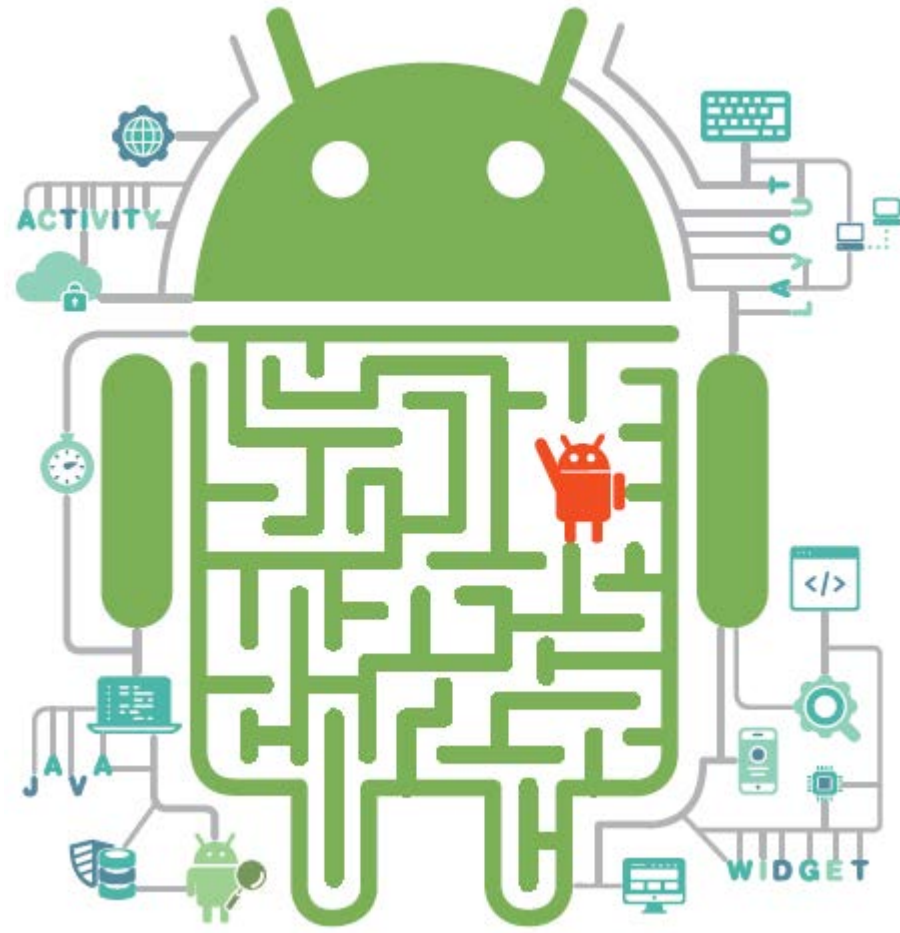


## 단계별로 배우는 안드로이드 프로그래밍

### [강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전재하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.



단계별로 배우는

# 안드로이드 프로그래밍

## Chapter 03. 위젯

# 목차

01 텍스트뷰

02 버튼

03 이미지뷰

04 프로그레스바

# 학습목표

- 위젯의 클래스 계층 관계를 이해한다.
- 텍스트뷰의 속성과 사용법을 익한다.
- 버튼의 속성과 사용법을 익한다.
- 이미지뷰의 속성과 사용법을 익한다.
- 프로그레스바의 속성과 사용법을 익한다.

## ■ 텍스트뷰

- 텍스트를 표시하는 기본 기능
- 속성을 추가하여 텍스트를 편집
- TextView: 자식 위젯에 공통 기능을 제공하므로 가장 먼저 학습
- CheckedTextView: 텍스트에 체크 표시 기능을 추가
- EditText: 텍스트를 편집하는 속성을 추가한 것
- Button과 그 이하 위젯: 서로 다른 특징을 가진 버튼 기능을 제공

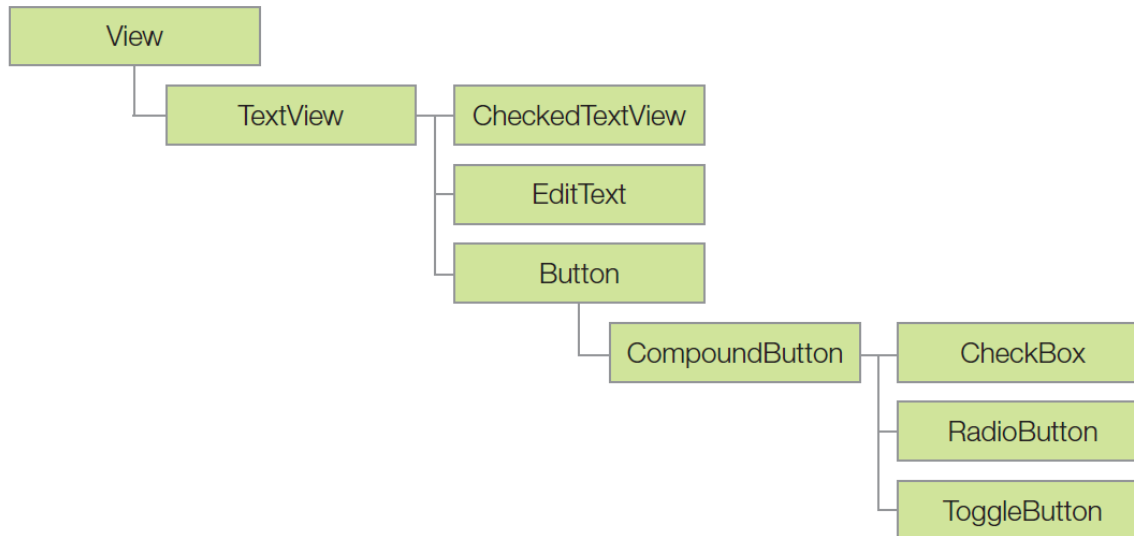


그림 3-1 텍스트뷰 관련 위젯들의 클래스 계층도

## ■ TextView 속성

- 2장에서 배운 View 클래스의 속성을 모두 가짐
- TextView만의 고유한 속성을 추가로 제공

표 3-1 TextView 속성 - 편집 기능 제외

XML 속성	관련 메서드	기능
android:text	setText()	표시할 텍스트를 지정한다.
android:textColor	setTextColor()	텍스트 색상을 지정한다.
android:textSize	setTextSize()	텍스트 크기를 지정한다.
android:typeface	setTypeface()	텍스트 서체를 "normal", "sans", "serif", "monospace" 중 하나로 지정한다.
android:textStyle		텍스트 스타일을 "bold", "italic", "bolditalic" 중 하나로 지정한다.
android:textAppearance	setTextAppearance()	텍스트의 기본 색상/크기/서체/스타일을 한꺼번에 지정한다.
android:letterSpacing	setLetterSpacing()	글자 간격을 실숫값(예: 0.1)으로 지정한다. 기본값은 "0"이며, 음수도 가능하다.
android:lineSpacingExtra	setLineSpacing()	줄 간격에 더할 값을 px/mm/in/pt/dp/sp 단위로 지정한다. 기본값은 "0dp"이며, 음수도 가능하다.
android:lineSpacingMultiplier		줄 간격에 곱할 값을 실수(예: 1.2)로 지정한다. 기본값은 "1"이며, 음수는 안 된다.
android:gravity	setGravity()	TextView 내부에 텍스트를 표시할 위치를 지정한다. 예를 들어 속성값으로 "right"를 주면 텍스트가 오른쪽에 정렬되어 표시된다.

# 01 텍스트뷰 ▶ TextView

android:autoLink	setAutoLinkMask()	텍스트에 포함된 웹 주소, 이메일 주소, 전화 번호, 거리 주소(미국식)가 자동으로 링크되어 동작한다. 클릭하면 웹사이트가 열리거나 이메일을 보내거나 전화를 걸거나 지도 앱이 실행된다.
android:singleLine	setSingleLine()	텍스트를 무조건 한 줄로 표시한다.
android:ellipsize	setEllipsize()	android:singleLine 속성을 준 상태에서 표시할 텍스트가 너무 길면 “...”을 표시한다. 속성값에 따라 앞(“start”), 중간(“middle”), 끝(“end”)에 “...”이 표시된다. “marquee” 값을 주면 전광판처럼 텍스트가 자동 스크롤된다.
android:drawableTop android:drawableBottom android:drawableLeft android:drawableRight	setCompoundDrawablesWithIntrinsicBounds()	텍스트의 위쪽/아래쪽/왼쪽/오른쪽에 그림을 표시한다.
android:drawablePadding	setCompoundDrawablePadding()	텍스트와 그림 사이에 여백을 준다. 기본값은 “0dp”이며, 음수도 가능하다.

