

레이아웃 익히기

Instructor: Park, JoonSeok (pjs50@pusan.ac.kr)

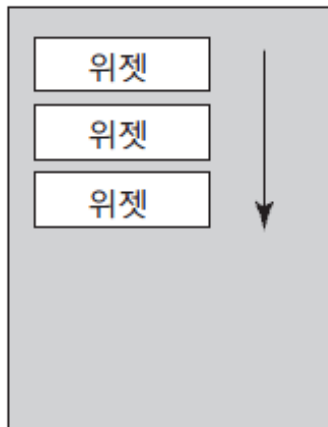
레이아웃 개요

▶ 레이아웃 대표적인 속성

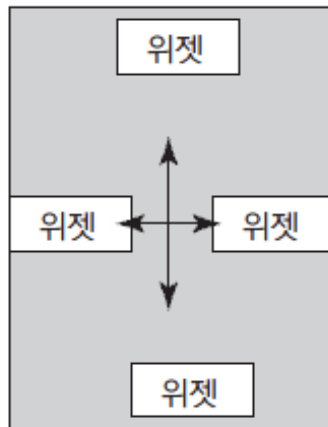
- ▶ orientation: 레이아웃 안에 배치할 위젯의 수직 또는 수평 방향 설정
- ▶ gravity: 레이아웃 안에 배치할 위젯의 정렬 방향을 좌측, 우측, 중앙 등으로 설정
- ▶ padding: 레이아웃 안에 배치할 위젯의 여백 설정
- ▶ layout_weight: 레이아웃이 전체 화면에서 차지하는 공간의 가중 값을 설정, 여러 개의 레이아웃이 중복될 때 사용
- ▶ baselineAligned: 레이아웃 안에 배치할 위젯들을 보기 좋게 정렬



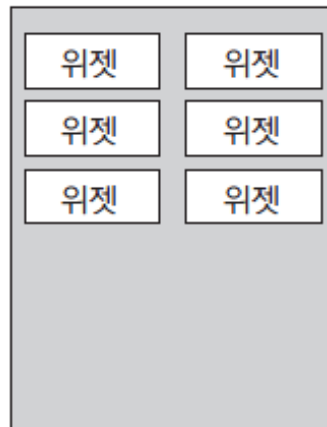
레이아웃 종류



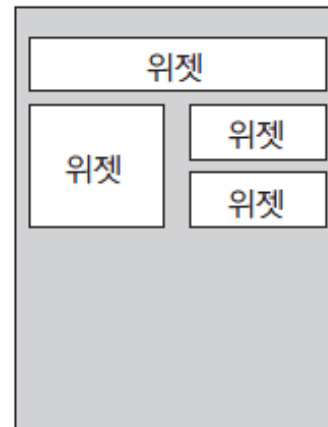
리니어레이아웃



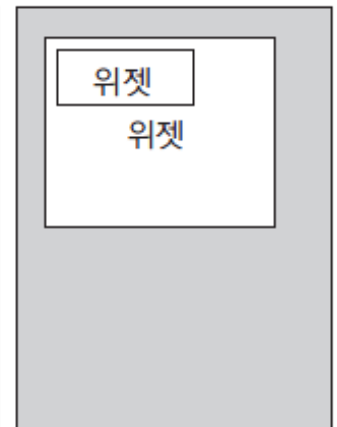
렐러티브레이아웃



테이블레이아웃



그리드레이아웃

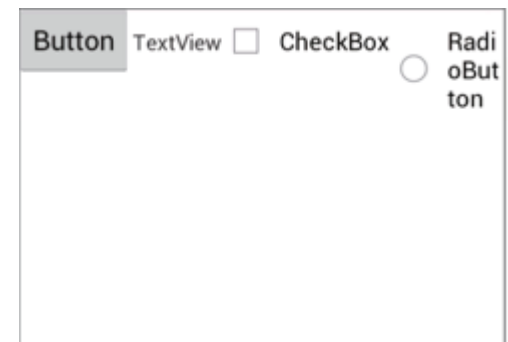
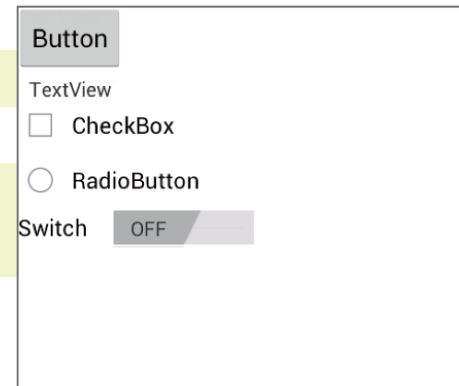


프레임레이아웃

리니어 레이아웃

▶ vertical은 수직 배열, horizontal은 수평배열

```
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>
```



```
1 <LinearLayout
2     android:orientation="horizontal" >
```

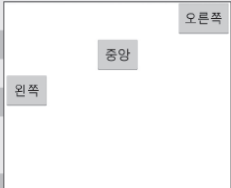
리니어 레이아웃

- ▶ **gravity: left, right, center, top, bottom**
 - ▶ 두개의 조합도 가능
- ▶ **layout_gravity**
 - ▶ 자신의 위치를 부모(주로 레이아웃)의 어디쯤에 위치시킬지 결정

```
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>
```



