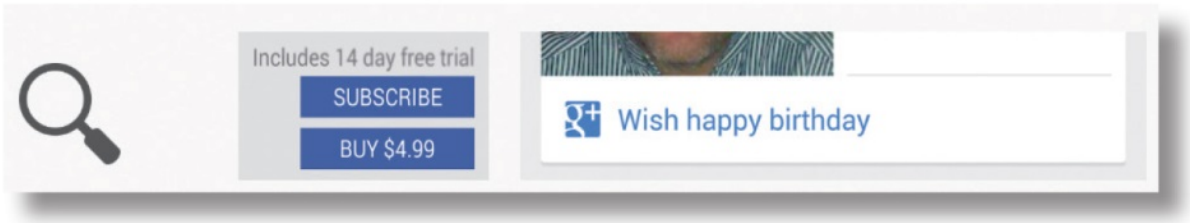
CHAP 3. 사용자 인터페이스 기초

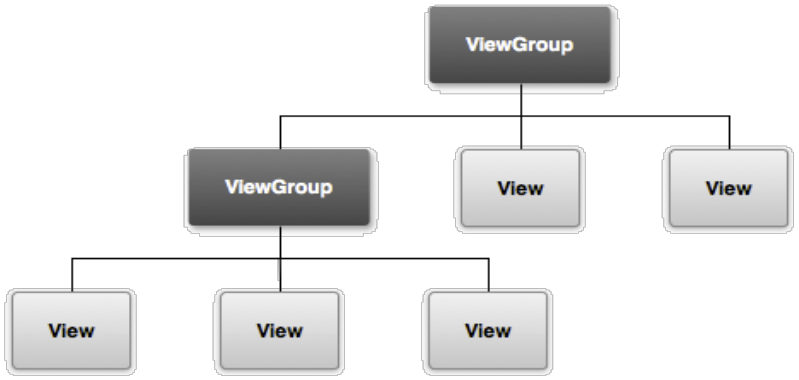
사용자 인터페이스 기초

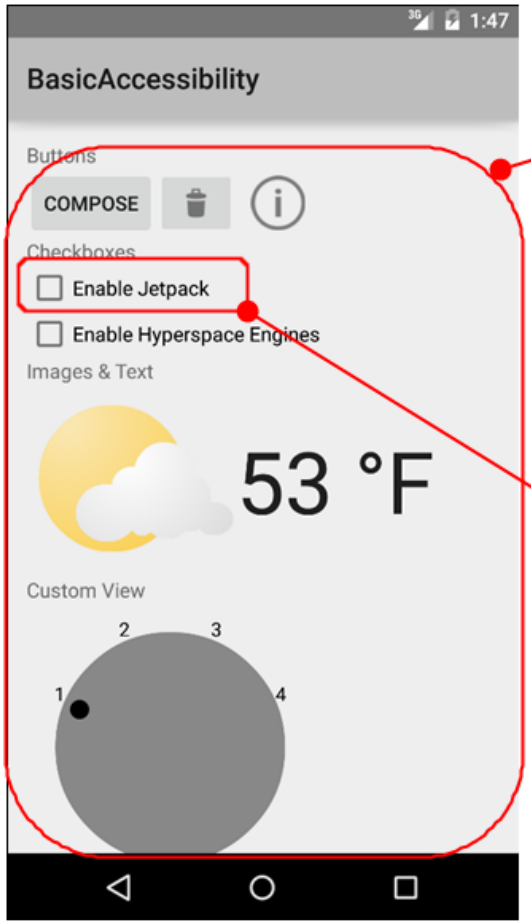
- 자바의 swing 은 사용하지 않음
  - 너무 리소스를 많이 잡아먹음!

- 독자적인 사용자 인터페이스 컨트롤 사용
  - 버튼, 리스트, 스크롤 바, 체크 박스, 메뉴, 대화 상자



View & ViewGroup





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

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

UI를 작성하는 절차

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

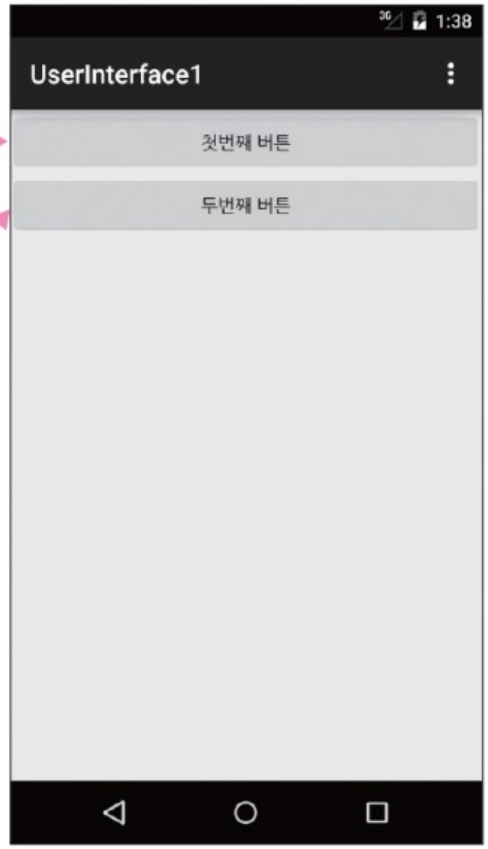
Ui를 작성하는 3가지 방법

① XML로 사용자 인터페이스 작성

```
...
<Button
android:text= "첫번째 버튼"
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content">
</Button>
...
```

