

레이아웃 구현





- 레이아웃 개요
- 리니어레이아웃(LinearLayout)
- 렐러티브레이아웃(RelativeLayout)
- 프레임레이아웃(FrameLayout)
- 테이블레이아웃(TableLayout)
- 그리드레이아웃(GridLayout)



레이아웃 개요 - 레이아웃 기본 개념

레이아웃 계층도

```
java.lang.Object
```

```
└ android.view.View
```

```
└ android.widget.ViewGroup
```

```
└ android.widget.LinearLayout
```

```
└ android.widget.TableLayout
```

```
└ android.widget.RelativeLayout
```

```
└ android.widget.FrameLayout
```

```
└ android.widget.GridLayout
```

Layout 계층도

- 레이아웃은 레이아웃 자체를 보이기 보다는, 레이아웃 안에 존재하는 위젯을 배치하기 위한 용도가 더 주요한 목적
- 레이아웃 중에서 가장 많이 사용되는 것은 리니어레이아웃



○ 레이아웃: 화면 구성

- 안드로이드에서는 다양한 화면 크기를 가지는 장치들을 지원
- 화면의 크기가 통일되어 있지 않아서 (예:120,150)과 같이 절대적(고정)인 화면 위치를 사용하여 뷰를 배치하는 것 보다 상대적으로 뷰를 배치하는 것이 바람직하다.





○ 레이아웃의 대표적인 속성

- orientation : 레이아웃 안에 배치될 위젯의 수직 또는 수평방향을 설정
- gravity : 레이아웃 안에 배치될 위젯을 좌측, 우측, 중앙 등 정렬할 방향을 설정
- padding : 레이아웃 안에 배치될 위젯의 여백을 설정
- layout_weight : 레이아웃이 전체 화면에서 차지하는 공간의 가중값을 설정하는데, 여러 개의 레이아웃이 중복될 때 주로 사용
- baselineAligned : 레이아웃 안에 배치할 위젯들을 보기 좋게 정렬



레이아웃 개요 - 레이아웃의 종류

레이아웃의 기본 형태

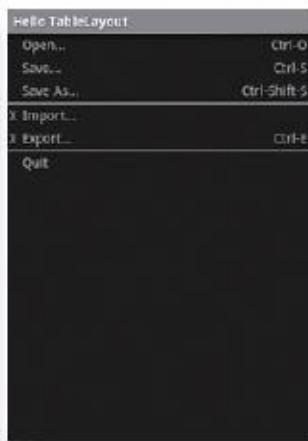


절대 좌표값으로 지정하는 앱슬루트레이아웃(AbsoluteLayout:절대적)도 있었으나, 해상도가 다른 안드로이드폰에서는 문제가 발생되기 때문에 더 이상 사용되지 않는다.

레이아웃 개요 - 레이아웃의 종류



LinearLayout



TableLayout



GridLayout



RelativeLayout



TabLayout



레이아웃 개요 - 레이아웃의 종류

○ 리니어 레이아웃(LinearLayout)

- 레이아웃의 왼쪽 위부터 아래쪽 또는 오른쪽으로 차례로 배치
- 자식들을 수직이나 수평으로 배치

○ 렐레티브 레이아웃(RelativeLayout)

- 위젯이 자신이 속한 레이아웃의 위/아래/왼쪽/오른쪽 위치를 지정하여 배치하거나, 다른 위젯으로부터 상대적인 위치를 지정
- 자식들을 부모나 다른 자식에 상대적으로 배치

○ 테이블 레이아웃(TableLayout)

- 위젯을 행과 열의 개수를 지정한 테이블 형태로 배열
- 자식들을 테이블 형태로 배치

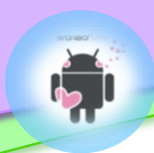
○ 그리드 레이아웃(GridLayout)

- 테이블 레이아웃과 비슷하지만, 행 또는 열을 확장하여 다양한 배치 가능
- 자식들을 바둑판 모양으로 배치

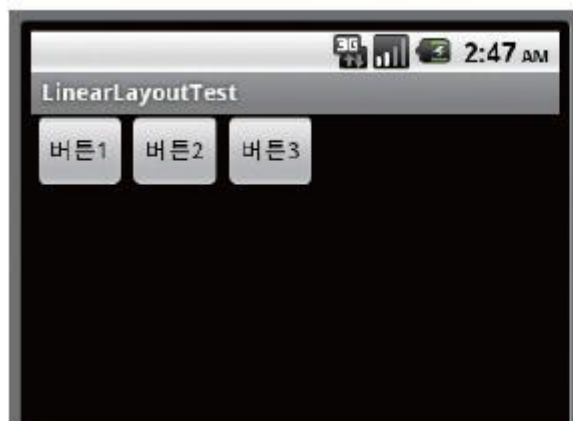
○ 프레임 레이아웃(FrameLayout)

- 위젯들을 왼쪽 위에 일률적으로 겹쳐서 배치하여 중복해서 보이는 효과를 낼 수 있다. 여러 개의 위젯들을 배치한 후 상황에 따라서 필요한 위젯을 보이는 방식에 주로 활용
- 모든 자식들을 좌측 상단에 겹치게 배치

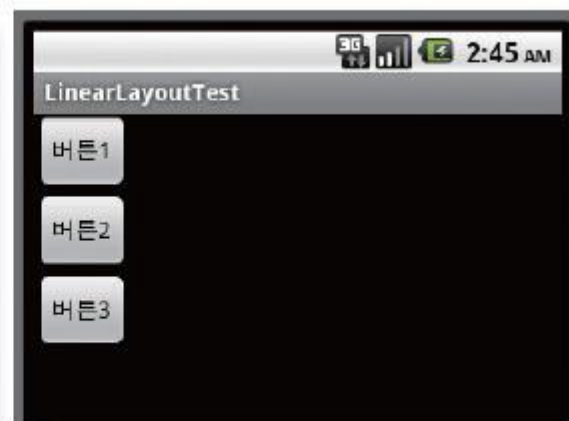
레이아웃 개요 - 기본 리니어레이아웃



XML 속성	관련 메소드	설명
orientation	setOrientation(int)	"horizontal"은 수평, "vertical"은 수직
gravity	setGravity(int)	X축과 y축 상에 자식을 어떻게 배치할 것인지를 지정
baselineAligned	setBaselineAligned(boolean)	false로 설정되면 자식 뷰들의 기 준선을 정렬하지 않는다.



(a) 수평 배치



(b) 수직 배치



레이아웃 개요 - 기본 리니어레이아웃

```
<?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="fill_parent"
```

```
    android:layout_height="fill_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>
```



레이아웃 개요 - 기본 리니어레이아웃

❖ orientation 속성과 예제

- vertical은 수직배열, horizontal은 수평배열

orientation 속성이 vertical 값인 XML 코드

```

1 <LinearLayout
2   android:orientation="vertical" >
3   <Button
4     android:layout_width="wrap_content"
5     android:layout_height="wrap_content"
6     android:text="Button" />
7   <TextView
8     android:text="TextView" />
9   <CheckBox
10    android:text="CheckBox" />
11  <RadioButton
12    android:text="RadioButton" />
13  <Switch
14    android:text="Switch" />
15 </LinearLayout>
  
```

수직 방향

내용에 맞게 조절
(wrap_content)

Button
TextView
☐ CheckBox
☐ RadioButton
Switch OFF

orientation 속성이 horizontal 값인 XML 코드

```

1 <LinearLayout
2   android:orientation="horizontal" >
3   <Button
4     android:layout_width="wrap_content"
5     android:layout_height="wrap_content"
6     android:text="Button" />
7   <TextView
8     android:text="TextView" />
9   ~~~ 중간 생략 ~~~
10 </LinearLayout>
  
```

수평 방향

Button TextView ☐ CheckBox ☐ RadioButton



레이아웃 개요 - 기본 리니어레이아웃

❖ gravity 속성

- gravity 속성은 레이아웃 안의 위젯들을 어디에 배치할 것인지를 결정
- left/right/center/top/bottom/조합(right|bottom) 등

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

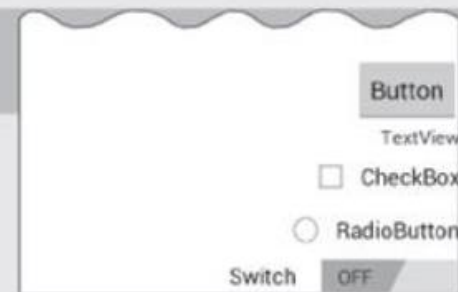
레이아웃 개요 - 기본 리니어레이아웃

❖ 예제

gravity 속성 XML 코드

```
1 <LinearLayout
2   android:orientation="vertical"
3   android:gravity="right|bottom" >
4   <Button
5     android:layout_width="wrap_content"
6     android:layout_height="wrap_content"
7     android:text="Button" />
8   <TextView
9     android:text="TextView" />
10   ~~~ 중간 생략 ~~~
11 </LinearLayout>
```

오른쪽 하단으로 배치



레이아웃 개요 - 기본 리니어레이아웃



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="fill_parent"
    android:layout_height="fill_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!

버튼1



레이아웃 개요 - 기본 리니어레이아웃



❖ layout_gravity 속성

- layout_gravity는 자신의 위치를 부모(주로 레이아웃)의 어디쯤에 위치할 것인지를 결정

❖ 예제

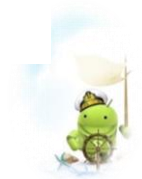
gravity는 자식(위젯)을 어디에 배치할 것인가?
layout_gravity는 부모(레이아웃)의 어디에 배치할 것인가?

layout_gravity 속성 XML 코드

```

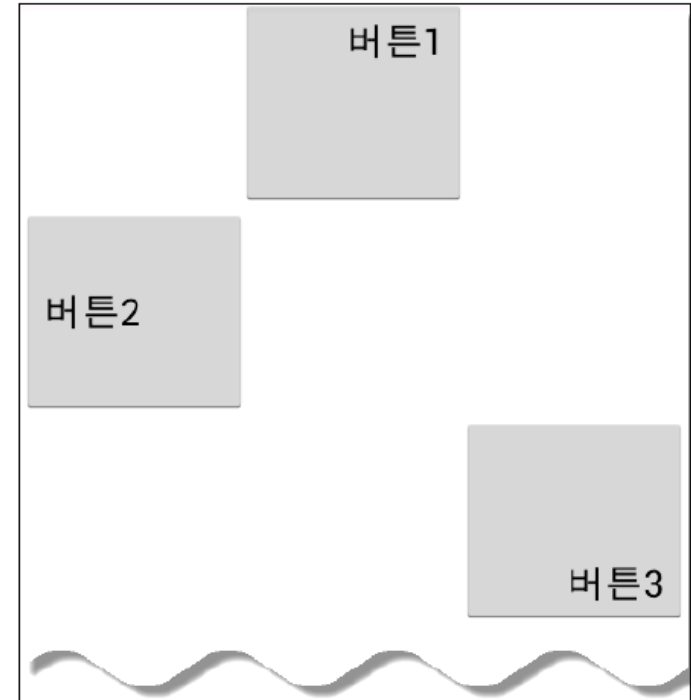
1 <LinearLayout
2   android:orientation="vertical" >
3   <Button
4     android:layout_gravity="right"
5     android:text="오른쪽" />
6   <Button
7     android:layout_gravity="center"
8     android:text="중앙" />
9   <Button
10    android:layout_gravity="left"
11    android:text="왼쪽" />
12 </LinearLayout>

```



직접 풀어보기

- 리니어레이아웃으로 다음 화면을 구성하는 XML을 작성하라.
 - ◎ 리니어레이아웃의 orientation은 "vertical"로 한다.
 - ◎ 버튼 3개를 생성하고, 버튼의 layout_width는 "110dp"로, layout_height는 "100dp"로 한다.
 - ◎ **[hint]** 버튼에 gravity와 layout_gravity를 모두 설정해야 한다.



