

# Ch04. 기본 위젯 익히기1

모바일앱프로그래밍  
김지심 교수

# 학습목표



뷰와 뷰 상속을 이해한다.  
기본 위젯을 작성할 수 있다.

# 목차

---

1. 뷰의 개요
2. 기본 위젯 다루기
3. 기본 위젯 활용하기



01



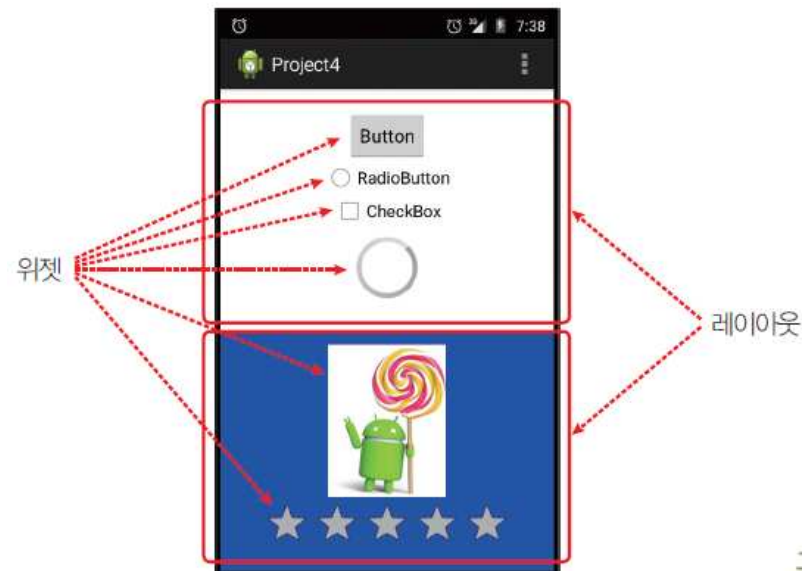
## 뷰의 개요



# 1. 뷰의 개요

## View 클래스

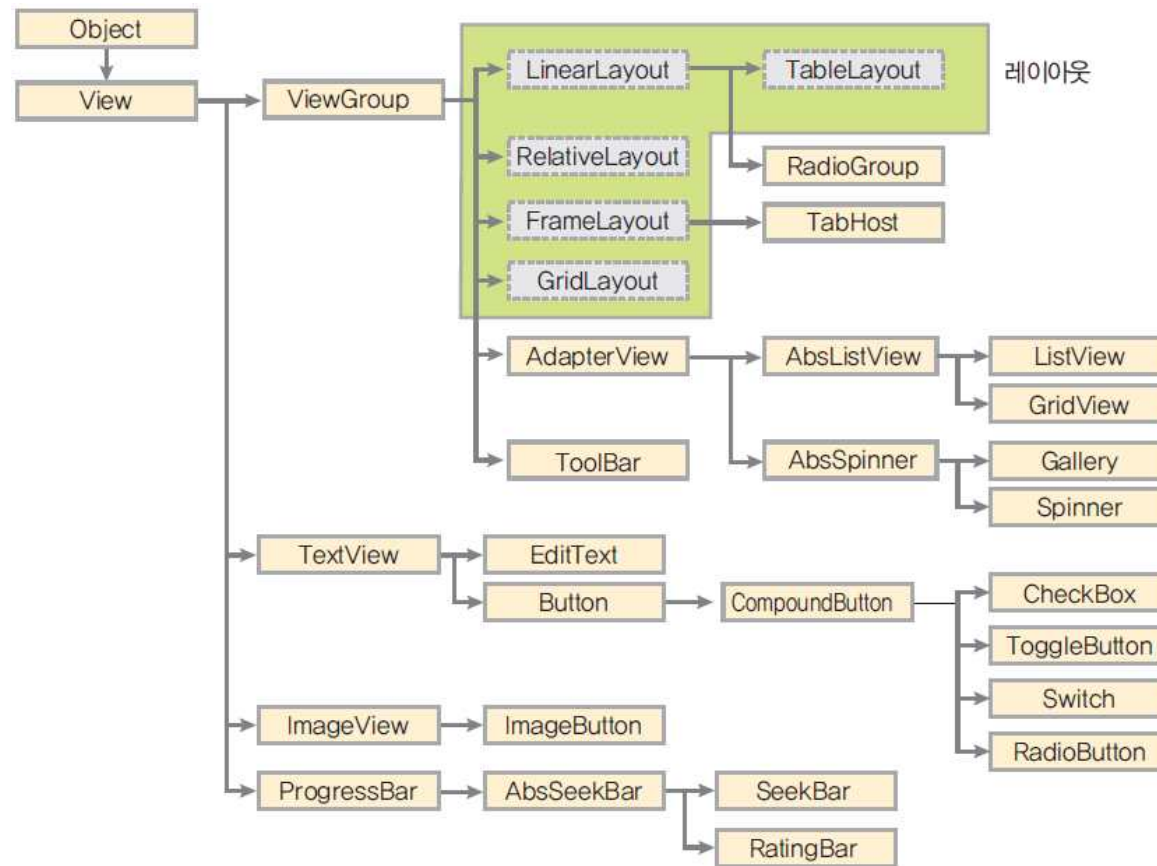
안드로이드 화면에서 실제로 사용되는 것들은 모두 View 클래스 상속을 받음  
뷰 클래스는 다른 말로 ' 위젯 ' 이라고도 함  
다른 위젯을 담을 수 있는 위젯은 레이아웃이라고 함  
레이아웃은 ViewGroup 클래스의 하위 클래스