② 코드로 사용자 인터페이스 작성

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



XML로 UI 작성

MainActivity | onCreate()

```
package com.example.mhkyung.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }
}
```

LinearLayout | Button

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.mhkyung.myapplication.MainActivity">


    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />
</LinearLayout>
```

LinearLayout 생성(뷰그룹)

버튼 생성

버튼 생성

코드로 UI 생성

MainActivity

```
package com.example.mhkyung.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//        setContentView(R.layout.activity_main);

        LinearLayout    container = new LinearLayout( context: this);
        container.setOrientation(LinearLayout.VERTICAL);
        Button  b1 = new Button( context: this);
        b1.setText("Button 1");
        container.addView(b1);

        Button  b2 = new Button( context: this);
        b2.setText("Button 2");
        container.addView(b2);

        setContentView(container);

    }
}
```

LinearLayout 생성(뷰그룹)

버튼 생성

버튼 생성

XML과 코드를 동시에 사용하는 방법

main.xml

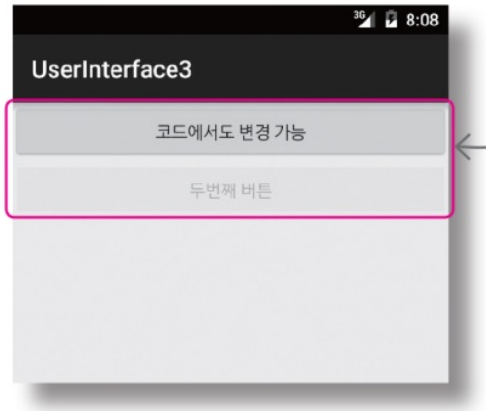
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:text="첫번째 버튼"
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </Button>
    <Button
        android:text="두번째 버튼"
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </Button>
</LinearLayout>
```

버튼에 식별자를 부여한다.

