

기본 위젯 익히기

학습목표

- 뷰와 뷰 상속을 이해한다.
- 기본 위젯을 다루는 방법을 익힌다.
- 안드로이드 앱의 기본적인 프로그래밍을 숙달한다.

목차

01 뷰의 개요

02 기본 위젯 다루기

03 기본 위젯 활용하기

01

뷰의 개요

1. 뷰와 뷰그룹

■ View 클래스

- 안드로이드 화면에서 실제로 사용되는 것들은 모두 View 클래스 상속을 받음
- 다른 말로 '위젯'이라고도 함
 - 화면에서 버튼 → 버튼 위젯 / 실제 코드에서 버튼 → 버튼 클래스
- 다른 위젯을 담을 수 있는 위젯은 레이아웃이라고 함
- 레이아웃은 ViewGroup 클래스 아래 존재
- 레이아웃도 크게 보면 위젯에 포함됨

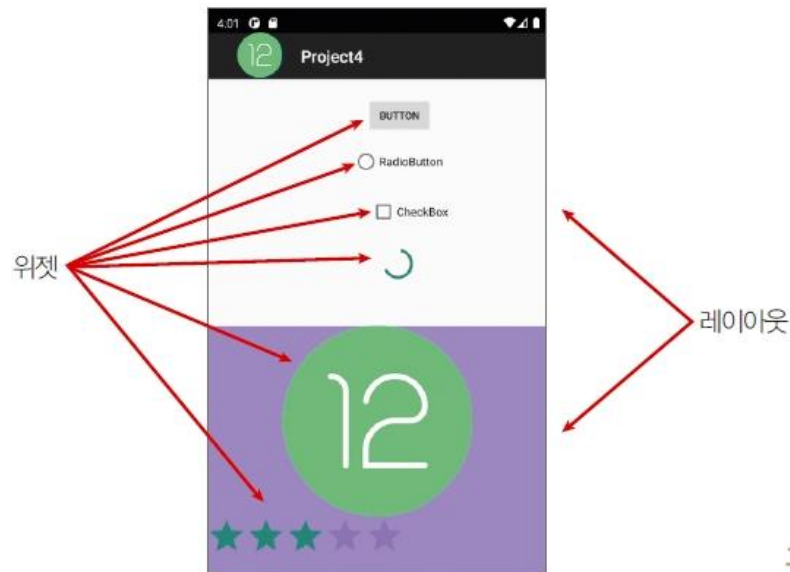


그림 4-1 위젯과 레이아웃

1. 뷰와 뷰그룹

■ View 클래스의 상속을 받은 클래스(위젯) 계층도

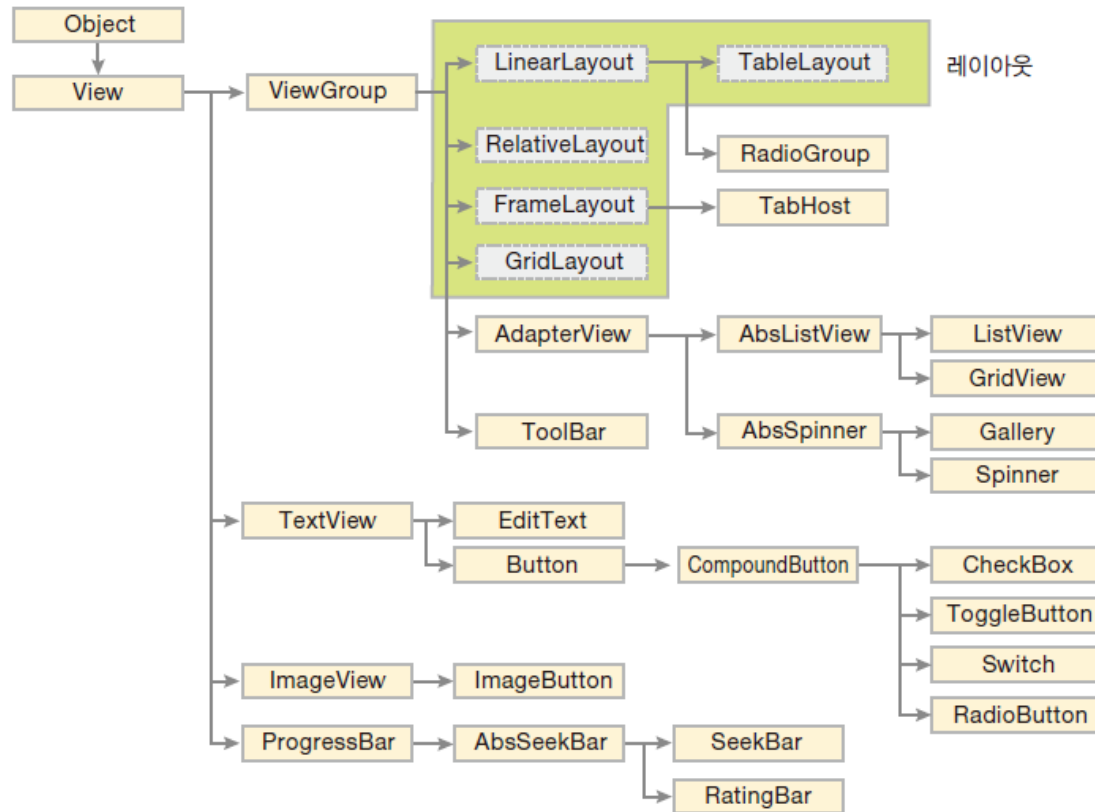


그림 4-2 안드로이드의 View 클래스 계층도

1. 뷰와 뷰그룹

■ Button의 XML 속성

- TextView나 View 클래스에서 상속받음
- 이럴 때는 TextView를 클릭해서 확인해봐야 하며 이후 안드로이드에서 사용되는 클래스에 대한 내용은 이러한 방식으로 찾아볼 수 있음
- 특히 화면을 구성하는 뷰는 대부분 android.widget 패키지에 포함되므로 이 부분을 살펴봐야 함

