

2017 LINC+ 사업단 사회맞춤형 JAVA 실무인력양성과정 안드로이드 앱 개발 기초

컴퓨터공학부
강승우

2. 사용자 인터페이스

2018.02.02

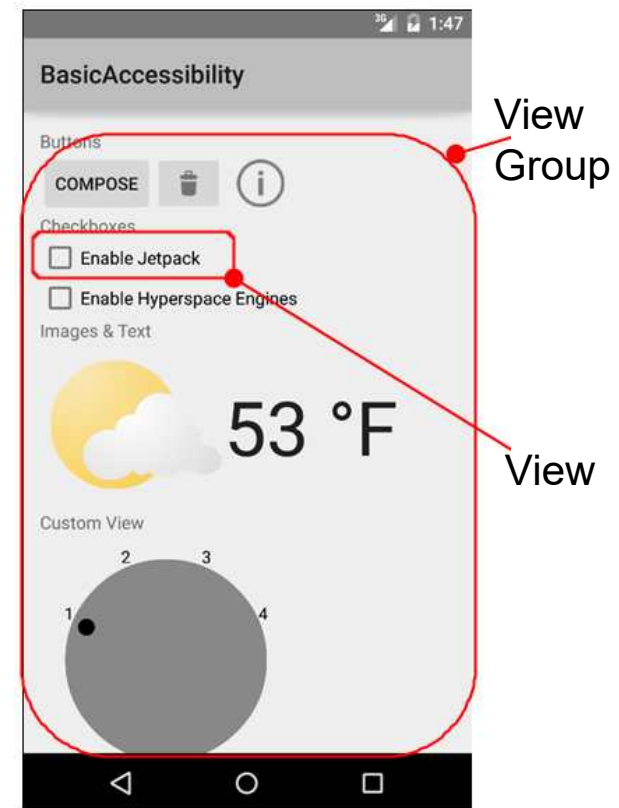
사용자 인터페이스 기초

- UI 개요
- 기본 뷰 (View)
- 레이아웃 (Layout)

UI 개요

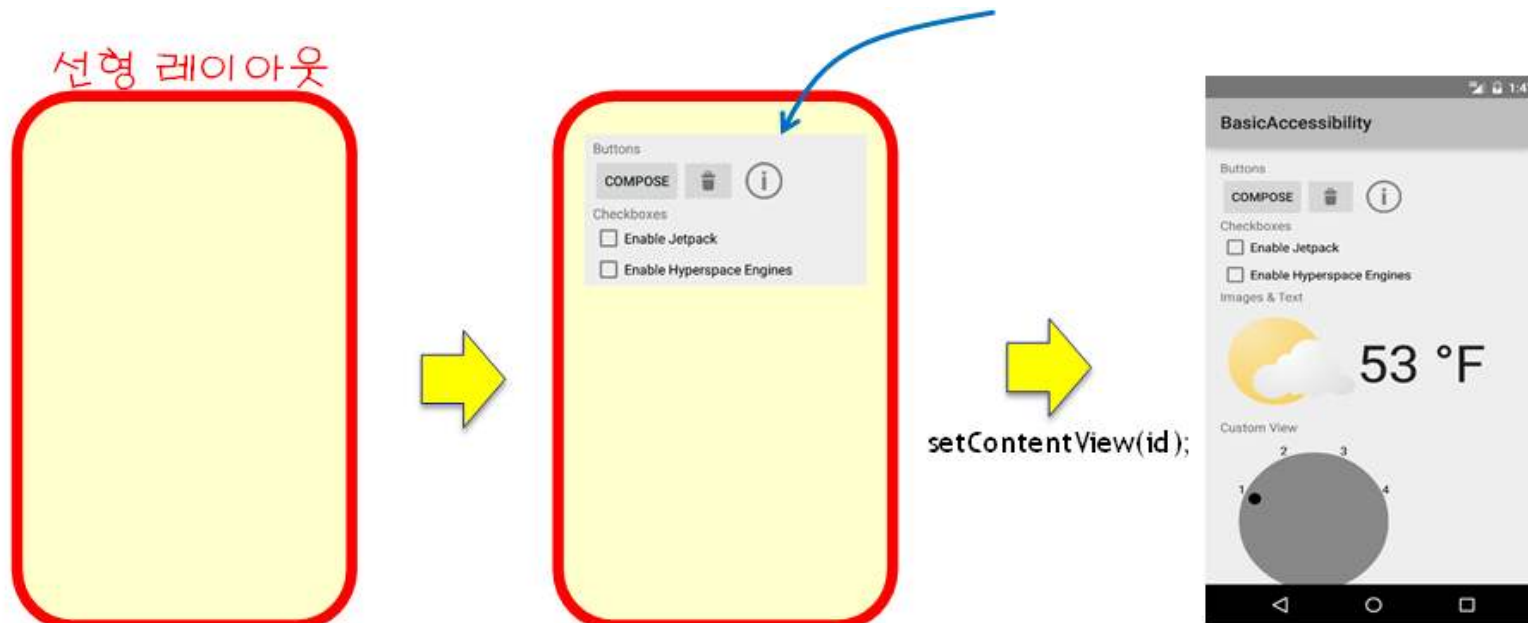
사용자 인터페이스 (UI)

