

기본 위젯

Instructor: Park, JoonSeok (pjs50@pusan.ac.kr)

뷰와 레이아웃

▶ 뷰란?

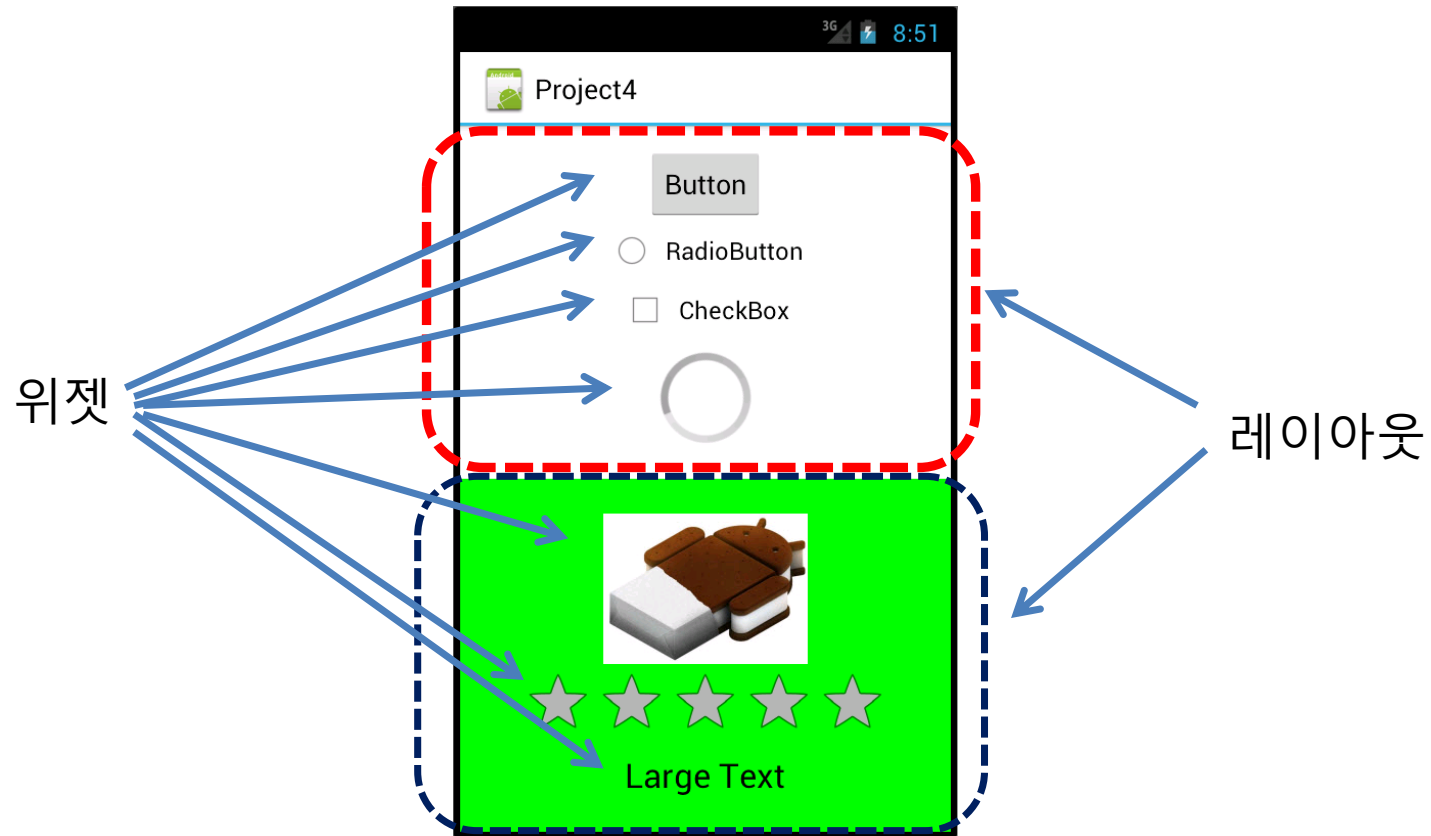
- ▶ 앱 실행 화면을 구성하는 요소 중에서 가장 상위에 있는 클래스
 - ▶ Ex: 텍스트 뷰, 버튼, 체크 박스 등 → 모두 뷰 클래스에서 상속
- ▶ 뷰 클래스에서 상속을 받은 클래스 위젯(또는 뷰)로 지칭

