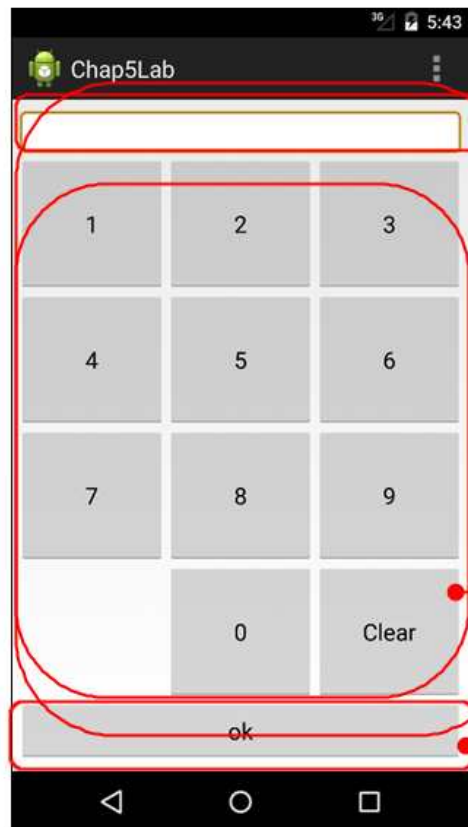


CHAP 3. 사용자 인터페이스 기초

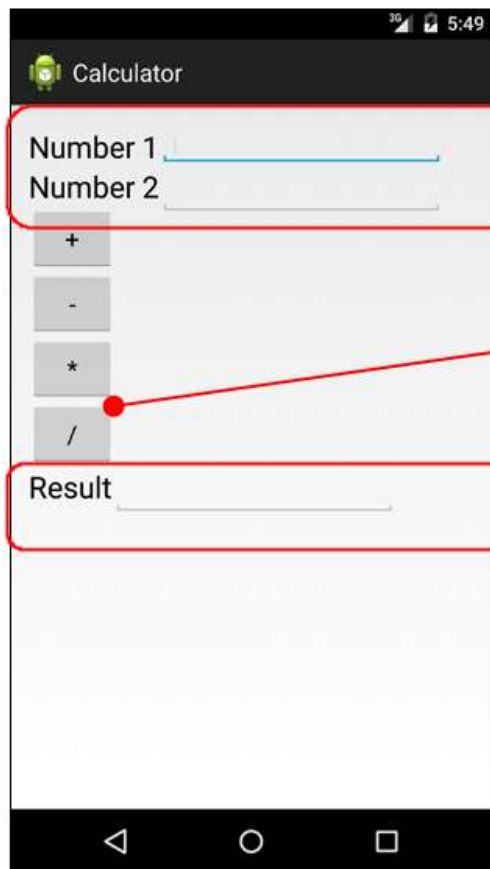


에디트 텍스트를 생성한다.

전체는 상대적 레이아웃

TableLayout으로 버튼들을 정렬한다.

버튼을 `android:layout_alignParentBottom="true"`로 하여 아래에 붙인다.



여기에 숫자 2개를 입력한다.

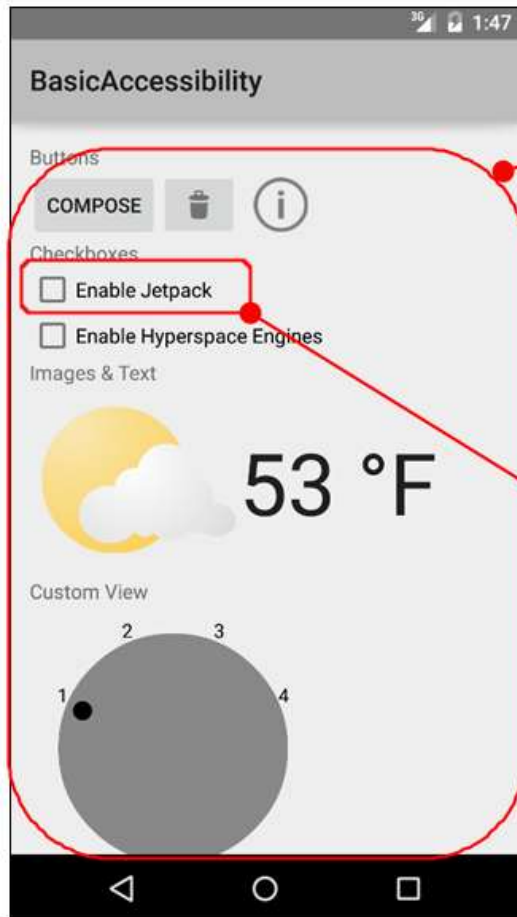
원하는 연산에 해당하는 버튼을 클릭한다.

여기에 연산의 결과를 표시한다.

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

뷰와 뷰그룹

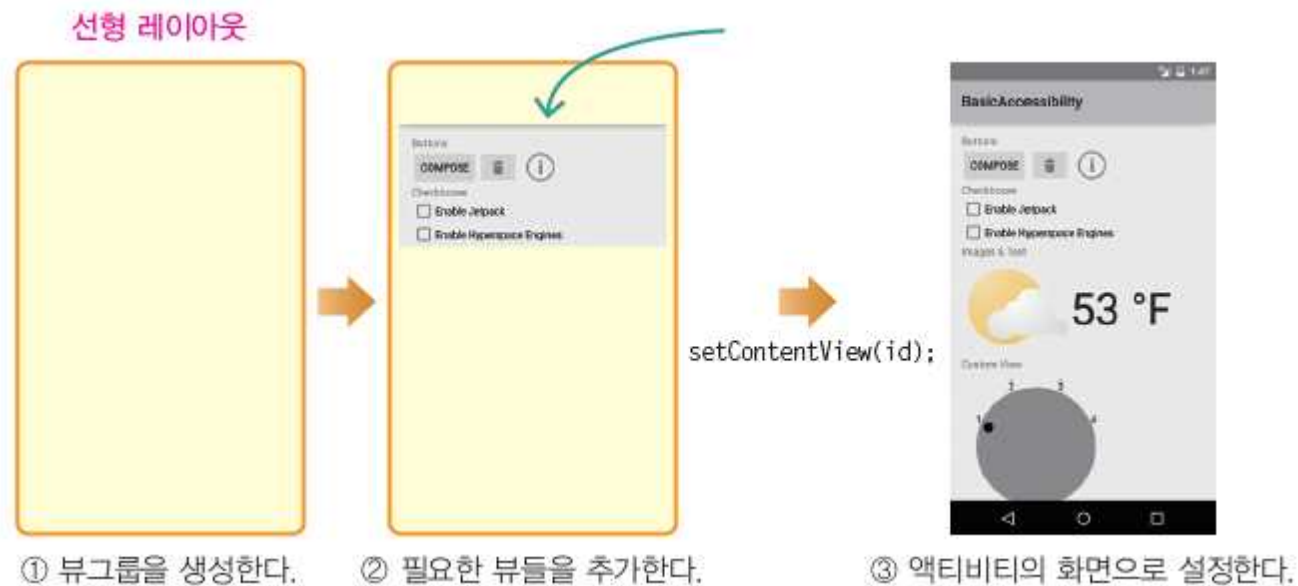


뷰그룹: 다른 뷰들을 담는 컨테이너 기능을 한다. 뷰그룹은 ViewGroup 클래스에서 상속받아서 작성된다. 흔히 레이아웃(layout)이라고 불리며 선형 레이아웃, 테이블 레이아웃, 상대적 레이아웃 등이 여기에 속한다. 각 레이아웃은 정해진 정책에 따라서 뷰들을 배치한다.

뷰: 컨트롤 또는 위젯이라고도 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.



