**常见Android面试题及答案(详细整理)**

1. 请描述一下Activity 生命周期。

答: 如下图所示。共有七个周期函数，按顺序分别是: onCreate(), onStart(), onRestart(), onResume(), onPause(),onStop(), onDestroy()。

onCreate(): 创建Activity时调用，设置在该方法中，还以Bundle的形式提供对以前存储的任何状态的访问。

onStart(): Activity变为在屏幕上对用户可见时调用。

onResume(): Activity开始与用户交互时调用(无论是启动还是重新启动一个活动，该方法总是被调用。

onPause(): Activity被暂停或收回cpu和其他资源时调用，该方法用户保护活动状态的，也是保护现场。

onStop(): Activity被停止并转为不可见阶段及后续的生命周期事件时调用。

onRestart(): Activity被重新启动时调用。该活动仍然在栈中，而不是启动新的Activity。

1、完整生命周期: 即从一个Activity从出现到消失，对应的周期方法是从onCreate()到onDestroy()。

2、可见生命周期: 当Activity处于可以用户看见的状态，但不一定能与用户交互时，将多次执行从onStart()到onStop()。

3、前景生命周期: 当Activity处于Activity栈最顶端，能够与其他用户进行交互时，将多次执行从onResume()到onPause()。

2. 两个Activity之间跳转时必然会执行的是哪几个方法。

答: 两个Activity之间跳转必然会执行的是下面几个方法。

onCreate()//在Activity生命周期开始时调用。

onRestoreInstanceState()//用来恢复UI状态。

onRestart()//当Activity重新启动时调用。

onStart()//当Activity对用户即将可见时调用。

onResume()//当Activity与用户交互时，绘制界面。

onSaveInstanceState()//即将移出栈顶保留UI状态时调用。

onPause()//暂停当前活动Activity，提交持久数据的改变，停止动画或其他占用GPU资源的东西，由于下一个Activity在这个方法返回之前不会resume，所以这个方法的[代码](http://www.xuebuyuan.com/" \o "代码" \t "_blank)执行要快。

onStop()//Activity不再可见时调用。

onDestroy()//Activity销毁栈时被调用的最后一个方法。

3. 横竖屏切换时候Activity的生命周期。

答: 1、不设置Activity的android: configChanges时，切屏会重新调用各个生命周期，切横屏时会执行一次，切竖屏时会执行两次。

2、设置Activity的android: configChanges=“orientation”时，切屏还是会重新调用各个生命周期，切横、竖屏时只会执行一次。

3、设置Activity的android: configChanges=“orientation|keyboardHidden”时，切屏不会重新调用各个生命周期，只会执行onConfiguration方法

4. 如何将一个Activity设置成窗口的样式。

答: 第一种方法，在styles.xml文件中，可以新建如下的类似Dialog的style。

<style name=“Theme.FloatActivity” parent=“android:style/Theme.Dialog”> </style>。

第二种方法，在AndroidManifest.xml中在需要显示为窗口的Activity中添加如下属性:  android: theme=“@style/Theme.FloatActivity”即可。也可以直接添加对应需要展示为Dialog style的Activity的android: theme属性为android: theme=“@ android: style/Theme.Dialog”。

5. 两个Activity之间怎么传递数据？

答: 可以在Intent对象中利用Extra来传递存储数据。

在Intent的对象请求中，使用putExtra(“键值对的名字”，”键值对的值”)；在另外一个Activity中将Intent中的请求数据取出来:

Intent intent = getIntent();

String value = intent.getStringExtra(“testIntent”);

6. 怎么让在启动一个Activity是就启动一个service？

答: 首先定义好一个service，然后在Activity的onCreate里面进行连接并bindservice或者直接startService。

7. Activity怎么和service绑定，怎么在activity中启动自己对应的service？

答: 1、activity能进行绑定得益于Serviece的接口。为了支持Service的绑定，实现onBind方法。

2、Service和Activity的连接可以用ServiceConnection来实现。需要实现一个新的ServiceConnection，重现onServiceConnected和OnServiceDisconnected方法，一旦连接建立，就能得到Service实例的引用。

3、执行绑定，调用bindService方法，传入一个选择了要绑定的Service的Intent(显示或隐式)和一个你实现了的ServiceConnection的实例

