# Android开发规范

1. 代码格式参考Google的规范和样例
   1. Android官方代码规范

<http://source.android.com/source/code-style.html>

* 1. Java代码规范

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

中文版本见《Code Conventions for Java》

1. 变量命名要求遵循见名知义的原则，如果你的命名别人看不懂，或者过了2周自己也不知道这个变量是做什么的，那这个命名就失败鸟
2. 大量的命名，要会使用前缀和后缀的方式，会使命名看起来很统一

例如：

//订单状态

typedef NS\_ENUM(NSInteger, GHSOrderStatus)

{

GHSOrderStatusCanceled = 0, //已取消

GHSOrderStatusPendingApproval = 9, //待审核

GHSOrderStatusProcessing = 10, //正在处理

GHSOrderStatusPendingPayment = 11, //待付款

GHSOrderStatusPendingDelivery = 20,//待发货

GHSOrderStatusDelivering = 30, //已发货

GHSOrderStatusReceived = 39, //已签收

GHSOrderStatusFinished = 40, //交易成功

GHSOrderStatusRejection = 63, //拒收

GHSOrderStatusProcessing1 = 198, //正在处理

GHSOrderStatusProcessing2 = 199 //正在处理

};

例如：

home\_logo.png

home\_item\_line.png

home\_icon\_search.png

home\_icon\_cancel.png

tab\_item\_0.png

tab\_item\_0\_h.png

tab\_item\_1.png

tab\_item\_1\_h.png

1. 养成注释的好习惯，在业务逻辑复杂的地方，或者变量需要解释的地方，或者暴露给别人看的接口的地方，等等，要添加注释

如果通过变量和命名已经很清晰业务逻辑，注释就是画蛇添足，例如：

if ( firstLaunch )

{

gotoRegister();

}

1. 养成使用常量的好习惯

如：

GHSPayTypeCashOnDelivery = 6, //货到付款

GHSPayTypeAlipay = 7 //支付宝

业务逻辑尽量使用常量，常量统一而且见名知义

1. 业务逻辑，不要使用显示值来做业务逻辑。

禁止下列行为：

if(status == “无货”)

{

disableBuyButton();

}

必须使用下列方式

if( stock == 0 || status == STOCK\_EMPTY)

{

disableBuyButton();

}

1. 如果一个方法写了50行以上，考虑一下是否有必要分开封装成多个方法
2. 如果一段代码被拷贝了3次，那就应该单独抽象成1个方法
3. 如果一个代码缩进了5层以上，那最好单独写个方法调用
4. 尽量少地暴露，变量作用域越小越好。
   1. 能用本模块完成的，就不用全局变量。
   2. 只在本模块用的变量，尽量定义成私有的
   3. 只在1个方法用到的变量，不要写成类变量
   4. 尽量少使用全局变量
5. 布局时，为了不拉伸图片，保持视觉效果，能固定数值的直接使用固定数值
   1. 单位使用dp，会保持此控件在任何密度的屏幕上大小不变
   2. 标题栏高度固定dp，按钮宽度和高度固定，图片的宽度和高度固定，边距固定
   3. ListView行高固定dp
   4. 尽量使点击区域不小于44dp X 44dp，如果图片太小，可以加padding
   5. 以上固定尺寸时dp的数值确定（效果图1080x1920）：

通常情况下，dp数值是效果图中px数值除以3.37四舍五入得到的，这个是给4.2寸以下屏幕用的；

sw360dp下的dimen.xml，dp数值是px数值除以3四舍五入得到的，这个是给大于4.2寸的屏幕用的。只有在大屏效果不好的情况下，才会填写这个文件指定尺寸。

* 1. 尽量使可点击的控件有按下效果，使用公司自定义的控件
  2. 字体，颜色，文本等建议放在资源文件中统一管理
  3. 界面实现尽量按照PSD来做，注意坐标、对齐、字体、颜色、阴影
  4. 把colors.xml、dimens.xml、strings.xml、styles.xml等用起来，能共用的部件比如导航栏、标题、字体样式等等最好使用style来写在同一个地方，界面上展现的字符串，要写在string.xml文件里
  5. 适配的时候，使用同1套layout，但是针对大于4.2寸的大屏幕，我们使用第2套dimens.xml，这个文件位于sw360dp目录下(意思就是宽度大于360dp的设备使用这套尺寸)，主要定义一些控件的宽高，边距，间距，字体大小等等。具体数值可以在布局编辑器里，使用5寸左右的手机进行查看和修改

1. Activity继承BaseActivity，使用它提供的加载框，各种提示框，这样可以在一个项目中统一样式，也减少了工作量
2. 优先使用公司库中封装好的控件或者工具类，因为统一成熟、有人测试有人维护。其次使用互联网上大家一致认可的第三方库，原因同上。实在找不到了才自己写。
3. xml布局文件上，必须有占位字符和占位图片，所见即所得的界面编辑器，不使用占位字符和占位图片就是浪费了所见即所得便利，因为看不见效果；禁止使用 “大保健”, “测试数据”等等数据做占位。尽量模仿真实数据占位。
4. 加载数据
   1. Activity创建时清空layout上的脏数据，加载上个页面传过来的数据。如果页面有缓存，需要加载缓存数据，请求回来后再更新一次数据
   2. 如果这个Activity只创建一次，后续还有刷新需求，在onResume里面根据刷新条件，重新刷新数据