Android Presentation Yantra 3.0

Presented By:-

Hem Shrestha
Android Developer @ Picovico

Yubaraj Poudel
Android Developer

Table of Content

Perquisites

Basic Java
(If else, loops, Data Types, Data structures)

Object Oriented Programming
(Class, Object, Inheritance)

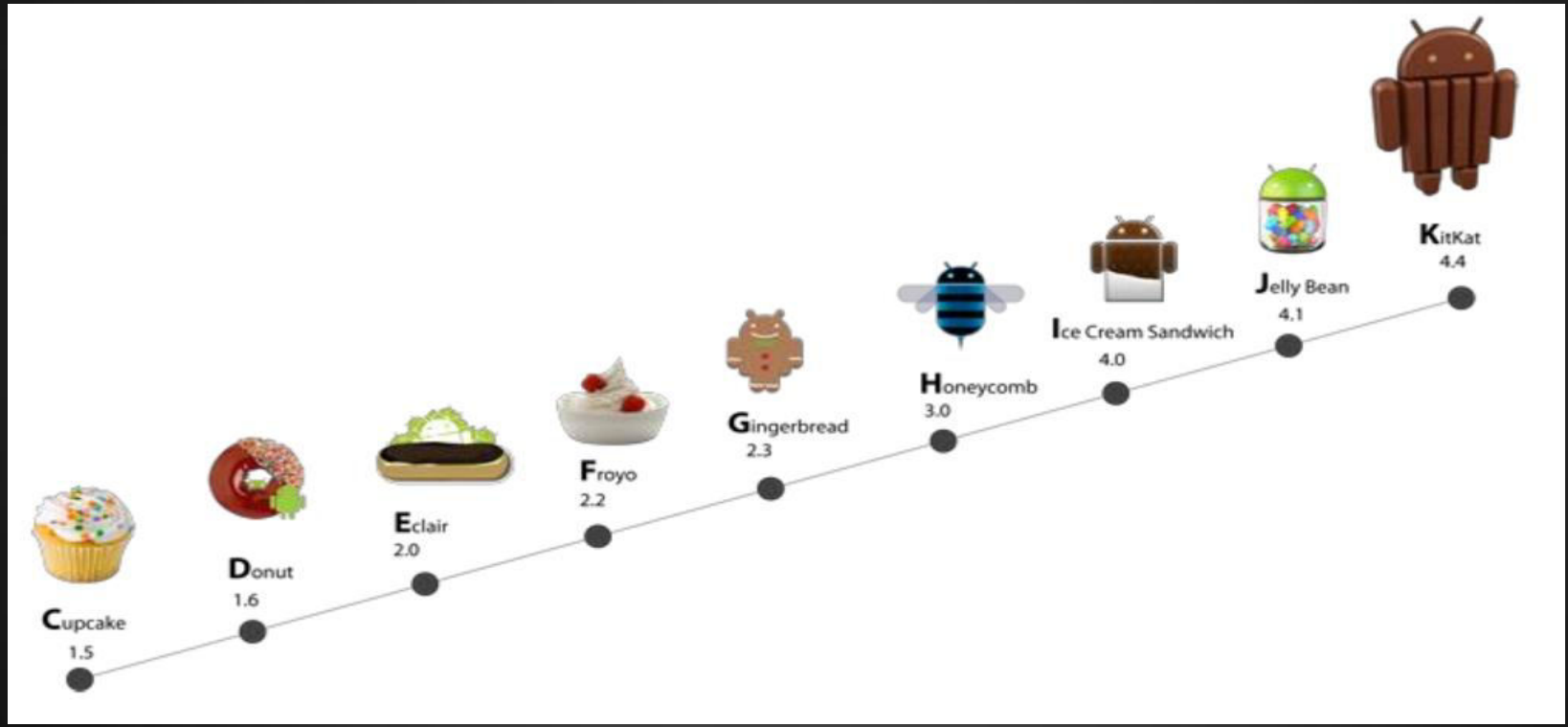
Android Platform Basics

Introduction and history
Android Versions
Android Architecture
Android Virtual Machine
(Dalvik VM, Ark VM is latest in 4.4)

Introduction

A Linux based Operating System designed primarily for touch screen mobile devices
Initially developed by Android Inc, lead by Andy Rubin and later purchased by Google in 2005
Android is Open Source and Google releases code under Apache2 license.