# 1. 뷰의 개요

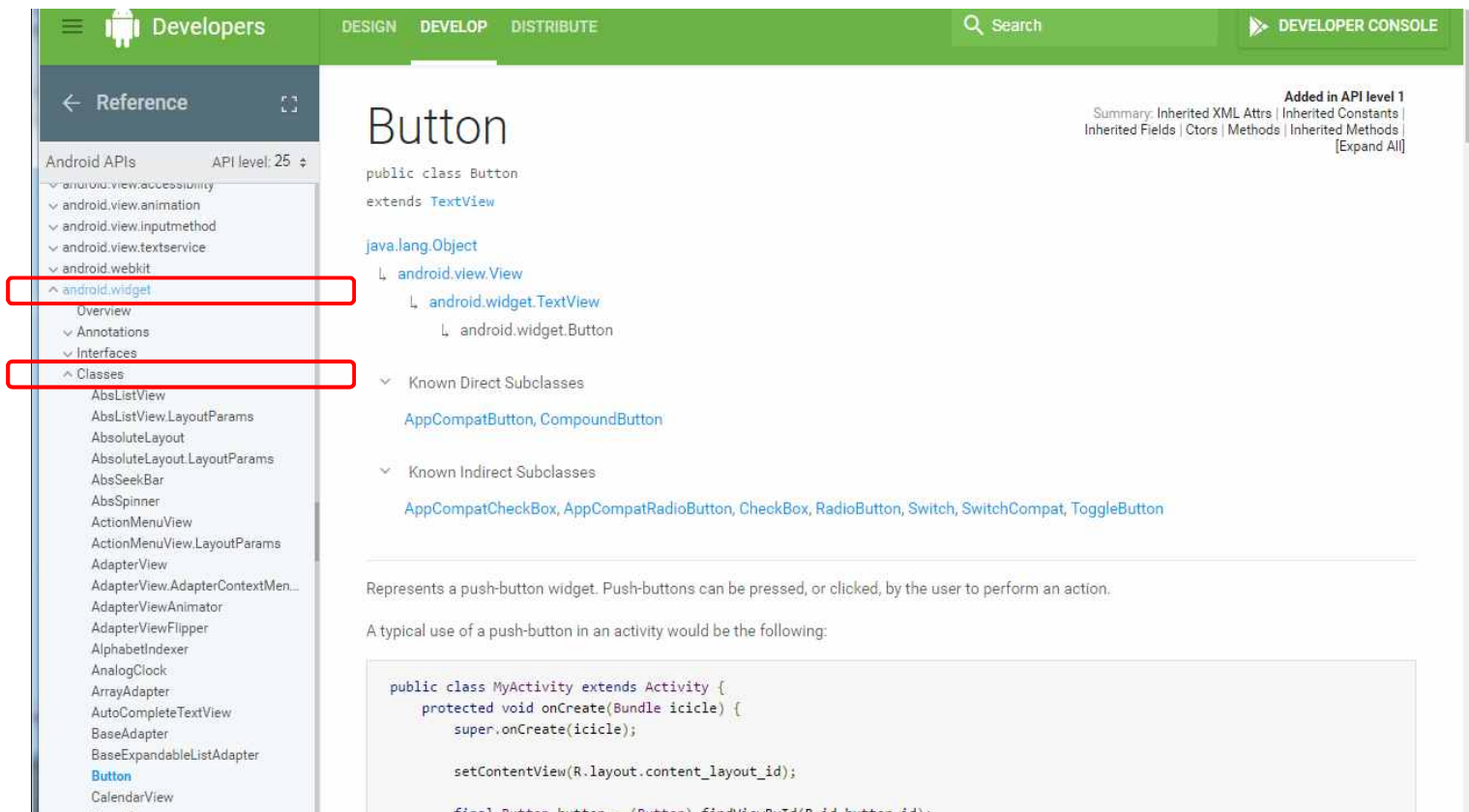
## View 클래스 계층도



# 1. 뷰의 개요

## 클래스 상속관계 찾기

<http://developer.android.com/reference>에서 자세히 확인  
android.widget > Classes > Button



# 1. 뷰의 개요

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

## id 속성

id 속성은 모든 뷰의 아이디를 나타냄

Java 코드에서 뷰에 접근할 때 id 속성에 지정한 아이디를 사용

id 속성은 뷰에 아이디를 새로 부여하는 개념이므로 “@+id/” 형식으로 지정  
/ 다음에는 새로 지정할 id를 적음

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

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



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

# 1. 뷰의 개요

id 속성 예제: Ex04\_01

The image shows the Android Studio interface. On the left, the XML layout file is open, displaying the following code:

```
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:id="@+id/tvGender"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:text="성별 선택"