- GUI (Graphical User Interface) 기반
- 안드로이드 SDK는 UI를 위한 다양한 요소를 제공
 - 버튼, 리스트, 스크롤 바, 체크 박스, 메뉴 등
- 뷰 (View)와 뷰 그룹 (View Group)
 - View
 - 안드로이드에서 UI를 구성하는 기본 building block
 - 컨트롤 또는 위젯이라고도 불림
 - 버튼, 텍스트 필드, 체크 박스 등
 - View 클래스를 상속받아서 작성됨
 - View Group
 - View를 담는 컨테이너, 흔히 레이아웃 (layout)이라고 불림
 - ViewGroup 클래스를 상속받아서 작성
 - linear layout, table layout, relative layout, constraint layout 등
 - 내부의 view를 배치하는 방식이 다름



UI 작성 절차

- View group 생성
- 필요한 view 추가
- Activity 화면으로 설정



UI 작성 방법

1. XML로 작성
2. 자바 코드로 작성
3. XML과 코드를 같이 사용하여 작성
4. Visual design 도구 이용

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

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  tools:context=".MainActivity"
  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
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="두번째 버튼" />

</LinearLayout>
```

예제 프로젝트 이름:
01_UI_Basic/Ch4UserInterface1



코드로 UI 작성

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

```
    @Override
```

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

```
    }  
}
```

예제 프로젝트 이름:
01_UI_Basic/Ch4UserInterface2



XML과 코드를 동시에 사용

<Button

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

<Button

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

@Override

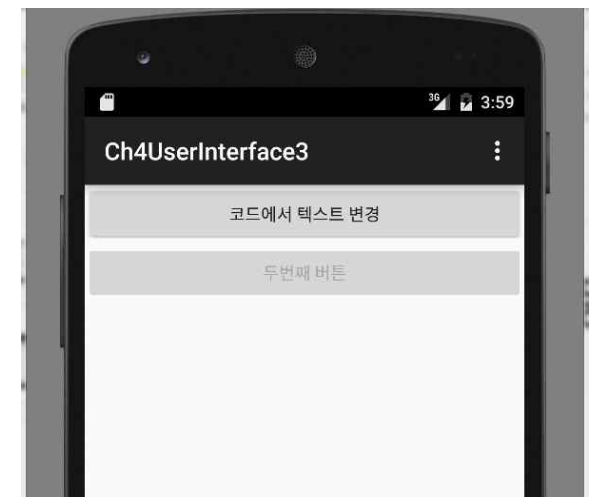
```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    Button b1 = (Button) findViewById(R.id.button1);  
    b1.setText("코드에서 텍스트 변경");
```

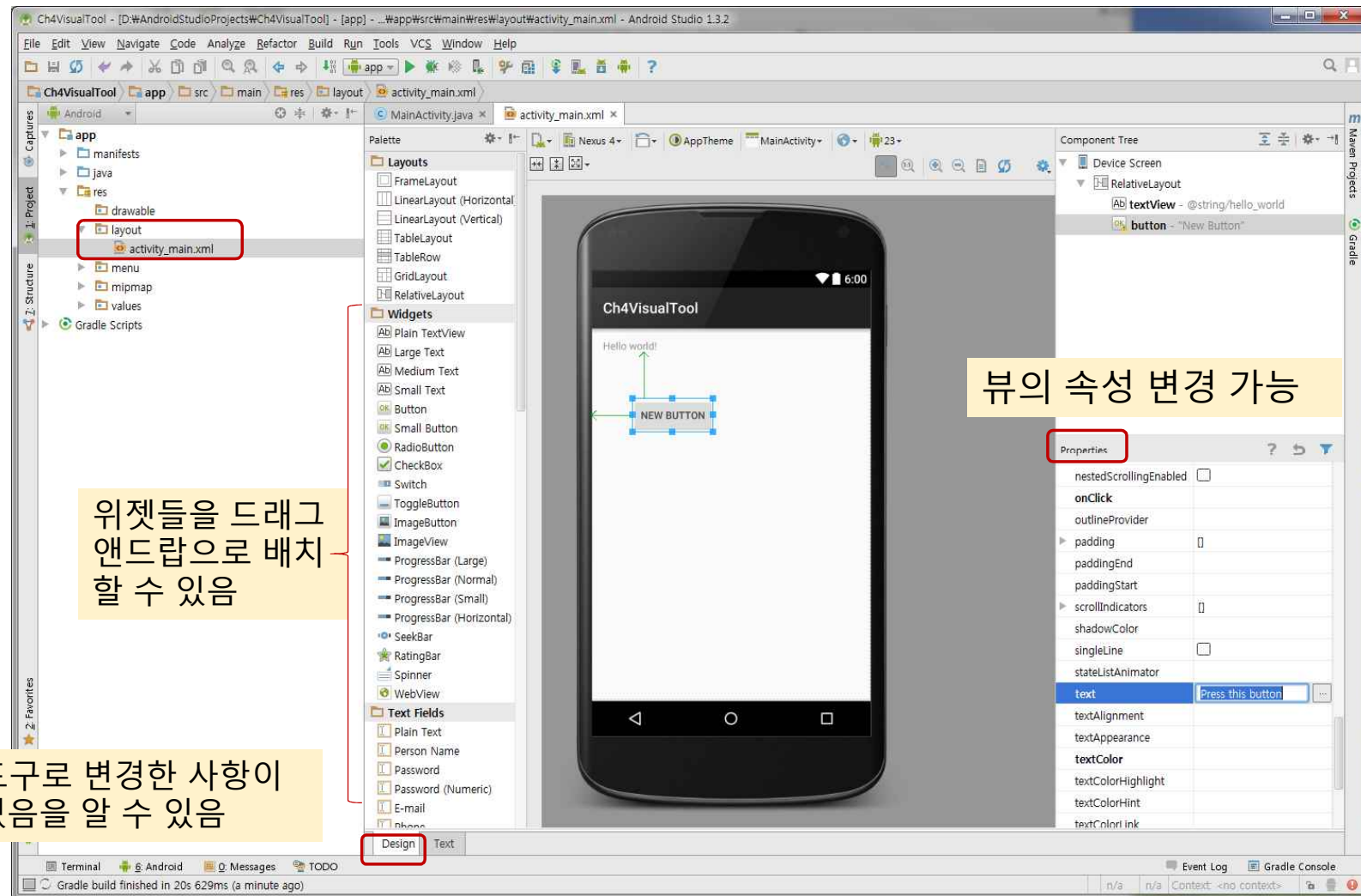
```
    Button b2 = (Button) findViewById(R.id.button2);  
    b2.setEnabled(false);
```