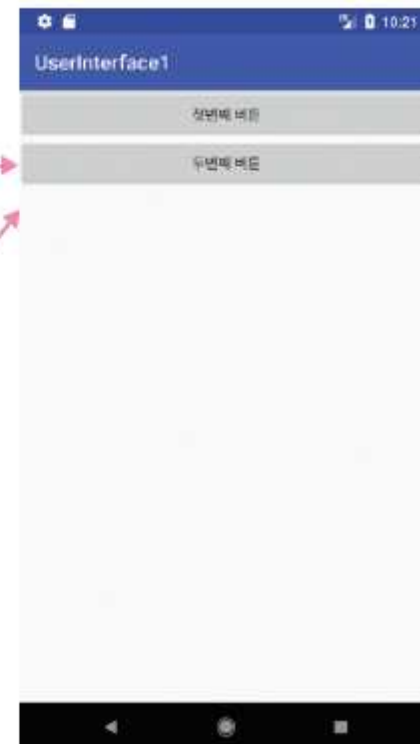
UI를 작성하는 2가지 방법

① XML로 사용자 인터페이스 기술

```
...  
<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

activity_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="match_parent"
        android:layout_height="wrap_content"
        android:text="첫번째 버튼" >
    </Button>

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

</LinearLayout>
```

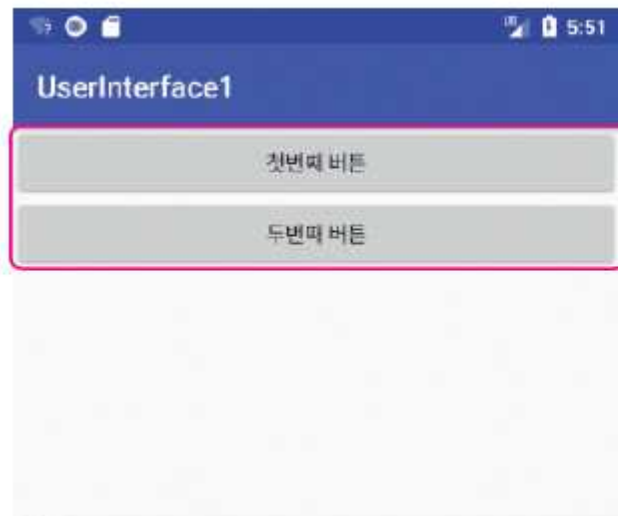
선형 레이아웃이
라는 뷰그룹을 생
성한다.

버튼이라는 뷰
를 생성한다.

자바 코드는 변경하지 않는다!

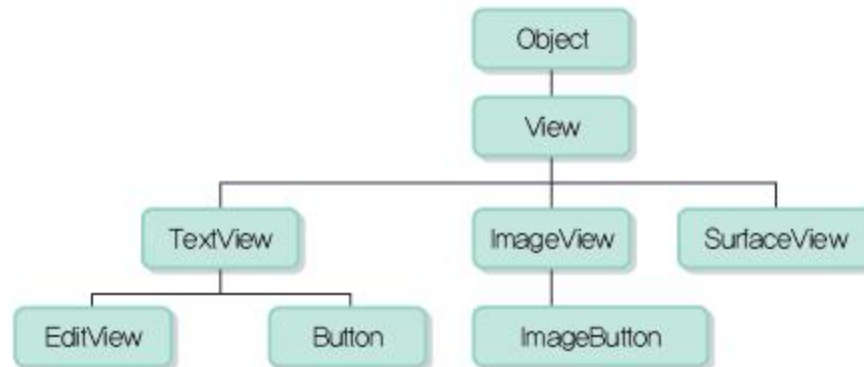
- 자동 생성된 코드 사용!

실행 결과



버튼만 2개 표시된다.

- **View** 클래스는 모든 뷰들의 부모 클래스이다.
- **View** 클래스가 가지고 있는 필드나 메소드는 모든 뷰에서 공통적으로 사용할 수 있다.



뷰의 필드와 메소드

- id
 - ▣ 뷰의 식별자
- 뷰의 위치와 크기

상수	설명
<code>match_parent</code>	부모의 크기를 꽉 채운다(fill_parent도 같은 의미).
<code>wrap_content</code>	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="@string/button_text" />`

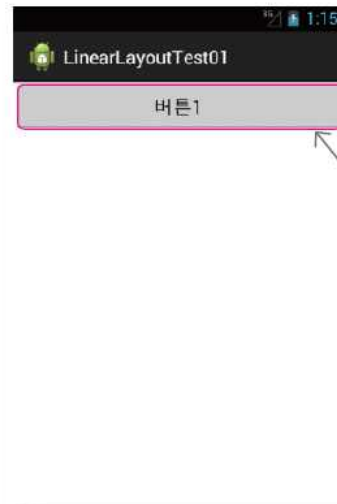
버튼의 위치와 크기를
결정한다.

layout_width와 layout_height가 모두 wrap_content이므로 가로 세로 길이가 모두 내용물에 맞추어진다.



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

layout_width가 match_parent이므로 가로 방향으로 부모공간을 모두 차지한다.



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

layout_height가 match_parent이므로 세로로 부모 공간을 모두 차지한다.



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

모두 match_parent이므로 가로와 세로로 부모 공간을 모두 차지한다.



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

뷰의 크기 단위

단위	설명
px(pixels)	화면의 실제 픽셀을 나타낸다. 픽셀은 권장되는 단위는 아님에 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

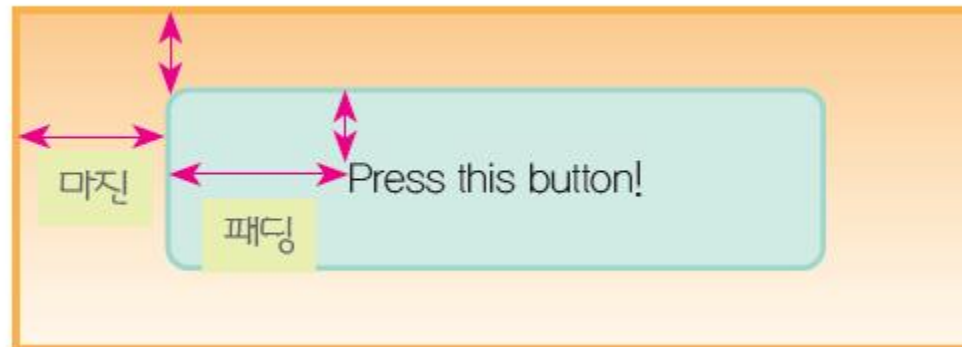
표시 방법	설명
#RRGGBB	RR은 빨간색 성분, GG는 녹색 성분, BB는 청색 성분을 나타낸다.
#AARRGGBB	AA는 투명도, RR은 빨간색 성분, GG는 녹색 성분, BB는 청색 성분을 나타낸다.

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.

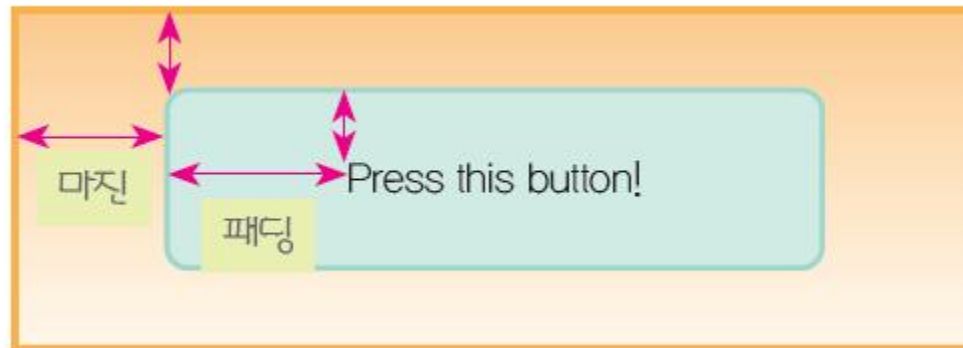
마진과 패딩

- 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



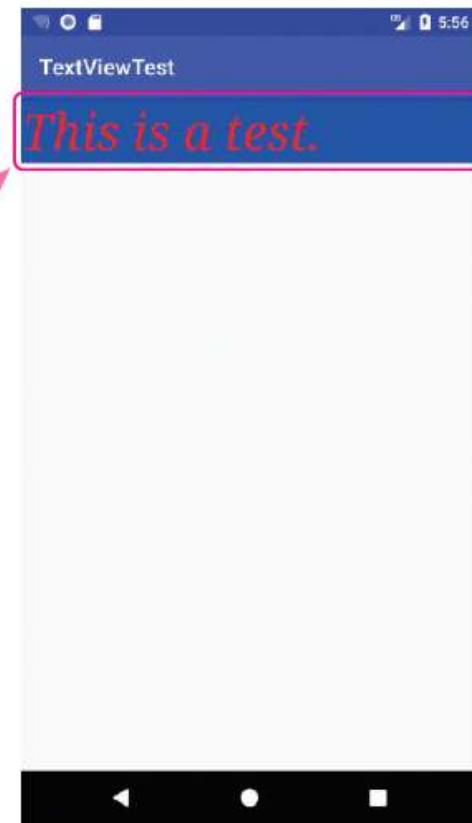
텍스트뷰

activity_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>
```



에디트 텍스트

Supercalifragilisticexpialidocious|

I'll be on my way then. see you
tomorrow|



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

에디트 텍스트의 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

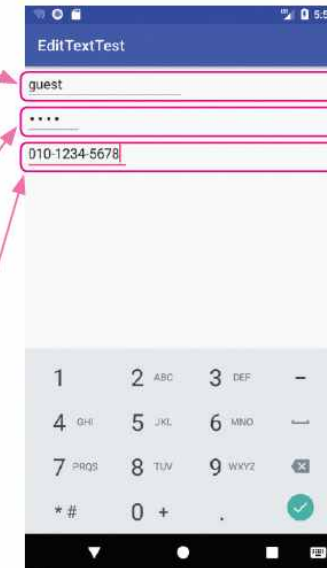
에디트 텍스트

