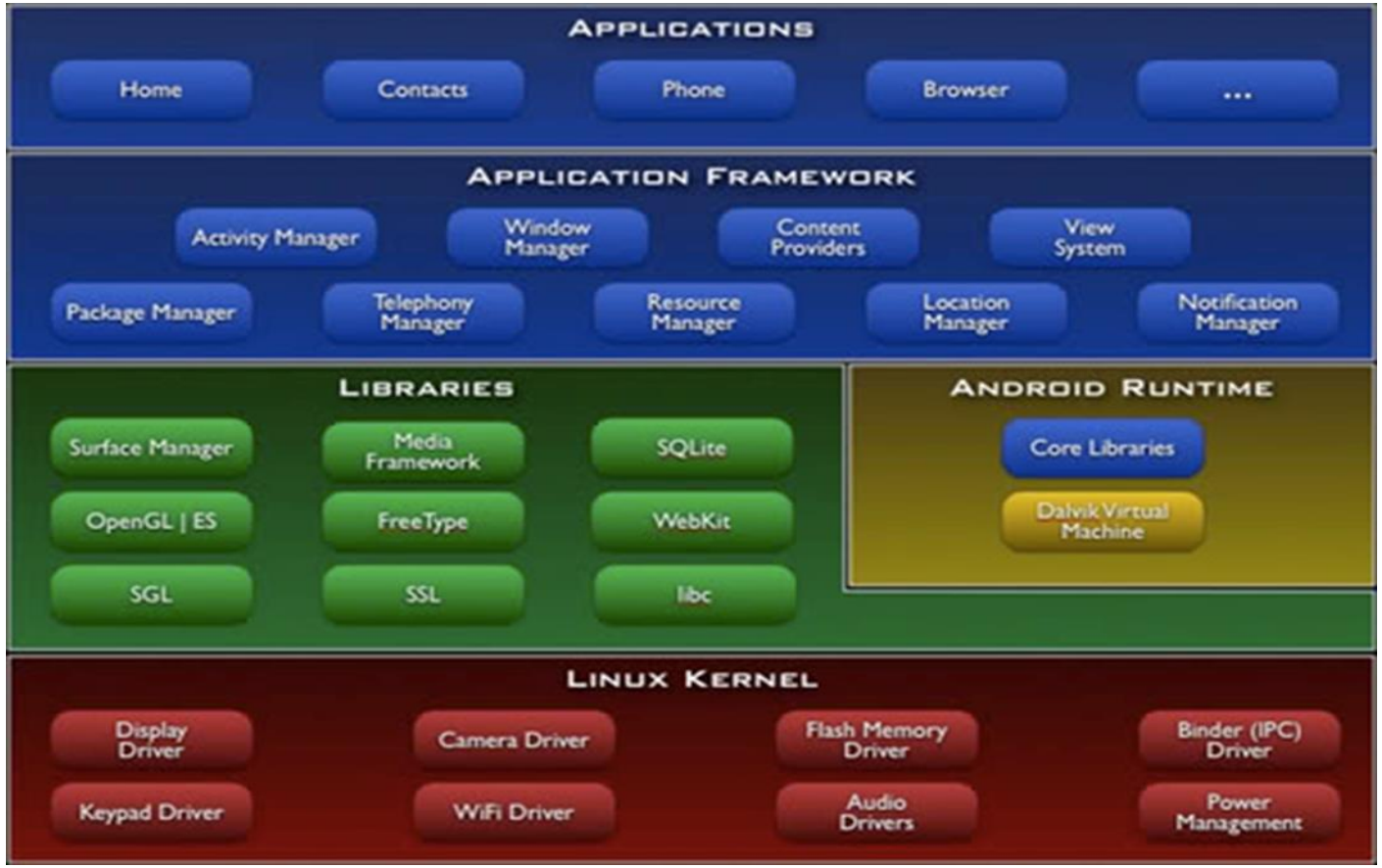
# Đặc tính của Android

- XML trong Android
  - Không giống như lập trình java thông thường, lập trình android ngoài các lớp được viết trong *.java còn sử dụng XML để thiết kế giao diện cho ứng dụng.
  - Nền tảng Java đã và đang hỗ trợ rất nhiều cách khác nhau để làm việc với XML trong thời gian nhất định, và hầu hết các API có liên quan đến XML của Java đều được hỗ trợ đầy đủ trên Android.
  - Layout được dùng để quản lý các thành phần giao diện theo 1 trật tự nhất định:

