

Android Mobile Programming

스타일과 셀렉터

조수연

010-4524-5468 / kgbsibbi@naver.com

스타일과 테마, 위젯 디자인과 셀렉터

스타일은 View나 Window의 모양과 형식(look and format)을 정의하는 속성들.

- height, padding, font color, font size, background color 등등.

스타일은 xml에 정의되며 이를 위젯들이 불러 쓰는 형태.

테마는 액티비티나 앱에 적용하는 스타일.

- 예를 들어 Activity 안의 모든 TextView에 공통으로 사용할 스타일을 테마로 지정하는 것.

위젯 디자인은 정의 된 스타일에 사용자가 정의한 위젯 이미지를 적용하여 완성할 수 있다.

이 때 위젯의 상태에 맞는 디자인을 적용하기 위해 셀렉터를 사용한다.

실습 프로젝트 만들기

Android Studio 실행

Start a new Android Studio project

Application name: DesignExample

Company Domain: style.com

Empty Activity로

MainActivity 추가하여 빈 프로젝트 생성

Application name:

DesignExample

Company Domain:

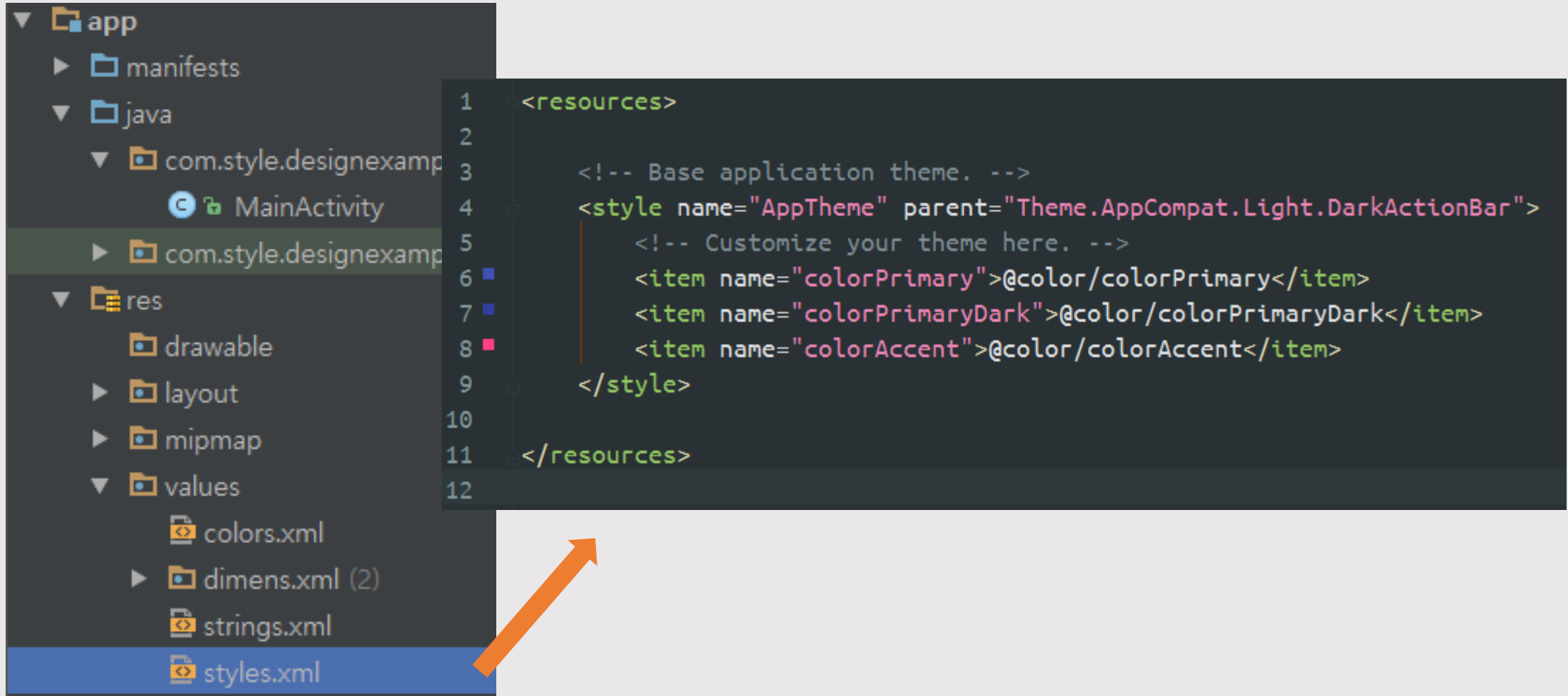
style.com

Package name:

com.style.designexample

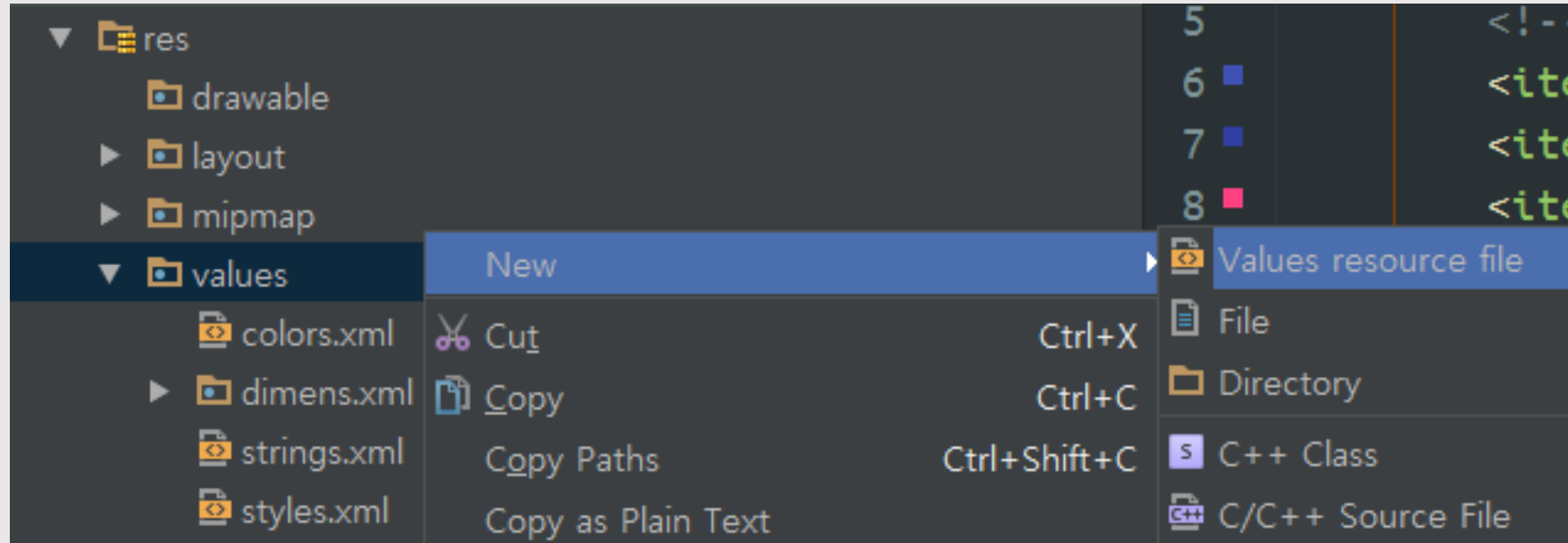
스타일 정의

res > values > styles.xml 파일에 앱의 기본 테마가 정의되어 있음.




스타일 정의

res > values > 마우스 오른쪽 클릭 > New > Values resource file



스타일 정의

파일 이름을 **button_styles.xml** 로 설정하여 파일 만들기

 New Resource File

File name:	<input type="text" value="button_styles.xml"/>	
Source set:	<input type="text" value="main"/>	▼
Directory name:	<input type="text" value="values"/>	