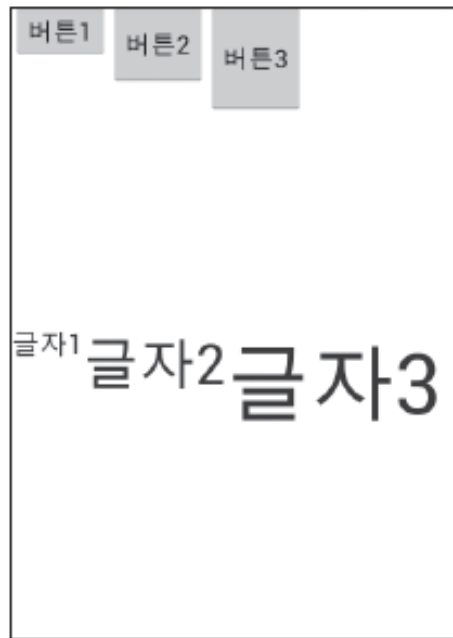
```
1 <LinearLayout
2   android:orientation="vertical" >
3   <Button
4     android:layout_gravity="right"
5     android:text="오른쪽" />
6   <Button
7     android:layout_gravity="center"
8     android:text="중앙" />
9   <Button
10    android:layout_gravity="left"
11    android:text="왼쪽" />
12 </LinearLayout>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



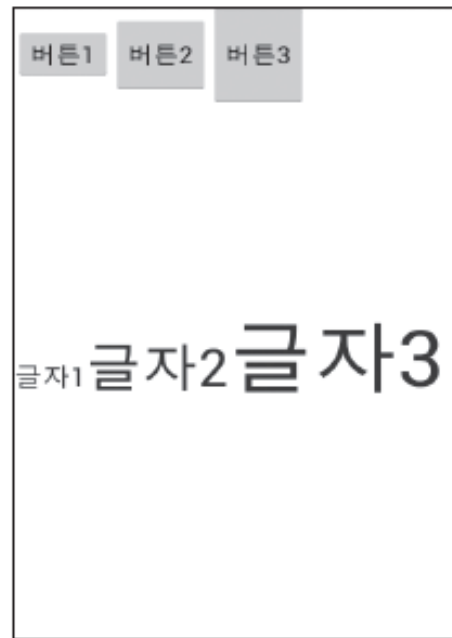
리니어 레이아웃

▶ baselineAligned

- ▶ true 와 false
- ▶ 크기가 다른 위젯들을 보기 좋게 정렬



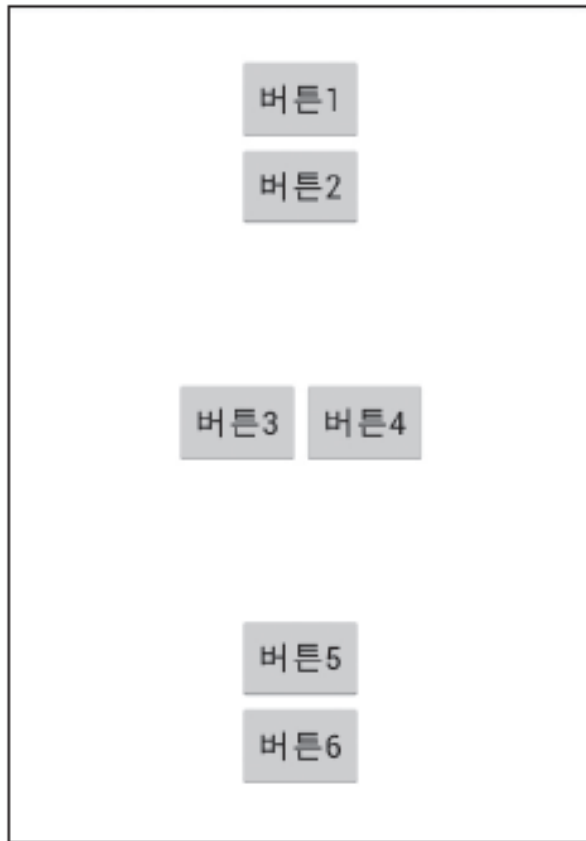
(a) false로 지정



(b) true로 지정하거나 생략

리니어 레이아웃

▶ 중복 레이아웃



```
<LinearLayout>
```

```
    <LinearLayout>
```

```
        위젯들...
```

```
    </LinearLayout>
```

```
    <LinearLayout>
```

```
        위젯들...
```

```
    </LinearLayout>
```

```
    <LinearLayout>
```

```
        위젯들...
```

```
    </LinearLayout>
```

```
</LinearLayout>
```

리니어 레이아웃

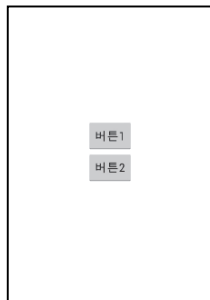
- ▶ **layout_weight**

- ▶ 리니어레이아웃을 여러 개 사용할 경우 각 레이아웃의 크기를 지정할 때 사용



리니어 레이아웃

```
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>
```

```
1 <LinearLayout
2   android:orientation="vertical" >
3   <LinearLayout
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     ~~~~ 중간 생략 ~~~~
7   </LinearLayout>
8   <LinearLayout
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    ~~~~ 중간 생략 ~~~~
12  </LinearLayout>
13  <LinearLayout
14    android:layout_width="match_parent"
15    android:layout_height="wrap_content"
16    ~~~~ 중간 생략 ~~~~
17  </LinearLayout>
18 </LinearLayout>
```