# Đặc tính của Android

- Linear Layout: tương đối được sử dụng nhiều để bố cục các View một cách đơn giản nhưng hiệu quả, các View được bao trong Linear Layout sẽ được hiển thị theo chiều dọc hoặc theo chiều ngang.
- Relative Layout: có thể cho người thiết kế tùy biến dễ dàng hơn trong giao diện của mình, các thành phần trong layout này có thể được định vị thông qua nhau bằng thuộc tính id .Đây là điều mà Linear layout không thể làm được.
- Table Layout: ta sẽ sử dụng layout này để dàn các View theo hàng.
- Frame Layout: chứa các View con và  đối với các View con chứa trong layout này sẽ hoạt động theo kiểu stack.
- Absolute Layout :cho phép định vị tọa độ của các View mà nó chứa.

# Cấu trúc nền tảng Android



# Cấu trúc nền tảng Android

- LINUX KERNEL

Được ra đời năm 1991,bất cứ 1 hệ điều hành nào muốn hoạt động được thì phải có 1 nền tảng nhất định.Và Linux Kernel chính là nền tảng của Android nhưng được chỉnh sửa lại cho tối ưu.

Lý do để chọn  Linux Kernel

- Có các driver để giao tiếp trực tiếp với phần cứng cho riêng từng mẫu di động. Muốn nhúng được android vào một mẫu di động mới, hãng phát triển sẽ phải viết lại toàn bộ hoặc một phần các driver trong kernel. Hiện tại linux kernel phiên bản mới nhất cũng hỗ trợ khá nhiều các loại phần cứng.
- Nền tảng này đảm nhiệm khá đầy đủ các chức năng cơ bản (quản lý bộ nhớ, lập lịch, quản lý process, file system, ...)
- Linux kernel đã được chứng minh qua thời gian.

# Cấu trúc nền tảng Android

## Libraries

Library được viết bằng C và C++.Đó là nơi chứa sức mạnh cốt lõi của nền tảng Android.Nó cung cấp các hàm cho các ứng dụng thực hiện các thuật toán phức tạp.Android có 9 nhóm thư viện chính

- **Surface Manager** : chịu trách nhiệm tạo ra các cửa sổ khác nhau của từng ứng dụng vào các thời điểm khác nhau trên màn hình
- **OpenGL | ES/ SGL** : phần cốt lõi của đồ họa bao gồm 3D/2D.Có thể kết hợp đồ họa 2D và 3D trong 1 ứng dụng.
- **Media Framework**: được cung cấp bởi PacketVideo thư viện hỗ trợ giải mã và ghi âm các chuẩn âm thanh, hình ảnh, video phổ biến (MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG).
- **SQLite**: đây là cốt lõi hầu hết của các dữ liệu lưu trữ.
- **FreeType**: thư viện giải mã và hiển thị các font chữ.
- **Libc**: thư viện biên dịch mã nguồn các chương trình viết bằng ngôn ngữ C.

# Cấu trúc nền tảng Android

- **Webkit**: là công cụ trình duyệt mở  trong Android và cũng được sử dụng trong Safari (Mac OS).Chúng được các kĩ sư Google chỉnh sửa để phù hợp hơn với thiết bị di động.
- **SSL**: thư viện hỗ trợ kết nối tới máy chủ qua giao thức SSL (giao thức kết nối có mã hõa và bảo mật)
- **SGL** là viết tắt của "Thư viện đồ họa Khả năng mở rộng" và các hệ thống đồ họa
được sử dụng Android. SGL là các thư viện đồ họa ở mức độ thấp thực hiện bản địa mã xử lý dựng hình. Nó hoạt động song song với các lớp cao cấp của khuôn khổ này (đặc biệt là WindowManager SurfaceManager) để thực hiện đồ họa Android

