

모바일 프로그래밍

02 사용자 인터페이스 기초 1

2017 2학기

강승우

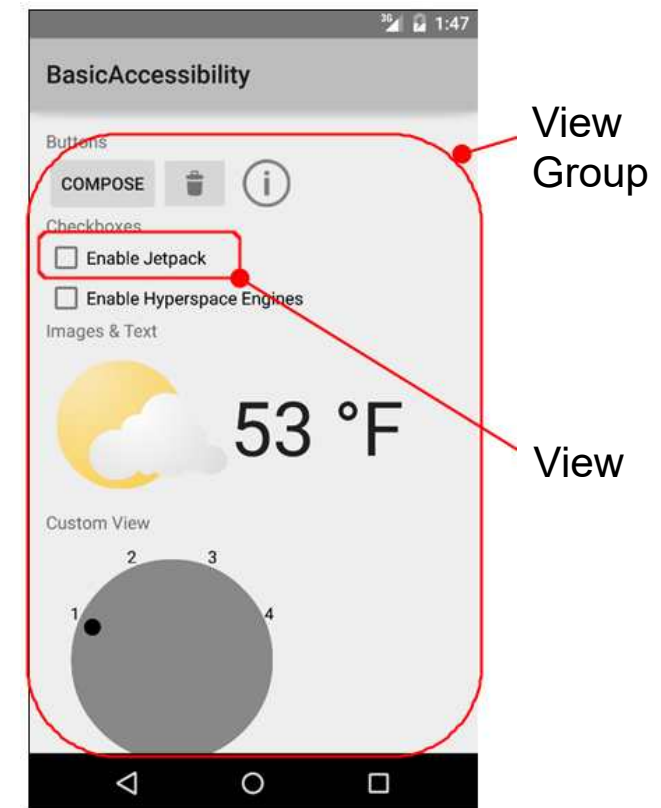
사용자 인터페이스 기초

- UI 개요
- 기본 뷰 (View)
- 레이아웃 (Layout)

UI 개요

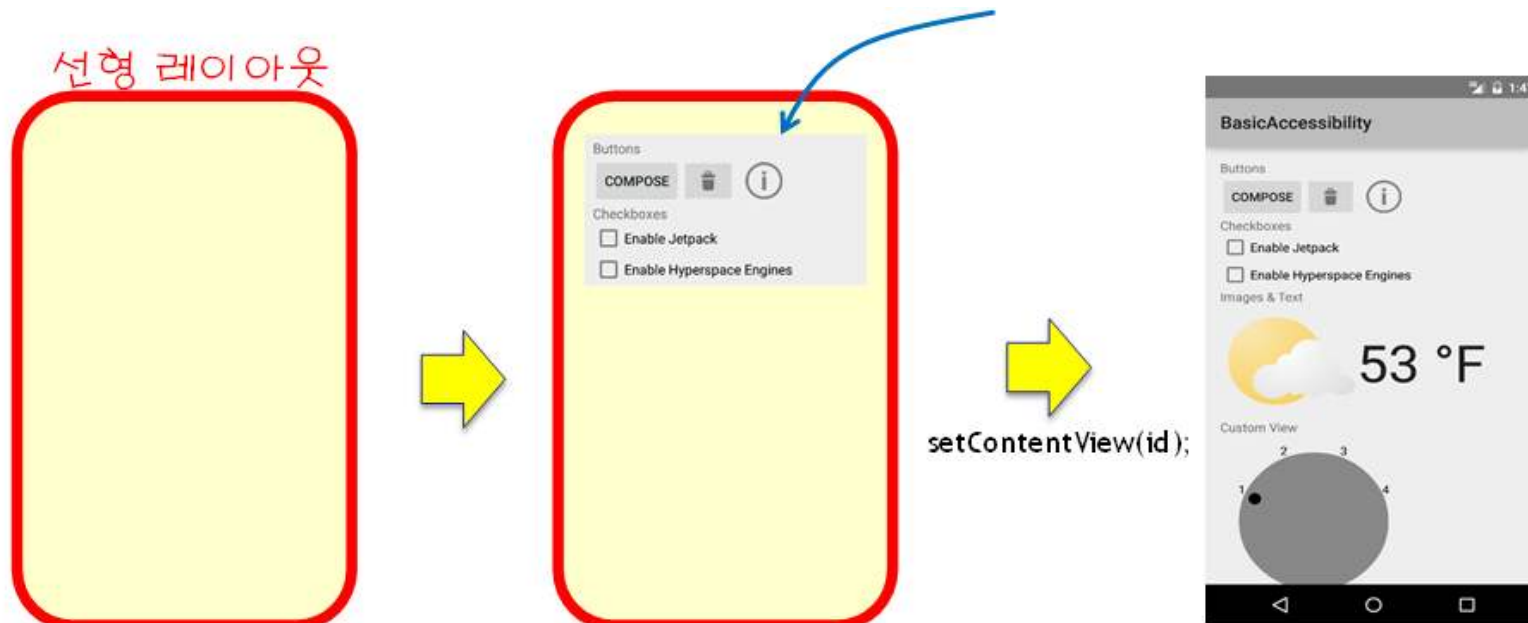
사용자 인터페이스 (UI)

