

Ch05. 레이아웃 익히기2

모바일앱프로그래밍
김지심교수

학습목표



레이아웃의 개념을 익힌다.
화면을 다양한 레이아웃으로 구성할 수 있다.
Java 코드만으로 화면을 작성할 수 있다.

목차

-
1. 레이아웃 개요
 2. 리니어레이아웃
 3. 기타 레이아웃
 4. 제약 레이아웃



03



기타 레이아웃



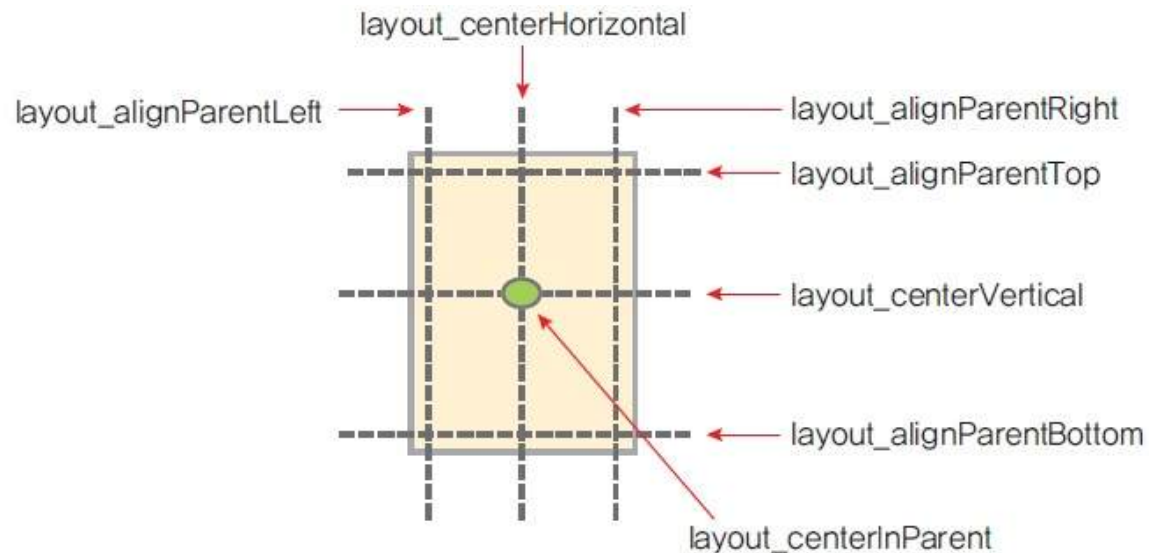
3. 기타 레이아웃 | RelativeLayout

RelativeLayout(상대레이아웃)

레이아웃 내부에 포함된 위젯들에 준하여 상대적인 위치로 배치

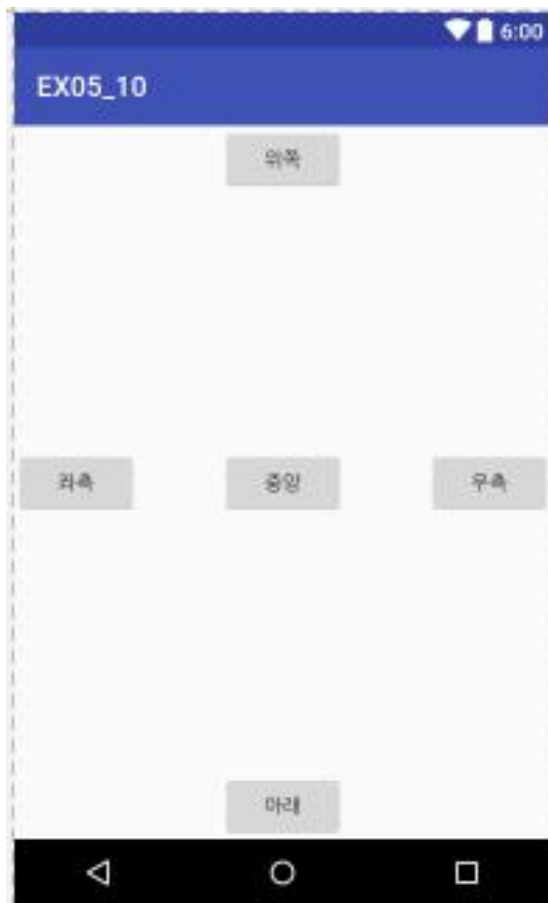
방법 1) 렐러티브레이아웃의 상하좌우에 배치

부모 레이아웃의 어느 위치에 적용할지 결정(true/false)



3. 기타 레이아웃 | RelativeLayout

예제5-10: Ex05_10
렐러티브레이아웃 예제



예제 5-10 렐러티브레이아웃 XML 코드 1

```
1 <RelativeLayout xmlns:android="http://~"
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" /> 함께 사용하여 가운데 배치
8   <Button
9     android:layout_alignParentLeft="true"
10    android:layout_centerVertical="true"
11    android:text="좌측" /> 함께 사용하여 가운데 배치
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>
```



3. 기타 레이아웃 | RelativeLayout

방법 2) 다른 위젯의 상대적 위치에 배치

기준위젯을 중심으로 우선 상대적으로 배치한 후 여타 위젯들에 상대적으로 배치
각 속성의 값은 다른 위젯의 id를 지정하면 됨
“@+id/기준 위젯의 아이디”와 같은 형식으로 사용

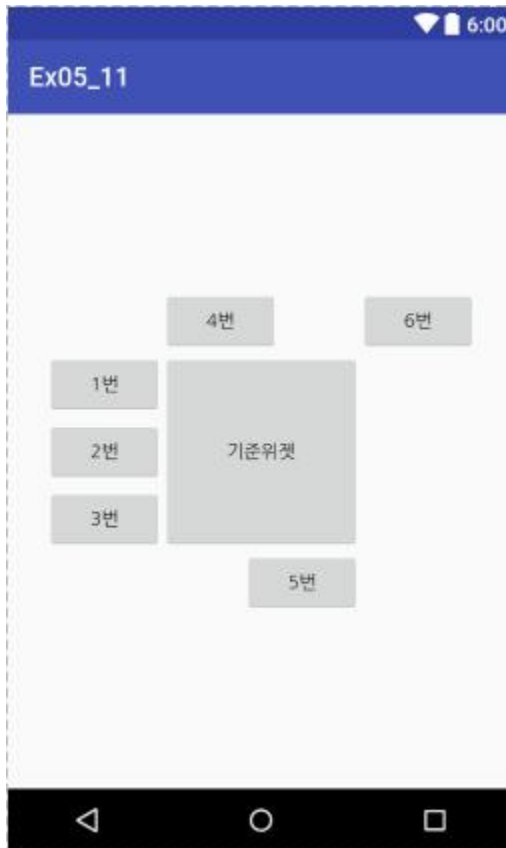


* 교재엔 속성명 layout_이 생략됨

3. 기타 레이아웃 | RelativeLayout

예제5-11: Ex05_11

방법1로 기준 위젯을 먼저 명시하고, 이를 기준으로 배치



android:layout_alignTop="@+id/baseBtn

예제 5-11 헬러티브레이아웃 XML 코드 2

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

android:layout_alignTop="@+id/baseBtn



3. 기타 레이아웃 | RelativeLayout

예제5-12: Ex05_12

여러 위젯을 기준으로 상대 위치를 지정(방법들 혼용)