# Cấu trúc nền tảng Android

ANDROID RUNTIME

- Máy ảo Dalvik được thiết kế đặc biệt cho Android nơi mà môi trường nhúng có nhiều hạn chế về pin,bộ nhớ và CPU. Máy ảo Dalvik tối ưu hóa các thiết bị phần cứng.
- Core libraries bao gồm các tiện tích và công cụ được xây dựng bằng ngôn ngữ Java

# Cấu trúc nền tảng Android

APPLICATION FRAMEWORK

- Các công cụ ,tiện ích và ứng dụng trong Application framework được viết bởi Google hoặc các nhà ứng dụng bằng ngôn ngữ Java. Cũng giống như .net cung cấp sẵn các class để cho người lập trình kế thừa lại một cách dễ dàng. Kèm theo đó là một môi trường phát triển phần mềm tích hợp (Android SDK) khiến cho việc tạo ra các phần mềm ứng dụng cho nền tảng này dễ dàng hơn, thuận lợi hơn. Dưới đây là 1 số Manager quan trọng
- **Activity Manager**: quản lý các Activity- phần quan trọng nhất của 1 ứng dụng giúp điều tiết tài nguyên cho các ứng dụng khác nhau khi chạy cùng 1 lúc nhằm đạt được sự tối ưu cao nhất tránh các tình trạng treo máy không cần thiết.
- **Package Manager**: quản lý các Package (gói) của ứng dụng được cài đặt trên thiết bị của bạn
- **Notification Manager**:cho phép các ứng dụng hiển thị cảnh báo trên thanh trạng thái.
- **Content Providers**: cho phép các ứng dụng được truy suất data của các ứng dụng khác hoặc chia sẻ data cho các ứng dụng khác.
- **Telephony Manager**: chứa các API (Applications Programming Interface)

# Cấu trúc nền tảng Android

APPLICATIONS

- Bao gồm tất cả những ứng dụng trong thiết bị bản đồ, trình duyệt, danh bạ, SMS …….. tất cả được viết bằng ngôn ngữ Java.
- Số lượng phần mềm trên Android Market đã vượt 200.000 nhưng còn chưa thể so sánh với các nền tảng khác (Symbian, iOS,…). Nhưng đây thực sự là 1 môi trường tốt cho những nhà phát triển phần mềm vì nếu xét về mặt tốc độ thì Android đang phát triển khá nhanh so với các nền tảng khác.

# Ứng dụng Android

- Mỗi ứng dụng đều chạy trên 1 process
- Mỗi process có 1 máy ảo Java riêng (Dalvik VM)
- Ứng dung Android không  có điểm bắt đầu ( không có hàm main)
- Thành phần cơ bản của 1 ứng dụng Android

1. View: Thành phần UI thiết lập giao diện người dùng, View có tính thứ cấp
2. AndroidManifest.xml
3. Activity
4. Content Provider
5. Intent
6. Service
7. BroadcastReceiver

# Vai trò của Manifest file trong Application

Manifest file khai báo tất cả thành phần của Application

- Chỉ rõ bất cứ sự cho phép các yêu cầu của ứng dụng
- Khai báo Level API thấp nhất được yêu cầu bởi ứng dụng
- Khai báo đặc điểm về phần cứng, phần mềm được sử dụng hay yêu cầu bởi ứng dụng , như Camera, Bluetooth
- API thư viện mà ứng dụng cần gọi đến như thư viện Google Maps
- Vài chức năng khác

Cấu trúc định nghĩa Manifest file

- Chỉ ra có bao nhiêu Activity được sử dụng, Service, Content Provider
- Thiết lập sự cho phép sử dụng của ứng dụng.
- Chỉ ra thư viện ngoài

# Các khái niệm cơ bản

- Các khái niệm trong ứng dụng Android: Một ứng dụng Android được xây dựng nên từ nhiều thành phần (component).
- Các component để xây dựng nên một ứng dụng Android được chia thành 4 component chính: Activity, Service, Content Provider, Broadcast Receiver cùng với 2 component khác là: Intent và Notification.