12    tools:text="Select Gender"
13  />
14
15  <RadioButton
16    android:layout_width="wrap_content"
17    android:layout_height="wrap_content"
18    android:id="@+id/rdFemale"
19    android:text="여성"
20    tools:text="Female"/>
21
22  <RadioButton
23    android:layout_width="wrap_content"
24    android:layout_height="wrap_content"
25    android:id="@+id/rdMale"
26    android:text="남성"
27    tools:text="Male"/>
28
29 </LinearLayout>
```

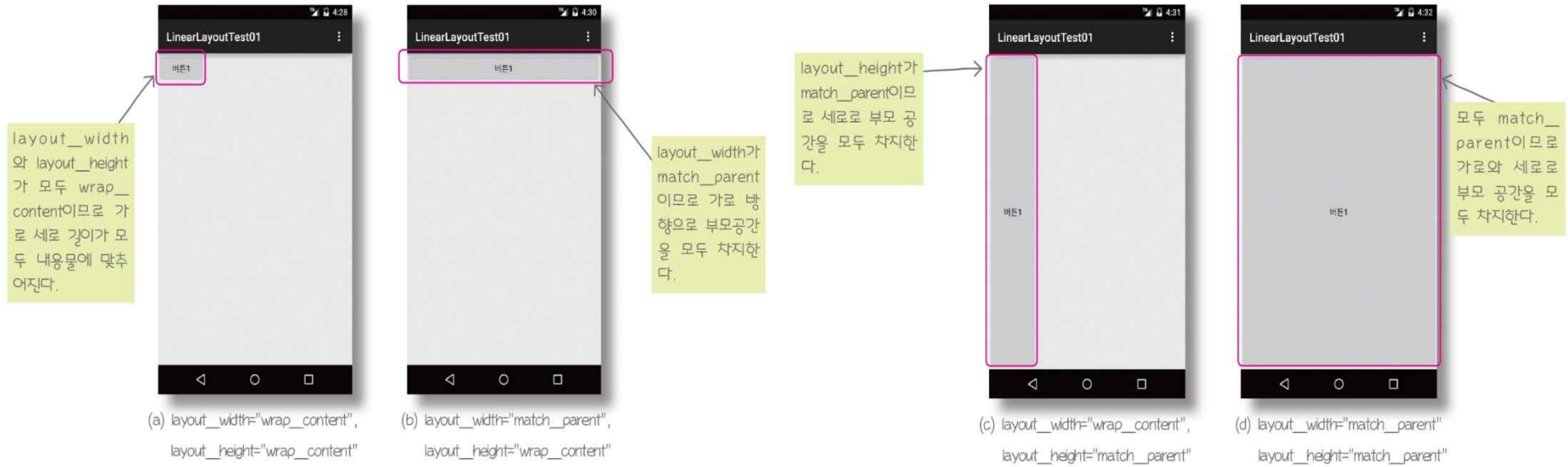
On the right, the preview window shows the rendered UI. The title bar indicates the app name "04\_4-1". The main content area displays the text "성별 선택" (Gender Selection) followed by two radio buttons: "여성" (Female) and "남성" (Male). The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