```
<?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" />
```



전화번호만
입력 가능

```
</LinearLayout>
```

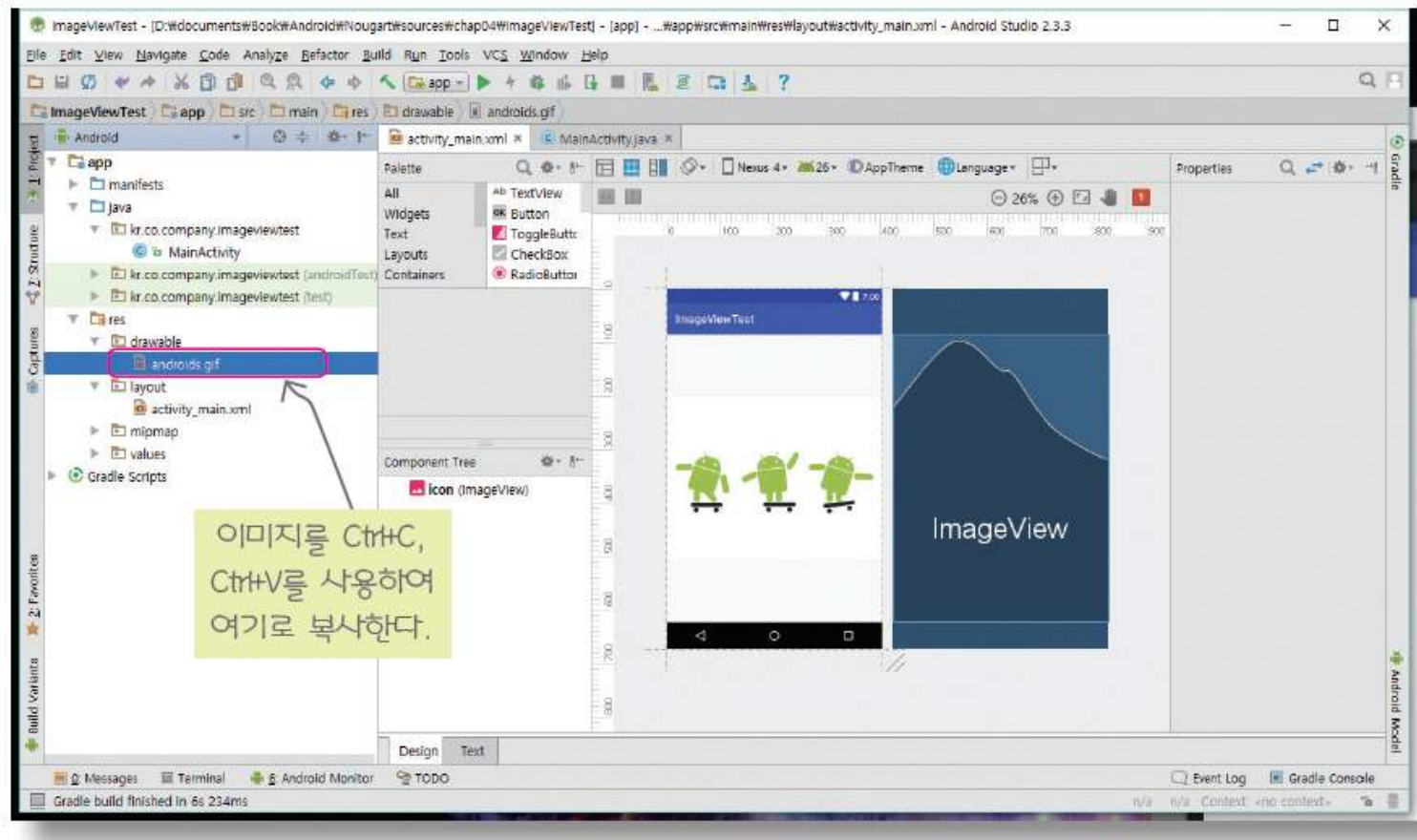
이미지뷰

- 아이콘과 같은 이미지들을 간단히 표시하는 데 사용

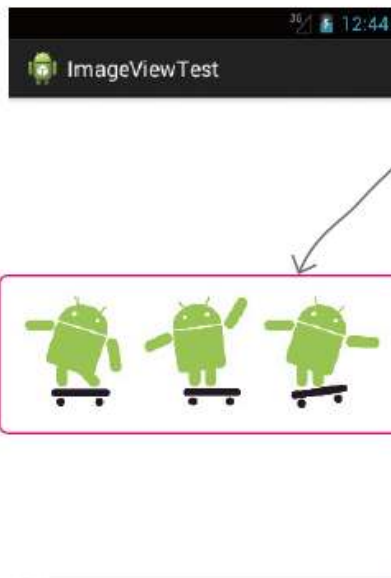
속성	설정 메소드	설명
<code>android:adjustViewBounds</code>	<code>setAdjustViewBounds(boolean)</code>	drawable의 종횡비를 유지하기 위하여 이미지 뷰의 가로, 세로를 조정
<code>android:cropToPadding</code>		true이면 패딩 안에 맞추어서 이미지를 자른다.
<code>android:maxHeight</code>	<code>setMaxHeight(int)</code>	이미지 뷰의 최대 높이
<code>android:maxLength</code>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

안드로이드에서 이미지 사용

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



이미지뷰



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"
/>
```

버튼

main.xml

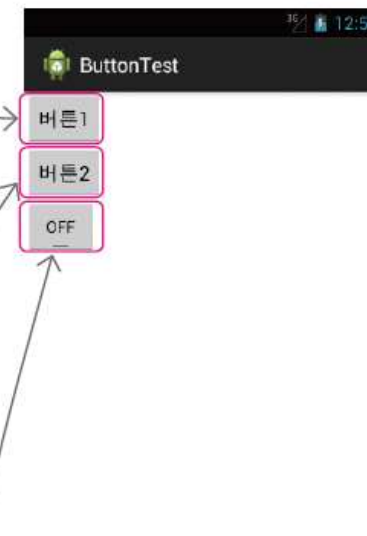
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
<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" />
```

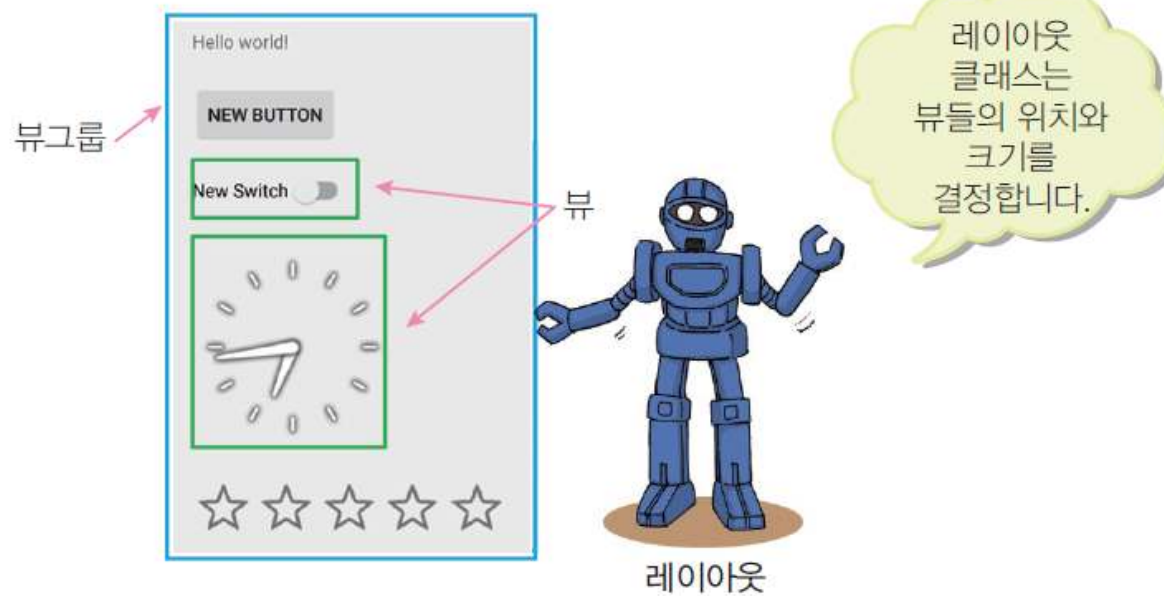
```
<ToggleButton
    android:id="@+id/button_toggle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="토글 버튼" />
```