# Các khái niệm cơ bản

- Vòng đời của một ứng dụng Android (Android Application Life Cycle): Android có cơ chế quản lý các process theo chế độ ưu tiên. Các process có độ ưu tiên (priority) thấp sẽ bị Android giải phóng mà không hề cảnh báo nhằm đảm bảo tài nguyên.
  - Foreground process: là process của ứng dụng hiện thời đang được người dùng tương tác.
  - Visible process: là process của ứng dụng mà activity đang hiển thị đối với người dùng (onPaused() của activity được gọi).
  - Service process: là Service đang running.

# Các khái niệm cơ bản

- Background process: là process của ứng dụng mà các activity của nó không hiển thị với người dùng (onStoped của activity được gọi).
- Empty process: process không có bất cứ 1 thành phần nào active.

Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động kill process, trước tiên là các empty process.

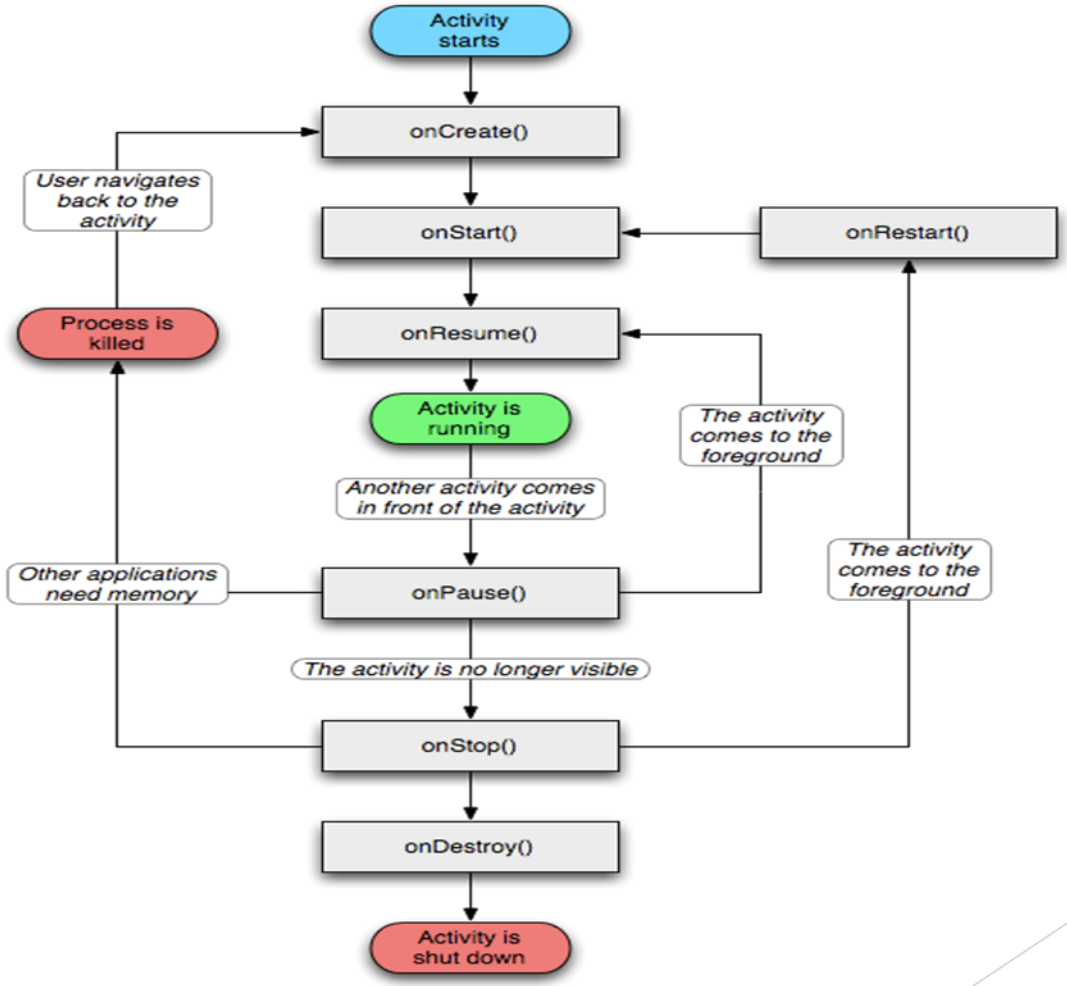
# Activity

- Activity là phần không thể thiếu và là phần quan trọng nhất trong 1 ứng dụng của Android.
- Activity là giao diện của ứng dụng,nơi người dùng có thể tương tác với ứng dụng.Theo đó, 1 ứng dụng có thể không có activity nào (ứng dụng chạy không có giao diện),có 1 hay nhiều hơn 1 activity

