



# Android 사용자 인터페이스

---

배 희호 교수  
경북대학교  
스마트IT과



# EditText

- EditText는 TextView로부터 파생된 클래스로 TextView는 단순히 Text를 보여주는 역할을 한다면 EditText는 Text를 입력 및 수정까지 가능한 View Widget
- EditText를 화면에 배치하는 방법은 Layout 리소스 XML에서 `<EditText></EditText>` 사이에 EditText의 속성을 정의하면 됨

```
<EditText  
    android:id="@+id/editText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:text="Sample EditText" />
```



# EditText



## ■ EditText 입력 및 수정 방지

- 부모 클래스로부터 상속받은 속성 중 enabled 속성을 사용하면 EditText에 텍스트를 입력 및 수정이 불가능한 상태로 설정할 수 있음
- default 속성값은 true로 사용 가능한 상태이며 false로 지정하게 되면 EditText를 사용할 수 없는 상태가 됨

```
<EditText
    android:id="@+id/Sample1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="editable = true" />
<EditText
    android:id="@+id/Sample2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="editable = false" />
```



# EditText



- EditText 속성(hint / textColorHint)
  - hint 속성은 EditText 배경에 속성값으로 지정한 문자열을 나타나게 함
  - Text 영역에 어떠한 내용을 입력하라는 간단한 안내 문구나 텍스트 입력 예시를 표시할 때 사용
  - textColorHint 속성은 hint 속성에 의해 지정된 문자열의 색상을 지정할 때 사용

```
<EditText  
    android:id="@+id/Sample1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="이메일을 입력하세요."  
    android:inputType="textPersonName"  
    android:textColorHint="@color/colorAccent" />
```



# EditText



- EditText 속성(selectAllOnFocus / textColorHighlight)
  - selectAllOnFocus 속성은 EditText를 클릭하였을 때 텍스트 영역을 전체 선택된 상태를 만들려고 할 때 사용
  - EditText를 클릭하고 텍스트를 입력하게 되면 기존에 있는 텍스트 내용은 일괄적으로 삭제되면서 새로 입력된 텍스트가 입력
  - textColorHighlight 속성은 EditText에서 Text에 해당하는 사각형 영역을 표현할 때 사용

```
<EditText  
    android:id="@+id/Sample1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:selectAllOnFocus="true"  
    android:text="Sample"  
    android:textColorHighlight="@color/colorPrimary" />
```



# EditText

## ■ EditText 속성(maxLength)

- maxLength 속성은 EditText에 입력 가능한 텍스트의 수를 지정할 수 있음

```
<EditText  
    android:id="@+id/editText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:maxLength="5"  
    android:text="Name" />
```

텍스트 길이는 5글자로 설정하였기 때문에  
위 결과화면에서 더 이상의 텍스트 입력은  
불가능



# EditText



- EditText 속성(background 값에 @null)
  - EditText 밑줄은 default 값으로 존재
  - 디자인 상으로 밑줄을 없애고 싶을 때, XML 파일에서 간단히 제거할 수 있음

```
<EditText  
    android:id="@+id/et_title"  
    android:hint="제목"  
    android:gravity="left|center"  
    android:textSize="16dp"  
    android:paddingLeft="10dp" />
```

```
<EditText  
    android:id="@+id/et_title"  
    android:hint="제목"  
    android:gravity="left|center"  
    android:textSize="16dp"  
    android:paddingLeft="10dp"  
    android:background="@null" />
```



# EditText



- EditText 속성 (imeOptions)
  - 입력 후 엔터 키 -> 키보드 사라지게

```
android:singleLine="true"  
android:imeOptions="actionDone"
```





# EditText



## ■ Keyboard의 <Enter> 버튼을 눌렀을 경우

```
input.setOnEditorActionListener(new OnEditorActionListener() {  
    @Override  
    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  
        if (actionId == EditorInfo.IME_ACTION_DONE ||  
            event.getKeyCode() == KeyEvent.KEYCODE_ENTER) {  
            return true;  
        }  
        return false;  
    }  
});
```



# 문자열 리소스(strings.xml)

- Android Project에서 /res/value 경로 아래에 strings.xml이라는 문자열 리소스 파일이 있음
- Project를 처음 생성하면 아래와 같이 기본으로 app\_name이라는 이름을 가진 문자열 리소스가 있음
- 사용할 문자열 추가하기
  - app\_name과 같은 형태로 태그로 작성하려는 문자열을 감싸주면 됨
  - 반드시 name="" 속성을 사용해서 해당 문자열 리소스의 이름을 지정해 주어야 하며, 지정한 이름으로 코드에서 사용할 수 있음

```
<resources>
  <string name="app_name">Sample</string>
  <string name="name">홍길동</string>
  <string name="nickname">별명이 무엇입니까?</string>
</resources>
```



