한국ICT인재개발원

[한국ICT인재개발원] 안드로이드

3. 안드로이드 위젯 숙련도 높이기

강의자 소개 : 채종훈 강사

前) 광고데이터 분석 1년

前) IT강의 경력 2년 6개월

前) 머신러닝을 활용한 데이터 분석 프로젝트반 운영 1년

前) 리그오브 레전드 데이터 분석 등...

現) 국비반 강의 진행중

위젯 개념 이해하기

안드로이드의 위젯은 마치 **html**의 태그요소와 비슷합니다.

텍스트뷰, 버튼, 라디오버튼, 이미지 등이 있으며 이 모든 부품들은

전부 **View**라는 클래스의 상속을 받습니다.

위젯을 담을 수 있는 공간을 레이아웃이라고 부르며

레이아웃 내부에 배치된 요소들을 위젯이라고 부르는 식입니다.

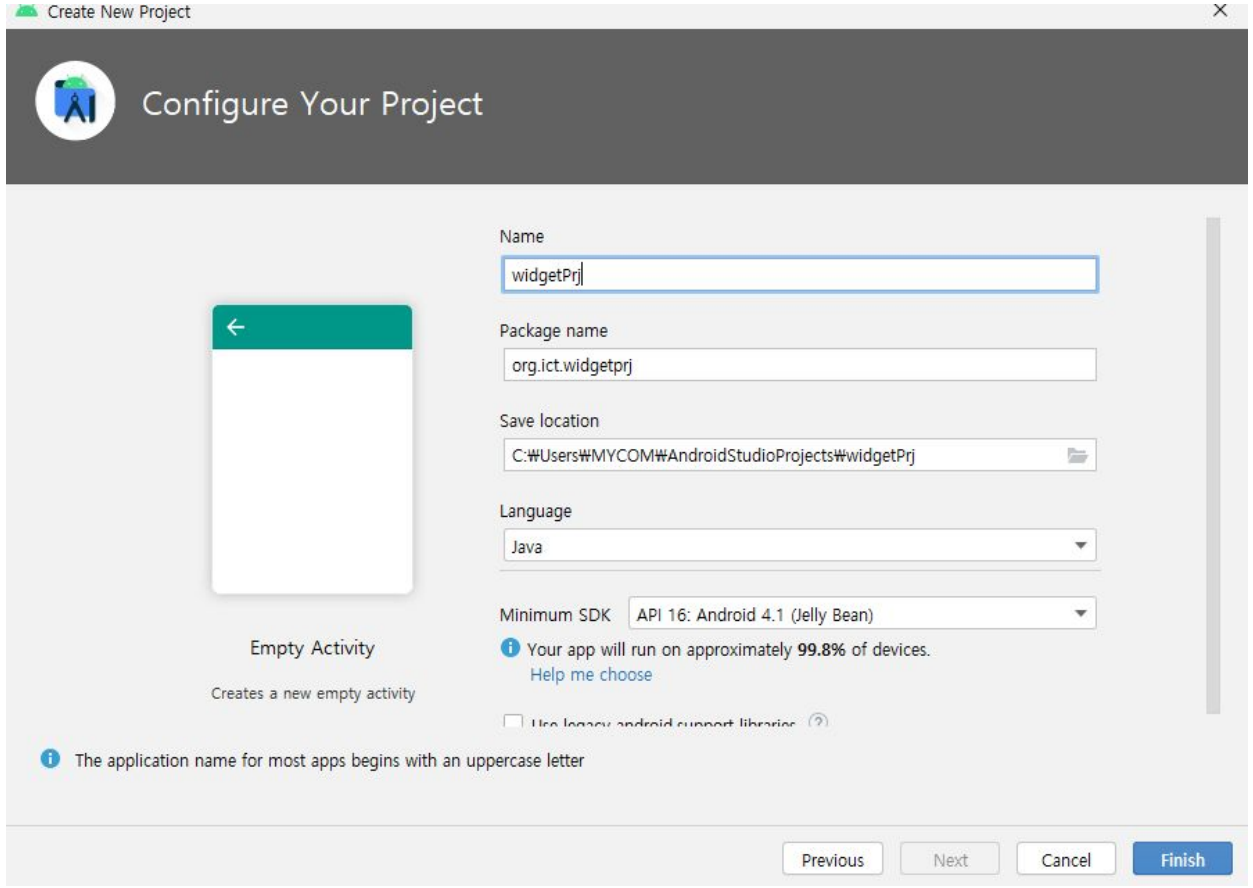
위젯 개념 이해하기


```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button1"
    android:text="@string/strBtn1"
    ></Button>
```

위젯 내부에는 **android:** 로 시작하는 속성값을 줄 수 있으며

이는 마치 **css** 내지는 **html**의 속성값(**id, class** 등)처럼 기능합니다.

위젯 개념 이해하기



먼저 **android:** 요소를 공부하기 위해 프로젝트를 생성하겠습니다.
widgetprj라는 이름으로 만들면 됩니다.