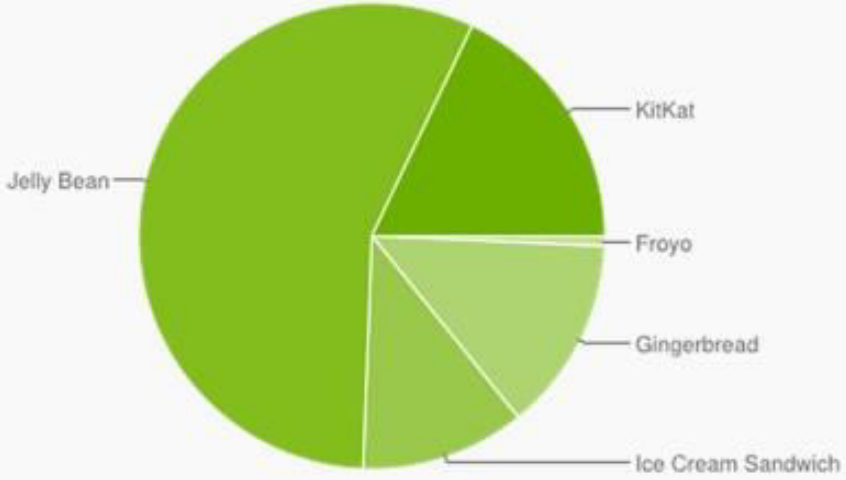
Android Versions



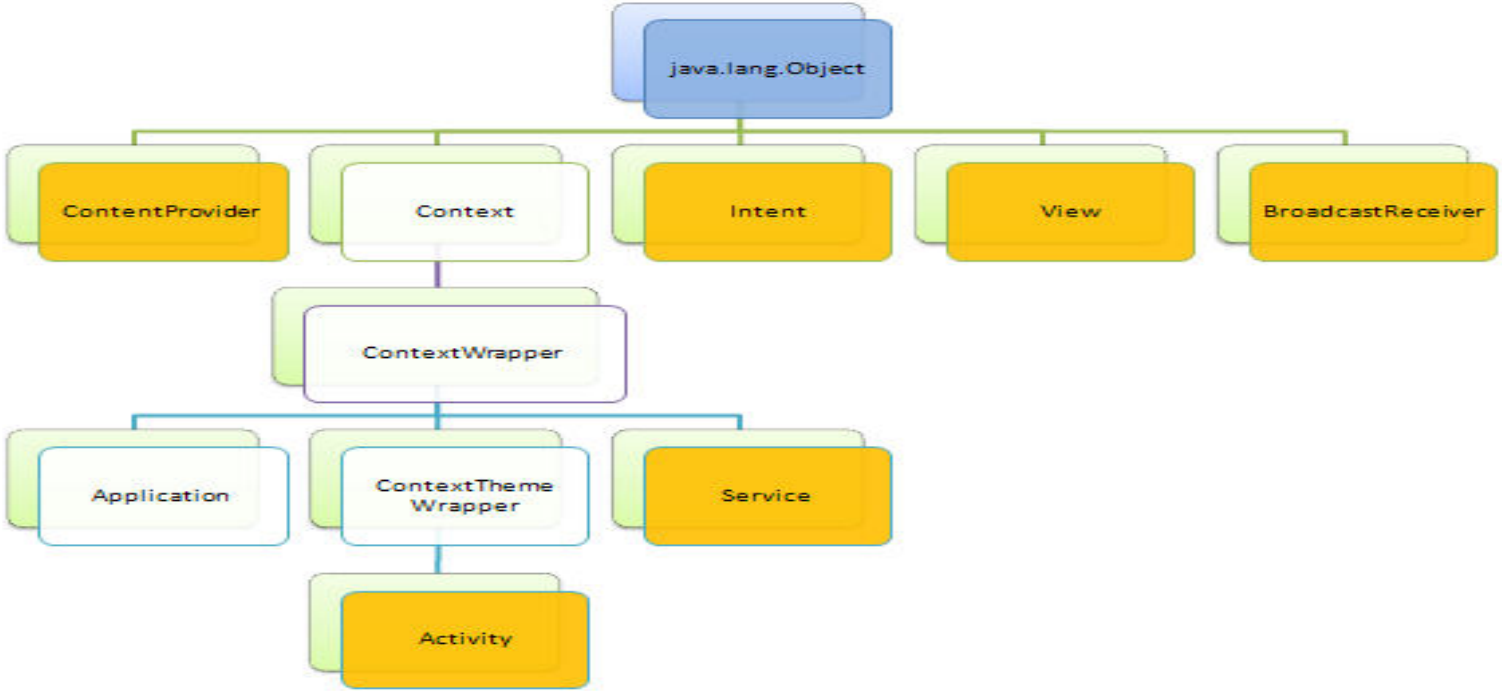
Android Versions Distribution

Version	Codename	API	Distribution
2.2	Froyo	8	0.7%
2.3.3 - 2.3.7	Gingerbread	10	13.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	11.4%
