

样式是属性的集合，例如定义属性fontColor、fontSize、layout_width、layout_height等，以独立的资源文件存放在XML文件中，并设置样式的名称。

Android Style类似网页设计中的级联样式CSS设计思路，可以让设计与内容分离，并且可以方便的继承、覆盖、重用

为使用样式的UI组件有很多冗余代码：

```
1.  <Button
2.      android:layout_width="match_parent"
3.      android:layout_height="match_parent"
4.      android:textSize="39sp"
5.      android:text="快乐的程序员" />
6.
7.  <Button
8.      android:layout_width="match_parent"
9.      android:layout_height="match_parent"
10.     android:textSize="39sp"
11.     android:text="快乐的程序员" />
12.
13. <Button
14.     android:layout_width="match_parent"
15.     android:layout_height="match_parent"
16.     android:textSize="39sp"
17.     android:text="快乐的程序员" />
```

style标签中name属性类似CSS中的class name，item标签中的name对应属性的名字，item标签对中的text对应属性的值

```
1.  <style name="mystyle">
2.      <item name="android:layout_width">wrap_content</item>
3.      <item name="android:layout_height">wrap_content</item>
4.      <item name="android:textSize">30sp</item>
5.  </style>
```

有两种方式来实现继承，一是通过style的parent属性，二是使用类似CSS中的命名规则来实现

```
1.  <style name="mystyle.son">
2.      <item name="android:textSize">40sp</item>
3.  </style>
```

Android中的主题：针对应用中所有Activity或者针对某个Activity设置样式，可以通过编辑AndroidManifest.xml来完成(样式和主题定义的规则都相同，都采用style，但是样式是给组件用的，主题是给app和activity用的)

```
1.  <application android:theme="@style/wrap_content">
```

设置某个指定的Activity主题

```
1.  <activity android:theme="@style/wrap_content">
```

android提供了许多自带的主题样式。例如Theme.Dialog、Theme.Translucent等等。使用方式也很简单

```
1. <activity android:theme="@android:style/Theme.Dialog">
```