

# 3. 기본 위젯 익히기

01 사용자 인터페이스 기토

02 뷰의 개요

03 기본 위젯 다루기

04 기본 위젯 활용하기

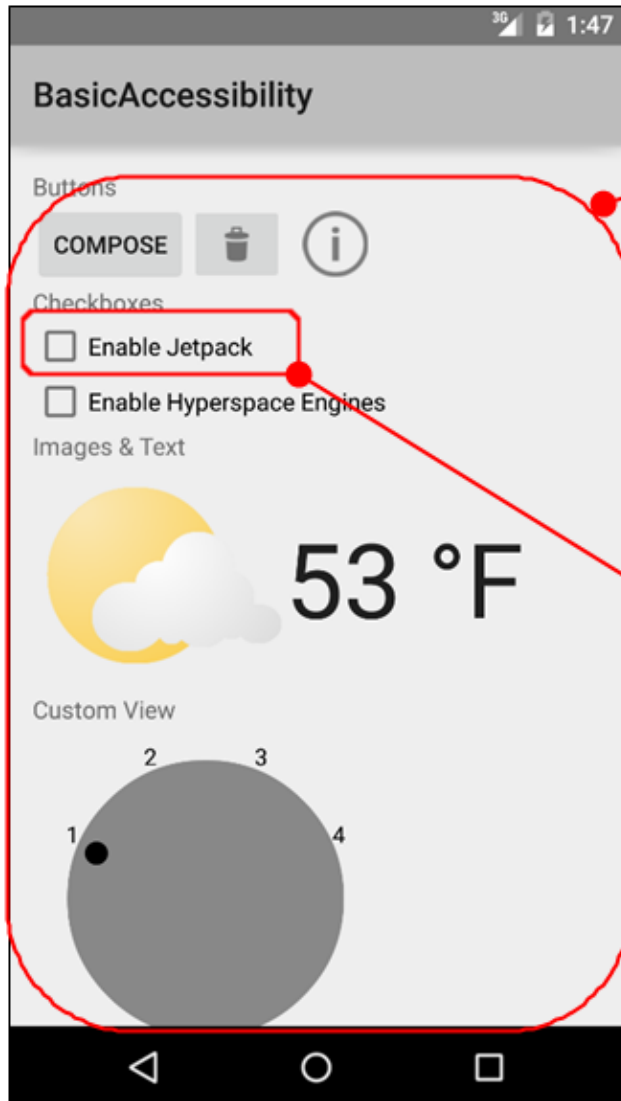


# 1. 사용자 인터페이스 기초

- **자바의 swing은 사용하지 않음**
  - ▣ 너무 리소스를 많이 잡아먹음!
- **독자적인 사용자 인터페이스 컨트롤 사용**
  - ▣ 버튼, 리스트, 스크롤 바, 체크 박스, 메뉴, 대화 상자

# 1. 사용자 인터페이스 기초

## □ 뷰와 뷰그룹



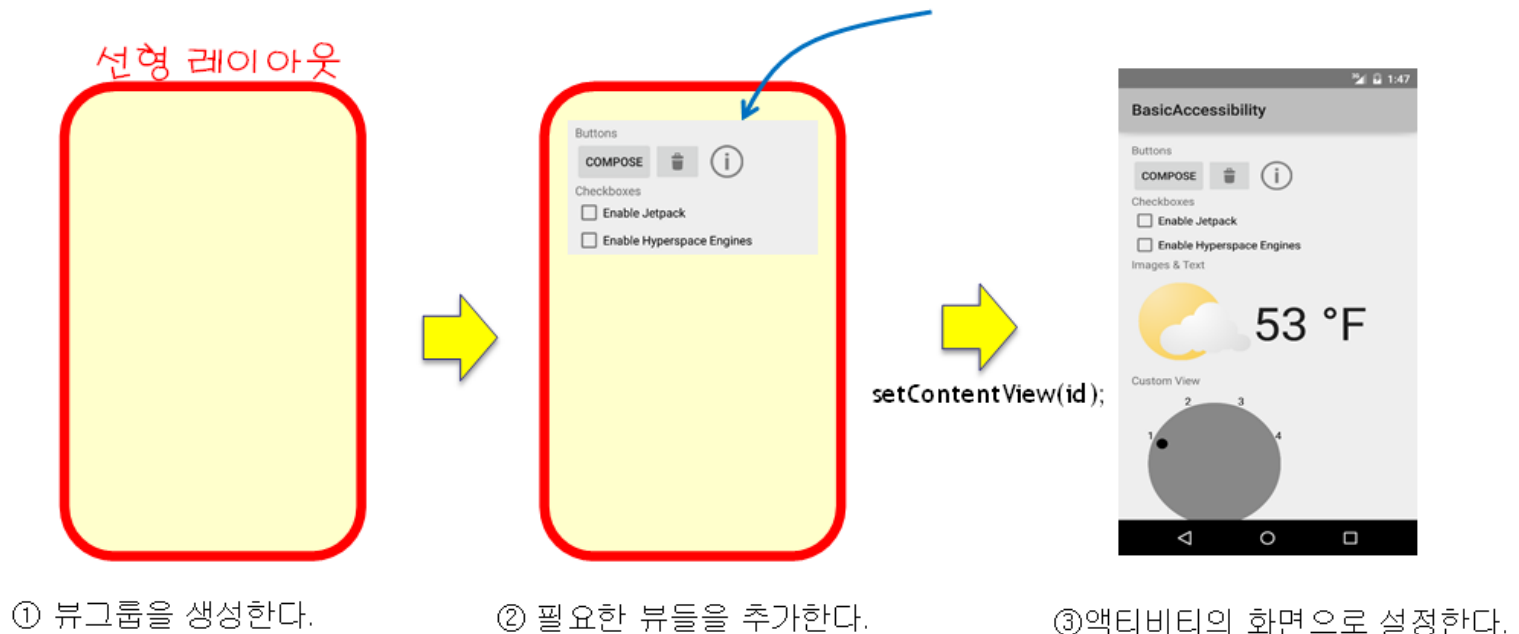
**뷰그룹:** 다른 뷰들을 담는 컨테이너 기능을 한다. 뷰그룹은 ViewGroup 클래스에서 상속받아서 작성된다. 흔히 레이아웃(layout)이라고 불리며 선형 레이아웃, 테이블 레이아웃, 상대적 레이아웃 등이 여기에 속한다. 각 레이아웃은 정해진 정책에 따라서 뷰들을 배치한다.

**뷰:** 컨트롤 또는 위젯이라고도 불린다. 사용자 인터페이스를 구성하는 기초적인 빌딩 블록이다. 버튼, 텍스트 필드, 체크박스 등이 여기에 속한다. 뷰들은 View 클래스를 상속받아서 작성된다.

# 1. 사용자 인터페이스 기초

## □ Ui를 작성하는 절차

1. 뷰그룹을 생성한다.
2. 필요한 뷰를 추가한다.
3. 액티비티 화면으로 설정한다.