# 1. 뷰의 개요

## layout\_width, layout\_height 속성

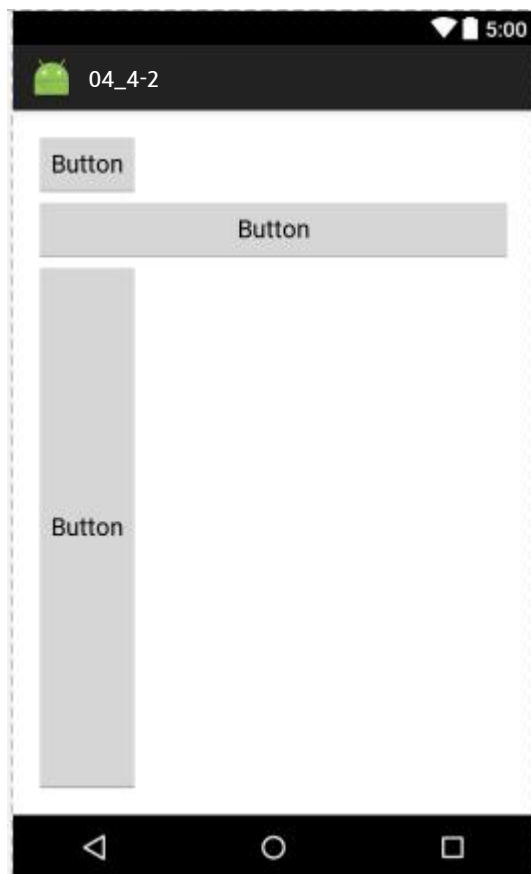
match\_parent : 자신의 부모(보통레이아웃)에 폭이나 높이를 맞춘다는 의미

wrap\_content : 자신의 폭이나 높이를 자신 안의 글자가 들어갈 정도로 설정



# 1. 뷰의 개요

Width, height 속성 예제: Ex04\_02



```
LinearLayout
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical" >
6
7      <Button
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="버튼입니다" />
11
12     <Button
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:text="버튼입니다" />
16
17     <Button
18         android:layout_width="wrap_content"
19         android:layout_height="match_parent"
20         android:text="버튼입니다" />
21 </LinearLayout>
```

# 1. 뷰의 개요

## background 속성 예제: Ex04\_07

background 속성은 뷰의 색상을 주로 #RRGGBB 값으로 지정



# 1. 뷰의 개요

## 예제 4-12: Ex04\_12

padding 속성

뷰의 경계와 내용 사이의 간격

layout\_margin 속성

뷰와 뷰 사이의 간격

```
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   android:padding="30dp" >
7
8   <TextView
9       android:layout_width="match_parent"
10      android:layout_height="wrap_content"
11      android:layout_margin="20dp"
12      android:text="아래에 이름을 입력 : " />
13
14   <EditText
15       android:layout_width="match_parent"
16       android:layout_height="wrap_content"
17       android:layout_margin="20dp"
18       android:hint="여기에 채우세요" />
19
20   <Button
21       android:id="@+id/button1"
22       android:layout_width="match_parent"
23       android:layout_height="wrap_content"
24       android:layout_margin="20dp"
25       android:text="확인" />
26
27 </LinearLayout>
```





# 1. 뷰의 개요

## 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.  |

# 1. 뷰의 개요

## 예제 4-13: Ex04\_13

### visibility 속성

visibility 속성은 뷰를 보일 것인지 여부를 결정

디폴트인 visible은 보이는 상태, invisible과 gone은 안 보이는 상태

Invisible vs. gone ?

```
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   <Button
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content"
10    android:text="버튼 1" />
11
12   <Button
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:text="버튼 2"
16     android:visibility="invisible"/>
17
18   <Button
19     android:layout_width="match_parent"
20     android:layout_height="wrap_content"
21     android:text="버튼 3"
22     android:visibility="visible" />
23
24   <Button
25     android:layout_width="match_parent"
26     android:layout_height="wrap_content"
27     android:text="버튼 4"
28     android:visibility="gone" />
29
30   <Button
31     android:layout_width="match_parent"
32     android:layout_height="wrap_content"
33     android:text="버튼 5" />
34
35 </LinearLayout>
```