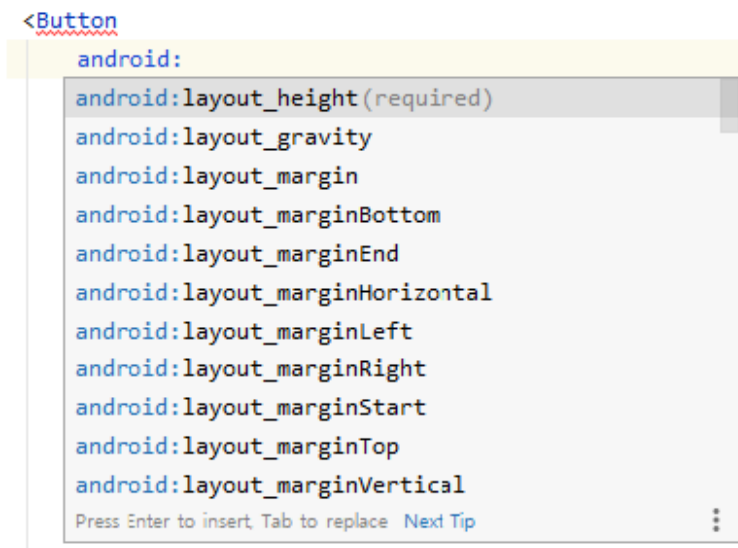


그림 4-3 Button의 XML 속성

1. 뷰와 뷰그룹

■ 클래스 상속 관계 찾기

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

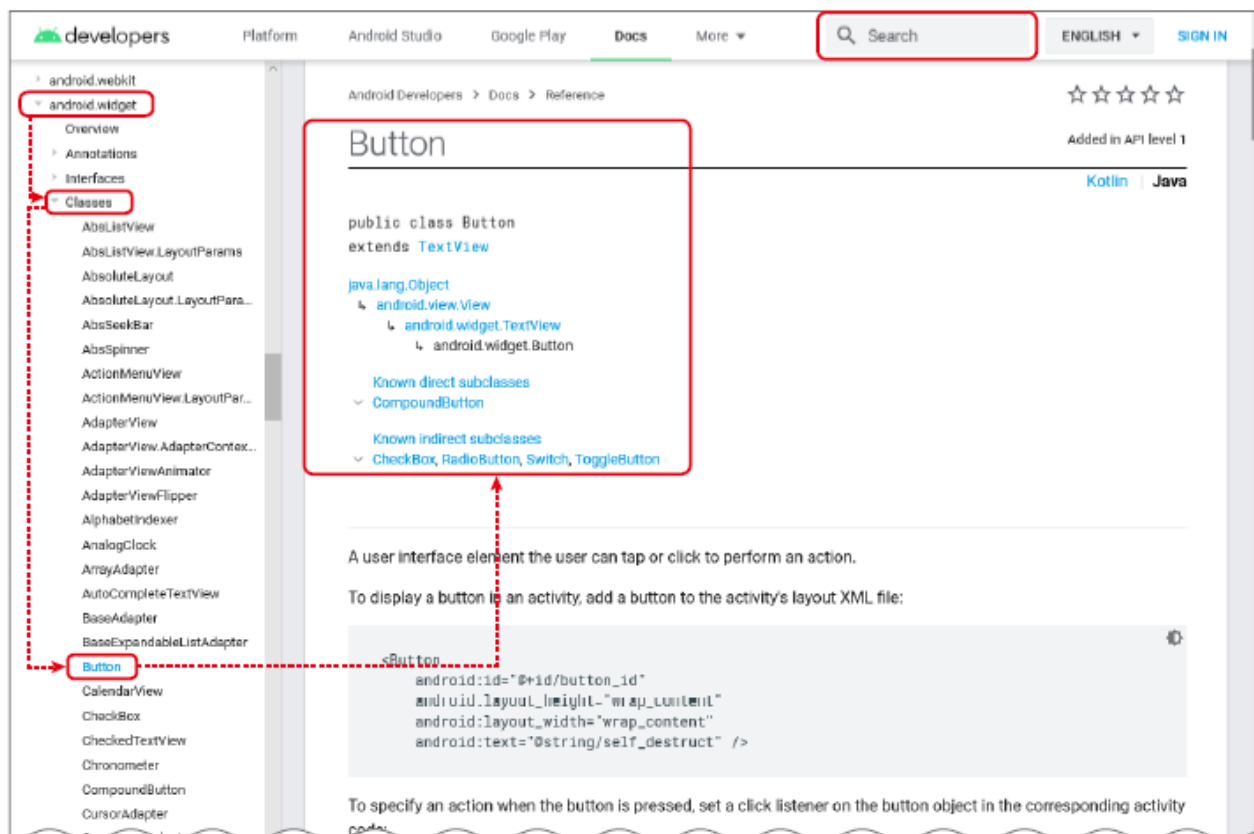


그림 4-4 클래스의 상속 관계를 찾는 방법

1. 뷰와 뷰그룹

▶ **작업 풀어보기 4-1**

<http://developer.android.com/reference>에서 ScrollView, DigitalClock, CalendarView의 각 클래스 상속 관계를 찾아보자.

2. View 클래스의 XML 속성

■ View 클래스로부터 상속받은 Button의 XML 속성

```
<Button
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#ff0000"
    android:text="버튼입니다"
/>
```

2. View 클래스의 XML 속성

■ id 속성

- id 속성은 모든 위젯의 아이디를 나타냄
- Java 코드에서 위젯에 접근할 때 id 속성에 지정한 아이디 사용
- id 속성은 위젯에 아이디를 새로 부여하는 개념이므로 “@+id/” 형식으로 지정
- / 다음에는 새로 지정할 id를 적음

→ android:id="@+id/btn1"의 의미 : 버튼 위젯의 아이디를 btn1로 부여한 것

```
위젯 변수 = (위젯형) findViewById(R.id.위젯id);
```

```
Button button1;  
button1 = (Button) findViewById(R.id.btn1);
```

2. View 클래스의 XML 속성

■ id 속성의 지정 유/무

- Button, RadioButton, CheckBox 등의 위젯은 일반적으로 클릭 또는 터치했을 때 어떤 동작을 하기 위한 것이므로 id 속성을 지정함
- 하지만 클릭이나 터치를 해도 아무 동작이 필요 없는 텍스트뷰, 이미지뷰 등은 굳이 id 속성을 지정하지 않아도 괜찮음