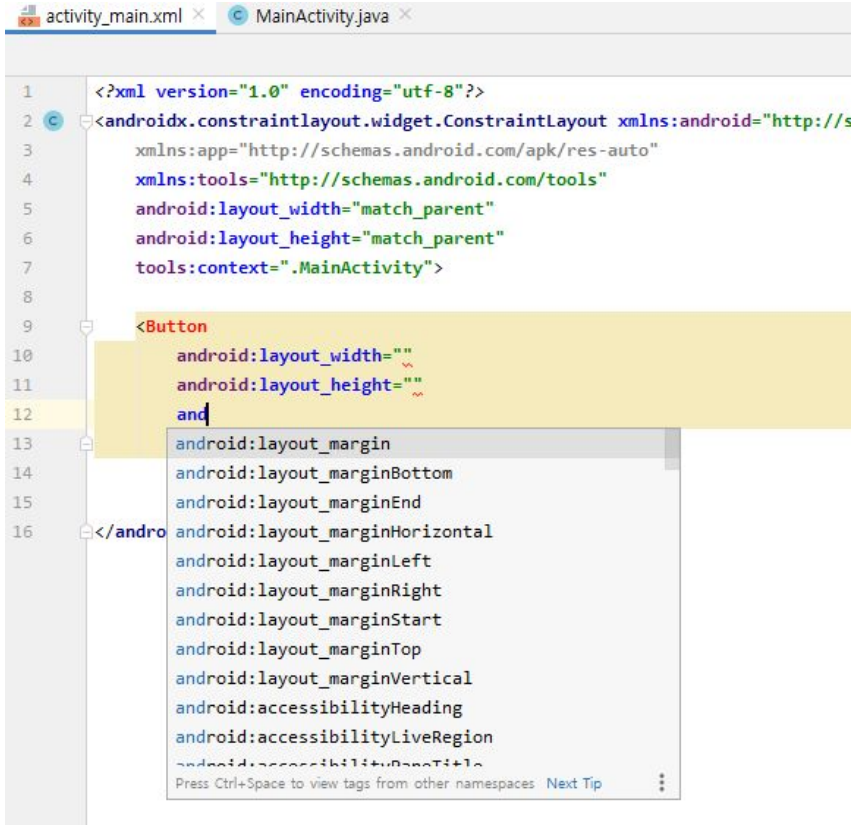
위젯 개념 이해하기

activity_main.xml MainActivity.java

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9
10
11 </androidx.constraintlayout.widget.ConstraintLayout>
```

activity_main.xml로 이동해 요소를 지워줍니다.

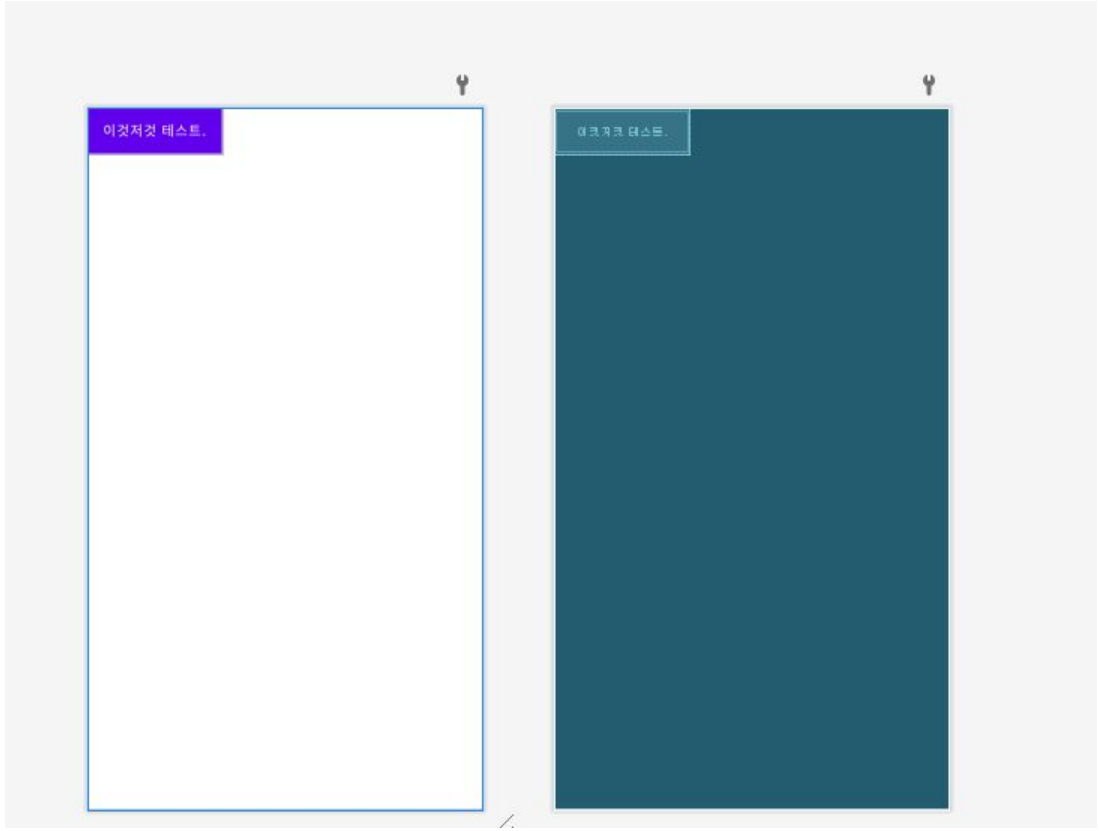
위젯 개념 이해하기



가장 많이 쓰는 **Button**을 예시로 들면 아래와 같이 **100**개가 넘는 속성을 지정할 수 있습니다.
따라서 이런 정보를 확인하려면
https://developer.android.com/reference/packages
로 접속해서 검색해보시면 됩니다.

위젯 개념 이해하기

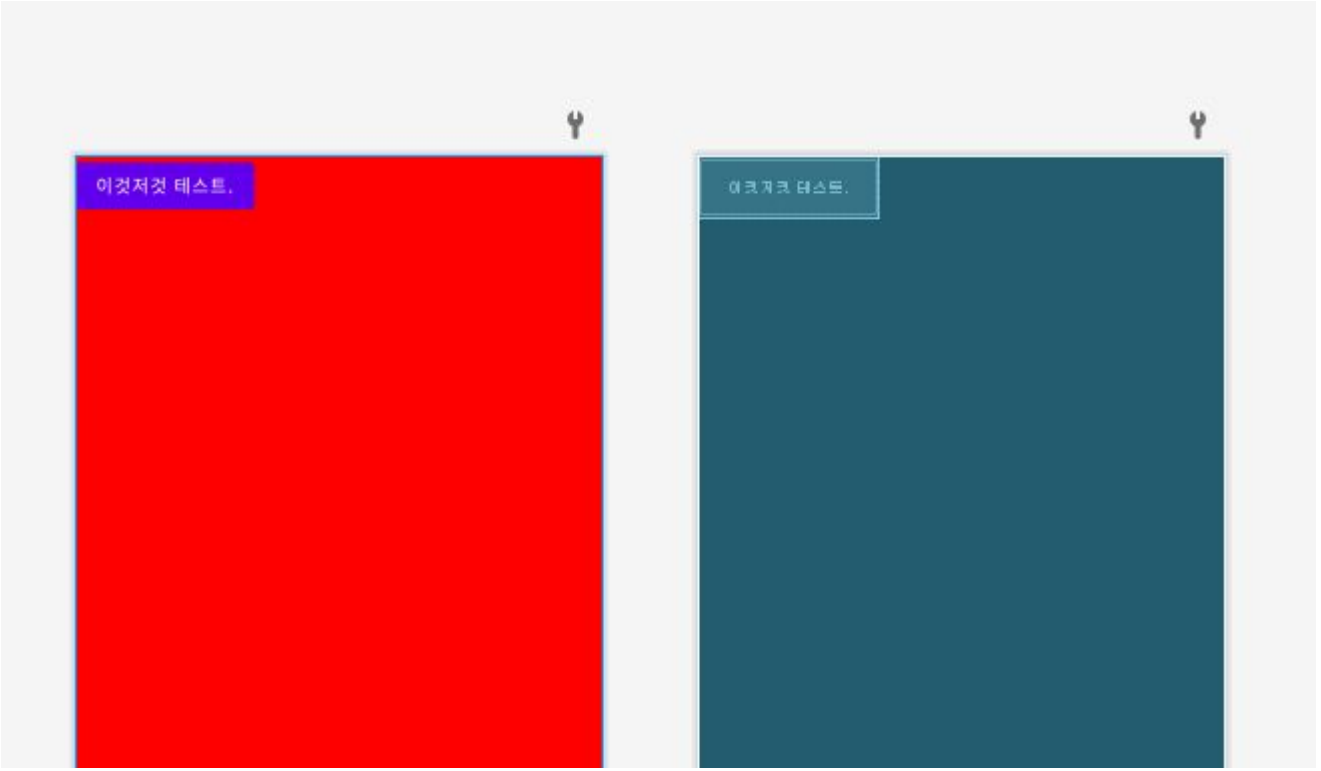