# Activity

- Activity : Được dùng để hiển thị một màn hình trong ứng dụng cho phép người dùng tương tác với ứng dụng.
- Các trạng thái (state) của Activity:
  - Active (running): Activity đang hiển thị trên màn hình (foreground).
  - Paused: Activity vẫn hiển thị (visible) nhưng không thể tương tác (lost focus).
  - Stop: Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop
  - Killed: Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái stop hoặc paused cũng có thể bị giải phóng và khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục hồi lại trạng thái trước đó.

# Life cycle callback



# Quản lý Activity Life Cycle

- Quản lý life Cycle của Activity được thực thi bởi phương thức callback
- Về bản chất tồn tại 3 trạng thái
  1. Resume : Activity được hiển thị ở ‘foreground’ của màn hình, và đang được ‘focus’
  2. Paused: Activity khác đang hiển thị ở ‘Foreground’, đang được ‘focus’, nhưng vẫn có thể thấy được
  3. Stopped: không còn nhìn thấy ở màn hình, đang ở chế độ ‘Background’. Vẫn có thể sống lại được, nhưng có thể bị tắt bởi hệ thống

# Các phương thức trong Activity

- Create() : được gọi khi activity được tạo lần đầu tiên
- Start() :  được gọi khi activity  visible với người dùng
- Resume : được gọi khi activity bắt đầu tương tác với người dùng
- Pause : được gọi hệ thống muốn resume 1 activity khác và activity hiện tại được dừng lại
- Stop : được gọi khi activity không hiển thị cho người sử dụng. Nó có thể diễn ra khi đang bị hủy, hay 1 activity khác vừa được resume và bao phủ lấy nó
- Restart : được gọi khi activity đã được dừng và đang khởi động 1 lần nữa
- Destroy : được gọi trước khi activity được hủy khỏi hệ thống. Đây là lời gọi cuối cùng activity này nhận được. Nó được gọi bởi activity được hoàn thành,hay bị hủy để tiết kiệm bộ nhớ

# Content Provider

- Một Content Provider cung cấp một tập chi tiết dữ liệu ứng dụng đến các ứng dụng khác. Thường được sử dụng khi chúng ta muốn tạo cơ sở dữ liệu dưới dạng public (các ứng dụng khác có thể truy xuất ).
  Dữ liệu thường được lưu trữ ở file hệ thống, hoặc trong một SQLite database.
  Đơn giản như : Danh bạ, Call log, cấu hình cài đặt...trên điện thoại là dữ liệu dưới dạng Content Provider.
- Content Provider hiện thực một tập phương thức chuẩn mà các ứng dụng khác có thể truy xuất và lưu trữ dữ liệu của loại nó điều khiển.
  Tuy nhiên, những ứng dụng không thể gọi các phương thức trực tiếp. Hơn thế chúng dùng lớp Content Resolver và gọi những phương thức đó. Một Content Resolver có thể giao tiếp đến nhiều content provider; nó cộng tác với các provider để quản lý bất kỳ giao tiếp bên trong liên quan.