```
}
```

예제 프로젝트 이름:
01_UI_Basic/Ch4UserInterface3

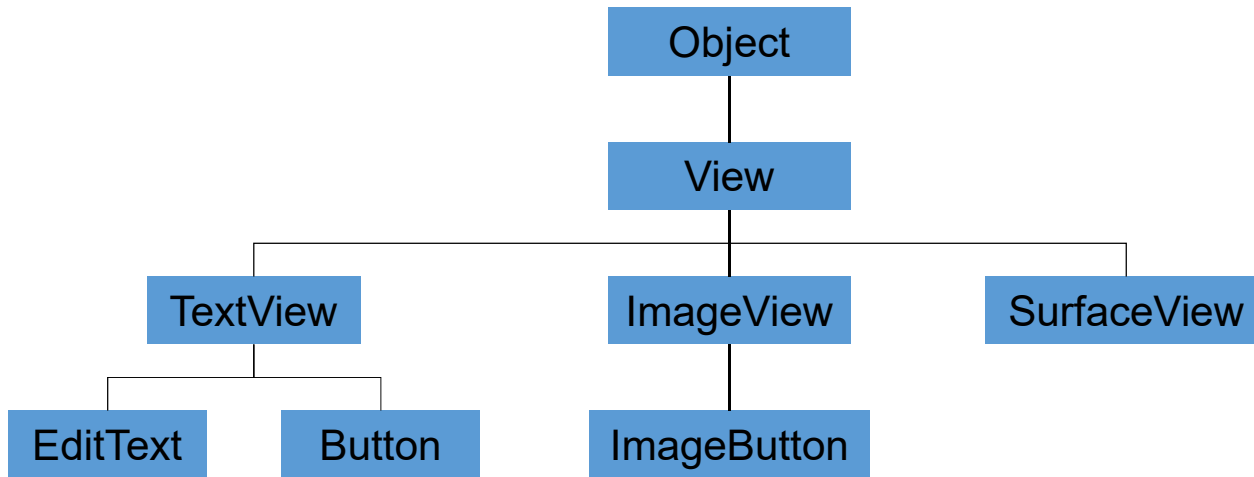


비주얼 도구 사용



뷰 속성

뷰 (View)



• View 클래스

- 모든 뷰들의 부모 클래스
- View 클래스의 필드나 메소드는 모든 뷰에서 공통적으로 사용 가능
- 관련 속성
 - id
 - size/location
 - color
 - visibility
 - margin/padding

뷰 속성

- id

<https://developer.android.com/reference/android/view/View.html>

- 뷰 식별자

android:id="@+id/my_button"

- XML의 각 속성 이름(attribute name)과 해당하는 자바 메소드 정보 참조

- 크기/위치

- android:layout_width / android:layout_height

<Button

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

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

예제 → android:layout_width="200px"

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.

크기 변경

- android:layout_width
- android:layout_height
- wrap_content
- match_parent

예제 프로젝트 이름:
01_UI_Basic/Ch4LayoutSizeTest



색상

- 뷰의 배경색 지정
 - android:background
 - 16 진수로 RGB와 투명도 설정
 - 투명도는 생략 가능

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.

android:background="#ff0000" : 빨간색
android:background="#00ff00" : 녹색
android:background="#ff000000": 완전 불투명 검정색
android:background="#ffffffff": 불투명 흰색

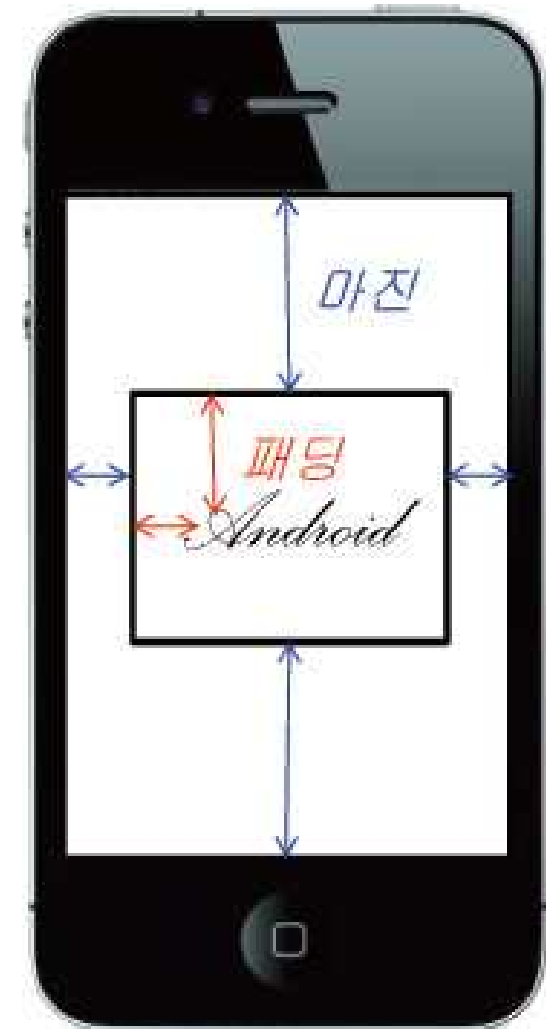
뷰의 화면 표시

- 뷰의 초기 가시성 제어
 - android:visibility

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

패딩과 마진

- 패딩 (padding)
 - 뷰의 경계와 뷰의 내용물 사이의 간격
- 마진 (margin)
 - 컨테이너와 뷰 사이의 간격



패딩과 마진

- 패딩

- `android:padding`
- `android:paddingLeft`, `android:paddingRight`,
`android:paddingTop`, `android:paddingBottom`

- 마진

- `android:margin`
- `android:marginLeft`, `android:marginRight`,
`android:marginTop`, `android:marginBottom`



패딩과 마진 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" />
```