# 1. 뷰의 개요

## 예제 4-14: Ex04\_14

### enabled, clickable 속성

enabled 속성 : 뷰의 동작 여부 (비활성화)

clickable 속성 : 클릭이나 터치가 가능하도록 함

```
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   <Button
8       android:layout_width="match_parent"
9       android:layout_height="wrap_content"
10      android:text="버튼 1" />
11
12   <Button
13       android:layout_width="match_parent"
14       android:layout_height="wrap_content"
15       android:enabled="false"
16       android:text="버튼 2" />
17
18   <Button
19       android:layout_width="match_parent"
20       android:layout_height="wrap_content"
21       android:clickable="false"
22       android:text="버튼 3" />
23
24 </LinearLayout>
```

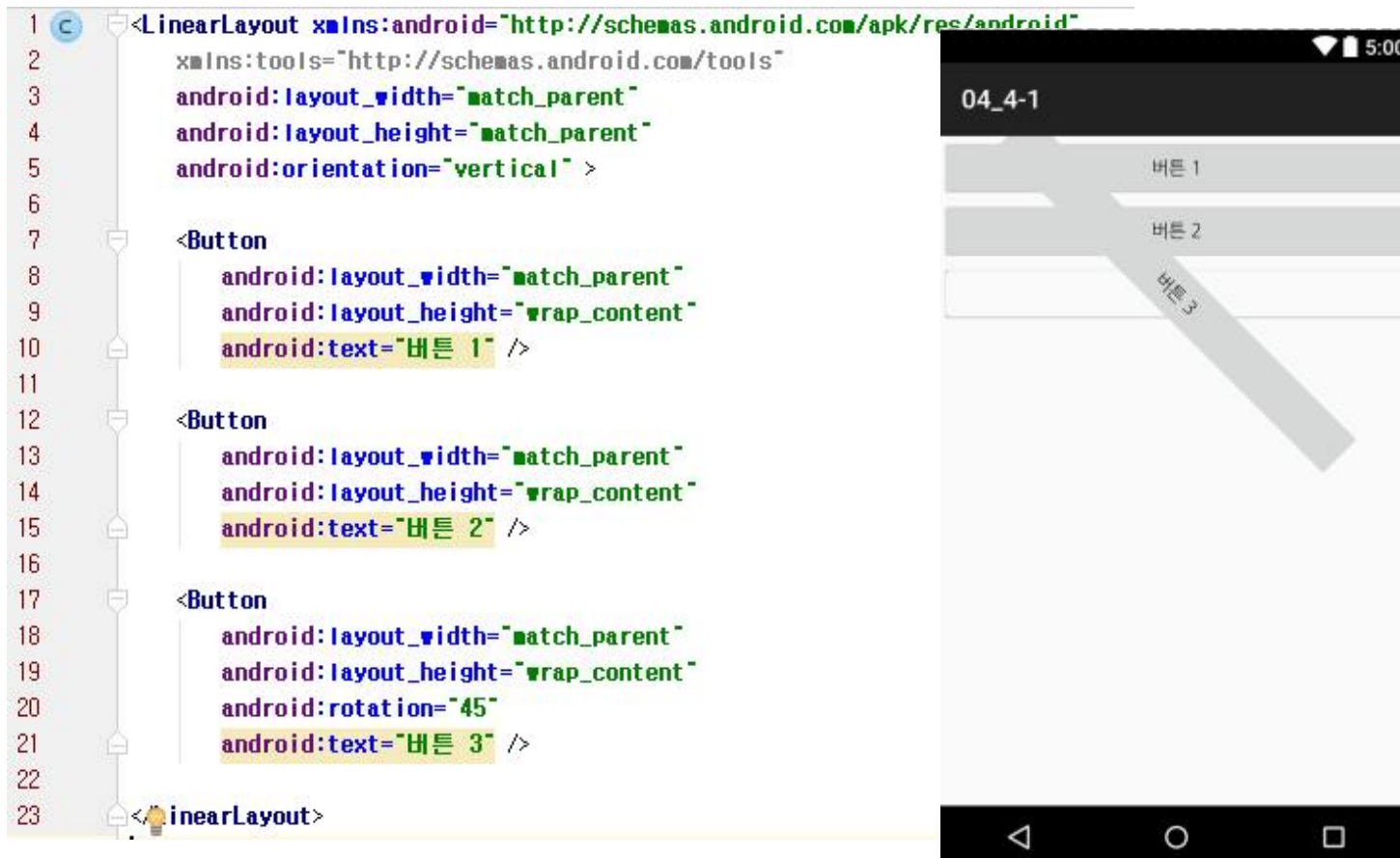


# 1. 뷰의 개요

## 예제 4-15: Ex04\_15

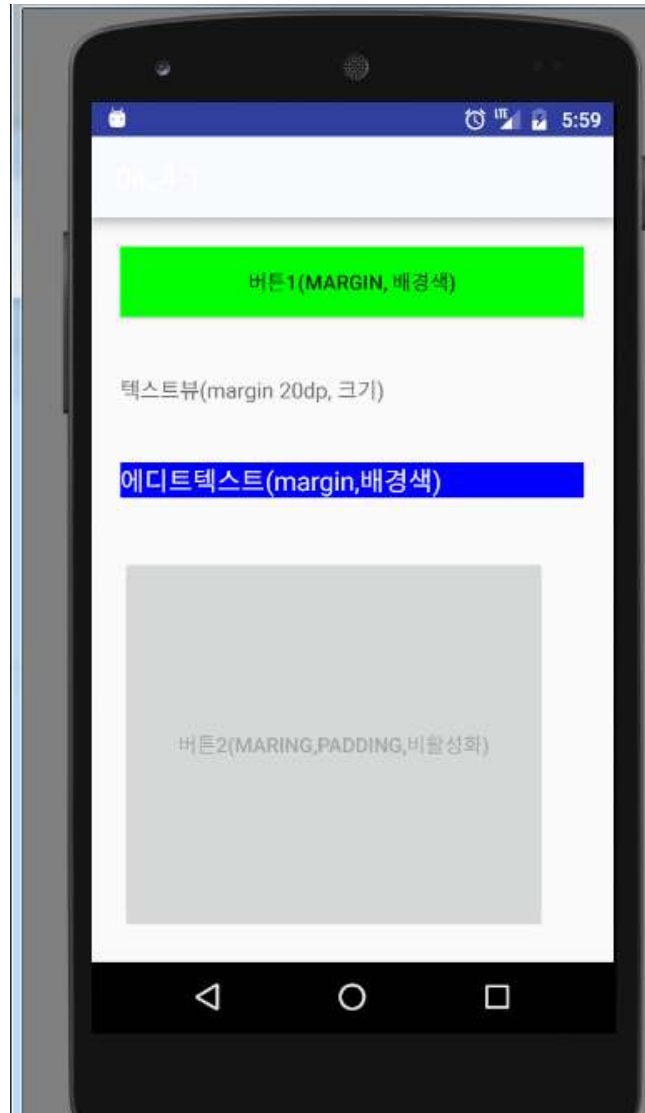
### rotation 속성

rotation은 뷰를 회전시켜서 출력  
값은 각도(degree)로 지정



# 1. 뷰의 개요

직접 풀어보기:Test04\_02







## 02 기본 뷰 다루기





## 2. 기본 뷰 다루기

### 계층도

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

TextView 계층도

textColor 속성 : 글자의 색상을 지정, #RRGGBB나 #AARRGGBB 형식

textSize 속성 : 글자의 크기를 sp, dp, px, in, mm 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 출력

값으로 true와 false를 설정할 수 있으며 디폴트는 false

## 2. 기본 뷰 다루기

예제 4-16: Ex04\_16

텍스트뷰 예제

activity\_main.xml로 화면을 기본적으로 디자인하고,  
java코드에서 재디자인



## 2. 기본 뷰 다루기

### 화면 디자인

#### activity\_main.xml

```
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="wrap_content"
9       android:layout_height="wrap_content"
10      android:text="textView 속성의 다양한 속성들은 이와 같음"
11      android:textSize="30dp"
12      android:textColor="#00FF00"
13      android:textStyle="bold|italic"
14      android:typeface="serif"
15      android:singleLine="true"
16   />
17
18   <TextView
19       android:text="TextView"
20       android:layout_width="match_parent"
21       android:layout_height="wrap_content"
22       android:id="@+id/tv2" />
23
24 </LinearLayout>
```

