

# Basi Android

- Cosa è Android?
  - Android is a **mobile operating system** based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.
- Android è “free and open source”
  - Basato su tecnologie open source
    - Licenza Open Source Apache 2.0
      - Chi vuole usare Android non deve pagare
    - <http://androidxref.com/>
  - La maggior parte dei device contiene codice proprietario
    - <https://www.makeuseof.com/tag/android-really-open-source-matter/>

# Perché studiamo Android

- Design Pattern
- Multi Thread Pattern
- Programmazione Client Server

# Android Developer

<https://developer.android.com/>

developers 

Platform

Android Studio

Google Play

Jetpack

Kotlin

Docs

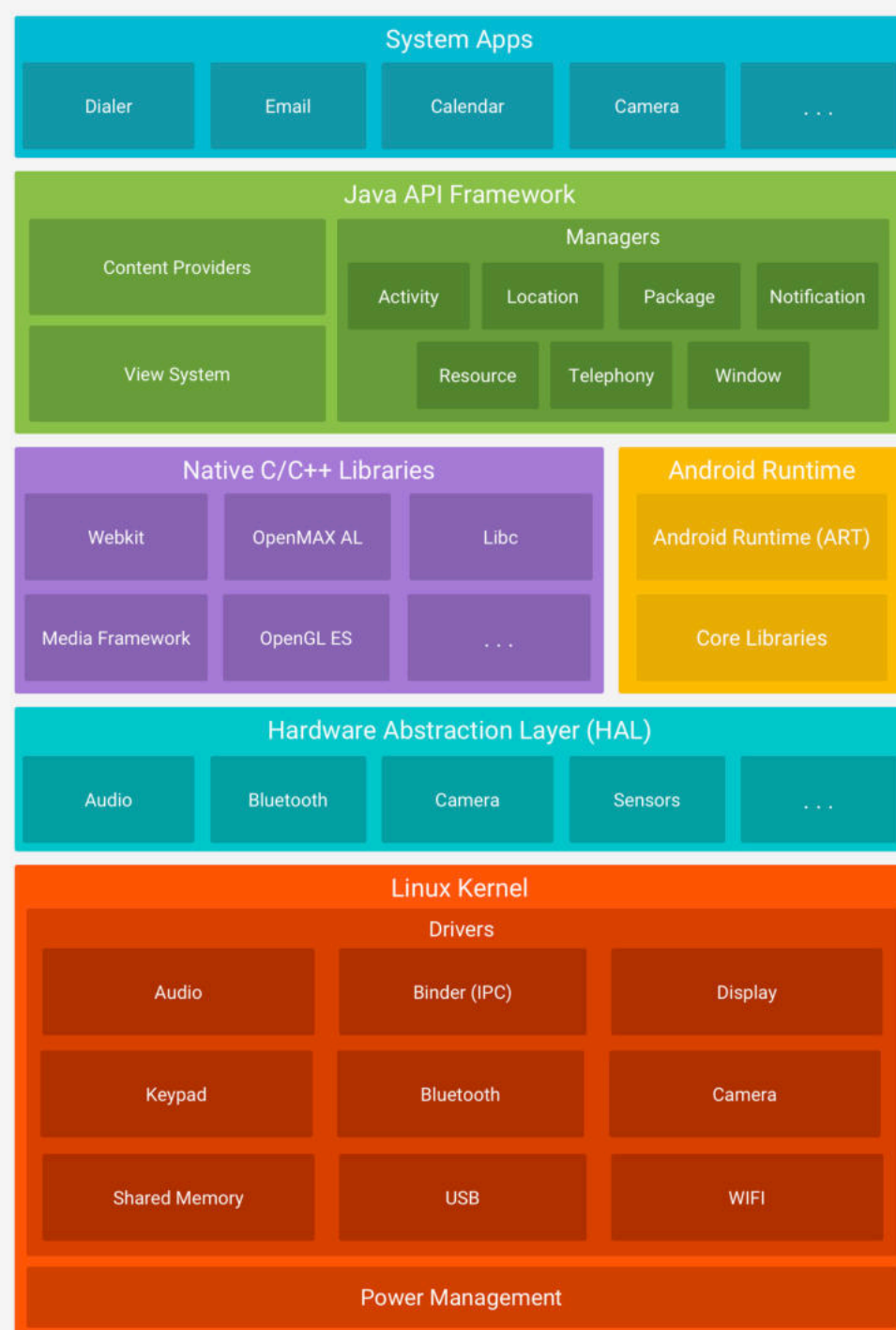
News

## Build anything on Android

 Search

Android 12 Developer Preview is here! [Try it out, and give us your feedback!](#)