# 1. 사용자 인터페이스 기초

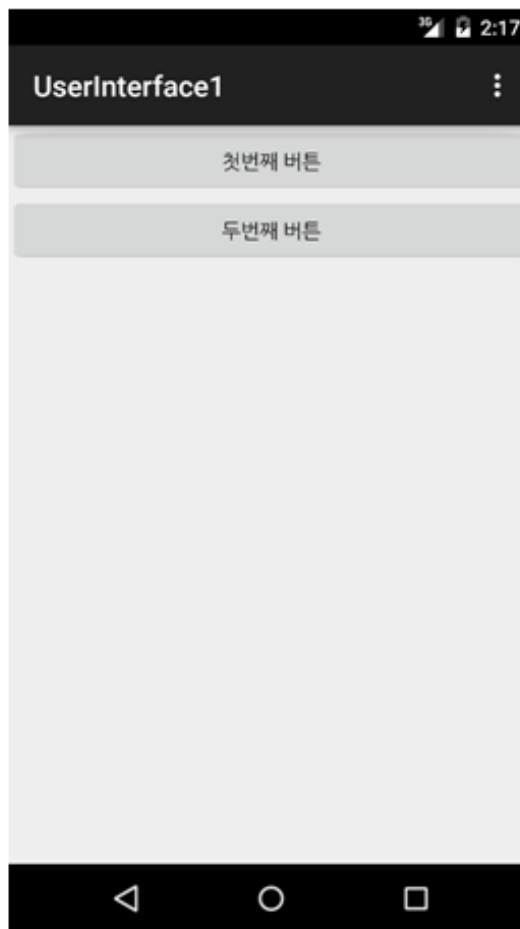
## □ Ui를 작성하는 3가지 방법

### 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);  
...
```



# 1. 사용자 인터페이스 기초

## □ XML로 ui 작성

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" >

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

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

</LinearLayout>
```

선형 레이아웃  
이라는 뷰그룹  
을 생성한다.

버튼이라는 뷰  
를 생성한다.

# 1. 사용자 인터페이스 기초

## □ 코드로 뷰를 생성하는 방법

UiThreadActivity.java

```
package company.co.kr.userinterface2;  
// 소스만 입력하고 Alt+Enter 키를 눌러서 import 문장을 자동으로 생성한다.
```

```
public class UserInterface2Activity extends ActionBarActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);
```

선형 레이아웃  
을 생성한다.

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

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

버튼을 선형 레  
이아웃에 추가  
한다.

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

```
        setContentView(container);
```

```
    }
```

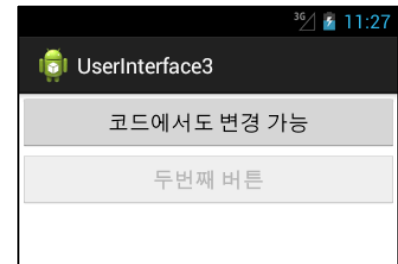
```
}
```

# 1. 사용자 인터페이스 기초

## □ XML과 코드를 동시에 사용하는 방법

```
<Button
    android:text="첫번째 버튼"
    android:id="@+id/button1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</Button>
<Button
    android:text="두번째 버튼"
    android:id="@+id/button2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</Button>
```