4.1.x	Jelly Bean	16	27.8%
4.2.x		17	19.7%
4.3		18	9.0%
4.4	KitKat	19	17.9%

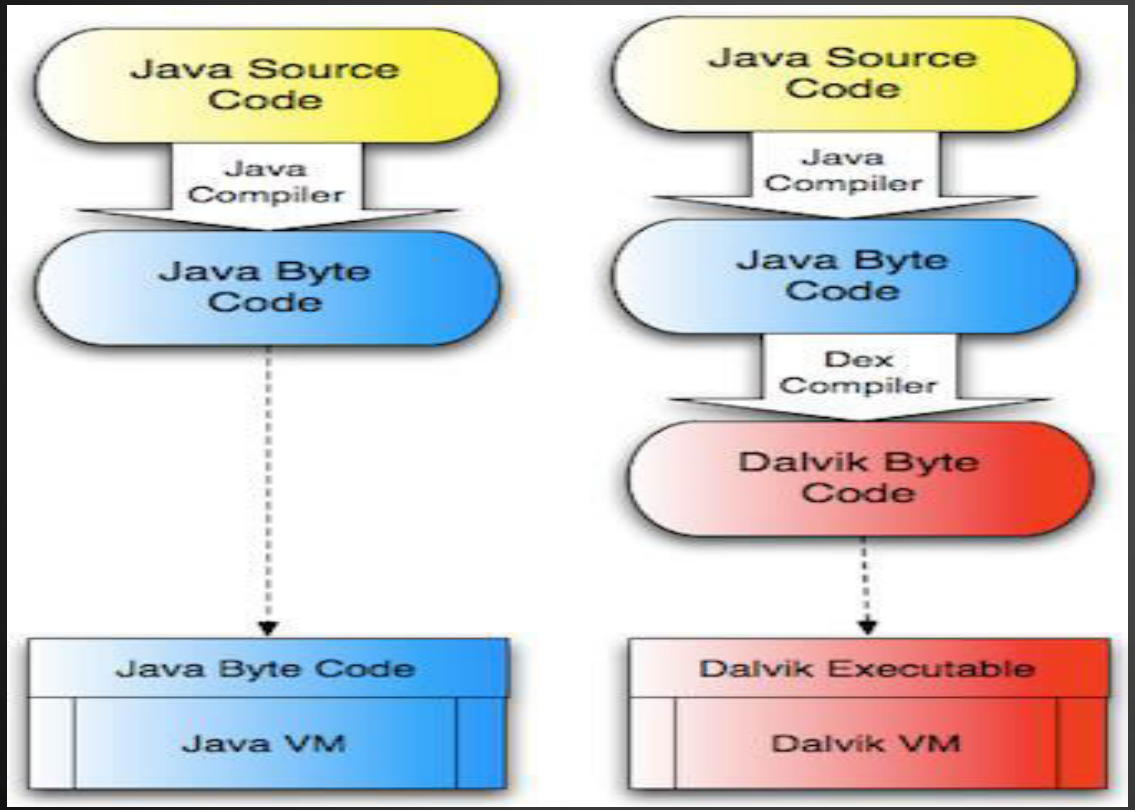
Data collected during a 7-day period ending on July 7, 2014.
Any versions with less than 0.1% distribution are not shown.