예제 4-1 id 속성의 XML 코드

```
1 <LinearLayout    ~~~ 중간 생략 ~~~
2     android:orientation="vertical" >
3     <TextView
4         android:id="@+id/textView1"
5         android:layout_width="wrap_content"
6         android:layout_height="wrap_content"
7         android:text="성별 선택" />
8     <RadioButton
9         android:id="@+id/female"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="여성" />
13    <RadioButton
14        android:id="@+id/male"
15        android:layout_width="wrap_content"
16        android:layout_height="wrap_content"
17        android:text="남성" />
18 </LinearLayout>
```

성별 선택
☐ 여성
☒ 남성

2. View 클래스의 XML 속성

■ layout_width, layout_height 속성

- match_parent : 자신의 부모(대개는 레이아웃)에 폭이나 높이를 맞춤
- wrap_content : 자신의 폭이나 높이를 자신 안의 글자가 들어갈 정도로만 설정

예제 4-2 layout_width, layout_height 속성의 XML 코드 1

```
1 <LinearLayout
2     ~~~~ 중간 생략 ~~~~ >
3     <Button
4         android:layout_width="wrap_content"
5         android:layout_height="wrap_content"
6         android:text="버튼입니다" />
7 </LinearLayout>
```

버튼입니다

부모 레이아웃

예제 4-3 layout_width, layout_height 속성의 XML 코드 2

```
1 <LinearLayout
2     ~~~~ 중간 생략 ~~~~ >
3     <Button
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content"
6         android:text="버튼입니다" />
7 </LinearLayout>
```

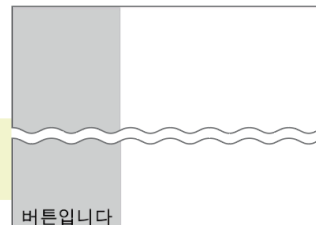
버튼입니다

2. View 클래스의 XML 속성

■ layout_width, layout_height 속성

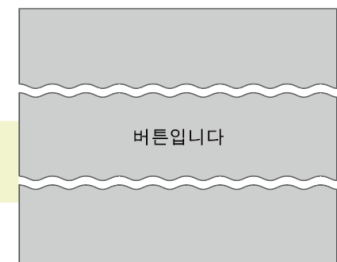
예제 4-4 layout_width, layout_height 속성의 XML 코드 3

```
1 <LinearLayout
2     ~~~~ 중간 생략 ~~~~ >
3     <Button
4         android:layout_width="wrap_content"
5         android:layout_height="match_parent"
6         android:text="버튼입니다" />
7 </LinearLayout>
```



예제 4-5 layout_width, layout_height 속성의 XML 코드 4

```
1 <LinearLayout
2     ~~~~ 중간 생략 ~~~~ >
3     <Button
4         android:layout_width="match_parent"
5         android:layout_height="match_parent"
6         android:text="버튼입니다" />
7 </LinearLayout>
```



예제 4-6 layout_width, layout_height 속성의 XML 코드 5

```
1 <LinearLayout
2     ~~~~ 중간 생략 ~~~~ >
3     <Button
4         android:layout_width="1080px"
5         android:layout_height="1920px"
6         android:text="버튼입니다" />
7 </LinearLayout>
```



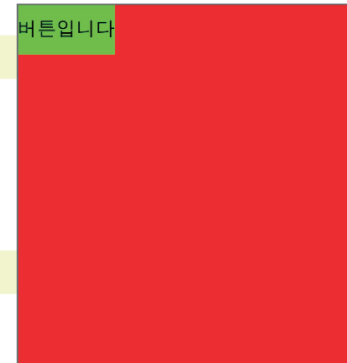
2. View 클래스의 XML 속성

■ background 속성

- background 속성은 위젯의 색상을 주로 #RRGGBB 값으로 지정
- 각 값은 빨간색, 초록색, 파란색을 의미함
- RR, GG, BB의 위치는 16진수 00~FF로 표현할 수 있음

예제 4-7 background 속성의 XML 코드

```
1 <LinearLayout
2   android:background="#ff0000"
3   ~~~~ 중간 생략 ~~~~ >
4   <Button
5     android:layout_width="wrap_content"
6     android:layout_height="wrap_content"
7     android:background="#00ff00"
8     android:text="버튼입니다" />
9 </LinearLayout>
```



2. View 클래스의 XML 속성

저자
한마디

책의 XML 코드 표기(★주의)

텍스트뷰 1개와 버튼 1개가 있는 기본적인 activity_main.xml의 전체 코드는 다음과 같다.

예제 4-8 전체 XML 코드

```
1 <LinearLayout xmlns:android=  
2             "http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:layout_width="match_parent"  
5     android:layout_height="match_parent"  
6     android:orientation="vertical" >  
7  
8     <TextView  
9         android:layout_width="match_parent"  
10        android:layout_height="wrap_content"  
11        android:text="Hello World!" />  
12  
13    <Button  
14        android:id="@+id/button1"  
15        android:layout_width="wrap_content"  
16        android:layout_height="wrap_content"  
17        android:text="Button" />  
18  
19 </LinearLayout>
```


2. View 클래스의 XML 속성

1~3행의 xmlns 이후에는 항상 동일한 내용이 나오고 4행, 5행, 9행, 10행, 15행, 16행에 나오는 layout_width와 layout_height는 모든 위젯의 필수 요소로 자주 등장할 것이다. 6행의 orientation 속성도 거의 고정되어 나온다. 따라서 이 책에서 이러한 반복적인 속성은 꼭 표현할 필요가 있을 때를 제외하고는 생략할 것이다. 즉 앞으로는 다음과 같이 필수 또는 반복되는 코드를 생략하고 간략하게 표현한다.

예제 4-9 앞으로 책에서 표기할 XML 코드

```
1 <LinearLayout>
2
3     <TextView
4         android:text="Hello World!" />
5
6     <Button
7         android:id="@+id/button1"
8         android:text="Button" />
9
10 </LinearLayout>
```

<LinearLayout>과 </LinearLayout>도 따로 추가할 설명이 없다면 생략하겠다. 간단히 추려놓으니 확인해야 할 코드가 명확하게 보인다. 앞으로는 이런 식으로 XML 코드를 표기하지만 실제로는 전체 코드를 다 작성해야 동작함을 잊지 말기 바란다. 주의할 점은 [예제 4-8] 6행의 orientation 속성을 생략하면 LinearLayout이 horizontal이 된다는 것이다. 이 책의 화면은 대부분 vertical 정렬이므로 예제에는 생략되어 있어도 6행을 잊지 말고 써야 한다. 이는 5장에서 다시 설명하겠다. 생략하지 않은 모든 소스코드는 자료실(<http://www.hanbit.co.kr/src/4584/>)에 올려놓았다.