▶ 레이아웃

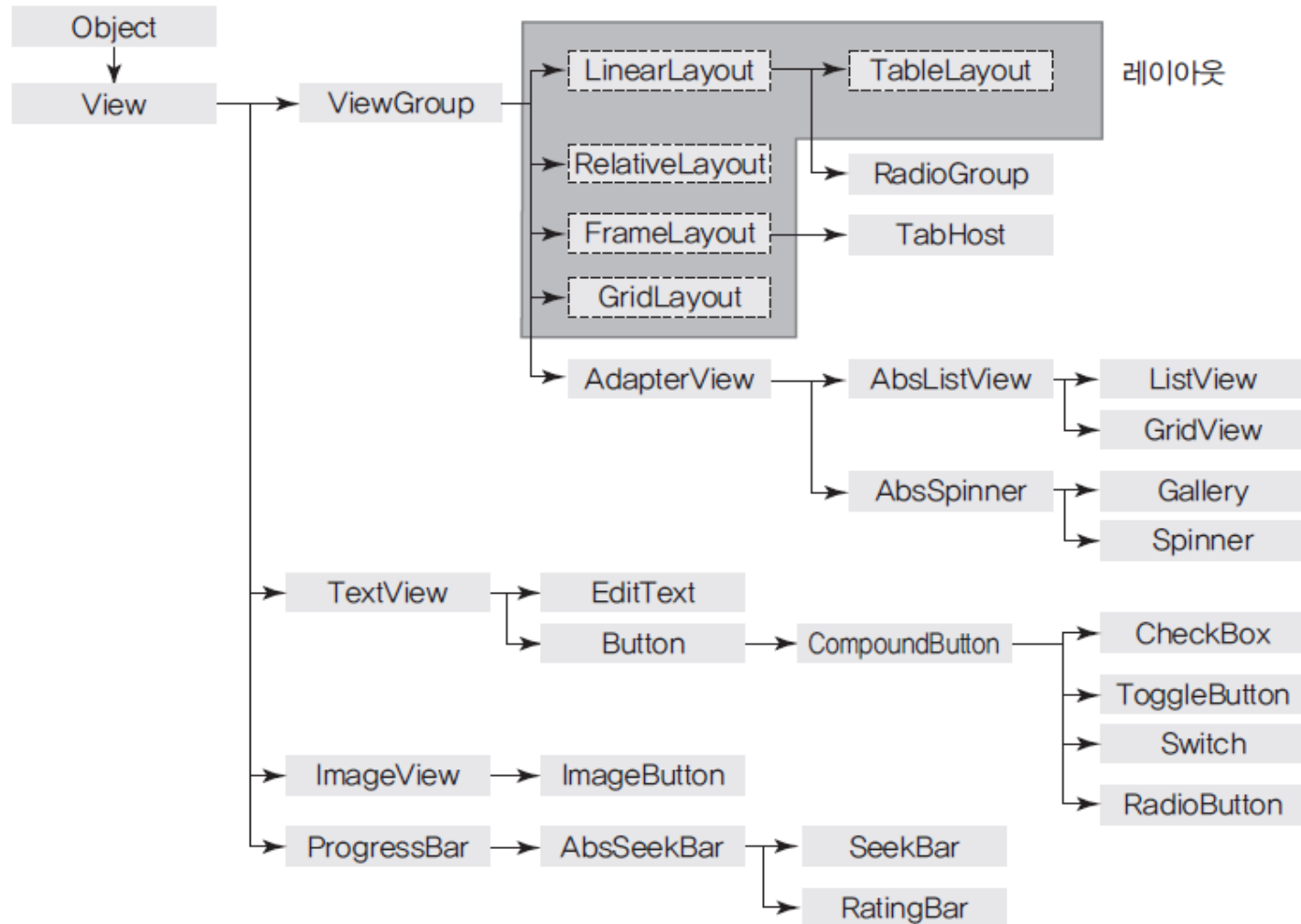
- ▶ 다른 위젯을 담을 수 있는 클래스



뷰와 레이아웃

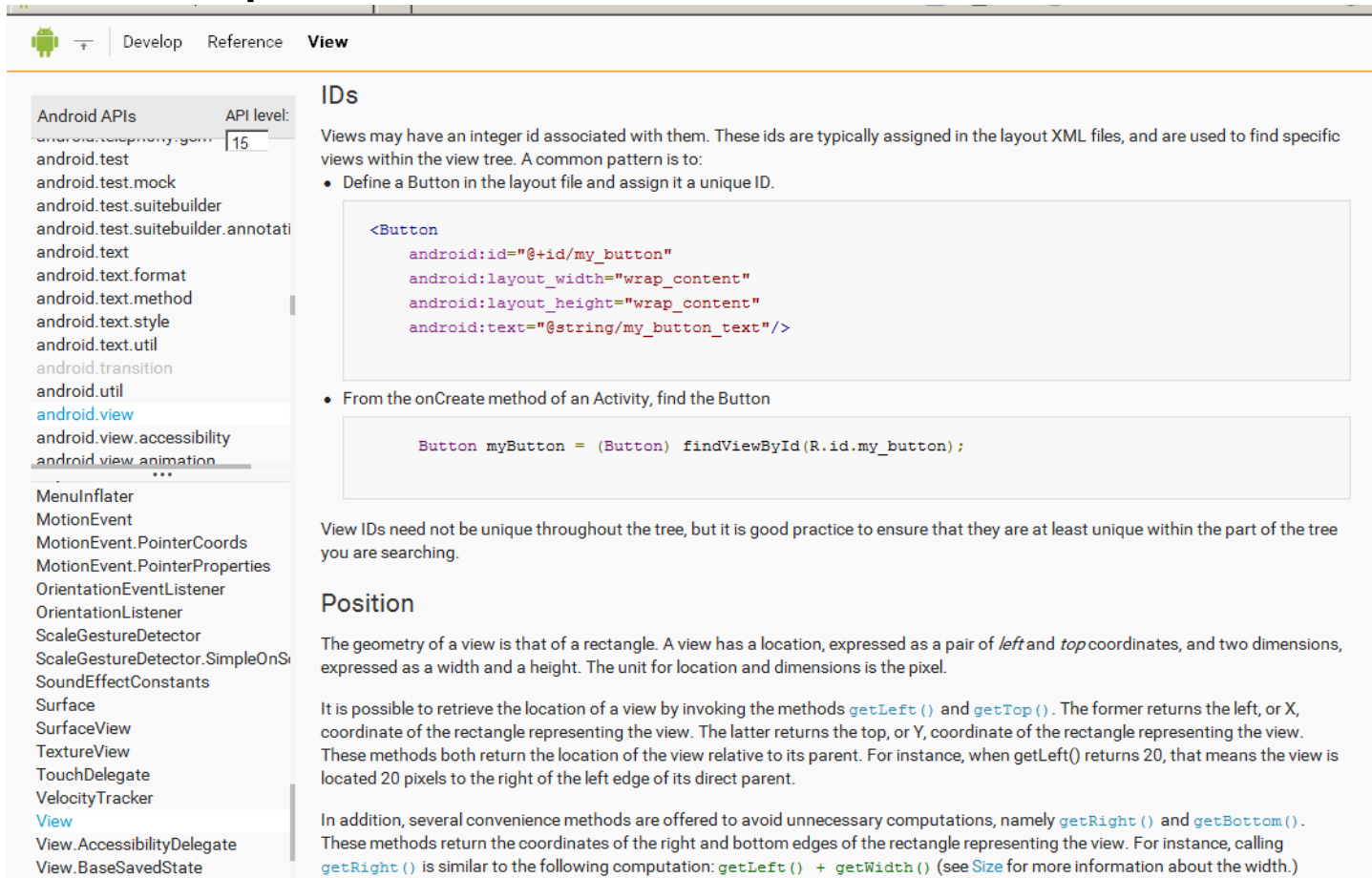


뷰 클래스 계층도



뷰 클래스의 상속 관계

▶ <http://developer.android.com>



The screenshot shows the Android Developer website. On the left, there's a sidebar with a list of Android APIs. The 'android.view' package is selected, and the 'View' class is highlighted. The main content area is titled 'IDs' and explains that views have integer IDs for identification. It includes a code snippet for defining a Button in XML and a Java snippet for finding it in code. Below this, the 'Position' section explains how to retrieve the location and dimensions of a view using methods like `getLeft()`, `getTop()`, `getRight()`, and `getBottom()`.

Android APIs API level: 15

- android.test
- android.test.mock
- android.test.suitebuilder
- android.test.suitebuilder.annotati
- android.text
- android.text.format
- android.text.method
- android.text.style
- android.text.util
- android.transition
- android.util
- android.view
- android.view.accessibility
- android.view.animation
- ...