```
<Button
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="이것저것 테스트."></Button>
```



보시는바와같이 레이아웃의 너비, 높이와 텍스트가 들어가는 것으로 실제 위젯이 배치되는것을 볼 수 있습니다.

위젯 개념 이해하기


```
androidx.constraintlayout.widget.ConstraintLayout xmlns
    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:background="#ff0000"
    tools:context=".MainActivity">
```



또한 배경에 **background** 속성값을 입혀주면 디자인이 바뀌는것을 볼 수 있습니다.

위젯 개념 이해하기

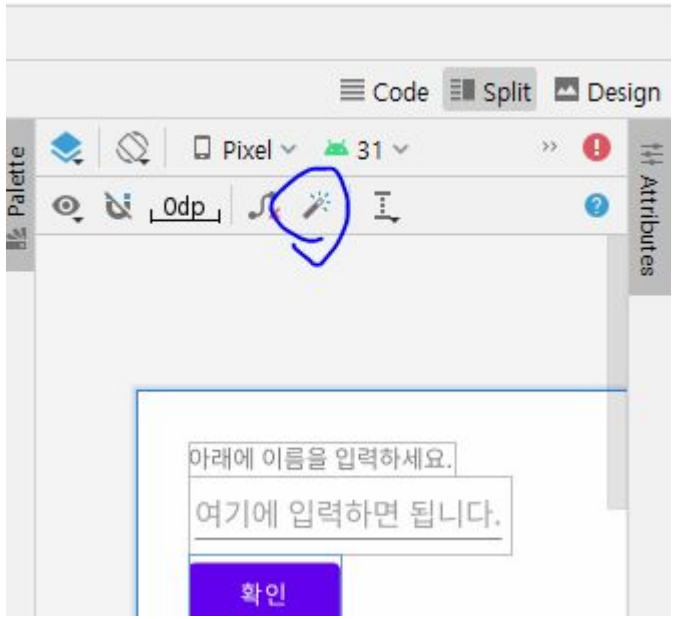
이 정보들은 꼭 **xml**로 변환하지 않아도 오른쪽 탭에서도 변환 가능하며 드래그앤 드롭으로 위치변경도 가능합니다.

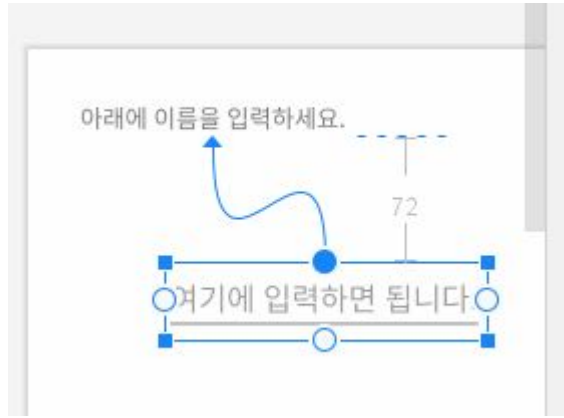
위젯 개념 이해하기