## 2. 기본 뷰 다루기

텍스트뷰 속성을 변경하기 위해 Java 코딩

MainActivity.java : XML로 정적으로 디자인한 것을 JAVA 코딩으로  
동적 디자인하여 재설정

```
1 package com.android.kjs.ex_04_16;
2
3 import android.graphics.Color;
4 import android.graphics.Typeface;
5 import android.os.Bundle;
6 import android.support.v7.app.AppCompatActivity;
7 import android.widget.TextView;
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         TextView tv2;
17         tv2 = (TextView) findViewById(R.id.tv2);
18
19         tv2.setText("동일한 텍스트뷰의 속성 테스트");
20         tv2.setTextColor(Color.RED);
21         tv2.setTextSize(30);
22         tv2.setTypeface(Typeface.MONOSPACE, Typeface.BOLD_ITALIC);
23         tv2.setSingleLine();
24     }
25 }
26
```

## 2. 기본 뷰 다루기

### 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 |

## 2. 기본 뷰 다루기

### 버튼

사용자의 이벤트를 처리하기 위한 가장 기본적인 뷰로 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?" />
```



## 2. 기본 뷰 다루기

### 버튼 계층도

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

Button 계층도

## 2. 기본 뷰 다루기

### 일반적인 버튼의 XML 코드

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

#### ❶ 버튼 변수 선언

```
Button mybutton;
```

#### ❷ 변수에 버튼 위젯 대입

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

#### ❸ 버튼 클릭할 때 동작하는 클래스 정의

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

## 2. 기본 뷰 다루기

### 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

## 2. 기본 뷰 다루기

### 일반적인 에디트텍스트의 XML 코드

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

#### ❶ 에디트텍스트 변수 선언

```
EditText myEdit;
```

#### ❷ 변수에 에디트텍스트 위젯 대입

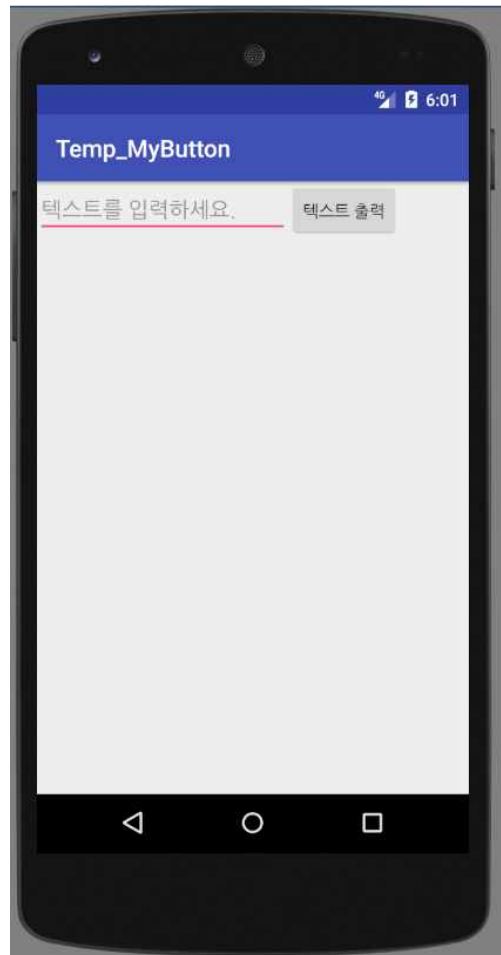
```
myEdit = (EditText) findViewById(R.id.edittext1);
```

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

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

## 2. 기본 뷰 다루기

예제: MyButton



## 2. 기본 뷰 다루기

예제: MyButton

화면 디자인

activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="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   tools:context="com.example.mybutton.MainActivity">
7
8   <EditText
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:text="텍스트를 입력하세요."
12    android:ems="10"
13    android:id="@+id/etInput"/>
14
15   <Button
16     android:layout_width="wrap_content"
17     android:layout_height="wrap_content"
18     android:text="텍스트 출력"
19     android:ems="10"
20     android:id="@+id/btnOutput"/>
21
22 </LinearLayout>
```



## 2. 기본 뷰 다루기

예제: MyButton

MainActivity.java

완성하기!