UserInterface3Activity.java

```
package kr.co.company.userinterface3;
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

public class UserInterface3Activity extends ActionBarActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

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

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

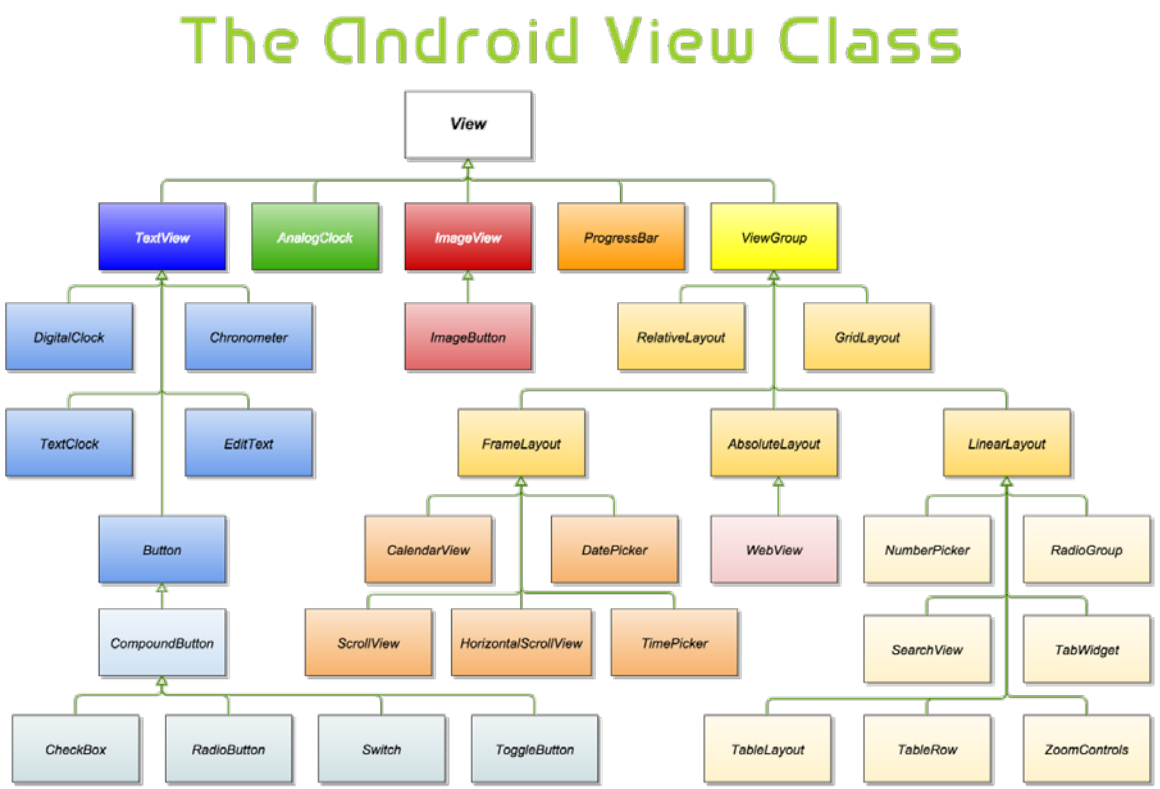
id가 button1인 버튼을 찾는다.



코드에 의하여 버튼의 텍스트와 가시성이 변경되었다.

View

- **View**는 모든 Widget들과 Layout들의 최상위 클래스이다.
- View의 필드와 메소드는 모든 하위 View에서 공통적으로 사용



View의 필드와 메소드

- id
  - View의 식별자
- View의 위치와 크기

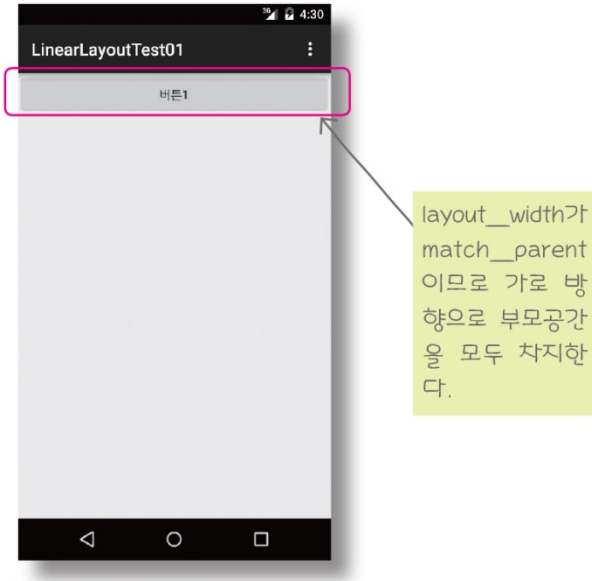
상수	설명
match_parent	부모의 크기를 꽉 채운다(fill_parent도 같은 의미).
wrap_content	뷰가 나타내는 내용물의 크기에 맞춘다.
숫자	크기를 정확히 지정한다.

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="버튼1" />
```

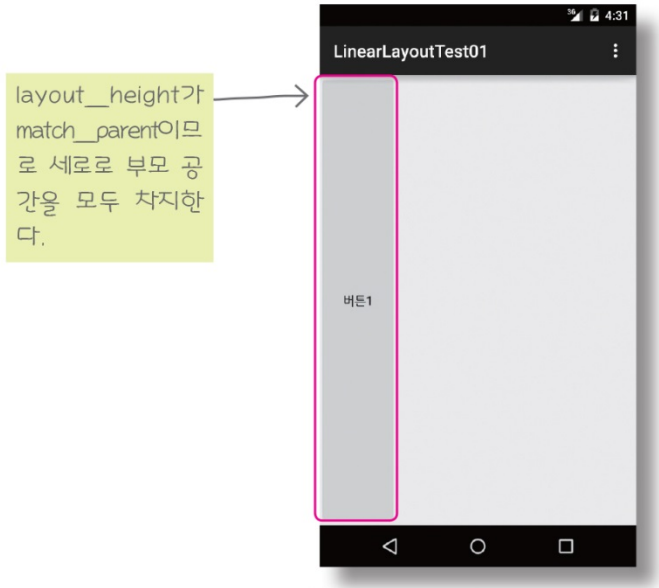
버튼의 위치와 크기를 결정한다.