기준1의 오른쪽 &
기준2의 위쪽

예제 5-12 렐라티브레이아웃 속성을 조합한 XML 코드

```
1 <RelativeLayout xmlns:android="http://www."
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>
```

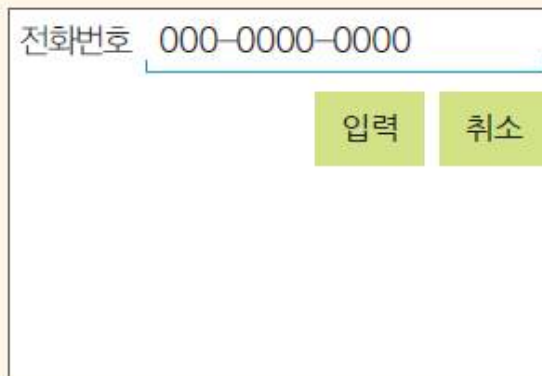


3. 기타 레이아웃 | RelativeLayout

실습(직접 풀어보기): Test05_04

▶ 직접 풀어보기 5-4

다음 화면의 XML 코드를 중복 리니어레이아웃과 렐러티브레이아웃으로 각각 작성해보자. 텍스트뷰 1개, 에디트텍스트 1개, 버튼 2개로 구성한다.



A UI mockup for a form. It features a text input field with the placeholder text "전화번호" (Phone Number) and the value "000-0000-0000". Below the input field are two green buttons: "입력" (Input) and "취소" (Cancel).

2개 버전으로 작성

Test05_04_Linear 프로젝트 1개,
Test05_04_Relative 프로젝트 1개

3. 기타 레이아웃 | TableLayout

TableLayout

주로 위젯을 표 형태로 배치할 때 사용

<TableRow>의 개수가 행의 개수가 됨

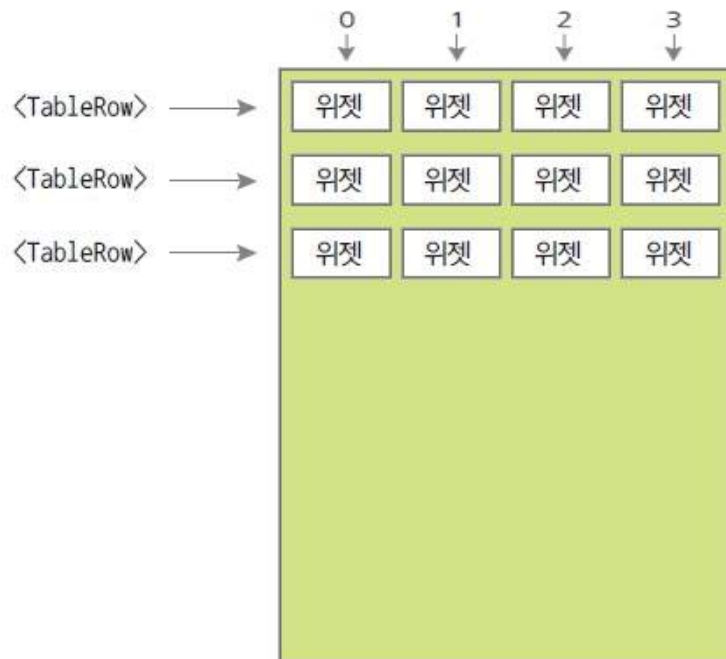
열의 개수는 <TableRow> 안에 포함된
위젯의 개수로 결정

속성

layout_column : 지정된 열에 현재
위젯을 표시

stretchColumns : 지정된 열의 폭을
늘려라

stretchColumns = "*" : 각 셀을 같은
크기로 확장, 전체 화면이 꽉 차는 효과



3. 기타 레이아웃 | TableLayout

예제5-13: Ex05_13
테이블레이아웃 예제

예제 5-13 테이블레이아웃 XML 코드

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

3. 기타 레이아웃 | **TableLayout**

실습5-2: Project05_02

테이블레이아웃을 활용하여 숫자 버튼까지 있는 계산기 작성

activity_main.xml

TableLayout 1개와 TableRow 9개로 구성

에디트텍스트 2개, 숫자 버튼 10개, 연산 버튼 4개,

텍스트뷰 1개를 생성

연산 버튼 위젯에는 layout_margin을 적절히 지정

결과를 보여줄 TextView는 색상을 빨간색, 글자 크기는 20sp

각 위젯의 id는 위에서부터 Edit1, Edit2, BtnNum0~9, BtnAdd, BtnSub, BtnMul, BtnDiv, TextResult



3. 기타 레이아웃 | TableLayout

activity_main.xml

TableLayout에 포함된 위젯은 width, height를 생략해도 됨
(테이블 레이아웃에서는 모든 셀들의 너비가 자동으로 결정됨)

```
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/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:stretchColumns="*"
8   tools:context="com.android.kjs.project05_02.MainActivity">
9
10  <TableRow>
11    <EditText
12      android:id="@+id/Edit1"
13      android:layout_span="5"
14      android:hint="숫자1 입력" />
15  </TableRow>
16
17  <TableRow>
18    <EditText
19      android:id="@+id/Edit2"
20      android:layout_span="5"
21      android:hint="숫자2 입력" />
22  </TableRow>
23
```

3. 기타 레이아웃 | TableLayout

```
24 <TableRow>
25 <Button
26     android:id="@+id/BtnNum0"
27     android:text="0" />
28 <Button
29     android:id="@+id/BtnNum1"
30     android:text="1" />
31 <Button
32     android:id="@+id/BtnNum2"
33     android:text="2" />
34 <Button
35     android:id="@+id/BtnNum3"
36     android:text="3" />
37 <Button
38     android:id="@+id/BtnNum4"
39     android:text="4" />
40 </TableRow>
41
42 <TableRow>
43 <Button
44     android:id="@+id/BtnNum5"
45     android:text="5" />
46 <Button
47     android:id="@+id/BtnNum6"
48     android:text="6" />
49 <Button
50     android:id="@+id/BtnNum7"
51     android:text="7" />
52 <Button
53     android:id="@+id/BtnNum8"
54     android:text="8" />
55 <Button
56     android:id="@+id/BtnNum9"
57     android:text="9" />
58 </TableRow>
59
60 <TableRow>
61 <Button android:id="@+id/BtnAdd"
62     android:layout_margin="5dp"
63     android:layout_span="5"
64     android:text="더하기" />
65 </TableRow>
66 <TableRow>
67 <Button
68     android:id="@+id/BtnSub"
69     android:layout_margin="5dp"
70     android:layout_span="5"
71     android:text="빼기" />
72 </TableRow>
73 <TableRow>
74 <Button
75     android:id="@+id/BtnMul"
76     android:layout_margin="5dp"
77     android:layout_span="5"
78     android:text="곱하기" />
79 </TableRow>
80 <TableRow>
81 <Button
82     android:id="@+id/BtnDiv"
83     android:layout_margin="5dp"
84     android:layout_span="5"
85     android:text="나누기" />
86 </TableRow>
87
88 <TableRow>
89 <TextView
90     android:id="@+id/TextResult"
91     android:layout_margin="5dp"
92     android:layout_span="5"
93     android:text="계산 결과 : "
94     android:textColor="#FF0000"
95     android:textSize="20dp" />
96 </TableRow>
97 </TableLayout>
```

