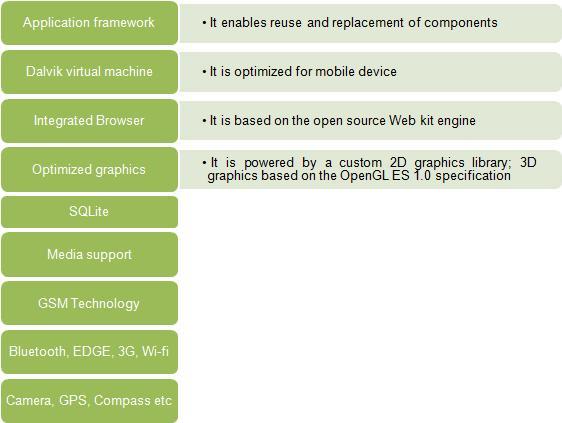
**#What is Android and why it is popular platform? Mention few features of Android operating system**.

[Operating Systems](http://www.engineersgarage.com/articles/operating-systems-tutorial) have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

**Features & Specifications**

**Android** is a powerful Operating System supporting a large number of applications in [Smart Phones](http://www.engineersgarage.com/articles/smart-phones). These applications make life more comfortable and advanced for the users. Hardwares that support Android are mainly based on [ARM architecture](http://www.engineersgarage.com/articles/arm-advanced-risc-machines-processors) platform. Some of the current features and specifications of android are:



Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides **Android SDK** (Software development kit). Read more about [open source software](http://www.engineersgarage.com/articles/open-source-software-history-advantages).

**#What is API ? Mention the android version name sequentially.**

In [computer programming](https://en.wikipedia.org/wiki/Computer_programming), an **application programming interface** (**API**) is a set of [subroutine](https://en.wikipedia.org/wiki/Subroutine) definitions, [protocols](https://en.wiktionary.org/wiki/Protocol), and tools for building [application software](https://en.wikipedia.org/wiki/Application_software). In general terms, it is a set of clearly defined methods of **communication** between various software components. A good API makes it easier to develop a [computer program](https://en.wikipedia.org/wiki/Computer_program) by providing all the building blocks, which are then put together by the [programmer](https://en.wikipedia.org/wiki/Programmer). An API may be for a web-based system, [operating system](https://en.wikipedia.org/wiki/Operating_system), [database system](https://en.wikipedia.org/wiki/Database_system), [computer hardware](https://en.wikipedia.org/wiki/Computer_hardware) or [software library](https://en.wikipedia.org/wiki/Library_(computing)).

List of android version:

**Cupcake:** The Android version 1.5 *Cupcake* is the first version, which got a name assigned, which is Cupcake. Cupcakes are muffin-like cakes, which mostly has a creamy hood.

**Donut:** Version 1.6 *Donut* got the name *Donut*.

**Éclair:** Android-Version 2 *Eclair* and 2.1 *Eclair* share the same name, Eclair. Eclairs are longly biscuits overdrawn with chocolate.

**Froyo:** *Frozen Yogurt* is a ice-like dessert made with milk and yogurt and is also the name of the Android version 2.2 *Froyo*.

**Gingerbread:** The name of Android version 2.3 *[verweis=Gingerbread](https://en.droidwiki.org/wiki/File:Gb.png)"Gingerbread"* is Gingerbread, which are cake-like biscuits mostly eaten around christmas time.

**Honeycomb:** The android version mostly made for tablets, Android 3.0 *Honeycomb*, is called *Honeycomb*. The bootanimation of this android version is also made like a honeycomb.

**Ice Cream Sandwich:** Android 4.0 is called after a sweet, which is made of delicious ice cream bewtween two biscuits: an Ice Cream Sandwhich.

**Jely Bean:** Jelly beans are colorful, mostly made out of sugar, sweets, which have a hard shell and are filled with jelly. The following android versions share this name:

* 4.1
* 4.2
* 4.3

**KitKat:** KitKat is a brand from Nestle, and is a chocolate bar filled with waffles. Android 4.4 is named after this sweet: 4.4 *[verweis=KitKat](https://en.droidwiki.org/wiki/File:Android_KitKat.png)"KitKat"*.