Android Architecture



Dalvik Virtual Machine



Android Development Environment

JDK
Eclipse IDE
ADT Plugin
Android SDK
AVD
Debugging with Android LogCat
Eclipse Perspective – Java/DDMS

Hello World !

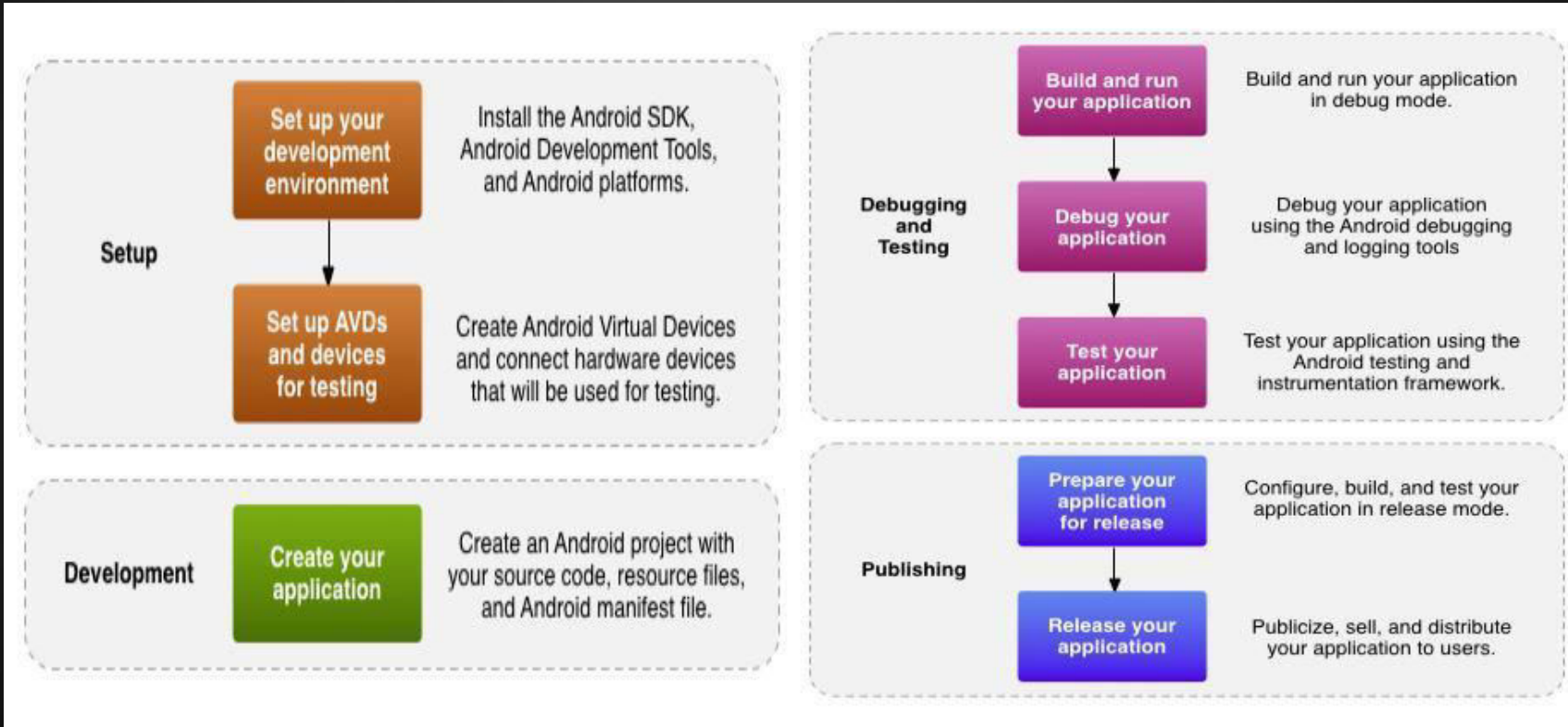
Real Device

Virtual Device

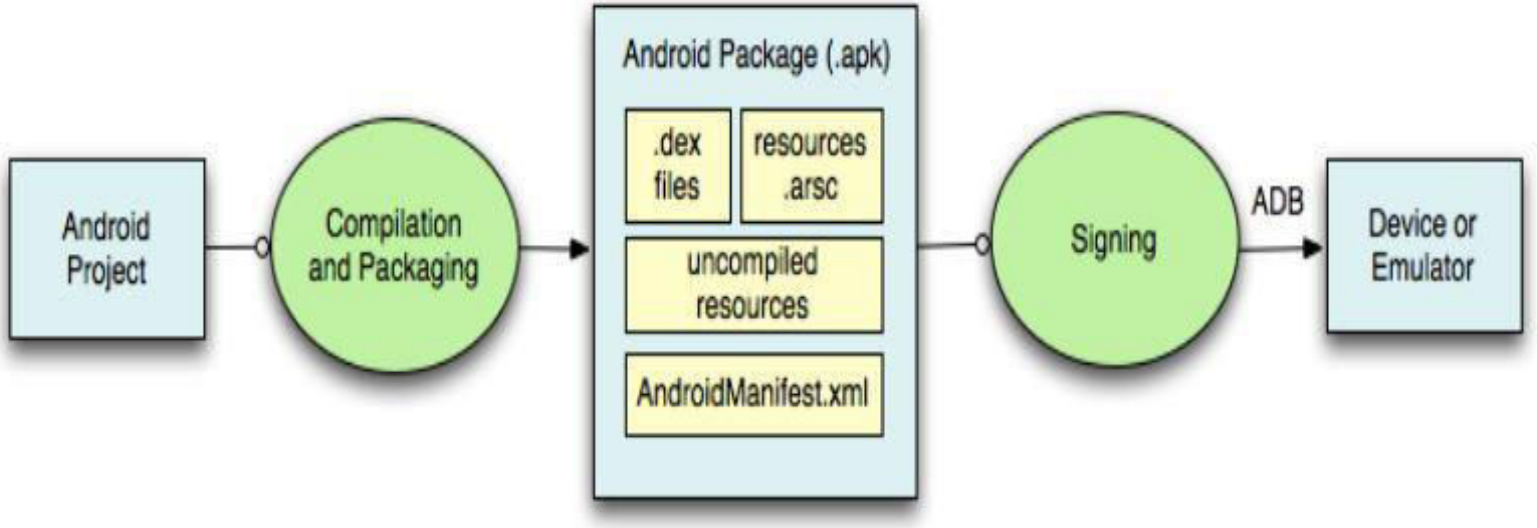
Application Fundamentals

- Apps Development Flow
- Build Cycle
- Directory Structure
- Android Manifest File
- Android Application