# Architettura



<https://developer.android.com/guide/platform>

# Virtual Machine

- Android usa la Android RunTime (ART)
  - Non la Java Virtual Machine
- Motivi
  - Ahead-of-time (AOT) and just-in-time (JIT) compilation
  - Optimized garbage collection (GC)
  - Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields
- Prima della V5.0 usava Dalvik VM

<https://source.android.com/devices/tech/dalvik/jit-compiler>

# Tool Android

- Software Development KIT
  - Tools per sviluppo applicazioni java
- Android Studio
  - IDE basati su IntelliJ
- ~~Eclipse Android Development Tools~~
  - ~~plug in di Eclipse non più supportato~~
- Native Development Kit
  - Sviluppo di applicazioni o parte di esse in C++

# Struttura SDK

- Tools

- Strumenti per la gestione dei progetti
  - `android`, `emulator`, etc.

- Platforms

- Versioni di android
  - 4.x, 5.x,...,10.x, Cellulari e Tablet, TV, ...
- API level
  - [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history)

- Extra

- librerie google ed altro

<https://developer.android.com/studio/features>

## Features overview

### Release notes

Android Studio

SDK command-line tools

Android emulator

SDK build tools

Android Gradle plugin

SDK platform

[SDK platform tools](#)



# Applicazioni Android

- Sono scritte in linguaggio java
  - oggi anche in Kotlin (<https://kotlinlang.org/>)
- Sono contenute in un Android package file .apk
  - in cui c'è il codice, le risorse e i file di configurazione
- Vivono in un ambiente isolato rispetto ad altre app
  - ogni app è eseguita da un utente Linux diverso
  - ogni app gira nella sua VM
  - i componenti di un app sono eseguiti in un processo separato

# Componenti di un app

- Sono i blocchi essenziali che compongono le app
  - hanno funzioni e ciclo "vitale" differente
  - ogni componente è un entry point attraverso il quale l'utente e/o il sistema accedono all'app
- Componenti
  - Activity
  - Service
  - Content Provider
  - Broadcast Receiver

# Activity

- Rappresenta una schermata dell'applicazione
- In generale una applicazione contiene varie Activity
- Quando avviamo una nuova attività la corrente è posta in background
  - ho uno stack di activity
- Le GUI si realizzano in XML o java
- Si controlla la vita dell'activity mediante callback
  - onCreate, onPause,...

# Service

- Servono ad eseguire operazioni in background
- Non hanno interfaccia grafica
- Servono ad operazioni come suonare la musica, gestire la rete, leggere la posizione, etc.
- Sono eseguiti indipendentemente da chi li ha creati
- Possono avere una interfaccia verso altri componenti

# Content Provider

- Sono usati per memorizzare e leggere dati rendendoli accessibili alle applicazioni
- Sono l'unico modo di condividere dati fra applicazioni
- Espongono una URI pubblica per individuare univocamente i propri dati
- I dati sono esposti in forma di una singola tabella di database
- ci sono CP predefiniti per contatti, media, etc.

# Broadcast Receiver

- Sono componenti che rispondono ad “annunci” inviati a tutti i componenti dal sistema
- Ex. lo schermo si spegne, la batteria e scarica, c’è una chiamata in arrivo, etc.
- Anche le app possono inviare annunci broadcast

# Manifest file

- Ogni applicazione contiene il file `AndroidManifest.xml`
- Presenta l'applicazione al sistema operativo
- Descrive i componenti dell'applicazione
- Dichiarare i permessi necessari all'applicazione
- Dichiarare le API minime necessarie all'applicazione

# ANDROID HELLO WORLD

<https://codelabs.developers.google.com/codelabs/android-training-hello-world/index.html?index=..%2F.index#0>





