Android布局篇

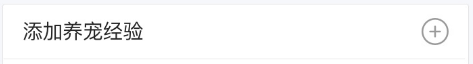
工程师 李洪江

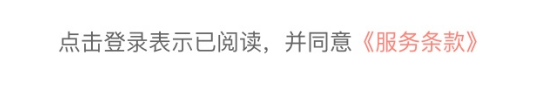
一 TextView组件使用

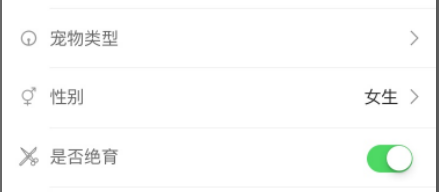
【1】利用DrawableLeft等属性来简化UI 布局。





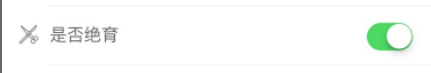






二 CheckedTextView使用

【1】支持checked属性，需要自己手动toggle。针对文字靠左的完美支持解决方案。







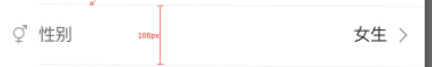
IMG_256

三 MaxWidth/MaxHeight等使用

【1】MaxWidth使用

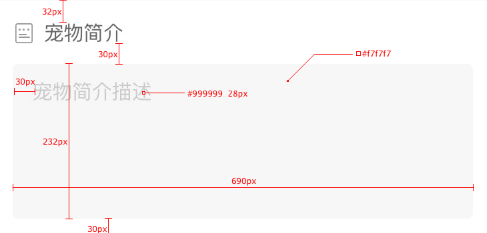


【2】MinHeight使用



四 minLines/maxLines使用

【1】固定Lines使用



五 weight 使用

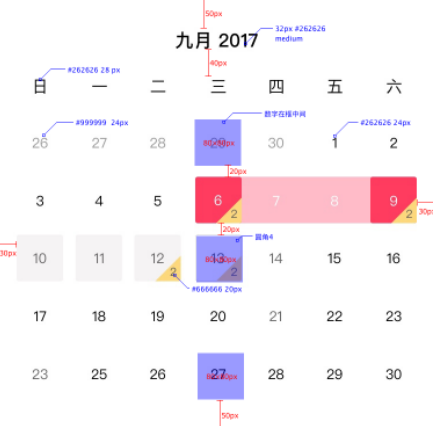


【1】使用weight一定要把对应的宽高设置为0，否则内容很长，会挤压后面weight元素，挤出非weight元素到屏幕外。

【2】一种典型布局是010，这种水平布局。也可以使用ReleativeLayout布局代替。



六 CheckBox使用



【1】 代替TextView使用，干掉默认按钮android:button=”@null”,禁止单击事件android:clickable=“false”；点击事件可以外包裹处理。

因为只有状态类控件才有checked属性。利用checked，selected，enable组合可设置8种选择器效果。

CheckedTextView不理想，文字居中有点问题。

七 FrameLayout使用

【1】常见四个角的角标icon控制。



【2】帧布局高度由内容的子元素最大高度决定。区别于相对布局高度测量必须明确知道子元素的高度。

【3】下图最好不使用相对布局。



1. ReleativeLayout使用



【1】 不规则布局使用。

【2】 使用依赖关系隐藏关联的布局，常见于分割线的控制。

【3】 使用相对布局最好能明确高，会影响测量性能。

九 padding/margin使用

【1】常见间隔线的使用。

【2】代替minHeight控制。

十 颜色选择器和背景选择器

【1】缓存不同背景，提高性能。

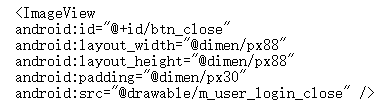
十一 形状的使用

【1】常见button效果。

十二 ImageView使用

【1】有时切图时候，不包含周边空白。此时我们只需要使用ImageView代替Button等布局。利用padding属性来设置空白，src设置icon，可以限制下宽高，这个宽高是padding+src宽高。不建议直接外包裹FrameLayout来做这种。

IMG_256



【2】包含间隔的不需要了。



十三 其他

【1】include复用布局使用。

【2】merge标签，自定义复合组件使用。

【3】ViewStub标签，不常用布局使用。

【4】fragment使用，复杂页面分成N个接口和布局加载。