Apps Development Flow

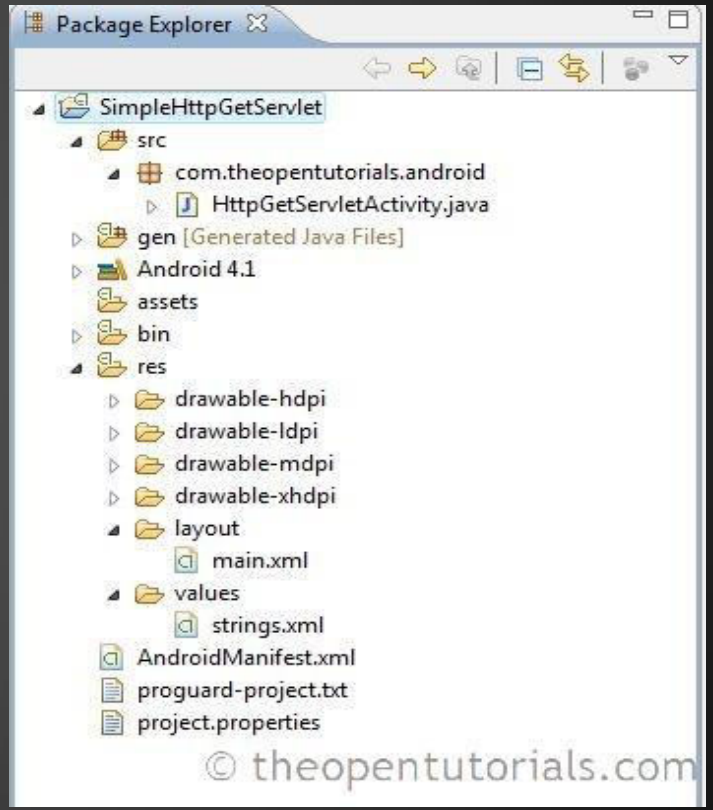


Apps Build Cycle



Directory Structure

src/
bin/ - Output directory of the build. final .apk
gen/ - R.java etc
libs/
assets/
res/ - Contains application resources, such as drawable files, layout files, and string values.
  drawable/
  layout/
  values/
  menu/
  xml/
AndroidManifest.xml



App Manifest File (AndroidManifest.xml)

"Starting point" of app
It describes:

- Package name
- Minimum level Android API
- Application name + icon
- Required Permissions
- The components of the application

(activities, services, broadcast receivers, and content providers)
Describes first activity to launch