8.什么是Service以及描述下它的生命周期。Service有哪些启动方法，有什么区别，怎样停用Service？

答: Android Service是运行在后台的代码，不能与用户交互，可以运行在自己的进程，也可以运行在其他应用[程序](http://www.xuebuyuan.com/)进程的上下文里。需要通过某一个Activity或者Context对象来调用。Service有两个启动方法，分别是Context.startService()和Context.bindService()。如果在Service执行耗时的操作需要启动一个新线程来执行。

Android Service只继承了onCreate(), onStart(),onDestroy()三个方法，当我们第一次启动Service时，先后调用onCreate(), onStart()这两个方法，当停止Service时，则执行onDestroy()方法时。如果Service已经启动了，当我们再次启动Service时，不会再执行onCreate()方法，而是直接执行onStart()方法。

9. 什么时候使用Service？

答: 比如播放多媒体的时候，用户启动了其他Activity，这个时候程序要在后台继续播放，比如检测SD卡上文件的变化，再或者在后台记录你的地理信息位置的改变等等。

10. 请描述一下Intent 和 Intent Filter。

答: Intent在Android中被翻译为”意图”，他是三种应用程序基本组件-Activity，Service和broadcast receiver之间相互激活的手段。在调用Intent名称时使用ComponentName也就是类的全名时为显示调用。这种方式一般用于应用程序的内部调用，因为你不一定会知道别人写的类的全名。而Intent Filter是指意图过滤，不出现在代码中，而是出现在android Manifest文件中，以<intent-filter>的形式。（有一个例外是broadcast receiver的intent  
filter是使用Context.registerReceiver()来动态设定的，其中intent filter也是在代码中创建的）

一个intent有action，data，category等字段。一个隐式intent为了能够被某个intent filter接收，必须通过3个测试，一个intent为了被某个组件接收，则必须通过它所有的intent filter中的一个。

11. Intent传递数据时，可以传递哪些类型数据？

答: intent间传送数据一般有两种常用的方法: 1、extra 2、data。

extra可以用Intent.putExtra放入数据。新启动的Activity可用Intent.getExtras取出Bundle，然后用Bundles.getLong,getInt,getBoolean,getString等函数来取放进去的值。

Data则是传输url。url可以是指我们熟悉的http，ftp等网络地址，也可以指content来指向ContentProvider提供的资源。Intent.setData可以放入数据，Intent.getData可以取出数据。

12. 说说Activity，Intent，Service是什么关系 ？

答: 一个Activity通常是一个单独的屏幕，每一个Activity都被实现为一个单独的类，这些类都是从Activity基类中继承而来的。Activity类会显示由视图控件组成的用户接口，并对视图控件的事件做出响应。

Intent的调用是用来进行屏幕之间的切换。Intent描述应用想要做什么。Intent数据结构中两个最重要的部分是动作和动作对应的数据，一个动作对应一个动作数据。

Service是运行在后台的代码，不能与用户交互，可以运行在自己的进程里，也可以运行在其他应用程序进程的上下文里。需要一个Activity或者其他Context对象来调用。

Activity跳转Activity，Activity启动Service，Service打开Activity都需要Intent表明意图，以及传递参数，Intent是这些组件间信号传递的承载着。

13. 请描述一下BroadcastReceiver。

答: Broadcast Receiver用于接收并处理广播通知(broadcast announcements)。多数的广播是系统发起的，如地域变换、电量不足、来电短信等。程序也可以播放一个广播。程序可以有任意数量的broadcast receivers来响应它觉得重要的通知。Broadcast receiver可以通过多种方式通知用户: 启动activity、使用NotificationManager、开启背景灯、振动设备、播放声音等，最典型的是在状态栏显示一个图标，这样用户就可以点它打开看通知内容。通常我们的某个应用或系统本身在某些事件(电池电量不足、来电短信)来临时会广播一个Intent出去，我们利用注册一个broadcast  
receiver来监听这些Intent并获取Intent中的数据。

14. 在manifest和代码中如何注册和使用 broadcast receiver 。

答: 在android的manifest中注册

<receiver android: name =“Receiver1”>

                <intent-filter>

                            <!----和Intent中的action对应--->

                            <actionandroid: name=“com.forrest.action.mybroadcast”/>

                </intent-filter>

</receiver>

在代码中注册

1、 IntentFilter filter = new IntentFilter(“com.forrest.action.mybroadcast”);//和广播中Intent的action对应;