```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/textView1"></TextView>

<RadioButton
    android:id="@+id/female"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="여성" />

<RadioButton
    android:id="@+id/male"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="남성" />
```

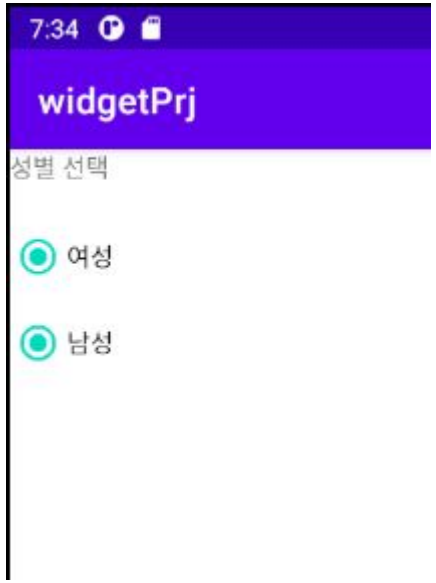
Radio버튼과 같이 html에서 form형식으로 친숙하게 사용하던 요소를 역시 사용할 수 있습니다.

위젯 개념 이해하기





작성 완료후에 바로 빌드하면 위젯들이 겹쳐져서 나옵니다.
따라서, 왼쪽과같이 작성 완료 후 위젯들을 위치에 맞게 배치한 다음 마법봉 모양 아이콘을 눌러주시거나, 혹은 우측과 같이 상하 위치를 잡아준뒤 확인해야 제대로 배열한 위치에 위젯이 배치됩니다.



라디오버튼과 **textView**를 조합한 모습입니다.
이런식으로 특정 웹 페이지를 복사해 앱으로 옮길 수도 있습니다.

위젯 개념 이해하기


```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    ></Button>
```

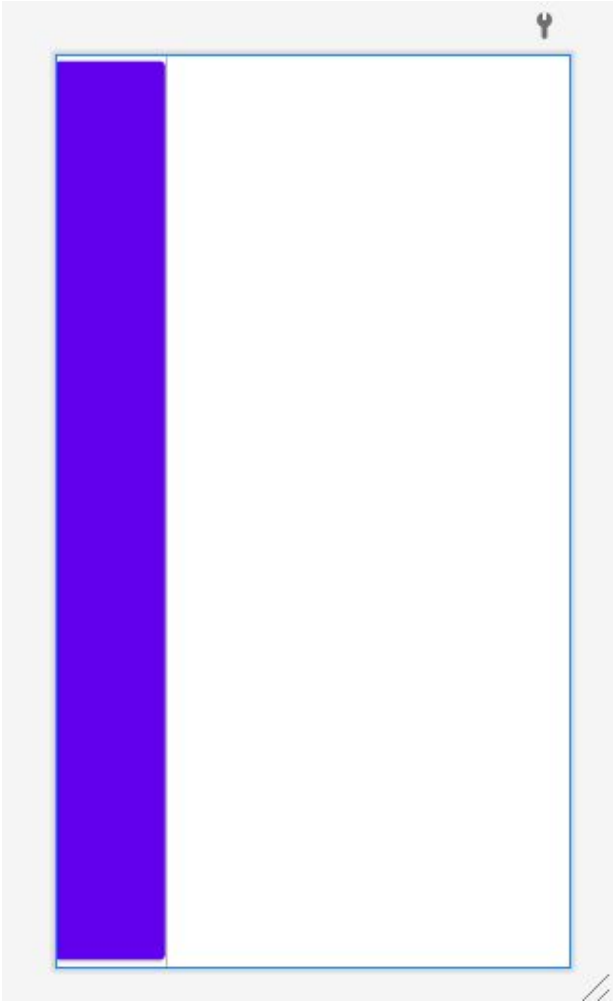
버튼의 위치를 이용해 **wrap_content, match_parent**에 대해 알아보겠습니다.

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    ></Button>
```

match_parent는 해당 방향으로 꽉 채우는것이고, **wrap_content**는 우선순위가 있는 방향에 배치합니다.
width가 **match_parent**이므로 좌우로는 가득 찬, 그리고 **height**가 **wrap_content**인데 높낮이는 위쪽이 우선순위이므로 상단에 가득 찬 버튼이 배치됩니다.

