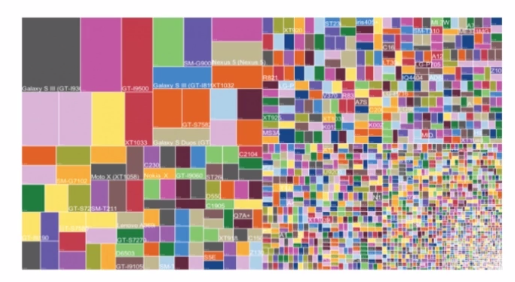
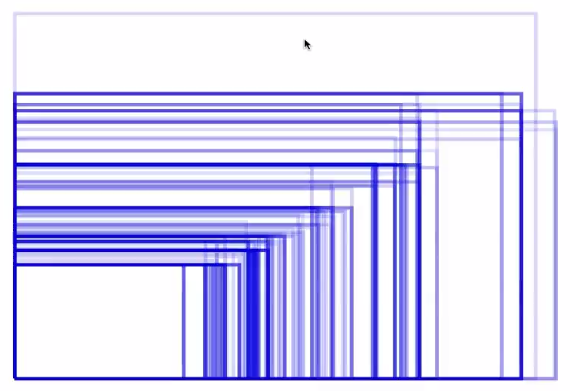
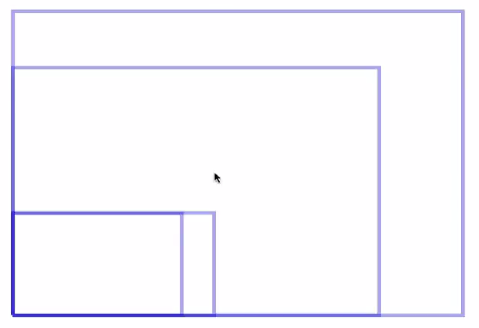
# 1、为什么要屏幕适配