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

```
<Button
    android:id="@+id/button1"
    android:layout_width="110dp"
    android:layout_height="100dp"
    android:layout_gravity="center"
    android:gravity="top|right"
    android:text="버튼1" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="110dp"
    android:layout_height="100dp"
    android:layout_gravity="left"
    android:gravity="center|left"
    android:text="버튼2" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="110dp"
    android:layout_height="100dp"
    android:layout_gravity="right"
    android:gravity="bottom|right"
    android:text="버튼3" />
```

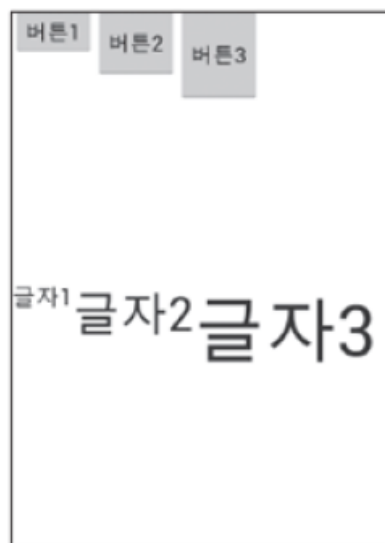
```
</LinearLayout>
```

레이아웃 개요 - 기본 리니어레이아웃

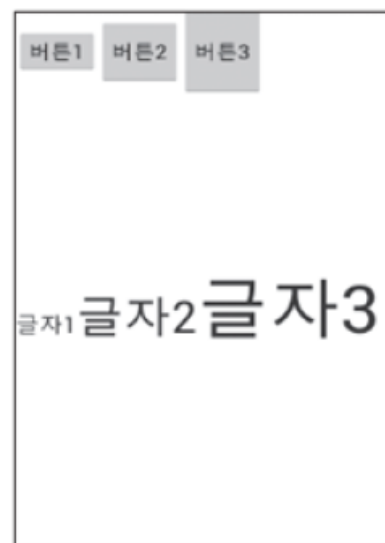
❖ **baselineAligned** 속성

- 크기가 다른 위젯들의 정렬을 보기 좋게 설정
- 디폴트 값은 "true"

❖ 예제



(a) false로 지정

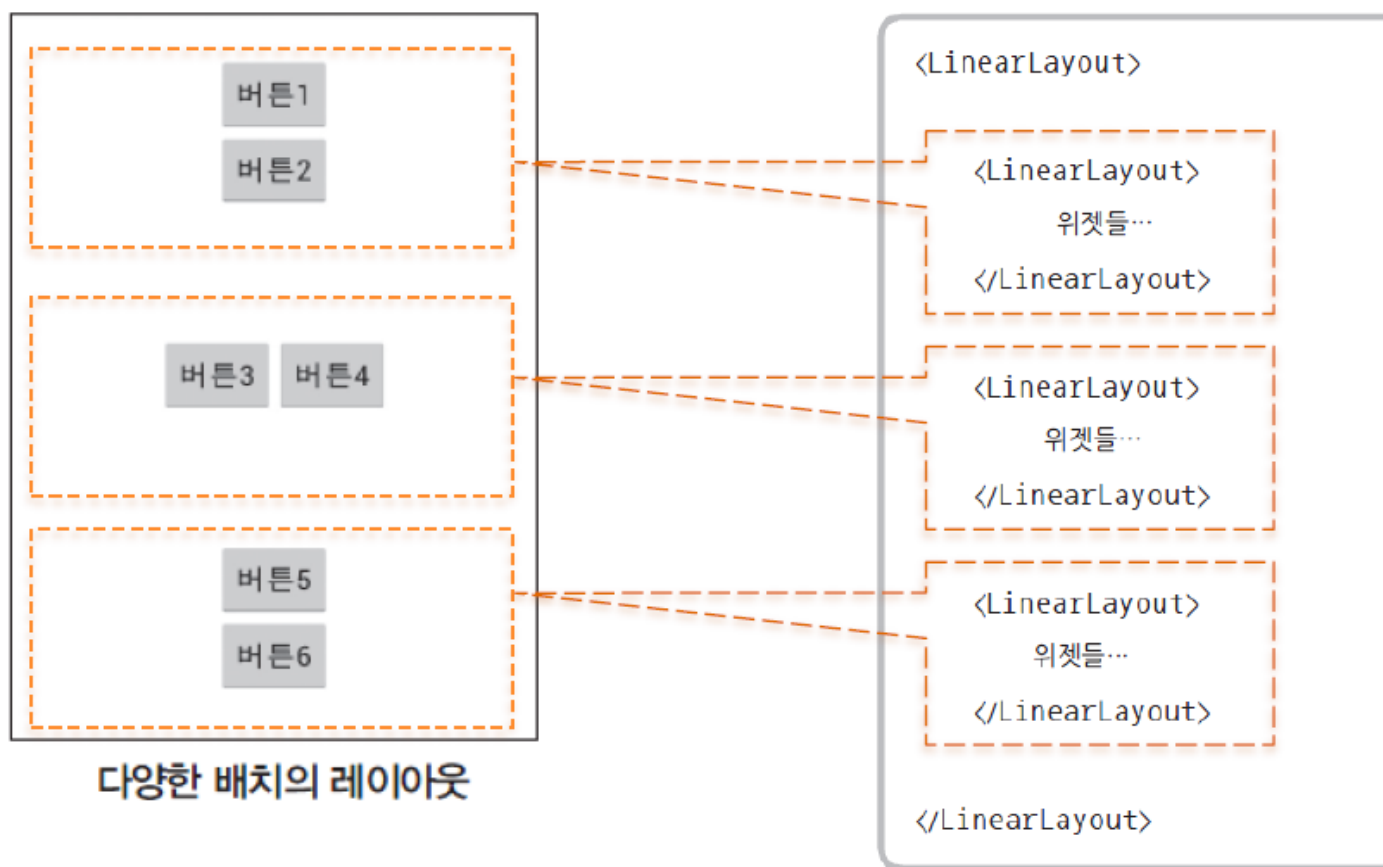


(b) true로 지정하거나 생략

baselineAligned 속성

레이아웃 개요 - 중복 리니어레이아웃

❖ 중복 리니어레이아웃 기본 형태



레이아웃 개요 - 중복 리니어레이아웃

○ layout_weight 속성

- 선형 레이아웃의 자식뷰에 **가중치**를 부여할 수 있다.
- 가중치가 높으면 자식 뷰가 부모 뷰 안의 비어 있는 공간으로 확장할 수 있다.
- 리니어레이아웃을 여러 개 사용하면, 각 레이아웃의 크기를 지정
- 대개 각 레이아웃을 전체 화면에서 몇 %로 지정하는 것이 일반적
- 자식뷰의 가중치를 지정하지 않으면 0값을 가진 것으로 확장되지 않는다.

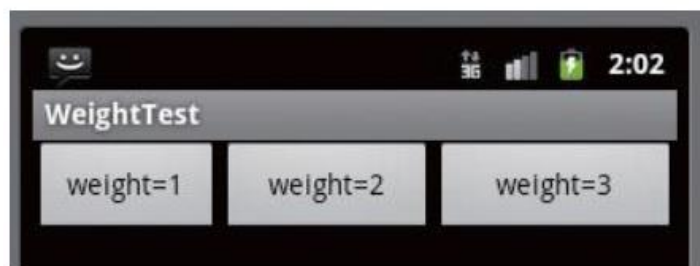
예)

선형레이아웃의 자식 뷰들의 가중치가 1:2:3이면,
남아 있는 공간의 1/6, 2/6, 3/6을 각각 할당

레이아웃 개요 - 중복 리니어레이아웃

○ 가중치

➤ 1:2:3



➤ 0:1:1



○ 예제 1 - 오류

세 개의 레이아웃으로 구분한 XML 코드

```

1 <LinearLayout
2   android:layout_width="match_parent"
3   android:layout_height="match_parent"
4   android:orientation="vertical" >
5   <LinearLayout
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:gravity="center"
9     android:orientation="vertical" >
10    <Button
11      android:text="버튼1" />
12    <Button
13      android:text="버튼2" />
14  </LinearLayout>

```

```

15  <LinearLayout
16    android:layout_width="match_parent"
17    android:layout_height="match_parent"
18    android:background="#00FF00"
19    android:gravity="center"
20    android:orientation="horizontal" >
21    <Button
22      android:text="버튼3" />
23    <Button
24      android:text="버튼4" />
25  </LinearLayout>
26  <LinearLayout
27    android:layout_width="match_parent"
28    android:layout_height="match_parent"
29    android:background="#0000FF"
30    android:gravity="center"
31    android:orientation="vertical" >
32    <Button
33      android:text="버튼5" />
34    <Button
35      android:text="버튼6" />
36  </LinearLayout>
37 </LinearLayout>

```

버튼1

버튼2



○ 예제 2(오류)와 예제 3(정상)

수정된 부분

android:layout_height = "wrap_content"

```
*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   <LinearLayout
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:gravity="center"
10    android:orientation="vertical" >
11     <Button
12       android:layout_width="wrap_content"
13       android:layout_height="wrap_content"
14       android:text="버튼1" />
15     <Button
16       android:layout_width="wrap_content"
17       android:layout_height="wrap_content"
18       android:text="버튼2" />
19   </LinearLayout>
```

체크



```
*activity_main.xml ✕
19     </LinearLayout>
20     <LinearLayout
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content"
23         android:background="#00FF00"
24         android:gravity="center"
25         android:orientation="horizontal" >
26         <Button
27             android:layout_width="wrap_content"
28             android:layout_height="wrap_content"
29             android:text="버튼3" />
30         <Button
31             android:layout_width="wrap_content"
32             android:layout_height="wrap_content"
33             android:text="버튼4" />
34     </LinearLayout>
35     <LinearLayout
36         android:layout_width="match_parent"
37         android:layout_height="wrap_content"
38         android:background="#0000FF"
39         android:gravity="center"
40         android:orientation="vertical" >
41         <Button
42             android:layout_width="wrap_content"
43             android:layout_height="wrap_content"
44             android:text="버튼5" />
45         <Button
46             android:layout_width="wrap_content"
47             android:layout_height="wrap_content"
48             android:text="버튼6" />
49     </LinearLayout>
50 </LinearLayout>
```