위젯 개념 이해하기

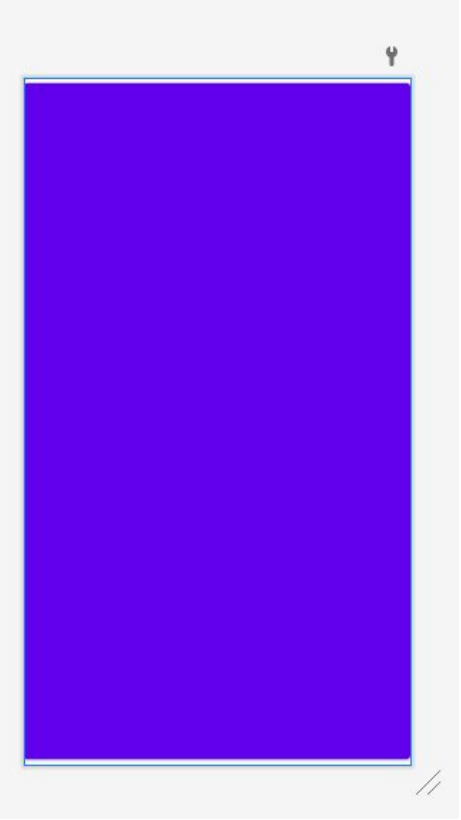

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    ></Button>
```



이번엔 **width**가 **wrap_content**이므로 왼쪽에 배치되고,
height가 **match_parent**이므로 위 아래를 가득 채웁니다.

위젯 개념 이해하기

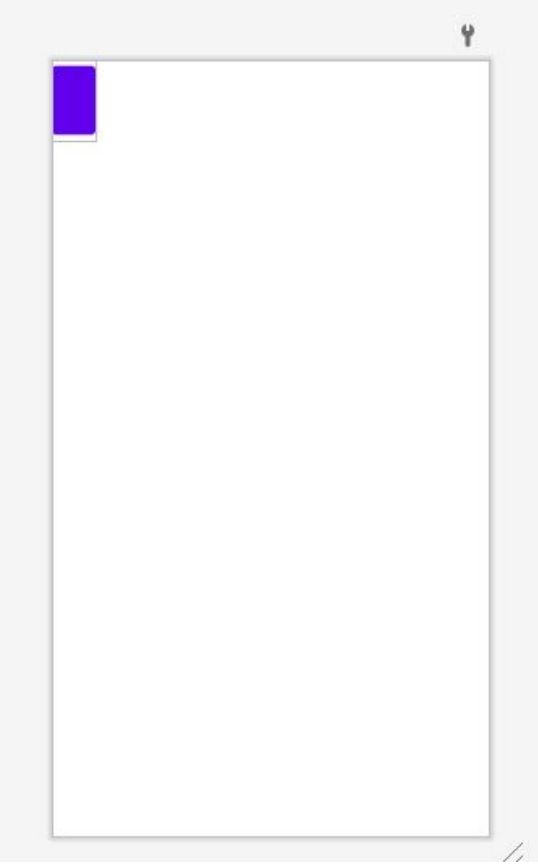

```
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    ></Button>
```



width, height 모두 화면을 꽉 채우므로 전체 화면이 버튼으로 구성됩니다.

위젯 개념 이해하기

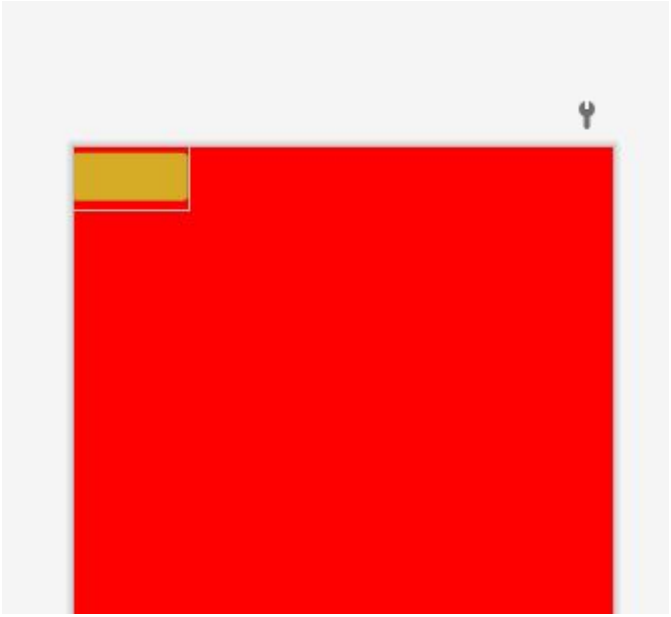

```
<Button
    android:layout_width="108px"
    android:layout_height="200px"
    ></Button>
```



혹은 위와 같이 크기를 절대값으로 지정할수도 있습니다.
이 경우 왼쪽 위가 우선순위를 가집니다.

위젯 개념 이해하기

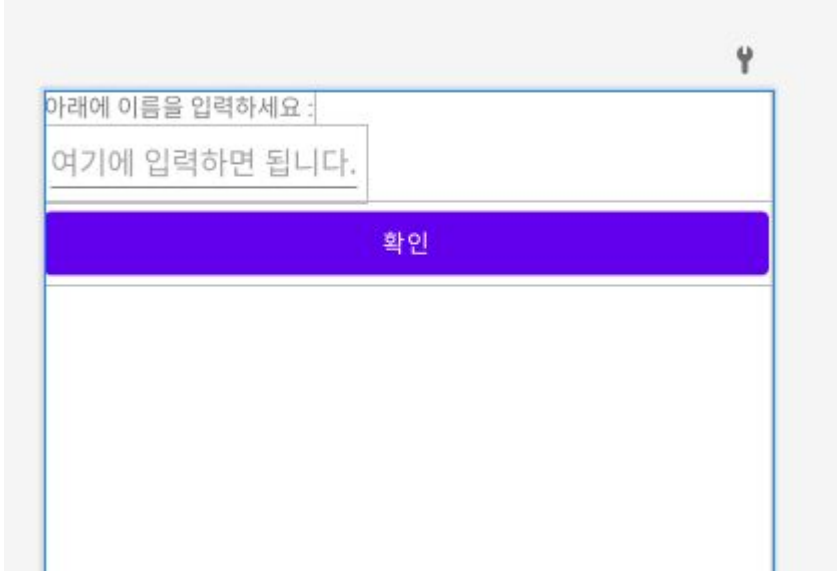

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#ff0000"
tools:context=".MainActivity">

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="#B4D1BB27"></Button>
```



레이아웃 색깔은 **background**속성으로 바꿀 수 있고
버튼색상은 **backgroundTint**속성을 부여해 바꿉니다.