## 2. 기본 뷰 다루기

### 실습(직접 풀어보기) 4-1: Project04\_01 초간단 계산기 앱 작성

#### 화면 디자인 및 편집

EditText 2개, Button 4개, TextView 1개 생성  
각 뷰에 layout\_margin을 적절히 지정  
TextView는 색상을 빨간색, 글자 크기를 30에  
각 뷰의 id는 위에서부터 차례로 Edit1, Edit2,  
BtnAdd, BtnSub, BtnMul, BtnDiv, TxtResult



## 2. 기본 뷰 다루기

### 화면 디자인 및 편집

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6     <EditText
7         android:id="@+id/Edit1"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:layout_margin="10dp"
11        android:hint="숫자1" />
12     <EditText
13         android:id="@+id/Edit2"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:layout_margin="10dp"
17         android:hint="숫자2" />
18     <Button
19         android:id="@+id/BtnAdd"
20         android:layout_width="match_parent"
21         android:layout_height="wrap_content"
22         android:layout_margin="10dp"
23         android:text="더하기" />
24     <Button
25         android:id="@+id/BtnSub"
26         android:layout_width="match_parent"
27         android:layout_height="wrap_content"
28         android:layout_margin="10dp"
29         android:text="빼기" />
30     <Button
31         android:id="@+id/BtnMul"
32         android:layout_width="match_parent"
33         android:layout_height="wrap_content"
34         android:layout_margin="10dp"
35         android:text="곱하기" />
36     <Button
37         android:id="@+id/BtnDiv"
38         android:layout_width="match_parent"
39         android:layout_height="wrap_content"
40         android:layout_margin="10dp"
41         android:text="나누기" />
42     <TextView
43         android:id="@+id/TxtResult"
44         android:layout_width="wrap_content"
45         android:layout_height="wrap_content"
46         android:layout_margin="10dp"
47         android:text="계산 결과 : "
48         android:textColor="#FF0000"
49         android:textSize="30sp" />
50 </LinearLayout>
```

## 2. 기본 뷰 다루기

### Java 코드 작성

변수 선언 및 뷰 대입

```
11 public class MainActivity extends AppCompatActivity {  
12  
13     EditText Edit1, Edit2;  
14     Button BtnAdd, BtnSub, BtnMul, BtnDiv;  
15     TextView TxtResult;  
16     String num1, num2;  
17     Integer result;  
18  
19     @Override  
20     protected void onCreate(Bundle savedInstanceState) {  
21         super.onCreate(savedInstanceState);  
22         setContentView(R.layout.activity_main);  
23         setTitle("초간단 계산기");  
24  
25         Edit1 = (EditText) findViewById(R.id.Edit1);  
26         Edit2 = (EditText) findViewById(R.id.Edit2);  
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  
32         TxtResult = (TextView) findViewById(R.id.TxtResult);  
33     }  
34 }
```

## 2. 기본 뷰 다루기

### Java 코드 작성

버튼에 터치이벤트 리스너 정의

터치 시에 동작하는 내용을 onTouch() 메소드 안에 코딩

```
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

BtnAdd.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        num1 = Edit1.getText().toString();
        num2 = Edit2.getText().toString();

        result = Integer.parseInt(num1) + Integer.parseInt(num2);

        TxtResult.setText("계산결과 : " + result.toString());

        return false;
    }
});

BtnSub.setOnTouchListener(new View.OnTouchListener() {
```

## 2. 기본 뷰 다루기

### Java 코드 수정1

빼기, 곱하기, 나누기 코드 추가 후 완성

### Java 코드 수정2

1) num1, num2, result를 Integer로 선언



## 2. 기본 뷰 다루기

### 실습(직접 풀어보기) 4-3: Test04\_03 초간단 계산기 수정

TextColorHint 사용

입력 숫자들은 실수값  
(나누기에서는 소수점 아래 1자리까지만 출력)

터치가 아닌 클릭으로 변경

나머지 값 구하기 버튼 추가

값을 입력하지 않고 버튼 클릭 시 오류 메시지 나타내기  
("값을 입력해 주세요.")

(num1,num2는 String으로 선언해서 하기)

0으로 나누면 오류 메시지 나타내기  
("0이 아닌 값을 입력해 주세요.")

액션바(타이틀바)에 본인 학번, 성명 명시하고,  
아이콘 출력(p.76 참고)