체크

체크



1:1:1 비율로 설정

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   <LinearLayout
7       android:layout_width="match_parent"
8       android:layout_height="match_parent"
9       android:layout_weight="1"
10      android:gravity="center"
11      android:orientation="vertical" >
12       <Button
13           android:layout_width="wrap_content"
14           android:layout_height="wrap_content"
15           android:text="버튼1" />
16       <Button
17           android:layout_width="wrap_content"
18           android:layout_height="wrap_content"
19           android:text="버튼2" />
20   </LinearLayout>
```

체크

버튼1

버튼2

버튼3

버튼4

버튼5

버튼6

activity_main.xml

```

20     </LinearLayout>
21     <LinearLayout
22         android:layout_width="match_parent"
23         android:layout_height="match_parent"
24         android:layout_weight="1"
25         android:background="#00FF00"
26         android:gravity="center"
27         android:orientation="horizontal" >
28         <Button
29             android:layout_width="wrap_content"
30             android:layout_height="wrap_content"
31             android:text="버튼3" />
32         <Button
33             android:layout_width="wrap_content"
34             android:layout_height="wrap_content"
35             android:text="버튼4" />
36     </LinearLayout>
37     <LinearLayout
38         android:layout_width="match_parent"
39         android:layout_height="match_parent"
40         android:layout_weight="1"
41         android:background="#0000FF"
42         android:gravity="center"
43         android:orientation="vertical" >
44         <Button
45             android:layout_width="wrap_content"
46             android:layout_height="wrap_content"
47             android:text="버튼5" />
48         <Button
49             android:layout_width="wrap_content"
50             android:layout_height="wrap_content"
51             android:text="버튼6" />
52     </LinearLayout>
53 </LinearLayout>

```

체크

체크



1:3:1 비율로 변경



activity_main.xml

```

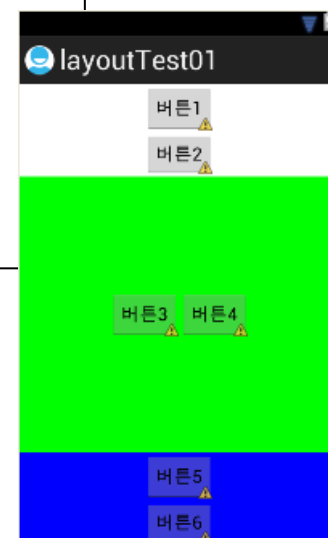
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6   <LinearLayout
7       android:layout_width="match_parent"
8       android:layout_height="0"
9       android:layout_weight="1"
10      android:gravity="center"
11      android:orientation="vertical" >
12       <Button
13           android:layout_width="wrap_content"
14           android:layout_height="wrap_content"
15           android:text="버튼1" />
16       <Button
17           android:layout_width="wrap_content"
18           android:layout_height="wrap_content"
19           android:text="버튼2" />
20   </LinearLayout>

```

수정

Android:layout_height="0"

Android:layout_weight="1"



activity_main.xml X

```
21 <LinearLayout
22     android:layout_width="match_parent"
23     android:layout_height="0"
24     android:layout_weight="3"
25     android:background="#00FF00"
26     android:gravity="center"
27     android:orientation="horizontal" >
28     <Button
29         android:layout_width="wrap_content"
30         android:layout_height="wrap_content"
31         android:text="버튼3" />
32     <Button
33         android:layout_width="wrap_content"
34         android:layout_height="wrap_content"
35         android:text="버튼4" />
36 </LinearLayout>
37 <LinearLayout
38     android:layout_width="match_parent"
39     android:layout_height="0"
40     android:layout_weight="1"
41     android:background="#0000FF"
42     android:gravity="center"
43     android:orientation="vertical" >
44     <Button
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:text="버튼5" />
48     <Button
49         android:layout_width="wrap_content"
50         android:layout_height="wrap_content"
51         android:text="버튼6" />
52 </LinearLayout>
53 </LinearLayout>
```

수정

Android:layout_height="0"
Android:layout_weight="3"

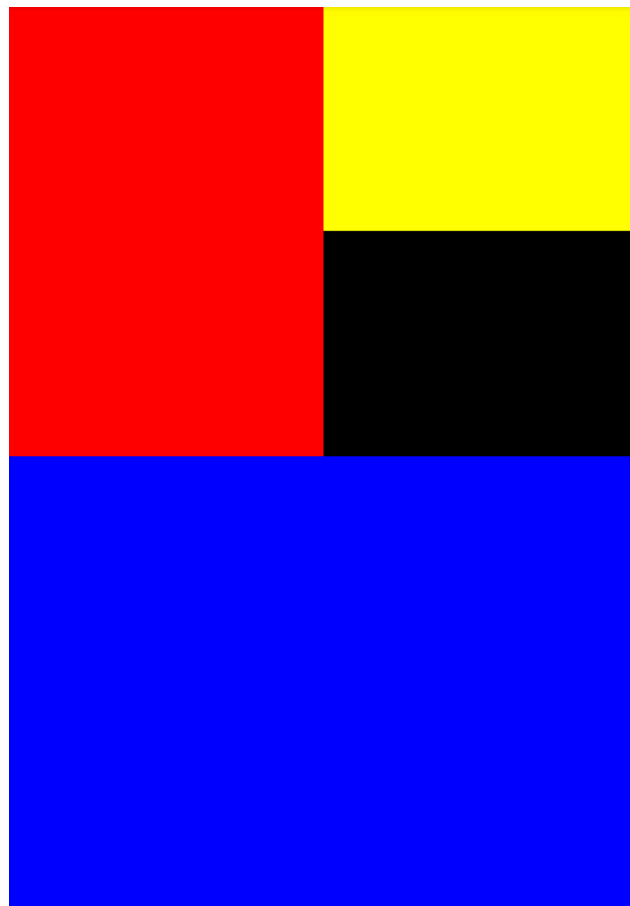
수정

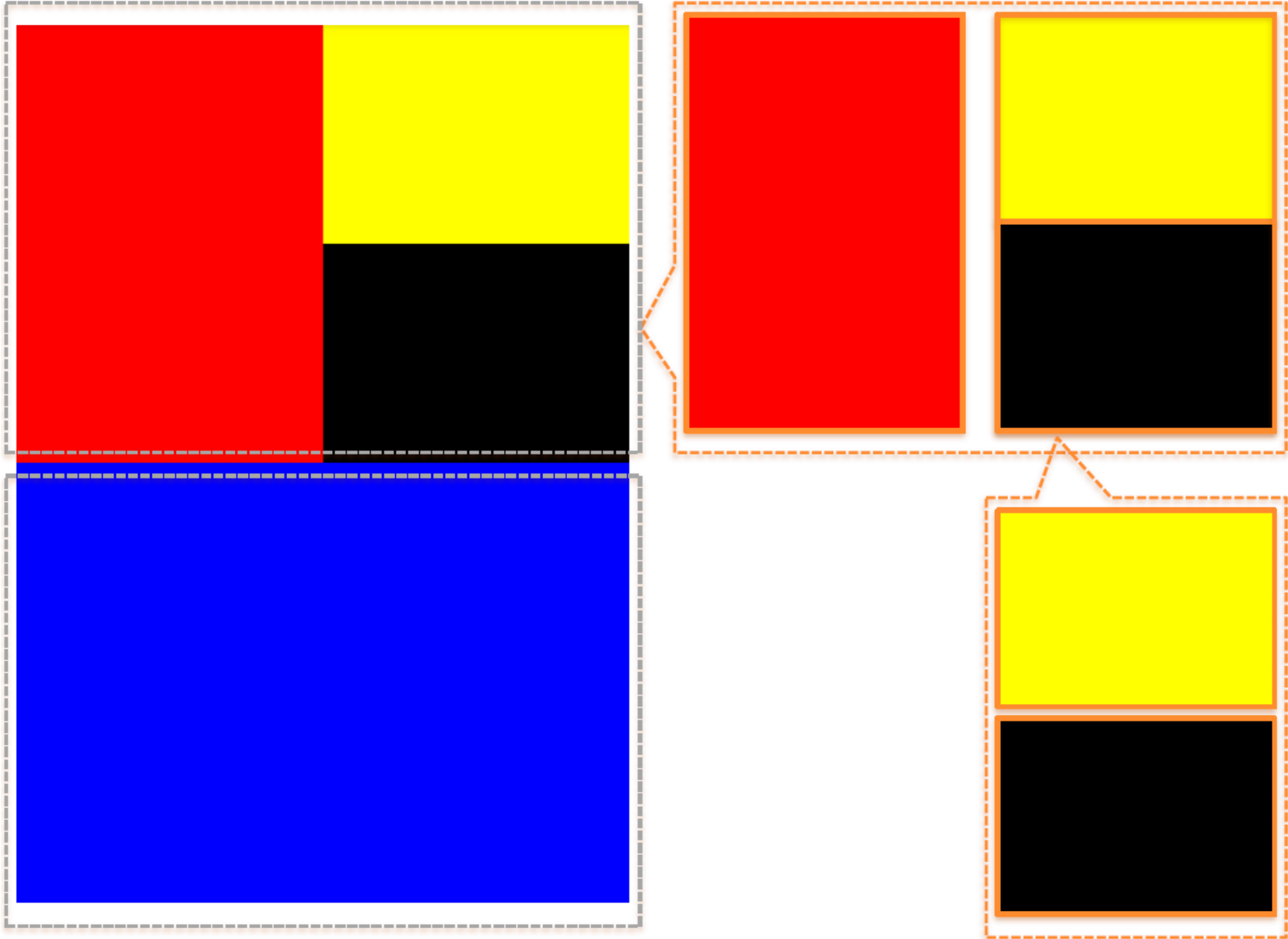
Android:layout_height="0"
Android:layout_weight="1"

직접 풀어보기



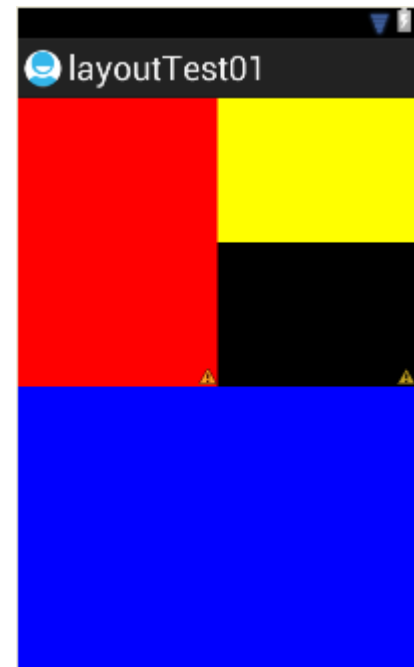
- 리니어레이아웃으로 다음 화면을 구성하는 XML을 작성하라.
 - ⦿ 레이아웃의 색상은 구분되어 보이도록 서로 다른 색으로 한다.
 - ⦿ hint : 레이아웃 안에 다시 레이아웃을 여러 번 중첩





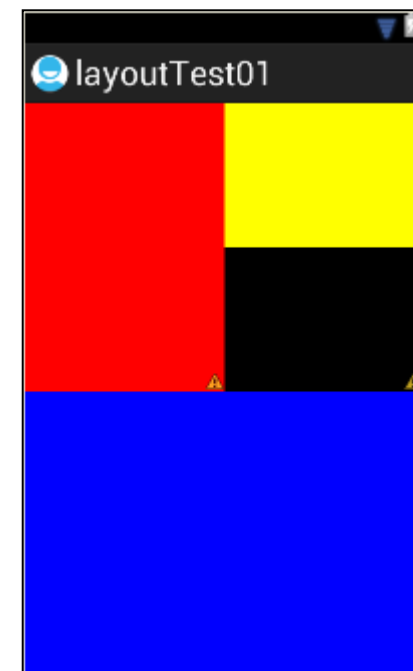
activity_main.xml