# 01 텍스트뷰 ▶ TextView



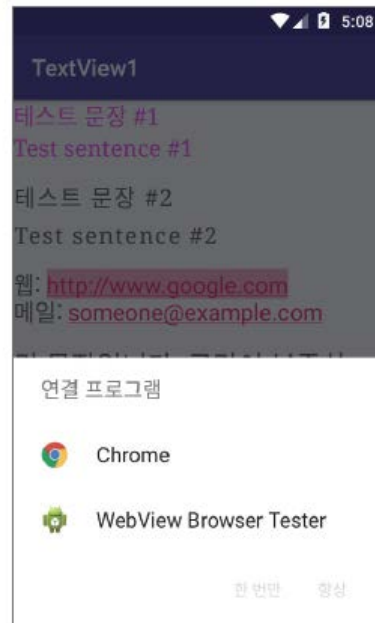
## 실습 3-1

## TextView1

### ■ TextView의 다양한 속성



(a) 초기 화면



(b) 웹 주소 클릭 시

그림 3-2 실행 화면



# 01 텍스트뷰 ▶ TextView



## 실습 3-1

## TextView1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <TextView
7          android:layout_width="wrap_content"
8          android:layout_height="wrap_content"
9          android:layout_marginBottom="16dp"
10         android:text="테스트 문장 #1\nTest sentence #1"
11         android:textColor="#ff00ff"
12         android:textSize="20sp"
13         android:typeface="serif"/>
14     <TextView
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:layout_marginBottom="16dp"
```

# 01 텍스트뷰 ▶ TextView



## 실습 3-1

## TextView1

```
18     android:letterSpacing="0.05"
19     android:lineSpacingMultiplier="1.2"
20     android:text="테스트 문장 #2\nTest sentence #2"
21     android:textSize="20sp"
22     android:typeface="serif"/>
23 <TextView
24     android:layout_width="wrap_content"
25     android:layout_height="wrap_content"
26     android:layout_marginBottom="16dp"
27     android:autoLink="web|email"
28     android:text="웹: http://www.google.com\n메일: someone@example.com"
29     android:textSize="20sp"/>
30 <TextView
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content"
33     android:layout_marginBottom="16dp"
34     android:ellipsize="end"
35     android:singleLine="true"
36     android:text="긴 문장입니다. 공간이 부족하면 생략 표시가 되는지 확인합니다."
37     android:textSize="24sp"/>
38 <TextView
39     android:layout_width="wrap_content"
40     android:layout_height="wrap_content"
41     android:drawablePadding="4dp"
42     android:drawableTop="@mipmap/ic_launcher"
43     android:text="앱 아이콘입니다."/>
44 </LinearLayout>
```

## ■ CheckedTextView

- TextView에 체크 표시 기능을 추가한 것

## ■ 주요 속성

표 3-2 CheckedTextView 속성

XML 속성	관련 메서드	기능
android:checked	setChecked()	값이 "true"이면 체크 상태로 시작한다. 기본값은 "false"이다.
android:checkMark	setCheckMarkDrawable()	체크 표시에 사용할 그림을 지정한다.

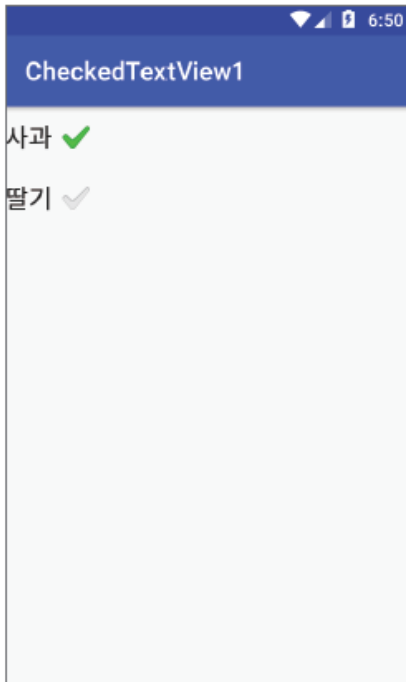
# 01 텍스트뷰 ▶ CheckedTextView



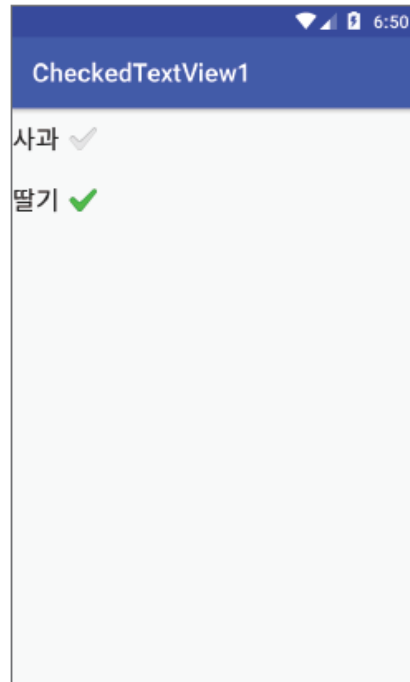
실습 3-2

CheckedTextView1

## ■ CheckedTextView를 이용한 출력과 입력 방법



(a) 초기 화면



(b) 사과와 딸기를 한 번씩 클릭

그림 3-3 실행 화면

# 01 텍스트뷰 ► CheckedTextView



## 실습 3-2

## CheckedTextView1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <CheckedTextView
7          android:layout_width="wrap_content"
8          android:layout_height="wrap_content"
9          android:checkMark="?android:attr/textCheckMark"
10         android:checked="true"
11         android:gravity="center_vertical"
12         android:onClick="mOnClick"
13         android:text="사과"
14         android:textSize="20dp"/>
15     <CheckedTextView
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:checkMark="?android:attr/textCheckMark"
19         android:gravity="center_vertical"
20         android:onClick="mOnClick"
21         android:text="딸기"
22         android:textSize="20dp"/>
23 </LinearLayout>
```