버튼에 식별자를 부여한다.



```
Button b1 = (Button) findViewById(R.id.button1);
b1.setText("코드에서도 변경 가능");

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

id가 button1인 버튼을 찾는다.

버튼의 텍스트를 변경한다.

버튼을 불활성으로 만든다.



## 2. 뷰의 개요

9

### □ 뷰와 뷰그룹[1/4]

#### ▣ View 클래스

- 안드로이드 화면에서 실제로 사용되는 것들은 모두 View 클래스 상속을 받음
- 뷰 클래스는 다른 말로 ' 위젯 ' 이라고도 함
- 다른 위젯을 담을 수 있는 위젯은 레이아웃이라고 함
- 레이아웃은 ViewGroup 클래스 아래 존재

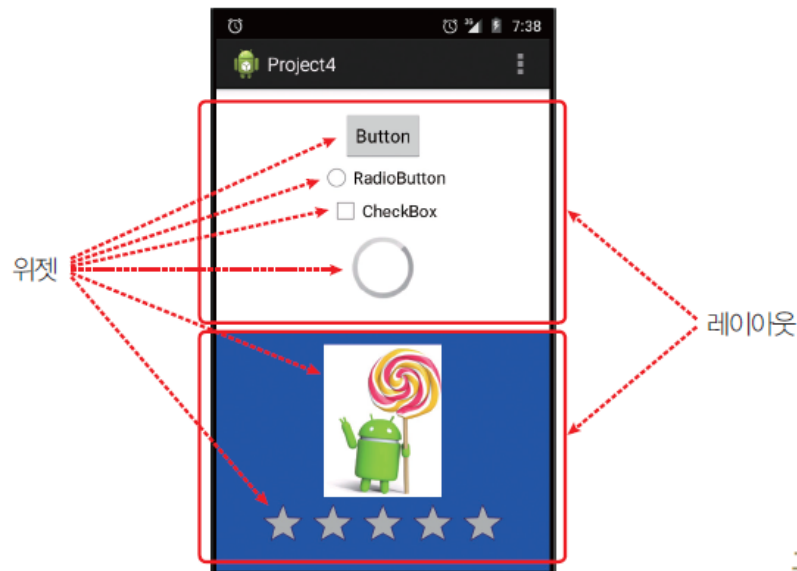


그림 4-1 위젯과 레이아웃

## 2. 뷰의 개요

10

### □ 뷰와 뷰그룹[2/4]

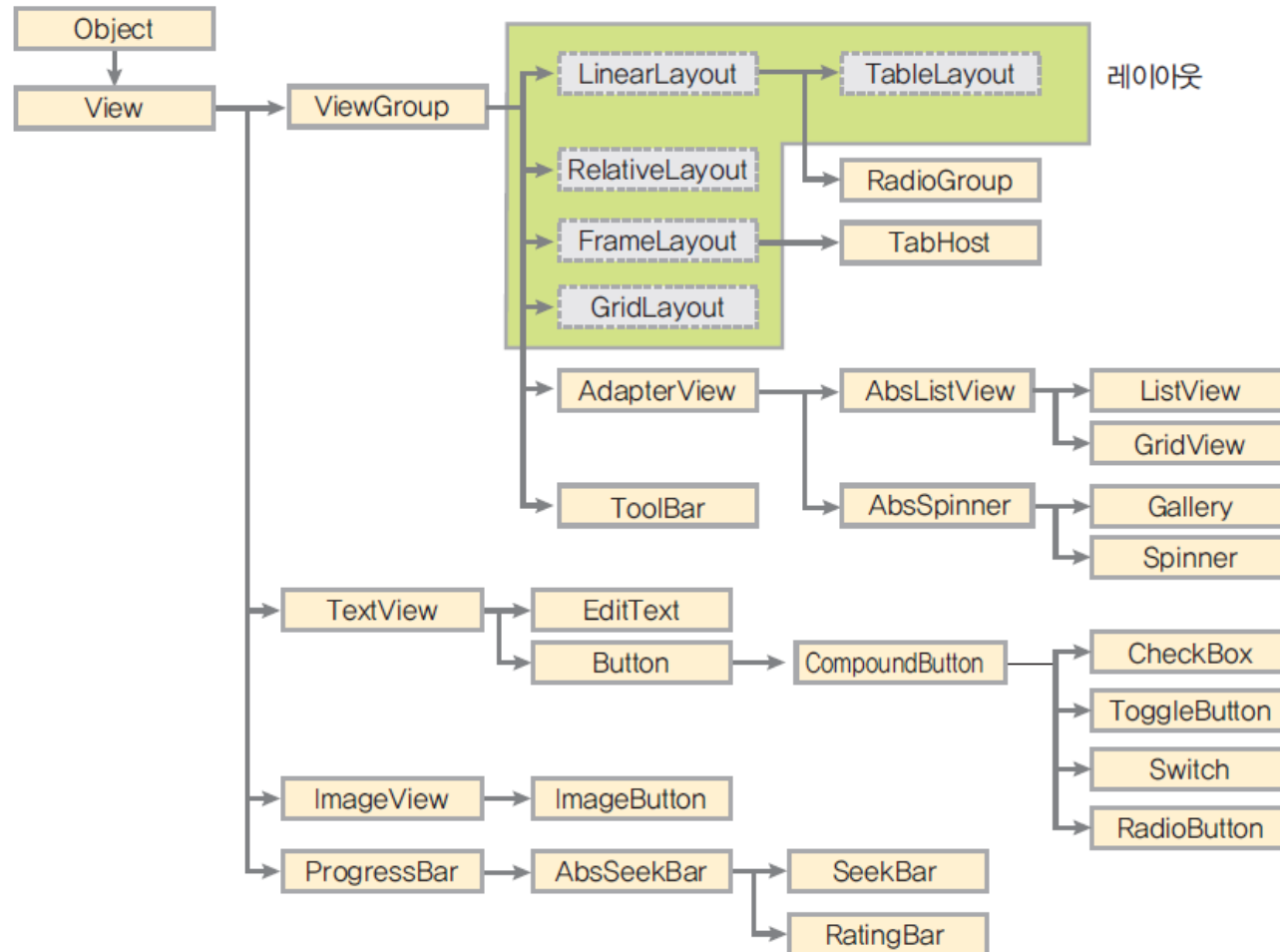


그림 4-2 안드로이드 View 클래스 계층도

# 1. 뷰의 개요

11

## □ 뷰와 뷰그룹[3/4]

### ❖ 클래스 상속관계 찾기

- ✓ <http://developer.android.com/reference>에서 자세히 확인할 수 있음

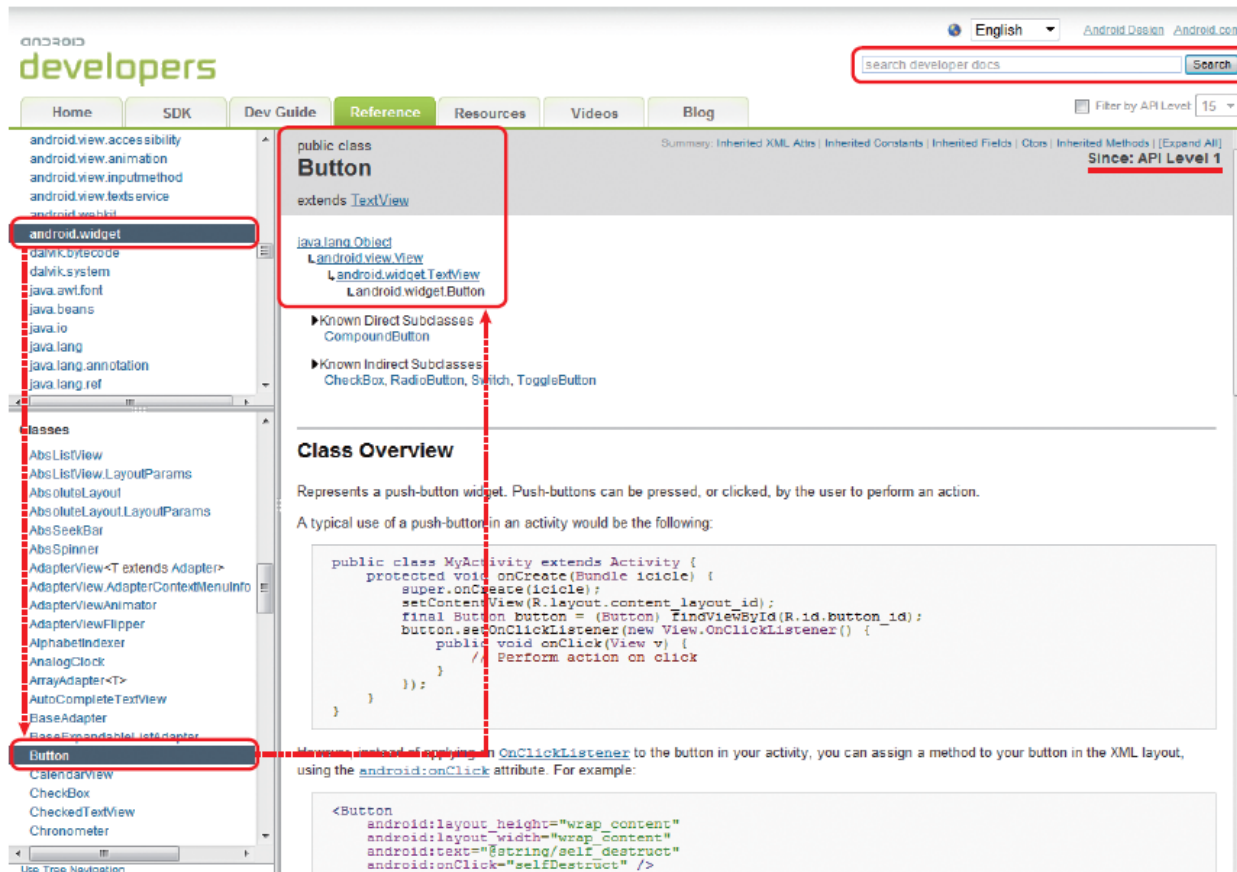


그림 4-4 클래스의 상속 관계를 찾는 방법

# 뷰의 필드와 메소드

## □ id

- ▣ 뷰의 식별자

## □ 뷰의 위치와 크기

상수	설명
<code>match_parent</code>	부모의 크기를 꼭 채운다( <code>fill_parent</code> 도 같은 의미).
<code>wrap_content</code>	뷰가 나타내는 내용물의 크기에 맞춘다.
숫자	크기를 정확히 지정한다.

<Button