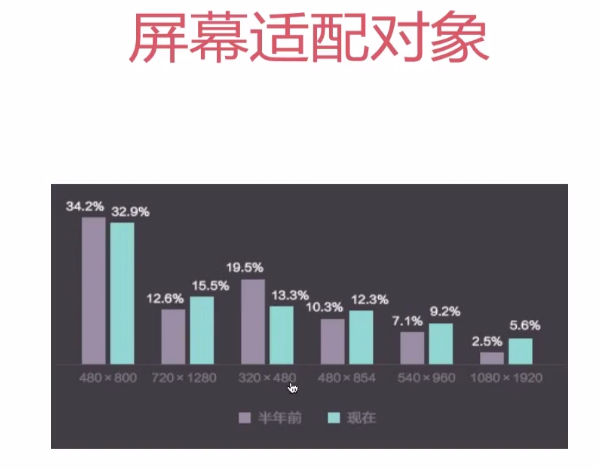


android品牌

android分辨率

iOS分辨率

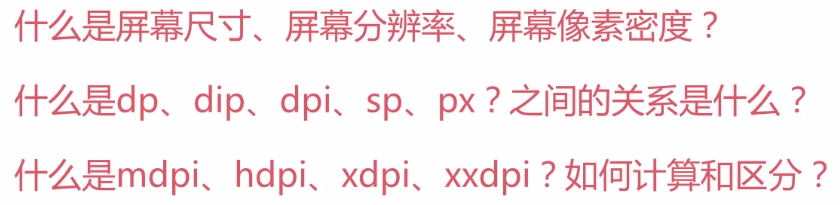
友盟的统计



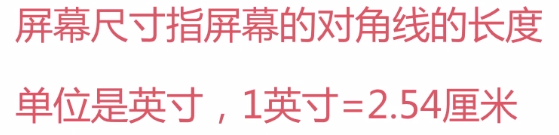
查看友盟指数：android设备分辨率趋势

对主要份额的屏幕进行适配。

# 2、屏幕尺寸像素密度

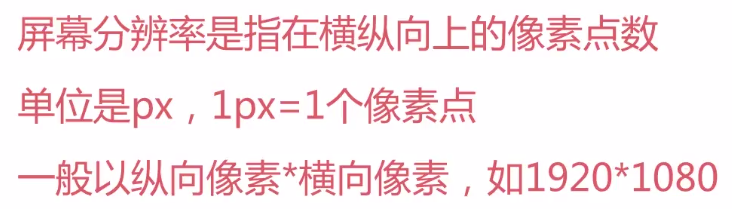


1. 屏幕尺寸

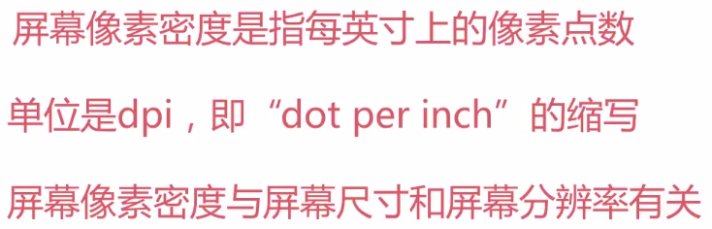


ZOL产品报价查询。

1. 屏幕分辨率: pixel



（3）屏幕像素密度



比如：Nexus 5 屏幕4.95inch，1920\*1080

计算：（1920^2 +1080^2）开平分 / 4.95 = 455，即dpi

1. px 像素

构成图像的最小单位

1. dp dip