# 01 텍스트뷰 ▶ CheckedTextView



실습 3-2

CheckedTextView1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {  
2  
3      @Override  
4      protected void onCreate(Bundle savedInstanceState) {  
5          super.onCreate(savedInstanceState);  
6          setContentView(R.layout.activity_main);  
7      }  
8  
9      public void mOnClick(View v) {  
10         CheckedTextView ctv = (CheckedTextView) v;  
11         ctv.setChecked(!ctv.isChecked());  
12         // ctv.toggle();  
13     }  
14 }
```

## ■ EditText

- 텍스트를 편집할 수 있도록 TextView의 기본 속성을 변경한 것
- 사실상 속성만 다른 TextView

## ■ TextView의 속성 중 편집 기능과 연관되어 사용되는 것

표 3-3 TextView(= EditText) 속성 - 편집 기능 관련

XML 속성	관련 메서드	기능
android:hint	setHint()	텍스트가 비어 있을 때 대신 표시할 텍스트를 지정한다.
android:ems	setEms()	TextView의 폭을 지정한다. em은 M을 발음대로 쓴 것인데 대략 글자 한 개의 폭을 의미한다. 예를 들어 android:ems="10"이면 글자 열 개의 폭이다.
android:lines	setLines()	TextView의 높이를 줄 수로 지정한다.
android:maxLength	setFilters()	입력 가능한 최대 글자 수를 지정한다.
android:selectAllOnFocus	setSelectAllOnFocus()	속성값이 "true"이면 TextView를 선택 시 텍스트 전체가 선택되어 하이라이트 표시된다.
android:digits	setKeyListener()	지정된 숫자만 입력받는다. 예를 들어 "12345"를 속성값으로 주면 1,2,3,4,5 숫자만 입력할 수 있다.
android:inputType	setRawInputType() 또는 setInputType()	입력 텍스트의 형식을 지정한다. 예를 들어 이메일 주소나 숫자만 입력할 수 있도록 제한할 수 있다. [그림 3-4]에서 보듯이 안드로이드 스튜디오의 Design 탭에서 Text를 선택하면 inputType만 다른 다양한 EditText를 볼 수 있다.



# 01 텍스트뷰 ▶ EditText

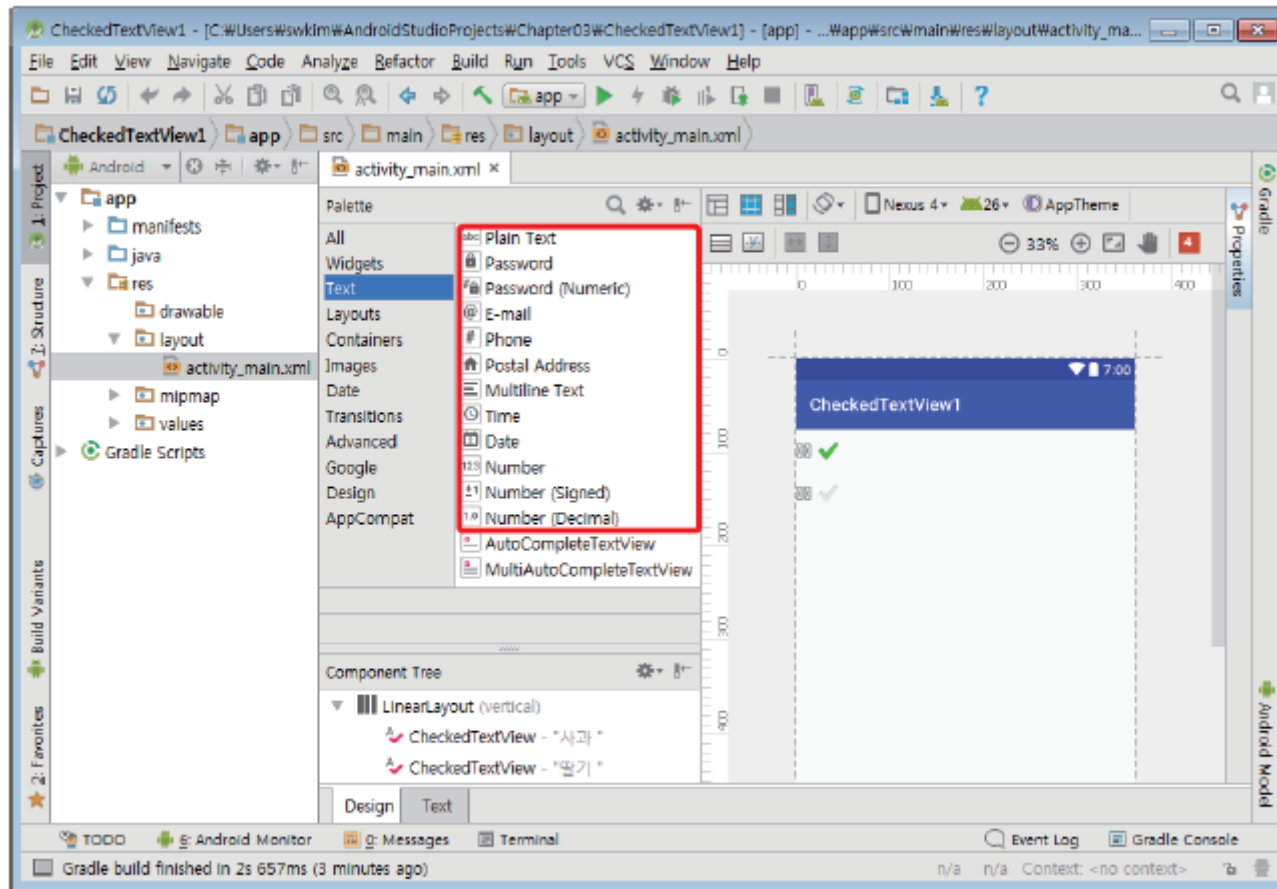


그림 3-4 inputType 속성에 따른 다양한 EditText

## ■ requestFocus

- XML 엘리먼트로, EditText와 더불어 유용하게 사용
- 여러 개의 EditText 중 키 입력을 최우선으로 받고 싶은 것이 있다면 requestFocus

```
<EditText  
    android:id="@+id/editName"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
    <requestFocus/>  
</EditText>
```

- EditText에 사용자가 입력한 내용은 자바 코드에서 getText ( ) 메서드로 획득
- 내용 변경 시 setText ( ) 메서드를 사용

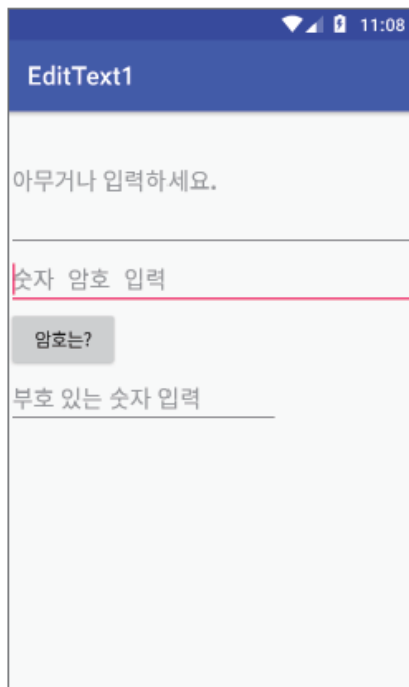
# 01 텍스트뷰 ▶ EditText



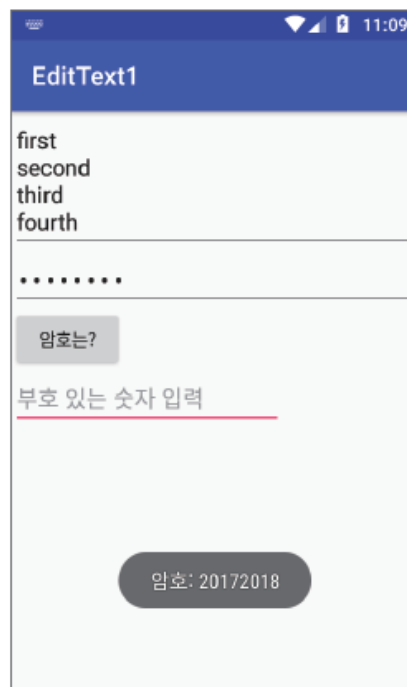
## 실습 3-3

## EditText1

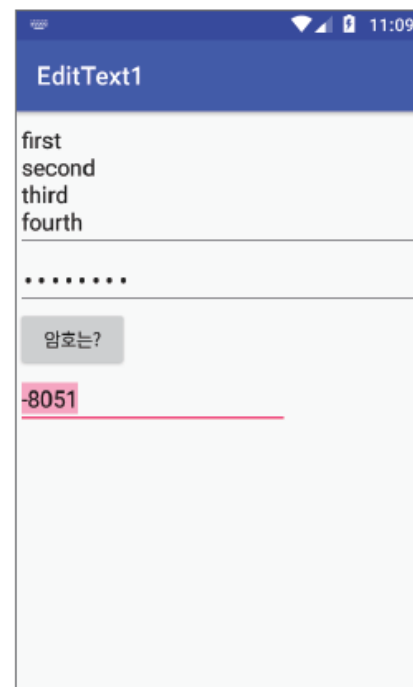
### ■ EditText (= TextView)의 다양한 속성



(a) 초기 화면



(b) 텍스트 입력 테스트 (1)



(c) 텍스트 입력 테스트 (2)

그림 3-5 실행 화면

# 01 텍스트뷰 ▶ EditText



## 실습 3-3

## EditText1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <EditText
7          android:layout_width="match_parent"
8          android:layout_height="wrap_content"
9          android:hint="아무거나 입력하세요."
10         android:lines="4"/>
11     <EditText
12         android:id="@+id/editPassword"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:hint="숫자 암호 입력"
16         android:inputType="numberPassword"/>
```

# 01 텍스트뷰 ▶ EditText



## 실습 3-3

### EditText1

```
17         android:maxLength="10">
18         <requestFocus/>
19     </EditText>
20     <Button
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:onClick="mOnClick"
24         android:text="암호는?"/>
25     <EditText
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28         android:ems="10"
29         android:hint="부호 있는 숫자 입력"
30         android:inputType="numberSigned"
31         android:selectAllOnFocus="true"/>
32 </LinearLayout>
```