```
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     <LinearLayout
7         android:layout_width="match_parent"
8         android:layout_height="match_parent"
9         android:layout_weight="1"
10        android:orientation="horizontal" >
11        <LinearLayout
12            android:layout_width="match_parent"
13            android:layout_height="match_parent"
14            android:layout_weight="1"
15            android:background="#ff0000"
16            android:orientation="vertical" >
17        </LinearLayout>
```



activity_main.xml

```
17     </LinearLayout>
18     <LinearLayout
19         android:layout_width="match_parent"
20         android:layout_height="match_parent"
21         android:layout_weight="1"
22         android:background="#00ff00"
23         android:orientation="vertical" >
24         <LinearLayout
25             android:layout_width="match_parent"
26             android:layout_height="match_parent"
27             android:layout_weight="1"
28             android:background="#ffff00"
29             android:orientation="vertical" >
30             </LinearLayout>
31             <LinearLayout
32                 android:layout_width="match_parent"
33                 android:layout_height="match_parent"
34                 android:layout_weight="1"
35                 android:background="#000000"
36                 android:orientation="vertical" >
37                 </LinearLayout>
38             </LinearLayout>
39         </LinearLayout>
40         <LinearLayout
41             android:layout_width="match_parent"
42             android:layout_height="match_parent"
43             android:layout_weight="1"
44             android:background="#0000ff"
45             android:orientation="vertical" >
46         </LinearLayout>
47     </LinearLayout>
```

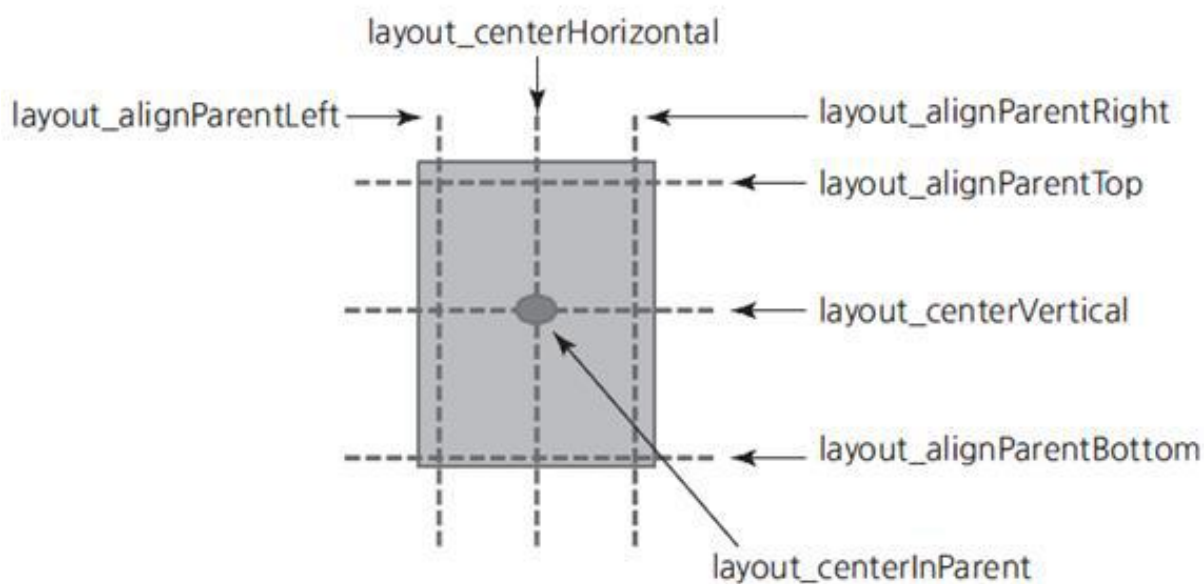


릴레이티브 레이아웃 (Relative Layout)

❖ 개념

- 렐러티브레이아웃(RelativeLayout)은 상대레이아웃이라고도 부르는데, 이름처럼 레이아웃 내부에 포함된 위젯들을 상대적인 위치로 배치

❖ 렐러티브레이아웃의 상하좌우에 배치



부모 레이아웃에서의 위치를 적용할 때 속성



릴레이티브 레이아웃 (Relative Layout)



속성	설명
layout_above	만약 true 이면 현재 뷰의 하단을 기준 뷰의 상단에 일치시킨다.
layout_alignBaseline	현재 뷰의 베이스라인을 기준 뷰의 베이스라인 위에 일치시킨다.
layout_alignBottom	현재 뷰의 하단을 기준 뷰의 하단과 일치시킨다.
layout_alignLeft	현재 뷰의 좌측단을 기준 뷰의 좌측단과 일치시킨다.
layout_alignParentBottom	현재 뷰의 하단을 부모의 하단과 일치시킨다.
layout_alignParentLeft	현재 뷰의 좌측단을 부모의 좌측단과 일치시킨다.
layout_alignParentRight	현재 뷰의 우측단을 부모의 우측단과 일치시킨다.
layout_alignParentTop	현재 뷰의 상단을 부모의 상단과 일치시킨다.
layout_alignRight	현재 뷰의 우측단을 기준 뷰의 우측단과 일치시킨다.
layout_alignTop	현재 뷰의 상단을 기준 뷰의 상단과 일치시킨다.
layout_alignWithParentIfMissing	기준 뷰가 없으면 보모를 대신 사용한다.
layout_below	현재 뷰의 상단을 기준 뷰의 하단에 위치시킨다.
layout_centerHorizontal	수평으로 현재 뷰의 중심을 보모와 일치시킨다.
layout_centerInparent	부모의 중심점에 현재 뷰를 위치시킨다.
layout_centerVertical	수직으로 현재 뷰의 중심을 부모와 일치시킨다.
layout_toLeftOf	현재 뷰의 우측단을 기준 뷰의 좌측에 위치시킨다.
layout_toRightOf	현재 뷰의 좌측단을 기준 뷰의 우측단에 위치시킨다.



릴레이티브 레이아웃 (Relative Layout)



<RelativeLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
```

<TextView

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

<EditText

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

address 아래에 배치

<Button

```
android:id="@+id/cancel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/input" />
```

input 아래에 배치



릴레이티브 레이아웃 (Relative Layout)

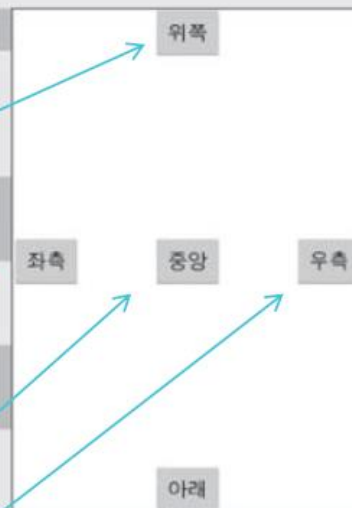
❖ 예제 1

릴레이티브레이아웃 XML 코드 1

```

1 <RelativeLayout xmlns:android="http://www."
2   android:layout_width="match_parent"
3   android:layout_height="match_parent" >
4   <Button
5     android:layout_alignParentTop="true"
6     android:layout_centerHorizontal="true"
7     android:text="위쪽" />
8   <Button
9     android:layout_alignParentLeft="true"
10    android:layout_centerVertical="true"
11    android:text="좌측" />
12  <Button
13    android:layout_centerInParent="true"
14    android:text="중앙" />
15  <Button
16    android:layout_alignParentRight="true"
17    android:layout_centerVertical="true"
18    android:text="우측" />
19  <Button
20    android:layout_alignParentBottom="true"
21    android:layout_centerHorizontal="true"
22    android:text="아래" />
23 </RelativeLayout>

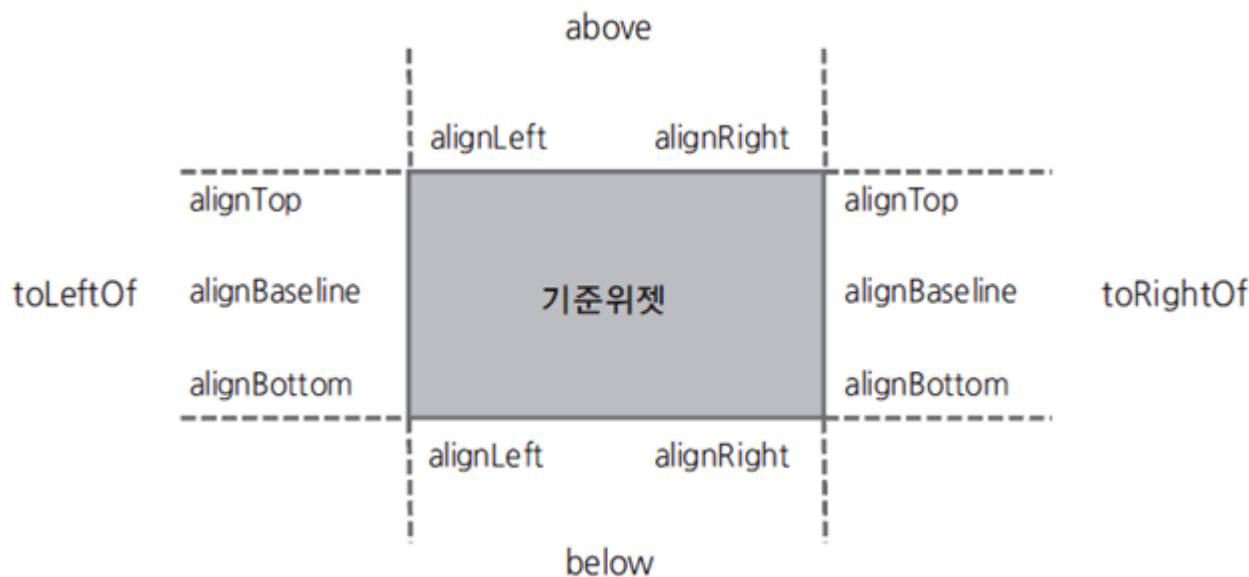
```



릴레이티브 레이아웃 (Relative Layout)

○ 다른 위젯의 상대위치에 배치

- ◎ 각 속성의 값은 다른 위젯의 id를 지정하면 되는데, "@+id/기준 위젯_아이디" 형식으로 사용
- ◎ 아래 그림을 간략하게 표현하기 위해서 나타난 속성 이름에서 앞의 "layout_"은 생략하였음



다른 위젯에서 상대적인 위치를 적용할 때의 속성

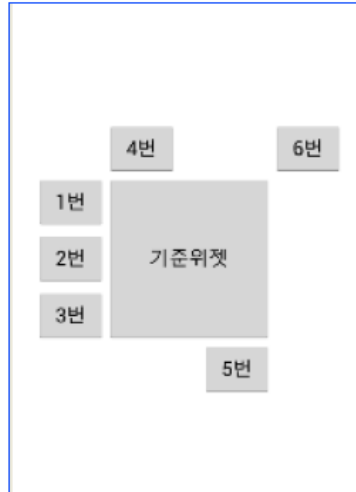


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

```
<Button
    android:id="@+id/baseBtn"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="기준 위젯" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/baseBtn"
    android:layout_toLeftOf="@+id/baseBtn"
    android:text="1번" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/baseBtn"
    android:layout_toLeftOf="@+id/baseBtn"
    android:text="2번" />
```