```
</LinearLayout>
```

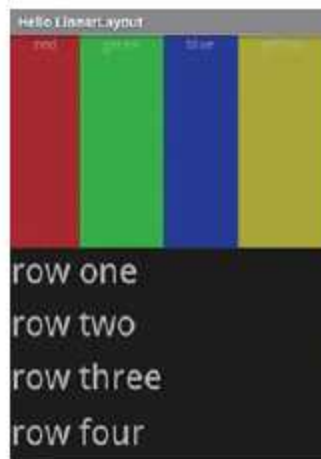


레이아웃

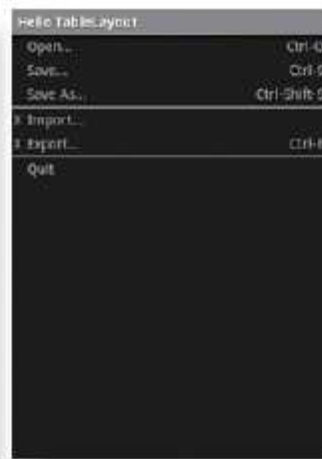
□ 뷰들을 화면에 배치하는 방법



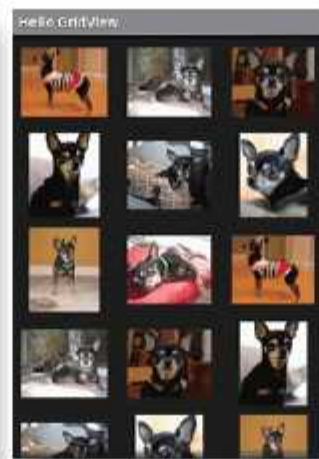
레이아웃의 종류



LinearLayout



TableLayout



GridLayout

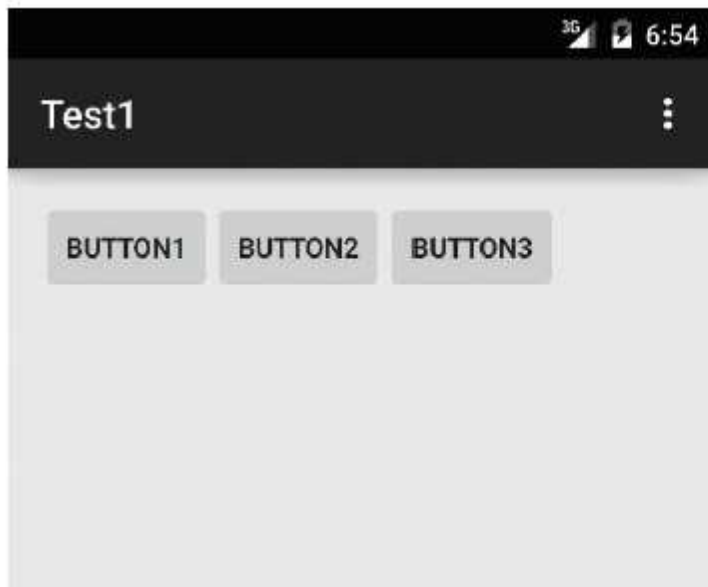


RelativeLayout

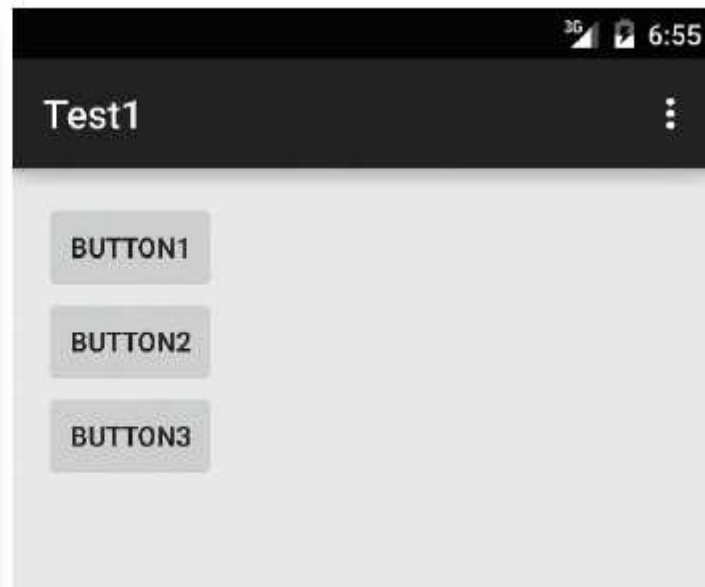


TabLayout

선형 레이아웃



(a) 수평 배치



(b) 수직 배치

선형 레이아웃 클래스의 속성

속성	관련 메소드	설명
orientation	setOrientation(int)	"horizontal"은 수평으로, "vertical"은 수직으로 배치한다.
gravity	setGravity(int)	x축과 y축 상에 자식을 어떻게 배치할 것인지를 지정한다.
baselineAligned	setBaselineAligned (boolean)	false로 설정되면 자식뷰들의 기준선을 정렬하지 않는다.

선형 레이아웃

activity_main.xml



사용자
인터페이스 작성

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="horizontal" ←
```

자식을 수평으로 배치

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">
```

```
    <Button
```

```
        android:id="@+id/button01"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="버튼 1"/>
```

```
    <Button
```

```
        android:id="@+id/button02"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="버튼 2"/>
```

```
    <Button
```

```
        android:id="@+id/button03"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="버튼 3"/>
```

```
</LinearLayout>
```



Gravity 속성 값

상수	값	설명
top	0x30	객체를 컨테이너의 상단에 배치, 크기를 변경하지 않음
bottom	0x50	객체를 컨테이너의 하단에 배치, 크기를 변경하지 않음
left	0x03	객체를 컨테이너의 좌측에 배치, 크기를 변경하지 않음
right	0x05	객체를 컨테이너의 우측에 배치, 크기를 변경하지 않음
center_vertical	0x10	객체를 컨테이너의 수직의 중앙에 배치, 크기를 변경하지 않음
fill_vertical	0x70	객체를 컨테이너의 수직을 채우도록 배치
center_horizontal	0x01	객체를 컨테이너의 수평의 중앙에 배치, 크기를 변경하지 않음
fill_horizontal	0x07	객체를 컨테이너의 수평을 채우도록 배치
center	0x11	객체를 컨테이너의 수평, 수직의 중앙에 배치
fill	0x77	객체가 컨테이너를 가득 채우도록 배치

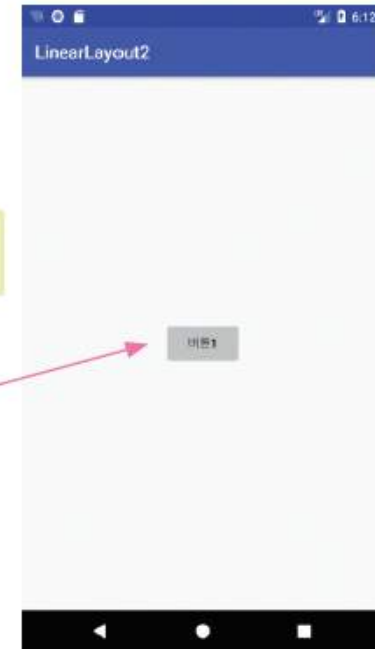
Gravity 설정

activity_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"
    android:gravity="center"
>

    <Button
        android:id="@+id/button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼1"
    />
</LinearLayout>
```

"자식뷰를 중앙에 배치할 것"이라는 의미이다.