# 01 텍스트뷰 ▶ EditText



## 실습 3-3

## EditText1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      private EditText mEditPassword;
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_main);
9          mEditPassword = (EditText) findViewById(R.id.editPassword);
10     }
11
12     public void mOnClick(View v) {
13         String password = mEditPassword.getText().toString();
14         if (password.length() > 0) {
15             Toast.makeText(this, "암호: " + password, Toast.LENGTH_SHORT).show();
16         } else {
17             Toast.makeText(this, "암호를 먼저 입력하세요!", Toast.LENGTH_SHORT).show();
18         }
19     }
20 }
```

### ■ 버튼

- 텍스트뷰의 일종으로 출력보다는 입력을 위해 주로 사용
- Button 클래스: TextView에 추가 속성이 없는 거의 동일한 클래스
- CompoundButton 클래스: 추상 클래스로서 체크와 언체크 두 가지 상태를 추가한 것
- CompoundButton의 서브클래스들: 모양만 조금 다를 뿐 추가 속성은 없다.

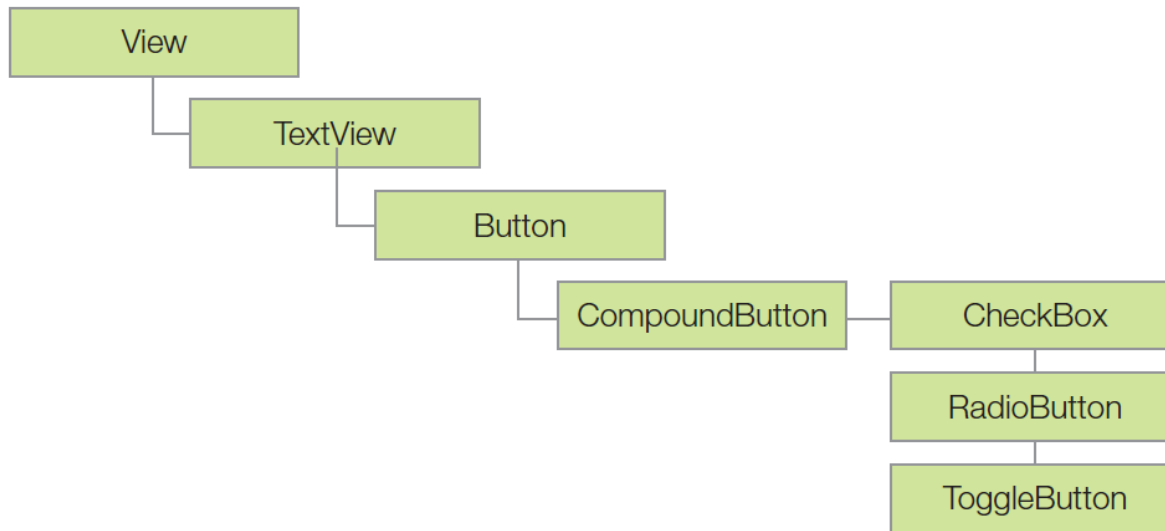


그림 3-6 텍스트뷰와 버튼 위젯들의 클래스 계층도

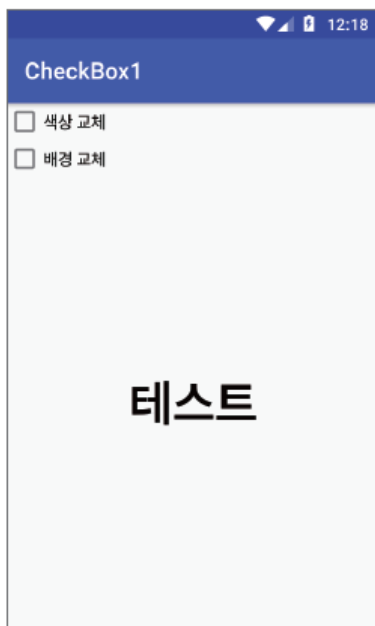
## 02 버튼 ▶ CheckBox



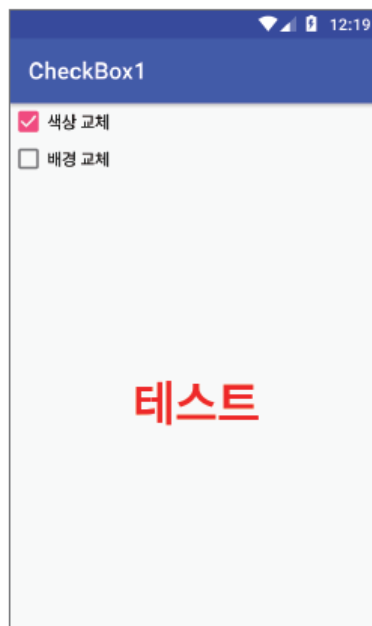
실습 3-4

CheckBox1

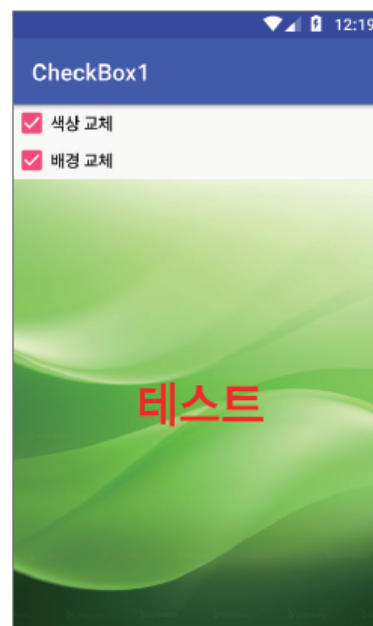
- 두 개의 CheckBox를 이용해서 텍스트의 색상이나 배경을 변경하기



(a) 초기 화면



(b) 첫 번째 CheckBox만 체크



(c) 두 번째 CheckBox도 체크

그림 3-7 실행 화면



## 02 버튼 ▶ CheckBox



실습 3-3

EditText1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <CheckBox
7          android:id="@+id/checkColor"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:onClick="mOnClick"
11         android:text="색상 교체"/>
12     <CheckBox
13         android:id="@+id/checkBackground"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
```

## 02 버튼 ▶ CheckBox



실습 3-3

EditText1

```
16         android:onClick="mOnClick"
17         android:text="배경 교체"/>
18     <TextView
19         android:id="@+id/textTest"
20         android:layout_width="match_parent"
21         android:layout_height="match_parent"
22         android:gravity="center"
23         android:text="테스트"
24         android:textColor="#000000"
25         android:textSize="40sp"
26         android:textStyle="bold"/>
27 </LinearLayout>
```

## 02 버튼 ▶ CheckBox



### 실습 3-3

### EditText1

MainActivity.java

```
01  public class MainActivity extends AppCompatActivity {
02
03      private CheckBox mCheckColor, mCheckBackground;
04      private TextView mTextTest;
05
06      @Override
07      protected void onCreate(Bundle savedInstanceState) {
08          super.onCreate(savedInstanceState);
09          setContentView(R.layout.activity_main);
10
11          mCheckColor = (CheckBox) findViewById(R.id.checkColor);
12          mCheckBackground = (CheckBox) findViewById(R.id.checkBackground);
13          mTextTest = (TextView) findViewById(R.id.textTest);
14      }
15
16      public void mOnClick(View v) {
17          switch (v.getId()) {
18              case R.id.checkColor:
19                  if (mCheckColor.isChecked())
20                      mTextTest.setTextColor(0xffff0000);
21                  else
22                      mTextTest.setTextColor(0xff000000);
23                  break;
```

## 02 버튼 ▶ CheckBox



실습 3-3

EditText1

```
24         case R.id.checkBackground:  
25             if (mCheckBackground.isChecked())  
26                 mTextTest.setBackgroundResource(R.drawable.testpic);  
27             else  
28                 mTextTest.setBackgroundResource(0);  
29             break;  
30         }  
31     }  
32 }
```

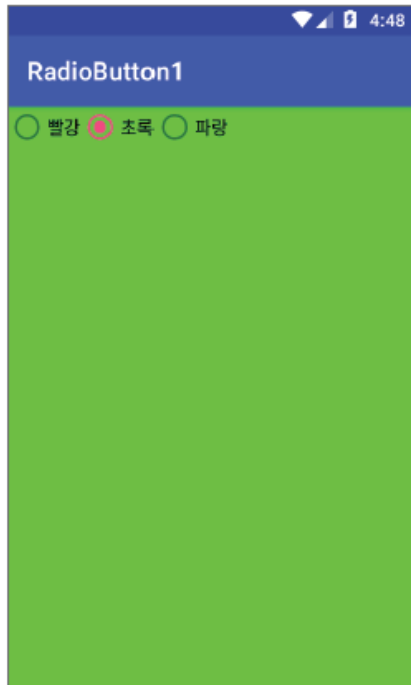
## 02 버튼 ▶ RadioButton



실습 3-5

RadioButton1

- 라디오 그룹에 여러 개의 RadioButton을 묶어서 사용



(a) 초기 화면



(b) '파랑'을 클릭하여 선택

그림 3-8 실행 화면

## 02 버튼 ▶ RadioButton



실습 3-5

RadioButton1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:id="@+id/layoutMain"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical">
7      <RadioGroup
8          android:id="@+id/groupColor"
9          android:layout_width="wrap_content"
10         android:layout_height="wrap_content"
11         android:orientation="horizontal">
12         <RadioButton
13             android:id="@+id/radioRed"
14             android:layout_width="wrap_content"
15             android:layout_height="wrap_content"
```

## 02 버튼 ▶ RadioButton



실습 3-5

RadioButton1