2. View 클래스의 XML 속성

■ padding 속성

- padding은 위젯의 경계선으로부터 위젯 안의 요소가 떨어지도록 설정

예제 4-10 간격이 없는 XML 코드

```
1 <LinearLayout >
2     <TextView
3         android:text="아래에 이름을 입력 : " />
4     <EditText
5         android:hint="여기에 채우세요" />
6     <Button
7         android:text="확인" />
8 </LinearLayout>
```

아래에 이름을 입력 :

여기에 채우세요

확인

예제 4-11 padding 속성의 XML 코드

```
1 <LinearLayout
2     android:padding="30dp" >
3     <TextView
4         android:text="아래에 이름을 입력 : " />
5     <EditText
6         android:hint="여기에 채우세요" />
7     <Button
8         android:text="확인" />
9 </LinearLayout>
```

아래에 이름을 입력 :

여기에 채우세요

확인

2. View 클래스의 XML 속성

■ layout_margin 속성

- 위젯과 위젯 사이에 여유를 두고 싶다면 layout_margin 속성을 사용

예제 4-12 layout_margin 속성의 XML 코드

```
1 <LinearLayout
2     android:padding="30dp" >
3     <TextView
4         android:layout_margin="20dp"
5         android:text="아래에 이름을 입력 :" />
6     <EditText
7         android:layout_margin="20dp"
8         android:hint="여기에 채우세요" />
9     <Button
10        android:layout_margin="20dp"
11        android:text="확인" />
12 </LinearLayout>
```

아래에 이름을 입력 :

여기에 채우세요

확인

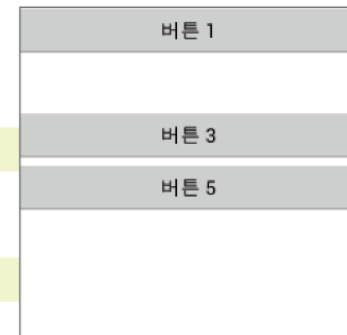
2. View 클래스의 XML 속성

■ visibility 속성

- visibility 속성은 위젯을 보일 것인지 여부를 결정
- 디폴트인 visible은 보이는 상태, invisible과 gone은 안 보이는 상태
- invisible : 보이지 않을 뿐 원래의 자리를 계속 유지함
- gone : 보이지 않으며 자리까지 없어짐

예제 4-13 visibility 속성의 XML 코드

```
1 <Button
2     android:text="버튼 1" />
3 <Button
4     android:visibility="invisible"
5     android:text="버튼 2" />
6 <Button
7     android:visibility="visible"
8     android:text="버튼 3" />
9 <Button
10    android:visibility="gone"
11    android:text="버튼 4" />
12 <Button
13    android:text="버튼 5" />
```



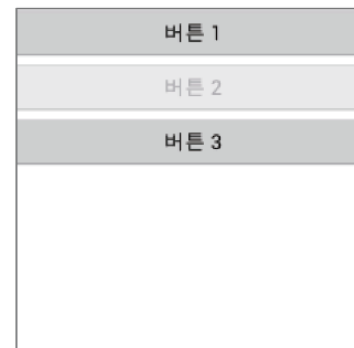
2. View 클래스의 XML 속성

■ enabled, clickable 속성

- enabled : 위젯의 동작 여부
- clickable : 클릭이나 터치가 가능하도록 함
- true와 false로 지정(디폴트 : true)
- XML 보다 Java 코드에서 주로 사용함

예제 4-14 enabled, clickable 속성의 XML 코드

```
1 <Button
2     android:text="버튼 1" />
3 <Button
4     android:enabled="false"
5     android:text="버튼 2" />
6 <Button
7     android:clickable="false"
8     android:text="버튼 3" />
```



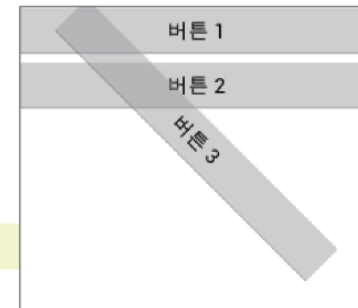
2. View 클래스의 XML 속성

■ rotation 속성

- rotation은 위젯을 회전시켜서 출력
- 값은 각도로 지정

예제 4-15 rotation 속성의 XML 코드

```
1 <Button
2     android:text="버튼 1" />
3 <Button
4     android:text="버튼 2" />
5 <Button
6     android:rotation="45"
7     android:text="버튼 3" />
```



2. View 클래스의 XML 속성

▶ 직접 풀어보기 4-2

다음과 같은 화면을 XML로 코딩하라. 버튼, 텍스트뷰, 에디트텍스트, 버튼을 차례로 지정하고 앞에서 배운 다양한 속성을 사용한다.

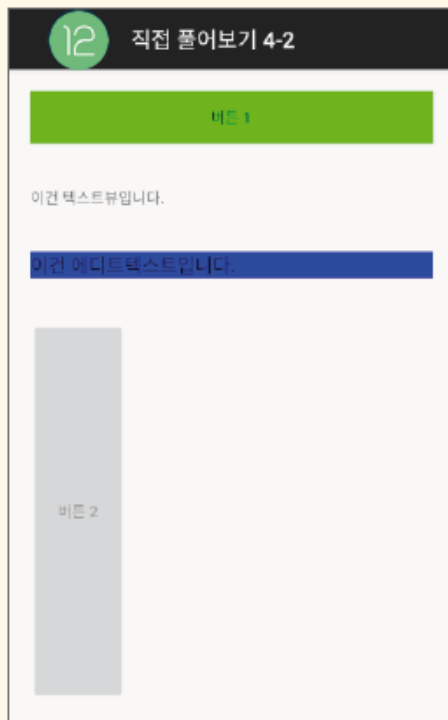


그림 4-5 다양한 XML 속성을 적용한 화면

02

기본 위젯 다루기

1. 텍스트뷰

■ 텍스트뷰

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

텍스트뷰 계층도

■ 텍스트뷰의 속성

- text : 텍스트뷰에 나타나는 문자열 표현
- textColor : 글자의 색상을 지정, #RRGGBB나 #AARRGGBB 형식
- textSize : 글자의 크기를 dp, px, in, mm, sp 단위로 지정
- typeface : 글자의 글꼴을 지정
 - 값으로 sans, serif, monospace 설정, 디폴트는 normal
- textStyle : 글자의 스타일을 지정
 - 값으로 bold, italic, bold|italic을 설정, 디폴트는 normal
- singleLine : 글이 길어 줄이 넘어갈 경우 강제로 한 줄만 출력하고 문자열의 맨 뒤에 '...'를 표시
 - 값으로 true와 false를 설정할 수 있으며 디폴트는 false