3. 기타 레이아웃 | **TableLayout**

MainActivity.java

전역변수 선언

숫자 버튼을 제외한 activity_main.xml의 7개 위젯에 대응할 위젯변수 7개

입력될 2개 문자열을 저장할 문자열 변수 2개

계산 결과를 저장할 정수 변수 1개

10개 숫자 버튼을 저장할 버튼 배열

10개 버튼의 id를 저장할 정수형 배열

증가 값으로 사용할 정수 변수

3. 기타 레이아웃 | TableLayout

```
12 public class MainActivity extends AppCompatActivity {  
13     EditText edit1, edit2;  
14     Button btnAdd, btnSub, btnMul, btnDiv;  
15     TextView txtResult;  
16     String num1, num2;  
17     Integer result;  
18     //10개 숫자 버튼 리소스를 배열로 처리  
19     Button[] numButtons = new Button[10];  
20     int[] numBtnIDs = {R.id.BtnNum0, R.id.BtnNum1, R.id.BtnNum2, R.id.BtnNum3, R.id.BtnNum4, R.id.BtnNum5,  
21         R.id.BtnNum6, R.id.BtnNum7, R.id.BtnNum8, R.id.BtnNum9};  
22     int i;  
23  
24     @Override  
25     protected void onCreate(Bundle savedInstanceState) {  
26         super.onCreate(savedInstanceState);  
27         setContentView(R.layout.activity_main);  
28         setTitle("테이블레이아웃 계산기");  
29  
30         edit1 = (EditText) findViewById(R.id.Edit1);  
31         edit2 = (EditText) findViewById(R.id.Edit2);  
32  
33         //for문으로 숫자 버튼 뷰들을 참조  
34         for (i = 0; i < numBtnIDs.length; i++) {  
35             numButtons[i] = (Button) findViewById(numBtnIDs[i]);  
36         }  
37  
38         btnAdd = (Button) findViewById(R.id.BtnAdd);  
39         btnSub = (Button) findViewById(R.id.BtnSub);  
40         btnMul = (Button) findViewById(R.id.BtnMul);  
41         btnDiv = (Button) findViewById(R.id.BtnDiv);  
42  
43         txtResult = (TextView) findViewById(R.id.TextResult);
```

int형의 뷰를 배열에 저장하여 객체 참조 시 이용하고
뷰 타입의 배열에 저장

반드시 복습하여!
위 방식으로 resId를 배열로 처리하도록 수정

3. 기타 레이아웃 | TableLayout

각 연산 버튼을 터치했을 때의 이벤트 처리

중복 금물!
반드시 복습하여!
onClick 메소드 등으로 수정

```
45 btnAdd.setOnTouchListener((v, event) -> {  
48     num1 = edit1.getText().toString();  
49     num2 = edit2.getText().toString();  
50     result = Integer.parseInt(num1) + Integer.parseInt(num2);  
51     txtResult.setText("계산 결과 : " + result.toString());  
52     return false;  
53 });  
55  
56 btnSub.setOnTouchListener((v, event) -> {  
59     num1 = edit1.getText().toString();  
60     num2 = edit2.getText().toString();  
61     result = Integer.parseInt(num1) - Integer.parseInt(num2);  
62     txtResult.setText("계산 결과 : " + result.toString());  
63     return false;  
64 });
```

*, / 연산도 동일하게 작성

3. 기타 레이아웃 | TableLayout

숫자 버튼 클릭시 이벤트 처리

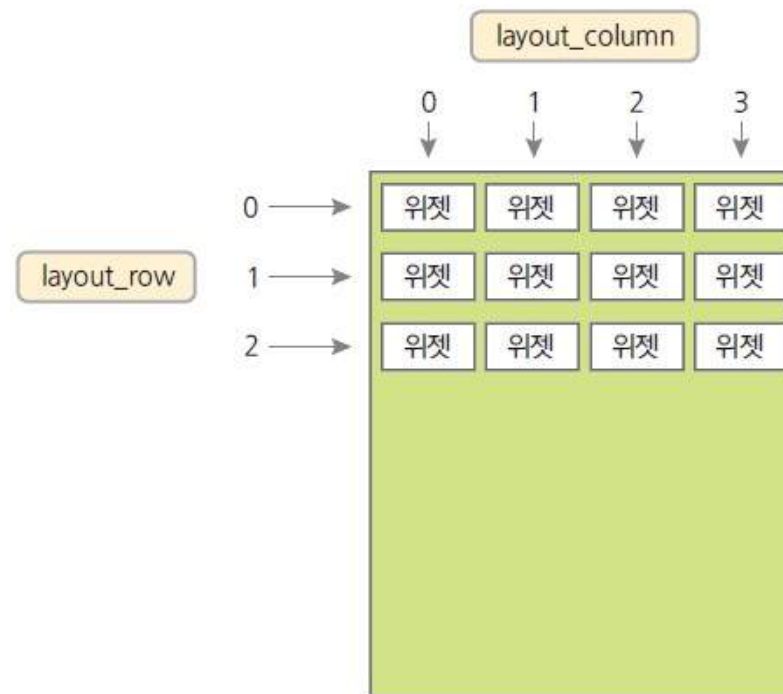
```
89 //숫자 버튼 터치 이벤트 처리
90 for (i = 0; i < numBtnIDs.length; i++) {
91
92     final int index = i;
93
94     numButtons[index].setOnClickListener((v) -> {
95         // 포커스받은 에디트텍스트에 숫자 추가
96         if (edit1.isFocused() == true) {
97             num1 = edit1.getText().toString() + numButtons[index].getText().toString();
98             edit1.setText(num1);
99         } else if (edit2.isFocused() == true) {
100             num2 = edit2.getText().toString() + numButtons[index].getText().toString();
101             edit2.setText(num2);
102         } else {
103             Toast.makeText(getApplicationContext(), "먼저 에디트텍스트를 선택하세요", Toast.LENGTH_SHORT).show();
104         }
105     });
106 }
107
108 }
109
110 }
111 }
```

onCreate의 지역변수이므로 Listener에 전달되지 않음
Listener에 전달하고 싶은 지역변수는 final로 처리

3. 기타 레이아웃 | GridLayout

GridLayout

테이블레이아웃처럼 위젯을 표 형태로 배치할 때 사용하는데 좀 더 직관적임



3. 기타 레이아웃 | GridLayout

그리드레이아웃 속성

<GridLayout>의 대표적 속성

rowCount : 행 개수

columnCount : 열 개수

orientation : 그리드를 수평 방향을 우선할지, 수직 방향을 우선할지를 결정

그리드레이아웃 안에 포함될 위젯의 대표적 속성

layout_row : 자신이 위치할 행 번호(0번부터 시작)

layout_column : 자신이 위치할 열 번호(0번부터 시작)