假如，像素密度是160dpi,则其1dp相当于1px

像素密度是240dpi,则1dp相当于1.5px

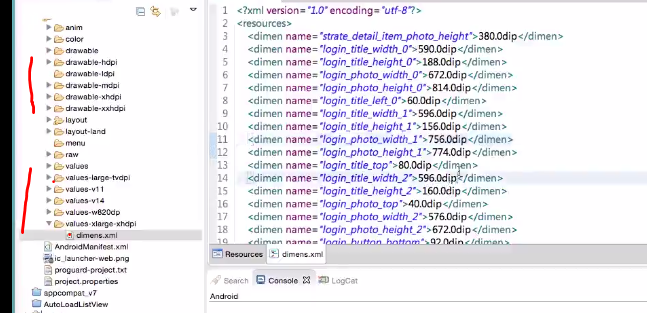
这样就统一了，使用320dp可以在不同的设备都是为等比例的显示，相同的效果。

1. sp 用于设置文字大小，尽量使用12sp\14sp\16sp\18sp。不要使用奇数，可能会造成精度丢失。

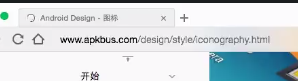


不同像素密度的区分，也是为适配的方便。

需要为不同像素密度的设备提供不同分辨率的图片。



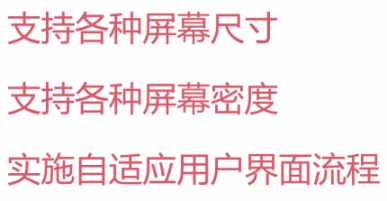




android design 设计规范指导



# 3、适配解决方案

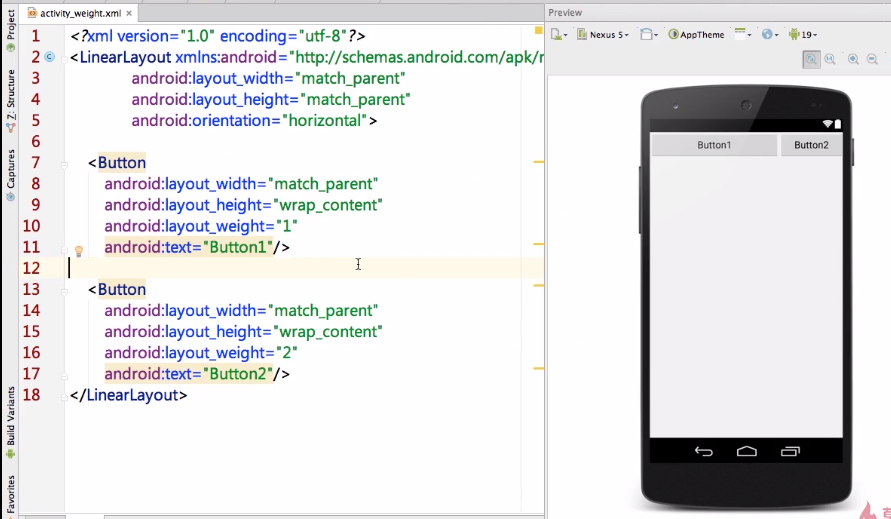


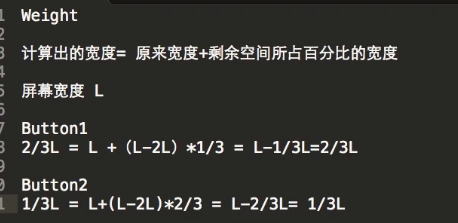
第3点，主要用于优化平板和手机之间的流程优化