# intent

- Intent là cấu trúc dữ liệu, được sử dụng để tải Activity, gửi thông tin cho BroadcastReceiver, hoặc gọi 1 Service
- Cung cấp những đặc trưng để thực hiện liên kết giữa những ứng dụng khác nhau
- Là cấu nối giữa các Activity, BroadcastReceiver, hay Service với nhau trên cùng ứng dụng hay trên các ứng dụng khác nhau
- Thuộc tính của Intent
  1. Action
  2. Category
  3. Type
  4. Data
  5. Component
  6. Extras

# Service

- Là một trong những thành phần của Android, được chạy trên background thực hiện những công việc tính toán không đòi hỏi giao diện
- Service giống như Activity thực hiện theo chu kỳ thời gian để quản lý sự thay đổi của trạng thái
- Có 2 loại Service:

1. "Started" Service: Service được gọi khi thành phần của ứng dụng gọi nó bằng phương thức startService(), và service sẽ chạy vô thời hạn. Loại service này thực hiện công việc đơn lẻ như download, tính toán và service sẽ dừng một khi công việc hoàn thành
2. "bound" Service: Service được gọi khi thành phần của ứng dụng gọi nó bằng phương thức bindService(). Loại service đề nghị giao diện Client – Server cho phép thành phần ứng dụng tương tác với nó, gửi yêu cầu, lấy kết quả. Một khi tất cả thành phần ứng dụng ngừng liên kết, Service kết thúc

# BroadcastReceiver

- Thành phần thu nhận các Intent và sẽ tự động khởi tạo ứng dụng phù hợp để đáp ứng các Intent. Được sử dụng để nhận những Intent message được gửi từ sendBroadcast() hoặc từ hệ thống.

# Application Resources

- Sự cần thiết cung cấp tài nguyên để lựa chọn cho cấu hình đặc thù của thiết bị cụ thể như ngôn ngữ khác nhau, kích thước màn hình.. Ngày càng trở nên quan trọng như nhiều thiết bị hỗ trợ Android có sẵn với các cấu hình khác nhau
- Để cung cấp khả năng tương thích với các cấu hình khác nhau, ta phải tổ chức các tài nguyên trong thư mục res dự án
- Bất cứ loại tài nguyên nào, có thể chỉ ra loại mặc định và tài nguyên được lựa chọn cho ứng dụng

1. Tài nguyên mặc định được sử dụng khi không quan tâm đến cấu hình của thiết bị hay không có bất kỳ tài nguyên để lựa chọn
2. Tài nguyên để lựa chọn được thiết kế cho những cấu hình đặc thù, bằng cách vào từ hạn định cấu hình phù hợp ở tên đường dẫn

# Application Resources

- Cung cấp tài nguyên
- Truy cập tài nguyên
- Các loại tài nguyên

# Cung cấp tài nguyên

- Cần đặt mỗi loại tài nguyên vào thư mục con riên

```
MyProject/
    src/
        MyActivity.java
    res/
        drawable/
            icon.png
        layout/
            main.xml
            info.xml
        values/
            strings.xml
```

- Thư mục tài nguyên được hỗ trợ trong thư mục /res của ứng dụng

# Cung cấp tài nguyên

- Lưu tài nguyên để lựa chọn vào riêng từng thư mục vừa tạo. Những tập tin tài nguyên này phải có tên giống như tập tin của tà

```
res/
    drawable/
        icon.png
        background.png
    drawable-hdpi/
        icon.png
        background.png
```

# Truy Cập Tài Nguyên

- Có 2 cách truy cập tài nguyên

1. Trong Code:

   Sử dụng lớp con của lớp R, ví dụ: R.string.hello

2. Trong XML tập tin:

   Sử dụng cú pháp đặc biệt XML phù hợp với ID tài nguyên được định nghĩa trong lớp R, ví dụ: @string/hello

string: là loại tài nguyên,

hello : là tên tài nguyên

# Các Loại Tài Nguyên

- Animation Resources - res/anim/, định nghĩa animation
- Color State List Resource - res/color/, định nghĩa màu
- Drawable Resources - res/drawable/, định nghĩa đồ họa khác nhau với bitmap,xml
- Layout Resource - res/layout/, định nghĩa layout của ứng dụng
- Menu Resource - res/menu/, định nghĩa nội dụng của Menu
- String Resources - res/values/, định nghĩa string , mảng string
- Style Resource - res/values/, định nghĩa cái nhìn, định dạng thành phần UI
- More Resource Types - res/values/, định nghĩa boolean, integers, dimensions, colors, ….