layout_rowSpan : 행을 지정된 개수만큼 확장

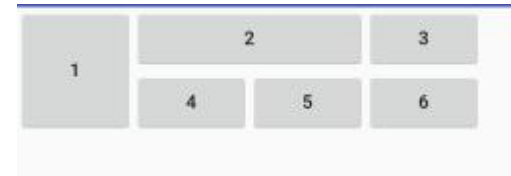
layout_columnSpan : 열을 지정된 개수만큼 확장

layout_gravity : 주로 fill, fill_vertical, fill_horizontal 등으로 지정

행 또는 열 확장시, 위젯을 확장된 셀에 꽉 채우는 효과를 냄

3. 기타 레이아웃 | GridLayout

예제5-18: Ex05_18
그리드레이아웃 예제



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:rowCount="2"
8   android:columnCount="3"
9   tools:context="com.example.kjs.temp.MainActivity">
10   <Button
11     android:text="1"
12     android:layout_row="0"
13     android:layout_column="0"
14     android:layout_rowSpan="2"
15     android:layout_gravity="fill_vertical"/>
16   <Button
17     android:text="2"
18     android:layout_row="0"
19     android:layout_column="1"
20     android:layout_columnSpan="2"
21     android:layout_gravity="fill_horizontal"/>
22   <Button
23     android:layout_row="0"
24     android:layout_column="3"
25     android:text="3"/>
26   <Button
27     android:text="4"
28     android:layout_row="1"
29     android:layout_column="1"/>
30   <Button
31     android:text="5"
32     android:layout_row="1"
33     android:layout_column="2"/>
34   <Button
35     android:text="6"
36     android:layout_row="1"
37     android:layout_column="3"/>
38 </GridLayout>
```

결과는??

3. 기타 레이아웃 | GridLayout

실습(직접풀어보기) 5-5: Test05_05

▶ 직접 풀어보기 5-5

[실습 5-2]를 그리드레이아웃으로 변경해서 실행해보자.

HINT Java 코드는 변경할 필요가 없고, XML만 변경하면 된다. XML 위젯의 id도 동일하게 사용한다.



3. 기타 레이아웃 | **FrameLayout**

FrameLayout

단순히 레이아웃 내의 위젯들은 왼쪽 상단부터 겹쳐서 출력
프레임레이아웃 자체로 사용하기보다는 탭 위젯 등과 혼용해서 사용할 때 유용



3. 기타 레이아웃 | **FrameLayout**

프레임레이아웃의 속성

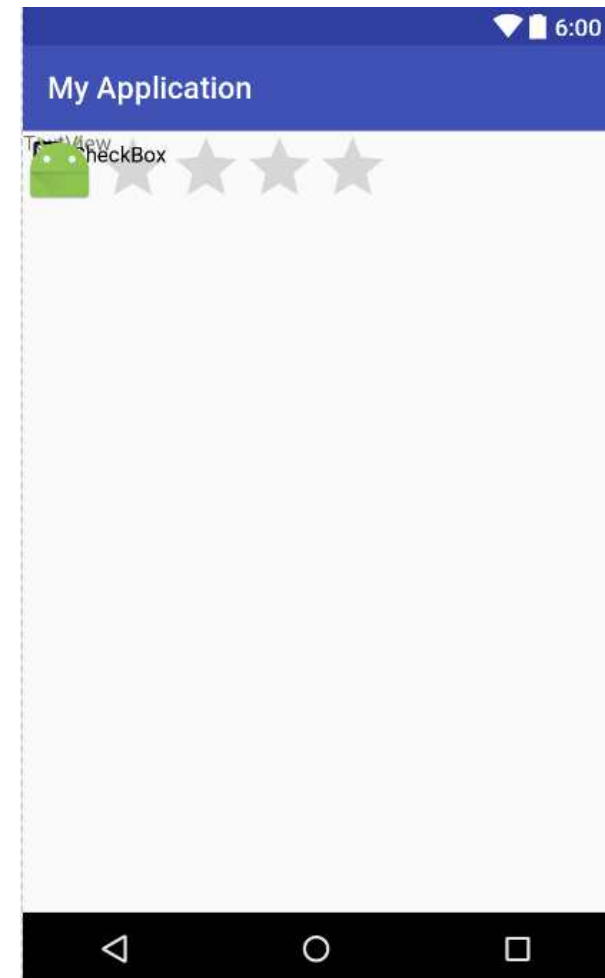
foreground : 프레임레이아웃의 전경 이미지를 지정

foregroundGravity : 전경 이미지의 위치를 지정(fill, right, left, top, bottom 등)

3. 기타 레이아웃 | FrameLayout

예제5-19: Ex05_19
프레임레이아웃 예제

```
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/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:foreground="@mipmap/ic_launcher"
8   android:foregroundGravity="top"
9   tools:context="com.example.user.myapplication.MainActivity">
10
11
12   <TextView
13     android:text="TextView"
14     android:layout_width="wrap_content"
15     android:layout_height="wrap_content"
16     android:id="@+id/textView" />
17
18   <RatingBar
19     android:layout_width="wrap_content"
20     android:layout_height="wrap_content"
21     android:id="@+id/ratingBar" />
22
23   <CheckBox
24     android:text="CheckBox"
25     android:layout_width="wrap_content"
26     android:layout_height="wrap_content"
27     android:id="@+id/checkBox" />
28 </FrameLayout>
29
```



3. 기타 레이아웃 | FrameLayout

실습: MyFrameLayout

‘이미지 바꾸기’ 버튼을 클릭할 때마다, 해당 이미지 출력
FrameLayout에 2개의 ImageView 배치하여 디자인



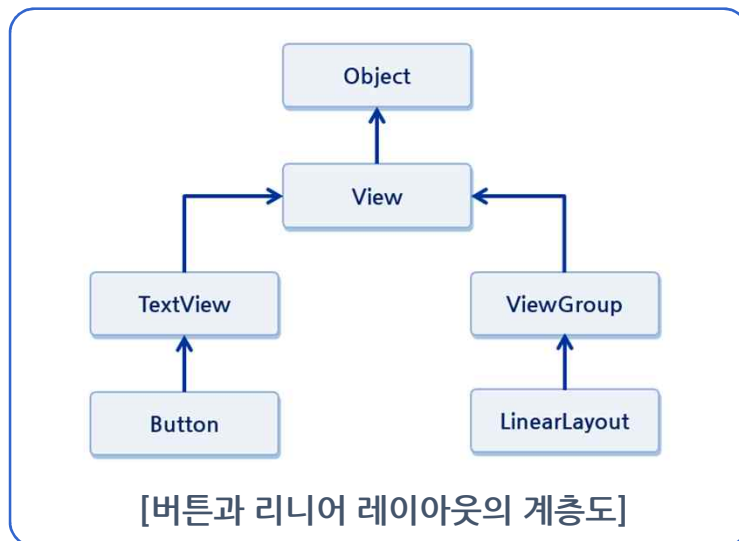
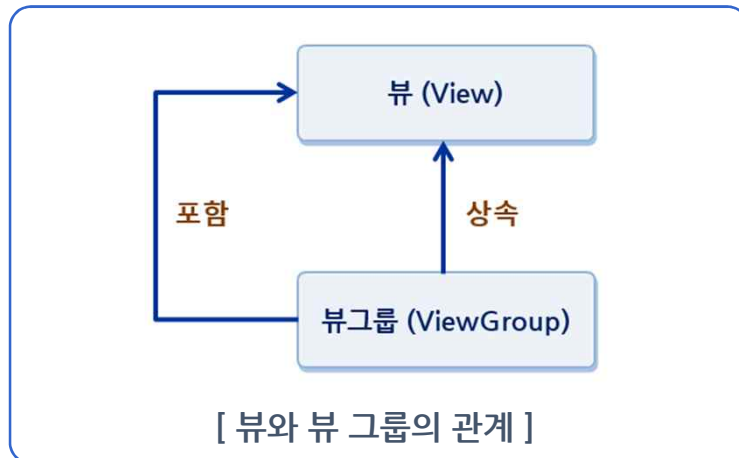
04



ConstraintLayout



ConstraintLayout



뷰(View)

화면에 보이는 각각의 것들 (버튼, 텍스트 등등)
흔히 컨트롤(Control)이나 위젯(Widget)이라 불리는
UI 구성 요소

뷰 그룹(ViewGroup)

뷰들을 여러 개 포함하고 있는 것
뷰 그룹도 뷰에서 상속하여 뷰가 됨
레이아웃, 어댑터 뷰 등

레이아웃(Layout)

뷰 그룹 중에서 내부에 뷰들을 포함하고 있으면서
그것들을 배치하는 역할을 하는 것

위젯(Widget)

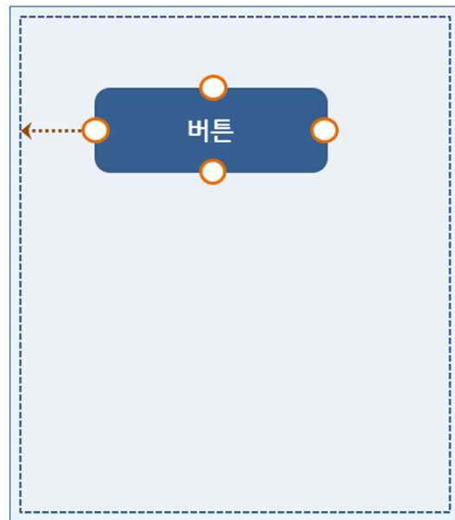
뷰 중에서 일반적인 컨트롤의 역할을 하고 있는 것
버튼, 텍스트뷰 등등

ConstraintLayout

제약레이아웃(ConstraintLayout)

뷰의 위치, 크기를 결정할 때 다른 View에 대한 제약조건을 사용하여 각 View를 배치하는 방식

뷰의 4면의 연결점을 타 뷰의 원하는 면과 연결하면 제약조건 생성



연결점을 연결할 수 있는 타깃

- 같은 레이아웃 안에 들어 있는 다른 뷰의 연결점
- 부모 레이아웃의 연결점
- 가이드라인(Guideline)

ConstraintLayout

상속 관계

ConstraintLayout