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="버튼1" />
```

(a) layout_width="wrap_content", layout_height="wrap_content"



(b) layout_width="match_parent", layout_height="wrap_content"



(c) layout_width="wrap_content", layout_height="match_parent"

(d) layout_width="match_parent" layout_height="match_parent"

뷰의 크기 단위

dp = px * 160/screen density

단위	설명
px(pixels)	화면의 실제 픽셀을 나타낸다. 픽셀은 권장되는 단위는 아닌데 그 이유는 장치마다 화면의 밀도가 다르기 때문이다.
dp(density-independent pixels)	dp는 화면의 밀도가 160dpi 화면에서 하나의 물리적인 픽셀을 말한다. 따라서 크기를 160dp로 지정하면 화면의 밀도와는 상관없이 항상 1인치가 된다. dp로 뷰의 크기를 지정하면 화면의 밀도가 다르더라도 항상 동일한 크기로 표시된다.
sp(scale-independent pixels)	화면 밀도와 사용자가 지정한 폰트 크기에 영향을 받아서 변환된다. 이 단위는 폰트 크기를 지정하는 경우에 추천된다.
pt(points)	1/72 인치를 표시한다.
mm(millimeters)	밀리미터를 나타낸다.
in(inches)	인치를 나타낸다.

색상

- 16진수로 투명도와 빛의 3원색인 RGB값을 표시

표시 방법	설명
#RRGGBB	RR은 빨간색 성분, GG는 녹색 성분, BB는 청색 성분을 나타낸다.
#AARRGGBB	AA는 투명도, RR은 빨간색 성분, GG는 녹색 성분, BB는 청색 성분을 나타낸다.

- 16진수로 투명도와 빛의 3원색인 RGB값을 표시

  Ex) Red - #FF0000

  White - #FFFFFF

  Black - #000000