MenuInflater
MotionEvent
MotionEvent.PointerCoords
MotionEvent.PointerProperties
OrientationEventListener
ScaleGestureDetector
ScaleGestureDetector.SimpleOnS
SoundEffectConstants
Surface
SurfaceView
TextureView
TouchDelegate
VelocityTracker
View
View.AccessibilityDelegate
View.BaseSavedState

IDs

Views may have an integer id associated with them. These ids are typically assigned in the layout XML files, and are used to find specific views within the view tree. A common pattern is to:

- Define a Button in the layout file and assign it a unique ID.

```
<Button
    android:id="@+id/my_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/my_button_text"/>
```

- From the onCreate method of an Activity, find the Button

```
Button myButton = (Button) findViewById(R.id.my_button);
```

View IDs need not be unique throughout the tree, but it is good practice to ensure that they are at least unique within the part of the tree you are searching.

Position

The geometry of a view is that of a rectangle. A view has a location, expressed as a pair of *left* and *top* coordinates, and two dimensions, expressed as a width and a height. The unit for location and dimensions is the pixel.

It is possible to retrieve the location of a view by invoking the methods `getLeft()` and `getTop()`. The former returns the left, or X, coordinate of the rectangle representing the view. The latter returns the top, or Y, coordinate of the rectangle representing the view. These methods both return the location of the view relative to its parent. For instance, when `getLeft()` returns 20, that means the view is located 20 pixels to the right of the left edge of its direct parent.

In addition, several convenience methods are offered to avoid unnecessary computations, namely `getRight()` and `getBottom()`. These methods return the coordinates of the right and bottom edges of the rectangle representing the view. For instance, calling `getRight()` is similar to the following computation: `getLeft() + getWidth()` (see [Size](#) for more information about the width.)

뷰 클래스 XML 속성

- ▶ 위젯과 레이아웃 등은 모두 View 클래스의 속성과 메소드를 상속받음

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

뷰 클래스의 XML 속성

▶ id

- ▶ 위젯의 아이디
- ▶ Java 코드에서 위젯에 접근할 때 id 속성에 지정한 아이디 사용

위젯 변수 = (위젯 형) findViewById(R.id.위젯id);

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



뷰 클래스의 XML 속성

- ▶ **layout_width, layout_height**
 - ▶ 위젯의 폭과 높이
 - ▶ 속성값 match_parent(=fill_parent)는 이름 그대로 자신의 부모(대개는 레이아웃)에 폭이나 높이를 맞추는 의미
 - ▶ 속성값 wrap_content는 자신의 폭이나 높이를 자신 안의 글자가 꼭 들어갈 정도로 설정한다는 의미



뷰 클래스 XML 속성

▶ padding

- ▶ 위젯의 경계선으로부터 위젯 안의 요소가 떨어지도록 설정

```
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

```
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

뷰 클래스의 XML 속성

- ▶ **layout_margin**
 - ▶ 위젯과 위젯 사이에 여백 설정

[예제 4-12] layout_margin 속성 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

뷰 클래스의 XML 속성

▶ visibility

- ▶ 위젯을 보일 것인지 여부 결정
- ▶ visible → 보이는 상태 (default)
- ▶ invisible → 보이지 않는 상태 (원래 자리는 유지)
- ▶ gone → 보이지 않는 상태 (원래 자리도 없어짐)

[예제 4-13] visibility 속성 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" />
```



뷰 클래스의 XML 속성

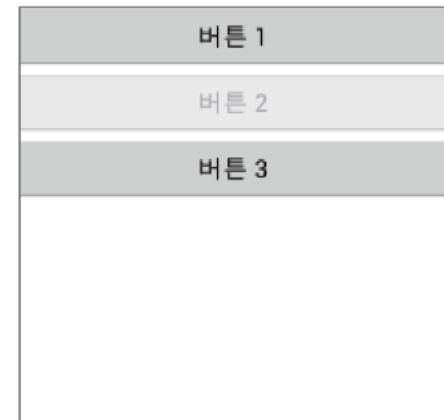
▶ enable

- ▶ 위젯의 동작 여부

▶ clickable

- ▶ 클릭이나 터치가 가능하도록 설정

```
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" />
```

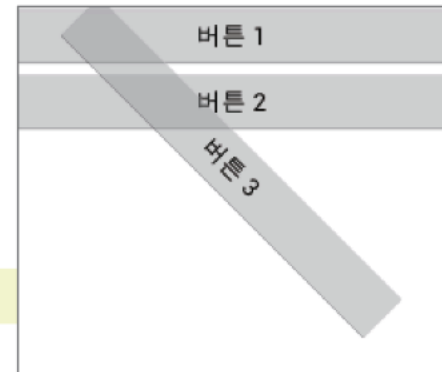


뷰 클래스의 XML 속성

▶ rotation

- ▶ 위젯을 회전시켜 출력
 - ▶ 값은 각도로 지정