# 문자열 리소스(strings.xml)

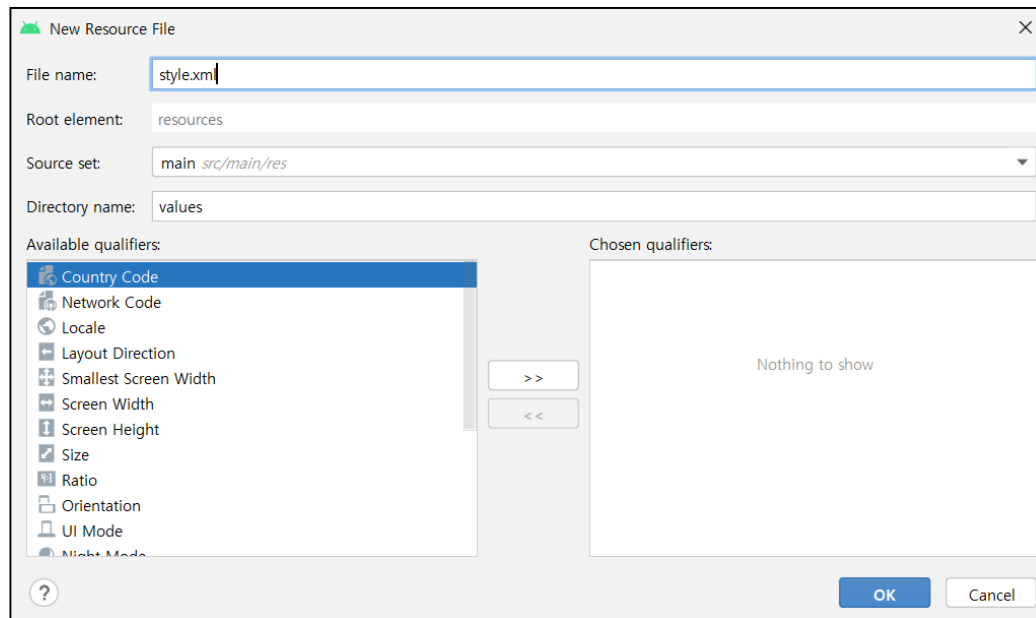
- XML 파일에서 사용하기

- 속성 중 android:text=""에 <string> 태그의 name 속성 값을 넣어줌
- 예) android:text="@string/name"
- 예) android:text="@string/nickname"



# Style 적용하기

- TextView와 EditText에서 속성 값으로 {fontFamily, textColor, textSize}가 중복으로 인해 코드가 불필요하게 많아짐 -> 이를 해결하기 위해서 style.xml을 사용
- Style.xml 파일 추가
  - /res/values 폴더를 우 클릭
  - New/Value Resource File을 클릭





# Style 적용하기

## ■ style.xml에 style 추가하기

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="MyStyle">
        <item name="android:fontFamily">@font/fontstyle</item>
        <item name="android:textColor">@color/black</item>
        <item name="android:textSize">20sp</item>
    </style>
</resources>
```



# Style 적용하기



## ■ 중복을 줄여보자

```
<TextView
    android:id="@+id/name_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="@font/fontstyle"
    android:text="name"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="20sp" />
```

```
<EditText
    android:id="@+id/editText_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="@font/fontstyle"
    android:hint="what_is_your_nickname"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="20sp" />
```



# Style 적용하기

## ■ 중복을 줄여보자

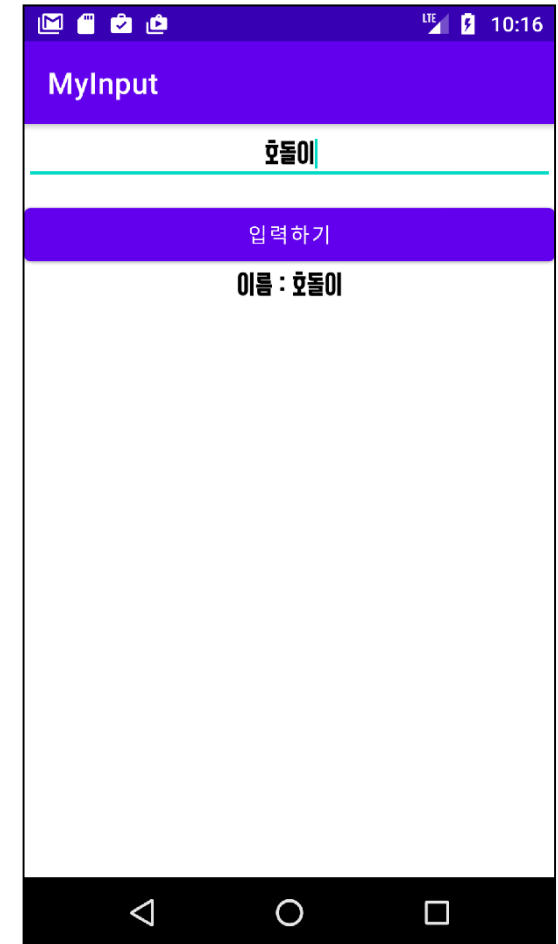
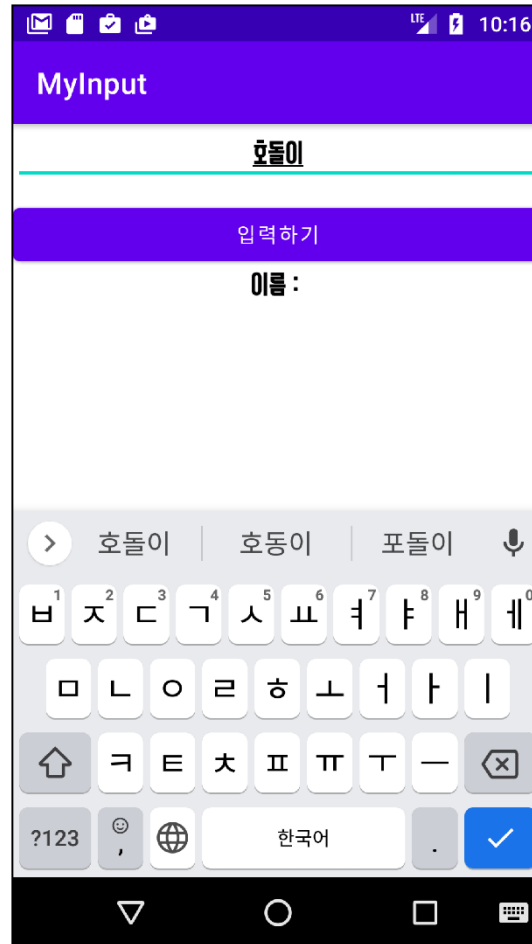
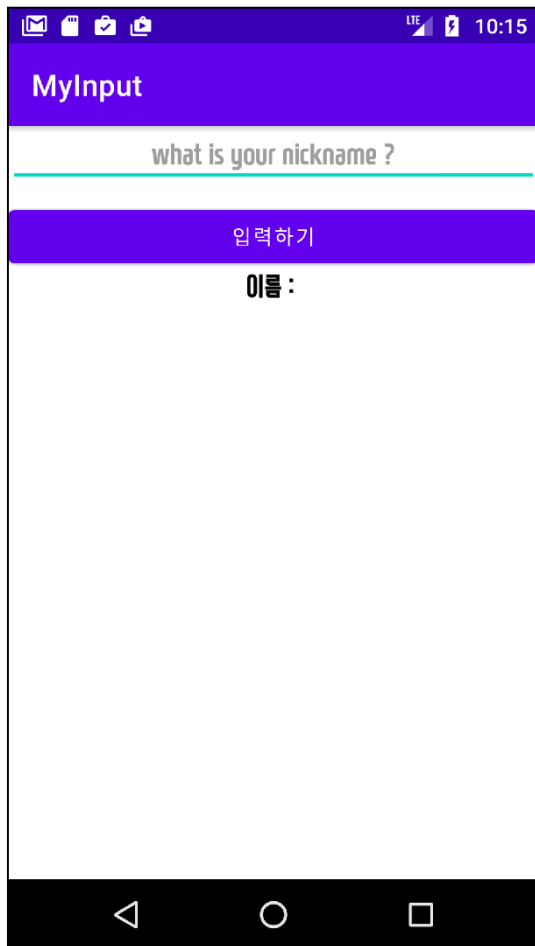
```
<TextView  
    android:id="@+id/name_text"  
    style="@style/MyStyle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="name"  
    android:textAlignment="center"/>
```

