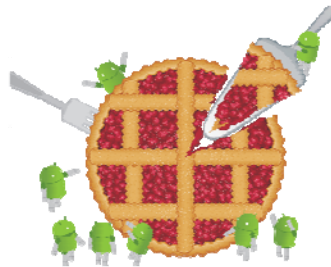


04. 기본 위젯 익히기

IT CookBook, Android Studio를 활용한 안드로이드 프로그래밍



학습목표

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

차례

- 01 뷰의 개요
- 02 기본 위젯 다루기
- 03 기본 위젯 활용하기

Page • 3

1. 뷰의 개요 ▶ 뷰와 뷰그룹[1/5]

❖ 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



Page • 4

그림 4-1 위젯과 레이아웃

1. 뷰의 개요 ▶ 뷰와 뷰그룹[2/5]

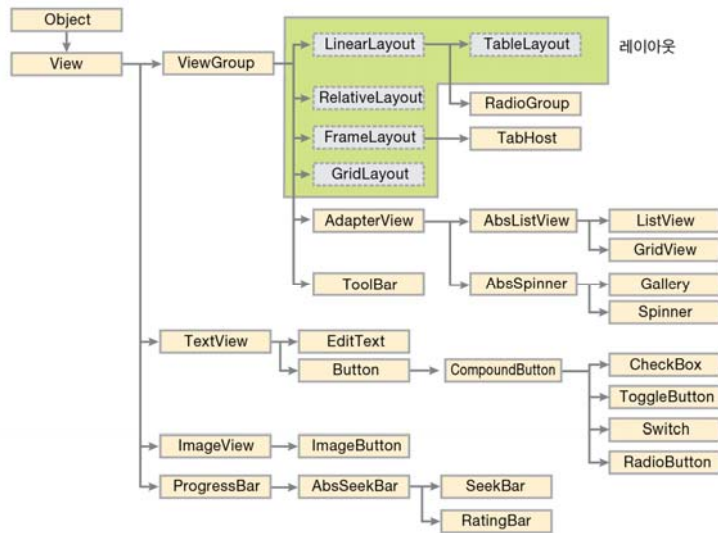


그림 4-2 안드로이드 View 클래스 계층도

1. 뷰의 개요 ▶ 뷰와 뷰그룹[3/5]

❖ Button의 XML 속성

```

<Button
    android:
        android:layout_height (required)
        android:layout_gravity
        android:layout_margin
        android:layout_marginBottom
        android:layout_marginEnd
        android:layout_marginLeft
        android:layout_marginRight
        android:layout_marginStart
        android:layout_marginTop
        android:layout_weight
    [Namespace Prefix]
    Ctrl+Down and Ctrl+Up will move caret down and up in the editor >>
    
```

그림 4-3 Button의 XML 속성

1. 뷰의 개요 ▶ 뷰와 뷰그룹[4/5]

❖ 클래스 상속관계 찾기

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

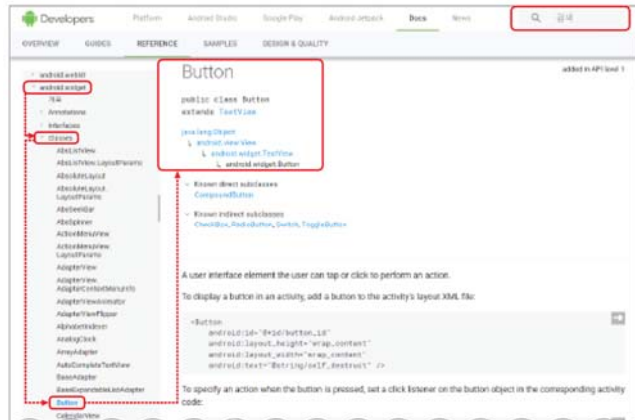


그림 4-4 클래스의 상속 관계를 찾는 방법

1. 뷰의 개요 ▶ View 클래스의 XML 속성[1/14]