**Lollipop:** Lollipops are sweets in different flavors on a stalk and is also the name of the following android versions:

* 5.0
* 5.1
* 5.1.1

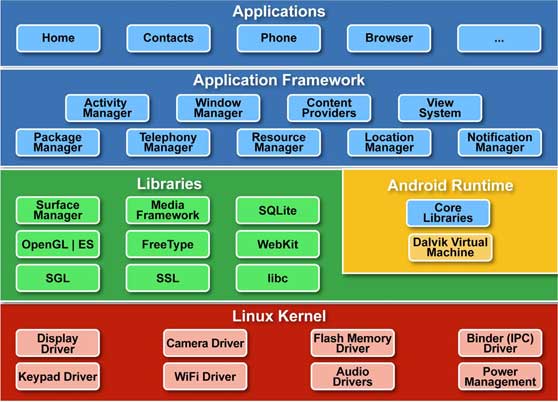
**Marshmallow:** The android version 6.0, announced at 17th of august in 2015 is called after the sweet Marshmallow.

**Nougat:** Android 7, also called [Android N](https://en.droidwiki.org/w/index.php?title=Android/N&action=edit&redlink=1), officialy got the name *Nougat*.

**Oreo:** Android 8, also called Android O genannt, carries the name Oreo, which are two crispy chocolate cookies with a vanilla cream center.

**# How many layers are in Android architecture? Mention the android sections name with few components.**

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



Link: [description](mailto:https://letsknowaboutandroid.wordpress.com/about/)

* **The Linux kernel**
  + has Linux Version 2.6.x for core system services and thus android handles only “Kernel” portion in Linux
  + **Core Libraries**
  + Uses the JAVA Programming Language
  + With the specific instance of the Dalvikvirtual machine (DVM), android Application operates in its own process.The DalvikVM is a Java based license free VM. It runs Register based VM and provides enhancement for low memory requirements. It executes files in the DalvikExecutable (.dex) format in which the DX tool changes classes to .dexformat.

**libraries**

1. *Libc* : it is c standard lib.
2. *SSL*: Secure Socket Layer for security
3. *SGL*: 2D picture engine where SGL is “Scalable Graphics Library”
4. *OpenGL/ES*: 3D image engine
5. *Media Framework*: essential part of Android multi-media
6. *SQLite*: Embedded database
7. *Lib web core*: Kernel of web browser
8. *Free Type*: Bitmap and Vector
9. *Surface Manager*: Manage different windows for different applications

**Application framework**

* unlimited application
* Equality of each apps
* Easy to embed web browser
* Simultaneous running

**The Design goal of Android**:

* Openness

–Be as flexible as possible

–How it handles access to data

–Rapid development (XML, Java)

* Develop Language

–App: Java

–Framework: Java

–Libraries: C/C++

–OS & Driver: C

**DEVELOP FOR ANDROID**

**APPLICATION ARCHITECTURE**

* Language: Java
* Virtual Machine:

–DalvikVM, not JVM and

–Open source

* Application: consists of one or more of the following classifications

–Activities and Services

–Content providers and Broadcast receivers

**# Explain AndroidManifest.xml file in detail.**

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code. Among other things, the manifest does the following:

* It **names the Java package** for the application. The package name serves as a unique identifier for the application.
* It **describes the components of the application** — the activities, services, broadcast receivers, and content providers that the application is composed of.
* It **names the classes** that implement each of the components and publishes their capabilities (for example, which [Intent](https://stuff.mit.edu/afs/sipb/project/android/docs/reference/android/content/Intent.html) messages they can handle). These declarations let the Android system know what the components are and under what conditions they can be launched.
* It **determines** which processes will **host application components**.
* It **declares** which **permissions** the application must have in order **to access protected parts** of the API and interact with other applications.
* It also **declares the permissions** that others are required to have in order **to interact** with the application's components.
* It **lists the**[**Instrumentation**](https://stuff.mit.edu/afs/sipb/project/android/docs/reference/android/app/Instrumentation.html)**classes** that provide profiling and other information as the application is running. These declarations are present in the manifest only while the application is being developed and tested; they're removed before the application is published.
* It **declares the minimum** level of the Android **API** that the application requires.
* It **lists the libraries** that the application must be linked against.

**# Sketch the android application development process.**

****

* **Planning**
* **Expert Market Research**
* **Critical Analysis**
* **App Implementation**
* **Testing & Integration**
* **Software Release**

**# What is orientation in Android?**

The **screenOrientation** is the attribute of activity element. The orientation of android activity can be portrait, landscape, sensor, unspecified etc. You need to define it in the AndroidManifest.xml file. For example:

1. **<activity**
2. android:name="com.example.screenorientation.MainActivity"
3. android:label="@string/app\_name"
4. android:screenOrientation="landscape"
5. **>**