```

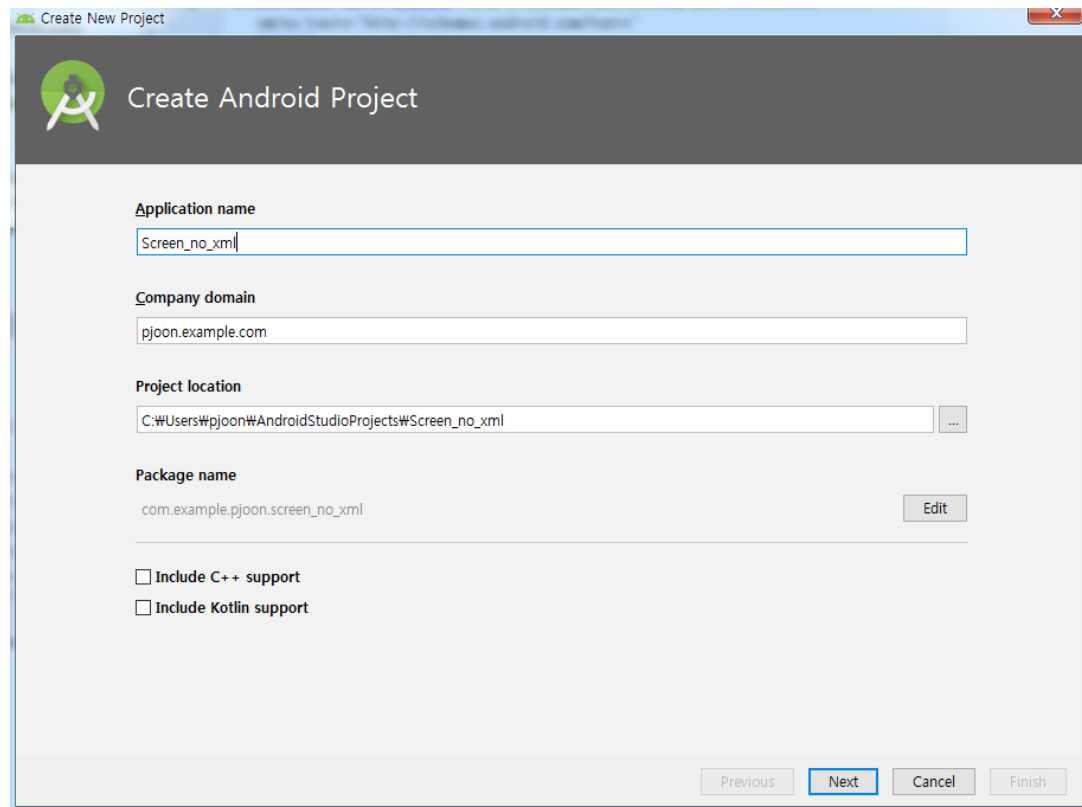
1 <LinearLayout
2     android:orientation="vertical" >
3     <LinearLayout
4         android:layout_width="match_parent"
5         android:layout_height="match_parent"
6         android:layout_weight="1"
7         ~~~~ 중간 생략 ~~~~
8     </LinearLayout>
9     <LinearLayout
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        android:layout_weight="1"
13        ~~~~ 중간 생략 ~~~~
14    </LinearLayout>
15    <LinearLayout
16        android:layout_width="match_parent"