❖ 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"
/>
```

1. 뷰의 개요 ▶ View 클래스의 XML 속성[2/14]

❖ id 속성

- ✓ id 속성은 모든 위젯의 아이디를 나타냄
 - ✓ Java 코드에서 위젯에 접근할 때 id 속성에 지정한 아이디를 사용
 - ✓ id 속성은 위젯에 아이디를 새로 부여하는 개념이므로 "@+id/" 형식으로 지정
 - ✓ / 다음에는 새로 지정할 id를 적음
- android:id="@+id/btn1"의 의미는 버튼 위젯의 아이디를 btn1로 부여한 것

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



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

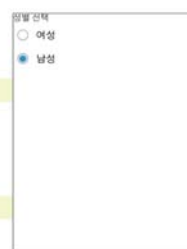
Page • 9

1. 뷰의 개요 ▶ View 클래스의 XML 속성[3/14]

❖ id 속성 예제

예제 4-1 id 속성의 XML 코드

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



Page

1. 뷰의 개요 ▶ View 클래스의 XML 속성[4/14]

❖ layout_width, layout_height 속성

- ✓ match_parent : 자신의 부모(대개는 레이아웃)에 폭이나 높이를 맞춘다는 의미
- ✓ wrap_content : 자신의 폭이나 높이를 자신 안의 글자가 들어갈 정도로 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>
```



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>
```



Page • 11

1. 뷰의 개요 ▶ View 클래스의 XML 속성[5/14]

❖ 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="wrap_content"
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" />
```



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="1540px"
6     android:text="버튼입니다" />
7 </LinearLayout>
```



Pag

1. 뷰의 개요 ▶ View 클래스의 XML 속성[6/14]

❖ background 속성

- ✓ background 속성은 위젯의 색상을 주로 #RRGGBB 값으로 지정

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>
```

버튼입니다



Page • 13

1. 뷰의 개요 ▶ View 클래스의 XML 속성[7/14]

저자
원마디

책의 XML 코드 표기(★ 주의)

텍스트뷰 1개와 버튼 1개가 있는 기본적인 activity_main.xml 코드의 전체는 다음과 같다.

예제 4-8 전체 XML 코드

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

1. 뷰의 개요 ▶ View 클래스의 XML 속성[8/14]

1행과 2행의 xmlns 이후는 항상 동일한 내용이 나오고 3행, 4행, 8행, 9행, 14행, 15행에 나오는 layout_width와 layout_height는 모든 위젯의 필수 요소로 자주 등장할 것이다. 5행의 orientation 속성도 거의 고정되어 나온다. 따라서 이 책에서 이러한 반복적인 속성은 꼭 표현할 필요가 있을 때를 제외하고는 생략할 것이다. 즉 앞으로는 다음과 같이 필수나 반복되는 코드를 생략하고 간략하게 표현한다.

예제 4-9 앞으로 책에서 표기할 XML 코드

```
1 <LinearLayout>
2
3   <TextView
4       android:text="@string/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 표기할 것인데, 실제로는 전체 코드를 다 표현해야 동작함을 잊지 말길 바란다. 주의할 점은 5행의 orientation 속성을 생략하면 LinearLayout이 horizontal이 된다는 것이다.

이 책의 화면은 대부분 vertical 정렬이므로 예제에는 생략되어 있어도 5행을 잊지 말고 쓰자. 이는 5장에서 다시 설명하겠다. 생략하지 않은 모든 소스코드는 한빛이카데미 사이트(<http://www.hanbit.co.kr/src/4438/>)에 등록해놓았다.

1. 뷰의 개요 ▶ View 클래스의 XML 속성[9/14]

❖ 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       andr
8 </LinearLayout>
```



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>
```



1. 뷰의 개요 ▶ View 클래스의 XML 속성[10/14]

❖ 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>
```



Page • 17

1. 뷰의 개요 ▶ View 클래스의 XML 속성[11/14]

❖ visibility 속성

- ✓ visibility 속성은 위젯을 보일 것인지 여부를 결정
- ✓ 디폴트인 visible은 보이는 상태, invisible과 gone은 안 보이는 상태

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" />
```



Page • 18

1. 뷰의 개요 ▶ View 클래스의 XML 속성[12/14]

❖ enabled, clickable 속성

- ✓ enabled 속성 : 위젯의 동작 여부
- ✓ clickable 속성 : 클릭이나 터치가 가능하도록 함

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" />
```