기본 뷰

TextView

<http://developer.android.com/reference/android/widget/TextView.html>

- 화면에 텍스트를 표시
 - 편집 불가능 텍스트
 - 설명이나 제목 등에 사용
- 속성
 - android:text
 - 표시할 텍스트
 - android:textColor
 - 텍스트 색상
 - android:textSize
 - 텍스트 크기
 - android:textStyle
 - 텍스트 스타일 (bold, italic, normal)
 - android:typeface
 - 텍스트 폰트
 - (normal, sans, serif, monospace)
 - android:width / android:height
 - 픽셀 단위 텍스트 뷰의 길이, 높이
 - android:lines
 - 줄 수 단위의 텍스트 뷰의 높이

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

예제 프로젝트 이름:
01_UI_Basic/Ch4EditTextTest

- 입력이 가능한 텍스트 필드
- TextView의 자식 클래스
- 속성

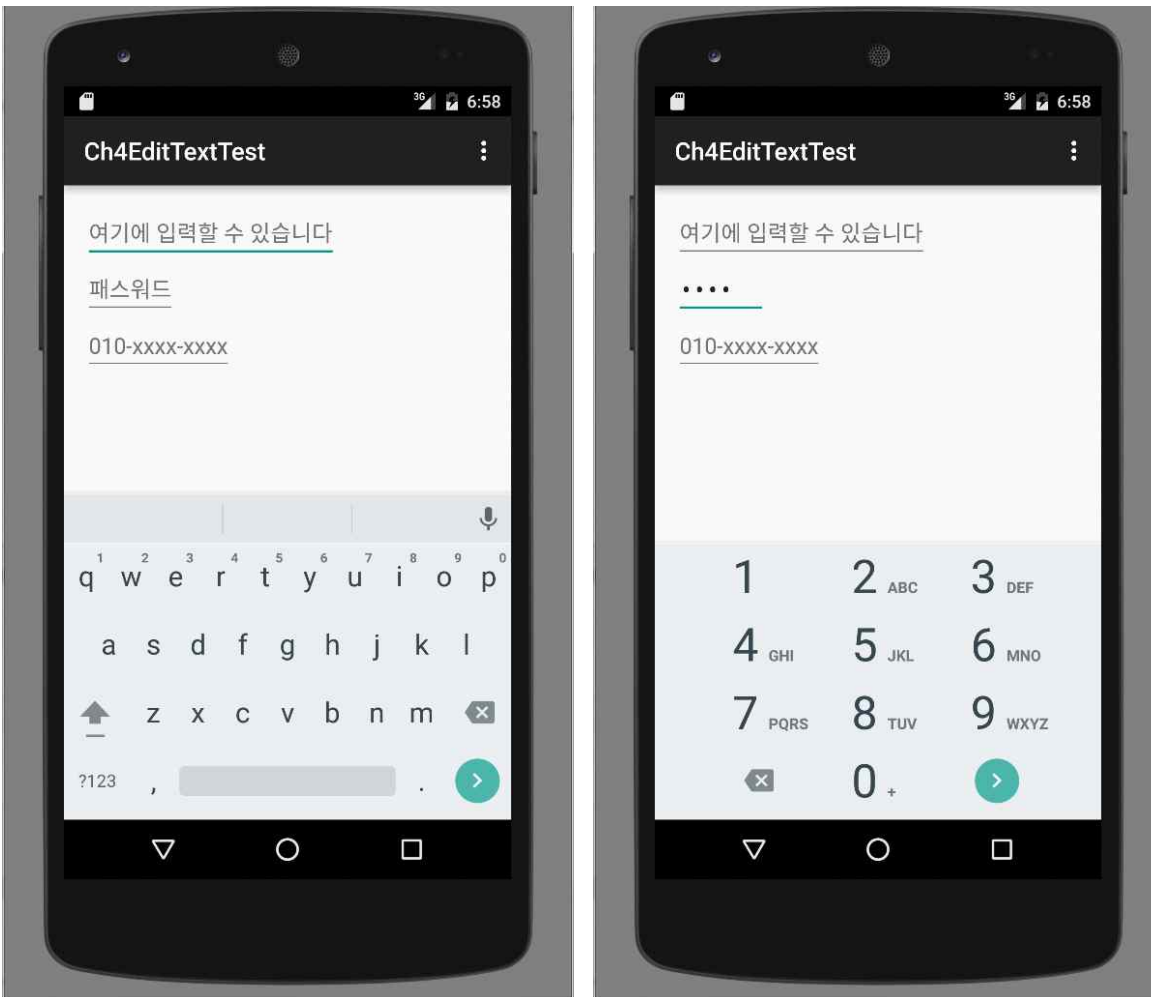
속성	설명
<code>android:autoText</code>	자동으로 타이핑 오류를 교정한다.
<code>android:drawableBottom</code>	텍스트의 아래에 표시되는 이미지 리소스이다.
<code>android:drawableRight</code>	텍스트의 오른쪽에 표시되는 이미지 리소스이다.
<code>android:editable</code>	편집가능
<code>android:text</code>	표시되는 텍스트이다.
<code>android:singleLine</code>	<code>true</code> 이면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

InputType 속성

- 입력되는 내용을 제한할 수 있음
 - 예: android:inputType="numberPassword"
 - 숫자만 입력할 수 있는 입력기가 화면에 표시

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

EditText 예제



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

```
...
```

```
<EditText  
    android:id="@+id/edit1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="여기에 입력할 수 있습니다"  
    android:inputType="text"  
    android:singleLine="true"  
/>
```

```
<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>
```

