# Android系统架构

Android是一个手机操作系统，系统架构分为五层：

## Linux内核层

包括了Android系统的核心服务，如：硬件驱动、进程管理、安全系统等等

## 运行环境（Dalvik与ART）

Dalvik:

包含了一整套Android运行环境虚拟机，每个APP都会分配Dalvik来保证互不干扰，并保持独立

特点：运行时编译

ART:

Android 5.X开始，ART取代了Dalvik，ART是在安装的时候编译，以后运行就不用再编译了

## Framework层（框架层）

Android APP framework

## Standard libraries

这里面包含了Android中的一些标准库，开发者在开源环境中可以直接使用的库

## 应用层

App

# Android APP 组件架构

Android 四大组件：Activity、BroadCastRecieve、 ContentProvider、service

四大组件组成了Android app的最基本的元素

## Android 四大组件如何协调工作的

1. Activity 最为人机交互的第一界面，负责向用户展示信息和处理信息
2. 这些信息的来源，可以通过资源获取、可以通过ContentProvider来获取其他应用的信息、或是service从后台计算下载处理结果、也可以通过BroadCastRecieve获取广播信息
3. Android提供了一个信使-----Intent，作为信息传递的载体
4. 组件与组件之间通过Intent载体来通信，传递信息、交互数据，正是通过这样的方式，使得各组件之间各自独立而又紧密联系

## 应用运行上下文对象

说明：Android系统中上下文对象，即Context中，为我们封装了这么一个“语境”，activity、service、application都继承自Context

Android应用程序，创建Context对象的几个时机：（创建Context对象的时机，就是创建Context实现类的时候）

1. 创建Application
2. 创建Activity
3. 创建Service

注意：

1. 当应用第一次启动的时候，Android系统会创建一个Application对象
2. 同时，也创建了一个Application Context对象
3. 所有的组件都共同拥有着这么一个Context,这个Context对象，贯穿整个应用程序的生命周期，为应用全局提供功能和环境支持
4. 而创建Activity、Service实例时，系统也会给它们提供运行的上下文对象，在匿名内部类中需要XXX\_ACTIVITY.this指定才能获得该Activity的Context对象
5. 也可以通过getApplicationContext()获取整个应用的Context，这与某个组件的上下文对象，在某些时候还是有区别的

# ADB命令的使用技巧

## Adb工具位于SDK的platform-tools目录下

## Adb命令使用

### adb version

查看adb 版本号

### adb shell

# 控件架构

# 自定义控件详解