2、 MyBroadcastReceiver br= new MyBroadcastReceiver();

3、 registerReceiver(br, filter);

15. 请介绍下ContentProvider是如何实现数据共享的。

答: 一个程序可以通过实现一个Content provider的抽象接口将自己的数据完全暴露出去，而且Content provider是以类似数据库中的表的方式将自己的数据暴露。Content provider存储和检索数据，通过它可以让所有的应用程序访问到，这也是应用程序之间唯一共享数据的方法。

要想使应用程序的数据公开化，可通过2种方法：创建一个数据自己的Content Provider或者将你的数据添加到一个已经存在的Content Provider中，前提是有相同数据类型并且有写入Content Provider的权限，Android提供了Content Resolverr，外界的程序可以通过Content Resolver接口访问Content Provider提供的数据。

16. 请介绍下Android的数据存储方式。

答: Android提供了5中存储数据的方式，分别是以下几种

1、使用Shared Preferences存储数据，用来存储key-value，pairs格式的数据，它是一个轻量级的键值存储机制，只可以存储基本数据类型。

2、使用文件存储数据，通过FileInputStream和FileOutputStream对文件进行操作。在Android中，文件是一个应用程序私有的，一个应用程序无法读写其他应用程序的文件。

3、使用SQLite数据库存储数据，Android提供的一个标准数据库，支持SQL语句。

4、使用Content Provider存储数据，是所有应用程序之间数据存储和检索的一个桥梁，它的作用就是使得各个应用程序之间实现数据共享。它是一个特殊的存储数据的类型，它提供了一套标准的接口用来获取数据，操作数据。系统也提供了音频、视频、图像和个人信息等几个常用的Content Provider。如果你想公开自己的私有数据，可以创建自己的Content Provider类，或者当你对这些数据拥有控制写入的权限时，将这些数据添加到Content Provider中实现共享。外部访问通过Content Resolver去访问并操作这些被暴露的数据。

5、使用网络存储数据

17. 请介绍下Android中常用的五种布局。

答: 最常用的布局方式为Absolute Layout、Relative Layout、Linear Layout、FrameLayout、TableLayout。其中Linear Layout和Relative Layout是最常用的方式，他们可以通过在xml配置文件或者代码中进行布局。

1、Frame Layout是最简单的布局方式，放置的控件都只能罗列到左上角，控件会有重叠，不能进行复杂的布局。

2、Linear Layout可以通过orientation属性设置线性排列的方向是垂直还是纵向的,每行或每列只有一个元素，可以进行复杂的布局。

3、Absolute Layout可以让子元素指定准确的x、y坐标值，并显示在屏幕上。Absolute Layout没有页边框，允许元素之间相互重叠。它是绝对坐标，所以在实际中不提倡使用。

4、Relative Layout允许子元素制定他们相对于其他元素或父元素的位置(通过ID制定)。因此，你可以以右对齐，或上下，或置于屏幕中央的形式来排列两个元素。元素按顺序排列，因此如果第一个元素在屏幕的中央，那么相对于这个元素的其他元素将以屏幕中央的相对位置来排列。这个是相对于Absolute Layout的，采用相对坐标，所以在实际中比较常用。

5、Table Layout将以子元素的位置分配到行或列。一个Table Layout由许多的Table Row组成，每个Table Row都会定义一个row。Table Layout容器不会显示row、column或者cell的边线框。每个row拥有0个或多个的cell； 和html中的table差不多。在实际中也经常使用。

18. 谈谈UI中， Padding和Margin有什么区别？

答: Padding是控件的内容相对控件的边缘的边距，而Margin是控件边缘相对于其他控件的边距。如下图所示：

19.  android本身的一些限制，比如apk包大小限制，读取大文件时的时间限。

答：apk包大小限制不好说，有的apk为100M，还是能装到手机上。一般的apk大小为5~10M左右。读取大文件的时间应该是在main线程里面，时间限制为5秒左右。

20. ListView如何提高其效率？

答:1、使用分页加载，不要一次性加载所有数据。

2、复用convertView。在getItemView中，判断converView是否为空，如果不为空，可复用。

3、异步加载图片。Item中如果包含有webimage，那么最好异步加载。

4、快速滑动时，不显示图片。当快速滑动列表(SCROLL\_STATE\_FLING)，item中的图片或获取需要消耗资源的view，可以不显示出来；而处于其他两种状态(SCROLL\_STATE\_IDLE和SCROLL\_STATE\_TOUCH\_SCROLL)，则将那些view显示出来