17        android:layout_height="match_parent"
18        android:layout_weight="1"
19        ~~~~ 중간 생략 ~~~~
20    </LinearLayout>
21 </LinearLayout>

```

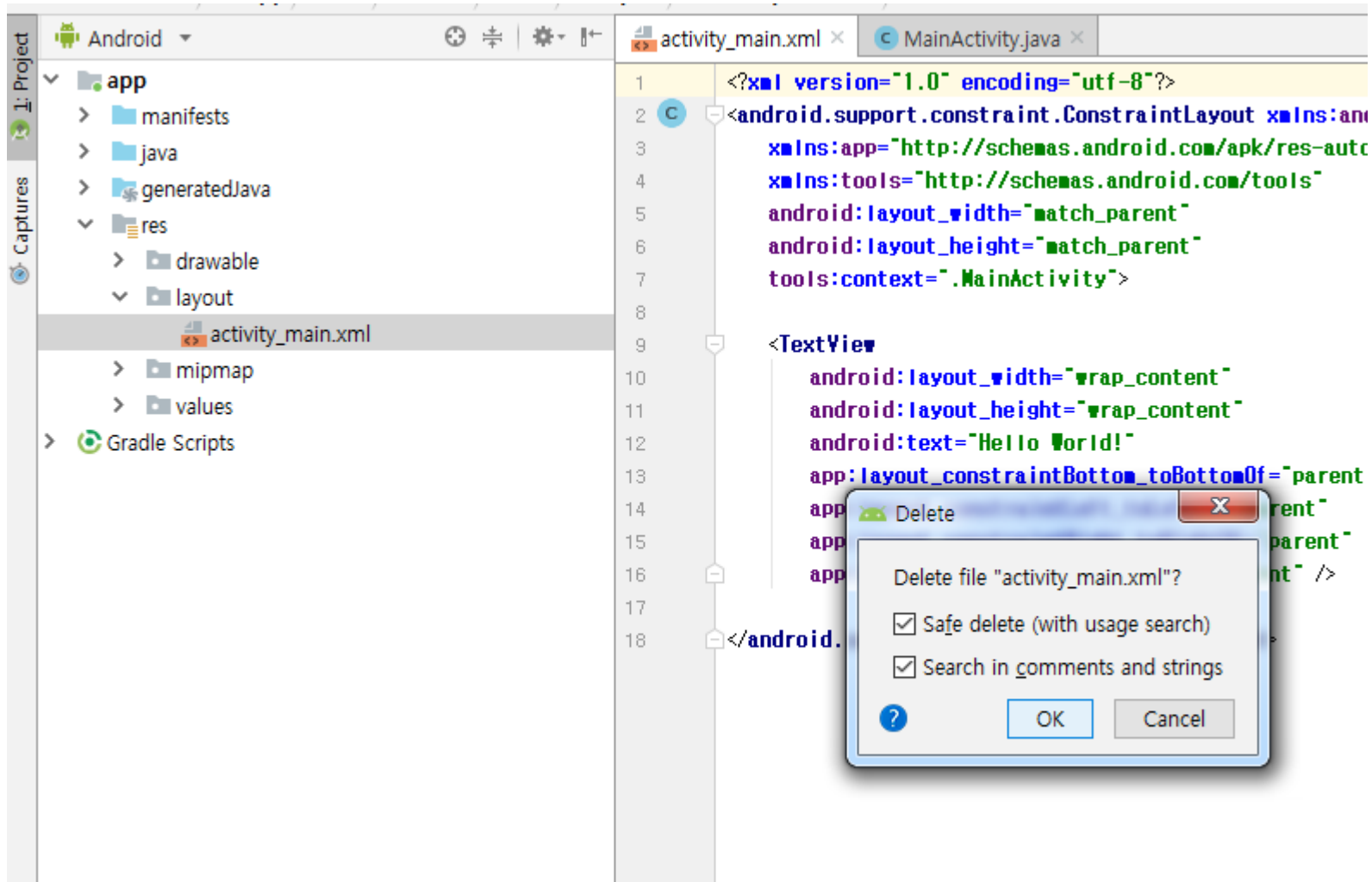


XML 없이 화면 코딩하기

- ▶ 버튼을 클릭하면 Toast 메시지가 출력되는 화면을 XML 파일 없이 Java로 코딩
 - ▶ 화면 디자인 및 편집
 - ▶ activity_main.xml 삭제



XML 없이 화면 코딩하기



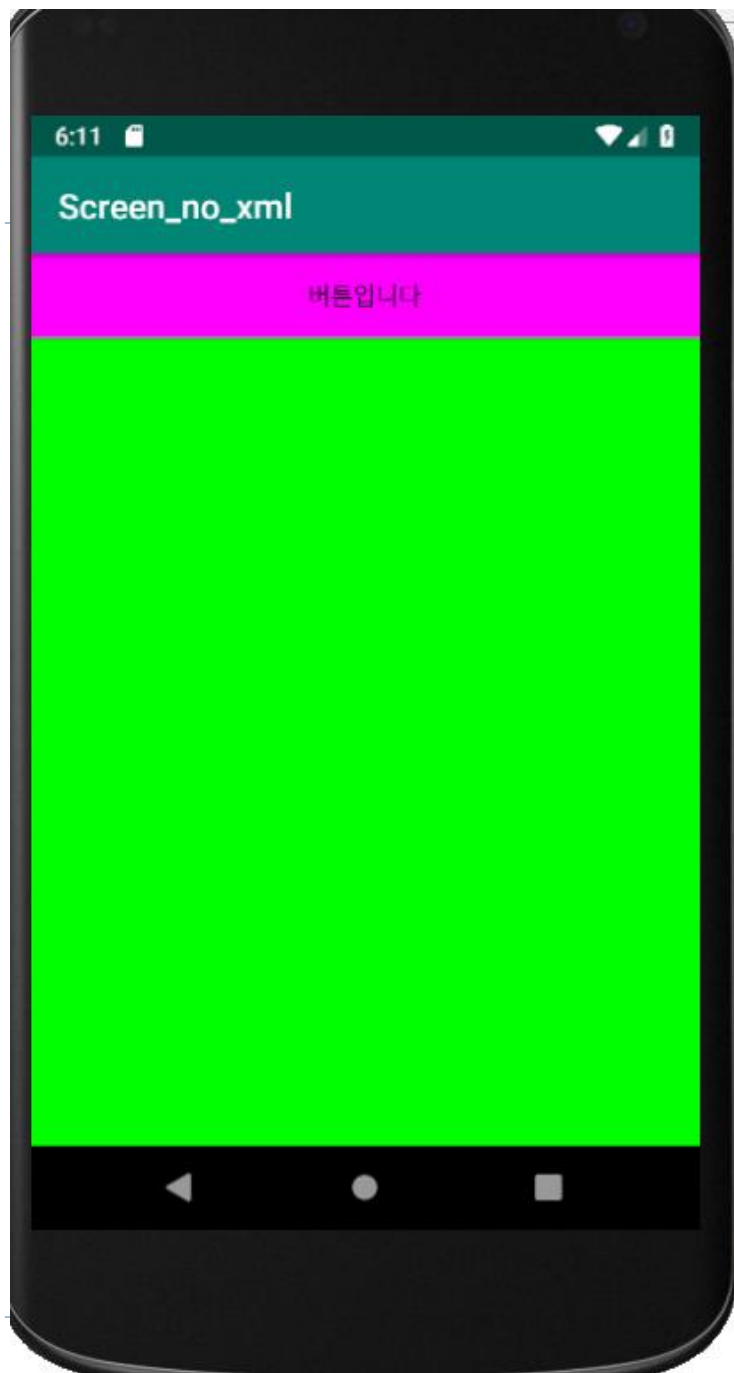
```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // setContentView(R.layout.activity_main);  
  
        LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(  
            LinearLayout.LayoutParams.MATCH_PARENT,  
            LinearLayout.LayoutParams.MATCH_PARENT);  
  
        LinearLayout baseLayout = new LinearLayout(context, this);  
        baseLayout.setOrientation(LinearLayout.VERTICAL);  
        baseLayout.setBackgroundColor(Color.rgb(red: 0, green: 255, blue: 0));  
        setContentView(baseLayout, params);  
    }  
}
```



```
Button btn = new Button( context this);
btn.setText("버튼입니다");
btn.setBackgroundColor(Color.MAGENTA);
baseLayout.addView(btn);

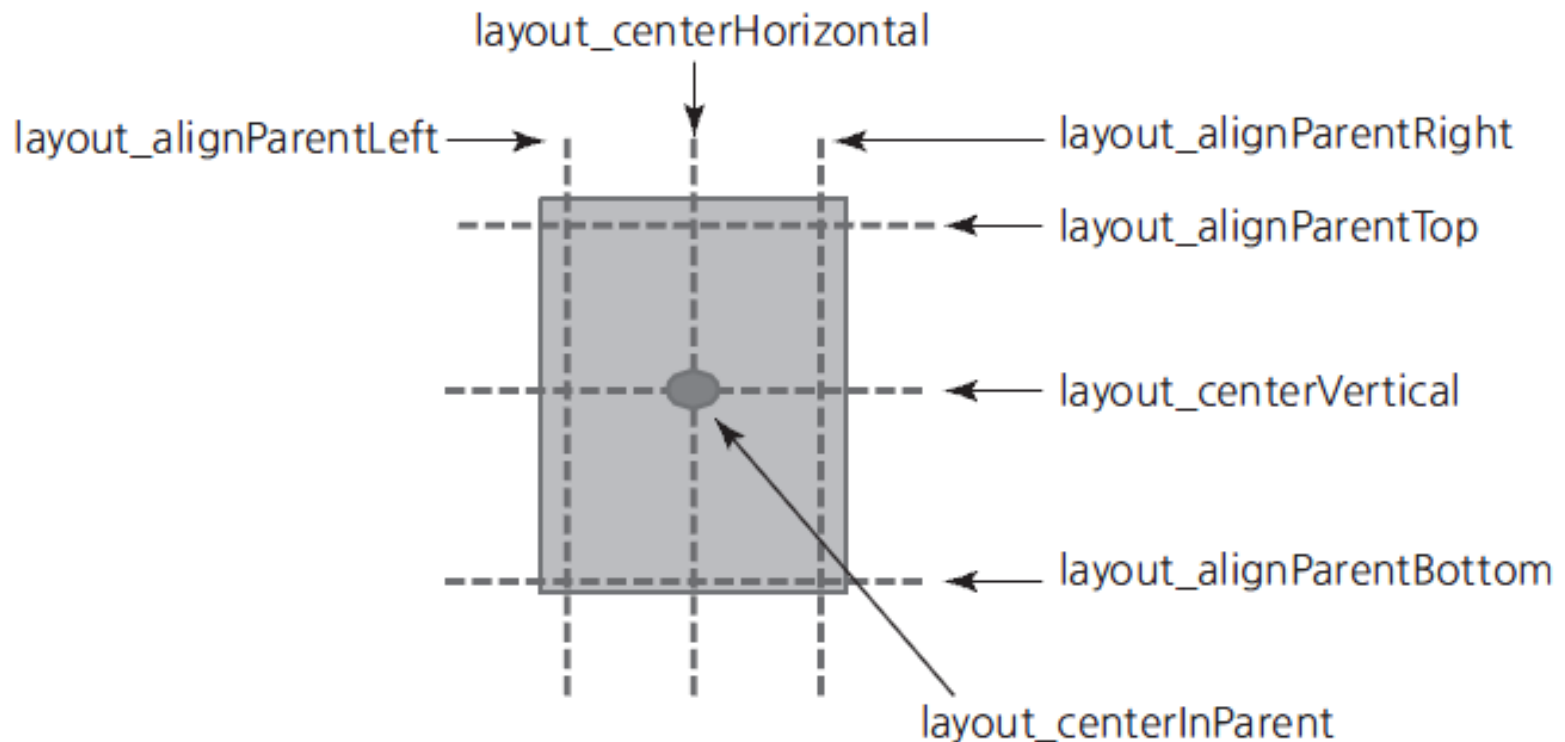
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), text "코드로 생성한 버튼입니다", Toast.LENGTH_SHORT).show();
    }
});
```





릴러티브 레이아웃

- ▶ 레이아웃 내부에 포함된 위젯들을 상대적인 위치로 배치



[그림 5-9] 부모 레이아웃에서의 위치를 적용할 때 속성

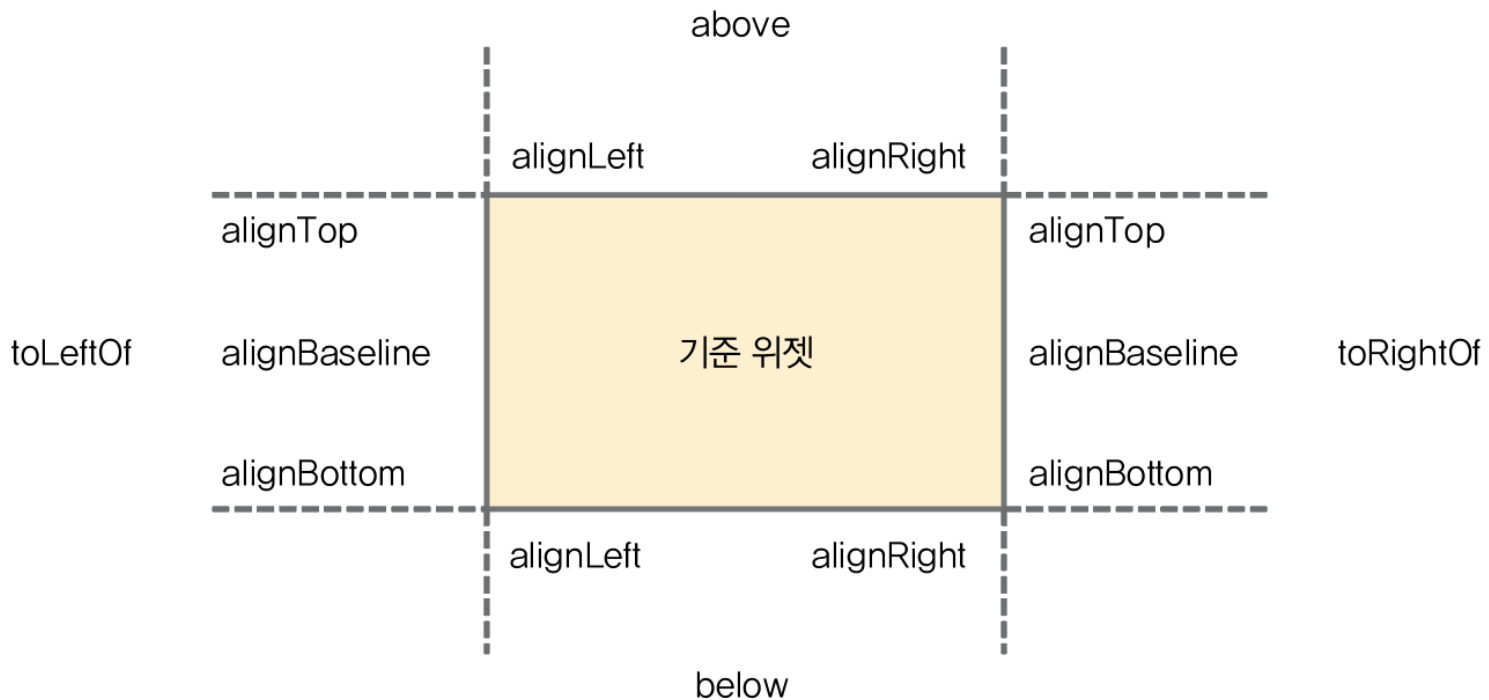
```

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>

```



- ▶ 다른 위젯의 상대 위치에 배치
 - ▶ 각 속성의 값은 다른 위젯의 id를 지정하면 되는데, "@+id/기준위젯_아이디" 형식으로 사용



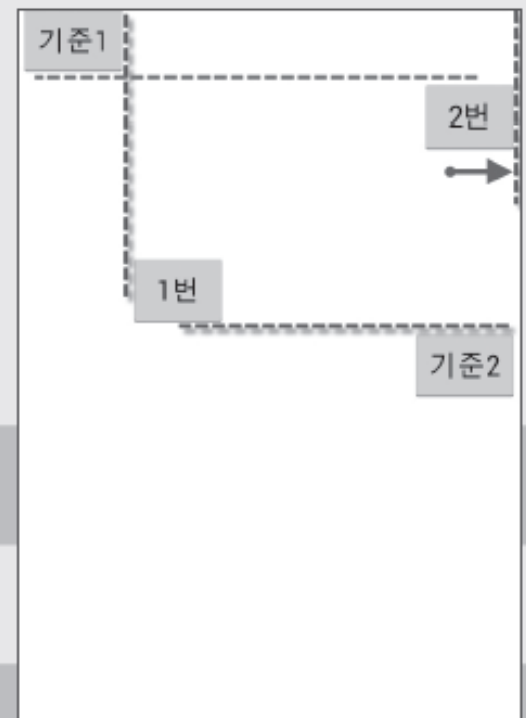
```