- Other stuff

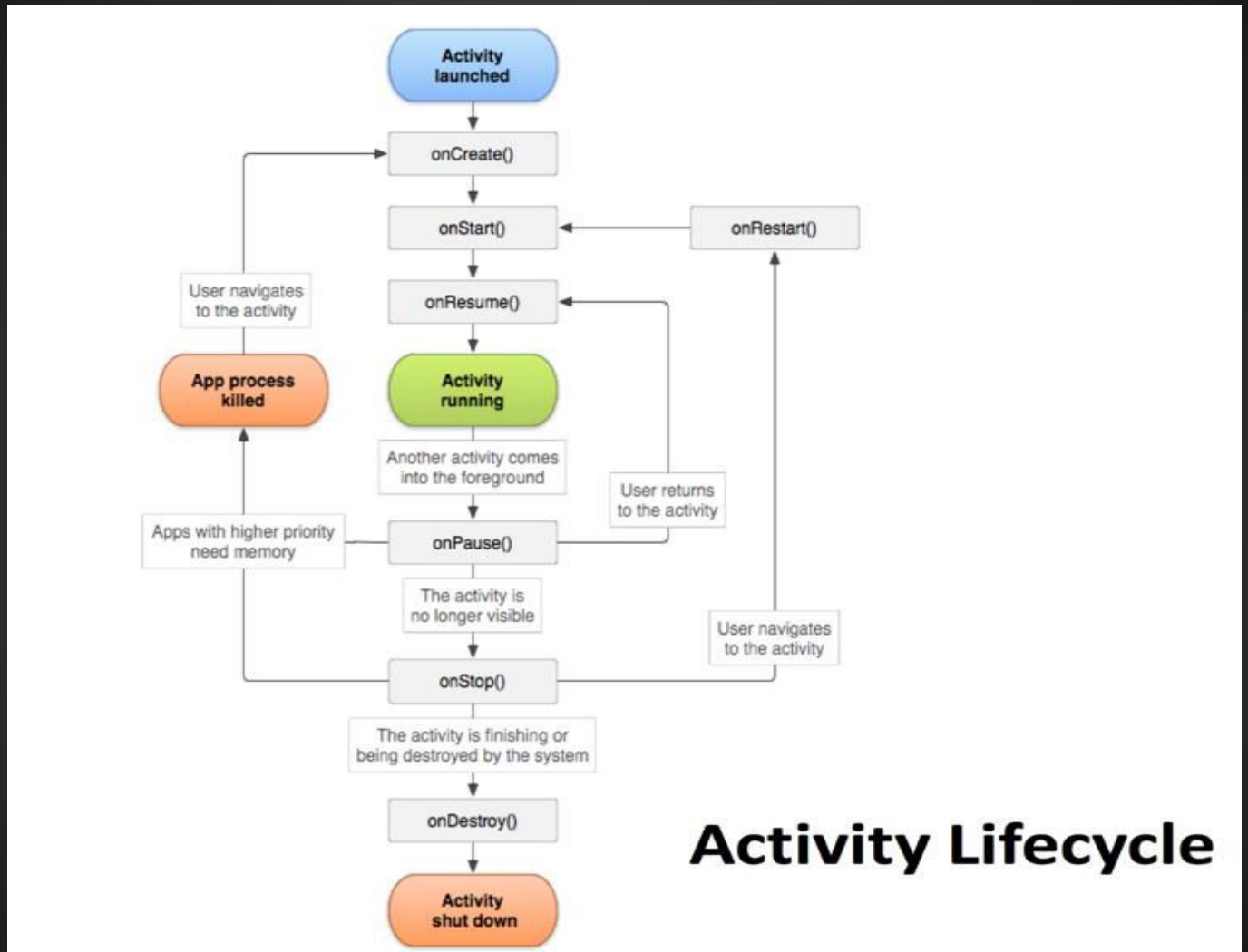
Android Application

Components :

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

Activity


Website – Pages ! App – Screens / Activities
Android Application = ∑ activity



Activity Lifecycle

Android UI - Design

Two ways to construct :

1. xml
2. code

Using xml layout

```
<?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>
```

```
    @Override
    public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
    }
```

Using Java Code

```