가중치(weight)

- 선형 레이아웃의 자식 뷰들의 가중치가 각각 1, 2, 3이면, 남아있는 공간의 $1/6$, $2/6$, $3/6$ 을 각각 할당받는다.



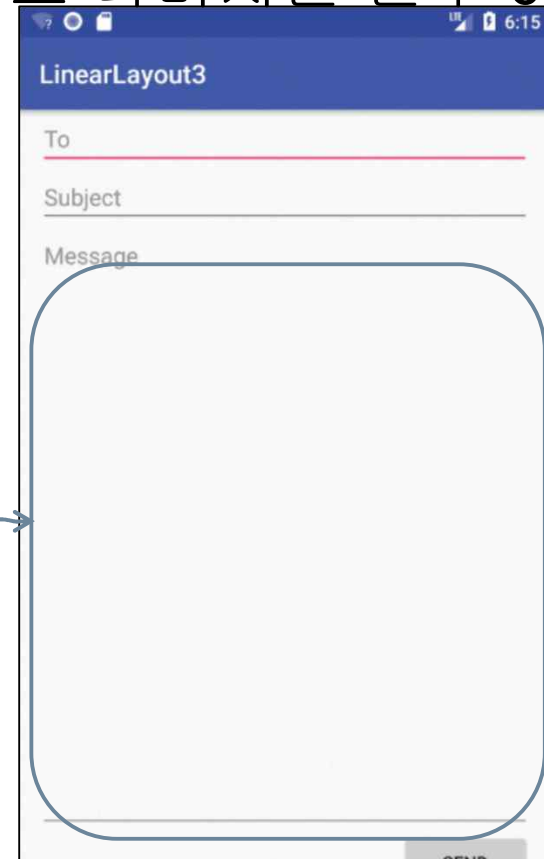
가중치(weight)

- 가중치를 1로 선언한 2개의 텍스트 뷰들은 남아있는 공간을 동일하게 차지할 것이다.



가중치 예제

- 버튼, 텍스트 뷰, 에디트 텍스트 등의 뷰들을 가중치를 다르게 하여 배치한 예
- 에디트 텍스트만 가중치가 1이고 나머지는 전부 0



activity_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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
```

가중치가 디폴트로 이어 된다.

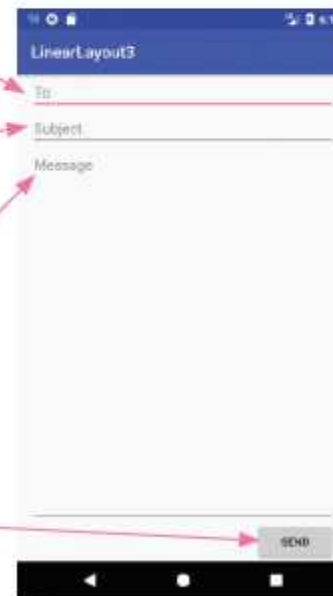
```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="To" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Subject" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="top"
    android:hint="Message" />
```

가중치를 1로 설정

```
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="Send" />
```



```
</LinearLayout>
```

테이블 레이아웃

activity_main.xml

```
<TableLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent" >
```

```
    <TableRow>
```

```
        <TextView android:text="주소" />
```

```
        <EditText android:text="서울시 종로구 120" />
```

```
    </TableRow>
```

```
    <TableRow>
```

```
        <TextView android:text="이름" />
```

```
        <EditText android:text="홍길동" />
```

```
    </TableRow>
```

```
    <TableRow>
```

```
        <Button android:text="저장" />
```

```
        <Button android:text="취소" />
```

```
    </TableRow>
```



상대적 레이아웃

속성	설명
<code>layout_above</code>	만약 true이면 현재 뷰의 하단을 기준 뷰의 위에 일치시킨다.
<code>layout_below</code>	현재 뷰의 상단을 기준 뷰의 하단에 위치시킨다.
<code>layout_centerHorizontal</code>	수평으로 현재 뷰의 중심을 부모와 일치시킨다.
<code>layout_centerInParent</code>	부모의 중심점에 현재 뷰를 위치시킨다.
<code>layout_centerVertical</code>	수직으로 현재 뷰의 중심을 부모와 일치시킨다.
<code>layout_toLeftOf</code>	현재 뷰의 우측단을 기준 뷰의 좌측단에 위치시킨다.
<code>layout_toRightOf</code>	현재 뷰의 좌측단을 기준 뷰의 우측단에 위치시킨다.

상대적



activity_main.xml

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

```
<TextView
    android:id="@+id/address"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:text="주소를 입력하세요" />
```

```
<EditText
    android:id="@+id/input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background"
    android:layout_below="@id/address" />
```

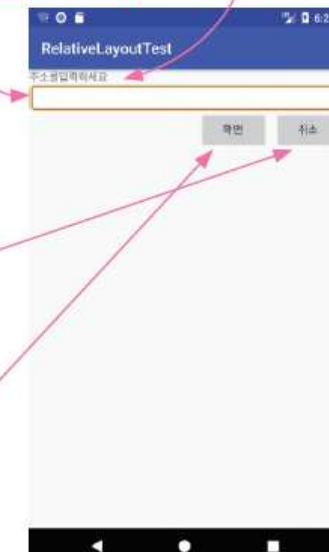
```
<Button
    android:id="@+id/cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/input"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="취소" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/cancel"
    android:layout_alignTop="@id/cancel"
    android:text="확인" />
```

address
아래에 배치

input
아래에 배치

cancel의
왼쪽에 배치



코드로 레이아웃 만들기



MainActivity.java

```
package.kr.co.company.userinterface2;
// 소스만 입력하고 Alt+Enter 키를 눌러서 import 문장을 자동으로 생성한다

public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout container = new LinearLayout(this);
        container.setOrientation(LinearLayout.VERTICAL);

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

        Button b2 = new Button(this);
        b2.setText("두번째 버튼");
        container.addView(b2);

        setContentView(container);
    }
}
```

선형 레이아웃을 생성한다.

버튼을 선형 레이아웃에 추가한다.

실행 결과



코드로 속성 변경하기

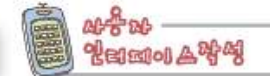
Nested Classes		
class	<code>LinearLayout.LayoutParams</code>	Per-child layout information associated with
XML Attributes		
Attribute Name	Related Method	Description
<code>android:baselineAligned</code>	<code>setBaselineAligned(boolean)</code>	When set to false, prevents the layout from aligning its children's baselines.
<code>android:baselineAlignedChildIndex</code>	<code>setBaselineAlignedChildIndex(int)</code>	When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which child TextView).
<code>android:divider</code>	<code>setDividerDrawable(Drawable)</code>	Drawable to use as a vertical divider between buttons.
<code>android:gravity</code>	<code>setGravity(int)</code>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
<code>android:measureWithLargestChild</code>	<code>setMeasureWithLargestChildEnabled(boolean)</code>	When set to true, all children with a weight will be considered having the minimum size of the largest child.
<code>android:orientation</code>	<code>setOrientation(int)</code>	Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column.
<code>android:weightSum</code>		Defines the maximum weight sum.
Inherited XML Attributes		
▶ From class <code>android.view.ViewGroup</code>		
▶ From class <code>android.view.View</code>		

일반적인 경우, XML의 속성과 메소드는 대응된다.

그림 5. 선형 레이아웃 클래스

XML로 UI 정의

activity_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="첫 번째 버튼"
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
    </Button>
    <Button
        android:text="두 번째 버튼"
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
    </Button>
</LinearLayout>
```

버튼에 식별자를 부여한다.