```
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" />
```



```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical" >
6
7   <Button
8       android:layout_width="match_parent"
9       android:layout_height="wrap_content"
10      android:text="# 1" />
11
12   <Button
13       android:layout_width="match_parent"
14       android:layout_height="wrap_content"
15      android:text="# 2" />
16
17   <Button
18       android:layout_width="match_parent"
19       android:layout_height="wrap_content"
20       android:rotation="45"
21      android:text="# 3" />
22
23 </LinearLayout>
24
```

텍스트 뷰

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

android:singleLine	setTransformationMethod(TransformationMethod)	Constrains the text to a single horizontally scrolling line instead of letting it wrap onto multiple lines, and advances focus instead of inserting a newline when you press the enter key.
android:text	setText(CharSequence,TextView.BufferType)	Text to display.
android:textAllCaps	setAllCaps(boolean)	Present the text in ALL CAPS.
android:textAppearance		Base text color, typeface, size, and style.
android:textColor	setTextColor(int)	Text color.
android:textColorHighlight	setHighlightColor(int)	Color of the text selection highlight.
android:textColorHint	setHintTextColor(int)	Color of the hint text.
android:textColorLink	setLinkTextColor(int)	Text color for links.
android:textIsSelectable	isTextSelectable()	Indicates that the content of a non-editable text can be selected.
android:textScaleX	setTextScaleX(float)	Sets the horizontal scaling factor for the text.
android:textSize	setTextSize(int,float)	Size of the text.
android:textStyle	setTypeface(Typeface)	Style (bold, italic, bolditalic) for the text.
android:typeface	setTypeface(Typeface)	Typeface (normal, sans, serif, monospace) for the text.
android:width	setWidth(int)	Makes the TextView be exactly this many pixels wide.



텍스트 뷰

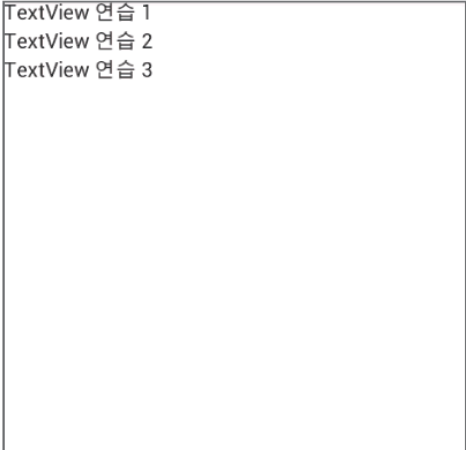
```
1 <TextView
2     android:textSize="30dp"
3     android:text="textSize 속성" />
4 <TextView
5     android:textSize="30dp"
6     android:textColor="#00FF00"
7     android:text="textColor 속성" />
8 <TextView
9     android:textSize="30dp"
10    android:textStyle="bold|italic"
11    android:text="textStyle 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...

텍스트 뷰

▶ XML 속성 설정

```
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

텍스트 뷰

▶ XML 속성을 변경하는 코드 - Java

```
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

에디트 텍스트

EditText

extends `TextView`

`java.lang.Object`

↳ `android.view.View`

↳ `android.widget.TextView`

↳ `android.widget.EditText`

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

① 에디트텍스트 변수 선언

```
EditText myEdit;
```

② 변수에 에디트텍스트 위젯 대입

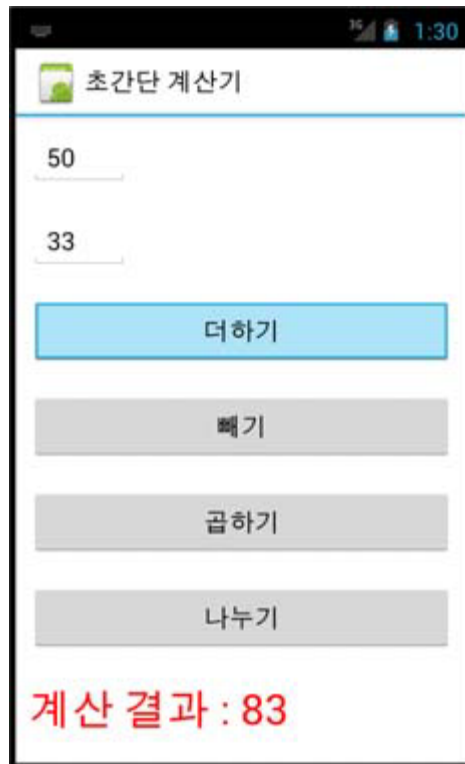
```
myEdit = (EditText) findViewById(R.id.edittext1);
```