1 <RelativeLayout xmlns:android="http://www."
2     android:layout_width="match_parent"
3     android:layout_height="match_parent" >
4     <Button
5         android:id="@+id/baseBtn"
6         android:layout_width="150dp"
7         android:layout_height="150dp"
8         android:layout_centerHorizontal="true"
9         android:layout_centerVertical="true"
10        android:text="기준 위젯" />
11    <Button
12        android:layout_alignTop="@+id/baseBtn"
13        android:layout_toLeftOf="@+id/baseBtn"
14        android:text="1번" />
15        ~~~~ 중간 생략 (버튼 2개) ~~~~
16    <Button
17        android:layout_above="@+id/baseBtn"
18        android:layout_alignLeft="@+id/baseBtn"
19        android:text="4번" />
20    <Button
21        android:layout_alignRight="@+id/baseBtn"
22        android:layout_below="@+id/baseBtn"
23        android:text="5번" />
24    <Button
25        android:layout_above="@+id/baseBtn"
26        android:layout_toRightOf="@+id/baseBtn"
27        android:text="6번" />
28 </RelativeLayout>

```

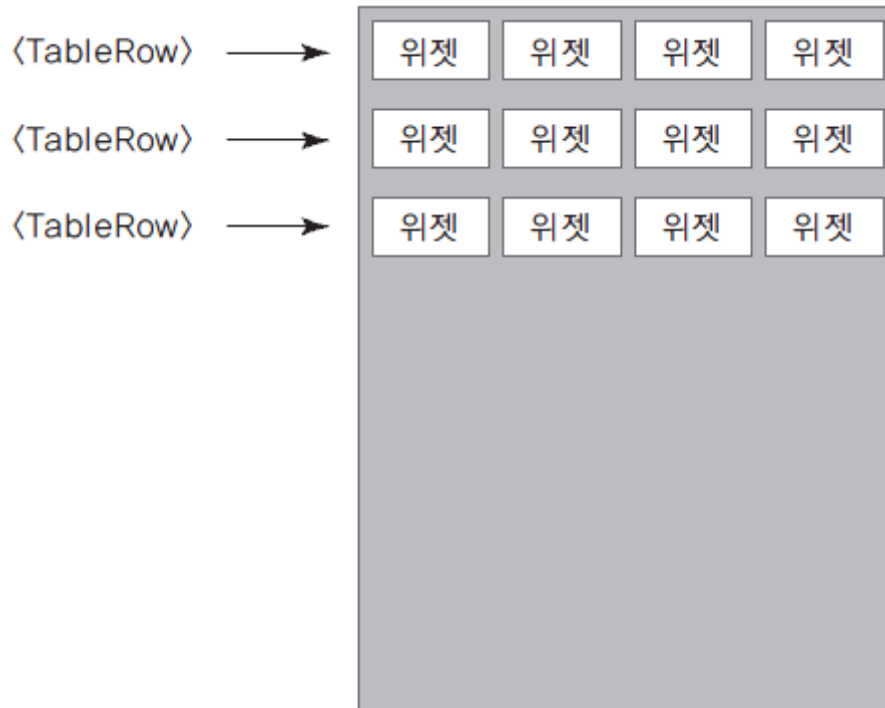


```
1 <RelativeLayout xmlns:android="http://~"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent" >
4     <Button
5         android:id="@+id/baseBtn1"
6         android:layout_alignParentLeft="true"
7         android:layout_alignParentTop="true"
8         android:text="기준1" />
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>
```



테이블 레이아웃

- ▶ 위젯을 표 형태로 배치
 - ▶ <TableRow> : 행의 개수
 - ▶ <TableRow> 안에 포함된 위젯의 개수: 열의 개수



▶ 속성

- ▶ `layout_span`은 열을 합쳐서 표시하라는 의미
- ▶ `layout_column`은 지정된 열에 현재 위젯을 표시하라는 의미
- ▶ `stretchColumns`는 `<TableLayout>` 자체에 설정하는 속성으로 지정된 열의 폭을 늘리라는 의미

```
1 <TableLayout xmlns:android="http://www."
2 >
3   <TableRow>
4     <Button
5       android:text="1" />
6     <Button
7       android:layout_span="2"
8       android:text="2" />
9     <Button
10      android:text="3" />
11   </TableRow>
```

```
12   <TableRow>
13     <Button
14       android:layout_column="1"
15       android:text="4" />
16     <Button
17       android:text="5" />
18     <Button
19       android:text="6" />
20   </TableRow>
21 </TableLayout>
```

1	2	3
	4	5
		6

테이블 레이아웃을 활용한 계산기 앱

Create New Project

Create Android Project

Application name
TableCalc

Company domain
pjoon.example.com

Project location
C:\Users#pjoon#AndroidStudioProjects#TableCalc

Package name
com.example.pjoon.tablecalc

☐ Include C++ support

☐ Include Kotlin support

Previous Next Cancel Finish



테이블레이아웃 계산기

숫자1 입력

숫자2 입력

0

1

2

3

4

5

6

7

8

9

더하기

빼기

곱하기

나누기

계산 결과 :

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/tableLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   >
8
9   <TableRow>
10
11     <EditText
12       android:layout_weight = "1"
13       android:id="@+id/Edit1"
14       android:layout_span="5"
15       android:hint="숫자1 입력" />
16   </TableRow>
17
18   <TableRow>
19
20     <EditText
21       android:layout_weight = "1"
22       android:id="@+id/Edit2"
23       android:layout_span="5"
24       android:hint="숫자2 입력" />
25   </TableRow>
```

```

27 <TableRow>
28
29   <Button
30       android:layout_weight = "1"
31       android:id="@+id/BtnNum0"
32       android:text="0" />
33
34   <Button
35       android:layout_weight = "1"
36       android:id="@+id/BtnNum1"
37       android:text="1" />
38
39   <Button
40       android:layout_weight = "1"
41       android:id="@+id/BtnNum2"
42       android:text="2" />
43
44   <Button
45       android:layout_weight = "1"
46       android:id="@+id/BtnNum3"
47       android:text="3" />
48
49   <Button
50       android:layout_weight = "1"
51       android:id="@+id/BtnNum4"
52       android:text="4" />
53 </TableRow>