위젯 개념 이해하기

```
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="아래에 이름을 입력하세요 :" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="여기에 입력하면 됩니다." />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="확인"></Button>

androidx.constraintlayout.widget.ConstraintLayout>
```



textview는 텍스트 내부 문장을 표현할 수 있고,
editText는 위와 같이 hint를 이용해 마치 html의 input 태그의 placeholder처럼 쓸 수 있습니다.

위젯 개념 이해하기

```
xmlns:tools="http://schemas.android.com/toc
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="30dp"
tools:context=".MainActivity">

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="아래에 이름을 입력하세요 :"
```

레이아웃의 **padding**값은 화면 가장 가장자리쪽을 기준으로 얼마나 요소가 떨어지게 배치할지를 의미합니다. 위와같이 **padding**만 작성시 **30dp**만큼 위 아래 양 옆에서 떨어지는 위치부터 요소가 배치됩니다.

위젯 개념 이해하기

```
xmlns:app="http://schemas.android.com/apk/res-a
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingTop="30dp"
android:paddingBottom="30dp"
android:paddingLeft="60dp"
android:paddingRight="100dp"
tools:context=".MainActivity">
```

레이아웃의 **padding**은 또한 **Top, Bottom, Left, Right**와 같이 일부 방향에만 적용 가능합니다.

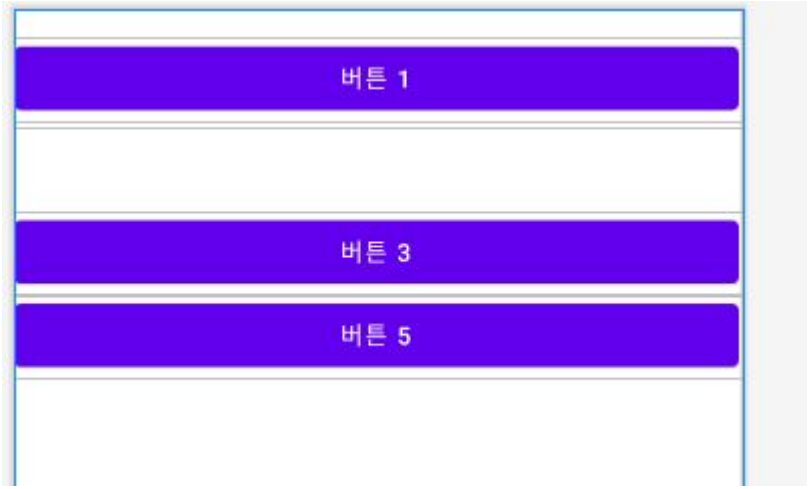
위젯 개념 이해하기


```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="아래에 이름을 입력하세요 :" />

<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:hint="여기에 입력하면 됩니다."
    app:layout_constraintTop_toBottomOf="@+id/textView"
    tools:layout_editor_absoluteX="0dp" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="확인"
    android:layout_margin="20dp"
    app:layout_constraintTop_toBottomOf="@+id/editText"
    tools:layout_editor_absoluteX="16dp"></Button>
```

padding이 레이아웃의 가장자리에서 얼마나 떨어진 쪽에 요소를 위치하게 만들지에 대한 것이라면, **margin**은 위젯간 최소 간격을 정의합니다.

위젯 개념 이해하기


```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 1"
    ></Button>
<Button
    android:visibility="invisible"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 2"
    ></Button>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 3"
    ></Button>
<Button
    android:visibility="gone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 4"
    ></Button>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 5"
    ></Button>
```



Button 위젯에는 visibility 속성을 부여할 수 있는데 크게 3가지 속성을 붙일 수 있습니다. visible은 기본값으로, 버튼이 화면에 보이며, invisible은 화면에는 보이지 않지만 자리를 점유하고, gone은 화면 밖에만 존재합니다.

위젯 개념 이해하기


```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 1"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="7dp"></Button>

<Button
    android:enabled="false"
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="버튼 2"
    app:layout_constraintTop_toBottomOf="@+id/button"
    tools:layout_editor_absoluteX="0dp"></Button>

<Button
    android:clickable="false"
    android:id="@+id/button3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="버튼 3"
    app:layout_constraintTop_toBottomOf="@+id/button2"
    tools:layout_editor_absoluteX="16dp"></Button>
```

버튼은 사용 가능 불가능, 클릭 가능 불가능 여부를 지정할 수 있습니다.
enabled는 사용 가능 불가능 여부이며 **true, false**로 각각 지정합니다.
clickable역시 클릭 가능 불가능 여부이며 **true, false**로 각각 지정합니다.

속성	비고
text	문자열 표시
textColor	문자의 색상 변경
textSize	글자의 크기를 dp, px, in, mm, sp 단위로 지정
typeface	글자의 글꼴 지정, sans, serif, monospace등을 설정가능하며, 디폴트는 normal
textStyle	글자 스타일 지정, bold, italic, bold|italic 등을 설정가능하며, 디폴트는 normal
singleLine	글이 너무 길어지면 개행 없이 ..... 을 붙이는 식으로 생략

텍스트뷰도 역시 여러 속성을 가지고 있습니다.
글자의 색상, 크기, 글꼴, 스타일(기울기, 볼드 등), 싱글라인(개행 없이 표시) 등을 가지고 있으며, 예제코드를 순차적으로 작성해보겠습니다.