③ 에디트텍스트에 입력된 값을 가져오기

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

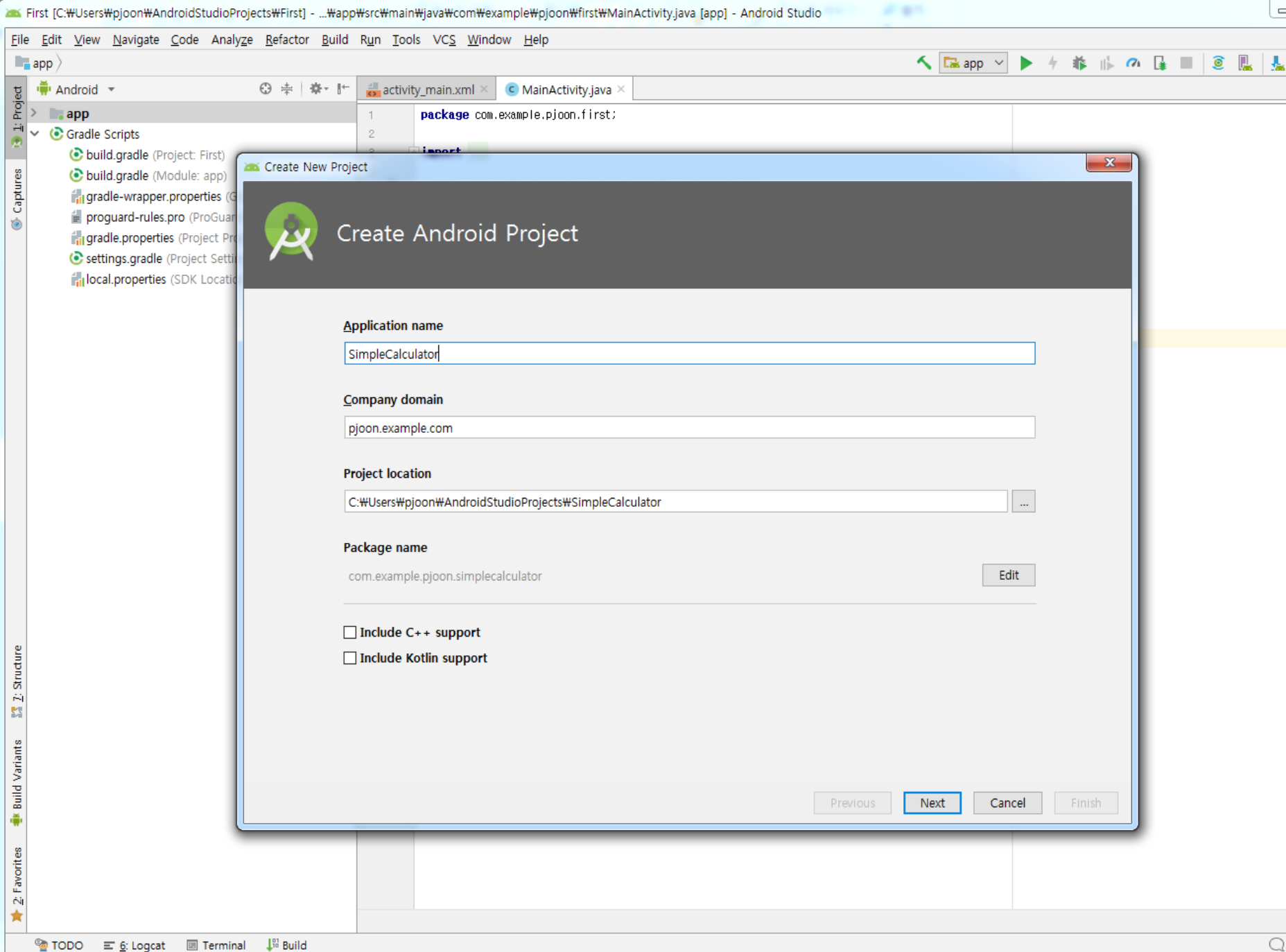


계산기 만들기



에디트텍스트 2개
버튼 4개
텍스트뷰 1개







Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**

API 15: Android 4.0.3 (IceCreamSandwich) ▼

By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)