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

```
    android:layout_alignBottom="@+id/baseBtn"
    android:layout_toLeftOf="@+id/baseBtn"
    android:text="3번" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/baseBtn"
    android:layout_alignLeft="@+id/baseBtn"
    android:text="4번" />
```

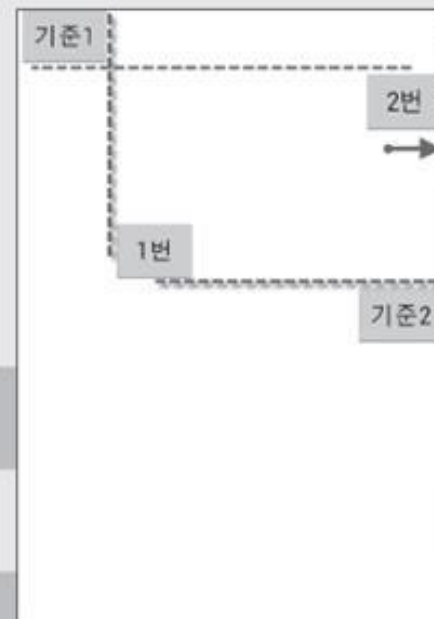
```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/baseBtn"
    android:layout_below="@+id/baseBtn"
    android:text="5번" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/baseBtn"
    android:layout_toRightOf="@+id/baseBtn"
    android:text="6번" />
```

```
</RelativeLayout>
```

렐러티브레이아웃 속성을 조합한 XML 코드

```
1 <RelativeLayout xmlns:android="http://~"
2   android:layout_width="match_parent"
3   android:layout_height="match_parent" >
4   <Button
5       android:id="@+id/baseBtn1"
6       android:layout_alignParentLeft="true"
7       android:layout_alignParentTop="true"
8       android:text="기준1" />
9   <Button
10      android:id="@+id/baseBtn2"
11      android:layout_alignParentRight="true"
12      android:layout_centerVertical="true"
13      android:text="기준2" />
14   <Button
15       android:layout_above="@+id/baseBtn2"
16       android:layout_toRightOf="@+id/baseBtn1"
17       android:text="1번" />
18   <Button
19       android:layout_alignParentRight="true"
20       android:layout_below="@+id/baseBtn1"
21       android:text="2번" />
22 </RelativeLayout>
```



직접 풀어보기

- 다음 화면의 XML 코드를 중복 **리니어레이아웃**과 **렐러티브레이아웃**으로 각각 작성하라.
 - ◎ 텍스트뷰 1개, 에디트텍스트 1개, 버튼 2개로 구성한다.

전화번호 000-0000-0000

입력 취소

전화번호: 부모에 왼쪽편으로 배치
000-0000-0000: "전화번호"에 오른쪽에 배치
취소: 부모에 오른쪽에 배치하고, "000-0000-0000"의 밑에 배치
입력: "취소"와 라인을 맞춰주고, "취소"의 왼쪽에 배치



```
NewFile.xml
1 <RelativeLayout 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
6   <TextView
7       android:id="@+id/textView1"
8       android:layout_width="wrap_content"
9       android:layout_height="wrap_content"
10      android:layout_alignParentLeft="true"
11      android:text="전화번호"
12      android:textSize="25sp" />
13
14   <EditText
15       android:id="@+id/editText1"
16       android:layout_width="match_parent"
17       android:layout_height="wrap_content"
18       android:layout_toRightOf="@+id/textView1"
19       android:hint="000-0000-0000" />
```

전화번호 000-0000-0000

입력

취소

NewFile.xml

```
20
21 <Button
22     android:id="@+id/btnCancel"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:layout_alignParentRight="true"
26     android:layout_below="@+id/editText1"
27     android:text="취소" />
28
29 <Button
30     android:id="@+id/btnOK"
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content"
33     android:layout_alignBaseline="@+id/btnCancel"
34     android:layout_toLeftOf="@+id/btnCancel"
35     android:text="입력" />
36
37 </RelativeLayout>
```

전화번호 000-0000-0000

입력

취소

```
NewFile.xml X
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   <LinearLayout
8       android:layout_width="match_parent"
9       android:layout_height="wrap_content"
10      android:orientation="horizontal" >
11
12      <TextView
13          android:id="@+id/textView1"
14          android:layout_width="wrap_content"
15          android:layout_height="wrap_content"
16          android:text="전화번호"
17          android:textSize="25sp" />
18
```

전화번호 000-0000-0000

입력

취소

NewFile.xml

```
18
19     <EditText
20         android:id="@+id/editText1"
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content"
23         android:hint="000-0000-0000" >
24     </EditText>
25 </LinearLayout>
26
27 <LinearLayout
28     android:layout_width="match_parent"
29     android:layout_height="wrap_content"
30     android:gravity="right"
31     android:orientation="horizontal" >
32
33     <Button
34         android:id="@+id/btnOK"
35         android:layout_width="wrap_content"
36         android:layout_height="wrap_content"
37         android:text="입력" />
38
39     <Button
40         android:id="@+id/btnCancel"
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:text="취소" />
44 </LinearLayout>
45
46 </LinearLayout>
```

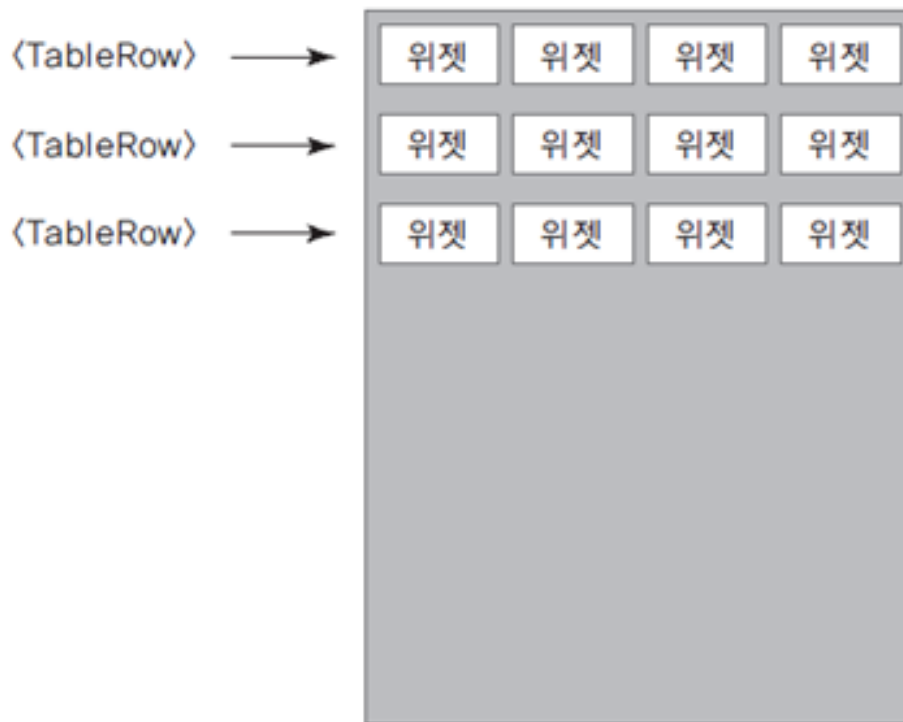
전화번호 000-0000-0000

입력

취소

테이블 레이아웃(TableLayout)

- 표 형태로 위젯을 배치할 때 주로 사용
- <TableRow>와 함께 사용되는데, <TableRow>의 개수가 바로 행의 개수가 된다. 열의 개수는 <TableRow>안에 포함된 위젯의 개수가 열의 개수로 결정된다.
- 행 4열의 테이블레이아웃은 다음과 같다.



테이블레이아웃



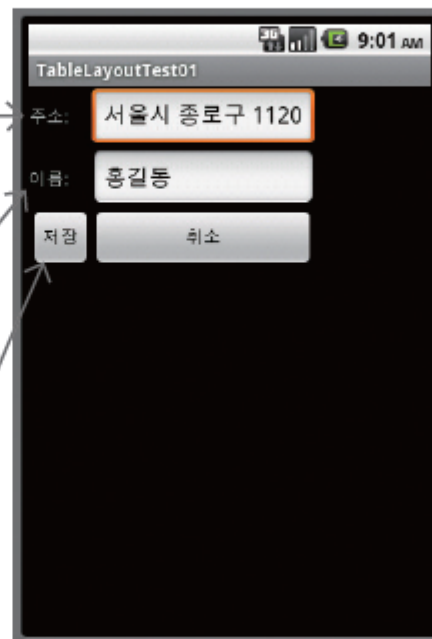
```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

테이블의 하나의 행

```
<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>
```

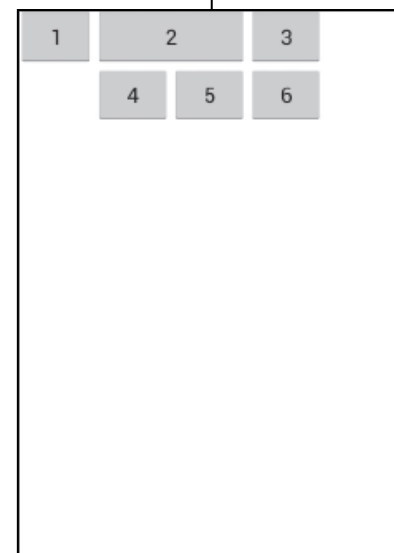


테이블 레이아웃

○ 속성

- ◎ **layout_span:** 열을 합쳐서 표시하라는 의미
(layout_span = '2' : 현재 셀부터 2개의 셀을 합쳐서 표시)
- ◎ **layout_column:** 지정된 열에 현재 위젯을 표시하라는 의미
- ◎ **stretchColumns:** <TableLayout> 자체에 설정하는 속성으로 지정된 열의 폭을 늘리라는 의미
 - ◎ stretchColumns="*"로 하면 각 셀을 모두 같은 크기로 확장해서 전체 화면이 꽉 차는 효과를 갖는다.
 - ◎ 열 번호는 0번부터

```
NewFile.xml
1 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:id="@+id/tableLayout1"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent" >
6
7   <TableRow>
8       <Button
9           android:layout_width="wrap_content"
10          android:layout_height="wrap_content"
11          android:text="1" />
12       <Button
13           android:layout_width="wrap_content"
14           android:layout_height="wrap_content"
15           android:layout_span="2"
16           android:text="2" />
17       <Button
18           android:layout_width="match_parent"
19           android:layout_height="wrap_content"
20           android:text="3" />
21   </TableRow>
```



1행에 위젯의 개수가 3개 배치 되어 3열을 구성한다. 그 중에 "2"를 현재 셀부터 2개의 셀을 합쳐 표시