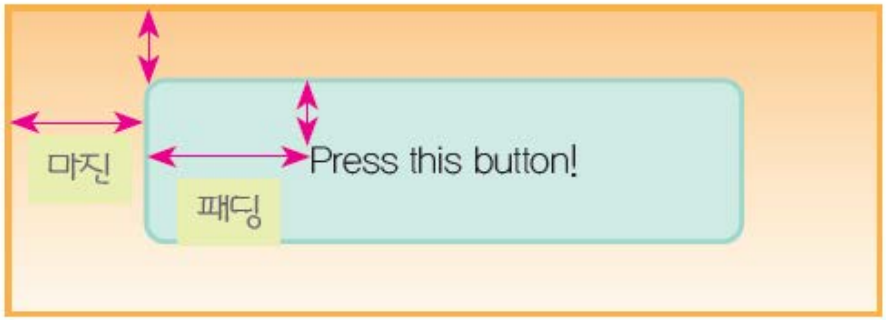
  알파값이 85인 Blue - #550000FF

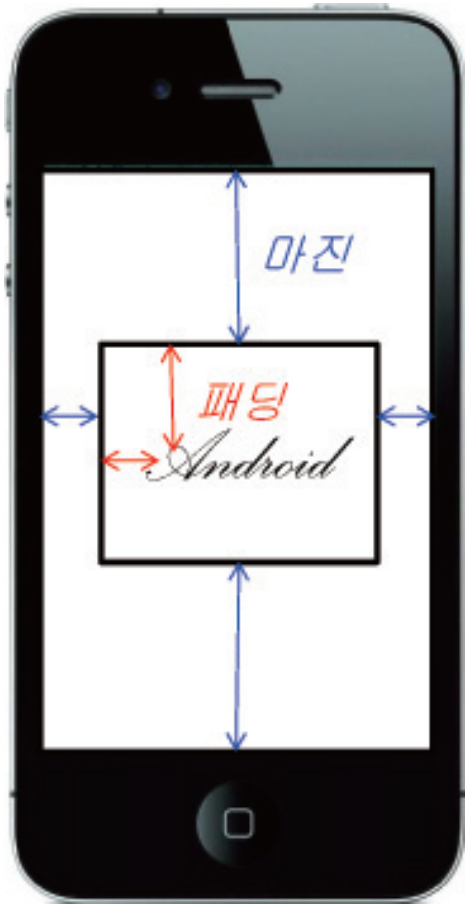
화면에 보이기 속성

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

마진과 패딩

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

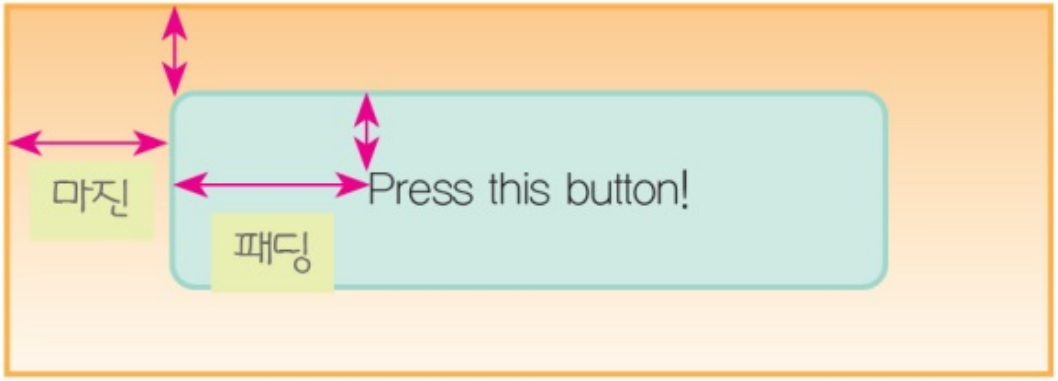




마진과 패딩

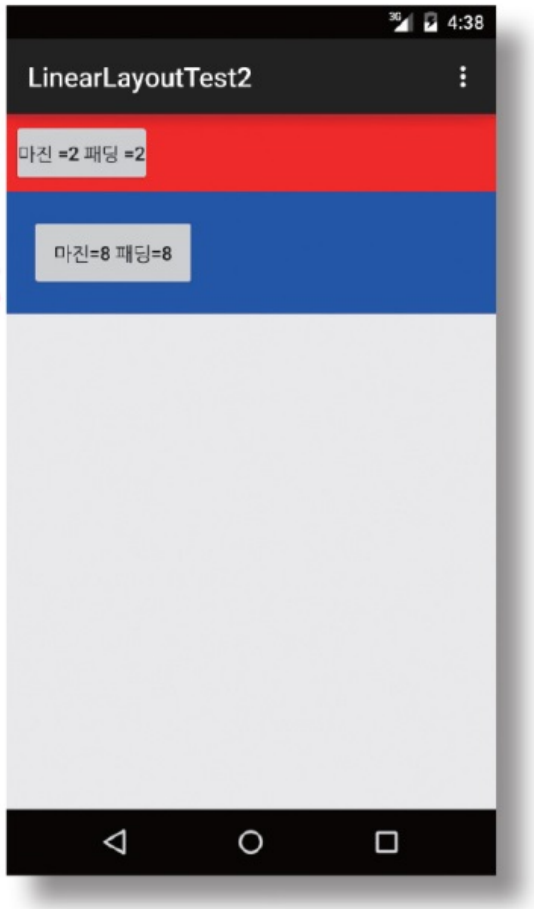
- paddingLeft, paddingRight, paddingTop, paddingBottom

- layout_marginLeft, layout_marginRight, layout_marginTop, layout_marginBottom



마진과 패딩의 예

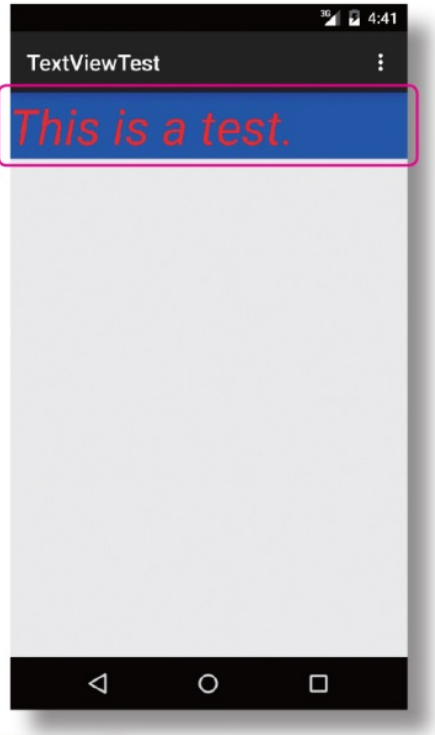
```
<Button
    android:id="@+id/button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="2pt"
    android:padding="2pt"
    android:text="마진=2 패딩=2" />
```



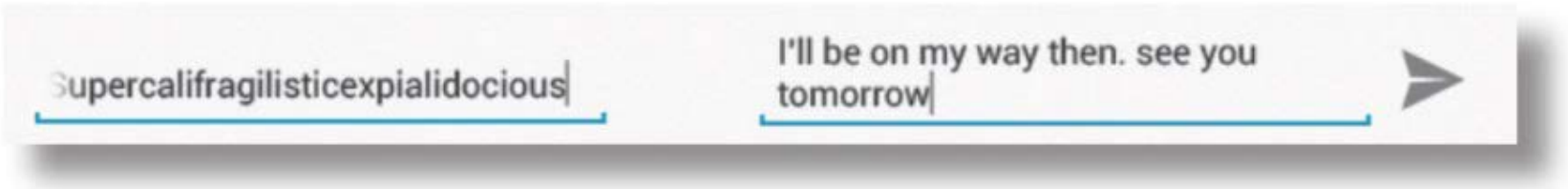
TextView

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#0000ff"
        android:text="This is a test."
        android:textColor="#ff0000"
        android:textSize="20pt"
        android:textStyle="italic"
        android:typeface="serif" />

</LinearLayout>
```