- GUI (Graphical User Interface) 기반
- 안드로이드 SDK는 UI를 위한 다양한 요소를 제공
 - 버튼, 리스트, 스크롤 바, 체크 박스, 메뉴 등
- 뷰 (View)와 뷰 그룹 (View Group)
 - View
 - 안드로이드에서 UI를 구성하는 기본 building block
 - 컨트롤 또는 위젯이라고도 불림
 - 버튼, 텍스트 필드, 체크 박스 등
 - View 클래스를 상속받아서 작성됨
 - View Group
 - View를 담는 컨테이너, 흔히 레이아웃 (layout)이라고 불림
 - ViewGroup 클래스를 상속받아서 작성
 - linear layout, table layout, relative layout 등
 - 내부의 view를 배치하는 방식이 다름



UI 작성 절차

- View group 생성
- 필요한 view 추가
- Activity 화면으로 설정



UI 작성 방법

1. XML로 작성
2. 자바 코드로 작성
3. XML과 코드를 같이 사용하여 작성
4. Visual design 도구 이용

① XML로 사용자 인터페이스 기술

```
...  
<Button  
    android:text="첫번째 버튼"  
    android:id="@+id/button1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
</Button>  
...
```

② 코드로 사용자 인터페이스 작성

```
...  
Button b1 = new Button(this);  
b1.setText("첫번째 버튼");  
container.addView(b1);  
...
```



XML로 UI 작성

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  tools:context=".MainActivity"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical" >

  <Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="첫번째 버튼" />

  <Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="두번째 버튼" />

</LinearLayout>
```



코드로 UI 작성

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);
```

```
        LinearLayout container = new LinearLayout(this);  
        container.setOrientation(LinearLayout.VERTICAL);
```

```
        Button b1 = new Button(this);  
        b1.setText("첫번째 버튼");  
        container.addView(b1);
```

```
        Button b2 = new Button(this);  
        b2.setText("두번째 버튼");  
        container.addView(b2);
```

```
        setContentView(container);  
        //setContentView(R.layout.activity_main);
```

```
    }  
}
```



XML과 코드를 동시에 사용

<Button

```
android:id="@+id/button1"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:text="첫번째 버튼" />
```

<Button

```
android:id="@+id/button2"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:text="두번째 버튼" />
```