코드로 속성 변경



MainActivity.java

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

public class MainActivity extends AppCompatActivity {

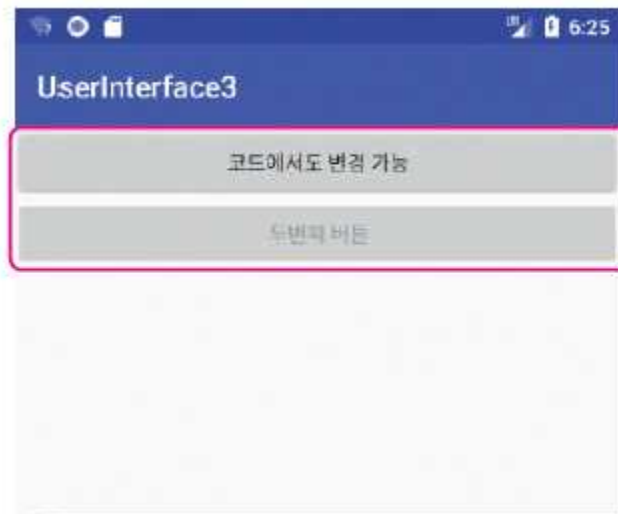
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        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.

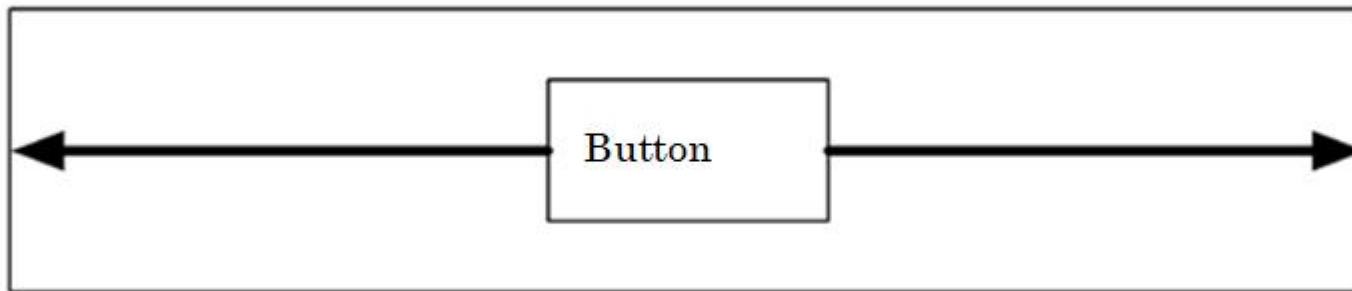
실행 화면



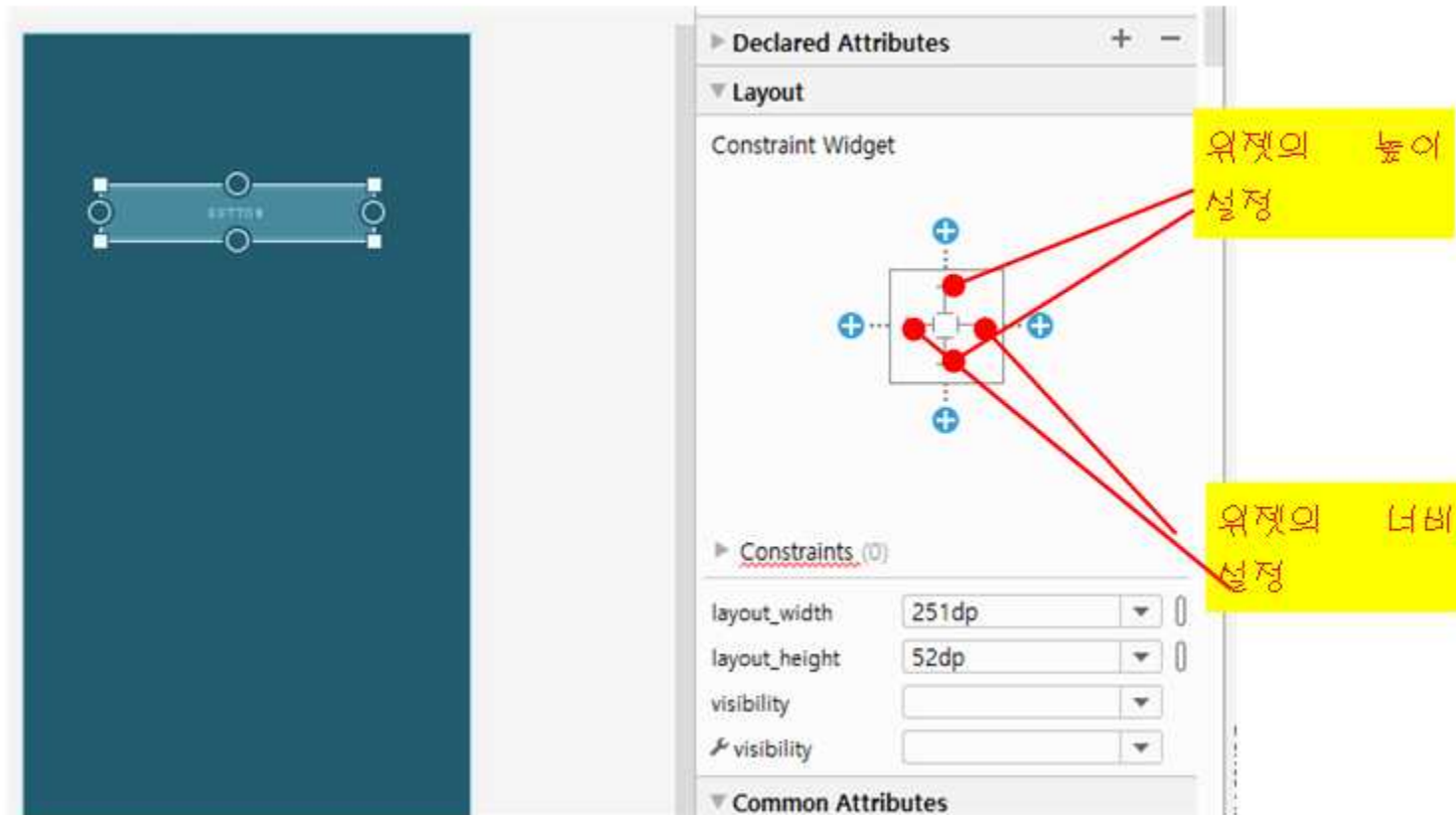
코드에 의하여 버튼
의 텍스트와 가시성
이 변경되었다.