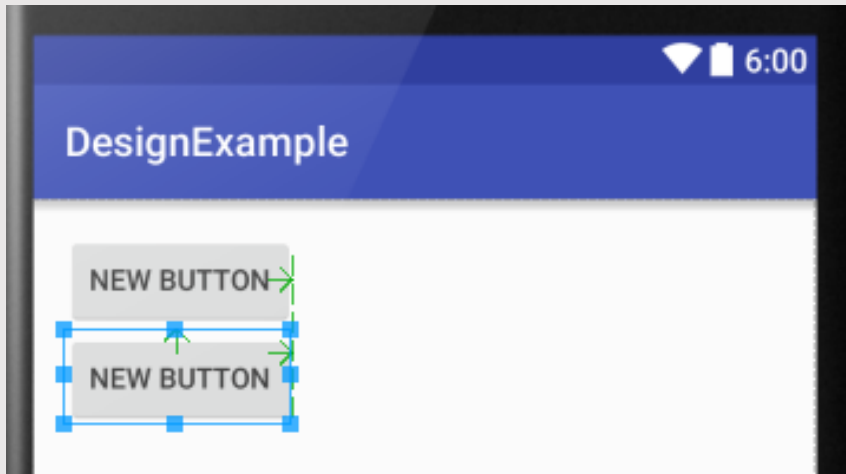
스타일 정의

button_styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="bigText"
        parent="@android:style/Widget.DeviceDefault.Button">
        <item name="android:textSize">30sp</item>
    </style>
</resources>
```

스타일 정의

activity_main.xml에 버튼을 두 개 추가



<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Button"  
    android:id="@+id/button"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentStart="true" />
```

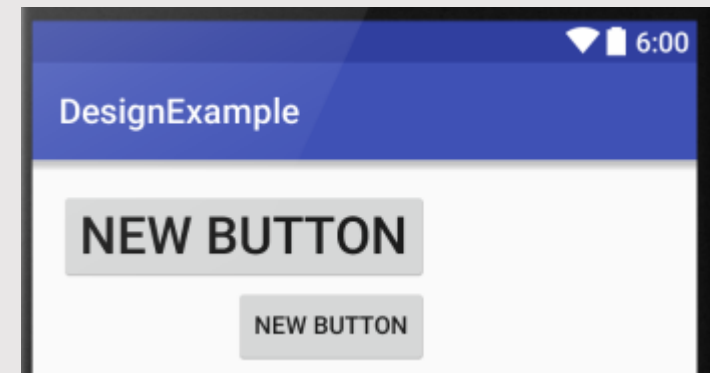
<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Button"  
    android:id="@+id/button2"  
    android:layout_below="@+id/button"  
    android:layout_alignEnd="@+id/button" />
```


스타일 정의

@+id/button 에 style부분 추가

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Button"  
    android:id="@+id/button"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentStart="true"  
    style="@style/bigText" />
```



스타일 정의

padding 및 글자 색 변경해보기: button_styles.xml 수정

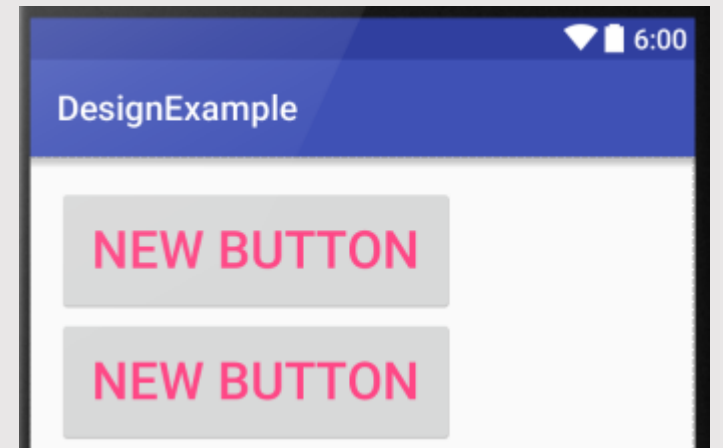
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="bigText"
        parent="@android:style/Widget.DeviceDefault.Button">
        <item name="android:textSize">30sp</item>
        <item name="android:padding">20dp</item>
        <item name="android:textColor">@color/colorAccent</item>
    </style>
</resources>
```

스타일 정의

정의 된 스타일은 앱 내의 위젯에게 반복적으로 설정할 수 있다.

@+id/button2에도 스타일을 적용해본다.

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Button"  
    android:id="@+id/button2"  
    android:layout_below="@+id/button"  
    android:layout_alignEnd="@+id/button"  
    style="@style/bigText" />
```



스타일 정의

안드로이드 스타일은 부모 스타일을 상속 받는다.