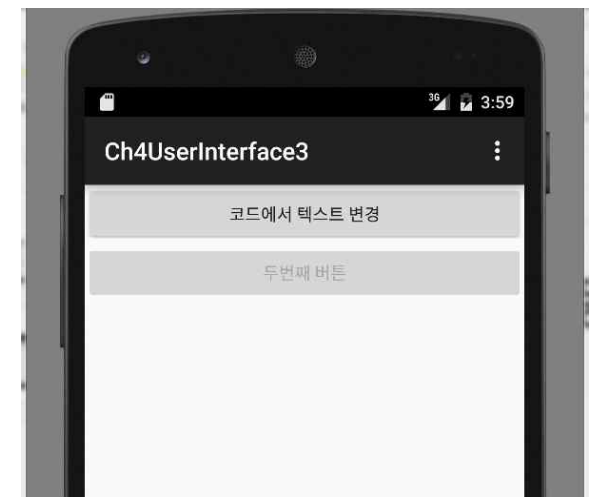
@Override

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

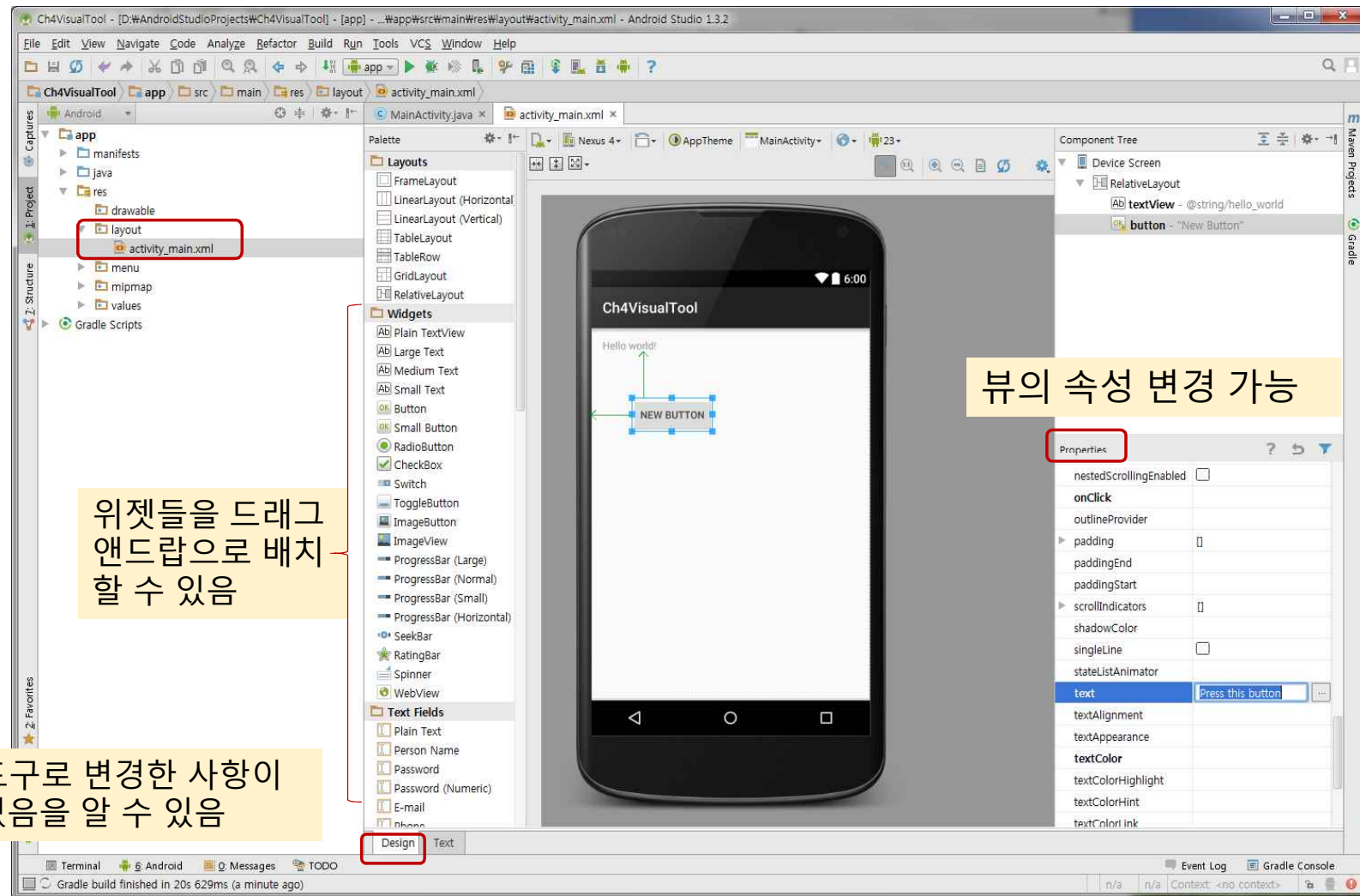
```
    Button b1 = (Button) findViewById(R.id.button1);  
    b1.setText("코드에서 텍스트 변경");
```

```
    Button b2 = (Button) findViewById(R.id.button2);  
    b2.setEnabled(false);
```

```
}
```