1. 텍스트뷰

예제 4-16 글자 관련 속성의 XML 코드

```
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 singleLine 속성" />
```

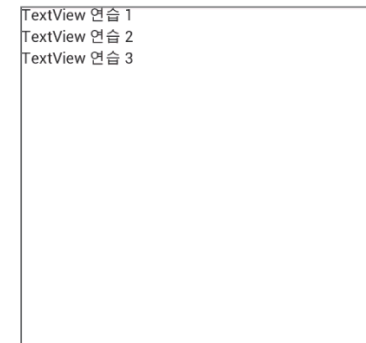
textSize 속성
textColor 속성
textStyle 속성
typeface 속성
singleLine 속성 single...

2. Java 코드로 XML 속성 설정

■ id 속성과 text만 설정한 XML 파일

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

2. Java 코드로 XML 속성 설정

■ XML 속성을 Java 코드로 설정

예제 4-18 텍스트 속성을 변경하는 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_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

2. Java 코드로 XML 속성 설정

■ 많이 사용되는 View 클래스 또는 TextView 클래스의 XML 속성과 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 |

3. 버튼과 에디트텍스트

■ 버튼과 에디트텍스트

- 사용자에게서 어떤 값을 입력받기 위한 가장 기본적인 위젯
- View와 TextView 클래스를 상속받으므로 거의 비슷하게 사용 가능

```
<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?" />
```

3. 버튼과 에디트텍스트

■ 버튼

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

버튼 계층도

■ 일반적인 버튼의 XML 코드

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

3. 버튼과 에디트텍스트

■ 버튼을 클릭했을 때 동작하는 Java 코드 3단계

① 버튼 변수 선언

- `Button mybutton;`

② 변수에 버튼 위젯 대입

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

③ 버튼을 클릭할 때 동작하는 클래스 정의

- ```
mybutton.setOnClickListener(new View.OnClickListener() {
 public void onClick(View v) {
 // 동작 내용을 이 부분에 코딩
 }
});
```



### 3. 버튼과 에디트텍스트

#### ■ 에디트텍스트

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

에디트텍스트 계층도

- 값을 입력받은 후 해당 값을 Java 코드에 가져와서 사용하는 용도로 많이 사용됨
- 일반적인 에디트텍스트의 XML 코드

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

### 3. 버튼과 에디트텍스트

#### ■ 에디트텍스트의 값을 가져오는 Java코드 3단계

① 에디트텍스트 변수 선언

- `EditText myEdit;`

② 변수에 에디트텍스트 위젯 대입

- `myEdit = (EditText) findViewById(R.id.editttext1);`

③ 에디트텍스트에 입력된 값 가져오기 → 주로 버튼 클릭 이벤트 리스너 안에 삽입

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

### 3. 버튼과 에디트텍스트

#### ■ 실습 4-1 초간단 계산기 앱 만들기

- 두 정수를 입력하고 버튼을 누르면 계산 결과가 나오는 계산기
- 가장 기본적인 위젯인 텍스트뷰, 에디트텍스트, 버튼을 이용한 앱

#### ■ 1 안드로이드 프로젝트 생성

- (1) 프로젝트 이름 : Project4\_1
- (2) 패키지 이름 : com.cookandroid.project4\_1

#### ■ 2 화면 디자인 및 편집

- (1) 에디트텍스트 2개, 버튼 4개, 텍스트뷰 1개를 생성
- (2) 각 위젯에 layout\_margin을 적절히 지정(예: 10dp)
- (3) TextView는 색상- 빨간색, 글자 크기- 30dp로 지정
- (4) 각 위젯의 id는 위에서부터 차례로 Edit1, Edit2, BtnAdd, BtnSub, BtnMul, BtnDiv, TextResult로 함

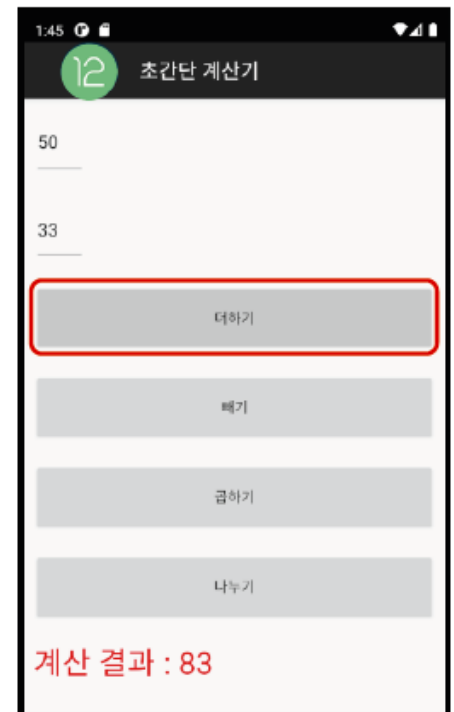


그림 4-6 초간단 계산기 앱 결과 화면

### 3. 버튼과 에디트텍스트

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 ~~~~ 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 ~~~~ 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>
```



### 3. 버튼과 에디트텍스트

#### ■ 실습 4-1 초간단 계산기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

- (1) Project Tree에서 [app]-[java]-[패키지 이름]-[MainActivity]에 아래 변수 선언
  - 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 ~~~ 중간 생략 ~~~
2 public class MainActivity extends AppCompatActivity {
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 ~~~ 중간 생략 ~~~
```