21. Android程序与Java程序的区别？

|  |  |  |
| --- | --- | --- |
| 区别 | Java程序 | Android程序 |
| 执行入口点 | Main | Android Mainfest.xml配置文件中的intent-filter |
| UI | Frame· | Activity |
| 是否基于配置文件 | 否 | 是 |
| 是否基于组件 | 否 | 是Activity，Service，Broadcast Receiver，Content Provider |

22. 谈谈对Android NDK的理解。

答: android NDK是一套工具，允许Android应用开发者嵌入从C、C++[源代码](http://www.xuebuyuan.com/)编译来的本地机器代码到各自的应用软件包中。

1、 NDK是一系列工具的集合。

NDK提供了一系列的工具，帮助开发者快速开发C（或C++）的动态库，并能自动将so和java应用一起打包成apk。这些工具对开发者帮助时巨大的。

NDK集成了交叉编辑器，并提供了相应的mk文件隔离CPU、平台、API等差异，开发人员只需要简单修改mk文件（指出“那些文件需要编译”、“编译特性要求”等），就可以创建出so。NDK可以自动将so和Java应用一起打包，极大的减轻了开发人员的打包工作。

2、NDK提供了一份稳定、功能有限的API头文件声明。这些API支持的功能非常有限，包含有：C标准库（libc）、标准数学库（libm）、压缩库（libz）、log库（liblog）。

23. 谈谈Android的优点和不足之处。

答: 优点如下所示：

1、android是开源、市场占有率大。

2、android能够有丰富的硬件选择。

3、android是一款基于Linux平台的开源操作系统，从而避免了专利壁垒，是一款完全免费的只能手机平台。

4、由于android是google开发的，因此能够无缝结合Google应用。

缺点如下所示：

1、安全与隐私。手机与互联网紧密联系，个人隐私很难得到保守。

24. Android系统中GC什么情况下会出现内存泄露呢？

答: 导致内存泄露主要的原因是，先前申请了内存空间而忘记了释放。如果程序中存在无用对象的引用，那么这些对象就会驻留内存，消耗内存，因为无法让垃圾回收器GC验证这些对象是否不再需要。如果存在对象的引用，这个对象就被定义为“有效的活动”，同时不会被释放。要确定对象所占内存将被回收，我们就要确认该对象不会再被使用。典型的做法是把对象数据成员设为null或者从集合中移除该对象。当出现以下情况时，会造成内存泄露:

1、 数据库的cursor没有关闭。

2、 构造adapter时，没有使用缓存contentview。

3、 Bitmap对象不使用时，采用recycle()释放内存。

4、 Activity中的对象的生命周期大于activity。

调试方法: DDMS==>HEAPSIZE==>dataobject==>[TotalSize]

25. Android UI中的View如何刷新。

答: Android中对View的更新方式有很多种，使用时要区分不同的应用场合。要分清的是：多线程和双缓冲。

1、不使用多线程和双缓冲

这种情况最简单，一般只希望View在发生改变时对UI进行重绘。你只需要Activity中显式调用View对象中的invalidate()方法即可。系统会自动调用View的onDraw()方法。

2、使用多线程和不使用双缓冲

这种情况下需要开启新的线程，新开的线程就不好访问View对象了。强行访问的话会报错：android.view.ViewRoot$ CalledFromWrongThreadException: only theoriginal thread that created a view hierarchy can touch its views。

这时候你需要创建一个继承了android.os.handler的子类，并重写handleMessage方法。Android.os.Handle是能发送和处理消息的，你需要在Activity中发出更新UI的消息，然后再你的Handler(可以使用匿名内部类)中处理消息(因为匿名内部类可以访问父类变量，你可以直接调用View对象中的invalidate()方法。也就是说：在新线程中创建并发送一个Message，然后在主线程中捕获、处理该消息。

3、使用多线程和双缓冲

Android的SurfaceView是View的子类，她同时也实现了双缓冲。你可以定义一个她的子类并实现Surfaceholder.Callback接口。由于SurfaceHolder.Callback接口，新线程就不要android.os.Handler帮忙了。SurfaceHolder中lockCanvas()方法可以锁定画布，绘制完新的图像后调用unlockCanvasand Post解锁。