```

```

55 <TableRow>
56
57   <Button
58       android:layout_weight = "1"
59       android:id="@+id/BtnNum5"
60       android:text="5" />
61
62   <Button
63       android:layout_weight = "1"
64       android:id="@+id/BtnNum6"
65       android:text="6" />
66
67   <Button
68       android:layout_weight = "1"
69       android:id="@+id/BtnNum7"
70       android:text="7" />
71
72   <Button
73       android:layout_weight = "1"
74       android:id="@+id/BtnNum8"
75       android:text="8" />
76
77   <Button
78       android:layout_weight = "1"
79       android:id="@+id/BtnNum9"
80       android:text="9" />
81 </TableRow>

```

```

83 <TableRow>
84
85     <Button
86         android:layout_weight = "1"
87         android:id="@+id/BtnAdd"
88         android:layout_margin="5dp"
89         android:layout_span="5"
90         android:text="더하기" />
91 </TableRow>
92
93 <TableRow>
94
95     <Button
96         android:layout_weight = "1"
97         android:id="@+id/BtnSub"
98         android:layout_margin="5dp"
99         android:layout_span="5"
100        android:text="빼기" />
101 </TableRow>
102
103 <TableRow>
104
105     <Button
106         android:layout_weight = "1"
107         android:id="@+id/BtnMul"
108         android:layout_margin="5dp"
109         android:layout_span="5"
110        android:text="곱하기" />
111 </TableRow>

```

```

112
113 <TableRow>
114
115     <Button
116         android:layout_weight = "1"
117         android:id="@+id/BtnDiv"
118         android:layout_margin="5dp"
119         android:layout_span="5"
120        android:text="나누기" />
121 </TableRow>
122
123 <TableRow>
124
125     <TextView
126         android:id="@+id/TextResult"
127         android:layout_margin="5dp"
128         android:layout_span="5"
129        android:text="계산 결과 : "
130        android:textColor="#FF0000"
131        android:textSize="20dp" />
132 </TableRow>
133
134 </TableLayout>

```

10

11 

12

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

```
24
25
26   
27
28
29
30 setTitle("테이블레이아웃 계산기");
31
32 edit1 = (EditText) findViewById(R.id.Edit1);
33 edit2 = (EditText) findViewById(R.id.Edit2);
34
35 btnAdd = (Button) findViewById(R.id.BtnAdd);
36 btnSub = (Button) findViewById(R.id.BtnSub);
37 btnMul = (Button) findViewById(R.id.BtnMul);
38 btnDiv = (Button) findViewById(R.id.BtnDiv);
39
40 textResult = (TextView) findViewById(R.id.TextResult);
41
```



```
41  
42 btnAdd.setOnTouchListener(new View.OnTouchListener() {  
43     ↑ public boolean onTouch(View arg0, MotionEvent arg1) {  
44         num1 = edit1.getText().toString();  
45         num2 = edit2.getText().toString();  
46         result = Integer.parseInt(num1) + Integer.parseInt(num2);  
47         textResult.setText("계산 결과 : " + result.toString());  
48         return false;  
49     }  
50 });
```

```
51  
52 btnSub.setOnTouchListener(new View.OnTouchListener() {  
53     ↑ public boolean onTouch(View arg0, MotionEvent arg1) {  
54         num1 = edit1.getText().toString();  
55         num2 = edit2.getText().toString();  
56         result = Integer.parseInt(num1) - Integer.parseInt(num2);  
57         textResult.setText("계산 결과 : " + result.toString());  
58         return false;  
59     }  
60 });
```

```
61  
62 btnMul.setOnTouchListener(new View.OnTouchListener() {  
63     ↑ public boolean onTouch(View arg0, MotionEvent arg1) {  
64         num1 = edit1.getText().toString();  
65         num2 = edit2.getText().toString();  
66         result = Integer.parseInt(num1) * Integer.parseInt(num2);  
67         textResult.setText("계산 결과 : " + result.toString());  
68         return false;  
69     }  
70 });
```

```
71  
72 btnDiv.setOnTouchListener(new View.OnTouchListener() {  
73     ↑ public boolean onTouch(View arg0, MotionEvent arg1) {  
74         num1 = edit1.getText().toString();  
75         num2 = edit2.getText().toString();  
76         result = Integer.parseInt(num1) / Integer.parseInt(num2);  
77         textResult.setText("계산 결과 : " + result.toString());  
78         return false;  
79     }  
80 });
```

```

85 // 숫자 버튼 10개를 대입
86 for (i = 0; i < numBtnIds.length; i++) {
87     numButtons[i] = (Button) findViewById(numBtnIds[i]);
88 }
89 // 숫자 버튼 10개에 대해서 클릭이벤트 처리
90 for (i = 0; i < numBtnIds.length; i++) {
91
92     final int index; // 주의! 꼭 필요함..
93     index = i;
94
95     numButtons[index].setOnClickListener(new View.OnClickListener() {
96         public void onClick(View v) {
97             // 포커스가 되어 있는 에디트텍스트에 숫자 추가
98             if (edit1.isFocused() == true) {
99                 num1 = edit1.getText().toString()
100                     + numButtons[index].getText().toString();
101                 edit1.setText(num1);
102             } else if (edit2.isFocused() == true) {
103                 num2 = edit2.getText().toString()
104                     + numButtons[index].getText().toString();
105                 edit2.setText(num2);
106             } else {
107                 Toast.makeText(getApplicationContext(),
108                     text: "먼저 에디트텍스트를 선택하세요", Toast.LENGTH_SHORT).show();
109             }
110         }
111     });
112 }
113 });
114
115 }