위젯 개념 이해하기


```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="1번 텍스트뷰"
    android:textSize="50dp"
    ></TextView>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="2번 텍스트뷰"
    android:textColor="#00ff00"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="89dp"></TextView>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="144dp"
    android:text="3번 텍스트뷰"
    android:textStyle="italic|bold"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteX="0dp"></TextView>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="4번 텍스트뷰"
    android:typeface="serif"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="195dp"></TextView>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="5번 텍스트뷰5번 텍스트뷰5번 텍스트뷰5번 텍스트뷰5번 텍스
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="249dp"
    android:singleLine="true"></TextView>
```

텍스트뷰도 역시 여러 속성을 가지고 있습니다.
글자의 색상, 크기, 글꼴, 스타일(기울기, 볼드 등), 싱글라인(개행 없이 표시)
등을 가지고 있으며, 예제코드를 순차적으로 작성해보겠습니다.

위젯 개념 이해하기

TextView 메서드	비고
.setText()	TextView 내부에 출력되는 문자를 바꿉니다.(문자열)
.setTextColor()	출력 문자의 색깔을 바꿉니다. (Color.색깔을 파라미터로 주어 변경합니다.)
.setTextSize()	출력 문자의 크기를 바꿉니다. (정수)
.setTypeFace()	출력 글꼴이나, 스타일을 바꿉니다. (Typeface.글꼴(스타일) 을 파라미터로 주어 변경합니다.)
.setSingleLine()	파라미터 없이 그냥 한 줄에 모든 문자열을 생략해가며 출력합니다.

XML쪽에 모든 속성을 작성할 수도 있지만, 버튼의 색상이 그랬듯 자바 코드를 이용해 여러가지 속성을 지정할 수 있습니다.
위는 메서드 예시입니다.

위젯 개념 이해하기


```xml
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView 연습 1"
    android:id="@+id/textView1" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="72dp"
    android:text="TextView 연습 2"
    app:layout_constraintTop_toBottomOf="@+id/textView1"
    tools:layout_editor_absoluteX="0dp" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="72dp"
    android:text="TextView 연습 3"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    tools:layout_editor_absoluteX="0dp" />
```

```java
// TextView 3개를 선언합니다.
TextView tv1, tv2, tv3;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //TextView 3개를 연결해보세요.
    tv1 = (TextView)findViewById(R.id.textView1);
    tv2 = (TextView)findViewById(R.id.textView2);
    tv3 = (TextView)findViewById(R.id.textView3);

    // TextView1의 문장을 변환하기
    tv1.setText("안녕하세요? TextView 1번입니다.");
    tv1.setTextColor(Color.CYAN);
    //TextView2의 글씨 크기를 크게 키워주세요.
    // 글꼴은 Serif로, 스타일은 italic으로 해 주세요.
    tv2.setTextSize(50);
    tv2.setTypeface(Typeface.SERIF, Typeface.ITALIC);

    //TextView3의 문장을 100글자 이상으로 늘려주세요.
    //singleLine을 걸어주세요.
    tv3.setText("가나다라마바사아자차카타파하가나다라마바
    tv3.setSingleLine();
```

위와 같이 **xml**에서 배치한 위젯에 대해, 자바코드상에서 연결한 후 추가적으로 속성을 변경하거나 부여할 수도 있습니다.
이를 통해 이벤트 처리까지 할 수 있습니다.

```
String Result = EditText.getText().toString()
```

다음으로 **EditText**에 있는 값을 가져오는 방법에 대해 보겠습니다.
getText()를 이용하면 값을 가져올 수 있고, 이를 문자로 바꾸기 위해서는 다시 뒤에 **toString()**을 붙이빈다.

위젯 개념 이해하기

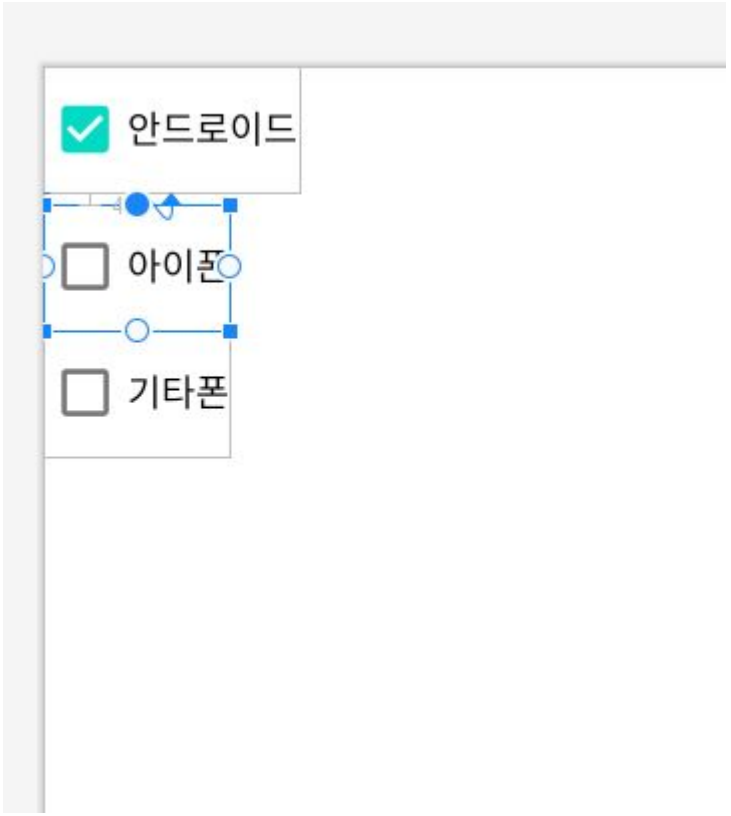
버튼과 EditText를 함께 활용해서 계산기를 만들어보겠습니다.