```
<EditText  
    android:id="@+id/editText_name"  
    style="@style/MyStyle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="what_is_your_nickname"  
    android:textAlignment="center" />
```



# 실습 예제

- EditText에 입력된 Text를 가져와 TextView의 Text에 출력하는 프로그램을 작성하여라.







# 실습 예제

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        style="@style/MyStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="what is your nickname ?"
        android:textAlignment="center"
        android:singleLine="true"
        android:imeOptions="actionDone"/>
```



# 실습 예제

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/show"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:text="입력하기" />
```

```
<TextView  
    android:id="@+id/textView"  
    style="@style/MyStyle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="이름 : "  
    android:textAlignment="center" />
```

```
</LinearLayout>
```



# 실습 예제

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    InputMethodManager manager;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        manager = (InputMethodManager) getSystemService(  
            Context.INPUT_METHOD_SERVICE);  
        EditText input = findViewById(R.id.editText);  
  
        TextView textView = findViewById(R.id.textView);  
        Button show = findViewById(R.id.show);  
    }  
}
```



# 실습 예제

## ■ MainActivity.JAVA

```
show.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String msg = input.getText().toString();  
        if (msg.equals("")) {  
            Toast.makeText(getApplicationContext(), "입력해주세요",  
                Toast.LENGTH_SHORT).show();  
        } else {  
            textView.setText("이름 : " + msg);  
        }  
    }  
});
```



# 실습 예제

## ■ MainActivity.JAVA

```
input.setOnEditorActionListener(new TextView.OnEditorActionListener() {  
    @Override  
    public boolean onEditorAction(TextView textView, int i,  
                                   KeyEvent keyEvent) {  
        if (i == EditorInfo.IME_ACTION_DONE) {  
            manager.hideSoftInputFromWindow(input.getWindowToken(), 0);  
            return true;  
        }  
        return false;  
    }  
});  
}
```



# Soft Keyboard

## ■ Android KeyBoard 입력 창 올리기/ 내리기

### ■ InputMethodManager 객체를 선언

```
InputMethodManager manager = (InputMethodManager)
    getSystemService(INPUT_METHOD_SERVICE);
EditText input = findViewById(R.id.editText);
```

### ■ KeyBoard 보이기/올리기

```
manager.showSoftInput(input, 0);
```

### ■ 키보드 숨기기/내리기

```
manager.hideSoftInputFromWindow(input.getWindowToken(), 0);
```



# Soft Keyboard



- AndroidManifest.xml에서의 KeyBoard 제어하기
  - AndroidManifest.xml에 설정값을 넣어줌으로써, Activity 시작 시에 자동으로 KeyBoard를 보이거나 숨길 수 있음
  - <activity> 설정에서 android:windowSoftInputmode의 value에 "stateAlwaysVisible"과 "stateAlwaysHidden"를 넣어주면 됨
  - KeyBoard 보이기

```
<activity android:name=".MainActivity"
          android:windowSoftInputMode="stateAlwaysVisible"/>
```

