作者：漫步  
链接：https://www.zhihu.com/question/27227425/answer/35973793  
来源：知乎  
著作权归作者所有，转载请联系作者获得授权。

受一个朋友邀请来回答这问题，以上各位都回答得很不错，不过对于一个Android开发新手来说，能有更详细的答案那可能会更有帮助。以下内容是来自eoe的一个网友的回答，但现在已经找不回那个帖子，文字已经珍藏起来。感谢原著作者的。

**一、Android编码规范**  
1.java代码中不出现中文，最多注释中可以出现中文；  
2.局部变量命名、静态成员变量命名：只能包含字母，单词首字母出第一个都为大写，其他字母都为小写；  
3.常量命名：只能包含字母和\_，字母全部大写，单词之间用\_隔开；  
4.layout中的id命名：命名模式为：view缩写\_模块名称\_view的逻辑名称  
view的缩写详情如下  
LinearLayout：ll  
RelativeLayout:rl  
TextView:tv  
ImageView:iv  
ImageButton:ib  
Button:btn  
5.activity中的view变量命名  
命名模式为：view缩写+逻辑名称  
建议：如果layout文件很复杂，建议将layout分成多个模块，每个模块定义一个moduleViewHolder，其成员变量包含所属view  
6.strings.xml中的id命名：

命名模式：activity名称\_功能模块名称\_逻辑名称 activity名称\_逻辑名称 common\_逻辑名称  
strings.xml中，使用activity名称注释，将文件内容区分开来  
7.drawable中的图片命名  
命名模式：activity名称\_逻辑名称/common\_逻辑名称  
7.styles.xml：将layout中不断重现的style提炼出通用的style通用组件，放到styles.xml中；  
8.使用layer-list和selector  
9.图片尽量分拆成多个可重用的图片  
10.服务端可以实现的，就不要放在客户端  
11.引用第三方库要慎重，避免应用大容量的第三方库，导致客户端包非常大  
12.处理应用全局异常和错误，将错误以邮件的形式发送给服务端  
13.图片的.9处理  
14.使用静态变量方式实现界面间共享要慎重  
15.Log(系统名称 模块名称 接口名称，详细描述)  
16.单元测试（逻辑测试、界面测试）  
17.不要重用父类的handler，对应一个类的handler也不应该让其子类用到，否则会导致message.what冲突  
18.activity中在一个View.OnClickListener中处理所有的逻辑  
19.strings.xml中使用%1$s实现字符串的通配  
20.如果多个Activity中包含共同的UI处理，那么可以提炼一个CommonActivity，把通用部分叫由它来处理，其他activity只要继承它即可  
21.使用button+activitgroup实现tab效果时，使用Button.setSelected(true)，确保按钮处于选择状态，并使activitygroup的当前activity与该button对应  
22.如果所开发的为通用组件，为避免冲突，将drawable/layout/menu/values目录下的文件名增加前缀  
23.数据一定要效验，例如  
字符型转数字型，如果转换失败一定要有缺省值；  
服务端响应数据是否有效判断；

**二、Android性能优化**  
1.http用gzip压缩，设置连接超时时间和响应超时时间  
http请求按照业务需求，分为是否可以缓存和不可缓存，那么在无网络的环境中，仍然通过缓存的httpresponse浏览部分数据，实现离线阅读。  
2.listview 性能优化  
1).复用convertView  
在getItemView中，判断convertView是否为空，如果不为空，可复用。如果couvertview中的view需要添加listerner，代码一定要在if(convertView==null){}之外。  
2).异步加载图片  
item中如果包含有webimage，那么最好异步加载  
3).快速滑动时不显示图片  
当快速滑动列表时（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  
3.使用线程池，分为核心线程池和普通线程池，下载图片等耗时任务放置在普通线程池，避免耗时任务阻塞线程池后，导致所有异步任务都必须等待  
4.异步任务，分为核心任务和普通任务，只有核心任务中出现的系统级错误才会报错，异步任务的ui操作需要判断原activity是否处于激活状态  
5.尽量避免static成员变量引用资源耗费过多的实例,比如Context  
6.使用WeakReference代替强引用，弱引用可以让您保持对对象的引用，同时允许GC在必要时释放对象，回收内存。对于那些创建便宜但耗费大量内存的对象，即希望保持该对象，又要在应用程序需要时使用，同时希望GC必要时回收时，可以考虑使用弱引用。  
7.超级大胖子Bitmap  
及时的销毁(Activity的onDestroy时，将bitmap回收)  
设置一定的采样率  
巧妙的运用软引用  
drawable对应resid的资源，bitmap对应其他资源8.保证Cursor 占用的内存被及时的释放掉，而不是等待GC来处理。并且 Android明显是倾向于编程者手动的将Cursor close掉  
9.线程也是造成内存泄露的一个重要的源头。线程产生内存泄露的主要原因在于线程生命周期的不可控  
10.如果ImageView的图片是来自网络，进行异步加载  
11.应用开发中自定义View的时候，交互部分，千万不要写成线程不断刷新界面显示，而是根据TouchListener事件主动触发界面的更新

**三、Android UI优化**  
1.layout组件化，尽量使用merge及include复用  
2.使用styles，复用样式定义  
3.软键盘的弹出控制，不要让其覆盖输入框  
4.数字、字母和汉字混排占位问题：将数字和字母全角化。由于现在大多数情况下我们的输入都是半角，所以 字母和数字的占位无法确定，但是一旦全角化之后，数字、字母的占位就和一个汉字的占位相同了，这样就可以避免由于占位导致的排版问题。  
5.英文文档排版：textview自动换行时要保持单词的完整性，解决方案是计算字符串长度，然后手动设定每一行显示多少个字母并加上‘n‘  
6.复杂布局使用RelativeLayout  
7.自适应屏幕，使用dp替代pix  
8.使用android:layout\_weight或者TableLayout制作等分布局  
9.使用animation-list制作动画效果

**四、其他的一些Android开发建议**

1.跟上时代的步伐，把Eclipse换成Android Studio，把SVN换成Git，这当然要适合项目开发属性的需要，Git学习中文网站：[https://git-scm.com/book/zh/v2](//link.zhihu.com/?target=https%3A//git-scm.com/book/zh/v2)

2.勤做总结，推荐使用印象笔记，把一些懂的经验总结起来，把还不懂的文章挂里面，有时间就慢慢消化；

3.定期code review，不断迭代，你总会发现一些不合理的代码，或者需要优化的地方。

4.关注一些知名的技术大V或网站，里面许多东西值得你去消化，推荐：[Android 开源项目集合](//link.zhihu.com/?target=http%3A//www.codekk.com/)，[http://tech.meituan.com/](//link.zhihu.com/?target=http%3A//tech.meituan.com/)，[stormzhang](https://www.zhihu.com/people/stormzhang)，但总归来说，去Android官网或者参考一份Java API文档虽说枯燥，但熟悉之后，你会有更大的进步。

5.如想更深入了解可阅读珍藏许久的文章：[Android应用程序开发以及背后的设计思想深度剖析](//link.zhihu.com/?target=http%3A//www.uml.org.cn/mobiledev/201211063.asp%231)

6.如果你公司没有强大的测试团队，发布应用前最好把应用放到测试平台去测测，比如云测之类的；

7.取应用包名的时候切忌取太容易重复的，如果同款手机已经有该包名，那么会因为签名不同而导致安装不上，这也怪中国安卓市场太多，无法像Google Play那样进行包名审核。