```
16         android:onClick="mOnClick"
17         android:text="빨강"/>
18     <RadioButton
19         android:id="@+id/radioGreen"
20         android:layout_width="wrap_content"
21         android:layout_height="wrap_content"
22         android:onClick="mOnClick"
23         android:text="초록"/>
24     <RadioButton
25         android:id="@+id/radioBlue"
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28         android:onClick="mOnClick"
29         android:text="파랑"/>
30 </RadioGroup>
31 </LinearLayout>
```

## 02 버튼 ▶ RadioButton



실습 3-5

RadioButton1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      private LinearLayout mLayoutMain;
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_main);
9          mLayoutMain = (LinearLayout) findViewById(R.id.layoutMain);
10
11         int id = R.id.radioGreen; // 초록을 기본값으로 선택한다.
12         RadioGroup ColorGroup = (RadioGroup) findViewById(R.id.groupColor);
13         ColorGroup.check(id); // 라디오 그룹에서 초록을 선택한다.
14         changeColor(id); // 초록을 클릭한 것처럼 처리한다.
15     }
16
```



## 02 버튼 ▶ RadioButton



### 실습 3-5

### RadioButton1

```
17     public void mOnClick(View v) {
18         changeColor(v.getId());
19     }
20
21     private void changeColor(int id) {
22         switch (id) {
23             case R.id.radioRed:
24                 mLayoutMain.setBackgroundColor(0xffff0000);
25                 break;
26             case R.id.radioGreen:
27                 mLayoutMain.setBackgroundColor(0xff00ff00);
28                 break;
29             case R.id.radioBlue:
30                 mLayoutMain.setBackgroundColor(0xff0000ff);
31                 break;
32         }
33     }
34 }
```

## 02 버튼 ▶ ToggleButton



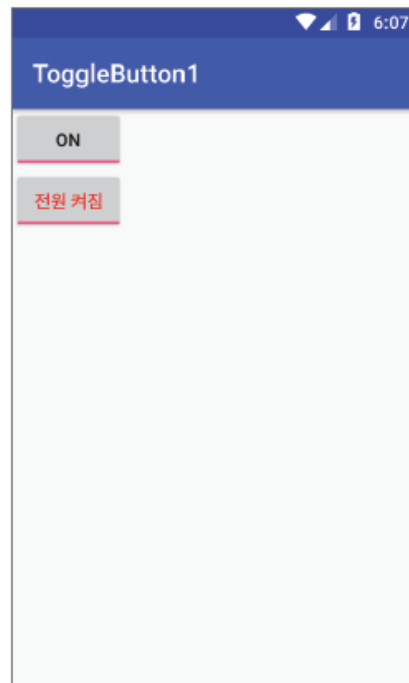
실습 3-6

ToggleButton1

- ToggleButton은 On/Off 두 가지 상태를 가지는 버튼



(a) 초기 화면



(b) 두 ToggleButton 클릭 후

그림 3-9 실행 화면

## 02 버튼 ▶ ToggleButton



실습 3-6

ToggleButton1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <ToggleButton
7          android:layout_width="wrap_content"
8          android:layout_height="wrap_content"/>
9      <ToggleButton
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:onClick="mOnClick"
13         android:textColor="#0000ff"
14         android:textOff="전원 꺼짐"
15         android:textOn="전원 켜짐"/>
16  </LinearLayout>
```

## 02 버튼 ▶ ToggleButton



실습 3-6

ToggleButton1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      @Override
4      protected void onCreate(Bundle savedInstanceState) {
5          super.onCreate(savedInstanceState);
6          setContentView(R.layout.activity_main);
7      }
8
9      public void mOnClick(View v) {
10         ToggleButton btn = (ToggleButton) v;
11         if (btn.isChecked())
12             btn.setTextColor(0xffff0000);
13         else
14             btn.setTextColor(0xff0000ff);
15     }
16 }
```

### ■ 이미지뷰

- 다양한 형식의 이미지를 화면에 출력하는 위젯
- ImageView 클래스: 이미지 출력에 필요한 공통의 속성과 기능을 모아놓은 것
- ImageButton 클래스: 이미지가 버튼처럼 동작하는 기능을 추가한 것

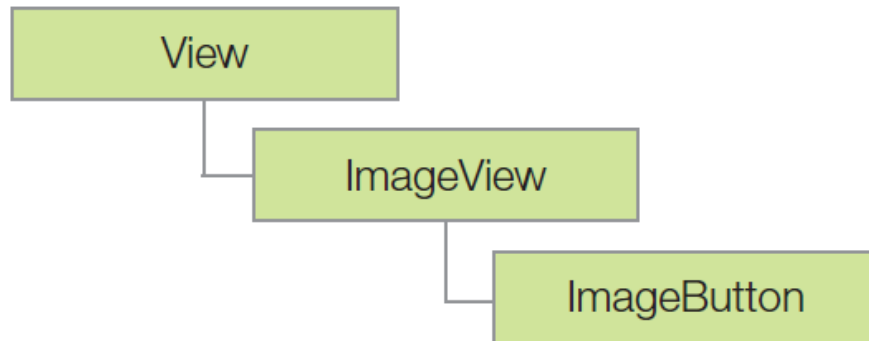


그림 3-10 이미지뷰 위젯들의 클래스 계층도

### ■ ImageView

- 리소스, 로컬 파일, Bitmap, Drawable을 출력할 수 있는 다재다능한 클래스이다.

### ■ 레이아웃 디자인에 자주 사용되는 순서의 주요 속성

표 3-4 ImageView 속성

XML 속성	관련 메서드	기능
android:src	setImageResource()	ImageView에 출력할 리소스를 지정한다.
android:scaleType	setScaleType()	ImageView에 출력할 이미지의 크기와 출력 위치를 제어한다. 상세 사항은 아래에서 추가 설명한다.
android:adjustViewBounds	setAdjustViewBounds()	속성값이 "true"이면 이미지의 종횡비에 맞춰 ImageView의 크기가 자동으로 조정된다. 예를 들어 이미지가 3:2 비율이면 이미지뷰도 3:2 비율이 된다. 기본값은 "false"이다.
android:maxWidth	setMaxWidth()	ImageView의 최대 폭을 지정한다.
android:maxHeight	setMaxHeight()	ImageView의 최대 높이를 지정한다.
android:cropToPadding	setCropToPadding()	속성값이 "true"이면 이미지의 출력이 항상 패딩 영역 내부로 제한된다. 속성값이 기본값인 "false"여도 이미지의 출력이 패딩 영역 내부로 여전히 제한되는데, 이미지를 원본 크기(1:1)로 출력할 경우에 한해 패딩 영역 내부로 제한되지 않고 출력된다.