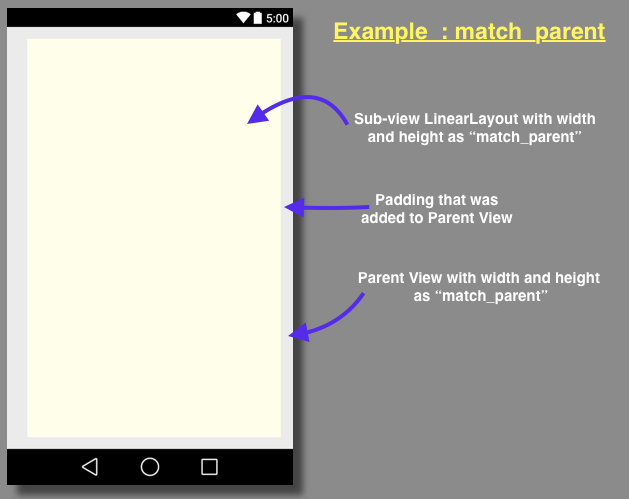
|  |  |
| --- | --- |
| **Value** | **Description** |
| unspecified | It is the default value. In such case, system chooses the orientation. |
| portrait | taller not wider |
| landscape | wider not taller |
| sensor | orientation is determined by the device orientation sensor. |

**# Draw Android activities interface and labeling it**.

#**wrap\_content and match\_parent**

**match\_parent**

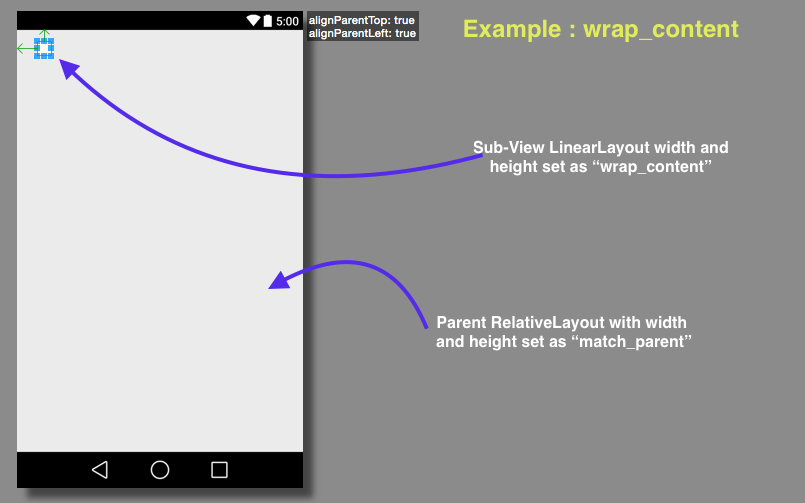
When you set layout width and height as match\_parent, it will occupy the complete area that the parent view has, i.e. it will be as big as the parent.   
  
**Note :** If parent is applied a padding then that space would not be included.



**wrap\_content**

If you set layout width or height as wrap\_content it will use space big enough for its contents to get enclosed.   
  
Lets create a new activity.xml file, with RelativeLayout parent View, now lets add a LinearLayout and set, 

android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"



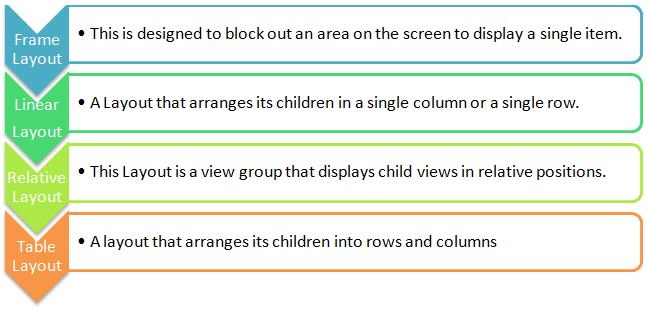
**NOTE:** This is same as match\_parent, fill\_parent was depreciated in API level 8. So if you are using API level 8 or above you must avoid using fill\_parent .

**Java vs Javascript:**

Key differences between **Java** and **JavaScript**: **Java** is an OOP programming language while **Java**Script is an OOP scripting language. **Java** creates applications that run in a virtual machine or browser while **JavaScript** code is run on a browser only. **Java** code needs to be compiled while **JavaScript**code are all in text.

**LinearLayout and RelativeLayout**

**RelativeLayout** : Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent). **LinearLayout** : A layout that organizes its children into a single horizontal or vertical row.



**IDE vs JDK:**

An **integrated development environment** (**IDE**) is a software application that provides comprehensive facilities to computer programmers for software development. An **IDE** normally consists of a source code editor, build automation tools, and a debugger. Most modern **IDEs** have intelligent code completion.

Software Developers: **JDK** (Java SE Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications. Administrators running applications on a server: Server JRE (Server Java Runtime Environment) For deploying Java applications on servers.