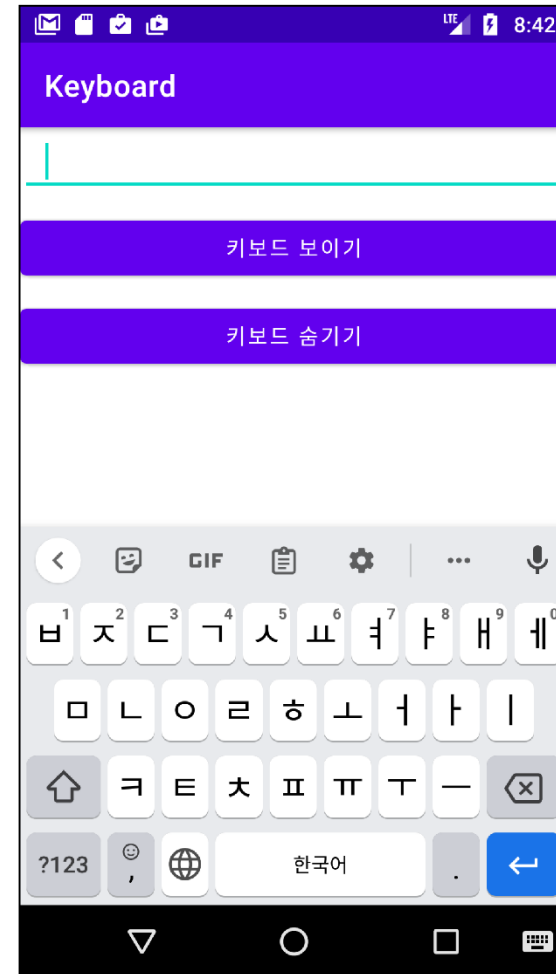
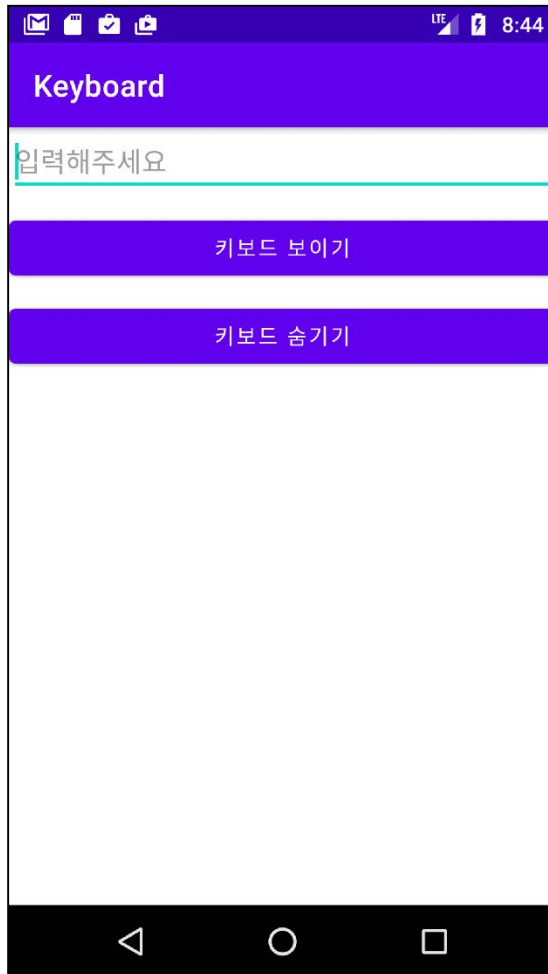
- KeyBoard 숨기기

```
<activity android:name=".MainActivity"
          android:windowSoftInputMode="stateAlwaysHidden"/>
```



# 실습 예제

## ■ Keyboard를 제어해보자







# 실습 예제

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="입력해주세요"
        android:textColor="#000000" />
```



# 실습 예제

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/show"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:text="키보드 보이기" />
```

```
<Button  
    android:id="@+id/hide"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:text="키보드 숨기기" />
```

```
</LinearLayout>
```



# 실습 예제

## ■ 사용자 인터페이스

```
public class MainActivity extends AppCompatActivity
                                   implements View.OnClickListener {

    InputMethodManager manager;
    EditText input;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        input = findViewById(R.id.editText);
        Button show = findViewById(R.id.show);
        Button hide = findViewById(R.id.hide);
        manager = (InputMethodManager) getSystemService(
                                                    INPUT_METHOD_SERVICE);

        show.setOnClickListener(this);
        hide.setOnClickListener(this);
    }
}
```



# 실습 예제

## ■ 사용자 인터페이스

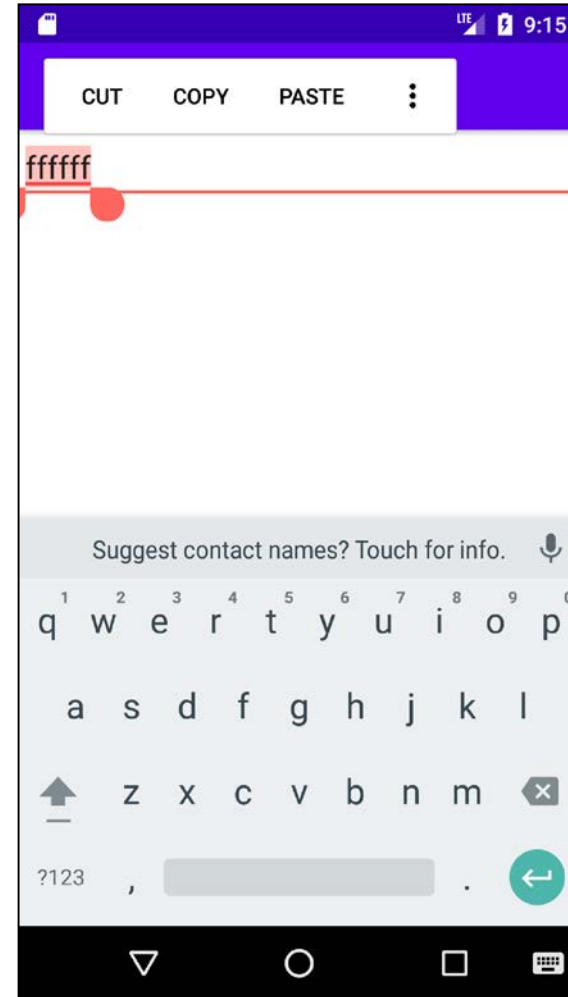
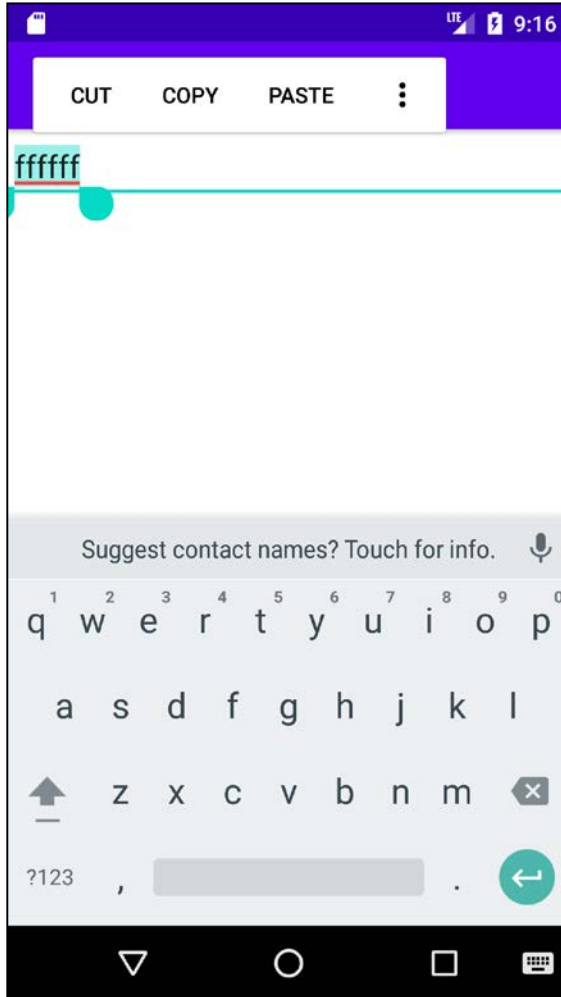
@Override

```
public void onClick(View view) {  
    switch (view.getId()) {  
        case R.id.show:  
            manager.showSoftInput(input, 0);  
            break;  
        case R.id.hide:  
            manager.hideSoftInputFromWindow(input.getWindowToken(), 0);  
            break;  
    }  
}
```