```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/button_text" />
```

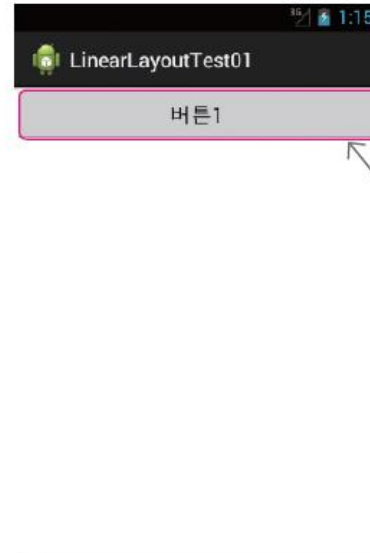
버튼의 위치와 크기를  
결정한다.

layout\_width와 layout\_height가 모두 wrap\_content이므로 가로 세로 길이가 모두 내용물에 맞추어진다.



(a) layout\_width="wrap\_content",  
layout\_height="wrap\_content"

layout\_width가 match\_parent이므로 가로 방향으로 부모공간을 모두 차지한다.



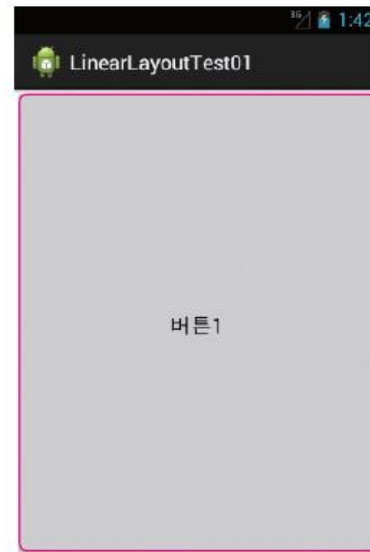
(b) layout\_width="match\_parent",  
layout\_height="wrap\_content"

layout\_height가 match\_parent이므로 세로로 부모 공간을 모두 차지한다.



(c) layout\_width="wrap\_content",  
layout\_height="match\_parent"

모두 match\_parent이므로 가로와 세로로 부모 공간을 모두 차지한다.



(d) layout\_width="match\_parent",  
layout\_height="match\_parent"

# 뷰의 크기 단위

단위	설명
px(pixels)	화면의 실제 픽셀을 나타낸다. 픽셀은 권장되는 단위는 아닌데 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.

# 색상

## □ 16진수로 투명도와 빛의 3원색인 RGB값을 표시

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.

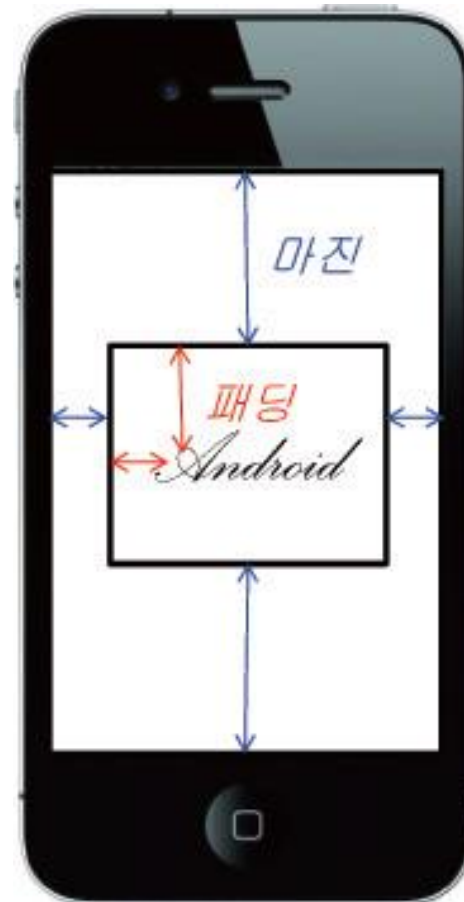
# 화면에 보이기 속성

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



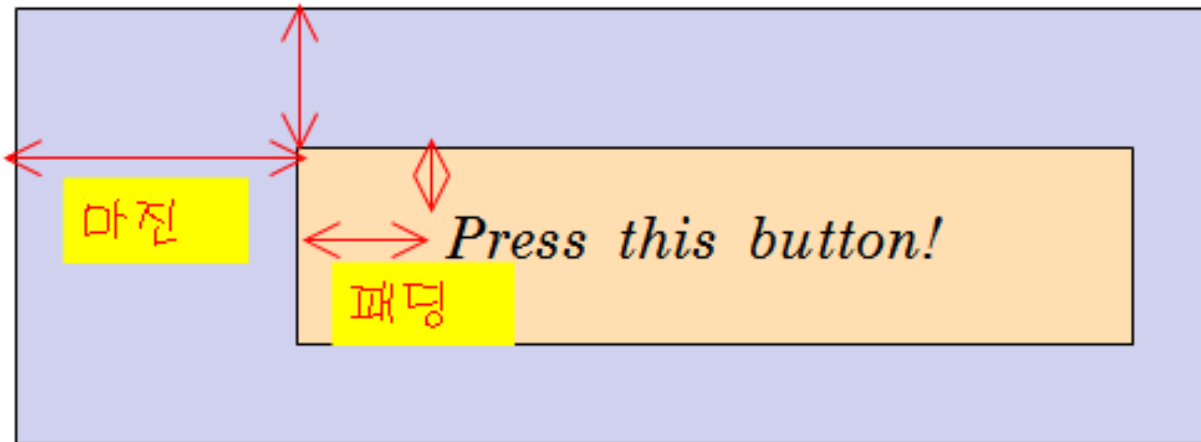
# 마진과 패딩

- 패딩이란 뷰의 경계와 뷰의 내용물 사이의 간격
- 마진이란 자식 뷰 주위의 여백



# 마진과 패딩

- `paddingLeft`, `paddingRight`, `paddingTop`, `paddingBottom`
- `layout_marginLeft`, `layout_marginRight`, `layout_marginTop`, `layout_marginBottom`



# 마진과 패딩의 예

```
<Button
```

```
    android:id="@+id/button01"
```

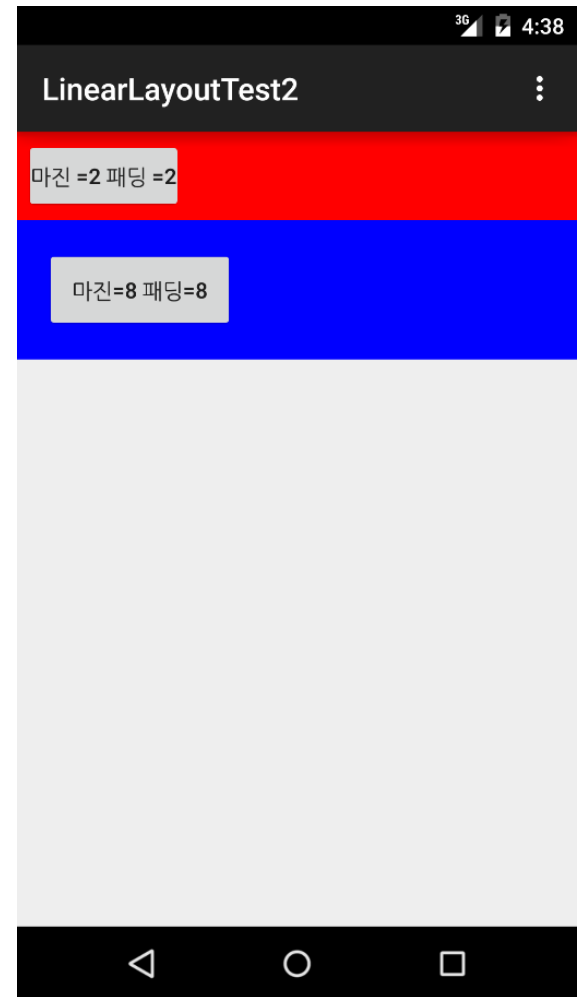
```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_margin="2pt"
```

```
    android:padding="2pt"
```

```
    android:text="마진 =2 패딩 =2" />
```



## □ View 클래스의 XML 속성

### ▶ 직접 풀어보기 4-2

다음 그림과 같은 화면을 XML로 코딩하자. 버튼, 텍스트뷰, 에디트텍스트, 버튼을 차례로 지정하고 앞에서 배운 다양한 속성을 지정해본다.



## 2. 기본 위젯 다루기

21

### □ 텍스트뷰[1/2]

#### ▣ 계층도

```
java.lang.Object
└ android.view.View
  └ android.widget.TextView
```

TextView 계층도

- textColor 속성 : 글자의 색상을 지정, #RRGGBB나 #AARRGGBB 형식
- textSize 속성 : 글자의 크기를 dp, px, in, mm, sp 단위로 지정
- typeface 속성 : 글자의 글꼴을 지정, 값으로 sans, serif, monospace 설정, 디폴트는 normal
- textStyle 속성 : 글자의 스타일을 지정, 값으로 bold, italic, bold|italic 을 설정, 디폴트는 normal
- singleLine 속성 : 글이 길어 줄이 넘어갈 경우 강제로 한 줄만 출력, 값으로 true와 false를 설정할 수 있으며 디폴트는 false

## □ 텍스트뷰[2/2]- 텍스트뷰 예제