EditText



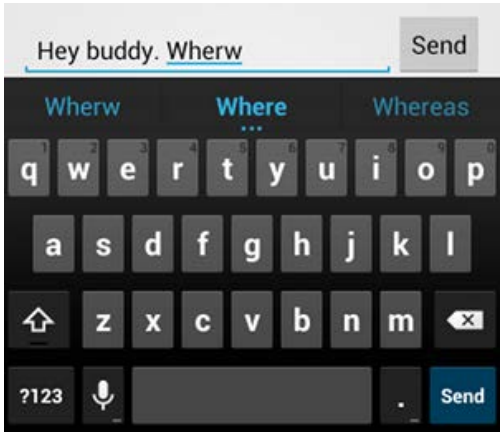
속성	설명
android:autoText	자동으로 타이핑 오류를 교정한다.
android:drawableBottom	텍스트의 아래에 표시되는 이미지 리소스이다.
android:drawableRight	텍스트의 오른쪽에 표시되는 이미지 리소스이다.
android:editable	편집가능
android:text	표시되는 텍스트이다.
android:singleLine	true이면 한 줄만 받음
android:inputType	입력의 종류
android:hint	입력 필드에 표시되는 힌트 메시지

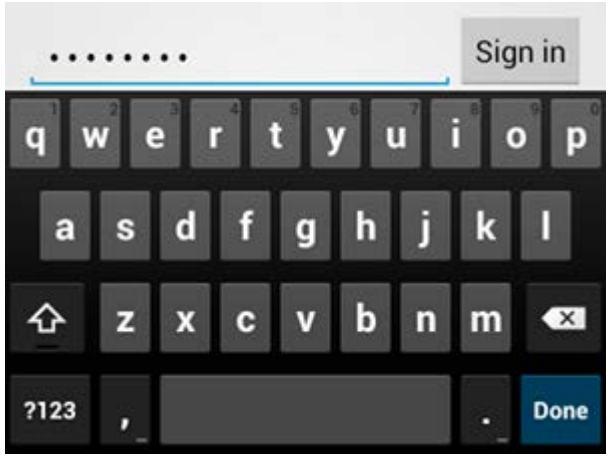
EditText의 inputType 속성

inputType	설명
none	편집이 불가능한 문자열
Text	일반적인 문자열
textMultiLine	여러 줄로 입력 가능
textPostalAddress	우편번호
textEmailAddress	이메일 주소
textPassword	패스워드
textVisiblePassword	패스워드 화면에 보인다.
number	숫자
numberSigned	부호가 붙은 숫자
numberDecimal	소수점이 있는 숫자
phone	전화번호
datetime	시간

android:inputType="phone"

android:inputType=
"textCapSentences |
textAutoCorrect"

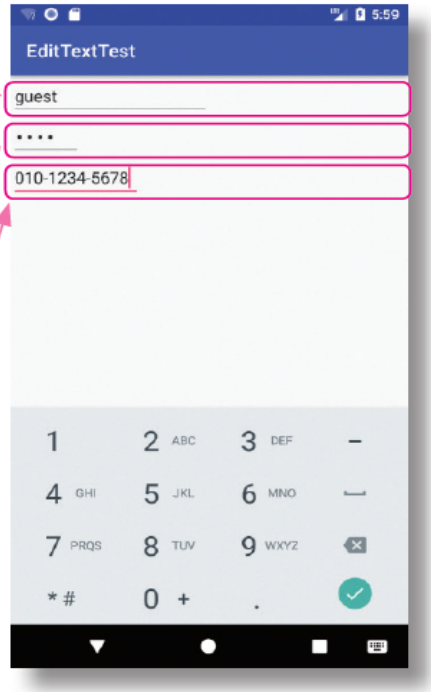




android:inputType="textPassword"

EditText

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/edit1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="아이디"
        android:inputType="text" />

    <EditText
        android:id="@+id/edit2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="패스워드"
        android:inputType="numberPassword" />

    <EditText
        android:id="@+id/edit3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="010-XXXX-XXXX"
        android:inputType="phone" />