# EditText의 Cursor 색상 변경하기

## ■ 실행 화면





# EditText의 Cursor 색상 변경하기

- EditText에 적용되는 색상 확인하기
- themes.xml에서 확인 및 변경 가능

```
activity_main.xml x colors.xml x themes.xml x styles.xml x MainActivity.java x
1 <resources xmlns:tools="http://schemas.android.com/tools">
2   <!-- Base application theme. -->
3   <style name="Theme.MyInput02" parent="Theme.MaterialComponents.DayNight"
4     <!-- Primary brand color. -->
5     <item name="colorPrimary">@color/purple_500</item>
6     <item name="colorPrimaryVariant">@color/purple_700</item>
7     <item name="colorOnPrimary">@color/white</item>
8     <!-- Secondary brand color. -->
9     <item name="colorSecondary">@color/teal_200</item>
10    <item name="colorSecondaryVariant">@color/teal_700</item>
11    <item name="colorOnSecondary">@color/black</item>
12    <!-- Status bar color. -->
13    <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimary</item>
14    <!-- Customize your theme here. -->
15  </style>
16 </resources>
```



# RGB 색상표



Google search results for "rgb 색상표".

Search results include:

- RGB 색상 코드 차트 - RT**  
[www.rapidtables.org](http://www.rapidtables.org) > web > color > RGB\_Color  
RGB 컬러 공간 또는 RGB 컬러 시스템은 상기의 조합으로부터 모든 색을 구성 R의 ED, G의 재 N 및 B의 루 색상. 빨강, 녹색 및 파랑은 각각 8 비트를 사용하며 0에서 255 ...
- 이미지**  
모두 보기  
A carousel of images showing various color charts and palettes.
- 256 color 색상표**  
[www.n2n.pe.kr](http://www.n2n.pe.kr) > lev-1 > color  
색이 변합니다. Your browser doesn't support java or java is not enabled! RGB값을 16진수로 바꾸어 드립니다. R, ...
- RGB색상표/16진수 컬러 코드표 - Hi!Penpal!**  
[www.hipenpal.com](http://www.hipenpal.com) > tool > html-color-charts-rgb-color-table-in-korean  
HTML 색상표/RGB색상코드표팔레트. H: °. S: %. B: %. R: G: B: #. 93DAFF. 98DFFF. 9DE4FF. A2E9FF. A7EEFF. ACF3FF. B0F7FF. B4FBFF. B9FFFF. C0FFFF. 87CEFA. 91D8FA.



# EditText의 Cursor 색상 변경하기

- EditText Cursor 색/밑줄 HighLight 색 변경하는 방법
- colors.xml 파일에 색상 추가

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="purple_200">#FFBB86FC</color>
4     <color name="purple_500">#FF6200EE</color>
5     <color name="purple_700">#FF3700B3</color>
6     <color name="teal_200">#FF03DAC5</color>
7     <color name="teal_700">#FF018786</color>
8     <color name="black">#FF000000</color>
9     <color name="white">#FFFFFFFF</color>
10
11     <color name="cursorColor">#ff665d</color>
12     <color name="nomalColor">#ffcd54</color>
```

```
<color name="cursorColor">#ff665d</color>
<color name="nomalColor">#ffcd54</color>
```





# EditText의 Cursor 색상 변경하기

- EditText Cursor 색/밑줄 HighLight 색 변경하는 방법
  - styles.xml 파일을 만들고 색상 추가

```
<resources>
  <style name="EditTheme" parent="Theme.AppCompat.Light.NoActionBar">
    <item name="colorAccent">@color/cursorColor</item>
    <item name="colorControlNormal">@color/nomalColor</item>
  </style>
</resources>
```

- 해당 Component에 theme 속성 추가

```
<EditText
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:textSize="20dp"
  android:theme="@style/EditTheme"/>
```



# EditText의 Cursor 색상 변경하기

- EditText Cursor 색상 변경하는 방법
  - Drawable 디렉토리에 Cursor 색상을 지정한 drawable\_cursor.xml 파일을 생성