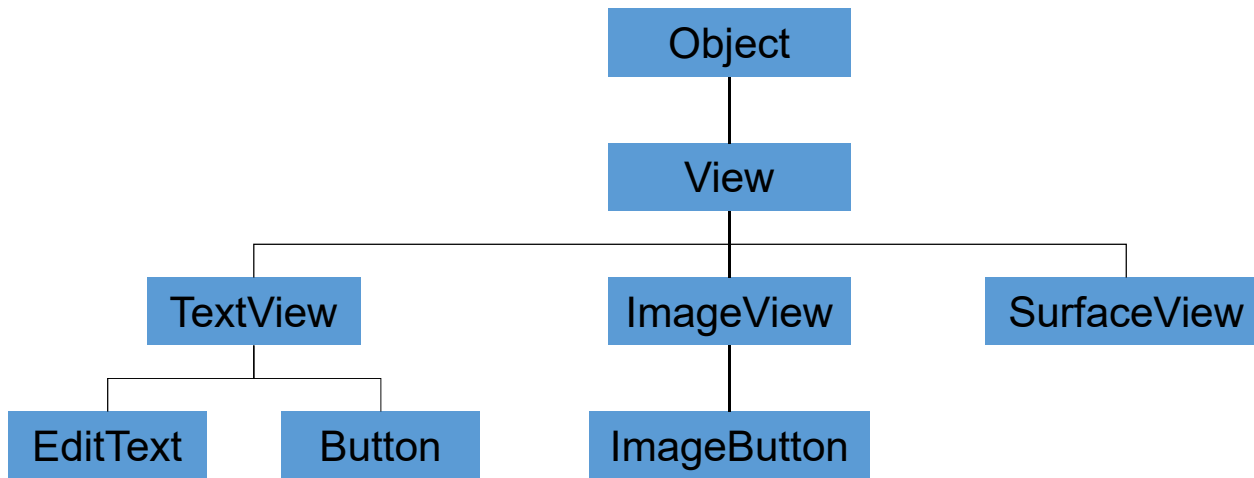


비주얼 도구 사용



뷰 속성

뷰 (View)



• View 클래스

- 모든 뷰들의 부모 클래스
- View 클래스의 필드나 메소드는 모든 뷰에서 공통적으로 사용 가능
- 관련 속성
 - id
 - size/location
 - color
 - visibility
 - margin/padding

뷰 속성

- id

<https://developer.android.com/reference/android/view/View.html>

- 뷰 식별자

android:id="@+id/my_button"

- XML의 각 속성 이름(attribute name)과 해당하는 자바 메소드 정보 참조

- 크기/위치

- android:layout_width / android:layout_height

<Button

```
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="첫 번째 버튼" />
```

상수	설명
match_parent	부모의 크기를 꽉 채운다(fill_parent도 같은 의미).
wrap_content	뷰가 나타내는 내용물의 크기에 맞춘다.
숫자	크기를 정확히 지정한다.

android:layout_width="200px"