```

1 <TextView
2     android:textSize="30dp"
3     android:text="textSize 속성" />
4 <TextView
5     android:textSize="30dp"
6     android:textColor="#00FF00"
7     android:text="textColor 속성" />
8 <TextView
9     android:textSize="30dp"
10    android:textStyle="bold|italic"
11    android:text="textStyle 속성" />
12 <TextView
13    android:textSize="30dp"
14    android:typeface="serif"
15    android:text="typeface 속성" />
16 <TextView
17    android:textSize="30dp"
18    android:singleLine="true"
19    android:text="singleLine 속성 singleLine 속성" />

```

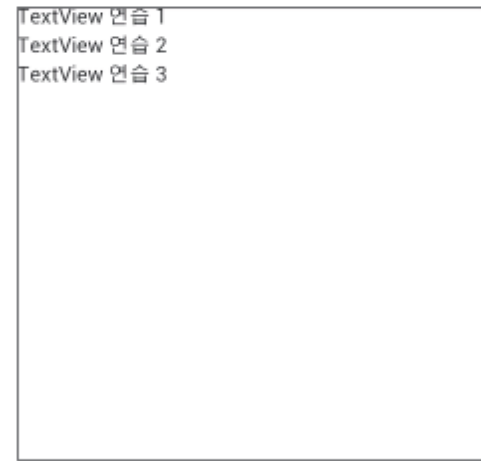
textSize 속성  
 textColor 속성  
 textStyle 속성  
 typeface 속성  
 singleLine 속성 single...

## □ Java 코드로 XML 속성 설정[1/3]

### ▣ 텍스트뷰 속성을 변경하는 예제

예제 4-17 텍스트뷰가 3개 있는 activity\_main.xml

```
1 <TextView
2     android:text="TextView 연습 1"
3     android:id="@+id/textView1" />
4 <TextView
5     android:text="TextView 연습 2"
6     android:id="@+id/textView2" />
7 <TextView
8     android:text="TextView 연습 3"
9     android:id="@+id/textView3" />
```



TextView 연습 1  
TextView 연습 2  
TextView 연습 3

## □ Java 코드로 XML 속성 설정[2/3]

### ▣ 텍스트뷰 속성을 변경하는 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {  
2     super.onCreate(savedInstanceState);  
3     setContentView(R.layout.main);  
4  
5     TextView tv1, tv2, tv3;  
6     tv1 = (TextView) findViewById(R.id.textView1);  
7     tv2 = (TextView) findViewById(R.id.textView2);  
8     tv3 = (TextView) findViewById(R.id.textView3);  
9  
10    tv1.setText("안녕하세요?");  
11    tv1.setTextColor(Color.RED);  
12    tv2.setTextSize(30);  
13    tv2.setTypeface(Typeface.SANS_SERIF, Typeface.BOLD_ITALIC);  
14    tv3.setText("가나다라마바사아자차카타파하가나다라마바사아자차카타파하");  
15    tv3.setSingleLine();  
16 }
```

안녕하세요?

#### TextView 연습 2

가나다라마바사아자차카타파하가나다라마바사아



## □ Java 코드로 XML 속성 설정[3/3]

### ▣ XML 속성과 관련 메소드

표 4-1 XML 속성과 관련 메소드

XML 속성	관련 메소드	비고
background	setBackgroundColor()	View 클래스
clickable	setClickable()	View 클래스
focusable	setFocusable()	View 클래스
id	setId()	View 클래스
longClickable	setLongClickable()	View 클래스
padding	setPadding()	View 클래스
rotation	setRotation()	View 클래스
scaleX, scaleY	setScaleX(), setScaleY()	View 클래스
visibility	setVisibility()	View 클래스
gravity	setGravity()	TextView 클래스
inputType	setRawInputType()	TextView 클래스
password	setTransformationMethod()	TextView 클래스
text	setText()	TextView 클래스
textColor	setTextColor()	TextView 클래스
textSize	setTextSize()	TextView 클래스

## □ 버튼과 에디트텍스트[1/12]

### ▣ 버튼

- 사용자에게서 어떤 값을 입력받기 위한 가장 기본적인 위젯으로 활용도가 높음

```
<TextView
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="나는 어떤 위젯일까요?" /a>
```

```
<Button
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="나는 어떤 위젯일까요?" />
```

## □ 버튼과 에디트텍스트[2/12]

### ▣ 버튼 계층도

```
java.lang.Object
└─ android.view.View
    └─ android.widget.TextView
        └─ android.widget.Button
```

Button 계층도

## □ 버튼과 에디트텍스트[3/12]

### ▣ 일반적인 버튼의 XML 코드

```
<Button
    android:id="@+id/button1"
    android:text="확인" />
```

#### ① 버튼 변수 선언

```
Button mybutton;
```

#### ② 변수에 버튼 위젯 대입

```
mybutton = (Button) findViewById(R.id.button1);
```

#### ③ 버튼 클릭할 때 동작하는 클래스 정의

```
mybutton.setOnClickListener( new View.OnClickListener() {
    public void onClick(View v) {
        // 동작할 내용을 코딩
    }
});
```

## □ 버튼과 에디트텍스트[4/12]

### ▣ 에디트텍스트

- 값을 입력받은 후 해당 값을 Java 코드에 가져와서 사용하는 용도로 많이 사용

```
java.lang.Object
└─ android.view.View
    └─ android.widget.TextView
        └─ android.widget.EditText
```

EditText 계층도

- **버튼과 에디트텍스트[5/12]**
  - ▣ **일반적인 에디트텍스트의 XML 코드**

```
<EditText  
    android:id="@+id/edittext1" />
```

① 에디트텍스트 변수 선언

```
EditText myEdit;
```

② 변수에 에디트텍스트 위젯 대입

```
myEdit = (EditText) findViewById(R.id.edittext1);
```

③ 에디트텍스트에 입력된 값을 가져오기 → 주로 버튼 클릭 이벤트 리스너 안에 넣음

```
String myStr = myEdit.getText().toString();
```

## □ 버튼과 에디트텍스트[6/12]

### 실습 4-1 초간단 계산기 앱 만들기

- 안드로이드 프로젝트 생성
  - 프로젝트 이름 : Project4\_1
  - 패키지 이름 : com.cookandroid.project4\_1
  
- 화면 디자인 및 편집
  - EditText 2개, Button 4개, TextView 1개 생성
  - 각 위젯에 layout\_margin을 적절히 지정
  - TextView는 색상을 빨간색, 글자 크기를 30에
  - 각 위젯의 id는 위에서부터 차례로 Edit1, Edit2, BtnAdd, BtnSub, BtnMul, BtnDiv, TextRes

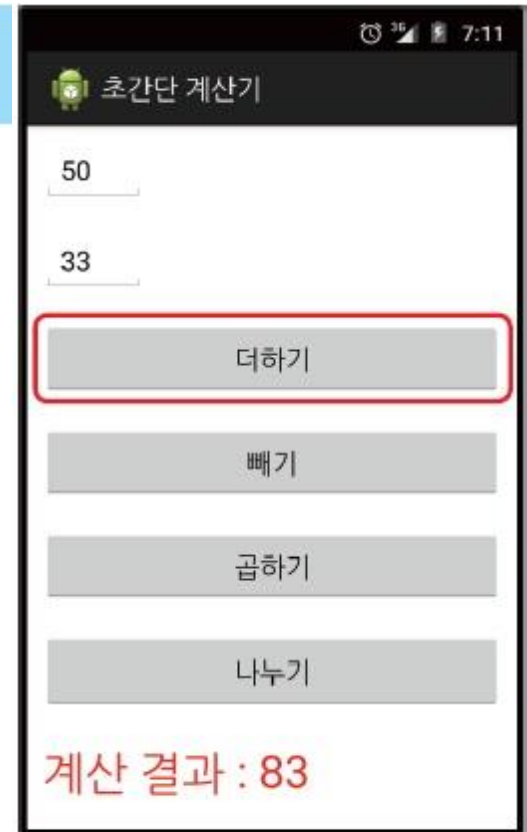


그림 4-5 초간단 계산기 결과 화면

## □ 버튼과 에디트텍스트[7/12]

### ▣ 화면 디자인 및 편집

예제 4-19 activity\_main.xml

```

1 <LinearLayout>
2   <EditText
3     android:id="@+id/Edit1"
4     android:layout_width="wrap_content"
5     android:layout_height="wrap_content"
6     android:layout_margin="10dp"
7     android:hint="숫자1" />
8   ~~~~ 중간 생략 (에디트텍스트 1개) ~~~~
9   <Button
10    android:id="@+id/BtnAdd"
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content"
13    android:layout_margin="10dp"
14    android:text="더하기" />
15   ~~~~ 중간 생략 (버튼 3개) ~~~~
16   <TextView
17     android:id="@+id/TextResult"
18     android:layout_width="wrap_content"
19     android:layout_height="wrap_content"
20     android:textSize="30dp"
21     android:textColor="#FF0000"
22     android:layout_margin="10dp"
23     android:text="계산 결과 : " />
24 </LinearLayout>