Page • 19

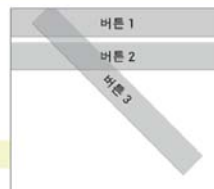
1. 뷰의 개요 ▶ View 클래스의 XML 속성[13/14]

❖ 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" />
```



Page • 20

1. 뷰의 개요 ▶ View 클래스의 XML 속성[14/14]

▶ 직접 들어보기 4-2

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



Page • 21

2. 기본 위젯 다루기 ▶ 텍스트뷰[1/2]

❖ 계층도

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

TextView 계층도

- ✓ **text** 속성 : 텍스트뷰에 나타나는 문자열 표현
- ✓ **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|italic을 설정, 디폴트는 normal
- ✓ **singleLine** 속성 : 글이 길어 줄이 넘어갈 경우 강제로 한 줄만 출력하고 문자열의 맨 뒤에 '...'를 표시. 값으로 true와 false를 설정할 수 있으며 디폴트는 false

Page • 22

2. 기본 위젯 다루기 ▶ 텍스트뷰[2/2]

❖ 텍스트뷰 예제

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...

Page •

2. 기본 위젯 다루기 ▶ 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

Page • 24

2. 기본 위젯 다루기 ▶ Java 코드로 XML 속성 설정[2/3]

❖ 텍스트뷰 속성을 변경하는 Java 코드

예제 4-18 텍스트 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("가나다라바사아자차카타파하가나다라바사아자차카타파하");
15    tv3.setSingleLine();
16 }

```

안녕하세요?

TextView 연습 2

가나다라바사아자차카타파하가나다라바사아

Page

2. 기본 위젯 다루기 ▶ 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()	password
		text
		textColor
		textSize
		setTransformationMethod()
		setText()
		setTextColor()
		setTextSize()

Page • 26

2. 기본 위젯 다루기 ▶ 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?" />
```



```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="나는 어떤 위젯일까요?" />
```

Page • 27

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[2/12]

❖ 버튼

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

Button 계층도

Page • 28

2. 기본 위젯 다루기 ▶ 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 버튼을 클릭할 때 동작하는 클래스 정의

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

Page • 28

2. 기본 위젯 다루기 ▶ 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

Page • 30

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[5/12]

❖ 일반적인 에디트텍스트의 XML 코드

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

❶ 에디트텍스트 변수 선언

```
EditText myEdit;
```

❷ 변수에 에디트텍스트 위젯 대입

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

❸ 에디트텍스트에 입력된 값 가져오기 → 주로 버튼 클릭 이벤트 리스너 안에 넣음

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

Page • 31

2. 기본 위젯 다루기 ▶ 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 30dp
- ✓ 각 위젯의 id는 위에서부터 차례로 Edit1, Edit2, BtnAdd, BtnSub, BtnMul, BtnDiv, TextResult로 함



그림 4-5 초간단 계산기 앱 결과 화면

Page • 32

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[7/12]

실습 4-1 초간단 계산기 앱 만들기

❖ 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="숫자!" />
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>

```



Page • 33

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[8/12]

실습 4-1 초간단 계산기 앱 만들기

❖ 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 ----- 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 -----

```

Page • 34

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[9/12]

실습 4-1 초간단 계산기 앱 만들기

❖ 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 }
```

Page • 35

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[10/12]

실습 4-1 초간단 계산기 앱 만들기

❖ 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 });
```

Page • 36

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[11/12]

실습 4-1 초간단 계산기 앱 만들기

❖ Java 코드 작성 및 수정

- ✓ 빼기, 곱하기, 나누기 코드 추가 후 완성

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    textView = (TextView) findViewById(R.id.textView);
15
16    btnAdd.setOnClickListener(new View.OnClickListener() {
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            textView.setText("덧셈 결과 : " + result.toString());
22            return false;
23        }
24    });
25
26    btnSub.setOnClickListener(new View.OnClickListener() {
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            textView.setText("뺄셈 결과 : " + result.toString());
32            return false;
33        }
34    });
35
36    btnMul.setOnClickListener(new View.OnClickListener() {
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            textView.setText("곱셈 결과 : " + result.toString());
42            return false;
43        }
44    });
45
46    btnDiv.setOnClickListener(new View.OnClickListener() {
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            textView.setText("나눗셈 결과 : " + result.toString());
52            return false;
53        }
54    });
55    }