뷰의 크기 단위

단위	설명
px(pixels)	화면의 실제 픽셀을 나타낸다. 픽셀은 권장되는 단위는 아님에 그 이유는 장치마다 화면의 밀도가 다르기 때문이다.
dp(density-independent pixels)	dp는 화면의 밀도가 160dpi 화면에서 하나의 물리적인 픽셀을 말한다. 따라서 크기를 160dp로 지정하면 화면의 밀도와는 상관없이 항상 1인치가 된다. dp로 뷰의 크기를 지정하면 화면의 밀도가 다르더라도 항상 동일한 크기로 표시된다.
sp(scale-independent pixels)	화면 밀도와 사용자가 지정한 폰트 크기에 영향을 받아서 변환된다. 이 단위는 폰트 크기를 지정하는 경우에 추천된다.
pt(points)	1/72 인치를 표시한다.
mm(millimeters)	밀리미터를 나타낸다.
in(inches)	인치를 나타낸다.

크기 변경

- android:layout_width
- android:layout_height
- wrap_content
- match_parent

와 match_

layout_width와 layout_height가 모두 wrap_content이므로 가로 세로 길이가 모두 내용물에 맞추어진다.



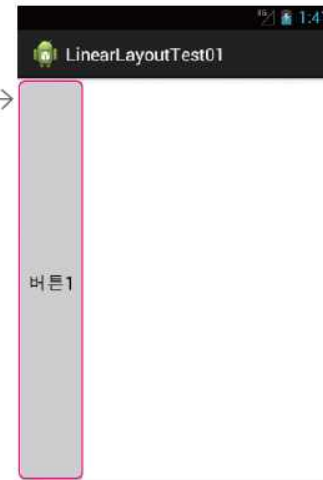
(a) layout_width="wrap_content",
layout_height="wrap_content"



(b) layout_width="match_parent",
layout_height="wrap_content"

layout_width가 match_parent이므로 가로 방향으로 부모공간을 모두 차지한다.

layout_height가 match_parent이므로 세로로 부모 공간을 모두 차지한다.



(c) layout_width="wrap_content",
layout_height="match_parent"



(d) layout_width="match_parent",
layout_height="match_parent"

모두 match_parent이므로 가로와 세로로 부모 공간을 모두 차지한다.

색상

- 뷰의 배경색 지정
 - android:background
 - 16 진수로 RGB와 투명도 설정
 - 투명도는 생략 가능

표시 방법	설명
#RRGGBB	RR은 빨간색 성분, GG는 녹색 성분, BB는 청색 성분을 나타낸다.
#AARRGGBB	AA는 투명도, RR은 빨간색 성분, GG는 녹색 성분, BB는 청색 성분을 나타낸다.

android:background="#ff0000" : 빨간색
android:background="#00ff00" : 녹색
android:background="#ff000000": 완전 불투명 검정색
android:background="#ffffffff": 불투명 흰색