### 3. 버튼과 에디트텍스트

#### ■ 실습 4-1 초간단 계산기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

- (2) 메인 메소드인 onCreate( ) 내부를 코딩 - <더하기> 부분
  - 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 }
```

### 3. 버튼과 에디트텍스트

#### ■ 실습 4-1 초간단 계산기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

- (3) 메인 메소드인 onCreate( ) 내부를 코딩 - <더하기> 버튼 클릭했을 때 동작
  - 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 });
```

### 3. 버튼과 에디트 텍스트

---

- 실습 4-1 초간단 계산기 앱 만들기
- 4 프로젝트 실행 및 결과 확인
  - (1) [Run As]-[Run 'app'] 선택 혹은 [Run 'app'] 아이콘 클릭하여 실행
- [반복] 3 Java 코드 작성 및 수정
  - (1) 빼기, 곱하기, 나누기 코드 작성
  - 완성된 코드 : 다음 슬라이드



### 3. 버튼과 에디트 텍스트

#### ■ 실습 4-1 초간단 계산기 앱 만들기

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}
```

### 3. 버튼과 에디트텍스트

```
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 }
```

### 3. 버튼과 에디트 텍스트

#### ▶ 직접 풀어보기 4-3

[실습 4-1]을 다음과 같이 수정하라.

- 터치가 아닌 클릭으로 변경한다.
- 나머지값을 구하는 버튼을 추가한다.
- 값을 입력하지 않고 버튼을 클릭할 때 오류 메시지를 토스트 메시지로 나타낸다.
- 실숫값을 계산한다.
- 0으로 나누면 토스트 메시지를 나타내고 계산하지 않는다.

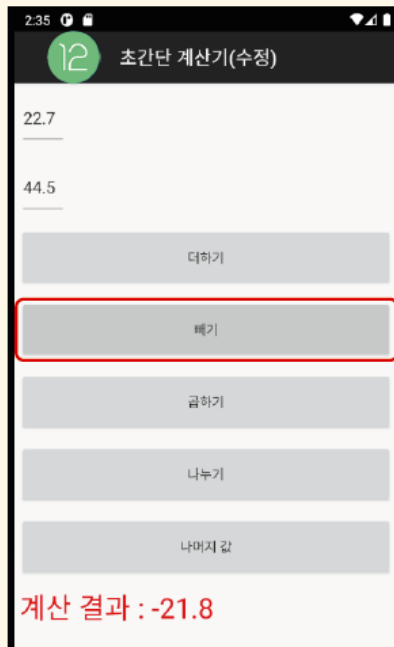


그림 4-7 수정된 초간단 계산기 앱

03

## 기본 위젯 활용하기

# 1. 컴파운드버튼

## ■ CompoundButton 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
```

컴파운드버튼 계층도

- Button 클래스의 하위 클래스
- 체크박스, 라디오버튼, 스위치, 토글버튼의 상위 클래스

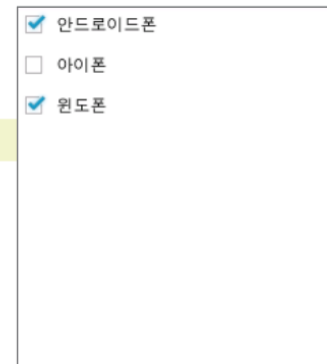
# 1. 컴파운드버튼

## ■ 체크박스

- 체크할 때마다 상태가 체크, 언체크로 변경
- 여러 개의 체크박스가 있어도 서로 독립적으로 동작
  - 따라서 여러 개를 동시에 체크할 수 있음

예제 4-24 CheckBox의 XML 코드

```
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" />
```



☒ 안드로이드폰

☐ 아이폰

☒ 윈도폰

# 1. 컴파운드버튼

## ■ 체크박스에서 체크와 언체크가 바뀔 때 Java 처리

- 버튼 클릭과 절차가 거의 동일함

### ① 체크박스 변수 선언

- `CheckBox mycheck;`

### ② 변수에 체크박스 위젯 대입

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

### ③ 체크박스가 변경될 때 동작하는 클래스 정의

- `mycheck.setOnCheckedChangeListener(new  
CompoundButton.OnCheckedChangeListener() {  
 public void onCheckedChanged(CompoundButton buttonView,  
 boolean isChecked) {  
 // 동작 내용을 이 부분에 코딩  
 }  
});`

