Android系统中自带的主题(theme)和样式(style)

**主题：**针对窗体级别，改变窗体样式，多用于所有的或某个activity

**一、系统自带的主题**：

Eg. @style/AppTheme.NoActionBar 就是一个系统自带主题

**二、自定义主题：**

在res\values\style.xml（向下兼容）文件下添加自定义主题

（res/values-vXX/style.xml是高的api level）

<style name="AppBaseTheme" parent="android:Theme.Light">

<!--  上面的name值必须是全局唯一的，parent是它所继承的系统自带或自定义的主题

   -->

  <!--  这里可以写很多下述item，定义自己想要的样式

    <item name="colorBackground">#dcdcdc</item>

<item name="panelTextColor">?panelForegroundColor</item>

panelForegroundColor前面的?表示的是name为panelForegroundColor的属性已经在父类主题中写过了，表示引用了theme的一个已有属性panelForegroundColor，现在panelTextColor值和panelForegroundColor的值一样

  -->

</style>

如果你喜欢一个主题，但是想做一些轻微的改变，你只需要将这个主题添加为父主题,并修改想修改的属性即可

**三、使用方法：**

1、在AndroidManifes.xmlt文件中application标签中的android:theme

2、在Activity创建时调用 setTheme()方法（必须在setContentView前面调用）来给某个Activity添加主题

**样式:** Style是针对窗体元素级别的，改变指定控件或者Layout的样式

**系统自带样式：**eg. "@android:style/TextAppearance"

**自定义样式：**每个item的name是属性的名字，值是属性的值

Eg. <style name="CodeFont.Red.Big">

<!-继承自自定义style时要在前面加上父style的名字，

比如这里父style的名字为CodeFont

-->

        <item name="android:textSize">30sp</item>

  </style>"@android:style/ CodeFont.Red.Big"

Android系统的themes.xml和style.xml(位于系统源代码frameworks\base\core\res\res\values\)包含了很多系统定义好的style，建议在里面挑个合适的，然后再继承修改