위젯 개념 이해하기

계산기 다음은 컴파운드 버튼이라 불리는 요소입니다.
체크박스, 라디오버튼, 스위치, 토글 등이 그것입니다.

위젯 개념 이해하기

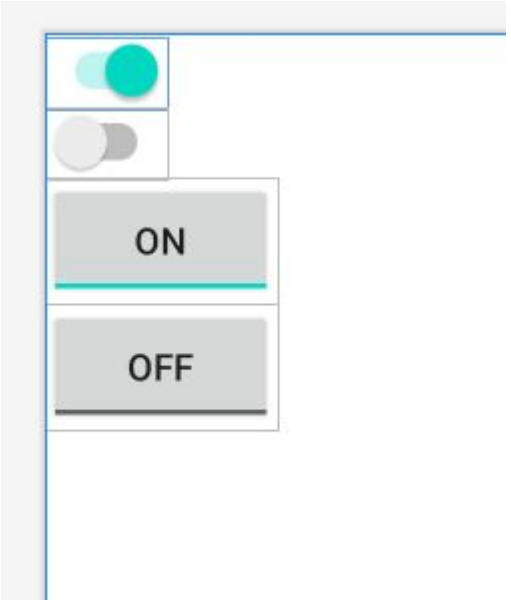

```
<CheckBox
    android:id="@+id/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true"
    android:text="안드로이드"></CheckBox>

<CheckBox
    android:id="@+id/ios"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp"
    android:text="아이폰"
    app:layout_constraintTop_toBottomOf="@+id/android"
    tools:layout_editor_absoluteX="0dp"></CheckBox>

<CheckBox
    android:id="@+id/others"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="기타폰"
    app:layout_constraintTop_toBottomOf="@+id/ios"
    tools:layout_editor_absoluteX="0dp"></CheckBox>
```



계산기 다음은 컴파운드 버튼이라 불리는 요소입니다.
체크박스, 라디오버튼, 스위치, 토글 등이 그것입니다.

위젯 개념 이해하기


```
<Switch
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true" />
<Switch
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="false" />
<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true" />
<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="false" />
```



계산기 다음은 컴파운드 버튼이라 불리는 요소입니다.
체크박스, 라디오버튼, 스위치, 토글 등이 그것입니다.

위젯 개념 이해하기


```
<RadioGroup
    android:id="@+id/rGroup1"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content">

    <RadioButton
        android:text="남성" />
    <RadioButton
        android:text="여성" />

</RadioGroup>
```

계산기 다음은 컴파운드 버튼이라 불리는 요소입니다.
체크박스, 라디오버튼, 스위치, 토글 등이 그것입니다.

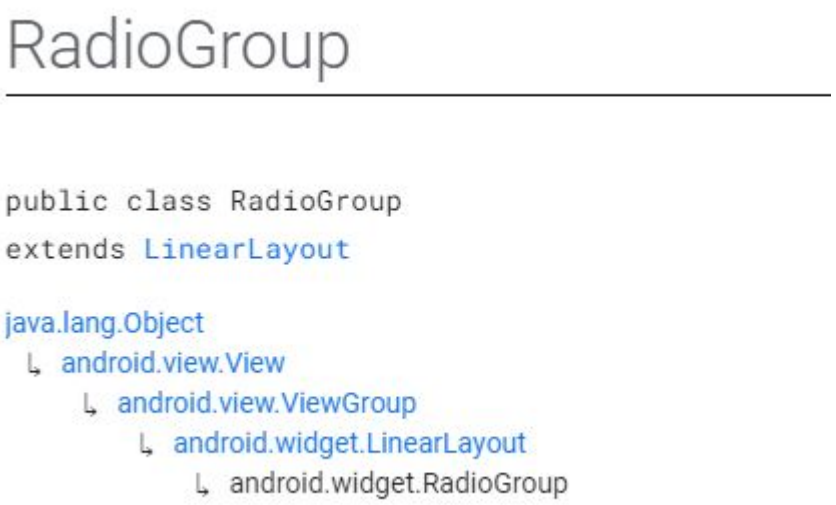
위젯 개념 이해하기


```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/thumbnail" />

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/thumbnail"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="127dp" />

<ImageView
    android:layout_width="300dp"
    android:layout_height="100dp"
    android:scaleType="fitCenter"
    android:src="@drawable/thumbnail"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="266dp" />

<ImageView
    android:layout_width="300dp"
    android:layout_height="100dp"
    android:scaleType="fitXY"
    android:src="@drawable/thumbnail"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="378dp" />
```

계산기 다음은 컴파운드 버튼이라 불리는 요소입니다.
체크박스, 라디오버튼, 스위치, 토글 등이 그것입니다.

```
checkbox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener(){
   public void onCheckedChanged(CompoundButton buttonView, boolean isChecked){
     코드작성..
   }
}
```

계산기 다음은 컴파운드 버튼이라 불리는 요소입니다.
체크박스, 라디오버튼, 스위치, 토글 등이 그것입니다.
위 요소들은 **CompoundButton**을 상속받아서 위와 같은 방식으로
이벤트를 연결할 수 있습니다.



라디오버튼은 **html**에서는 원래 **name**속성이 일치하는 요소들끼리 하나의 그룹으로 묶였지만, 안드로이드에서는 **RadioGroup**라는 태그 내부에 **RadioButton**을 새로 작성하는 방식으로 묶어줍니다.

위젯 개념 이해하기

이미지뷰는 그림을 출력하는 위젯입니다.
그림파일은 일반적으로 프로젝트의 **res - drawable** 폴더에 집어넣습니다.
접근은 **@drawable/그림파일아이디** 형식으로 접근합니다.