특별히 내가 선언하지 않은 스타일은 부모 스타일에 정의 된 내용을 그대로 사용하게 된다.

즉, 모든 스타일 속성을 다 선언할 필요는 없으며 **부모 스타일에서 고치고 싶은 부분만 재정의**하면 된다.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="bigText"
        parent="@android:style/Widget.DeviceDefault.Button">
        <item name="android:textSize">30sp</item>
        <item name="android:padding">20dp</item>
        <item name="android:textColor">@color/colorAccent</item>
    </style>
</resources>
```

스타일 정의

내가 정의한 스타일을 상속받아서 새 스타일 만들기: 먼저 정의된 스타일의 이름을 적고 . 뒤에 새로운 이름을 적으면 먼저 정의한 스타일을 자동으로 상속받는다.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="bigText"
        parent="@android:style/Widget.DeviceDefault.Button">
        <item name="android:textSize">30sp</item>
        <item name="android:padding">20dp</item>
        <item name="android:textColor">@color/colorAccent</item>
    </style>
    <style name="bigText.wide">
        <item name="android:layout_width">match_parent</item>
    </style>
</resources>
```

스타일 정의

@+id/button2 에 적용하기 → 새로 적용할 스타일과 충돌되는 설정은 삭제하거나 수정한다.

1. layout_width 삭제: 스타일에서 정의하고 있음.
2. layout_alignEnd를 layout_alignParentStart로 수정.

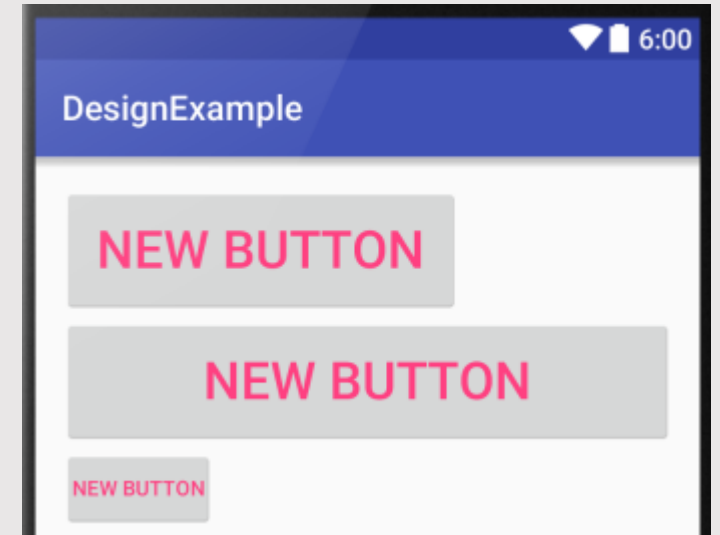
```
<Button  
    android:layout_height="wrap_content"  
    android:text="New Button"  
    android:id="@+id/button2"  
    android:layout_below="@+id/button"  
    android:layout_alignParentStart="true"  
    style="@style/bigText.wide" />
```



스타일 정의

실습

1. button_styles.xml 파일에 smallFont 스타일을 추가하라.
 1. textColor는 동일하게 accen로 설정
 2. fontSize는 12sp
 3. paddin은 5dp
2. activity_main.xml에 버튼을 하나 더 추가하고, smallFont 스타일을 적용하라.



스타일 정의

button_styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="bigText"
        parent="@android:style/Widget.DeviceDefault.Button">
        <item name="android:textSize">30sp</item>
        <item name="android:padding">20dp</item>
        <item name="android:textColor">@color/colorAccent</item>
    </style>
    <style name="bigText.wide">
        <item name="android:layout_width">match_parent</item>
    </style>
    <style name="smallFont">
        <item name="android:textSize">12sp</item>
        <item name="android:textColor">@color/colorAccent</item>
        <item name="android:padding">5dp</item>
    </style>
</resources>
```


스타일 정의

activity_main.xml 내의 @+id/button3

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Button"  
    android:id="@+id/button3"  
    android:layout_below="@+id/button2"  
    android:layout_alignParentStart="true"  
    style="@style/smallFont" />
```

Selector

스타일을 정의할 때, 또는 스타일이 아니더라도 화면상의 위젯의 디자인을 하나만 변경할 때 하나의 고정 이미지 또는 고정 색으로 정의하기 힘든 경우가 많다.