```
public class ConstraintLayout
```

```
extends ViewGroup
```

```
java.lang.Object
```

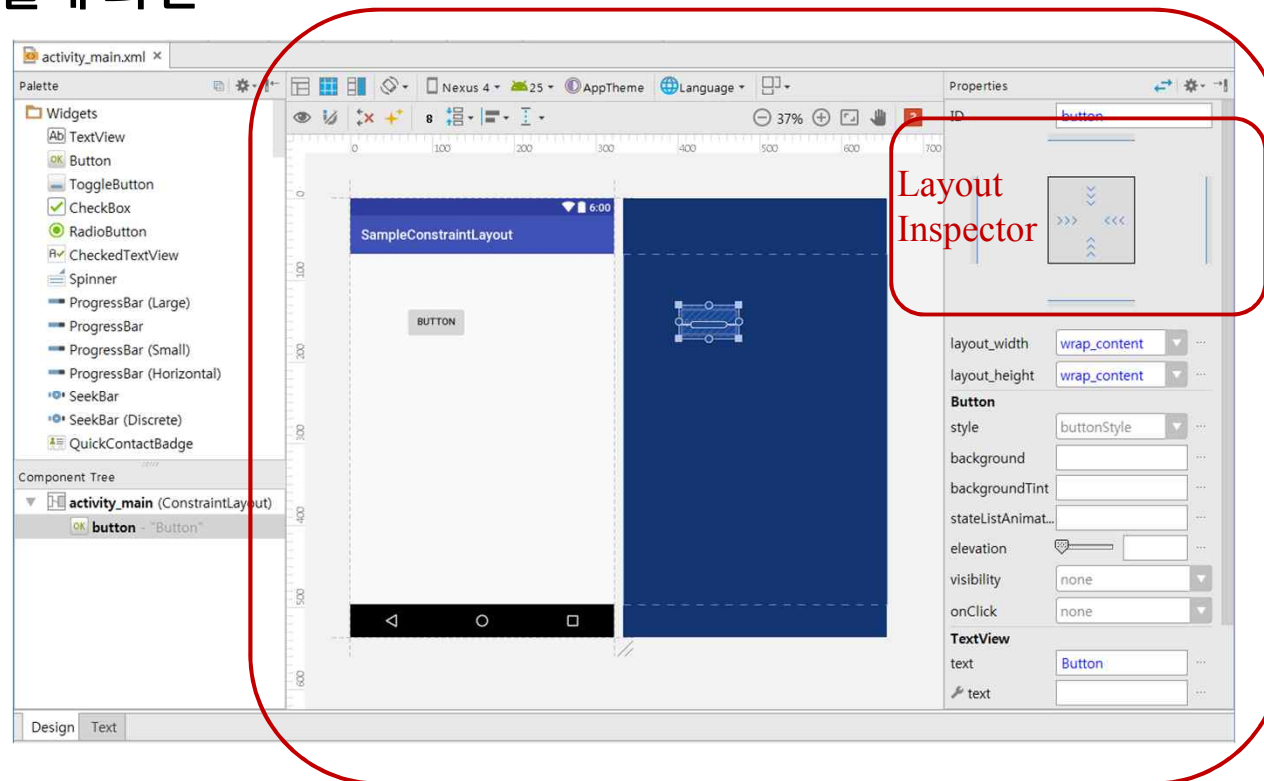
```
↳ android.view.View
```