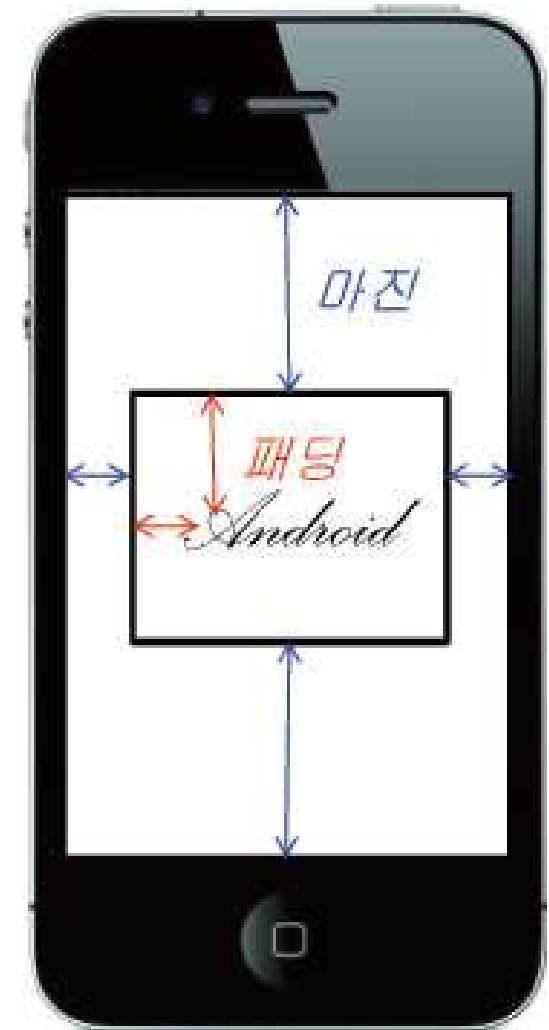
뷰의 화면 표시

- 뷰의 초기 가시성 제어
 - android:visibility

상수	값	설명
visible	0	화면에 보이게 한다. 디폴트 값
invisible	1	표시되지 않는다. 그러나 배치에서 공간을 차지한다.
gone	2	완전히 숨겨진다.

패딩과 마진

- 패딩 (padding)
 - 뷰의 경계와 뷰의 내용물 사이의 간격
- 마진 (margin)
 - 컨테이너와 뷰 사이의 간격



패딩과 마진

- 패딩
 - `android:padding`
 - `android:paddingLeft`, `android:paddingRight`,
`android:paddingTop`, `android:paddingBottom`
- 마진
 - `android:margin`
 - `android:marginLeft`, `android:marginRight`,
`android:marginTop`, `android:marginBottom`



패딩과 마진 예

```
<Button  
    android:id="@+id/button01"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_margin="2pt"  
    android:padding="2pt"  
    android:text="마진 =2 패딩 =2" />
```



기본 뷰

TextView

<http://developer.android.com/reference/android/widget/TextView.html>

- 화면에 텍스트를 표시
 - 편집 불가능 텍스트
 - 설명이나 제목 등에 사용
- 속성
 - android:text
 - 표시할 텍스트
 - android:textColor
 - 텍스트 색상
 - android:textSize
 - 텍스트 크기
 - android:textStyle
 - 텍스트 스타일 (bold, italic, bolditalic)
 - android:typeface
 - 텍스트 폰트
 - (normal, sans, serif, monospace)
 - android:width / android:height
 - 픽셀 단위 텍스트 뷰의 길이, 높이
 - android:lines
 - 줄 수 단위의 텍스트 뷰의 높이

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#0000ff"
    android:text="This is a test."
    android:textColor="#ff0000"
    android:textSize="20pt"
    android:textStyle="italic"
    android:typeface="serif" />
```