# Môi trường phát triển Android

- Hệ điều hành hỗ trợ
  1. Windows XP (32-bit) or Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
  2. Mac OS X 10.4.8 or later (x86 only)
  3. Linux (tested on Ubuntu Linux, Lucid Lynx)
- Môi trường phát triển hỗ trợ
  1. Eclipse 3.5 (Galileo) hoặc lớn hơn
  2. JDK 5 or JDK 6 (JRE alone is not sufficient)
  3. Android Development Tools plugin (optional)
  4. Android Studio
  5. Kotlin
- Môi trường phát triển hoặc IDEs khác
  1. JDK 5 or JDK 6 (JRE alone is not sufficient)
  2. Eclipse

# Công cụ hỗ trợ





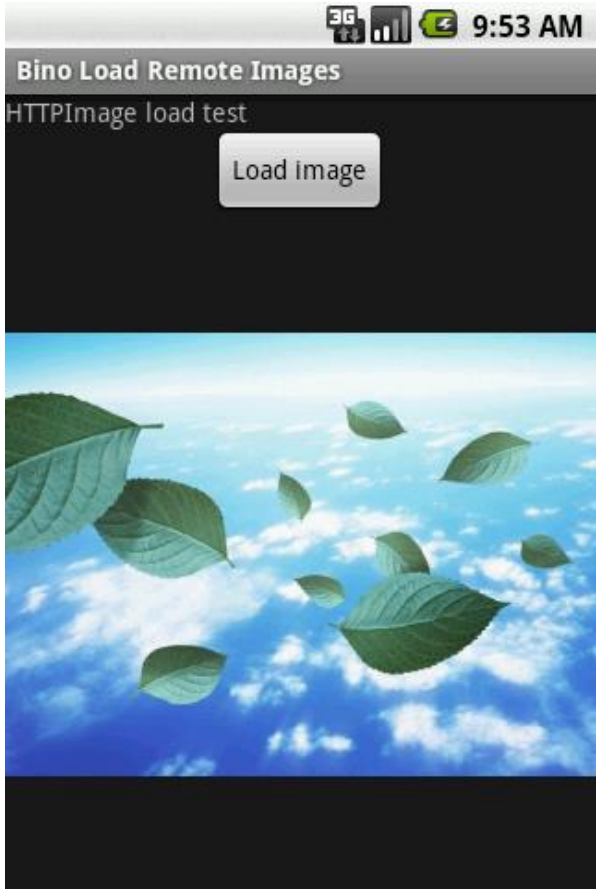
Kotlin

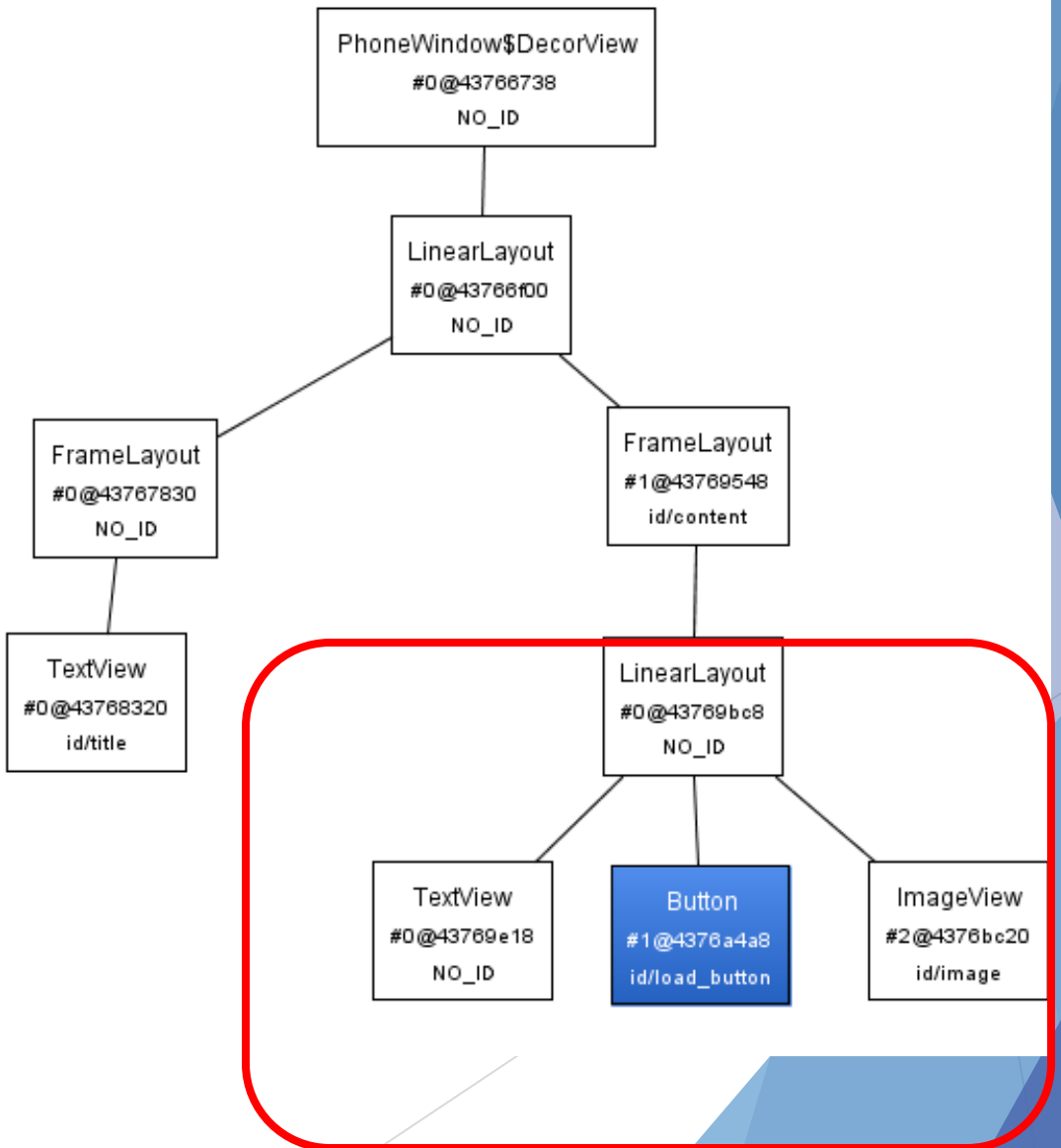
# Thiết kế giao diện trên Android

# Thiết kế giao diện

- Trong Android, dùng Activity để hiển thị màn hình.
- Mỗi activity sẽ chứa các View theo dạng cấu trúc cây, nghĩa là một Layout gốc chứa các view/layout con bên trong hoặc chỉ có 1 view duy nhất. (lưu ý Layout cũng là một view)
- Có thể thiết kế giao diện trong code java hoặc trong file xml trong thư mục layout.

# Tree view


9:53 AM
Bino Load Remote Images
HTTPImage load test
Load image


PhoneWindow$DecorView
#0@43766738
NO_ID
LinearLayout