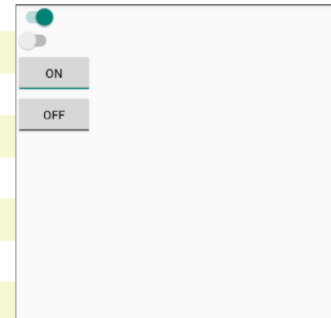
# 1. 컴파운드버튼

## ■ 스위치와 토글버튼

- 스위치와 토글버튼은 모양만 조금 다를 뿐 용도는 거의 동일함
- 스위치의 주 용도는 온/오프 상태 표시임

예제 4-25 스위치와 토글버튼의 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" />
```





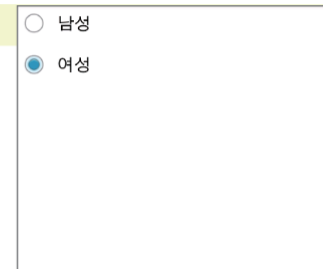
# 1. 컴파운드버튼

## ■ 라디오버튼과 라디오그룹

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

예제 4-26 RadioGroup과 RadioButton의 XML 코드

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



○ 남성

☒ 여성

## 2. 이미지뷰와 이미지버튼

### ■ 이미지뷰(Image View)

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

이미지뷰 계층도

- 그림을 출력하는 위젯으로 그림이 필요하거나 화면을 화려하게 구성할 때 사용함
- 이미지뷰에 보여줄 그림 파일은 프로젝트의 [res]-[drawable]에 있어야 함
- XML에서 “@drawable/그림 아이디” 형식으로 접근함

## 2. 이미지뷰와 이미지버튼

### ■ 이미지뷰 및 이미지버튼의 XML 속성

- src : 이미지의 경로를 나타냄
- maxHeight/maxWidth : 이미지의 크기를 지정
- scaleType : 이미지의 확대/축소 방식 지정
- 그림 파일이 [res]-[drawable] 폴더에 있어야 사용 가능함

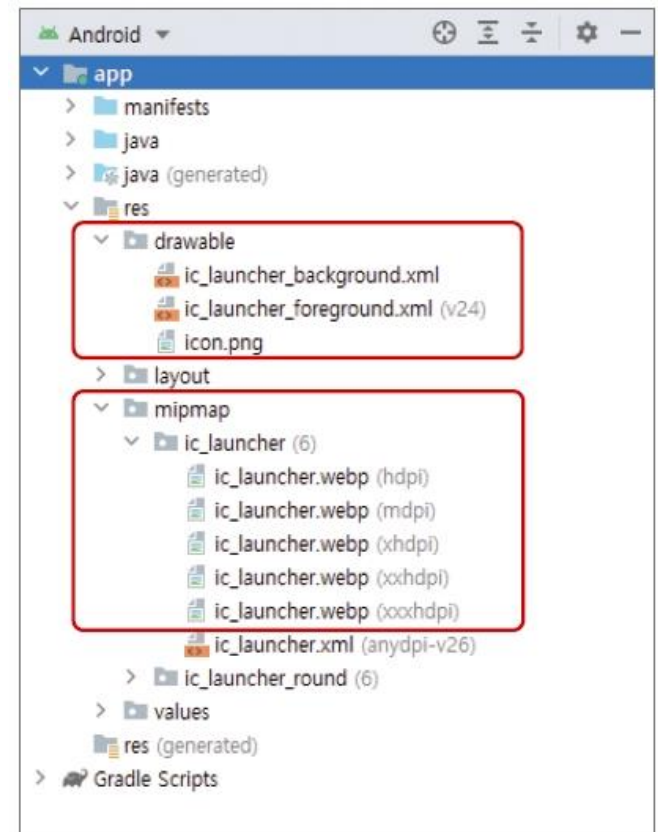
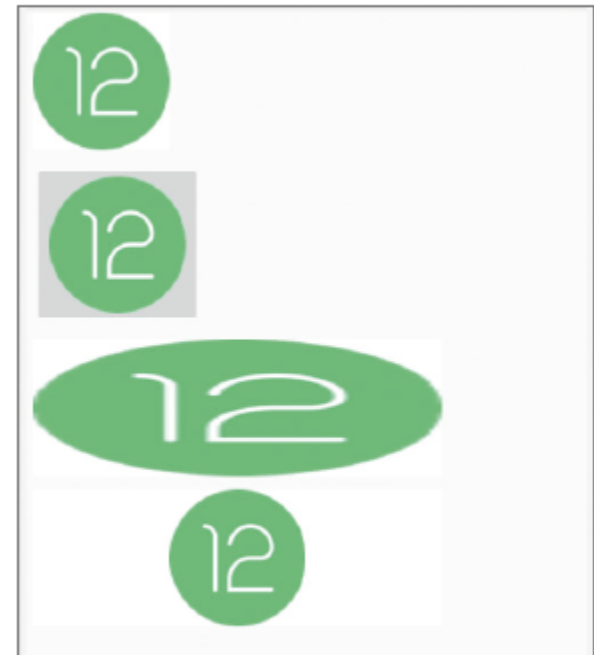


그림 4-8 drawable 폴더와 mipmap 폴더

## 2. 이미지뷰와 이미지버튼

예제 4-27 이미지뷰와 이미지버튼의 XML 코드