</LinearLayout>
```



전화번호만 입력 가능

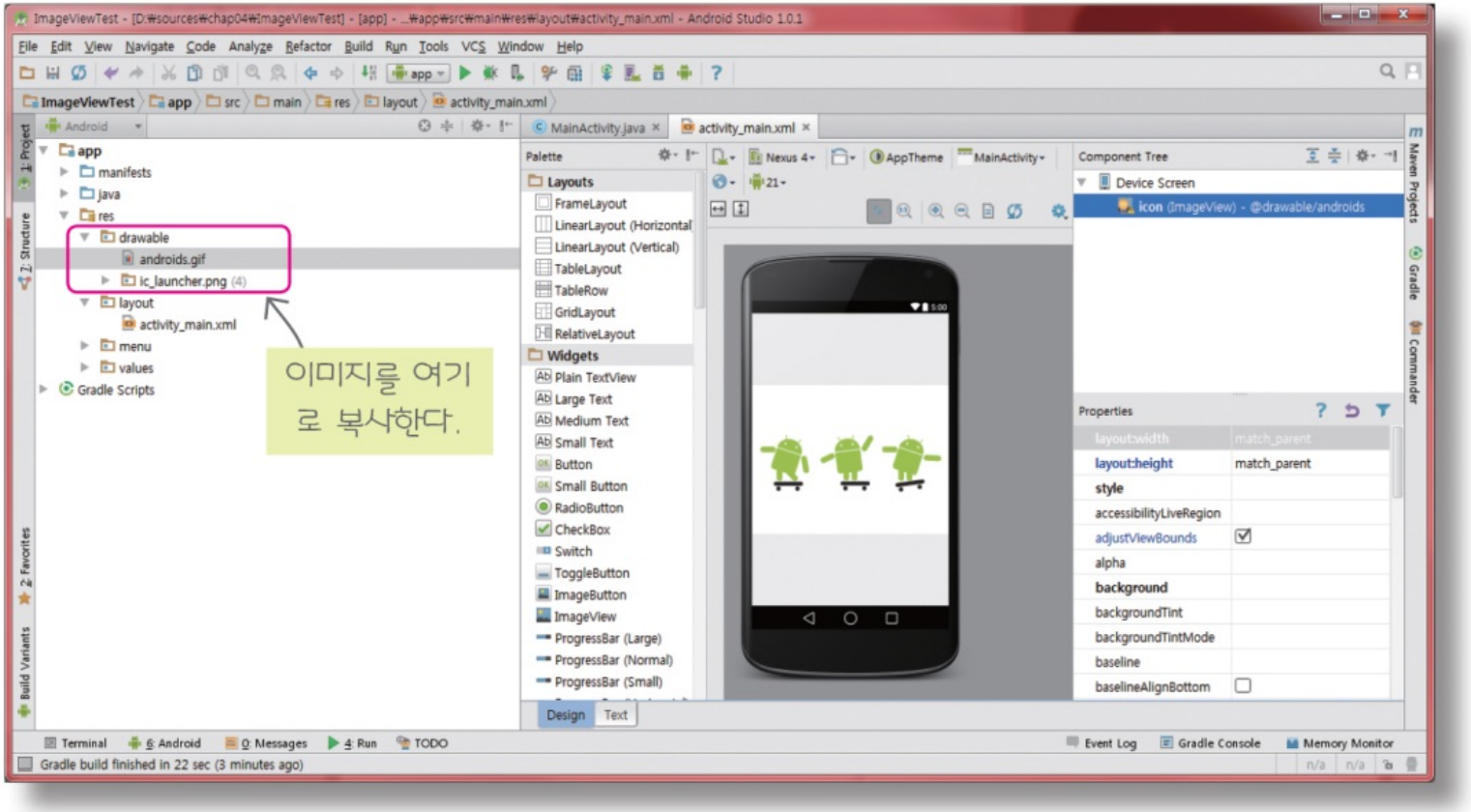
ImageView

- 아이콘과 같은 이미지들을 간단히 표시하는 데 사용

속성	설정 메소드	설명
android:adjustViewBounds	setAdjustViewBounds(boolean)	drawable의 종횡비를 유지하기 위하여 이미지 뷰의 가로, 세로를 조정
android:cropToPadding		true이면 패딩 안에 맞추어서 이미지를 자른다.
android:maxHeight	setMaxHeight(int)	이미지 뷰의 최대 높이
android:maxWidth	setMaxWidth(int)	이미지 뷰의 최대 너비
android:scaleType	setScaleType(ImageView.ScaleType)	이미지 뷰의 크기에 맞추어 어떻게 확대나 축소할 것인지 방법 선택
android:src	setImageResource(int)	이미지 소스
android:tint	setColorFilter(int, PorterDuff.Mode)	이미지 배경 색상

안드로이드에서 이미지 사용

- 이미지를 drawable 폴더로 복사한다.(Ctrl+C, Ctrl+V)



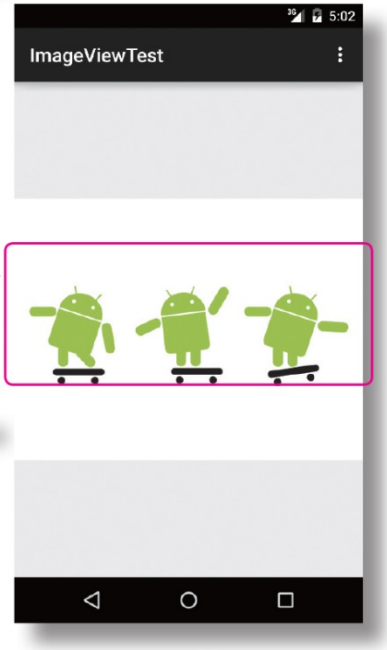
ImageView

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ImageView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/icon"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:adjustViewBounds="true"
    android:src="@drawable/androids"
/>
```

이것이 이미지 파일 이름이다.