```
<?xml version="1.0" encoding="utf-8" ?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#FF33ff" />
    <size android:width="5dp" />
</shape>
```

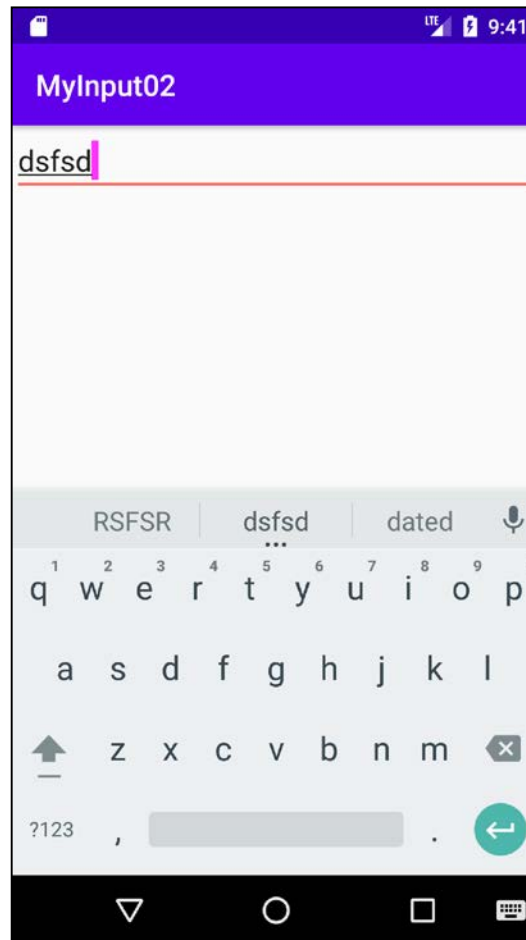
- 해당 Component에 textCursorDrawable 속성에 지정

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textCursorDrawable="@drawable/drawable_cursor"
    android:textSize="20dp"
    android:theme="@style/EditTheme" />
```



# EditText의 Cursor 색상 변경하기

## ■ 최종 결과





# Selector를 이용한 Button 만들기

- Android Studio는 프로젝트 생성시에
  - Theme.MaterialComponents.DayNights.DarkActionBar를 디폴트 테마로 설정되어 있음
  - 해당 테마에서 Button 생성시 MaterialButton을 생성
  - MaterialButton의 background는 자체적인 background를 가지고 있어, android:background 속성의 설정 값을 무시



# Selector를 이용한 Button 만들기

## ■ 2가지 해결 방법

- 단순 Button 대신 androidx.appcompat.widget.  
AppCompatButton을 사용

```
<androidx.appcompat.widget.AppCompatButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@drawable/clicked"  
    android:text="버튼 클릭" />
```

- res/values/themes/themes.xml에서 상단의  
parent="Theme.MaterialComponents.DayNight.DarkAct  
ionBar" 부분을  
parent="Theme.AppCompat.DayNight.DarkActionBar"  
로 변경



# Selector를 이용한 Button 만들기

## ■ selector 속성의 종류

### ■ android:state\_pressed

- View가 눌렸을 때 (터치나 클릭이 발생했을 때)

### ■ android:state\_focused

- View에 포커스가 위치했을 때  
(EditText를 입력할 수 있을 때)

### ■ android:state\_selected

- View를 선택했을 때  
(방향키로 이동하다가 선택했을 때)

### ■ android:state\_checkable

- 체크 가능한 상태일 때  
(체크 박스를 체크할 수 있는 상태일 때)

### ■ android:state\_checked

- 체크된 상태일 때 (체크박스가 체크된 상태일 때)

### ■ android:state\_enabled

- 사용할 수 있는 상태 일 때  
(터치나 클릭 이벤트 등을 받을 수 있는 상태일 때)



# Selector를 이용한 Button 만들기

- Selector를 이용한 Button 만드는 방법
  - selector 속성을 적용하기 위한 Button 만들기
    - /res/drawable 디렉터리에 위치한 drawable 리소스로 만들어야 함
    - 누르지 않았을 때의 Button 디자인
    - 눌렀을 때의 Button 디자인
    - <shape> 태그 사용
  - selector 작성
    - <selector> 태그 내부에 중첩 태그로 상태별 item 태그를 추가
    - <item> 태그의 drawable 속성에 상태별로 적용할 리소스(이미지, shape 등) 설정
    - selector는 /res/drawable 디렉터리에 위치한 drawable 리소스로 만들어야 함
  - Button의 android:drawable의 속성 값 지정



# Selector를 이용한 Button 만들기

- Button 디자인을 위한 <shape> 태그 사용방법
  - <shape> 태그를 사용하면, 선과 면이 있는 도형을 정의할 수 있음
  - <shape> 태그는 아래의 중첩 태그를 이용해 도형을 정의
    - solid: 도형의 배경색(면)
    - stroke: 선(테두리)
    - corners: 둥근 모서리(radius)
    - gradient: 그라데이션
    - padding: 패딩(내부 여백)
    - size: 크기(높이, 너비)





# Selector를 이용한 Button 만들기

## ■ <shape> 태그

```
<shape android:shape="rectangle | oval | line | ring" >
  <solid android:color="color" />
  <stroke
    android:width="integer"
    android:color="color"
    android:strokeWidth="integer"
    android:strokeDash="integer" />
  <corners
    android:radius="integer"
    android:topLeftRadius="integer"
    android:topRightRadius="integer"
    android:bottomLeftRadius="integer"
    android:bottomRightRadius="integer" />
```