```
↳ android.view.ViewGroup
```

```
↳ android.support.constraint.ConstraintLayout
```

ConstraintLayout

ConstraintLayout 기본 설계 화면

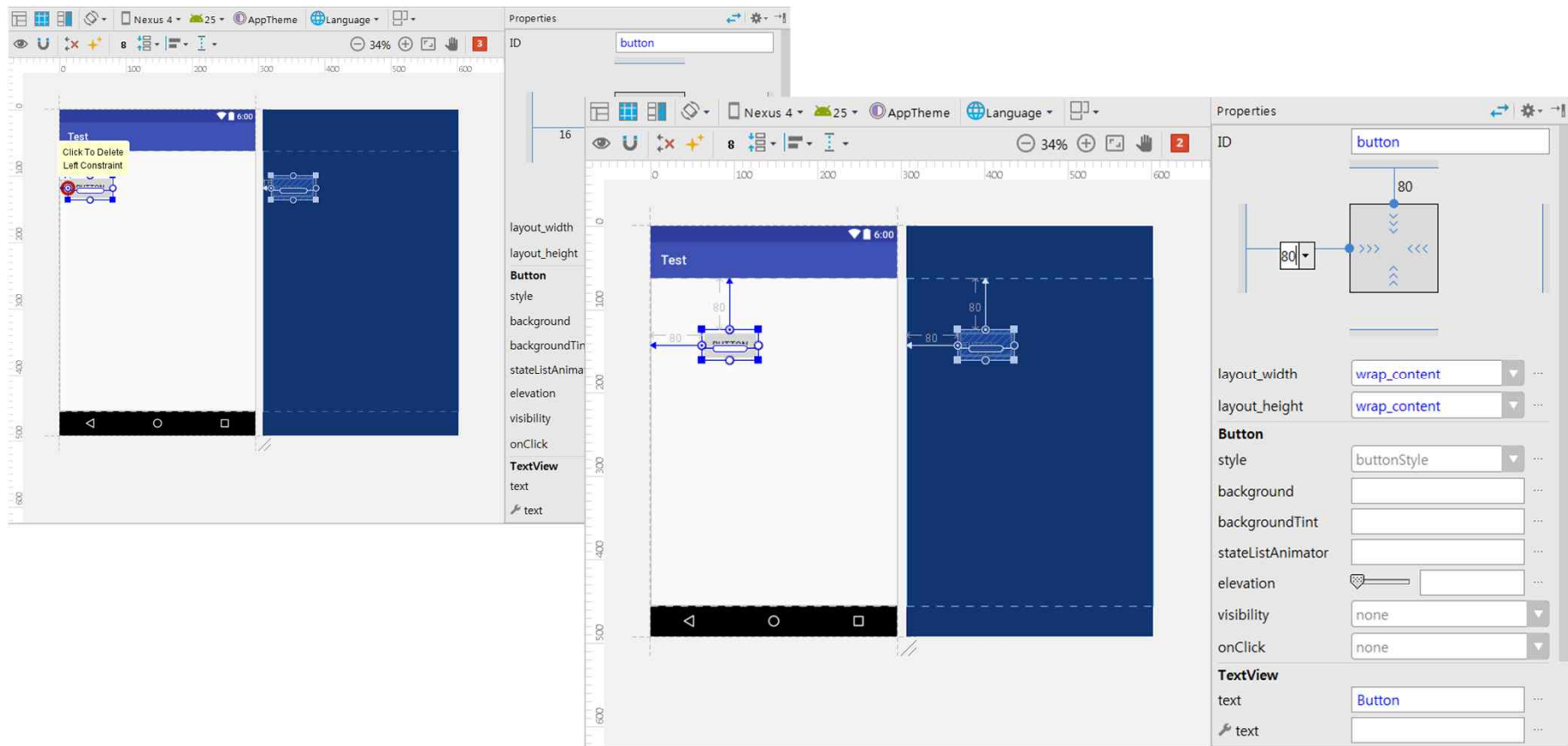


Design(Layout Editor)

ConstraintLayout

ConstraintLayout 기본

왼쪽과 위쪽에 있는 연결점을 부모 레이아웃의 면과 연결

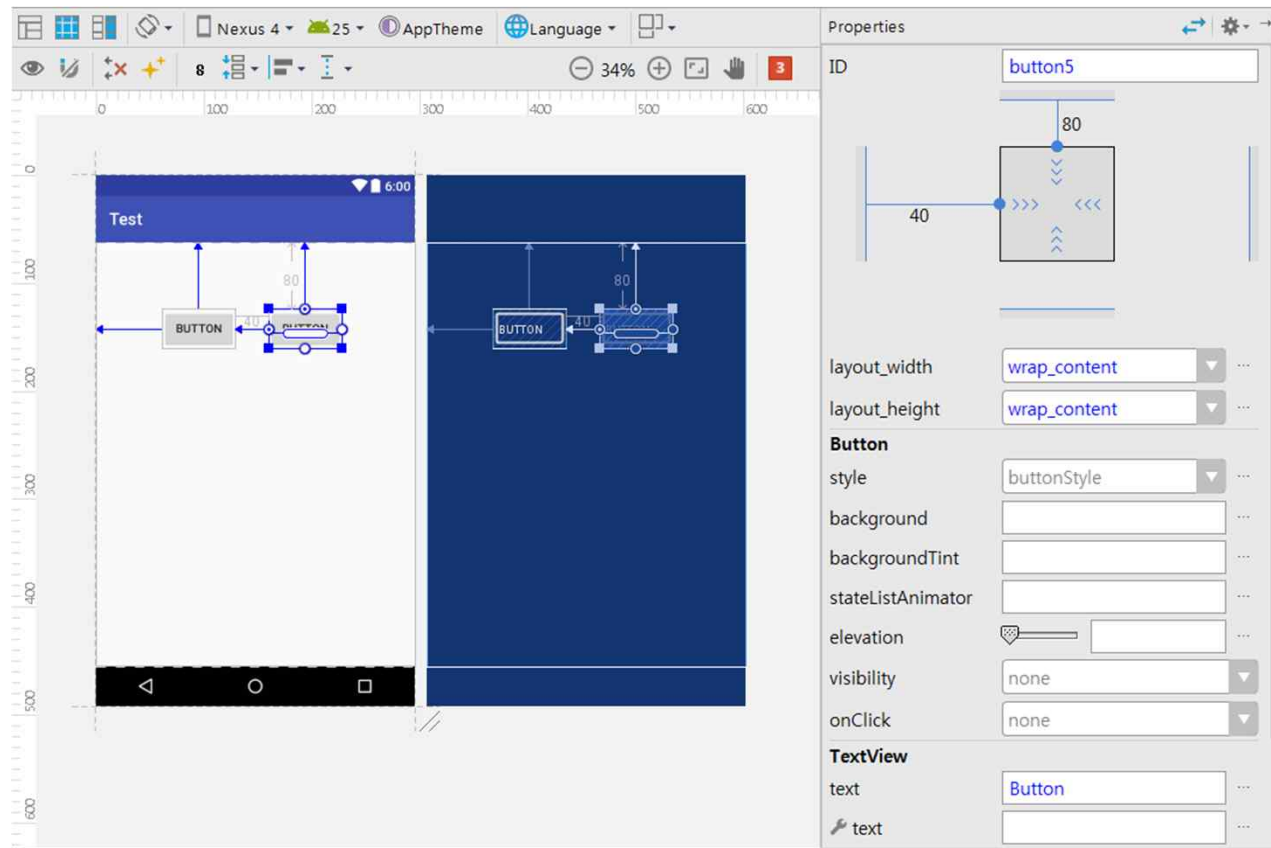


