# **Android面试题目**

**1.equals与==的区别**

==是判断两个变量或实例是不是指向同一个内存空间 equals是判断两个变量或实例所指向的内存空间的值是不是相

**2.String、StringBuffer与StringBuilder的区别**

String 类型和 StringBuffer 类型的主要性能区别其实在于 String 是不可变的对象 StringBuffer和StringBuilder底层是 char[]数组实现的 StringBuffer是线程安全的，而StringBuilder是线程不安全的

**3.Java的四种引用的区别**

强引用：如果一个对象具有强引用，它就不会被垃圾回收器回收。即使当前内存空间不足，JVM 也不会回收它，而是抛出 OutOfMemoryError 错误，使程序异常终止。如果想中断强引用和某个对象之间的关联，可以显式地将引用赋值为null，这样一来的话，JVM在合适的时间就会回收该对象 。

软引用：在使用软引用时，如果内存的空间足够，软引用就能继续被使用，而不会被垃圾回收器回收，只有在内存不足时，软引用才会被垃圾回收器回收。

弱引用：具有弱引用的对象拥有的生命周期更短暂。因为当 JVM 进行垃圾回收，一旦发现弱引用对象，无论当前内存空间是否充足，都会将弱引用回收。不过由于垃圾回收器是一个优先级较低的线程，所以并不一定能迅速发现弱引用对象。

虚引用：顾名思义，就是形同虚设，如果一个对象仅持有虚引用，那么它相当于没有引用，在任何时候都可能被垃圾回收器回收。

**4.HashMap和 HashTable 的区别**

HashTable比较老，是基于Dictionary 类实现的，HashTable 则是基于 Map接口实现的； HashTable 是线程安全的， HashMap 则是线程不安全的； HashMap可以让你将空值作为一个表的条目的key或value。

**5.wait()和sleep()的区别**

sleep来自Thread类，和wait来自Object类；调用sleep()方法的过程中，线程不会释放对象锁。而 调用 wait 方法线程会释放对象锁； sleep睡眠后不出让系统资源，wait让出系统资源其他线程可以占用CPU； sleep(milliseconds)需要指定一个睡眠时间，时间一到会自动唤醒。

**6.Android 线程间通信有哪几种方式**

共享内存(变量);文件；数据库；Handler； Java 里的 wait()，notify()，notifyAll()

**7.http协议了解多少，说说里面的协议头部有哪些字段?**

http(超文本传输协议)是一个基于请求与响应模式的、无状态的、应用层的协议;http请求由三部分组成，分别是：请求行、消息报头、请求正文。HTTP消息报头包括普通报头、请求报头、响应报头、实体报头

**8.https的了解**

HTTPS(全称：Hyper Text Transfer Protocol over Secure Socket Layer)，是以安全为目标的HTTP通道，简单讲是HTTP的安全版。即HTTP下加入SSL层，HTTPS的安全基础是SSL，因此加密的详细内容就需要SSL。

Https网页里不允许加载Http资源，在PC端会发出警告，可手动通过安全警告来加载http资源，但是移动端是不会给出警告，且默认不加载资源。

**9.Fragment 在你们项目中的使用**

Fragment 是 android3.0 以后引入的的概念，做局部内容更新更方便，原来为了到达这一点要；把多个布局放到一个 activity 里面，现在可以用多 Fragment 来代替，只有在需要的时候才加载。

**10.如何切换 fragement,不重新实例化**

正确的切换方式是 add()，切换时 hide()，add()另一个 Fragment;再次切换时，只需 hide()当前， show()另一个。

**11.Activity和Fragment生命周期有哪些?**

Activity：onCreate->onStart->onResume->onPause->onStop->onDestroy

Fragment：onAttach->onCreate->onCreateView->onActivityCreated->onStart->onResume->onPause->onStop->onDestroyView->onDestroy->onDetach

**12.启动service的两种方法?有什么区别?**

一种是startService(),另一种是bindService()。这两者的区别是第一种方式调用者开启了服务，即会与服务失去联系，两者没有关联。即使访问者退出了，服务仍在运行。如需解除服务必须显式的调用stopService方法。主要用于调用者与服务没有交互的情况下，也就是调用者不需要获取服务里的业务方法。比如电话录音。而后者调用者与服务绑定在一起的。当调用者退出的时候，服务也随之退出。用于需要与服务交互。

**13.Service是否在main thread中执行, service里面是否能执行耗时的操作？**

默认情况,如果没有显示的指 servic 所运行的进程, Service 和 activity 是运行在当前 app 所在进程的 main thread(UI 主线程)里面。service 里面不能执行耗时的操作( 网络请求,拷贝数据库,大文件 )。

特殊情况 ,可以在清单文件配置 service 执行所在的进程 ,让 service 在另外的进程中执行

<service android:name="com.baidu.location.f" android:enabled="true" android:process=":remote" > </service>

**14.为什么要用 ContentProvider ? 它和 sql 的实现上有什么差别?**

ContentProvider 屏蔽了数据存储的细节,内部实现对用户完全透明,用户只需要关心操作数据的uri 就可以了，ContentProvider 可以实现不同 app 之间共享。Sql 也有增删改查的方法，但是 sql 只能查询本应用下的数据库。而 ContentProvider 还可以去增删改查本地文件. xml 文件的读取等。

**15.ListView卡顿的原因与性能优化，越多越好**

1.重用converView： 通过复用converview来减少不必要的view的创建，另外Infalte操作会把xml文件实例化成相应的View实例，属于IO操作，是耗时操作。

2.减少findViewById()操作： 将xml文件中的元素封装成viewholder静态类，通过converview的setTag和getTag方法将view与相应的holder对象绑定在一起，避免不必要的findviewbyid操作

3.避免在 getView 方法中做耗时的操作: 例如加载本地 Image 需要载入内存以及解析 Bitmap ，都是比较耗时的操作，如果用户快速滑动listview，会因为getview逻辑过于复杂耗时而造成滑动卡顿现象。用户滑动时候不要加载图片，待滑动完成再加载，可以使用这个第三方库glide

4.Item的布局层次结构尽量简单，避免布局太深或者不必要的重绘

5.使用 RecycleView 代替listview： 每个item内容的变动，listview都需要去调用notifyDataSetChanged来更新全部的item，太浪费性能了。RecycleView可以实现单个item的局部刷新，并且引入了增加和删除的动态效果，在性能上和定制上都有很大的改善

6.ListView 中元素避免半透明： 半透明绘制需要大量乘法计算，在滑动时不停重绘会造成大量的计算，在比较差的机子上会比较卡。 在设计上能不半透明就不不半透明。实在要弄就把在滑动的时候把半透明设置成不透明，滑动完再重新设置成半透明。

**16.每个应用程序对应多少个Dalvik虚拟机**

每一个Android应用在底层都会对应一个独立的Dalvik虚拟机实例，其代码在虚拟机的解释下得以执行 ，而所有的Android应用的线程都对应一个Linux线程。

**17.横竖屏切换时Activity的生命周期**

此时的生命周期跟清单文件里的配置有关系。不设置Activity的android:configChanges时，切屏会重新调用各个生命周期默认首先销毁当前activity,然后重新加载；设置Activity android:configChanges=”orientation|keyboardHidden|screenSize”时，切屏不会重新调用各个生命周期，只会执行onConfigurationChanged方法。