```
NewFile.xml X
22
23     <TableRow>
24         <Button
25             android:layout_width="wrap_content"
26             android:layout_height="wrap_content"
27             android:layout_column="1"
28             android:text="4" />
29         <Button
30             android:layout_width="wrap_content"
31             android:layout_height="wrap_content"
32             android:text="5" />
33         <Button
34             android:layout_width="wrap_content"
35             android:layout_height="wrap_content"
36             android:text="6" />
37     </TableRow>
38 </TableLayout>
```

1	2	3	
	4	5	6

1열에 표시하라.

1	2	3
4	5	6

만약, `layout_column="1"`이 없다면, 0열에 표시 되어 다음과 같이 표시된다.

테이블 레이아웃



[실습] 테이블레이아웃을 활용한 계산기

- <실습>의 초간단 계산기를 좀 더 개선시켜서, 테이블레이아웃을 활용하여 숫자버튼까지 있는 계산기를 만들자.



화면 디자인



- TableLayout 1개와 TableRow 9개로 구성
- EditText 2개, 숫자버튼 10개, 연산 Button 4개, TextView 1개를 생성
- 각 연산 버튼 위젯에는 layout_margin을 적절히 지정
- 결과를 보여줄 TextView는 색상을 빨간색으로, 글자크기는 20dp 정도로 지정
- 각 위젯의 id는 위에서부터 Edit1, Edit2, BtnNum0~9, BtnAdd, BtnSub, BtnMul, BtnDiv, TextResult로 지정

숫자1 입력

숫자2 입력

0

1

2

3

4

5

6

7

8

9

더하기

빼기

곱하기

나누기

계산 결과 :



```

main.xml
1 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:id="@+id/tableLayout1"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent" >
6
7   <TableRow>
8       <EditText
9           android:id="@+id/Edit1"
10          android:layout_span="5"
11          android:hint="숫자1 입력" />
12   </TableRow>
13   <TableRow>
14       <EditText
15           android:id="@+id/Edit2"
16           android:layout_span="5"
17           android:hint="숫자2 입력" />
18   </TableRow>

```

5열 크기 만큼 표시

<TableRow>
</TableRow>

1 행

숫자1 입력				
숫자2 입력				
0	1	2	3	4
5	6	7	8	9
더하기				
빼기				
곱하기				
나누기				
계산 결과 :				

main.xml

```

19 <TableRow>
20     <Button
21         android:id="@+id/BtnNum0"
22         android:text="0" />
23     <Button
24         android:id="@+id/BtnNum1"
25         android:text="1" />
26     <Button
27         android:id="@+id/BtnNum2"
28         android:text="2" />
29     <Button
30         android:id="@+id/BtnNum3"
31         android:text="3" />
32     <Button
33         android:id="@+id/BtnNum4"
34         android:text="4" />
35 </TableRow>
36 <TableRow>
37     <Button
38         android:id="@+id/BtnNum5"
39         android:text="5" />
40     <Button
41         android:id="@+id/BtnNum6"
42         android:text="6" />
43     <Button
44         android:id="@+id/BtnNum7"
45         android:text="7" />
46     <Button
47         android:id="@+id/BtnNum8"
48         android:text="8" />
49     <Button
50         android:id="@+id/BtnNum9"
51         android:text="9" />
52 </TableRow>

```

main.xml

```

53 <TableRow>
54     <Button
55         android:id="@+id/BtnAdd"
56         android:layout_span="5"
57         android:text="더하기" />
58 </TableRow>
59 <TableRow>
60
61     <Button
62         android:id="@+id/BtnSub"
63         android:layout_span="5"
64         android:text="빼기" />
65
66 </TableRow>
67 <TableRow>
68     <Button
69         android:id="@+id/BtnMul"
70         android:layout_span="5"
71         android:text="곱하기" />
72 </TableRow>
73 <TableRow>
74     <Button
75         android:id="@+id/BtnDiv"
76         android:layout_span="5"
77         android:text="나누기" />
78 </TableRow>
79 <TableRow>
80     <TextView
81         android:id="@+id/TextResult"
82         android:layout_span="5"
83         android:text="계산 결과: "
84         android:textColor="#FF0000"
85         android:textSize="20dp" />
86 </TableRow>
87 </TableLayout>

```

숫자1 입력

숫자2 입력

0	1	2	3	4
5	6	7	8	9

더하기

빼기

곱하기

나누기

계산 결과:

Project5_2Activity.java ✕

```
1 package kr.ac.project5_2;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.MotionEvent;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.TextView;
10 import android.widget.Toast;
11
12 public class Project5_2Activity extends Activity {
13     EditText edit1, edit2;
14     Button btnAdd, btnSub, btnMul, btnDiv;
15     TextView textResult;
16     String num1, num2;
17     Integer result;
18
19     // 10개 숫자 버튼 배열
20     Button[] numButtons = new Button[10];
21     // 10개 숫자 버튼의 id 값 배열
22     Integer[] numBtnIDs = { R.id.BtnNum0, R.id.BtnNum1, R.id.BtnNum2,
23         R.id.BtnNum3, R.id.BtnNum4, R.id.BtnNum5, R.id.BtnNum6,
24         R.id.BtnNum7, R.id.BtnNum8, R.id.BtnNum9 };
25     int i; // 증가값 용도
26 }
```

에디트텍스트에 입력할 문자열 저장 변수

연산 결과 저장변수

```

27 @Override
28 public void onCreate(Bundle savedInstanceState) {
29     super.onCreate(savedInstanceState);
30     setContentView(R.layout.main);
31
32     setTitle("테이블레이아웃 계산기");
33
34     edit1 = (EditText) findViewById(R.id.Edit1);
35     edit2 = (EditText) findViewById(R.id.Edit2);
36
37     btnAdd = (Button) findViewById(R.id.BtnAdd);
38     btnSub = (Button) findViewById(R.id.BtnSub);
39     btnMul = (Button) findViewById(R.id.BtnMul);
40     btnDiv = (Button) findViewById(R.id.BtnDiv);
41
42     textResult = (TextView) findViewById(R.id.TextResult);
43
44     btnAdd.setOnClickListener(new View.OnClickListener() {
45         public boolean onTouch(View arg0, MotionEvent arg1) {
46             num1 = edit1.getText().toString();
47             num2 = edit2.getText().toString();
48             result = Integer.parseInt(num1) + Integer.parseInt(num2);
49             textResult.setText("계산 결과 : " + result.toString());
50             return false;
51         }
52     });

```



"+"더하기

에디트텍스트의
문자열을 받는다.

연산 결과 출력

```
53
54 btnSub.setOnClickListener(new View.OnClickListener() {
55     public boolean onTouch(View arg0, MotionEvent arg1) {
56         num1 = edit1.getText().toString();
57         num2 = edit2.getText().toString();
58         result = Integer.parseInt(num1) - Integer.parseInt(num2);
59         textResult.setText("계산 결과 : " + result.toString());
60         return false;
61     }
62 });
63
64 btnMul.setOnClickListener(new View.OnClickListener() {
65     public boolean onTouch(View arg0, MotionEvent arg1) {
66         num1 = edit1.getText().toString();
67         num2 = edit2.getText().toString();
68         result = Integer.parseInt(num1) * Integer.parseInt(num2);
69         textResult.setText("계산 결과 : " + result.toString());
70         return false;
71     }
72 });
73
74 btnDiv.setOnClickListener(new View.OnClickListener() {
75     public boolean onTouch(View arg0, MotionEvent arg1) {
76         num1 = edit1.getText().toString();
77         num2 = edit2.getText().toString();
78         result = Integer.parseInt(num1) / Integer.parseInt(num2);
79         textResult.setText("계산 결과 : " + result.toString());
80         return false;
81     }
82 });
```

"- 빼기

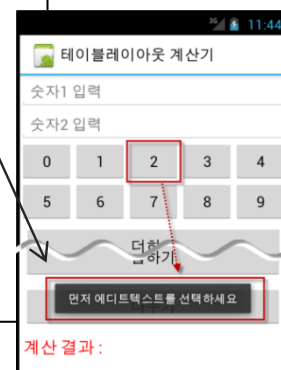
"* 곱

"/ 나누기

```

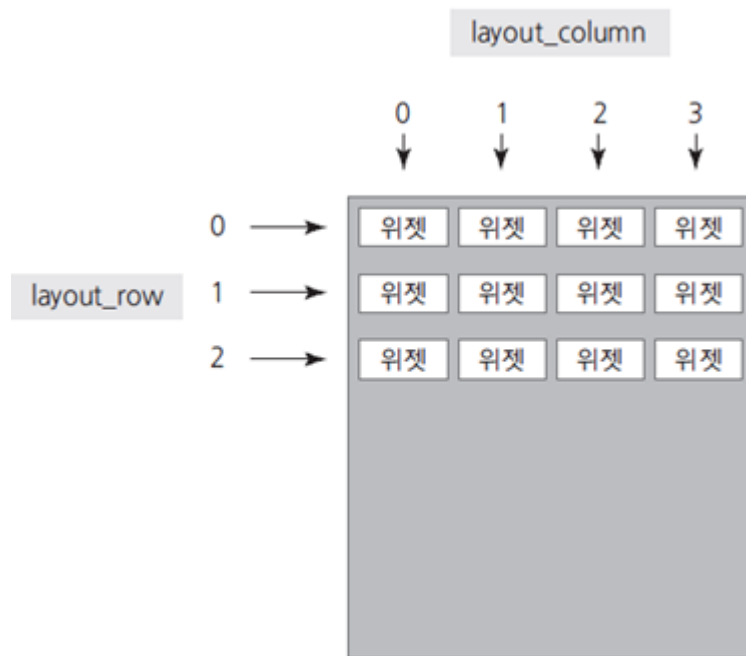
83
84 // 숫자 버튼 10개를 대입
85 for (i = 0; i < numBtnIDs.length; i++) {
86     numButtons[i] = (Button) findViewById(numBtnIDs[i]);
87 }
88 // 숫자 버튼 10개에 대해서 클릭이벤트 처리
89 for (i = 0; i < numBtnIDs.length; i++) {
90
91     final int index; // 주의! 꼭 필요함..
92     index = i;
93
94     numButtons[index].setOnClickListener(new View.OnClickListener() {
95         public void onClick(View v) {
96             // 포커스가 되어 있는 에디트텍스트에 숫자 추가
97             if (edit1.isFocused() == true) {
98                 num1 = edit1.getText().toString()
99                     + numButtons[index].getText().toString();
100                 edit1.setText(num1);
101             } else if (edit2.isFocused() == true) {
102                 num2 = edit2.getText().toString()
103                     + numButtons[index].getText().toString();
104                 edit2.setText(num2);
105             } else {
106                 Toast.makeText(getApplicationContext(),
107                     "먼저 에디트텍스트를 선택하세요", 0).show();
108             }
109         }
110     });
111 }
112 }
113 }

```



5. 그리드 레이아웃(Grid Layout)

- 테이블레이아웃에서 좀 어려웠던 행을 확장하는 기능도 간단하게 할 수 있어서 유연한 화면 구성에 적합
- 행, 열을 지정하는 것은 상당히 직관적
- Android 4.0 (아이스크림 샌드위치, API 14)부터 지원.



그리드레이아웃 개요

그리드 레이아웃

○ 속성

- ◎ <GridLayout> 자체에 자주 사용되는 속성
 - ⊙ rowCount : 행 개수
 - ⊙ columnCount : 열 개수
 - ⊙ orientation : 수평방향/수직방향 중 어디를 우선할 것인지를 결정
- ◎ 그리드레이아웃 안에 포함될 위젯에서 사용되는 속성
 - ⊙ layout_row : 자신이 위치할 행 번호 (0번부터 카운트)
 - ⊙ layout_column : 자신이 위치할 열 번호 (0번부터 카운트)
 - ⊙ layout_rowSpan : 지정된 개수만큼 행을 확장
 - ⊙ layout_columnSpan : 지정된 개수만큼 열을 확장
 - ⊙ layout_gravity : 위젯을 확장된 셀에 꽂 채우는 효과 (fill, fill_vertical, fill_horizontal 등)