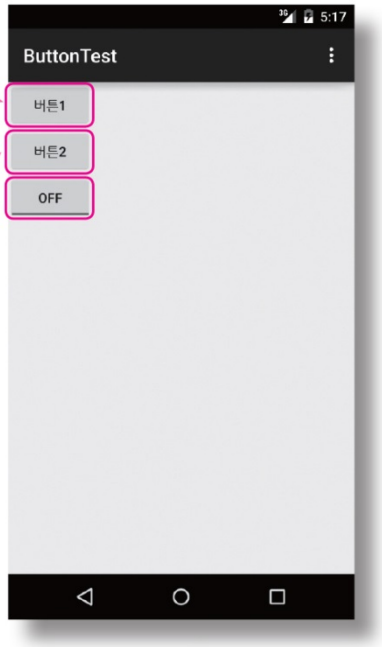
src 속성이 이미지 파일 이름을 가지고 있다.



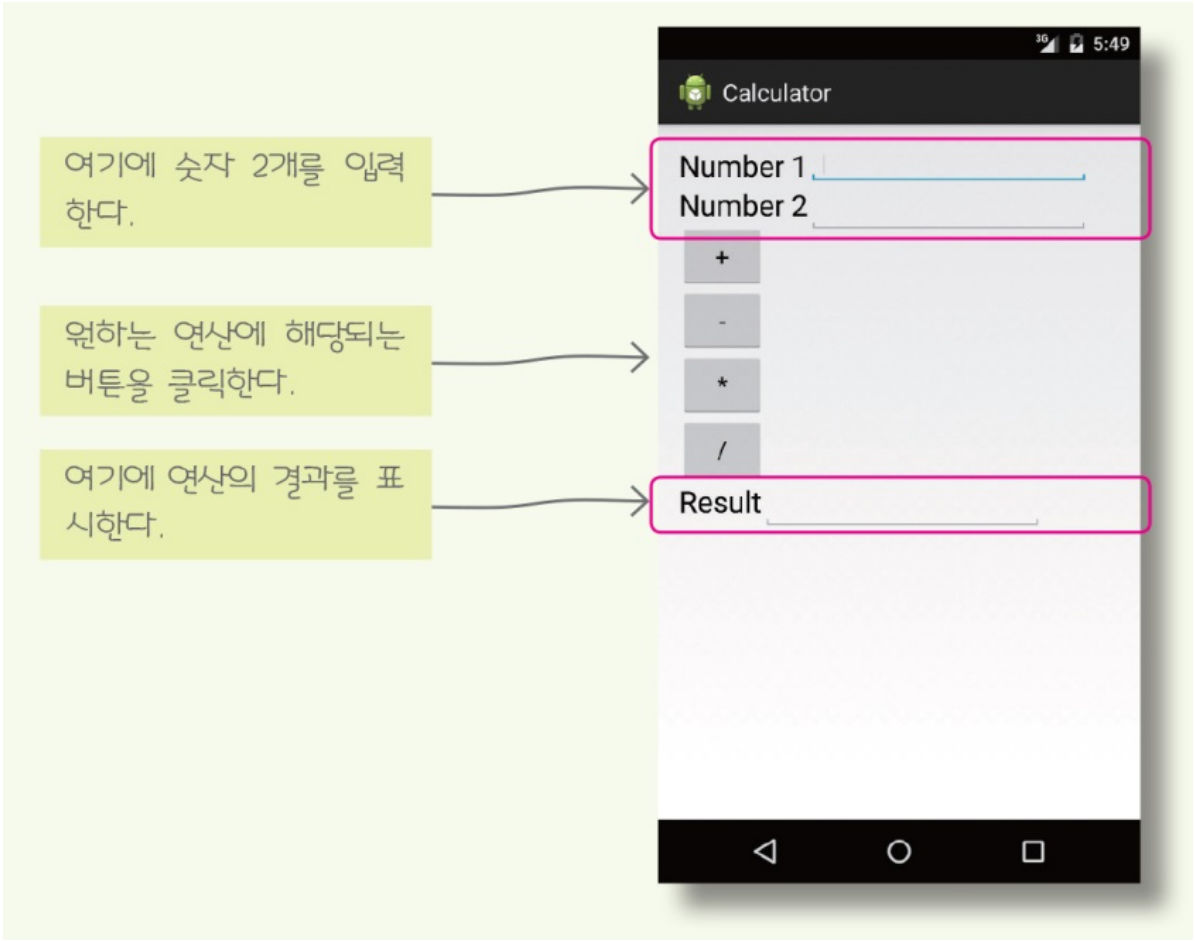
Button

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼2" />

    <ToggleButton
        android:id="@+id/button_toggle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="토글 버튼" />

</LinearLayout>
```



Lab: 계산기 앱 작성


여기에 숫자 2개를 입력한다.
원하는 연산에 해당되는 버튼을 클릭한다.
여기에 연산의 결과를 표시한다.
5:49
Calculator
Number 1
Number 2
+
-
*
/
Result