# Selector를 이용한 Button 만들기

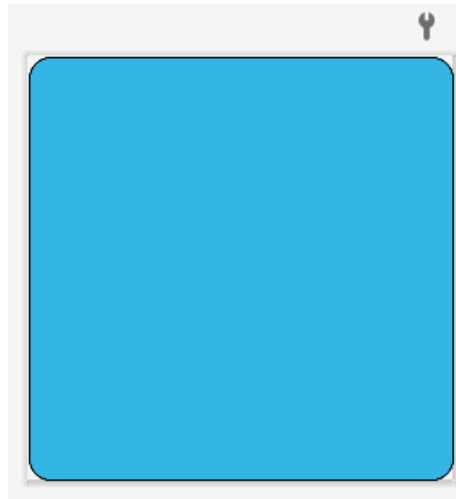
```
<gradient
    android:angle="integer"
    android:centerX="float"
    android:centerY="float"
    android:centerColor="integer"
    android:endColor="color"
    android:gradientRadius="integer"
    android:startColor="color"
    android:type="linear | radial | sweep"
    android:useLevel="true | false" />
<padding
    android:left="integer"
    android:top="integer"
    android:right="integer"
    android:bottom="integer" />
<size
    android:width="integer"
    android:height="integer" />
</shape>
```



# Selector를 이용한 Button 만들기

## ■ 누르지 않았을 때의 Button 디자인 (unclicked.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke
        android:width="2dp"
        android:color="@android:color/black" />
    <solid android:color="@android:color/holo_blue_light" />
    <corners android:radius="25dp" />
</shape>
```

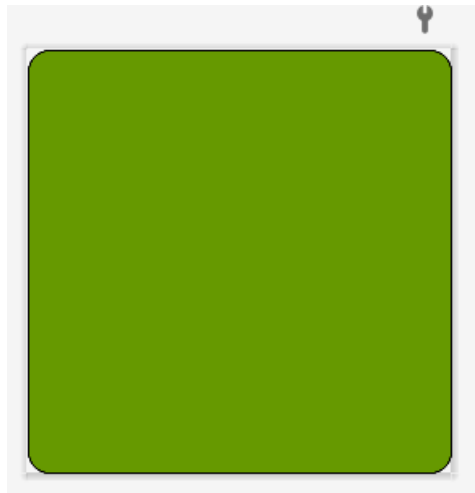




# Selector를 이용한 Button 만들기

## ■ 눌렀을 때의 Button 디자인 (clicked.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke
        android:width="2dp"
        android:color="@android:color/black" />
    <solid android:color="@android:color/holo_green_dark" />
    <corners android:radius="25dp" />
</shape>
```





# Selector를 이용한 Button 만들기

## ■ selector 작성 (button\_selector.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_enabled="false">
        <shape android:shape="rectangle">
            <corners android:radius="25dp" />
            <solid android:color="@color/white" />
            <stroke android:width="2dp" android:color="@color/purple_200" />
        </shape>
    </item>

    <item android:drawable="@drawable/unclicked"
        android:state_pressed="false" />

    <item android:drawable="@drawable/clicked"
        android:state_pressed="true" />
</selector>
```



# Selector를 이용한 Button 만들기

## ■ selector 작성

- 앞의 코드처럼 상태별 <item> 태그 내부의 중첩 태그로 shape를 생성해도 됨
- 별도로 XML 파일을 만들어 <shape>를 최상위 태그로 작성한 다음, selector 파일의 상태별 item 태그의 drawable 속성에 설정해도 됨
- 순서가 중요함



# Selector를 이용한 Button 만들기

## ■ Button의 android:drawable의 속성 값 지정

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">
```



# Selector를 이용한 Button 만들기

## ■ Button의 android:drawable의 속성 값 지정

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

```
    <Button
```

```
        android:id="@+id/button1"  
        android:layout_width="0dp"  
        android:layout_height="wrap_content"  
        android:layout_weight="1"  
        android:text="Enable OK Button" />
```

```
    <Button
```

```
        android:id="@+id/button2"  
        android:layout_width="0dp"  
        android:layout_height="wrap_content"  
        android:layout_weight="1"  
        android:text="Disable OK Button" />
```

```
</LinearLayout>
```





# Selector를 이용한 Button 만들기

## ■ Button의 android:drawable의 속성 값 지정

<Button

```
    android:id="@+id/okButton"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="20dp"  
    android:background="@drawable/button_selector"  
    android:text="OK Button" />
```

<TextView

```
    android:id="@+id/textView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="20dp"  
    android:gravity="center"  
    android:textSize="20dp" />
```

</LinearLayout>





# Selector를 이용한 Button 만들기

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        TextView textView = findViewById(R.id.textView);  
        final Button button1 = findViewById(R.id.okButton);  
  
        Button enableOkButton = findViewById(R.id.button1);  
        enableOkButton.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                button1.setEnabled(true);  
                textView.setText("Ok Button is enabled.");  
            }  
        });  
    }  
};
```



# Selector를 이용한 Button 만들기

## ■ MainActivity.JAVA