### ■ 이미지뷰가 액티비티 화면 전체를 차지

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <ImageView xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:scaleType="속성값"
6      android:src="@drawable/testpic"/>
```

## 03 이미지뷰 ▶ ImageView

### ■ scaleType 속성값에 따른 ScaleType 예제의 실행 화면



(a) matrix



(b) fitXY



(c) fitStart

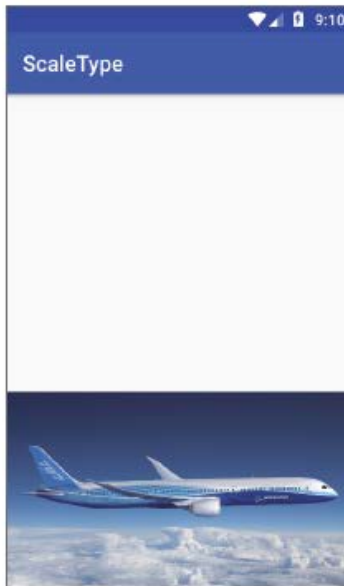


(d) fitCenter



## 03 이미지뷰 ▶ ImageView

### ■ scaleType 속성값에 따른 ScaleType 예제의 실행 화면



(e) fitEnd



(f) center



(g) centerCrop

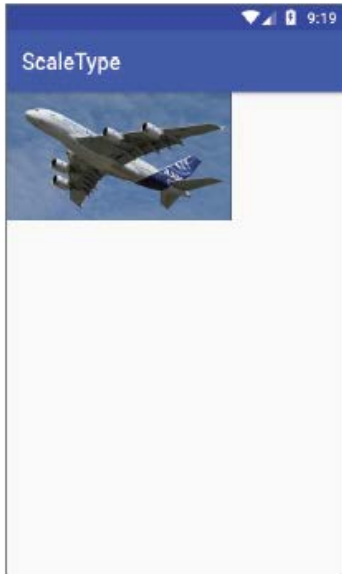


(h) centerInside

그림 3-11 scaleType 속성값에 따른 이미지 출력 결과(이미지가 큰 경우)

## 03 이미지뷰 ▶ ImageView

### ■ scaleType 속성값에 따른 ScaleType 예제의 실행 화면



(a) matrix



(b) fitXY



(c) fitStart



(d) fitCenter

## 03 이미지뷰 ▶ ImageView

### ■ scaleType 속성값에 따른 ScaleType 예제의 실행 화면



(e) fitEnd



(f) center



(g) centerCrop



(h) centerInside

그림 3-12 scaleType 속성값에 따른 이미지 출력 결과(이미지가 작은 경우)

### ■ scaleType 속성값

표 3-5 scaleType 속성값에 따른 이미지 출력 방식

scaleType 속성값	기능
matrix	뷰의 좌상단을 기준으로 출력하되 setImageMatrix() 메서드로 행렬(matrix)을 조작하지 않으면 1:1, 즉 원본 그대로 출력한다.
fitXY	이미지의 종횡비를 유지하지 않으면서 뷰 화면을 가득 채운다.
fitStart	뷰의 좌상단을 기준으로 출력하되 이미지의 종횡비를 유지하면서 가득 채운다.
fitCenter	뷰의 정중앙을 기준으로 출력하되 이미지의 종횡비를 유지하면서 가득 채운다.
fitEnd	뷰의 우하단을 기준으로 출력하되 이미지의 종횡비를 유지하면서 가득 채운다.
center	이미지가 뷰의 정중앙에 위치하되 1:1, 즉 원본 그대로 출력한다.
centerCrop	이미지가 뷰의 정중앙에 위치하되 이미지의 폭과 높이가 뷰의 폭과 높이보다 크거나 같도록 (필요시) 크기를 조정한다. 이때 이미지의 종횡비는 유지한다. <b>참고</b> ▶ 이미지가 뷰보다 크면 어느 한쪽 면이 뷰에 닿을 때까지 이미지를 줄이고, 이미지가 뷰보다 작으면 어느 한쪽 면이 뷰에 닿을 때까지 이미지를 키운다.
centerInside	이미지가 뷰의 정중앙에 위치하되 이미지의 폭과 높이가 뷰의 폭과 높이보다 작거나 같도록 (필요시) 크기를 조정한다. 이때 이미지의 종횡비는 유지한다. <b>참고</b> ▶ 이미지가 뷰보다 크면 모든 면이 뷰의 내부에 들어올 때까지 이미지를 줄이고, 이미지가 뷰보다 작으면 그대로 둔다.

## 03 이미지뷰 ▶ ImageView



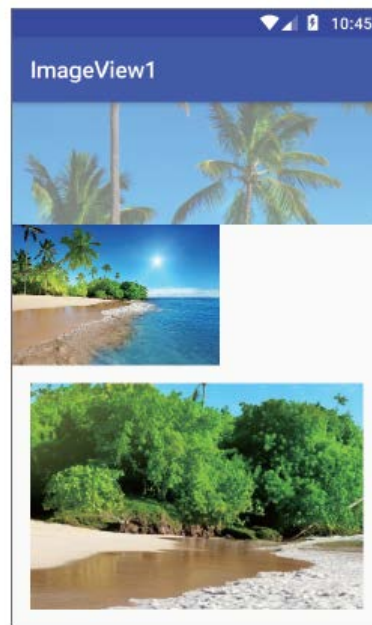
실습 3-7

ImageView1

### ■ 실행 화면



(a) 초기 화면



(b) 아래로 스크롤한 후

그림 3-13 실행 화면

## 03 이미지뷰 ▶ ImageView



실습 3-7

ImageView1

- 1장의 Hello 예제처럼 프로젝트를 생성하되 이름은 ImageView1
- 사용할 그림 준비한 후, 탐색기에서 Ctrl+C를 눌러 복사



(a) 준비한 그림 파일 ("testpic.jpg")

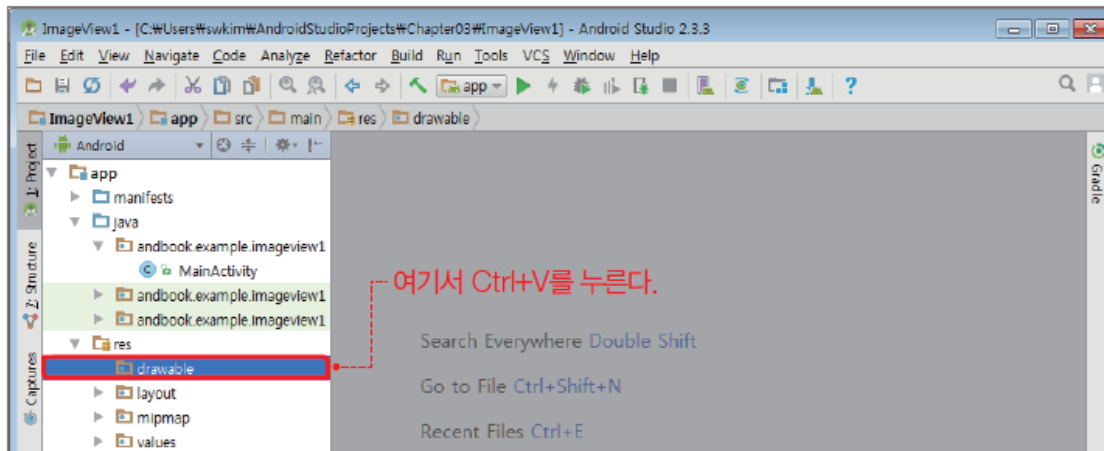
## 03 이미지뷰 ▶ ImageView



실습 3-7

ImageView1

- 좌측 프로젝트 창의 res/drawable에서 Ctrl+V 클릭



(b) 그림 파일 붙여넣기

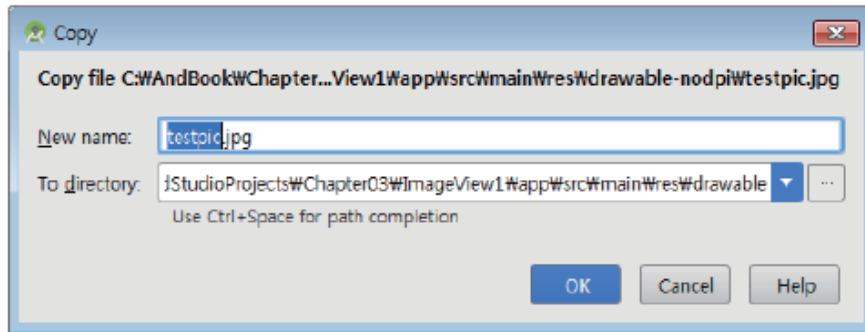
## 03 이미지뷰 ▶ ImageView



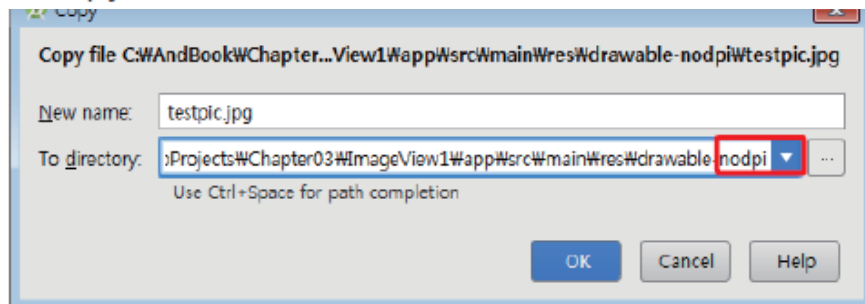
실습 3-7

ImageView1

### ■ 대화상자가 열림



(c) Copy 대화상자 (초기 상태)



(d) Copy 대화상자 (폴더 이름 끝에 - nodpi 추가)

그림 3-14 drawable 리소스 준비



## 03 이미지뷰 ▶ ImageView



실습 3-7

ImageView1

### ■ res/layout/activity\_main.xml 수정

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5      <LinearLayout
6          android:layout_width="match_parent"
7          android:layout_height="wrap_content"
8          android:orientation="vertical">
9          <ImageView
10             android:id="@+id/imageTest"
11             android:layout_width="wrap_content"
12             android:layout_height="wrap_content"
13             android:adjustViewBounds="true"
```

## 03 이미지뷰 ▶ ImageView



실습 3-7

ImageView1

```
14         android:scaleType="matrix"
15         android:src="@drawable/testpic"/>
16     <ImageView
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:adjustViewBounds="true"
20         android:maxLength="180dp"
21         android:scaleType="fitXY"
22         android:src="@drawable/testpic"/>
23     <ImageView
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:adjustViewBounds="true"
27         android:cropToPadding="true"
28         android:padding="16dp"
29         android:scaleType="center"
30         android:src="@drawable/testpic"/>
31 </LinearLayout>
32 </ScrollView>
```