☐ Include Android Instant App support

☐ **Wear OS**

API 23: Android 6.0 (Marshmallow) ▼

☐ **TV**

API 21: Android 5.0 (Lollipop) ▼

☐ **Android Auto**

☐ **Android Things**

API 24: Android 7.0 (Nougat) ▼

Previous

Next

Cancel

Finish

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical" >
6
7   <EditText
8       android:id="@+id/Edit1"
9       android:layout_width="wrap_content"
10      android:layout_height="wrap_content"
11      android:layout_margin="10dp"
12      android:hint="숫자1" />
13
14   <EditText
15       android:id="@+id/Edit2"
16       android:layout_width="wrap_content"
17       android:layout_height="wrap_content"
18       android:layout_margin="10dp"
19       android:hint="숫자2" />
20
21   <Button
22       android:id="@+id/BtnAdd"
23       android:layout_width="match_parent"
24       android:layout_height="wrap_content"
25       android:layout_margin="10dp"
26       android:text="더하기" />
```

```
28 <Button
29     android:id="@+id/BtnSub"
30     android:layout_width="match_parent"
31     android:layout_height="wrap_content"
32     android:layout_margin="10dp"
33     android:text="빼기" />
```

```
35 <Button
36     android:id="@+id/BtnMul"
37     android:layout_width="match_parent"
38     android:layout_height="wrap_content"
39     android:layout_margin="10dp"
40     android:text="곱하기" />
```

```
42 <Button
43     android:id="@+id/BtnDiv"
44     android:layout_width="match_parent"
45     android:layout_height="wrap_content"
46     android:layout_margin="10dp"
47     android:text="나누기" />
```

```
48
49 <TextView
50     android:id="@+id/TextResult"
51     android:layout_width="wrap_content"
52     android:layout_height="wrap_content"
53     android:layout_margin="10dp"
54     android:text="계산 결과: "
55     android:textColor="#FF0000"
56     android:textSize="30dp" />
```

```
58 </LinearLayout>
```



```

1 package com.example.pjoon.simplecalculator;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.widget.TextView;
6
7 public class MainActivity extends AppCompatActivity {
8
9     EditText edit1, edit2;
10    Button btnAdd, btnSub, btnMul, btnDiv;
11    TextView textResult;
12    String num1, num2;
13    Integer result;
14
15    @Override
16    protected void onCreate(Bundle savedInstanceState) {
17        super.onCreate(savedInstanceState);
18        setContentView(R.layout.activity_main);
19    }
20 }

```

@Override

```
protected 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);  
    textResult = (TextView) findViewById(R.id.TextResult);  
}
```



```

22 @Override
23 protected void onCreate(Bundle savedInstanceState) {
24     super.onCreate(savedInstanceState);
25     setContentView(R.layout.activity_main);
26
27     setTitle("초간단 계산기");
28
29     edit1 = (EditText) findViewById(R.id.Edit1);
30     edit2 = (EditText) findViewById(R.id.Edit2);
31     btnAdd = (Button) findViewById(R.id.BtnAdd);
32     textResult = (TextView) findViewById(R.id.TextResult);
33

```

```


34     btnAdd.setOnT



```

```

35 }
36
37 @Override
38 public bool
39     // Infl
40     getMenu
41     return
42 }
43
44 @Override
45 public bool
46     // Hand
47     // auto
48     // as y
49     int id
50     if (id
51     ret
52 }
53 return super.onOptionsItemSelected(item);

```

 Enable intelligent code completion?
☒ setOnTouchListener(OnTouchListener l) : void - View

Press 'Ctrl+Space' to show Template Proposals

Code Recommenders is available in your Eclipse installation. If you want to make intelligent code completion your default Content Assist, click [here](#) or simply press Return. If you do not want to make it the default, click [here](#).

You can configure Code Recommenders in the [Code Recommenders > Completions](#) preferences.

To learn more about Code Recommenders, please visit the [project homepage](#).

Eclipse Code Recommenders

protected void

```
protected 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);  
    textResult = (TextView) findViewById(R.id.TextResult);  
  
    btnAdd.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public boolean onTouch(View v, MotionEvent event) {  
            num1 = edit1.getText().toString();  
            num2 = edit2.getText().toString();  
            result = Integer.parseInt(num1) + Integer.parseInt(num2);  
            textResult.setText("계산 결과 : " + result.toString());  
            return false;  
        }  
    });  
}
```



4:05



초간단 계산기

30

20

더하기

빼기

곱하기

나누기

계산 결과 : 50

계산기 확장

- ▶ 터치가 아닌 클릭으로 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