비주얼 도구 사용하는 방법

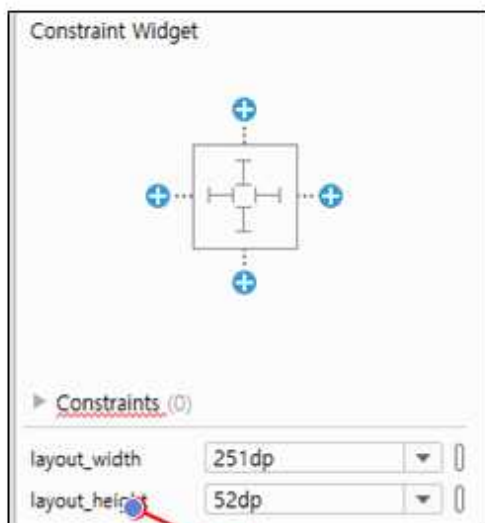
□ 구속조건



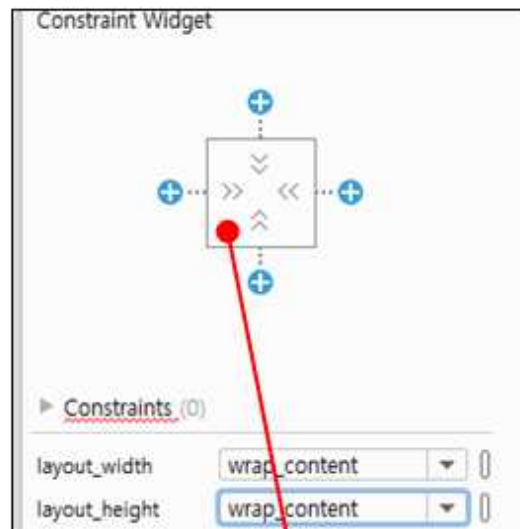
위젯의 크기 조정



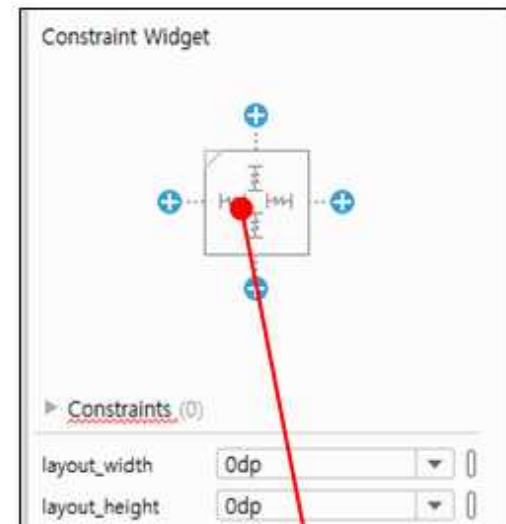
위젯의 크기 조정



고정된 크기

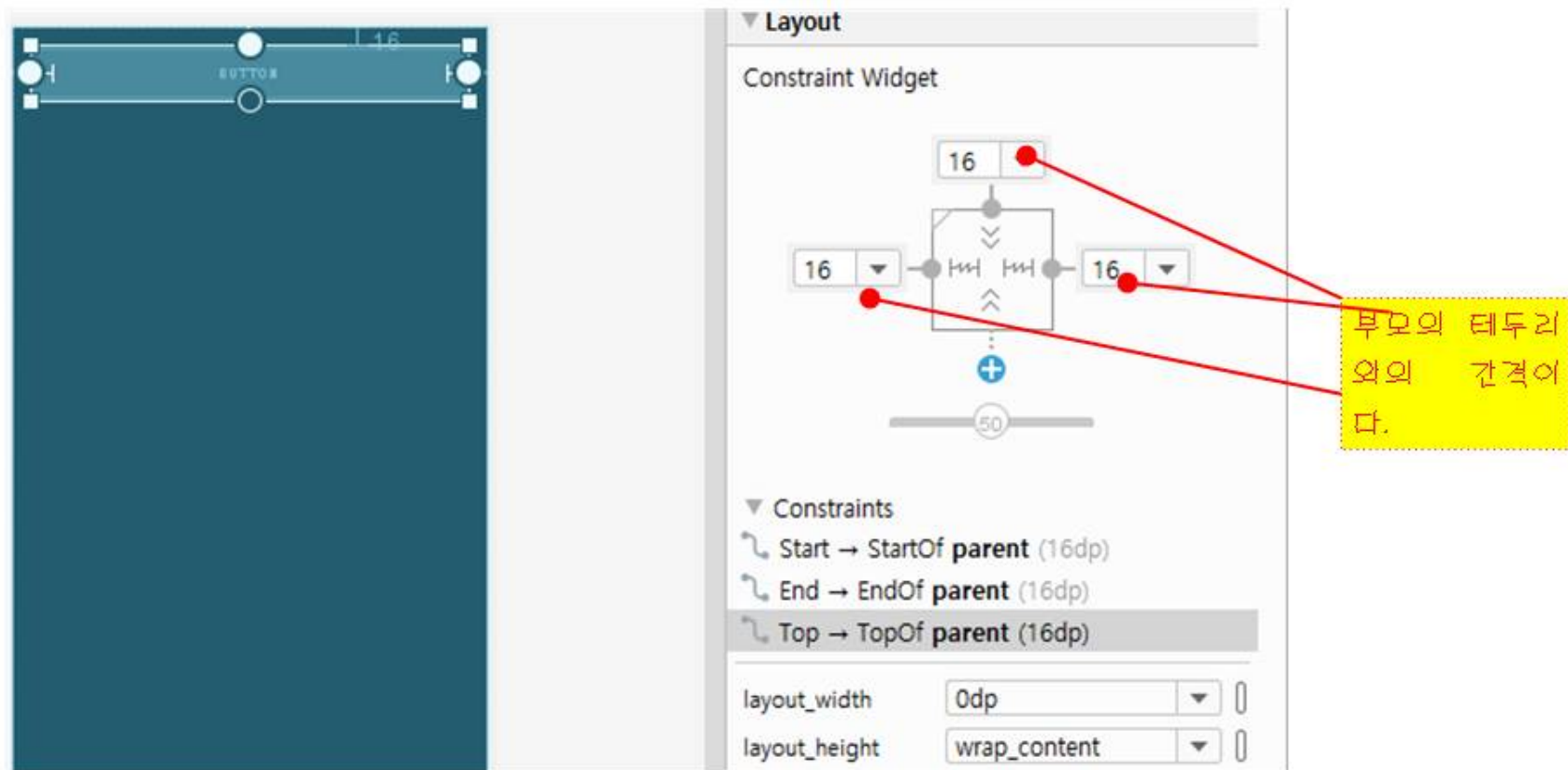


wrap_content

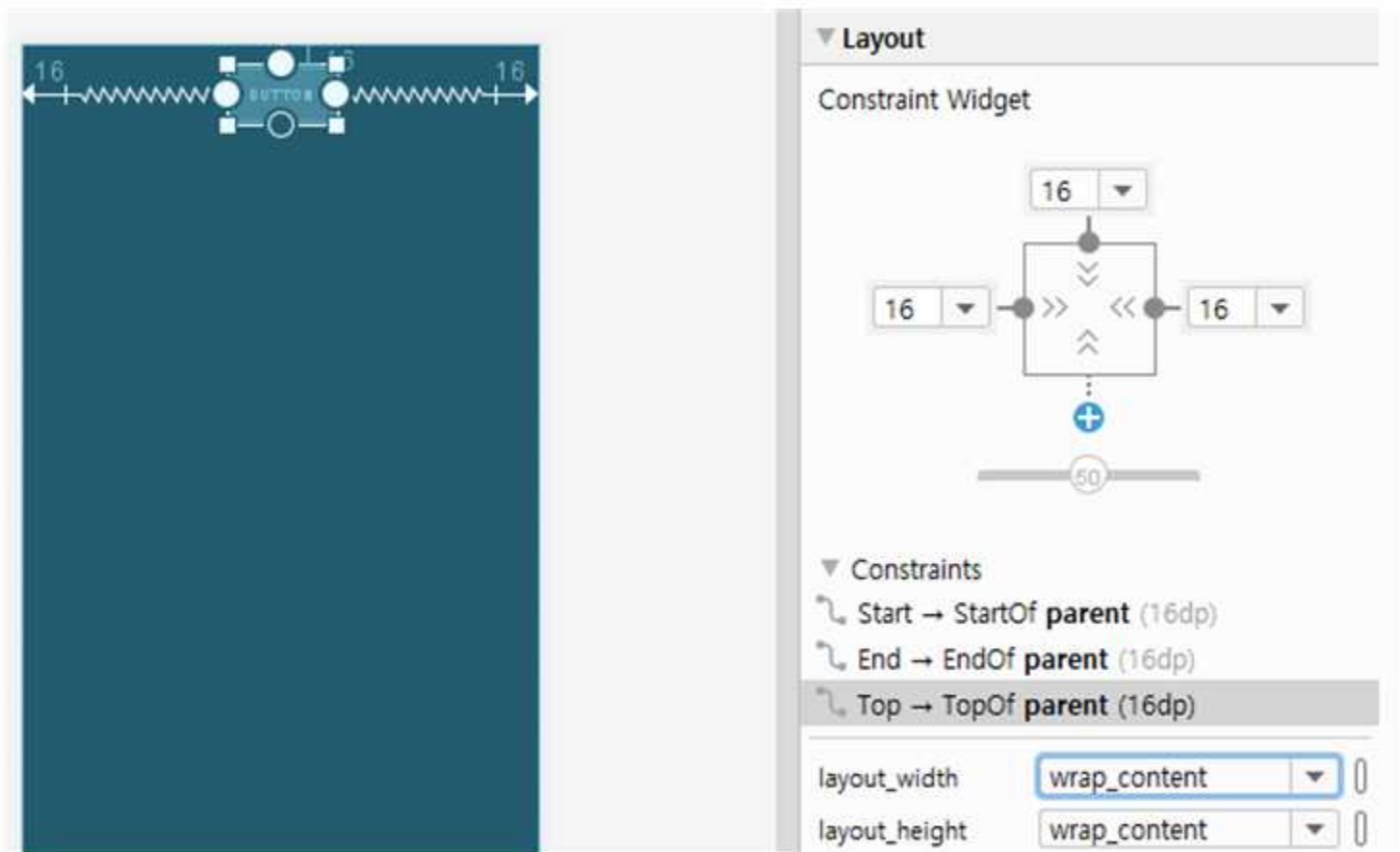


match_constraint

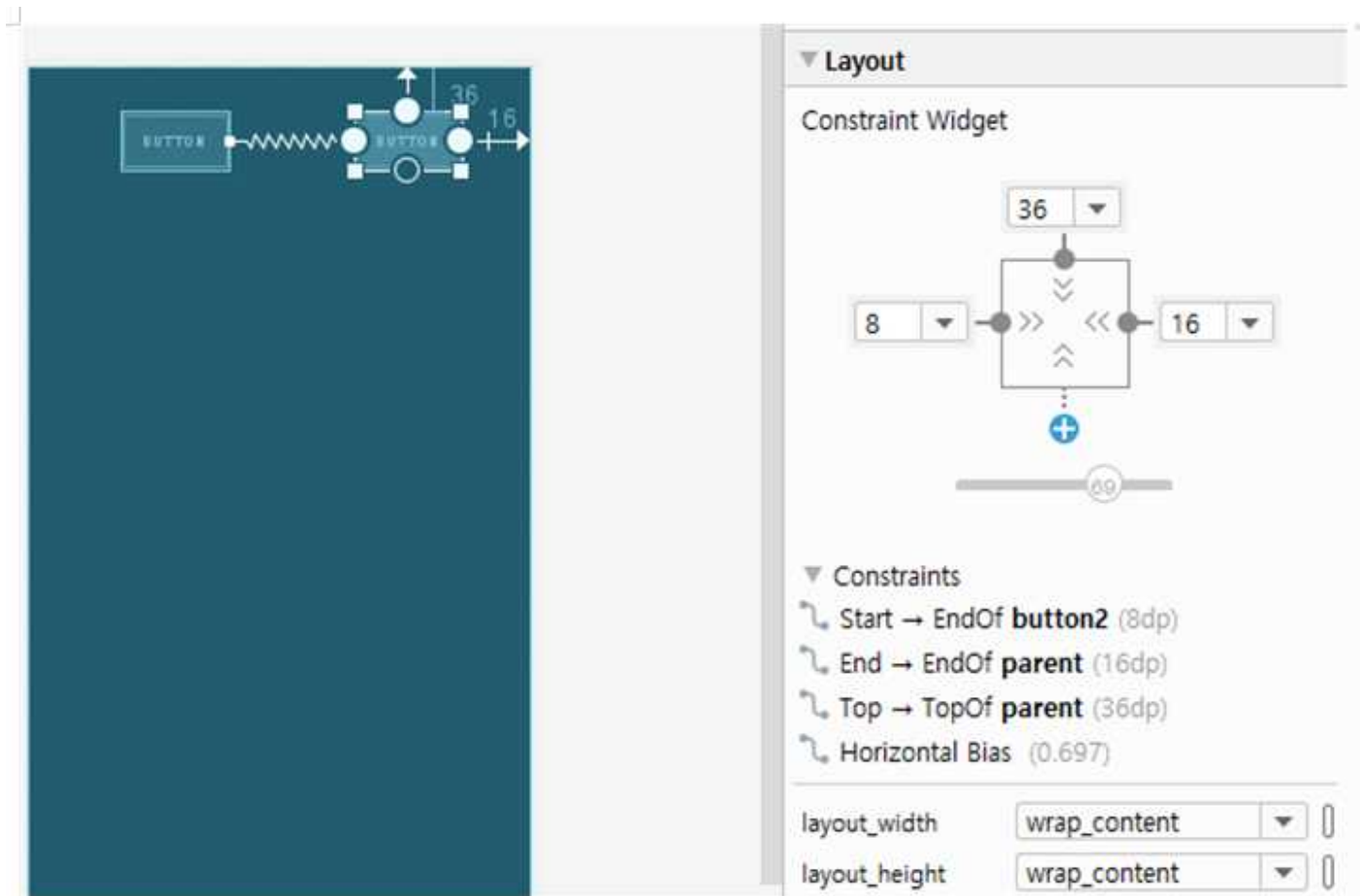
구속 조건 설정



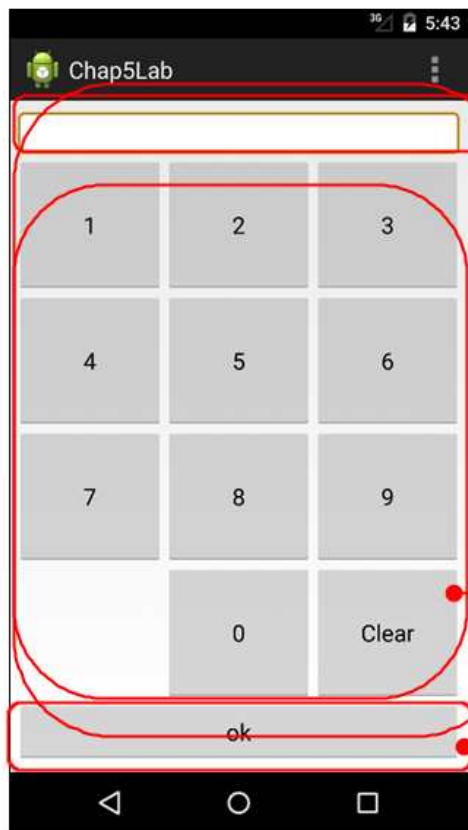
구속 조건 설정



다른 위젯에 붙일 때



Lab: 계산기 앱 작성



에디트 텍스트를 생성한다.

전체는 상대적 레이아웃

TableLayout으로 버튼들을 정렬한다.

버튼을 `android:layout_alignParentBottom="true"`로 하여 아래에 붙인다.

상단 구현

main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="7dip">
    <EditText android:id="@+id/number"
        android:background="@android:drawable/editbox_background"
        android:cursorVisible="false"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    ...
</RelativeLayout>
```



버튼 구현

main.xml

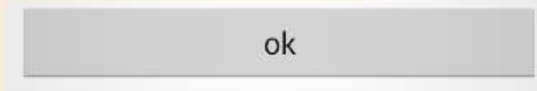
```
<TableLayout android:id="@+id/row1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/numberSeparator2"