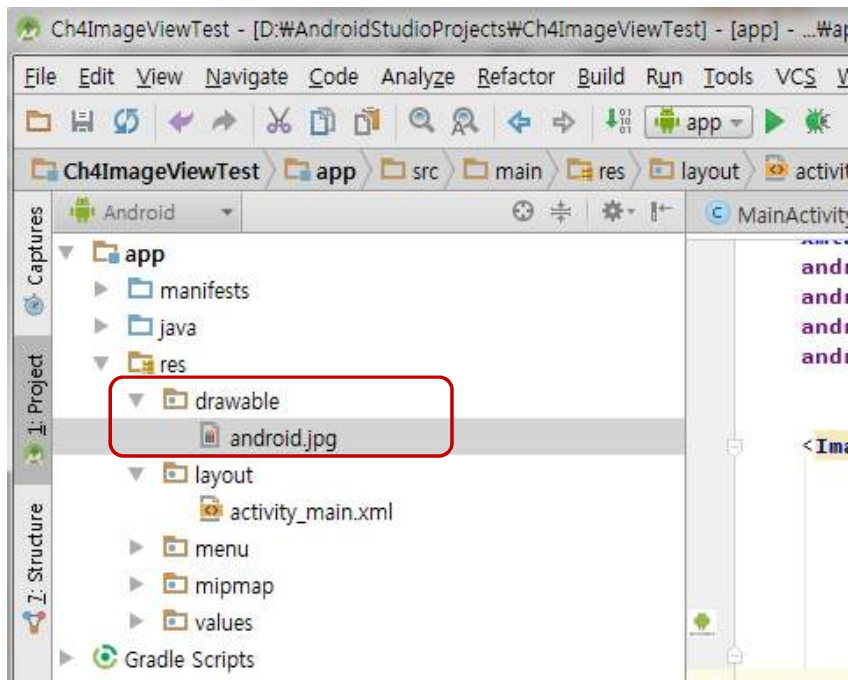
ImageView

- 아이콘 같은 이미지를 표시하는데 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

ImageView 예제

예제 프로젝트 이름:
01_UI_Basic/Ch4ImageViewTest



<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/android"
/>
```

Button

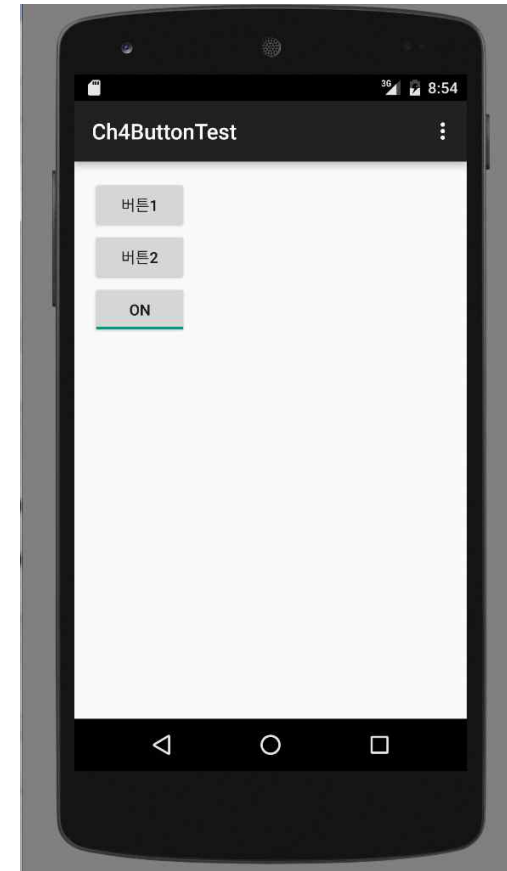
예제 프로젝트 이름:
01_UI_Basic/Ch4ButtonTest

- 사용자가 클릭할 수 있는 가장 기본적인 위젯
 - TextView 클래스를 상속 받음
 - TextView의 모든 속성 사용 가능
- ToggleButton
 - on/off 두 가지 상태를 왔다 갔다 하는 버튼

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

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

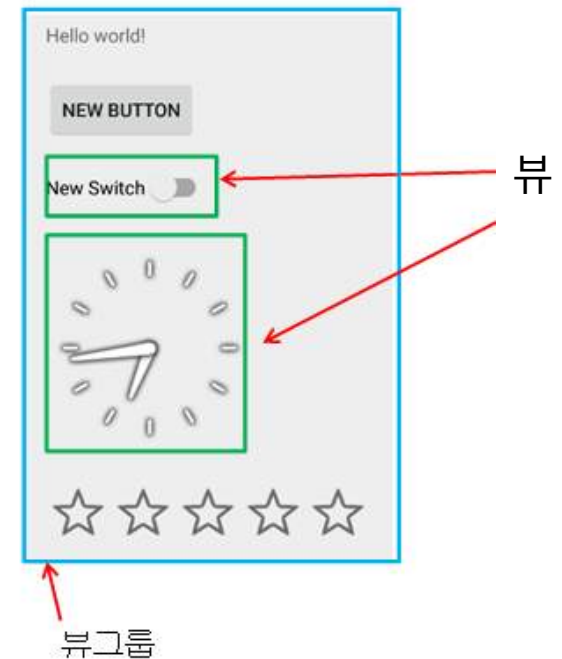
<ToggleButton
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="토글 버튼"
    android:id="@+id/button_toggle"/>
```



레이아웃 (Layout)

레이아웃

- 뷰를 화면에 배치하는 방법
- 절대 위치(좌표) vs. 상대 위치
 - 안드로이드 기기 (스마트폰, 태블릿)는 화면 크기와 해상도가 다양하기 때문에 절대 좌표를 가지고 뷰의 위치를 지정하는 방식이 아니라 뷰들의 상대적인 배치를 지정하는 방식이 쓰임