```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical" >
6
7   <EditText
8       android:id="@+id/Edit1"
9       android:layout_width="wrap_content"
10      android:layout_height="wrap_content"
11      android:layout_margin="6dp"
12      android:hint="숫자1"
13      android:textColorHint="#00FF00" />
14
15   <EditText
16       android:id="@+id/Edit2"
17       android:layout_width="wrap_content"
18       android:layout_height="wrap_content"
19       android:layout_margin="6dp"
20       android:hint="숫자2"
21       android:textColorHint="#FFFF00" />
22
23   <Button
24       android:id="@+id/BtnAdd"
25       android:layout_width="match_parent"
26       android:layout_height="wrap_content"
27       android:layout_margin="6dp"
28       android:text="더하기" />
```

```
<Button
    android:id="@+id/BtnSub"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="6dp"
    android:text="빼기" />
```

```
<Button
    android:id="@+id/BtnMul"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="6dp"
    android:text="곱하기" />
```

```
<Button
    android:id="@+id/BtnDiv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="6dp"
    android:text="나누기" />
```

```
<Button
    android:id="@+id/BtnRem"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="6dp"
    android:text="나머지 구하기" />
```



```
58     <TextView
59         android:id="@+id/TextResult"
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:layout_margin="6dp"
63         android:text="계산 결과: "
64         android:textColor="#FF0000"
65         android:textSize="30dp" />
66
67 </LinearLayout>
```

```
package com.example.pjoon.calc2;
```

```
import ...
```

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

```
    EditText edit1, edit2;
```

```
    Button btnAdd, btnSub, btnMul, btnDiv, btnRem;
```

```
    TextView textResult;
```

```
    String num1, num2;
```

```
    Double result; // 실수값으로 계산
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

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

```
        setTitle("초간단 계산기(수정)");
```

```
        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);
```

```
        btnRem = (Button) findViewById(R.id.BtnRem);
```

```
        textResult = (TextView) findViewById(R.id.TextResult);
```

```
    }
```

```
java / com / example / pjoon / calc2 / MainActivity  
1 package com.example.pjoon.calc2;  
2  
3 import ...  
4  
5  
6  
7  
8  
9  
10 public class MainActivity extends AppCompatActivity {  
11  
12     EditText edit1, edit2;  
13     Button btnAdd, btnSub, btnMul, btnDiv, btnRem;  
14     TextView textResult;  
15     String num1, num2;  
16     Double result; // 실수값으로 계산  
17  
18     @Override  
19     protected void onCreate(Bundle savedInstanceState) {  
20         super.onCreate(savedInstanceState);  
21         setContentView(R.layout.activity_main);  
22         setTitle("초간단 계산기(수정)");  
23  
24         edit1 = (EditText) findViewById(R.id.Edit1);  
25         edit2 = (EditText) findViewById(R.id.Edit2);  
26  
27         btnAdd = (Button) findViewById(R.id.BtnAdd);  
28         btnSub = (Button) findViewById(R.id.BtnSub);  
29         btnMul = (Button) findViewById(R.id.BtnMul);  
30         btnDiv = (Button) findViewById(R.id.BtnDiv);  
31         btnRem = (Button) findViewById(R.id.BtnRem);  
32  
33         textResult = (TextView) findViewById(R.id.TextResult);  
34  
35         btnRem.setOnClickListener(new View.OnClickListener() {  
36  
37  
38  
39
```

Click or press Alt+Enter

- Implement methods
- Add import for 'android.view.View.OnClickListener' ▶
- Annotate interface 'OnClickListener' as @Deprecated ▶

Select Methods to Implement

android.view.View.OnClickListener

onClick(v:View):void

☐ Copy JavaDoc

☒ Insert @Override

OK

Cancel

```

35 btnRem.setOnClickListener(new View.OnClickListener() {
36     @Override
37     public void onClick(View v) {
38         num1 = edit1.getText().toString();
39         num2 = edit2.getText().toString();
40         // num1이나 num2가 비어 있다면
41         if (num1.trim().equals("") || num2.trim().equals("")) {
42             Toast.makeText(getApplicationContext(), "입력 값이 비었습니다", Toast.LENGTH_SHORT)
43                 .show();
44         } else {
45             // num2가 0이면 나누지 않는다.
46             if (num2.trim().equals("0")) {
47                 Toast.makeText(getApplicationContext(),
48                     "0으로 나머지 값 안됩니다!", Toast.LENGTH_SHORT).show();
49             } else {
50                 result = Double.parseDouble(num1)
51                     % Double.parseDouble(num2);
52                 textResult.setText("계산 결과 : " + result.toString());
53             }
54         }
55     }
56 });
57
58 }

```