```

숫자1

숫자2

더하기

빼기

곱하기

나누기

계산 결과 :



## □ 버튼과 에디트텍스트[8/12]

### ▣ Java 코드 작성 및 수정

- activity\_main.xml의 7개 위젯에 대응할 위젯 변수 7개
- 입력될 2개 문자열을 저장할 문자열 변수 2개
- 계산 결과를 저장할 정수 변수 1개

#### 예제 4-20 Java 코드 1

```
1  ~~~~ 중간생략 ~~~~
2  public class MainActivity extends Activity {
3      EditText edit1, edit2;
4      Button btnAdd, btnSub, btnMul, btnDiv;
5      TextView textResult;
6      String num1, num2;
7      Integer result;
8
9      @Override
10     public void onCreate(Bundle savedInstanceState) {
11         ~~~~ 중간생략 ~~~~
```

## □ 버튼과 에디트텍스트[9/12]

### ▣ Java 코드 작성 및 수정

- 에디트텍스트 2개를 변수에 대입
- 버튼(더하기) 1개를 변수에 대입
- 텍스트뷰 1개를 변수에 대입

#### 예제 4-21 Java 코드 2

```
1 public void onCreate(Bundle savedInstanceState) {  
2     super.onCreate(savedInstanceState);  
3     setContentView(R.layout.activity_main);  
4     setTitle("초간단 계산기");  
5  
6     edit1 = (EditText) findViewById(R.id.Edit1);  
7     edit2 = (EditText) findViewById(R.id.Edit2);  
8     btnAdd = (Button) findViewById(R.id.BtnAdd);  
9     textResult = (TextView) findViewById(R.id.TextResult);  
10 }
```

## □ 버튼과 에디트텍스트[10/12]

### ▣ Java 코드 작성 및 수정

- 버튼에 터치 이벤트 리스너 정의
- 터치 시에 동작하는 내용을 onTouch( ) 메소드 안에 코딩

#### 예제 4-22 Java 코드 3

```
1 btnAdd.setOnTouchListener(new View.OnTouchListener() {  
2     public boolean onTouch(View arg0, MotionEvent arg1) {  
3         num1 = edit1.getText().toString();  
4         num2 = edit2.getText().toString();  
5         result = Integer.parseInt(num1) + Integer.parseInt(num2);  
6         textResult.setText("계산 결과 : " + result.toString());  
7         return false;  
8     }  
9 });
```

## □ 버튼과 에디트텍스트[11/12]

### ▣ Java 코드 작성 및 수정

#### ■ 빼기, 곱하기, 나누기 코드 추가

예제 4-23 완성된 Java 코드

```

1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4     setTitle("초간단 계산기");
5
6     edit1 = (EditText) findViewById(R.id.Edit1);
7     edit2 = (EditText) findViewById(R.id.Edit2);
8
9     btnAdd = (Button) findViewById(R.id.BtnAdd);
10    btnSub = (Button) findViewById(R.id.BtnSub);
11    btnMul = (Button) findViewById(R.id.BtnMul);
12    btnDiv = (Button) findViewById(R.id.BtnDiv);
13
14    textResult = (TextView) findViewById(R.id.TextResult);
15
16    btnAdd.setOnTouchListener(new View.OnTouchListener() {
17        public boolean onTouch(View arg0, MotionEvent arg1) {
18            num1 = edit1.getText().toString();
19            num2 = edit2.getText().toString();
20            result = Integer.parseInt(num1) + Integer.parseInt(num2);
21            textResult.setText("계산 결과 : " + result.toString());
22            return false;
23        }
24    });

```

```

25
26    btnSub.setOnTouchListener(new View.OnTouchListener() {
27        public boolean onTouch(View arg0, MotionEvent arg1) {
28            num1 = edit1.getText().toString();
29            num2 = edit2.getText().toString();
30            result = Integer.parseInt(num1) - Integer.parseInt(num2);
31            textResult.setText("계산 결과 : " + result.toString());
32            return false;
33        }
34    });
35
36    btnMul.setOnTouchListener(new View.OnTouchListener() {
37        public boolean onTouch(View arg0, MotionEvent arg1) {
38            num1 = edit1.getText().toString();
39            num2 = edit2.getText().toString();
40            result = Integer.parseInt(num1) * Integer.parseInt(num2);
41            textResult.setText("계산 결과 : " + result.toString());
42            return false;
43        }
44    });
45
46    btnDiv.setOnTouchListener(new View.OnTouchListener() {
47        public boolean onTouch(View arg0, MotionEvent arg1) {
48            num1 = edit1.getText().toString();
49            num2 = edit2.getText().toString();
50            result = Integer.parseInt(num1) / Integer.parseInt(num2);
51            textResult.setText("계산 결과 : " + result.toString());
52            return false;
53        }
54    });
55 }

```

## □ 버튼과 에디트텍스트[12/12]

### ▶ 직접 물어보기 4-3

[실습 4-1]에 다음과 같이 기능을 추가하거나 변경해보자.

- 터치(Touch)가 아닌 클릭(Click)으로 변경
- 나머지 값 구하기 버튼 추가
- 값을 입력하지 않고, 버튼을 클릭할 때 오류 메시지를 토스트로 나타내기
- 실수 값 계산하기
- 0으로 나누면 토스트 메시지를 나타내고 계산하지 않기

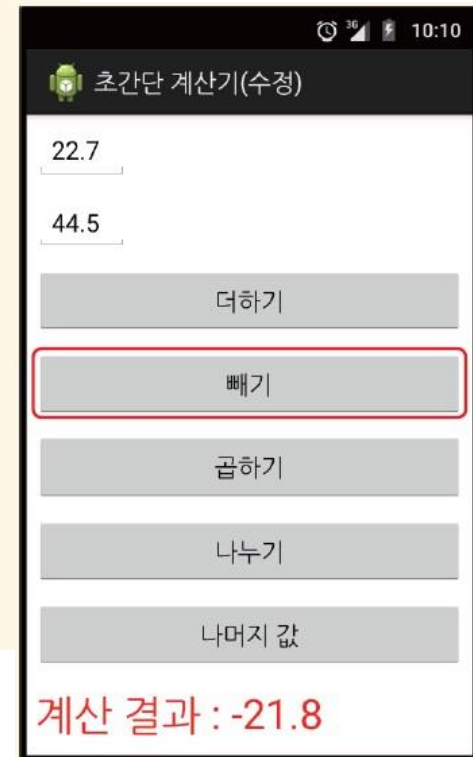


그림 4-6 개선된 계산기

# 3. 기본 위젯 활용하기

38

## □ 컴파운드버튼[1/4]

### ▣ 컴파운드버튼

- Button 클래스의 하위 클래스
- 체크박스(CheckBox), 라디오버튼(RadioButton), 스위치(Switch), 토글버튼(ToggleButton)의 상위 클래스

```
java.lang.Object
└ android.view.View
    └ android.widget.TextView
        └ android.widget.Button
            └ android.widget.CompoundButton
                └ android.widget.CheckBox
                └ android.widget.RadioButton
                └ android.widget.Switch
                └ android.widget.ToggleButton
```