# AMBIENTE DI SVILUPPO

# Progetto Android

- Creare un progetto Android
  - Se vogliamo creare un app Android non basta scrivere un file:
  - Devo usare i tool a disposizione
- Due tipi di progetto
  - dipende dal sistema di compilazione
- Compilatori
  - ~~– ant – quello legato ad ADT~~
  - gradle – quello usato in Android Studio

# Struttura progetto HelloWorld

**gradle** – tool di compilazione

**src** – tutto il codice e le risorse

**build.gradle** - file per la compilazione

altri file per la compilazione

# Compilare un progetto gradle



- Android per compilare usa gradle
  - Sistema di compilazione/installazione automatico
- Il comando da dare è
  - > `./gradlew assembleDebug`
- Per vedere tutti comandi
  - `./gradlew tasks`

# Android Virtual Devices

- **emulator**
  - Applicazione che emula un device Android completo
    - Linux per ARM, librerie native, DVM, packages, applicazioni
  - Usa QEMU per emulare il device fisico
  - Vuole come parametro un AVD
- **Android Virtual Device**
  - Sono le caratteristiche del device emulato
    - Es. Hardware, target, display, memoria, etc.
- **Creare AVD**
  - > `android create avd -n my_avd -t 1`
- **Lanciare l'emulatore**
  - > `emulator -avd my_avd`

# Installare ed eseguire

- Vedere la lista dei device

- adb devices

- emulator-5554 device

- Installare

- > adb -s emulator-5554 install HelloWorld-debug.apk

- Eseguire ...

# Android Debug Bridge (adb)

- è un command-line tool per interagire con un device android
- Comprende
  - un client (adb in platform-tool)
  - un demone che gira nel dispositivo
    - va attivato (USB Debugging)
  - un server nella nostra macchina

<https://developer.android.com/studio/command-line/adb>

Esempi di uso

```
adb shell am start -n com.package.name/com.package.name.ActivityName
```

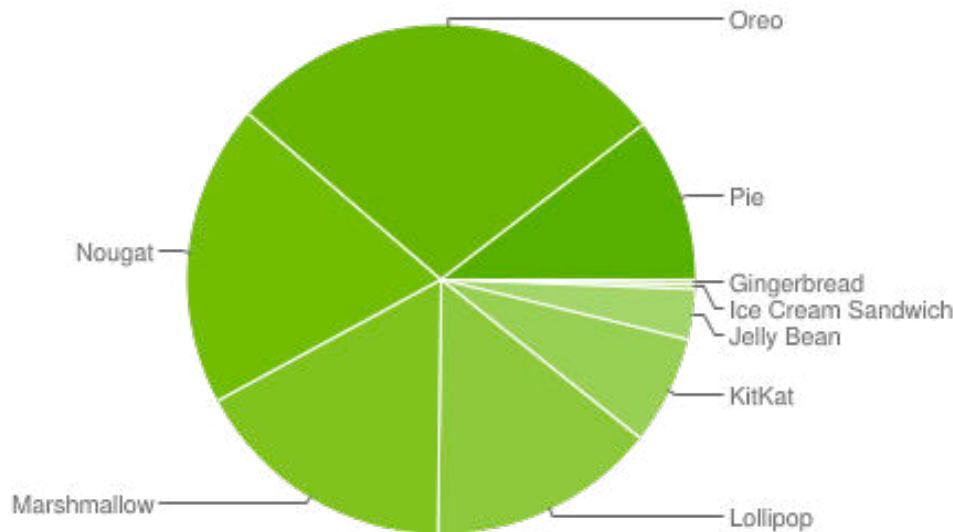


# Android Studio

- Sistema di sviluppo sviluppato intorno ad IntelliJ
  - Si connette ai tool di sviluppo presenti nel sistema
    - es. java jdk, sdk android, g++ gcc gdb, etc.
- Integra gradle
- Semplifica le operazioni fatte prima
  - ci evita di operare a linea di comando

# Version: Min e Target

- La minSDK è il livello minimo di SDK che serve all'app
  - Google Play non la mostra
- La target SDK è quella usata per sviluppare



<https://developer.android.com/about/dashboards/index.html#Platform>