```
</LinearLayout>
```



EditText

- 입력이 가능한 텍스트 필드
- TextView의 자식 클래스
- 속성

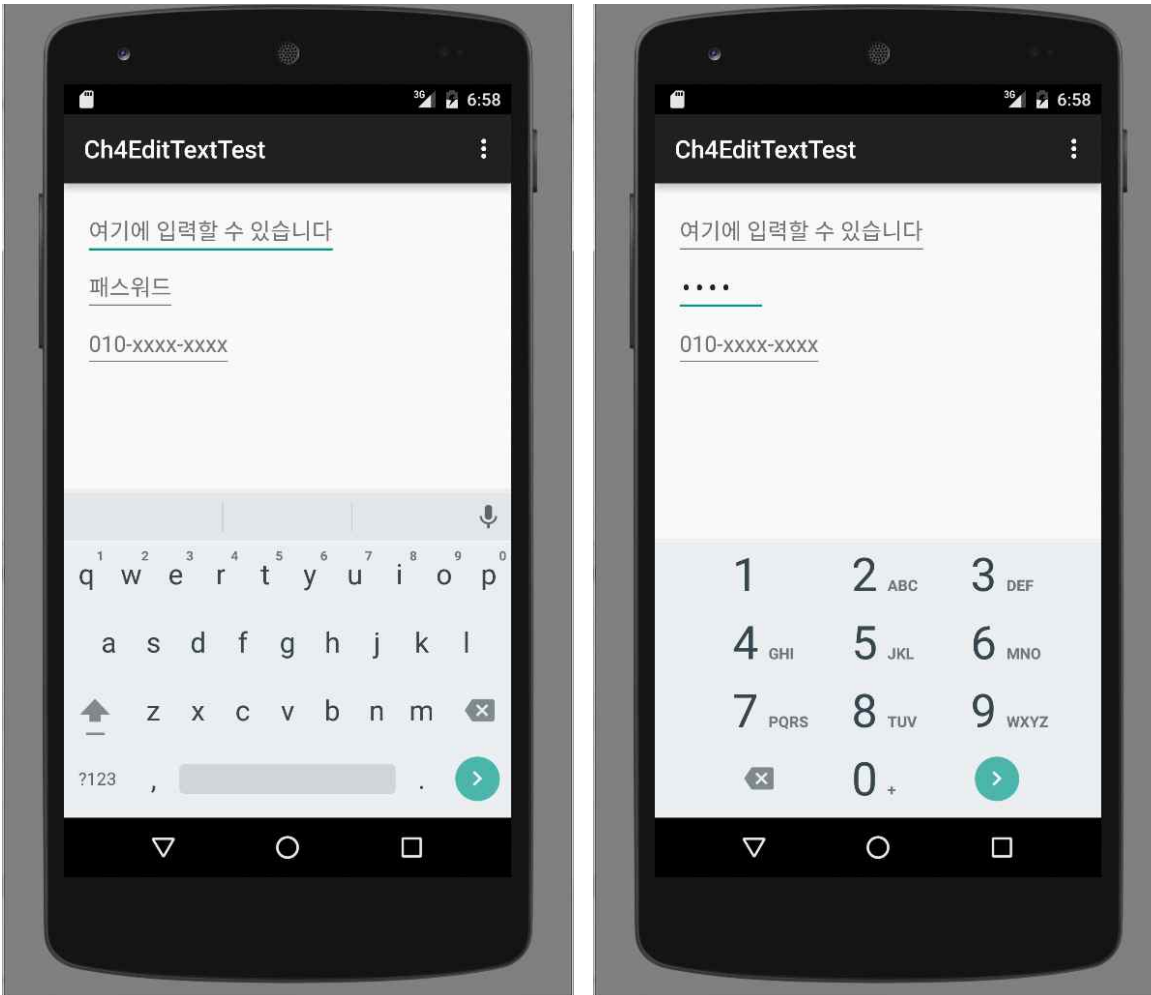
속성	설명
<u>android:autoText</u>	자동으로 타이핑 오류를 교정한다.
<u>android:drawableBottom</u>	텍스트의 아래에 표시되는 이미지 리소스이다.
<u>android:drawableRight</u>	텍스트의 오른쪽에 표시되는 이미지 리소스이다.
<u>android:editable</u>	편집가능
<u>android:text</u>	표시되는 텍스트이다.
<u>android:singleLine</u>	true이면 한줄만 받음
<u>android:inputType</u>	입력의 종류
<u>android:hint</u>	입력 필드에 표시되는 힌트 메시지

InputType 속성

- 입력되는 내용을 제한할 수 있음
 - 예: android:inputType="numberPassword"
 - 숫자만 입력할 수 있는 입력기가 화면에 표시

inputType	설명
none	편집이 불가능한 문자열
Text	일반적인 문자열
textMultiLine	여러 줄로 입력 가능
textPostalAddress	우편번호
textEmailAddress	이메일 주소
textPassword	패스워드
textVisiblePassword	패스워드 화면에 보인다.
number	숫자
numberSigned	부호가 붙은 숫자
numberDecimal	소수점이 있는 숫자
phone	전화번호
datetime	시간

EditText 예제



```
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
...
```

```
<EditText  
    android:id="@+id/edit1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="여기에 입력할 수 있습니다"  
    android:inputType="text"  
    android:singleLine="true"  
/>
```