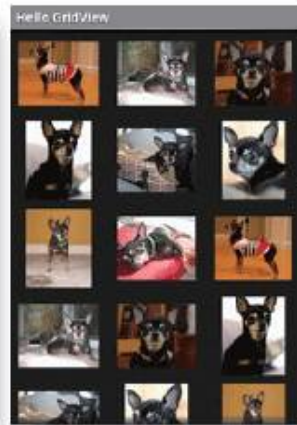
레이아웃 종류



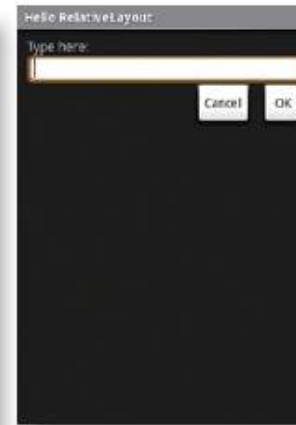
LinearLayout



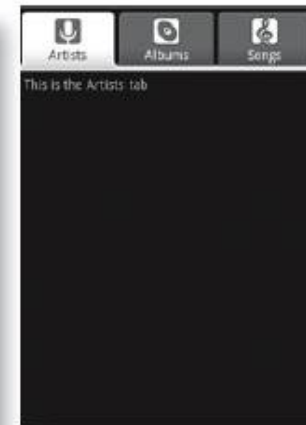
TableLayout



GridLayout



RelativeLayout

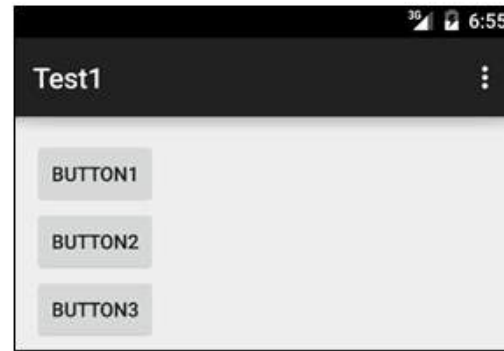
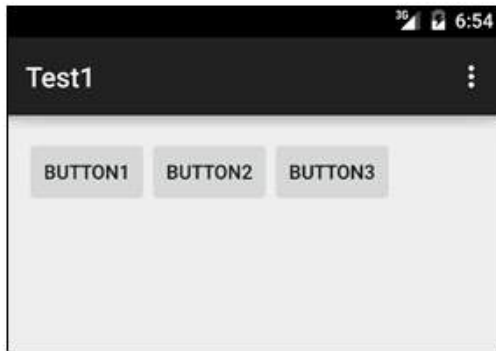


TabLayout

레이아웃 클래스	설명
LinearLayout	하위 뷰(자식)를 수직이나 수평으로 배치
TableLayout	하위 뷰(자식)를 테이블 형태로 배치 (행과 열)
GridLayout	바둑판 모양으로 배치
RelativeLayout	상대적으로 배치 (예: 다른 뷰의 왼쪽, 오른쪽)
ConstraintLayout	RelativeLayout과 유사. Android Studio 3.0 버전부터 기본 layout으로 적용
TabLayout	탭을 이용하여 겹쳐진 하위 뷰 중 하나를 선택할 수 있도록 배치

LinearLayout

예제 프로젝트 이름:
01_UI_Basic/Ch5LinearLayoutTest



속성	관련 메소드	설명
orientation	setOrientation(int)	“horizontal”: 수평 배치, “vertical”: 수직 배치
gravity	setGravity(int)	x축과 y축 상에 하위 뷰를 어떻게 배치할 것인지를 지정
baselineAligned	setBaselineAligned(boolean)	하위 뷰의 하단 기준선 정렬 여부 지정 (true: 정렬)

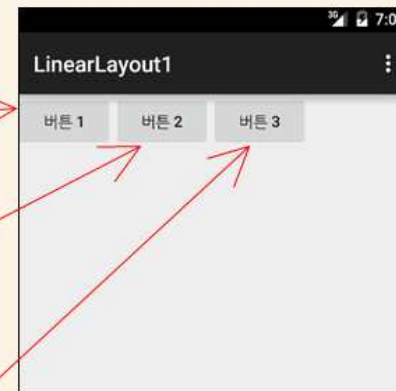
<https://developer.android.com/reference/android/widget/LinearLayout.html>

horizontal 이용 예제

main.xml

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

자식을 수평으로 배치



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 우단에 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 컨테이너 가득 채우도록 배치

gravity 속성 예제

main.xml

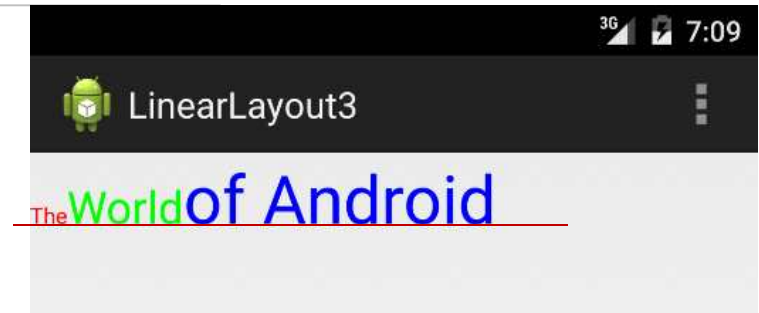
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.
    android:orientation="vertical"
    android:layout width="match parent"
    android:layout height="match parent"
    android:gravity="center"
>
<Button android:id="@+id/button01"
    android:layout width="wrap content"
    android:layout height="wrap content"
    android:text="버튼1"
/>
</LinearLayout>
```

“자식뷰를 중앙에 배치할
것!”이라는 의미이다.

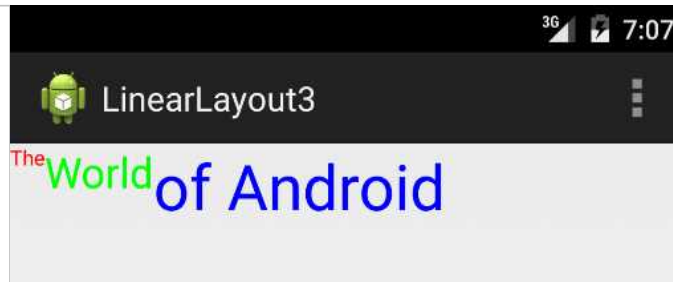


baselineAligned 예제

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:baselineAligned="true"
>
```

