

# Selector

## 1.1 知识解析

如果想动态的改变不同状态下组件的样式，例如：获得焦点、失去焦点、按下时等等，可以使用 selector 来达到这个目的。

在 drawable 目录或者 layout 目录下,定义一个 selector,然后在需要使用 selector 中的组件中使用 android:background、android:textColor、listSelector (ListView 中) 等属性来调用。

可以使用的事件：

- android:state\_pressed=[ “true” | “false” ]：如触摸按钮或者点击按钮
- android:state\_focused=[ “true” | “false” ]：如使用方向键或者软键盘上的“下一个”让按钮 focused
- android:state\_selected=[ “true” | “false” ]：如打开一个 tab 时
- android:state\_checkable=[ “true” | “false” ]：只适用于在 checkable 和 uncheckable 之间切换的组件，当改组件可以被选中时
- android:state\_checked=[ “true” | “false” ]
- android:state\_enabled=[ “true” | “false” ]：当组件可用的时候
- android:state\_window\_focused=[ “true” | “false” ]：当应用在前景的时候，例如，弹出对话框后，原来 activity 的窗口此状态为 false，关闭对话框后为 true

例如：

```
<?xml version="1.0" encoding="utf-8"?>
<selector
xmlns:android="http://schemas.android.com/apk/res/and
roid" >
```

```
<item
    android:state_pressed="true"
    android:color="#FF0000"></item>

<item
    android:state_focused="true"
    android:color="#00FF00"></item>

<item
    android:color="#0000FF"></item>

</selector>
```

然后可以在 Button 组件中使用这个 selector:

```
<Button
    android:textColor="@drawable/button_text_selector"
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button1"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="75dp"
    android:text="Button" />
```

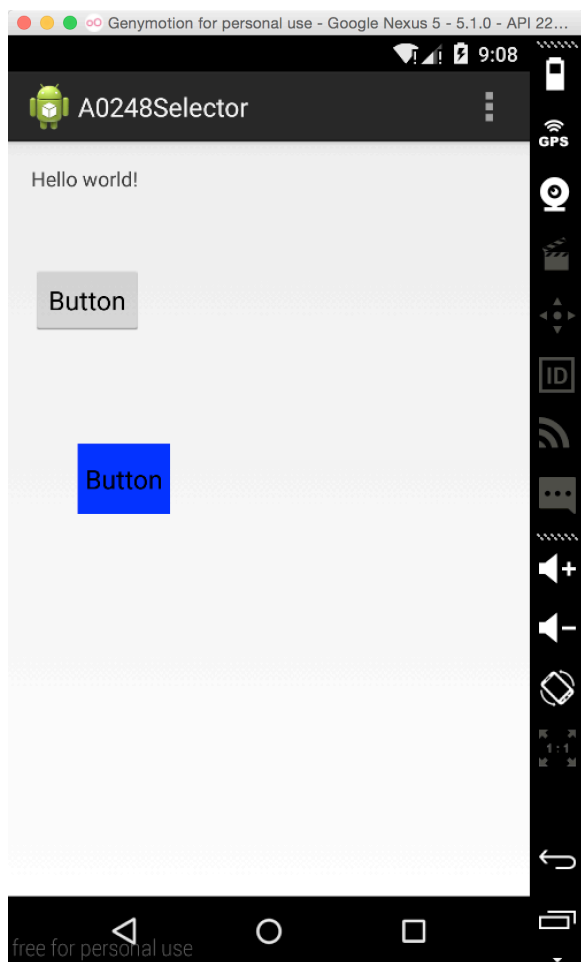
### 注意:

每个 item 对应一种状态。

在定义 selector 的时候,根据不同的目的,选择不同的 item 属性,例如,为了定义组件的背景,需要使用 android:drawable,为了定义文本颜色等,需要使用 android:color 等。

statelist 中第一个匹配当前状态的 item 会被使用。因此,如果第一个 item 没有任何状态特性的话,那么它将每次都被使用,这也是为什么默认的值必须总是在最后。

## 1.2 功能演示



## 1.4 职业素质

我们可以通过 `Selector` 等方式自定义一些我们想要的风格的控件样式。如果想动态的改变不同状态下组件的样式,例如:获得焦点、失去焦点、按下时等等,可以使用 `selector` 来达到这个目的。