```
app:layout_constraintLeft_toLeftOf="parent"  
app:layout_constraintTop_toTopOf="parent"
```

ConstraintLayout

ConstraintLayout 기본

버튼 추가 후 부모 레이아웃 및 기존 버튼과 연결하여 제약조건 생성



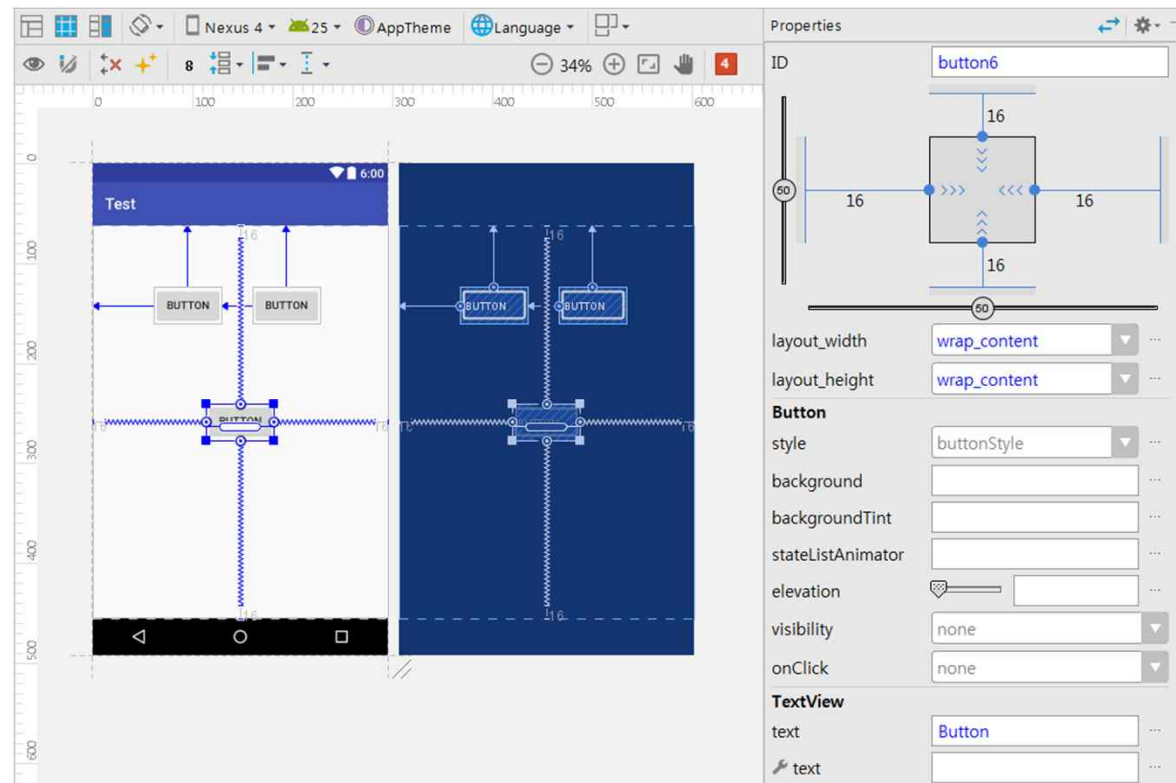
ConstraintLayout

ConstraintLayout 기본

버튼 추가 후 위, 아래, 왼쪽, 오른쪽 모두 부모 레이아웃과 연결선 생성

좌우 또는 상하를 쌍으로 연결하면 그 가운데에 배치됨

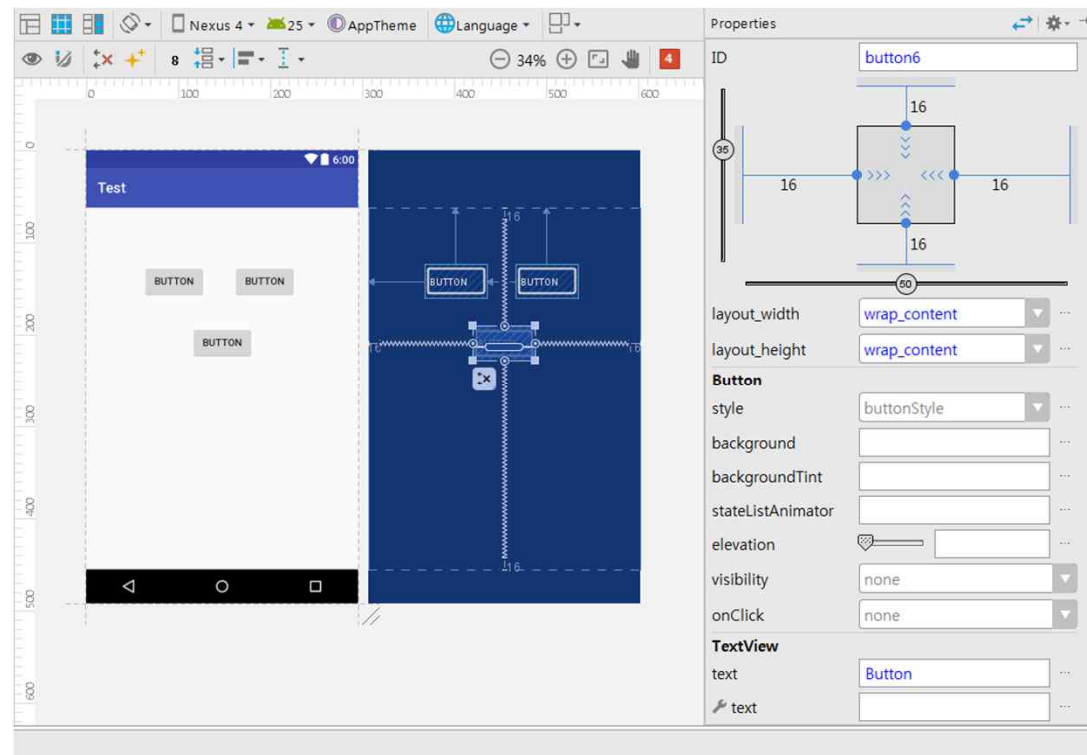
*** 사면 중 2개 면의 제약조건은 필수로 형성되어야 함**



ConstraintLayout

ConstraintLayout 기본

양쪽 또는 상하로 연결된 경우(해당 측면의 가운데 위치하는 경우)