```

Page • 37

Coding Exercise 실습 4-1

```

• package com.cookandroid.project4_1;

• import android.support.v7.app.AppCompatActivity;
• import android.os.Bundle;
• import android.view.MotionEvent;
• import android.view.View;
• import android.widget.Button;
• import android.widget.EditText;
• import android.widget.TextView;

• public class MainActivity extends AppCompatActivity {
•     EditText edit1, edit2;
•     Button btnAdd, btnSub, btnMul, btnDiv;
•     TextView textView;
•     String num1, num2;
•     Integer result;

```

Page • 38

Coding Exercise 실습 4-1

```
• @Override
• public void onCreate(Bundle savedInstanceState) {
•     super.onCreate(savedInstanceState);
•     setContentView(R.layout.activity_main);
•     setTitle("초간단 계산기");
•     edit1 = (EditText) findViewById(R.id.Edit1);
•     edit2 = (EditText) findViewById(R.id.Edit2);
•     btnAdd = (Button) findViewById(R.id.BtnAdd);
•     btnSub = (Button) findViewById(R.id.BtnSub);
•     btnMul = (Button) findViewById(R.id.BtnMul);
•     btnDiv = (Button) findViewById(R.id.BtnDiv);

•     textResult = (TextView) findViewById(R.id.TextResult);
•     btnAdd.setOnTouchListener(new View.OnTouchListener() {
•         public boolean onTouch(View arg0, MotionEvent arg1) {
•             num1 = edit1.getText().toString();
•             num2 = edit2.getText().toString();
•             result = Integer.parseInt(num1) + Integer.parseInt(num2);
•             textResult.setText("계산 결과 : " + result.toString());
•             return false;
•         }
•     });
```

Page * 39

Coding Exercise 실습 4-1

```
• btnSub.setOnTouchListener(new View.OnTouchListener() {
•     public boolean onTouch(View arg0, MotionEvent arg1) {
•         num1 = edit1.getText().toString();
•         num2 = edit2.getText().toString();
•         result = Integer.parseInt(num1) - Integer.parseInt(num2);
•         textResult.setText("계산 결과 : " + result.toString());
•         return false;
•     }
• });

• btnMul.setOnTouchListener(new View.OnTouchListener() {
•     public boolean onTouch(View arg0, MotionEvent arg1) {
•         num1 = edit1.getText().toString();
•         num2 = edit2.getText().toString();
•         result = Integer.parseInt(num1) * Integer.parseInt(num2);
•         textResult.setText("계산 결과 : " + result.toString());
•         return false;
•     }
• });
```

Page * 40

Coding Exercise 실습 4-1

```
• btnDiv.setOnClickListener(new View.OnClickListener() {  
•     public boolean onTouch(View arg0, MotionEvent arg1) {  
•         num1 = edit1.getText().toString();  
•         num2 = edit2.getText().toString();  
•         result = Integer.parseInt(num1) / Integer.parseInt(num2);  
•         textResult.setText("계산 결과 : " + result.toString());  
•         return false;  
•     }  
• }  
  
• }  
  
• }
```

Page • 41

2. 기본 위젯 다루기 ▶ 버튼과 에디트텍스트[12/12]

▶ 직접 알아보기 4-3

[실습 4-1]에 다음과 같이 기능을 추가하거나 변경하라.