```
Button disableOkButton = findViewById(R.id.button2);
disableOkButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        button1.setEnabled(false);
        textView.setText("Ok Button is disabled.");
    }
});
```



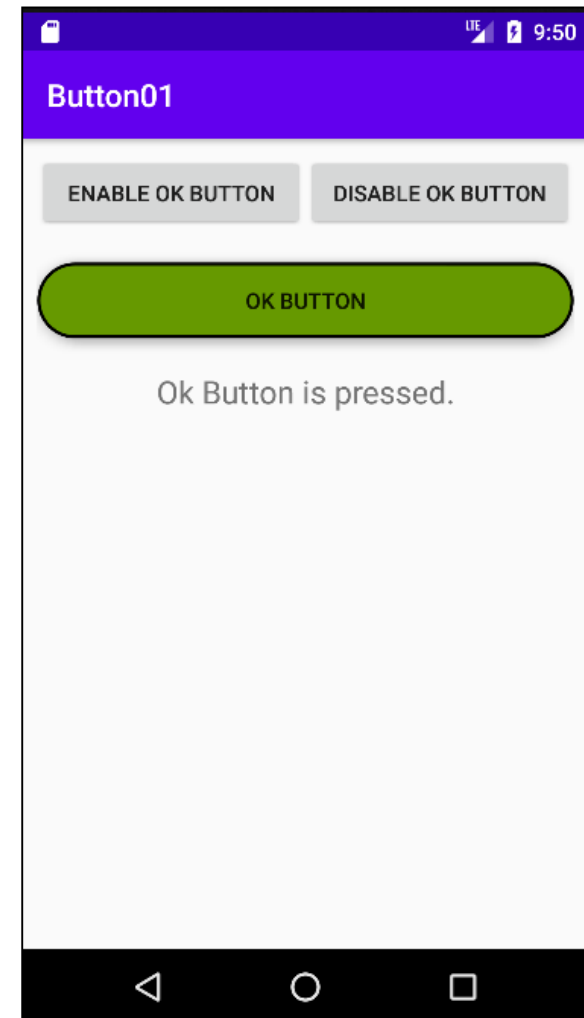
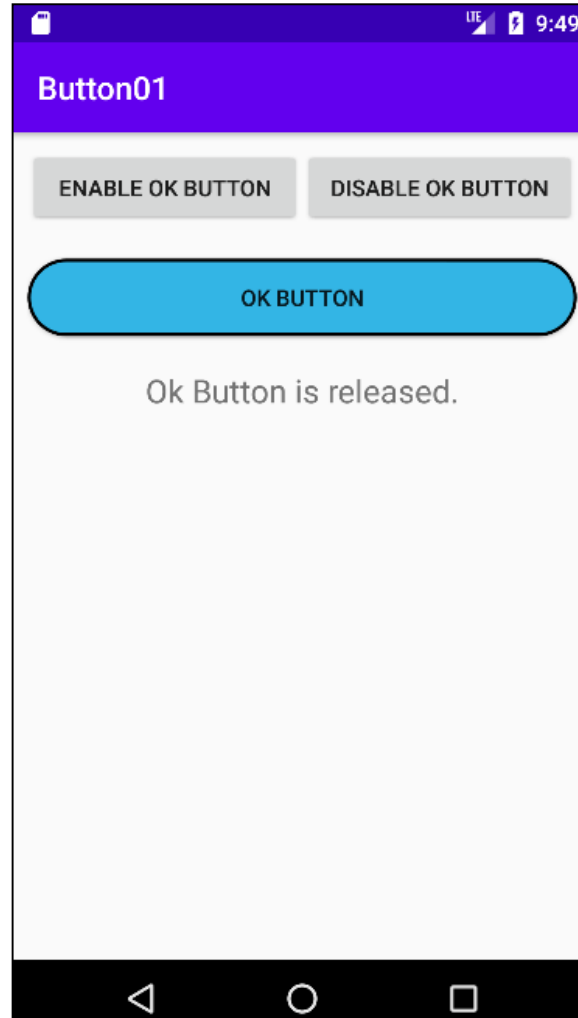
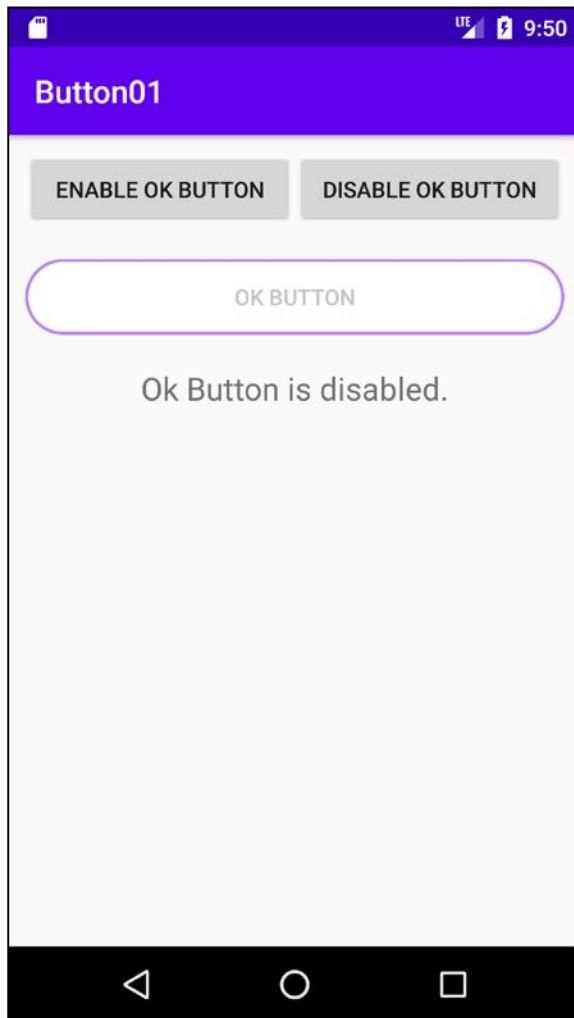
# Selector를 이용한 Button 만들기

## ■ MainActivity.JAVA

```
button1.setOnTouchListener(new View.OnTouchListener() {  
    @Override  
    public boolean onTouch(View view, MotionEvent motionEvent) {  
        int action = motionEvent.getAction();  
        if(action == MotionEvent.ACTION_DOWN) {  
            textView.setText("Ok Button is pressed.");  
        }else if(action == MotionEvent.ACTION_UP) {  
            textView.setText("Ok Button is released.");  
        }  
        return false;  
    }  
});  
}
```



# Selector를 이용한 Button 만들기





# Selector를 이용한 Button 만들기

## ■ 그라데이션 효과

- 위 button\_selcctor.xml 파일에서 android:color를 빼고, gradient 요소를 추가

```
<gradient  
  android:angle="0"  
  android:startColor="#FD8569"  
  android:endColor="#FD699C"  
  android:type="linear" />
```