버튼의 경우 일반적인 상태와 클릭 상태, 그리고 Disable 의 색이나 이미지가 달라야 한다.

텍스트 입력 창 역시 일반적인 상태와 클릭하여 글자를 입력 중인 상태, 그리고 Disable의 상태가 구분 되어야 한다.

체크박스나 라디오 버튼의 경우 Enable/Disable 각각에 대하여 선택이 되었느냐 되지 않았느냐의 구분이 다시 필요하다.

각 상태마다 다른 색 또는 이미지를 설정할 수 있는 도구가 필요하며 안드로이드에서 제공하는 도구가 Selector 이다.

Selector

색상 추가하기

- 버튼에서 사용할 각 상황에서의 색상을 선언한다.

[배경 색]

1. Enabled + Normal 상태 : 연한 회색
2. Enabled + Pressed 상태 : accent (핑크)
3. Disabled 상태 : 진한 회색

[글자 색]

1. Enabled + Normal 상태 : accent (핑크)
2. Enabled + Pressed 상태 : 연한 회색
3. Disabled 상태 : 회색

Selector

res > values > colors.xml 파일에 다음과 같이 추가

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>

    <color name="enabledGray">#DDDDDD</color>
</resources>
```

Selector

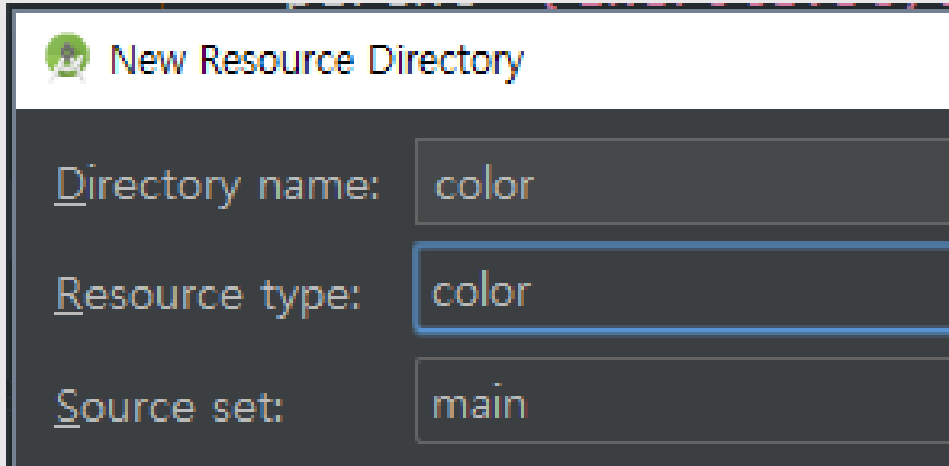
res > drawable 폴더 > button_background.xml 파일 추가

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="false"
        android:drawable="@color/enabledGray">
    </item>
    <item android:state_pressed="true"
        android:drawable="@color/colorAccent">
    </item>
</selector>
```

Selector

res > 마우스 오른쪽 버튼 클릭 > New > Android Resource Directory

Resource type을 color로 선택하여 완료



New Resource Directory

Directory name: color

Resource type: color

Source set: main

Selector

res > color 폴더 안에 다음 파일 생성 > button_text_color.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="false"
        android:color="@color/colorAccent">
    </item>
    <item android:state_pressed="true"
        android:color="@color/enabledGray">
    </item>
</selector>
```

Selector

만들어진 Selector를 Style에 적용: button_styles.xml 수정

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="bigText"
        parent="@android:style/Widget.DeviceDefault.Button">
        <item name="android:textSize">30sp</item>
        <item name="android:padding">20dp</item>
        <item name="android:textColor">@color/button_text</item>
        <item name="android:background">@drawable/button_background</item>
    </style>
```

아래 생략. 수정 사항 없음.