```
1 <ImageView
2 android:src="@drawable/icon.png" />
3 <ImageButton
4 android:src="@drawable/icon.png" />
5 <ImageView
6 android:layout_width="300dp"
7 android:layout_height="100dp"
8 android:scaleType="fitXY"
9 android:src="@drawable/icon.png" />
10 <ImageView
11 android:layout_width="300dp"
12 android:layout_height="100dp"
13 android:scaleType="fitCenter"
14 android:src="@drawable/icon.png" />
```



### 3. 버튼과 에디트텍스트

#### ■ 실습 4-2 애완동물 사진 보기 앱 만들기

- 보고 싶은 애완동물의 사진을 출력하는 앱
- '시작함'에 체크 : 애완동물 사진 세 가지 중에서 하나를 선택하라는 내용이 나옴
- 선택 후에 <선택 완료> 버튼 클릭 : 해당 애완동물의 이미지가 나타남

#### ■ 1 안드로이드 프로젝트 생성

- (1) 프로젝트 이름 : Project4\_2
- (2) 패키지 이름 : com.cookandroid.project4\_2

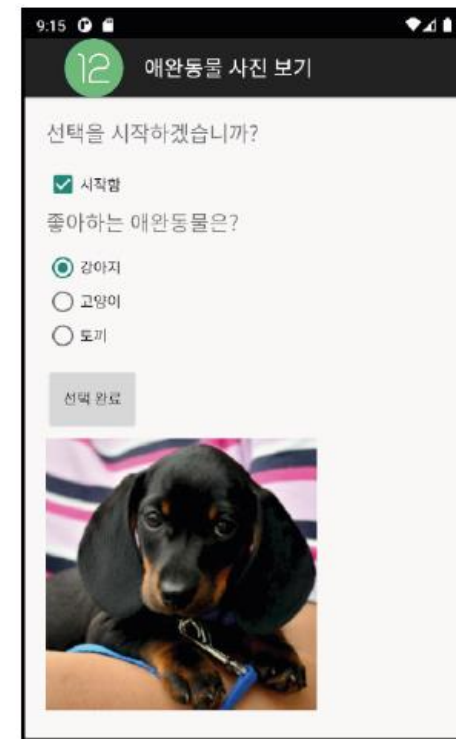


그림 4-9 애완동물 사진 보기 앱 결과 화면

### 3. 버튼과 에디트텍스트

- 실습 4-2 애완동물 사진 보기 앱 만들기
- 2 화면 디자인 및 편집
  - (1) 프로젝트의 [res]-[drawable]에 강아지, 고양이, 토끼 그림 파일을 미리 복사

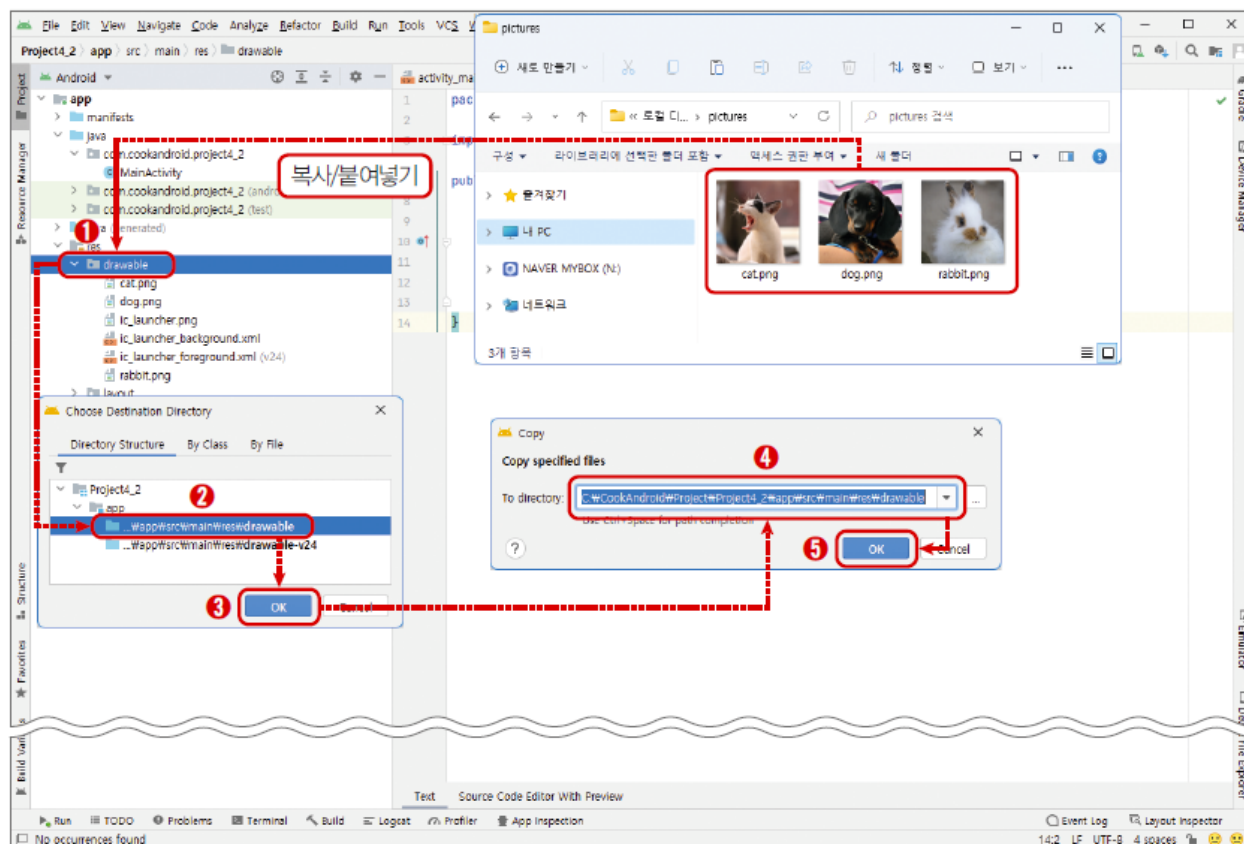


그림 4-10 그림 파일 복사

### 3. 버튼과 에디트텍스트

#### ■ 실습 4-2 애완동물 사진 보기 앱 만들기

#### ■ 2 화면 디자인 및 편집

- (2) Project Tree에서 [app]-[res]-[layout]-[activity\_main.xml]에서 코딩
  - TextView, CheckBox, TextView, RadioGroup, RadioButton 각각 3개, Button, ImageView의 차례로 만들
  - 레이아웃에 padding을 적절히 지정
  - 맨 위의 TextView와 CheckBox를 제외하고, 나머지 위젯은 visibility 속성을 invisible로 지정
  - 위젯의 id는 위에서부터 Text1, ChkAgree, Text2, Rgroup1, RdoDog, RdoCat, RdoRabbit, BtnOK, ImgPet으로 함

### 3. 버튼과 에디트텍스트

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 android:id="@+id/Text2"
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 ~~~~ 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



### 3. 버튼과 에디트텍스트

#### ■ 실습 4-2 애완동물 사진 보기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

- (1) Project Tree에서 [app]-[java]-[패키지 이름]-[MainActivity]에 전역변수 선언
  - activity\_main.xml의 9개 위젯에 대응할 위젯 변수 9개

예제 4-29 Java 코드 1

```
1 ~~~~ 중간 생략 ~~~~
2 public class MainActivity extends AppCompatActivity {
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 ~~~~
```

### 3. 버튼과 에디트텍스트

- 실습 4-2 애완동물 사진 보기 앱 만들기
- 3 Java 코드 작성 및 수정
  - (2) 각 위젯을 변수에 대입. onCreate( ) 메소드 안에서 처리

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 }
```

### 3. 버튼과 에디트텍스트

#### ■ 실습 4-2 애완동물 사진 보기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

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

예제 4-31 Java 코드 3

```
1 chkAgree.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener
 Listener() {
2 public void onCheckedChanged(CompoundButton buttonView, boolean
 isChecked) {
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 });
```

### 3. 버튼과 에디트텍스트

#### ■ 실습 4-2 애완동물 사진 보기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

- (4) <선택 완료>를 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 });
```

### 3. 버튼과 에디트텍스트

- 실습 4-2 애완동물 사진 보기 앱 만들기
- 4 프로젝트 실행 및 결과 확인
  - 프로젝트를 실행하여 결과를 확인

#### ▶ 직접 풀어보기 4-4

[실습 4-2]를 다음과 같이 수정하라.

- ‘좋아하는 안드로이드 버전은?’으로 질문을 변경한다.
- ‘시작함’을 스위치로 변경한다.
- <선택 완료>를 없애고, 라디오버튼을 선택할 때마다 즉시 해당 이미지가 나오도록 변경한다.
- 마지막에 <종료>와 <처음으로> 버튼을 추가한다. <종료>를 클릭하면 응용 프로그램이 완전히 종료되게 하고, <처음으로>를 클릭하면 다시 초기화되고 처음 화면이 나오게 한다.

그림 4-11 안드로이드 사진 보기 앱

