

# 样式的继承

## 知识解析

### 样式的继承

有两种方式可以实现样式继承：

第一种方式可以在定义样式时使用 `parent` 属性来继承样式，使用这种方式，我们可以继承一个我们自己定义好的样式，也可以继承一个 `android` 平台自带的样式甚至整个主题：

```
<style
    name="GreenText"
    parent="@android:style/TextAppearance">
<item
    name="android:textColor">#00FF00</item>
</style>

<style
    name="CustomTheme" parent="android:Theme.Light">
<item
    name="android:windowBackground">@color/custom_theme_color</item>
</style>
```

另一种继承的方式是使用用户自定义的样式作为前缀即可。这种方式只适用于继承用户自定义样式。

```
<style name="TextStyle1.Red">
<item name="android:textColor">#FF0000
</item>
</style>
```

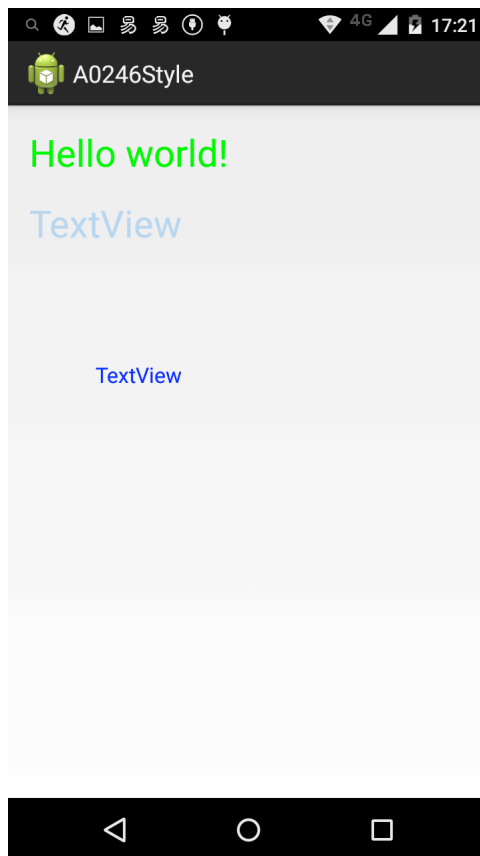
## 在 Activity/Application 中使用样式

通过在 AndroidManifest.xml 中的 application 或者 activity 节点指定 android:theme 属性即可指定预先定义好的主题：

```
<activity
    android:name=".MainActivity"
    android:theme="@style/my_style"
    android:label="@string/app_name">
```

对于 activity，也可以在 Java 中使用 `setTheme()` 方法，但需要在调用 `setContentView()`、`inflate()` 之前调用，否则不起效果。

## 功能演示



## 实战操作

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.exam.a0246style.MainActivity" >
    <TextView
        android:id="@+id/textView2"
        style="@style/StyleText1.StyleText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <TextView
        android:id="@+id/textView1"
        style="@style/StyleText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="15dp"
        android:text="TextView" />
    <TextView
        android:id="@+id/textView3"
        style="@style/StyleText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="84dp"
```

```
        android:text="TextView" />
    </RelativeLayout>
```

## 职业素质

- 1.在 `res/values` 目录下新建一个名叫 `style.xml` 的文件。增加一个 `<resources>` 根节点。
- 2.对每一个风格和主题，给 `<style>` element 增加一个全局唯一的名字，也可以选择增加一个父类属性。在后边我们可以用这个名字来应用风格，而父类属性标识了当前风格是继承于哪个风格。
- 3.在 `<style>` 元素内部，申明一个或者多个 `<item>`，每一个 `<item>` 定义了一个名字属性，并且在元素内部定义了这个风格的值。
- 4.你可以应用在其他 XML 定义的资源。