- 터치가 아닌 클릭으로 변경
- 나머지값 구하기 버튼 추가
- 값을 입력하지 않고, 버튼을 클릭할 때 오류 메시지를 토스트로 나타내기
- 실숫값 계산하기
- 0으로 나누면 토스트 메시지를 나타내고 계산하지 않기



그림 4-6 개선된 계산기

Page • 42

3. 기본 위젯 활용하기 ▶ 컴파운드버튼[1/4]

❖ 컴파운드버튼

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

Page • 43

3. 기본 위젯 활용하기 ▶ 컴파운드버튼[2/4]

❖ 체크박스

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="윈도우폰"
11  android:checked="true" />
```



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) {
        // 이 부분에 동작할 내용을 코딩
    }
});
```

Page • 44

3. 기본 위젯 활용하기 ▶ 컴파운드버튼[3/4]

❖ 스위치와 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" />
```



Page • 45

3. 기본 위젯 활용하기 ▶ 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

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>
```



Page • 46

3. 기본 위젯 활용하기 ▶ 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

Page • 47

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[2/11]

❖ 이미지뷰 및 이미지버튼의 XML :

- ✓ src : 이미지의 경로를 나타냄
- ✓ maxHeight/maxWidth :
이미지의 크기를 지정
- ✓ scaleType : 이미지의 확대/축소 방식 지정

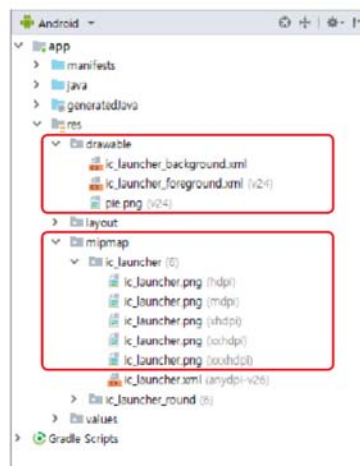


그림 4-7 drawable 폴더와 mipmap 폴더

Page • 48

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[3/11]

❖ 이미지뷰와 이미지버튼의 XML 코드

예제 4-27 ImageView와 ImageButton의 XML 코드

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



Page • 49

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[4/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ 안드로이드 프로젝트 생성

- ✓ 프로젝트 이름 : Project4_2
- ✓ 패키지 이름 : com.cookandroid.project4_2

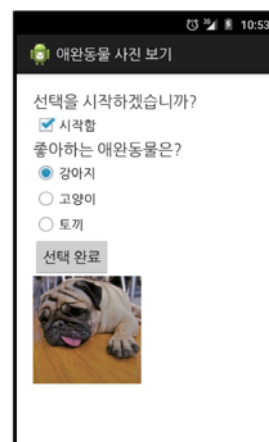


그림 4-8 실행 화면

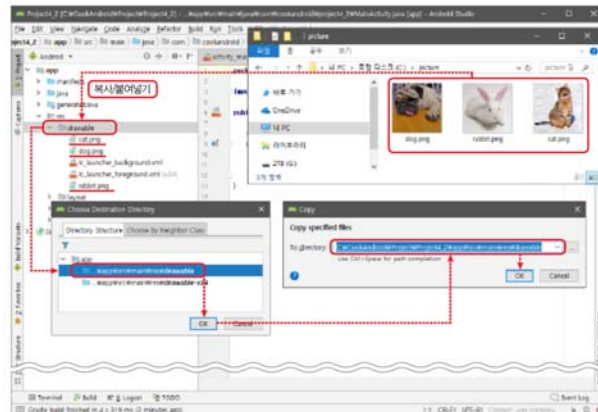
Page • 50

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[5/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ 화면 및 디자인 편집

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



Page • 51

그림 4-9 그림 파일 복사

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[6/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ 화면 및 디자인 편집

- ✓ TextView, CheckBox, TextView, RadioGroup, RadioButton 각각 3개, Button, ImageView의 차례로 만들
- ✓ 레이아웃에 padding을 적절히 지정
- ✓ 맨 위의 TextView와 CheckBox를 제외하 나머지 위젯은 visibility 속성을 invisible로 지정
- ✓ 위젯의 id는 위에서부터 Text1, ChkAgree, Text2, Rgroup1, RdoDog, RdoCat, RdoRabbit, BtnOK, ImgPet으로 함



Page • 52

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[7/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ 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 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 ~~~~
```

Page • 53

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[8/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ 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 }
```