## 03 이미지뷰 ▶ ImageView



실습 3-7

ImageView1

### ■ MainActivity 클래스에 다음 코드 작성

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {  
2  
3      private ImageView mImageTest;  
4  
5      @Override  
6      protected void onCreate(Bundle savedInstanceState) {  
7          super.onCreate(savedInstanceState);  
8          setContentView(R.layout.activity_main);  
9  
10         mImageTest = (ImageView) findViewById(R.id.imageTest);  
11         mImageTest.setAlpha(128); /* 안드로이드 1.0부터 사용 가능 */  
12         //mImageTest.setImageAlpha(128); /* 안드로이드 4.1부터 사용 가능 */  
13     }  
14 }
```

### ■ ImageButton

- 버튼처럼 누르고 떼는 효과를 ImageView에 추가한 것
- 추가 속성은 없으며 ImageView처럼 src 속성으로 출력할 대상을 지정

## 03 이미지뷰 ▶ ImageButton



실습 3-8

ImageButton1

### ■ 실행 화면

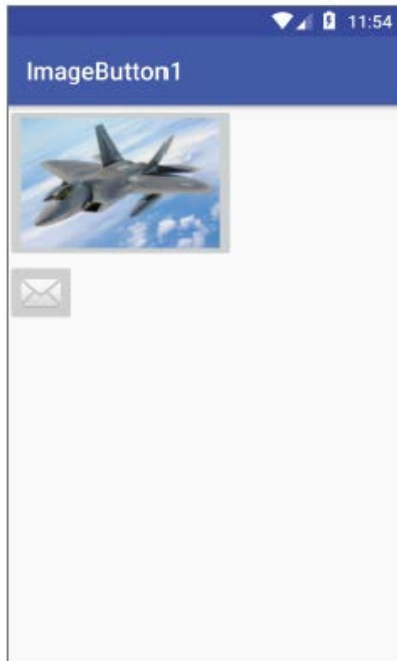


그림 3-15 실행 화면

## 03 이미지뷰 ▶ ImageButton



실습 3-8

ImageButton1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <ImageButton
7          android:layout_width="wrap_content"
8          android:layout_height="wrap_content"
9          android:src="@drawable/testpic"/>
10     <ImageButton
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:src="@android:drawable/ic_dialog_email"/>
14 </LinearLayout>
```

### ■ 프로그레스바

- 작업의 진행 상태를 표시하는 것이 주 기능
- 대표 클래스인 ProgressBar는 주로 출력용으로 사용
- 추상 클래스인 AbsSeekBar에서 파생된 RatingBar와 SeekBar 클래스는 주로 입력용

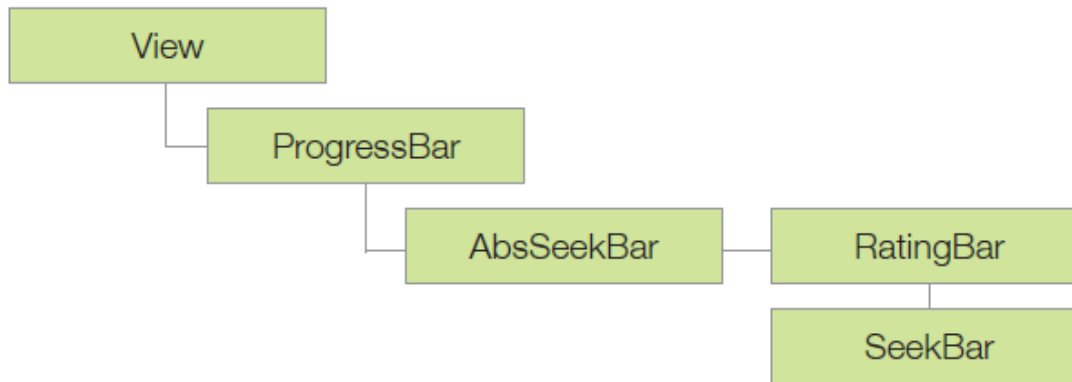


그림 3-16 프로그레스바 위젯들의 클래스 계층도

### ■ ProgressBar

- 작업의 진행 상황을 회전 바퀴나 수평 막대로 보여줌
- 확정(indeterminate) 모드와 불확정(indeterminate) 모드가 있음
- 불확정 모드에서는 진행값을 보여주지 않고 애니메이션으로만 표시
- 수평 막대는 확정 모드에서 최대 두 개의 진행값을 보여줌
- 각각 기본 진행값(default progress value)과 2차 진행값(secondary progress value)이라 부름



(a) 회전 바퀴: 확정/불확정 모드 동일



(b) 수평 막대: 확정 모드  
(기본 진행값을 50%, 2차 진행값을  
75%로 설정한 예)



(c) 수평 막대: 불확정 모드

그림 3-17 ProgressBar 형태



### ■ ProgressBar의 주요 속성

표 3-6 ProgressBar 속성

XML 속성	관련 메서드	기능
android:max	setMax()	최대 진행값을 지정한다. 예를 들어 값이 "255"면 진행값은 0~255 범위가 된다.
android:progress	setProgress()	기본 진행값을 지정한다.
android:secondaryProgress	setSecondaryProgress()	2차 진행값을 지정한다.
android:indeterminate	setIndeterminate()	값이 "true"이면 불확정 모드가 된다.

## 04 프로그레스바 ▶ ProgressBar



실습 3-9

ProgressBar1

### ■ ProgressBar를 확정 모드로 사용하면서 진행값을 변경하는 방법

- 좌측의 [-/+ ] 버튼은 기본 진행값을 증감
- 우측의 [-/+ ] 버튼은 2 차 진행값을 증감



activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6     <ProgressBar
7         android:id="@+id/progStatus"
```

그림 3-18 실행 화면

## 04 프로그레스바 ▶ ProgressBar



실습 3-9

ProgressBar1

```
8         style="?android:attr/progressBarStyleHorizontal"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:max="120"
12        android:progress="60"
13        android:secondaryProgress="90"/>
14    <LinearLayout
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content">
17        <Button
18            android:id="@+id/btnDecrease1"
19            android:layout_width="60dp"
20            android:layout_height="wrap_content"
21            android:onClick="mOnClick"
22            android:text="-"/>
23        <Button
24            android:id="@+id/btnIncrease1"
25            android:layout_width="60dp"
26            android:layout_height="wrap_content"
```

## 04 프로그레스바 ▶ ProgressBar



실습 3-9

ProgressBar1

```
27         android:onClick="mOnClick"
28         android:text="+"/>
29     <Button
30         android:id="@+id/btnDecrease2"
31         android:layout_width="60dp"
32         android:layout_height="wrap_content"
33         android:onClick="mOnClick"
34         android:text="-"
35         android:textColor="#0000ff"/>
36     <Button
37         android:id="@+id/btnIncrease2"
38         android:layout_width="60dp"
39         android:layout_height="wrap_content"
40         android:onClick="mOnClick"
41         android:text="+"
42         android:textColor="#0000ff"/>
43     </LinearLayout>
44 </LinearLayout>
```

## 04 프로그레스바 ▶ ProgressBar



실습 3-9

ProgressBar1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      private ProgressBar mProgStatus;
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_main);
9          mProgStatus = (ProgressBar) findViewById(R.id.progStatus);
10     }
11
12     public void mOnClick(View v) {
13         switch (v.getId()) {
14             case R.id.btnDecrease1:
15                 mProgStatus.setProgress(mProgStatus.getProgress() - 5);
16                 break;
17             case R.id.btnIncrease1:
18                 mProgStatus.setProgress(mProgStatus.getProgress() + 5);
19                 break;
20             case R.id.btnDecrease2:
21                 mProgStatus.setSecondaryProgress(mProgStatus.getSecondaryProgress() - 5);
22                 break;
23             case R.id.btnIncrease2:
24                 mProgStatus.setSecondaryProgress(mProgStatus.getSecondaryProgress() + 5);
25                 break;
26         }
27     }
28 }
```

- RatingBar는 별의 개수로 평점을 입력받을 때 사용
- 주요 속성

표 3-7 RatingBar 속성

XML 속성	관련 메서드	기능
android:numStars	setNumStars()	표시할 별의 개수를 지정한다.
android:rating	setRating()	기본 평점을 지정한다.
android:stepSize	setStepSize()	스텝 크기를 지정한다. 예를 들어 값이 "0.5"이면 한 단계 값을 증감할 때 별 1/2만큼 평점이 증감한다.
android:isIndicator	setIsIndicator()	"true"이면 사용자 입력을 받을 수 없고 출력만 한다. 기본값은 "false"이므로 사용자 입력을 받을 수 있다.