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

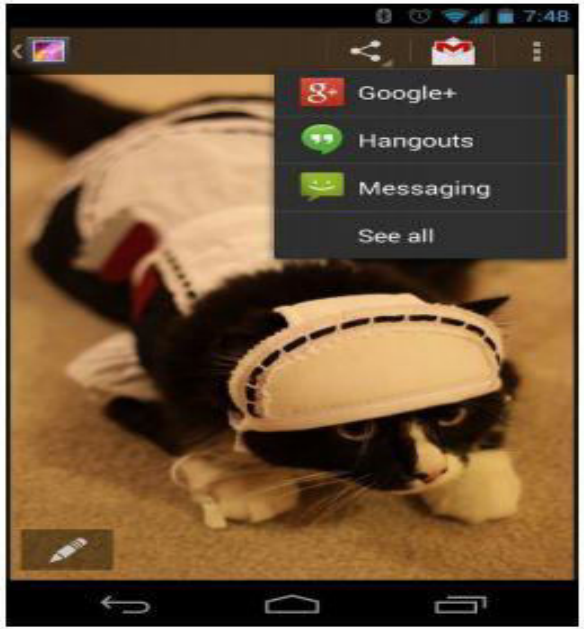
Android UI – Components (Elements)

Android UI – Components (Elements)

Layouts
- Linear Layout
- Relative Layout
- List View
- Grid View

Input Controls
- Buttons
- Text Fields
- Checkboxes
- Radio Buttons
- Toggle Buttons
- Spinners
- Pickers

Input Events
Menus
Action Bar
Settings
Dialogs
Notifications
Toasts
Search
Drag and Drop
Styles and Themes
Custom Components



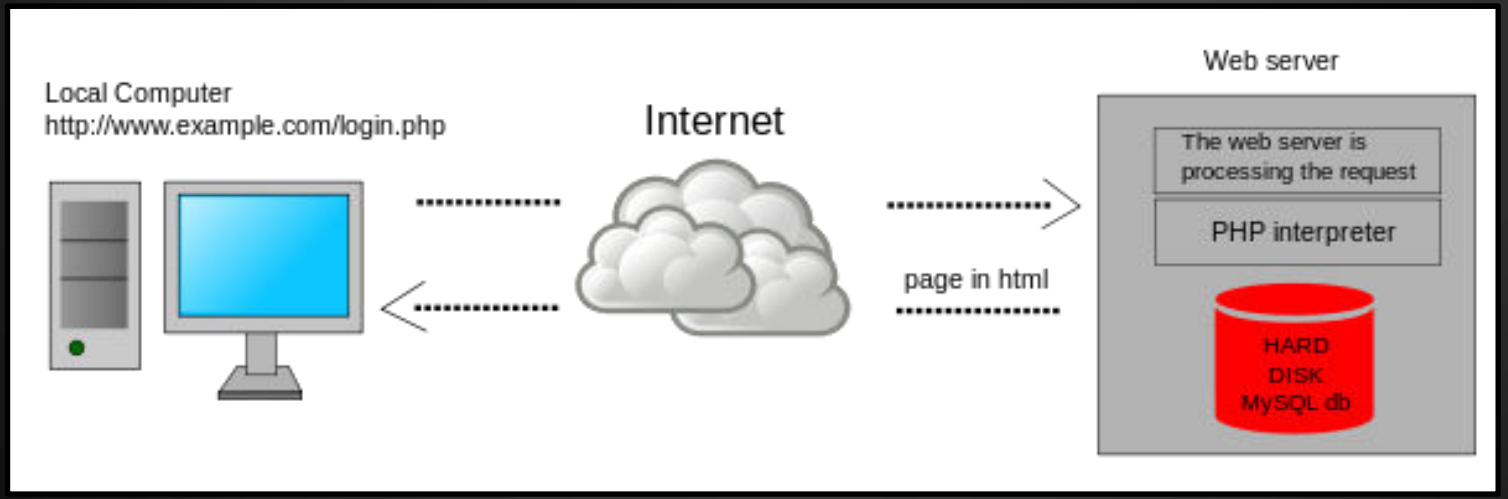
Click Event Handling