Page • 54

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[9/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ 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 buttonView, boolean
3         isChecked) {
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    });
```

Page • 55

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[10/11]

실습 4-2 애완동물 사진 보기 앱 만들기

❖ Java 코드 작성 및 수정

- ✓ <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(), "동물을 먼저 선택하세요", Toast.
15                    LENGTH_SHORT).show();
16            }
17    });
```

Page • 56

Coding Exercise 실습 4-2

```
• package com.cookandroid.project4_2;

• import android.support.v7.app.AppCompatActivity;
• import android.os.Bundle;
• import android.view.View;
• import android.widget.Button;
• import android.widget.CheckBox;
• import android.widget.CompoundButton;
• import android.widget.ImageView;
• import android.widget.RadioButton;
• import android.widget.RadioGroup;
• import android.widget.TextView;
• import android.widget.Toast;

• public class MainActivity extends AppCompatActivity {
•     TextView text1, text2;
•     CheckBox chkAgree;
•     RadioGroup rGroup1;
•     RadioButton rdoDog, rdoCat, rdoRabbit;
•     Button btnOK;
•     ImageView imgPet;
```

Page • 57

Coding Exercise 실습 4-2

```
• @Override
•     public void onCreate(Bundle savedInstanceState) {
•         super.onCreate(savedInstanceState);
•         setContentView(R.layout.activity_main);
•         setTitle("애완동물 사진 보기");

•         // 위젯을 변수에 대입
•         text1 = (TextView) findViewById(R.id.Text1);
•         chkAgree = (CheckBox) findViewById(R.id.ChkAgree);

•         text2 = (TextView) findViewById(R.id.Text2);
•         rGroup1 = (RadioGroup) findViewById(R.id.Rgroup1);
•         rdoDog = (RadioButton) findViewById(R.id.RdoDog);
•         rdoCat = (RadioButton) findViewById(R.id.RdoCat);
•         rdoRabbit = (RadioButton) findViewById(R.id.RdoRabbit);

•         btnOK = (Button) findViewById(R.id.BtnOK);
•         imgPet = (ImageView) findViewById(R.id ImgPet);
```

Page • 58

Coding Exercise 실습 4-2

```
• // 동의함 체크박스의 체크가 변경되면
•     chkAgree.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
•         public void onCheckedChanged(CompoundButton arg0, boolean arg1) {
•             // 체크되면 모두 보이도록 설정
•             if (chkAgree.isChecked() == true) {
•                 text2.setVisibility(android.view.View.VISIBLE);
•                 rGroup1.setVisibility(android.view.View.VISIBLE);
•                 btnOK.setVisibility(android.view.View.VISIBLE);
•                 imgPet.setVisibility(android.view.View.VISIBLE);
•             } else {
•                 text2.setVisibility(android.view.View.INVISIBLE);
•                 rGroup1.setVisibility(android.view.View.INVISIBLE);
•                 btnOK.setVisibility(android.view.View.INVISIBLE);
•                 imgPet.setVisibility(android.view.View.INVISIBLE);
•             }
•         }
•     });
```

Page • 59

Coding Exercise 실습 4-2

```
• // 선택확인 버튼을 클릭하면
•     btnOK.setOnClickListener(new View.OnClickListener() {
•         public void onClick(View arg0) {
•
•             switch (rGroup1.getCheckedRadioButtonId()) {
•                 case R.id.RdoDog:
•                     imgPet.setImageResource(R.drawable.dog);
•                     break;
•                 case R.id.RdoCat:
•                     imgPet.setImageResource(R.drawable.cat);
•                     break;
•                 case R.id.RdoRabbit:
•                     imgPet.setImageResource(R.drawable.rabbit);
•                     break;
•                 default:
•                     Toast.makeText(getApplicationContext(), "동물 먼저 선택하세요",
•                         Toast.LENGTH_SHORT)
•                         .show();
•             }
•         }
•     });
```

Page • 60

3. 기본 위젯 활용하기 ▶ 이미지뷰와 이미지버튼[11/11]

▶ 작업 들어보기 4-4

[실습 4-2]를 다음과 같이 수정하라.

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



그림 4-10 안드로이드 사진 보기

Question ?

IT CookBook, Android Studio를 활용한 안드로이드 프로그래밍