```


6:40



테이블레이아웃 계산기

1

2

0

1

2

3

4

5

6

7

8

9

더하기

빼기

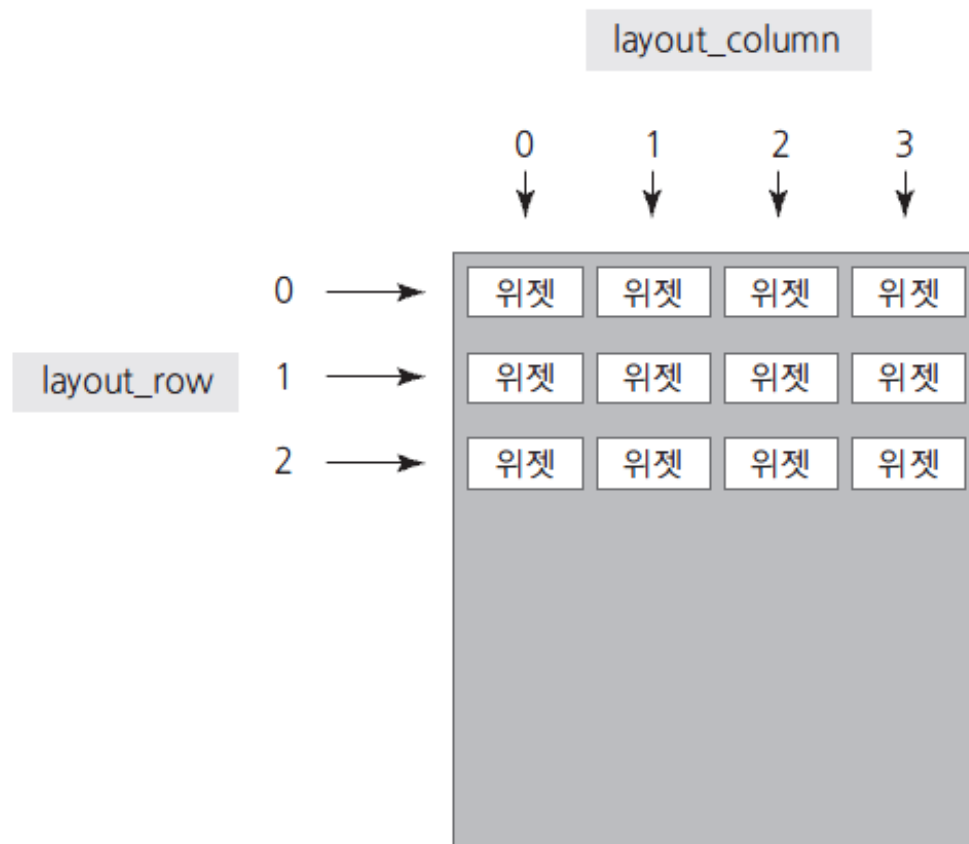
곱하기

나누기

계산 결과 : 3

그리드 레이아웃

- ▶ 테이블 레이아웃과 마찬가지로 위젯을 표 형태로 배치
- ▶ 테이블레이아웃보다는 유연한 화면 구성에 적합



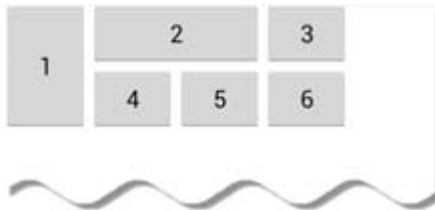
그리드 레이아웃

▶ 속성

- ▶ rowCount: 행 개수
- ▶ columnCount: 행 개수
- ▶ layout_row: 자신이 위치할 행 번호 (0부터 시작)
- ▶ layout_column: 자신이 위치할 열 번호 (0부터 시작)
- ▶ layout_rowSpan: 행을 지정된 개수만큼 확장
- ▶ layout_columnSpan: 열을 지정된 개수만큼 확장
- ▶ layout_gravity: 주로 fill, fill_vertical, fill_horizaontal 등으로 지정, layout_rowSpan이나 layout_columnSpan으로 행 또는 열이 확장되었을 때, 위젯을 확장된 셀에 꽉 채우는 효과



그리드 레이아웃



```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:columnCount="4"
    android:rowCount="2" >
```

```
<Button
    android:layout_column="0"
    android:layout_gravity="fill_vertical"
    android:layout_row="0"
    android:layout_rowSpan="2"
    android:text="1" />
```

```
<Button
    android:layout_column="1"
    android:layout_columnSpan="2"
    android:layout_gravity="fill_horizontal"
    android:layout_row="0"
    android:text="2" />
```

```
<Button
    android:layout_column="3"
    android:layout_row="0"
    android:text="3" />
```

```
<Button
    android:layout_column="1"
    android:layout_row="1"
    android:text="4" />
```

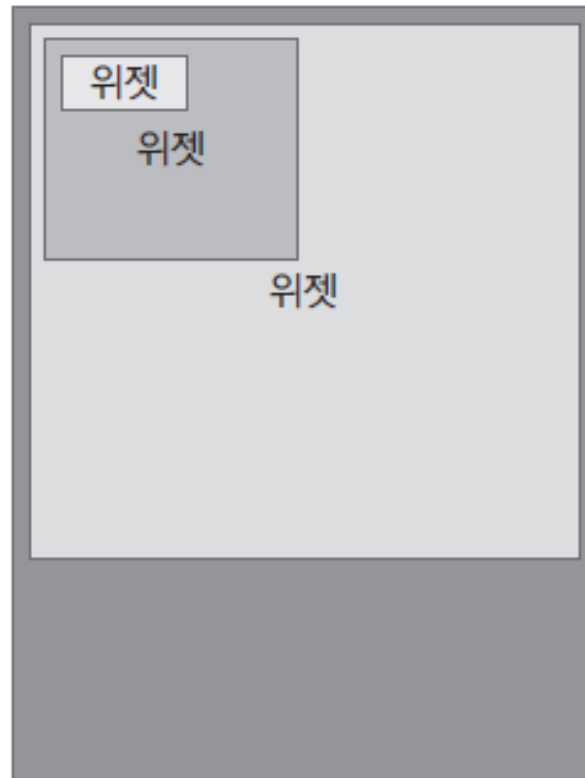
```
<Button
    android:layout_column="2"
    android:layout_row="1"
    android:text="5" />
```

```
<Button
    android:layout_column="3"
    android:layout_row="1"
    android:text="6" />
```

```
</GridLayout>
```

프레임 레이아웃

- ▶ 단순히 레이아웃 내의 위젯들은 왼쪽 상단부터 겹쳐서 출력함
- ▶ 프레임레이아웃 그 자체를 사용하기 보다는 탭 위젯 등과 혼용해서 사용할 때 유용함



[그림 5-17] 프레임레이아웃 개요

프레임 레이아웃

▶ 속성

- ▶ foreground: 프레임레이아웃의 전경 이미지를 지정
- ▶ foregroundGravity: 전경 이미지의 위치를 지정 (fill, right, left, top, bottom 등의 값)



프레임 레이아웃

```
<FrameLayout 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"
    android:foreground="@drawable/dog"
    android:foregroundGravity="fill_horizontal" >

    <RatingBar
        android:id="@+id/ratingBar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher" />

    <CheckBox
        android:id="@+id/checkbox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CheckBox" />

</FrameLayout>
```