```
OnClickListener mClickListener = new OnClickListener() {
@Override
  public void onClick(View v) {
    switch(v.getId()){
      case R.id.btn1:
        // do action1
        break;
      case R.id.btn2:
        // do action2
        break;
    }
  }
};

findViewById(R.id.btn1).setOnClickListener(mClickListener);
findViewById(R.id.btn2).setOnClickListener(mClickListener);
```

Working with Web Services

Concept of API
Data format (JSON)
Android Stuffs



API Concept

- Application Program Interface
- An interface by software components to communicate with each other
- A set of routines, protocols, and tools for building software applications.
- A good API makes it easier to develop a program by providing all the building blocks.

JSON Data

- JSON (JavaScript Object Notation) is a lightweight text-based open standard designed for human-readable data interchange...
- Easy for machines to parse and generate
- Familiar to programmers of the C-family of languages
- JSON Object and JSON Array
- Collection of name/value pairs

- Alternative: XML

JSON Vs. XML

```
{
 "res": "success",
 "data": {
   "id": "1",
   "username": ”hereshem",
   “fname": "Hem",
   “lname": "Shrestha",
   ”c_home": "014488535",
   ”c_mobile": "9843096958",
   ”c_office": "014782582",
   "created": "2012-12-21 05:52:03 pm"
 }
}
```

```
<?xml version="1.0" encoding="UTF-8" ?
 <res>success</res>
 <data>
  <id>1</id>
  <username>hereshem</username>
  <fname>Hem</fname>
  <lname>Shrestha</lname>
  <c_home>014488535</c_home>
  <c_mobile>9843096958</c_mobile>
  <c_office>014782582</c_office>
  <created>2012-12-21 05:52:03 pm</created>
 </data>
```

Android Stuffs

Adding permission
Writing server request file
Check for internet connection
Sync and Async Task
Background and foreground process

Keep on touch

```
public void showMyContact(){
        String username = “hereshem”;  // here‘s Hem

        Log.i(“GitHub”, “github.com/” + username);
        Log.i(“Facebook”, “facebook.com/” + username);
        Log.i(“Twitter”, “twitter.com/” + username);
        Log.i(“Blog”, username + “.blogspot.com”);
        Log.i(“Email”, username + “@gmail.com”);

}
```

Building App

Best of luck for the upcoming events cheers…….