```
<EditText  
    android:id="@+id/edit2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="패스워드"  
    android:inputType="numberPassword"  
/>
```

```
<EditText  
    android:id="@+id/edit3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="010-xxxx-xxxx"  
    android:inputType="phone"  
/>
```

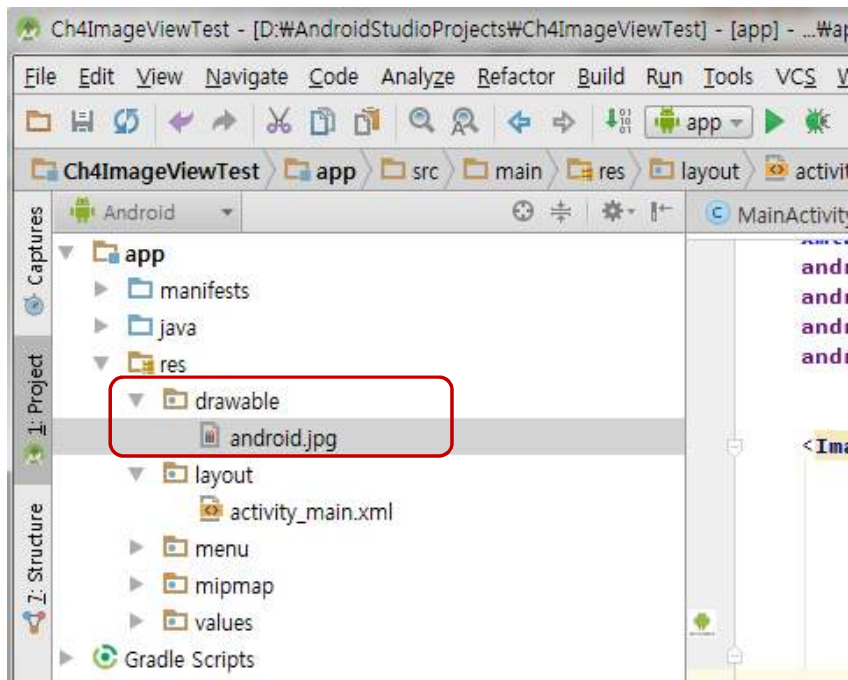
```
</LinearLayout>
```

ImageView

- 아이콘 같은 이미지를 표시하는데 사용

속성	설정 메소드	설명
<code>android:adjustViewBounds</code>	<code>setAdjustViewBounds(boolean)</code>	drawable의 종횡비를 유지하기 위하여 이미지 뷰의 가로, 세로를 조정
<code>android:cropToPadding</code>		true이면 패딩 안에 맞추어서 이미지를 자른다.
<code>android:maxHeight</code>	<code>setMaxHeight(int)</code>	이미지 뷰의 최대 높이
<code>android:maxLength</code>	<code>setMaxLength(int)</code>	이미지 뷰의 최대 너비
<code>android:scaleType</code>	<code>setScaleType(ImageView.ScaleType)</code>	이미지 뷰의 크기에 맞추어 어떻게 확대나 축소할 것인지 방법 선택
<code>android:src</code>	<code>setImageResource(int)</code>	이미지 소스
<code>android:tint</code>	<code>setColorFilter(int, PorterDuff.Mode)</code>	이미지 배경 색상

ImageView 예제



<ImageView

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/icon"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:adjustViewBounds="true"
android:src="@drawable/android"
/>
```

Button

- 사용자가 클릭할 수 있는 가장 기본적인 위젯
 - TextView 클래스를 상속 받음
 - TextView의 모든 속성 사용 가능
- ToggleButton
 - on/off 두 가지 상태를 왔다 갔다 하는 버튼

```
<Button  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:text="버튼1"  
    android:id="@+id/button1"/>
```

```
<Button  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:text="버튼2"  
    android:id="@+id/button2"/>
```

```
<ToggleButton  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:text="토글 버튼"  
    android:id="@+id/button_toggle"/>
```