## 04 프로그레스바 ▶ RatingBar

### 실습 3-10

### RatingBar1

#### ■ 실행 화면



(a) 초기 화면



(b) 평점 1/2 증가 후 [투표 결과는?] 클릭

그림 3-19 실행 화면

## 04 프로그레스바 ▶ RatingBar



실습 3-10

RatingBar1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <RatingBar
7          android:id="@+id/ratingVote"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:numStars="5"
11         android:rating="3"
12         android:stepSize="0.5"/>
13     <LinearLayout
14         android:layout_width="match_parent"
15         android:layout_height="wrap_content">
16         <Button
17             android:id="@+id/btnDec"
18             android:layout_width="wrap_content"
19             android:layout_height="wrap_content"
20             android:onClick="mOnClick"
21             android:text="-"/>
22         <Button
```



## 04 프로그레스바 ▶ RatingBar



실습 3-10

RatingBar1

```
23         android:id="@+id/btnInc"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:onClick="mOnClick"
27         android:text="+"/>
28     <Button
29         android:id="@+id/btnResult"
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:onClick="mOnClick"
33         android:text="투표 결과는?"/>
34 </LinearLayout>
35 </LinearLayout>
```

## 04 프로그레스바 ▶ RatingBar



실습 3-10

RatingBar1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      private RatingBar mRatingVote;
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_main);
9          mRatingVote = (RatingBar) findViewById(R.id.ratingVote);
10     }
11
12     public void mOnClick(View v) {
13         switch (v.getId()) {
14             case R.id.btnDec:
15                 mRatingVote.incrementProgressBy(-1);
16                 break;
17             case R.id.btnInc:
18                 mRatingVote.incrementProgressBy(1);
```

## 04 프로그레스바 ▶ RatingBar



실습 3-10

RatingBar1

```
19         break;
20     case R.id.btnResult:
21         Toast.makeText(this, "현재값 = " + mRatingVote.getRating(),
22             Toast.LENGTH_SHORT).show();
23         break;
24     }
25 }
26 }
```

### ■ SeekBar

- 프로그레스바의 일종
- 썸(thumb)을 이용해 사용자가 진행값을 조작
- 모양이나 동작이 윈도우즈 운영 체제의 스크롤바(scroll bar)와 거의 같음
- 썸을 조작하면 이벤트가 발생하는데 SeekBar.OnSeekBarChangeListener 인터페이스 객체를 통해 처리

### ■ SeekBar.OnSeekBarChangeListener 인터페이스에 포함된 추상 메서드

표 3-8 SeekBar.OnSeekBarChangeListener 인터페이스

추상 메서드	의미
<code>void onStartTrackingTouch(SeekBar seekBar)</code>	사용자가 터치하여 썸을 끌기를 시작했다.
<code>void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser)</code>	썸의 위치가 변경되었다. 현재 위치값은 progress이며 사용자의 행위가 아닌 자바 코드로 위치값을 변경한 경우에는 fromUser의 값이 "false"가 된다.
<code>void onStopTrackingTouch(SeekBar seekBar)</code>	사용자가 손을 떼어 썸을 끌기를 중단했다.

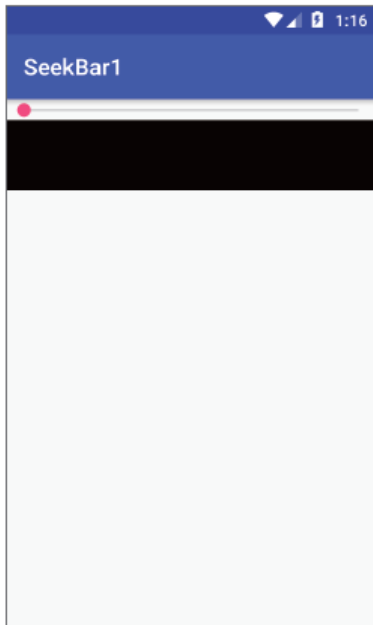
## 04 프로그레스바 ▶ SeekBar



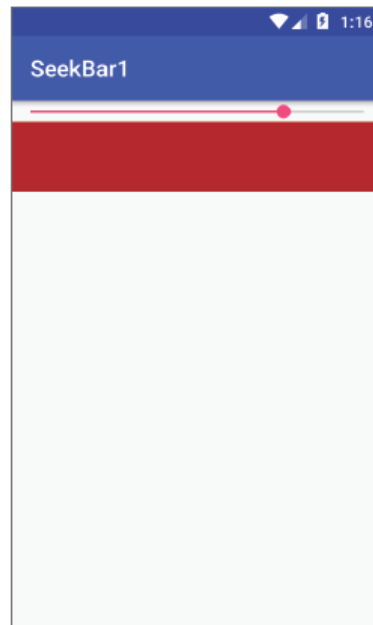
실습 3-11

SeekBar1

### ■ 실행 화면



(a) 초기 화면



(b) 썸 조작 후

그림 3-20 실행 화면

## 04 프로그레스바 ▶ SeekBar



실습 3-11

SeekBar1

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="wrap_content"
5      android:orientation="vertical">
6      <SeekBar
7          android:id="@+id/seekRed"
8          android:layout_width="match_parent"
9          android:layout_height="wrap_content"
10         android:max="255"/>
11     <View
12         android:id="@+id/viewColor"
13         android:layout_width="match_parent"
14         android:layout_height="60dp"/>
15 </LinearLayout>
```

## 04 프로그레스바 ▶ SeekBar



실습 3-11

SeekBar1

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      private SeekBar mSeekRed;
4      private View mViewColor;
5
6      @Override
7      protected void onCreate(Bundle savedInstanceState) {
8          super.onCreate(savedInstanceState);
9          setContentView(R.layout.activity_main);
10
11         mSeekRed = (SeekBar) findViewById(R.id.seekRed);
12         mViewColor = findViewById(R.id.viewColor);
13
14         mViewColor.setBackgroundColor(Color.argb(255, mSeekRed.getProgress(), 0, 0));
15
16         mSeekRed.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
17
18             @Override
```

## 04 프로그레스바 ▶ SeekBar

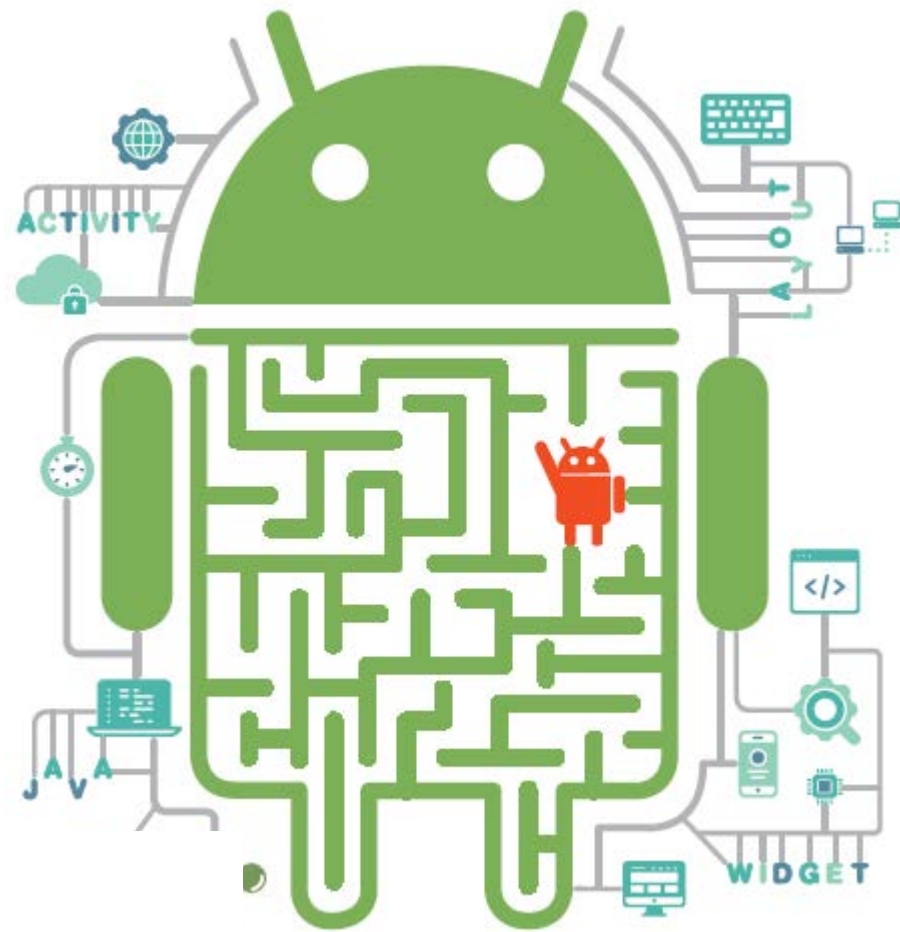


실습 3-11

SeekBar1

```
19         public void onProgressChanged(SeekBar seekBar, int progress, boolean
20             if (fromUser) {
21                 mViewColor.setBackgroundColor(Color.argb(255, progress, 0, 0));
22             }
23         }
24
25         @Override
26         public void onStartTrackingTouch(SeekBar seekBar) {
27         }
28
29         @Override
30         public void onStopTrackingTouch(SeekBar seekBar) {
31         }
32     });
33 }
34 }
```





단계별로 배우는

# 안드로이드 프로그래밍