CompoundButton 계층도

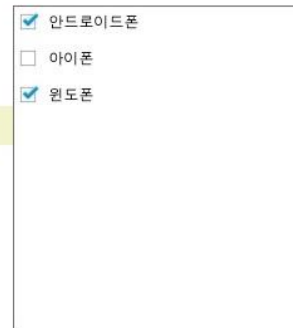
## □ 컴파운드버튼[2/4]

### ▣ 체크박스

```

1 <CheckBox
2     android:id="@+id/android"
3     android:text="안드로이드폰"
4     android:checked="true"/>
5 <CheckBox
6     android:id="@+id/iphone"
7     android:text="아이폰" />
8 <CheckBox
9     android:id="@+id/window"
10    android:text="윈도폰"
11    android:checked="true" />

```



#### ❶ 체크박스 변수 선언

```
CheckBox mycheck;
```

#### ❷ 변수에 체크박스 위젯 대입

```
mycheck = (CheckBox) findViewById(R.id.android);
```

#### ❸ 체크박스가 변경될 때 동작하는 클래스 정의

```

mycheck.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton arg0, boolean arg1) {
        // 이 부분에 동작할 내용을 코딩
    }
});

```



## □ 컴파운드버튼[3/4]

### ▣ 스위치, 토글버튼

- 스위치와 토글버튼은 모양만 조금 다를 뿐 용도는 거의 동일

#### 예제 4-25 Switch와 ToggleButton의 XML 코드

```
1 <Switch
2     android:checked="true" />
3 <Switch
4     android:checked="false" />
5 <ToggleButton
6     android:checked="true" />
7 <ToggleButton
8     android:checked="false" />
```



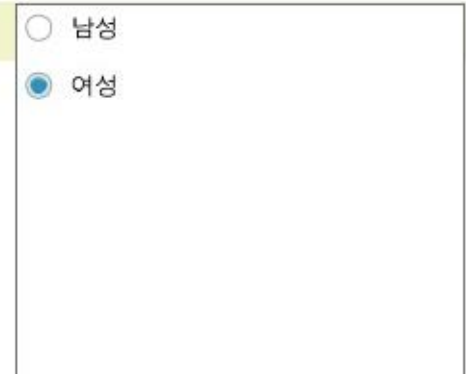


## □ 컴파운드버튼[4/4]

### ▣ 라디오버튼과 라디오그룹

- 라디오버튼 : 여러 개 중 하나만 선택해야 하는 경우에 사용
- 라디오그룹 : 라디오버튼만 여러 개 나열하면 클릭하는 것마다 모두 중복 선택이 되므로 라디오그룹과 함께 사용해야 함

```
1 <RadioGroup
2     android:id="@+id/rGroup1" >
3     <RadioButton
4         android:text="남성" />
5     <RadioButton
6         android:text="여성" />
7 </RadioGroup>
```



○ 남성

● 여성

## □ 이미지뷰와 이미지버튼[1/11]

### ▣ 이미지뷰(Image View)

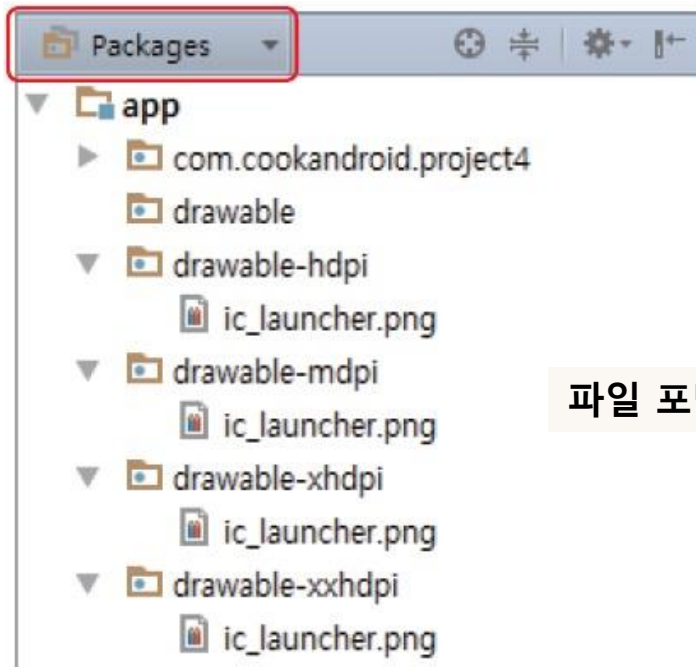
- 그림을 출력하는 위젯으로 그림이 필요하거나 화면을 화려하게 구성할 때 사용
- 이미지뷰에 보여줄 그림 파일은 프로젝트의 [res]-[drawable]에 있어야 함

```
java.lang.Object
└ android.view.View
    └ android.widget.ImageView
        └ android.widget.ImageButton
```

ImageView 계층도

## □ 이미지뷰와 이미지버튼[2/11]

- ▣ 이미지뷰 및 이미지버튼의 XML 속성
  - src : 이미지의 경로를 나타냄
  - MaxHeight/maxLength : 이미지의 크기를 지정
  - scaleType : 이미지의 확대/축소 방식 지정



파일 포맷은 png, jpg, gif를 지원하지만 png나 jpg 권장

그림 4-7 한 이미지를 네 가지 해상도로 저장한 drawable 리소스(Packages 모드로 볼 때)

- 이미지뷰와 이미지버튼[3/11]
  - ▣ 이미지뷰와 이미지버튼의 XML 코드

```
1 <ImageView
2     android:src="@drawable/ic_launcher" />
3 <ImageButton
4     android:src="@drawable/ic_launcher" />
5 <ImageView
6     android:layout_width="200dp"
7     android:layout_height="100dp"
8     android:scaleType="fitXY"
9     android:src="@drawable/ic_launcher" />
10 <ImageView
11     android:layout_width="200dp"
12     android:layout_height="100dp"
13     android:scaleType="fitCenter"
14     android:src="@drawable/ic_launcher" />
```



## □ 이미지뷰와 이미지버튼[4/11]

### 실습 4-2 좋아하는 애완동물 선택 앱 만들기

- 안드로이드 프로젝트 생성
  - 프로젝트 이름 : Project4\_2
  - 패키지 이름 : com.cookandroid.project4\_2