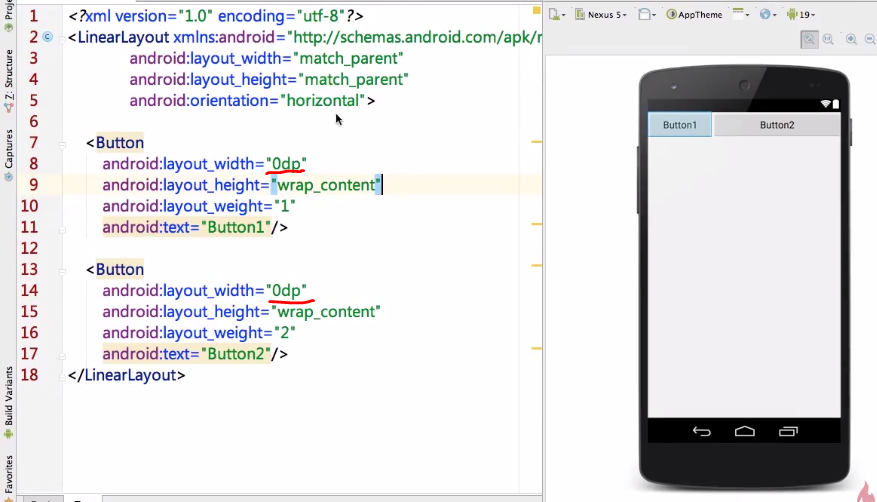
## **3.1 支持各种屏幕尺寸**



## 3.2 weight 权重的使用









## 3.3 使用相对布局

## 3.4使用large限定符

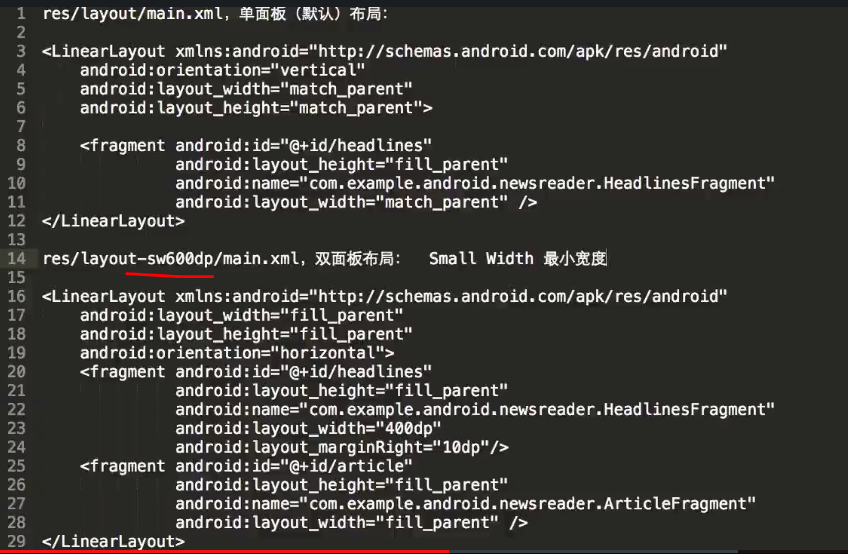




当在平板上使用，就会使用large模式。

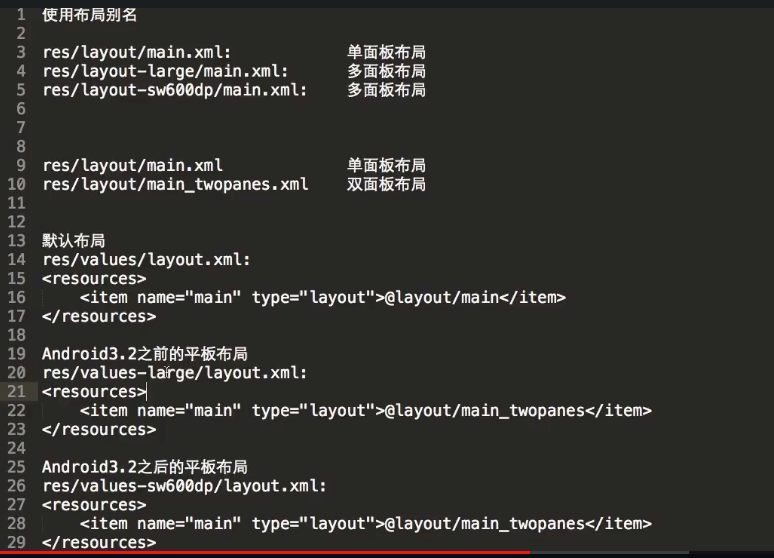
适配手机与平板。

## 3.5最小宽度限定符



主要设备的宽或者高小的那边尺寸大于600，就会使用这个布局。

## 3.6使用布局别名



setContentView(R.layout.main)；

取别名。

## 3.7使用屏幕方向限定符



land 水平方向 大于600dp

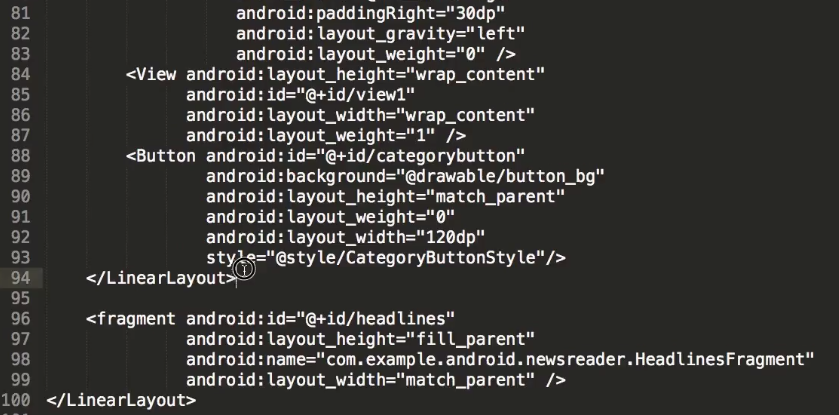
port 竖直方向 大于 600dp

1. 小例子