#0@43766f00
NO_ID
FrameLayout
#0@43767830
NO_ID
FrameLayout
#1@43769548
id/content
TextView
#0@43768320
id/title
LinearLayout
#0@43769bc8
NO_ID
TextView
#0@43769e18
NO_ID
Button
#1@4376a4a8
id/load_button
ImageView
#2@4376bc20
id/image

# Layout mẫu của helloworld

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent”>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text=“@string/hello“ />
</LinearLayout>
```

# Một số thuộc tính cơ bản

- Layout_width, layout_height: chiều rộng của view (fill_parent là to bằng kích thước của layout chứa view này, wrap_content là vừa đủ nội dung cần hiển thị của view)
- Orientation: với LinearLayout, việc sắp xếp các view là nằm kề nhau theo hàng ngang hoặc hàng dọc, ta khai báo orientation để chọn sắp theo kiểu nào (horizontal/vertical)

# Một số thuộc tính cơ bản

- Gravity: thuộc tính này qui định các view nằm bên trong layout sẽ đặt theo vị trí nào so với layout (trung tâm, trái , phải, trên dưới…)
- Weight: để các view phân chia tỉ lệ diện tích hiển thị trên màn hình (tỉ lệ tính theo weight của từng view trên tổng số weight, các view ko khai báo weight thì sẽ xem qua width và height)

# Q/A