## □ 이미지뷰와 이미지버튼[5/11]

### 실습 4-2 좋아하는 애완동물 선택 앱 만들기

#### ❖ 화면 및 디자인 편집

- ✓ 프로젝트의 [res]-[drawable]에 강아지, 고양이, 토끼 그림 파일을 미리 복사

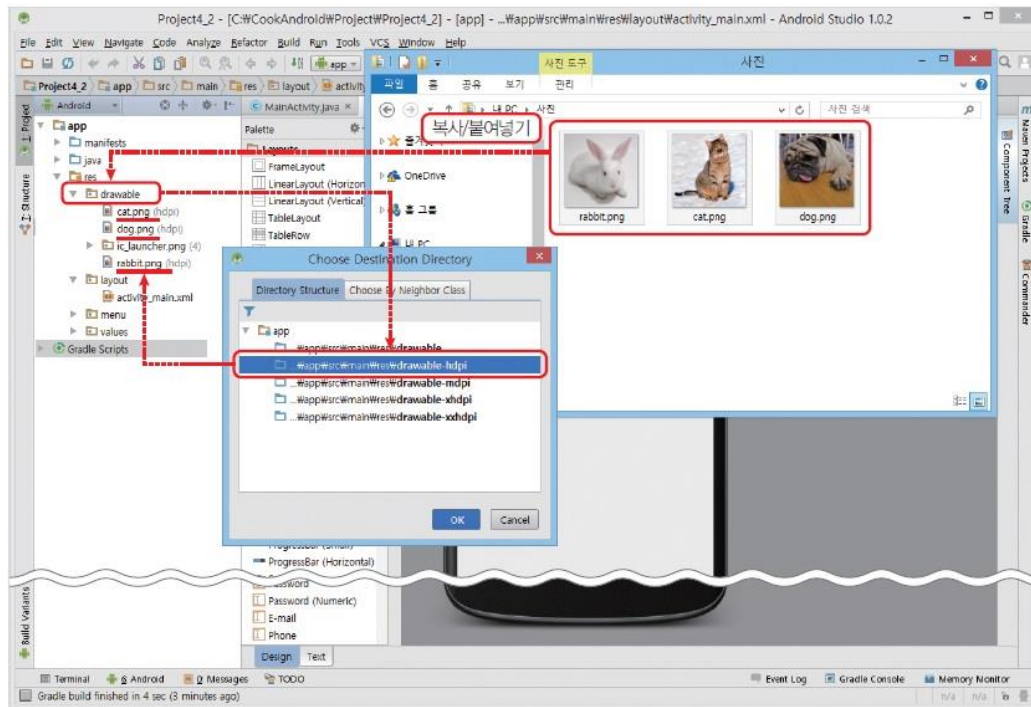


그림 4-9 그림 파일 복사

## □ 이미지뷰와 이미지버튼[6/11]

### 실습 4-2 좋아하는 애완동물 선택 앱 만들기

#### □ 화면 및 디자인 편집

- TextView, CheckBox, TextView, RadioGroup, RadioButton 각각 세 개, Button, ImageView의 차례로 만들 레이아웃에 padding을 적절히 지정
- 위의 TextView와 CheckBox를 제외하고, 나머지 위젯은 visibility 속성을 invisible로 지정
- 위젯의 id는 위에서부터 Text1, ChkAgree, Text2, Rgroup1, RdoDog, RdoCat, RdoRabbit, BtnOK, ImgPet으로 함

예제 4-28 activity\_main.xml 코드

```

1 <TextView
2     android:id="@+id/Text1"
3     android:text="선택을 시작하겠습니까?" />
4 <CheckBox
5     android:id="@+id/ChkAgree"
6     android:text="시작함" />
7 <TextView
8     android:text="좋아하는 애완동물은?"
9     android:visibility="invisible" />
10 <RadioGroup
11     android:id="@+id/Rgroup1"
12     android:visibility="invisible" >
13     <RadioButton
14         android:id="@+id/RdoDog"
15         android:text="강아지" />
16     ~~~~ 중간 생략 (라디오버튼 2개) ~~~~
17 </RadioGroup>
18 <Button
19     android:id="@+id/BtnOK"
20     android:text="선택 완료"
21     android:visibility="invisible" />
22 <ImageView
23     android:id="@+id/ImgPet"
24     android:visibility="invisible" />

```

선택을 시작하겠습니까?  
☐ 시작함

## □ 이미지뷰와 이미지버튼[7/11]

### ▣ Java 코드 작성 및 수정

- activity\_main.xml의 9개 위젯에 대응할 위젯 변수 9개

#### 예제 4-29 Java 코드 1

```
1  ~~~~ 중간 생략 ~~~~
2  public class MainActivity extends Activity {
3      TextView text1, text2;
4      CheckBox chkAgree;
5      RadioGroup rGroup1;
6      RadioButton rdoDog, rdoCat, rdoRabbit;
7      Button btnOK;
8      ImageView imgPet;
9
10     @Override
11     public void onCreate(Bundle savedInstanceState) {
12     ~~~~ 중간 생략 ~~~~
```



## □ 이미지뷰와 이미지버튼[8/11]

### ❖ Java 코드 작성 및 수정

- ✓ 각 위젯을 변수에 대입

예제 4-30 Java 코드 2

```
1 public void onCreate(Bundle savedInstanceState) {  
2     super.onCreate(savedInstanceState);  
3     setContentView(R.layout.activity_main);  
4     setTitle("애완동물 사진 보기");  
5  
6     text1 = (TextView) findViewById(R.id.Text1);  
7     chkAgree = (CheckBox) findViewById(R.id.ChkAgree);  
8  
9     text2 = (TextView) findViewById(R.id.Text2);  
10    rGroup1 = (RadioGroup) findViewById(R.id.Rgroup1);  
11    rdoDog = (RadioButton) findViewById(R.id.RdoDog);  
12    rdoCat = (RadioButton) findViewById(R.id.RdoCat);  
13    rdoRabbit = (RadioButton) findViewById(R.id.RdoRabbit);  
14  
15    btnOK = (Button) findViewById(R.id.BtnOK);  
16    imgPet = (ImageView) findViewById(R.id.ImgPet);  
17 }
```

## □ 이미지뷰와 이미지버튼[9/11]

### ▣ Java 코드 작성 및 수정

- <시작함> 체크박스를 체크/언체크할 때 동작하는 리스너를 onCreate( ) 내부에 정의

예제 4-31 Java 코드 3

```
1  chkAgree.setOnCheckedChangeListener(new CompoundButton.  
    OnCheckedChangeListener() {  
2      public void onCheckedChanged(CompoundButton arg0, boolean arg1) {  
3  
4          if (chkAgree.isChecked() == true) {  
5              text2.setVisibility(android.view.View.VISIBLE);  
6              rGroup1.setVisibility(android.view.View.VISIBLE);  
7              btnOK.setVisibility(android.view.View.VISIBLE);  
8              imgPet.setVisibility(android.view.View.VISIBLE);  
9          } else  
10         {  
11             text2.setVisibility(android.view.View.INVISIBLE);  
12             rGroup1.setVisibility(android.view.View.INVISIBLE);  
13             btnOK.setVisibility(android.view.View.INVISIBLE);  
14             imgPet.setVisibility(android.view.View.INVISIBLE);  
15         }  
16     }  
17 });
```

## □ 이미지뷰와 이미지버튼[10/11]

### ▣ Java 코드 작성 및 수정

- <선택 확인>을 클릭하면 동작하는 리스너를 onCreate( ) 메소드 내부에 정의

예제 4-32 Java 코드 4

```
1 btnOK.setOnClickListener(new View.OnClickListener () {  
2     public void onClick(View arg0) {  
3         switch(rGroup1.getCheckedRadioButtonId()) {  
4             case R.id.RdoDog:  
5                 imgPet.setImageResource(R.drawable.dog);  
6                 break;  
7             case R.id.RdoCat:  
8                 imgPet.setImageResource(R.drawable.cat);  
9                 break;  
10            case R.id.RdoRabbit:  
11                imgPet.setImageResource(R.drawable.rabbit);  
12                break;  
13            default:  
14                Toast.makeText(getApplicationContext(), "동물 먼저 선택하세요", Toast.  
15                    LENGTH_SHORT).show();  
16            }  
17 });
```

## □ 이미지뷰와 이미지버튼[11/11]

### ▶ 직접 풀어보기 4-4

[실습 4-2]를 다음과 같이 수정해보자.

- “좋아하는 안드로이드 버전은?”으로 질문 변경
- <시작함>을 Switch로 변경
- <선택 완료>를 없애고, 라디오버튼을 선택할 때마다 즉시 해당 이미지가 나오도록 변경
- 제일 마지막에 <종료>와 <처음부터>를 추가하고 <종료> 클릭시에는 응용프로그램이 완전히 종료되도록 함. <처음부터> 클릭 시에는 다시 초기화가 되고 처음 화면이 나오도록 함.



그림 4-10 안드로이드 사진 보기 프로젝트