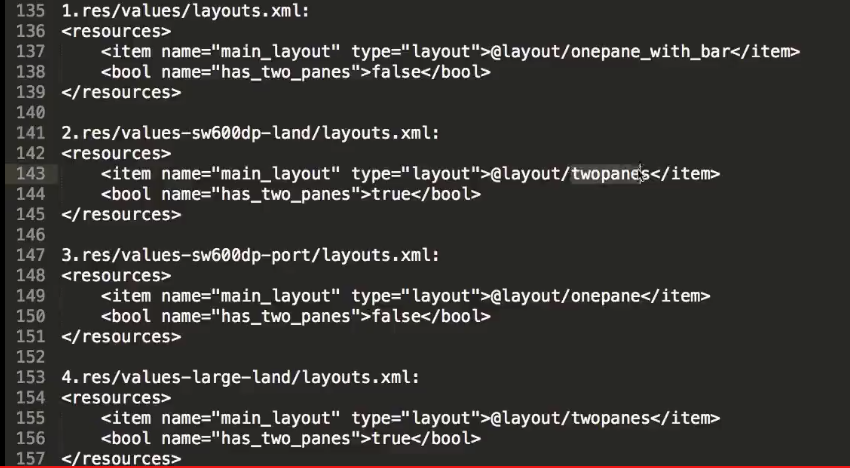




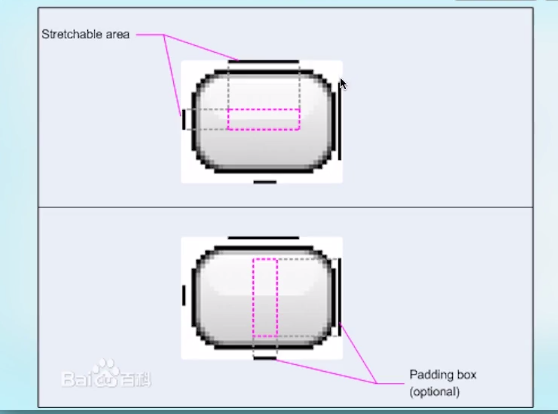




再利用上面的布局，进行设置布局别名：



## 3.8 .9图



拉伸区：就是可以拉伸的区域

间隔区：就是内容区

.9图的使用

点一个像素点就可以

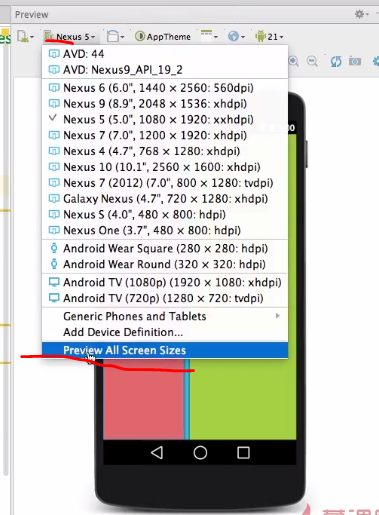
上下对应的比例拉伸，就要相同的像素点

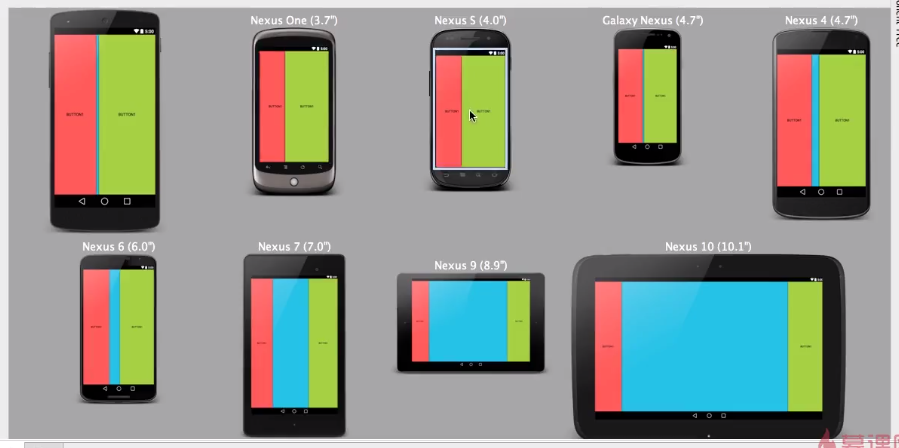
## 3.10 支持各种屏幕密度

使用非密度制约像素

提供备用位图

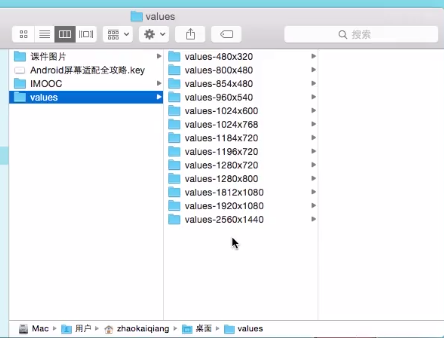
（1）使用非密度制约像素





如图，就是为什么要使用非像素密度像素的原因。

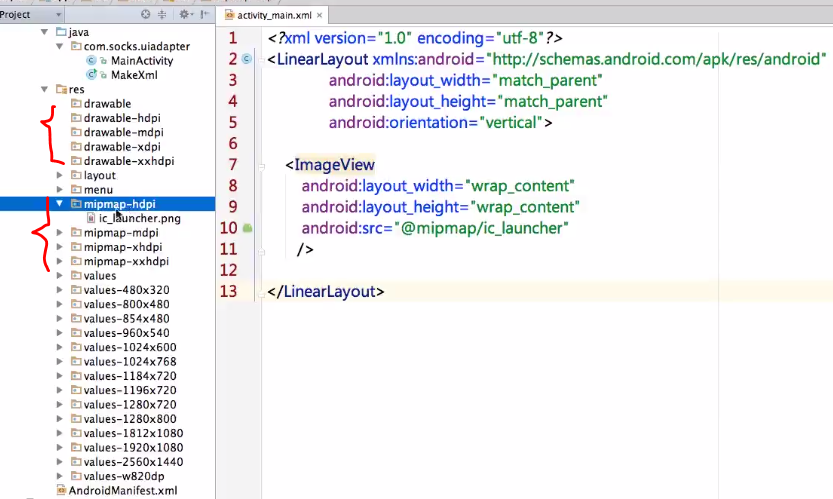
这是一个大的适配原因。