Selector

주의 할 점.

1. `textColor` 는 `color` 폴더 안의 `selector`를 사용한다.
 1. `color` 폴더 안의 `selector item`에는 `color` 항목이 들어가야 한다.
2. `background`는 `drawable` 폴더 안의 `selector`를 사용한다.
 1. `drawable` 폴더 안의 `selector item`에는 `drawable` 항목이 들어가야 한다.
 2. `drawable` 항목의 값은 `@drawable` 로 시작하는 실제 그림이 될 수도 있고 `@color`로 시작하는 색이 될 수도 있다.

Selector

버튼의 모서리 둥글게: button_background.xml 을 다음과 같이 수정한다.

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="false">
    <shape android:shape="rectangle">
      <corners android:radius="15dp"></corners>
      <solid android:color="@color/enabledGray"></solid>
    </shape>
  </item>
  <item android:state_pressed="true">
    <shape android:shape="rectangle">
      <corners android:radius="15dp"></corners>
      <solid android:color="@color/colorAccent"></solid>
    </shape>
  </item>
</selector>
```

Selector

버튼의 테두리 색상/두께/점선 지정

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="false">
    <shape android:shape="rectangle">
      <corners android:radius="15dp"></corners>
      <stroke android:color="@color/colorAccent"
        android:width="2dp" android:dashWidth="5dp" android:dashGap="5dp"></stroke>
      <solid android:color="@color/enabledGray"></solid>
    </shape>
  </item>
  <item android:state_pressed="true">
    <shape android:shape="rectangle">
      <corners android:radius="15dp"></corners>
      <solid android:color="@color/colorAccent"></solid>
    </shape>
  </item>
</selector>
```

Selector

버튼의 색상을 그라데이션으로 지정: solid 항목을 gradient로 변경

위쪽 생략

```
<item android:state_pressed="true">
    <shape android:shape="rectangle">
        <corners android:radius="15dp"></corners>
        <gradient android:angle="270"
            android:startColor="@color/enabledGray"
            android:endColor="@color/colorAccent"></gradient>
    </shape>
</item>
</selector>
```

정리

1. res > values > colors.xml에 필요한 색상을 지정
2. res > color 또는 res > drawable에 Selector 생성
3. res > values 에 style 파일 생성
4. 상속 받을 부모 클래스를 정한 뒤 원하는 항목만 내가 재정의 해서 스타일을 만듦
5. 레이아웃에 있는 위젯들이 그 스타일을 사용함.

Edit Text 스타일

EditText: 글 상자의 색 및 글자 색

1. res / drawable 폴더에 edit_text_background.xml 파일 생성
2. 글자 색은 버튼과 동일하게 사용하기 위해 따로 xml파일을 만들지 않음.
3. res/values 폴더에 edit_text_style.xml 파일을 생성.
4. activity_main.xml에 EditText Widget을 하나 추가 하고 스타일을 지정.

Edit Text 스타일

res / drawable 폴더에 edit_text_background.xml 파일

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="false">
    <shape android:shape="rectangle">
      <corners android:radius="15dp"></corners>
      <stroke android:color="@color/colorAccent" android:width="1dp"></stroke>
      <solid android:color="@color/enabledGray"></solid>
    </shape>
  </item>
  <item android:state_pressed="true">
    <shape android:shape="rectangle">
      <corners android:radius="15dp"></corners>
      <stroke android:color="@color/enabledGray" android:width="1dp"></stroke>
      <solid android:color="@color/colorAccent"></solid>
    </shape>
  </item>
</selector>
```

Edit Text 스타일

res / values 폴더에 edit_text_style.xml 파일

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="vividColor"
        parent="@android:style/Widget.DeviceDefault.EditText">
        <item name="android:textColor">@color/button_text</item>
        <item name="android:background">@drawable/edit_text_background</item>
        <item name="android:padding">20dp</item>
    </style>
</resources>
```


Edit Text 스타일

activity_main.xml 에 EditText를 추가

```
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/editText"  
    android:layout_below="@+id/button3"  
    android:layout_alignParentStart="true"  
    style="@style/vividColor" />
```