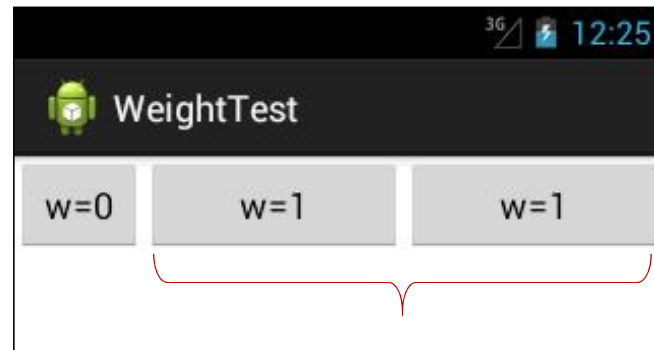
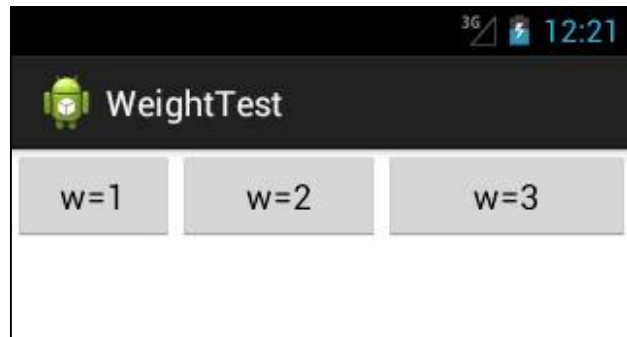


```
<LinearLayout
...
...
    android:baselineAligned="false"
>
```



weight 속성

- 공간 할당의 가중치를 줄 수 있음
 - 예: 자식 뷰들의 가중치가 각각 1, 2, 3이면,
남아있는 공간의 1/6, 2/6, 3/6을 각각 할당 받음
- 각 View에서 weight 속성을 설정하지 않으면 0으로 인식
 - 확장되지 않음



남은 공간을 동일하게 나누어서 차지

weight 예제

- 각 view의 android:layout_weight 속성

```
<Button
    android:id="@+id/button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:text="클 등록" />

<TextView
    android:id="@+id/textview01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:text="이름" />

<EditText
    android:id="@+id/edittext01"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0" />

<TextView
    android:id="@+id/textview02"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:text="의견" />

<EditText
    android:id="@+id/edittext02"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1" />

</LinearLayout>
```

가중치를 0으로 설정

가중치를 1으로 설정

LinearLayout4

클 등록

이름

love is a open door

의견

Okay, can I ...

RelativeLayout

- 자식 뷰의 위치를 부모 뷰나 다른 자식 뷰 들에 상대적으로 지정
 - 먼저 등장하는 뷰를 기준으로 상대적인 위치를 결정하게 됨
 - 따라서 기준이 되는 뷰는 먼저 나와야 함

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

<https://developer.android.com/reference/android/widget/RelativeLayout.html>

RelativeLayout 예제

예제 프로젝트 이름:
01_UI_Basic/Ch5RelativeLayoutTest

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" />
</RelativeLayout>
```



address 아래에 배치

input 아래에 배치

cancel의 왼쪽에 배치

TableLayout

- 자식 뷰들을 테이블 형태로 배치
 - 하나의 테이블에는 여러 개의 TableRow로 구성
 - 각 TableRow에 여러 개의 셀이 포함. 하나의 셀에는 하나의 뷰 객체

```
<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="서울시종로구 120"/>
    </TableRow>

    <TableRow>
        <TextView android:text="이름"/>
        <EditText android:text="홍길동"/>
    </TableRow>

    <TableRow>
        <Button android:text="저장"/>
        <Button android:text="취소"/>
    </TableRow>
</TableLayout>
```

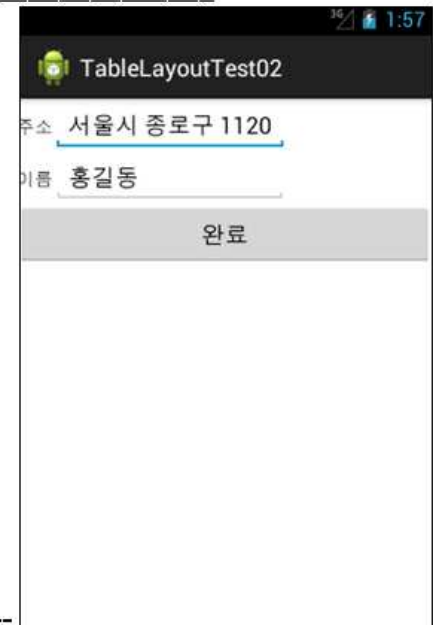


TableLayout 예제 2

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 1120"/>
    </TableRow>
    <TableRow>
        <TextView android:text="이름"/>
        <EditText android:text="홍길동"/>
    </TableRow>
    <Button android:text="완료"/>
</TableLayout>
```

TableRow 안에 넣지
않더라도 뷰 객체가
하나의 행으로 배치



<https://developer.android.com/reference/android/widget/TableLayout.html>