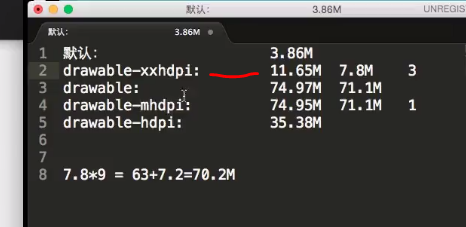
根据手机的像素生成资源文件，比如480分成320份。。。



1. 提高备用位图

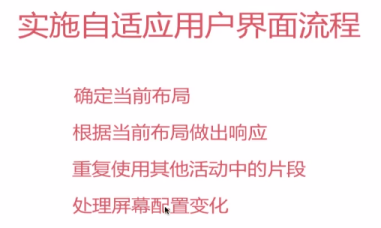


drawable mipmap的区别？

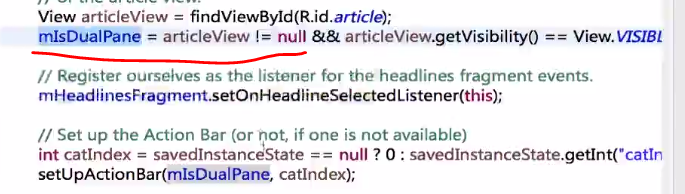


采用非对应的图片，会进行拉伸或压缩，不是优的,内存占用高。

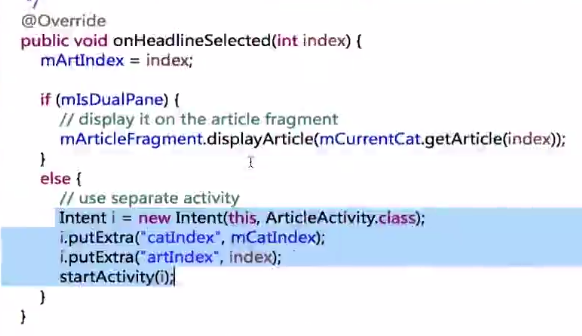
所以，最好还是按照不同的屏幕密度进行切图。

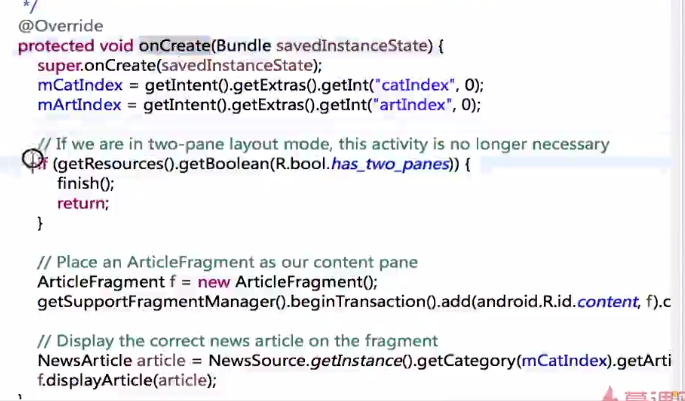


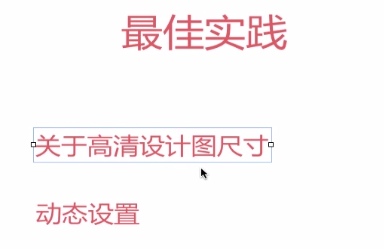
判断当前布局的模式，是单面板还是双面板？



区分操作：







## 3.11 动态设置

比如 pipWindow setLocation px，需要动态设置。

## 3.12 多看谷歌开发指导！！！