    android:layout_above="@id/ok"
    android:layout_weight="1">
    <TableRow android:layout_weight="1">
        <Button android:id="@+id/n1"
            android:layout_width="0dip"
            android:layout_height="fill_parent"
            android:text="1"
            android:layout_weight="1" />
        <Button android:id="@+id/n2"
            android:layout_width="0dip"
            android:layout_height="fill_parent"
            android:text="2"
            android:layout_weight="1" />
        <Button android:id="@+id/n3"
            android:layout_width="0dip"
            android:layout_height="fill_parent"
            android:text="3"
            android:layout_weight="1" />
    </TableRow>
```



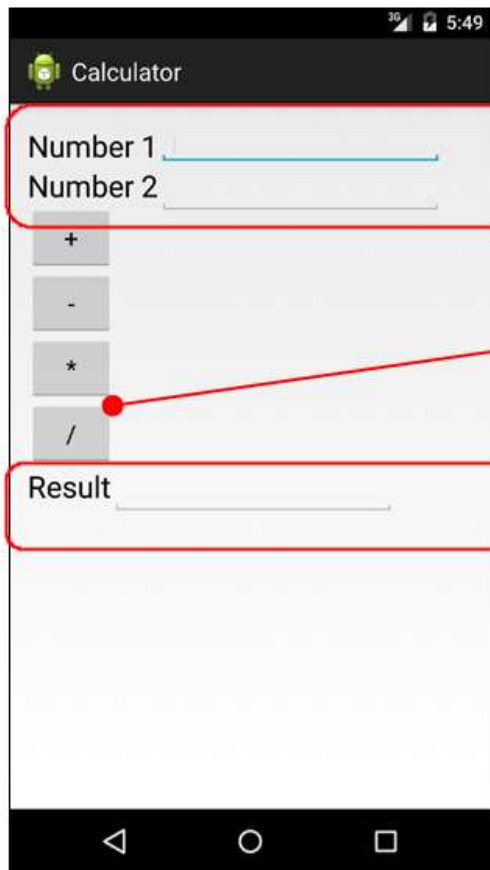
하단 구현

main.xml

```
<Button android:id="@+id/ok"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:text="ok"  
        android:layout_alignParentBottom="true" />
```



Lab: 계산기 앱 작성



여기에 숫자 2개를 입력한다.

원하는 연산에 해당하는 버튼을 클릭한다.

여기에 연산의 결과를 표시한다.