```

1 <GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="fill_parent"
4   android:layout_height="wrap_content"
5   android:columnCount="4"
6   android:rowCount="2" >
7
8   <Button
9       android:layout_column="0"
10      android:layout_row="0"
11      android:layout_gravity="fill_vertical"
12      android:layout_rowSpan="2"
13      android:text="1" />
14
15   <Button
16       android:layout_column="1"
17       android:layout_row="0"
18       android:layout_columnSpan="2"
19       android:layout_gravity="fill_horizontal"
20       android:text="2" />
21

```

행(4)과 열(2)의개수 지정

0행 0열에 위치

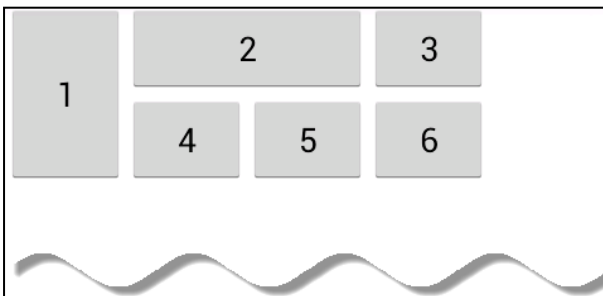
셀의 수직 방향으로 확장한 만큼
꼭 채워라

행의 확장(2행만큼)

0행 1열에 위치

열을 확장(2만큼)

셀을 수평 방향으로 확장한 만큼
확장

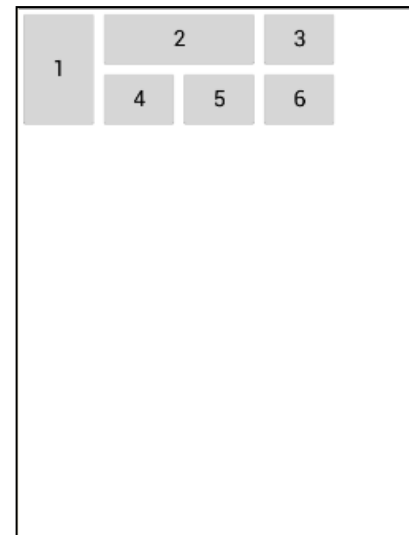



```
*main.xml x
22 <Button
23     android:layout_column="3"
24     android:layout_row="0"
25     android:text="3" />
26
27 <Button
28     android:layout_column="1"
29     android:layout_row="1"
30     android:text="4" />
31
32 <Button
33     android:layout_column="2"
34     android:layout_row="1"
35     android:text="5" />
36
37 <Button
38     android:layout_column="3"
39     android:layout_row="1"
40     android:text="6" />
41
42 </GridLayout>
```

0행 3열에 위치

1행1열에 위치

1행2열에 위치



직접 풀어보기

- 그리드레이아웃으로 변경해서 실행하라.
 - ⊙ hint : Java 코드는 변경할 필요가 없고, XML만 변경하면 된다. 또 XML 위젯의 id도 동일하게 사용한다.

숫자1 입력				
숫자2 입력				
0	1	2	3	4
5	6	7	8	9
더하기				
빼기				
곱하기				
나누기				
계산 결과 :				



*main.xml

```

1 <GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:id="@+id/tableLayout1"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:columnCount="5"
8     android:rowCount="9" >
9
10    <EditText
11        android:id="@+id/Edit1"
12        android:layout_columnSpan="5"
13        android:layout_gravity="fill_horizontal"
14        android:hint="숫자1 입력" />
15
16    <EditText
17        android:id="@+id/Edit2"
18        android:layout_columnSpan="5"
19        android:layout_gravity="fill_horizontal"
20        android:hint="숫자2 입력" />
21
22    <Button
23        android:id="@+id/BtnNum0"
24        android:layout_column="0"
25        android:layout_row="2"
26        android:text="0" />
27

```

9행 5열 그리드 생성

열을 확장(5열)

확장한 만큼(5) 수평으로 꽉 채움

숫자1 입력

숫자2 입력

0

1

2

3

4

5

6

7

8

9

더하기

빼기

곱하기

나누기

2행 0열에 위치

계산 결과 :

```

28 <Button
29     android:id="@+id/BtnNum1"
30     android:layout_column="1"
31     android:layout_row="2"
32     android:text="1" />
33
34 <Button
35     android:id="@+id/BtnNum2"
36     android:layout_column="2"
37     android:layout_row="2"
38     android:text="2" />
39
40 <Button
41     android:id="@+id/BtnNum3"
42     android:layout_column="3"
43     android:layout_row="2"
44     android:text="3" />
45
46 <Button
47     android:id="@+id/BtnNum4"
48     android:layout_column="4"
49     android:layout_row="2"
50     android:text="4" />
51
52 <Button
53     android:id="@+id/BtnNum5"
54     android:layout_column="0"
55     android:layout_row="3"
56     android:text="5" />
57

```

2행 1열에 위치

2행 2열에 위치

2행 3열에 위치

2행 4열에 위치

2행 5열에 위치

숫자1 입력				
숫자2 입력				
0	1	2	3	4
5	6	7	8	9
더하기				
빼기				
곱하기				
나누기				
계산 결과:				

main.xml

```

58 <Button
59     android:id="@+id/BtnNum6"
60     android:layout_column="1"
61     android:layout_row="3"
62     android:text="6" />
63
64 <Button
65     android:id="@+id/BtnNum7"
66     android:layout_column="2"
67     android:layout_row="3"
68     android:text="7" />
69
70 <Button
71     android:id="@+id/BtnNum8"
72     android:layout_column="3"
73     android:layout_row="3"
74     android:text="8" />
75
76 <Button
77     android:id="@+id/BtnNum9"
78     android:layout_column="4"
79     android:layout_row="3"
80     android:text="9" />
81
82 <Button
83     android:id="@+id/BtnAdd"
84     android:layout_column="0"
85     android:layout_row="4"
86     android:layout_columnSpan="5"
87     android:layout_gravity="fill_horizontal"
88     android:text="더하기" />
89

```

3행1열에 위치

3행2열에 위치

3행3열에 위치

3행4열에 위치

4행0열에 위치

열을 확장(5열)

확장한 만큼(5) 수평으로
꽉 채움

숫자1 입력

숫자2 입력

0	1	2	3	4
5	6	7	8	9
더하기				
빼기				
곱하기				
나누기				

계산 결과 :

```

*main.xml
90 <Button
91     android:id="@+id/BtnSub"
92     android:layout_column="0"
93     android:layout_row="5"
94     android:layout_columnSpan="5"
95     android:layout_gravity="fill_horizontal"
96     android:text="빼기" />
97
98 <Button
99     android:id="@+id/BtnMul"
100    android:layout_column="0"
101    android:layout_row="6"
102    android:layout_columnSpan="5"
103    android:layout_gravity="fill_horizontal"
104    android:text="곱하기" />
105
106 <Button
107     android:id="@+id/BtnDiv"
108     android:layout_column="0"
109     android:layout_row="7"
110     android:layout_columnSpan="5"
111     android:layout_gravity="fill_horizontal"
112     android:text="나누기" />
113
114 <TextView
115     android:id="@+id/TextResult"
116     android:layout_column="0"
117     android:layout_row="8"
118     android:layout_columnSpan="5"
119     android:layout_gravity="fill_horizontal"
120     android:text="계산 결과: "
121     android:textColor="#FF0000"
122     android:textSize="20dp" />
123 </GridLayout>

```

5행 0열에 위치

열을 확장(5열)

확장한 만큼(5) 수
평으로 꽉 채움

숫자1 입력

숫자2 입력

0	1	2	3	4
5	6	7	8	9
더하기				
빼기				
곱하기				
나누기				

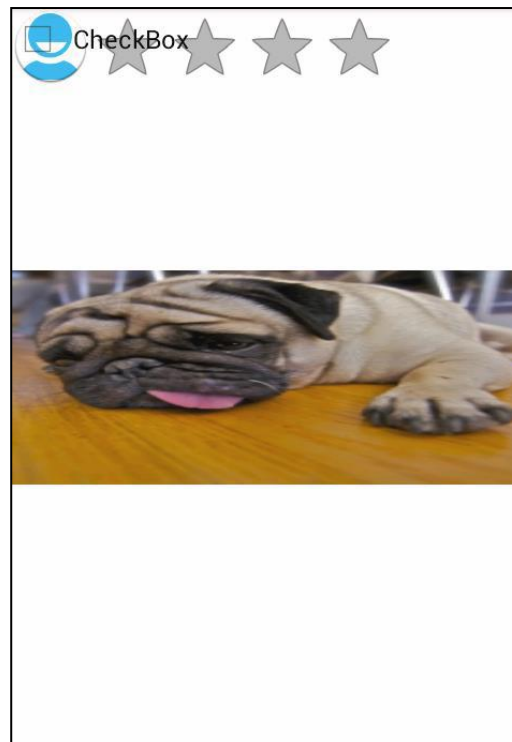
계산 결과:

프레임 레이아웃

○ 속성

- ◉ foreground : 프레임레이아웃의 전경이미지를 지정
- ◉ foregroundGravity : 전경이미지의 위치를 지정
(fill, right, left, top, bottom 등)

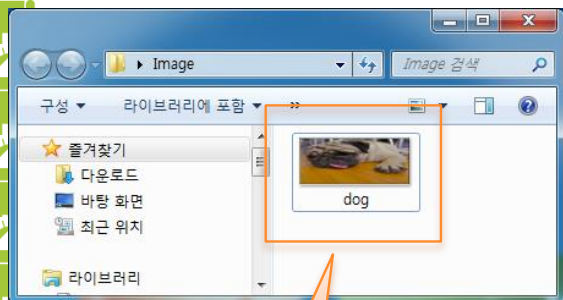
프로젝트 명: framelayout01



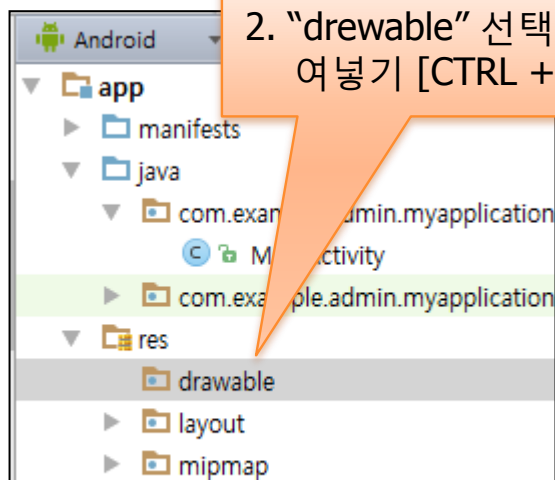
이미지 복사

○ dog.png 그림 파일을 복사

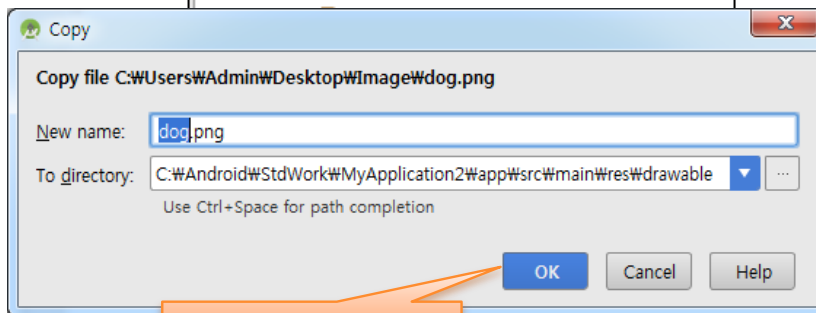
◎ [res] → [drawable]에 dog.png 파일 복사



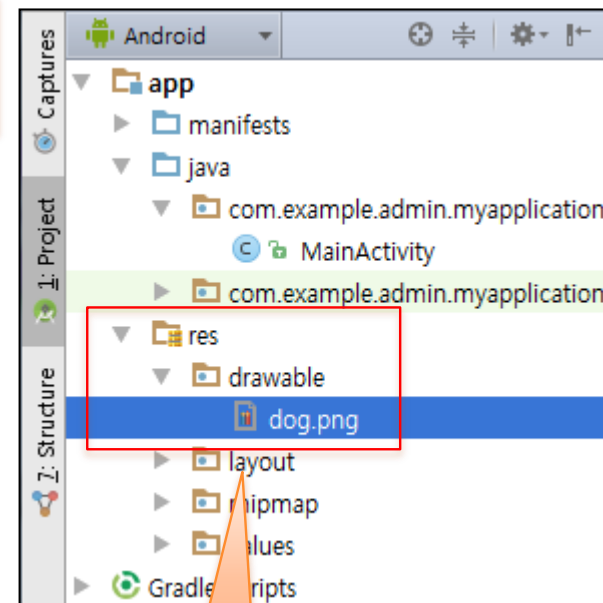
1. 이미지 복사
[CTRL + C]



2. "drewable" 선택 후 붙여넣기 [CTRL + V]

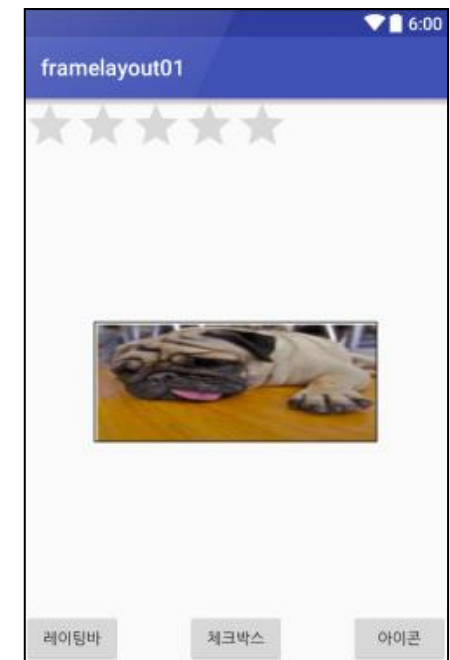


3. 클릭



4. 복사 확인

5. 코딩



```

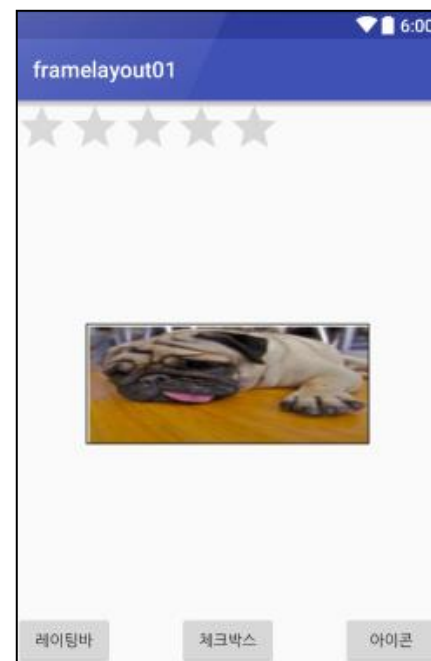
1  <?xml version="1.0" encoding="utf-8"?>
2  <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:id="@+id/framelayout"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:foreground="@drawable/dog"
8      android:foregroundGravity="center">
9
10     <RatingBar
11         android:id="@+id/ratingBar1"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content" />
14
15     <ImageView
16         android:id="@+id/img"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:src="@mipmap/ic_launcher"
20         android:visibility="invisible"/>
21
22     <CheckBox
23         android:id="@+id/checkbox1"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:text="CheckBox"
27         android:visibility="invisible"/>
28
29     <Button

```

```
activity_main.xml ×  
23 android:layout="@+id/checkbox" />  
24 android:layout_width="wrap_content"  
25 android:layout_height="wrap_content"  
26 android:text="CheckBox"  
27 android:visibility="invisible"/>
```

6. 코딩

```
28  
29 <Button  
30     android:id="@+id/btn1"  
31     android:layout_width="wrap_content"  
32     android:layout_height="wrap_content"  
33     android:layout_gravity="bottom|left"  
34     android:text="레이팅바" />  
35  
36 <Button  
37     android:id="@+id/btn2"  
38     android:layout_width="wrap_content"  
39     android:layout_height="wrap_content"  
40     android:layout_gravity="bottom|center"  
41     android:text="체크박스" />  
42  
43 <Button  
44     android:id="@+id/btn3"  
45     android:layout_width="wrap_content"  
46     android:layout_height="wrap_content"  
47     android:layout_gravity="bottom|right"  
48     android:text="아이콘" />  
49 </FrameLayout>
```



```
1 package com.example.admin.frameLayout01;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.CheckBox;
8 import android.widget.FrameLayout;
9 import android.widget.ImageView;
10 import android.widget.RatingBar;
11
12 public class MainActivity extends AppCompatActivity {
13
14     RatingBar rating;
15     CheckBox checkBox;
16     ImageView img;
17     FrameLayout frameLayout;
18     Button btn1, btn2, btn3;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
```

Import 추후 확인

7. 코딩

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

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

```
    rating = (RatingBar) findViewById(R.id.ratingBar1);
```

```
    checkBox = (CheckBox) findViewById(R.id.checkBox1);
```

```
    img = (ImageView) findViewById(R.id.img);
```

```
    frameLayout = (FrameLayout) findViewById(R.id.frameLayout);
```

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

```
    btn2 = (Button) findViewById(R.id.btn2);
```

```
    btn3 = (Button) findViewById(R.id.btn3);
```

```
    btn1.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View v) {
```

```
            rating.setVisibility(View.VISIBLE);
```

```
            checkBox.setVisibility(View.INVISIBLE);
```

```
            img.setVisibility(View.INVISIBLE);
```

```
        }
```

```
    });
```

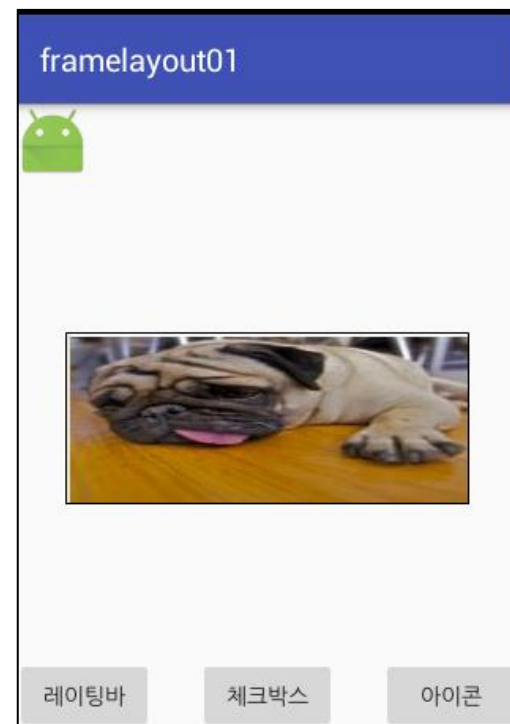
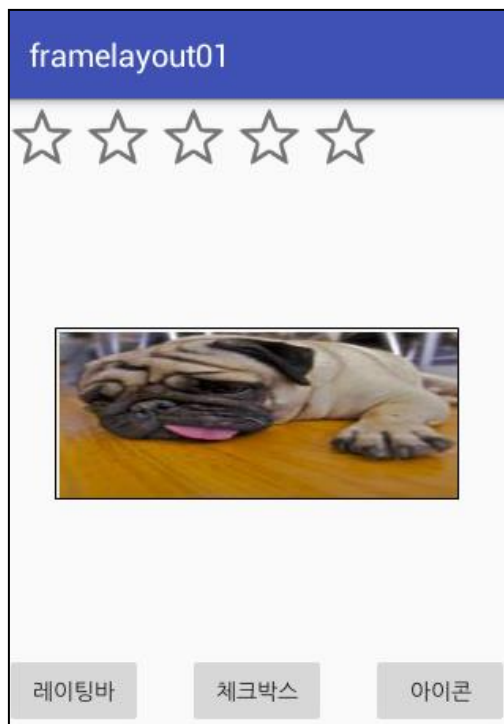
```
    btn2.setOnClickListener(new View.OnClickListener() {
```

8. 코딩

보여주세요

```
35 public void onClick(View v) {  
36     rating.setVisibility(View.VISIBLE);  
37     checkBox.setVisibility(View.INVISIBLE);  
38     img.setVisibility(View.INVISIBLE);  
39 }  
40 }  
41 btn2.setOnClickListener(new View.OnClickListener() {  
42     @Override  
43     public void onClick(View v) {  
44         rating.setVisibility(View.INVISIBLE);  
45         checkBox.setVisibility(View.VISIBLE);  
46         img.setVisibility(View.INVISIBLE);  
47     }  
48 });  
49 btn3.setOnClickListener(new View.OnClickListener() {  
50     @Override  
51     public void onClick(View v) {  
52         rating.setVisibility(View.INVISIBLE);  
53         checkBox.setVisibility(View.INVISIBLE);  
54         img.setVisibility(View.VISIBLE);  
55     }  
56 });  
57 }  
58 }
```

9. 코딩





○ 레이아웃 종류

- 리니어레이아웃(LinearLayout), 렐러티브레이아웃(RelativeLayout), 프레임레이아웃(FrameLayout), 테이블레이아웃(TableLayout), 그리드레이아웃(GridLayout)

○ 레이아웃 주요 속성

- orientation, gravity, padding, layout_weight, baselineAligned
- 테이블레이아웃은 표 형태로 위젯을 배치할 때 사용되는데, 행의 개수는 **<TableRow>**로 결정된다
- 그리드레이아웃은 테이블레이아웃보다 좀 더 직관적으로 구성할 수 있으며, 행 단위의 확장도 가능해서 다양한 형태의 표를 구성할 수 있다.
- 프레임레이아웃은 레이아웃 안의 위젯들을 좌측상단에 겹쳐서 출력한다