layout Inspector 창에서 바이어스(Bias)를 드래그하여 위치 조정

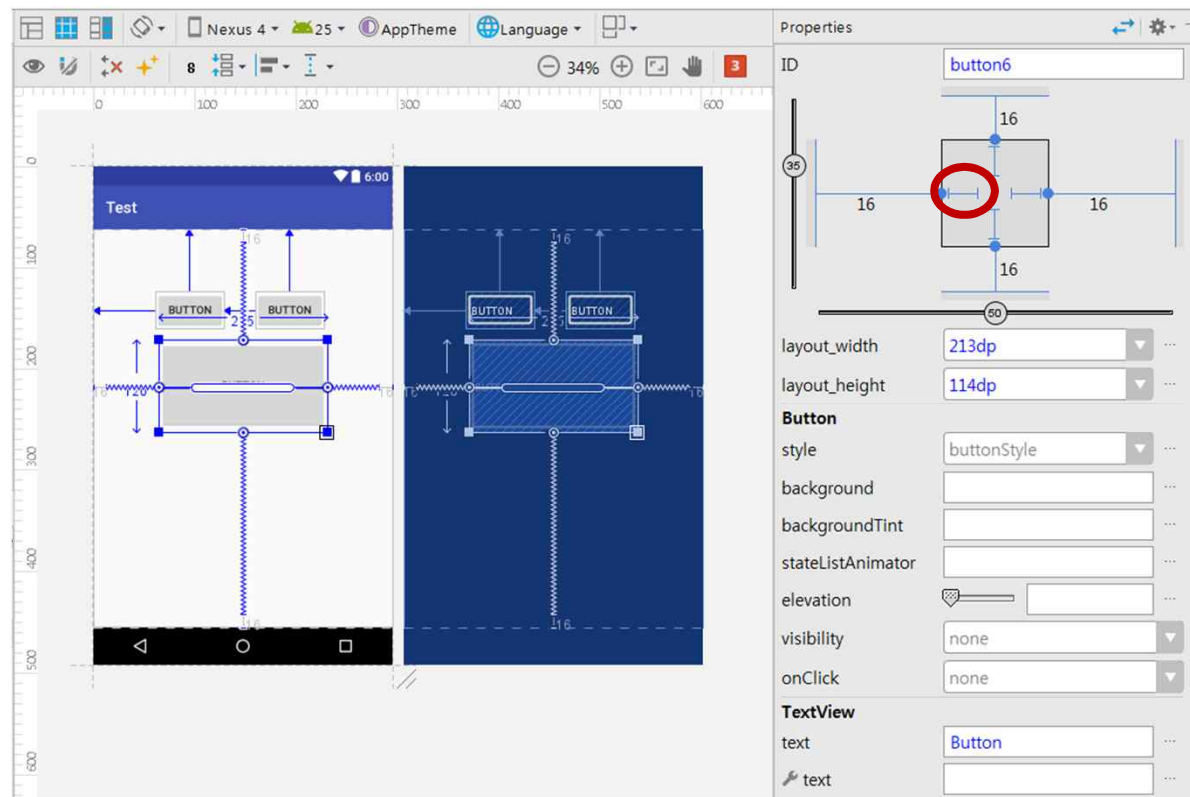


`app:layout_constraintVertical_bias="0.36"`

ConstraintLayout

ConstraintLayout 기본

모서리에 있는 핸들(Handle)을 이용해 뷰의 크기 조정

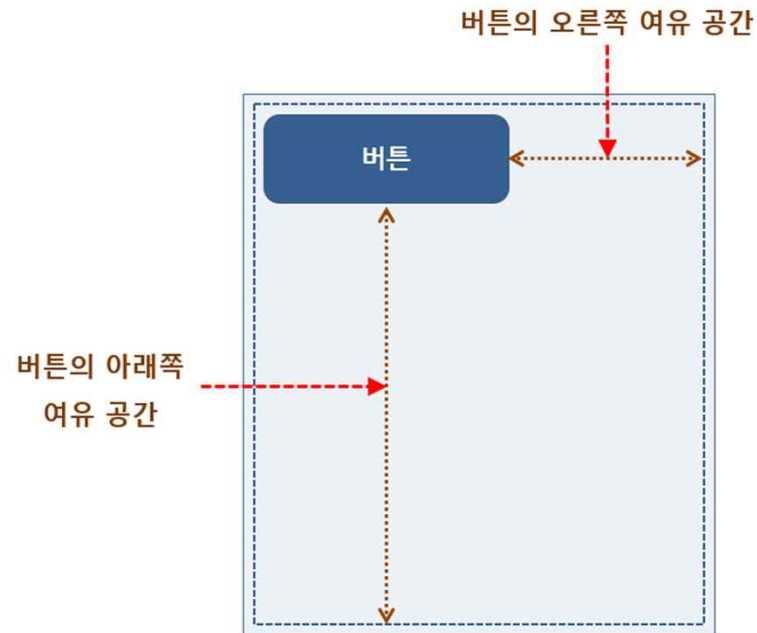


3가지 속성
Fixed,
match_parent,
wrap_parent

ConstraintLayout

ConstraintLayout 기본

부모 레이아웃 안에서 뷰가 차지할 수 있는 여유공간은 레이아웃에 의해 결정됨

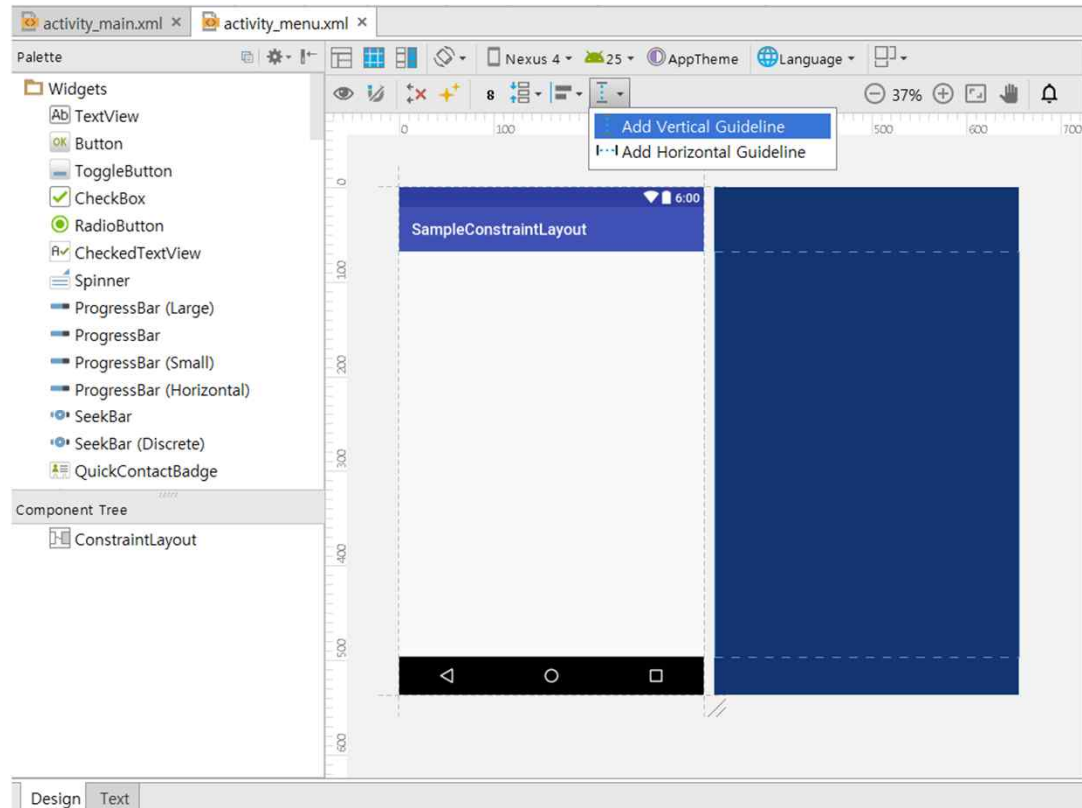
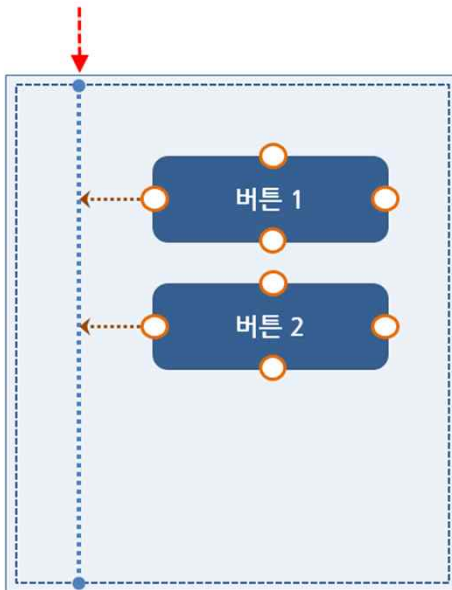


ConstraintLayout

ConstraintLayout 기본

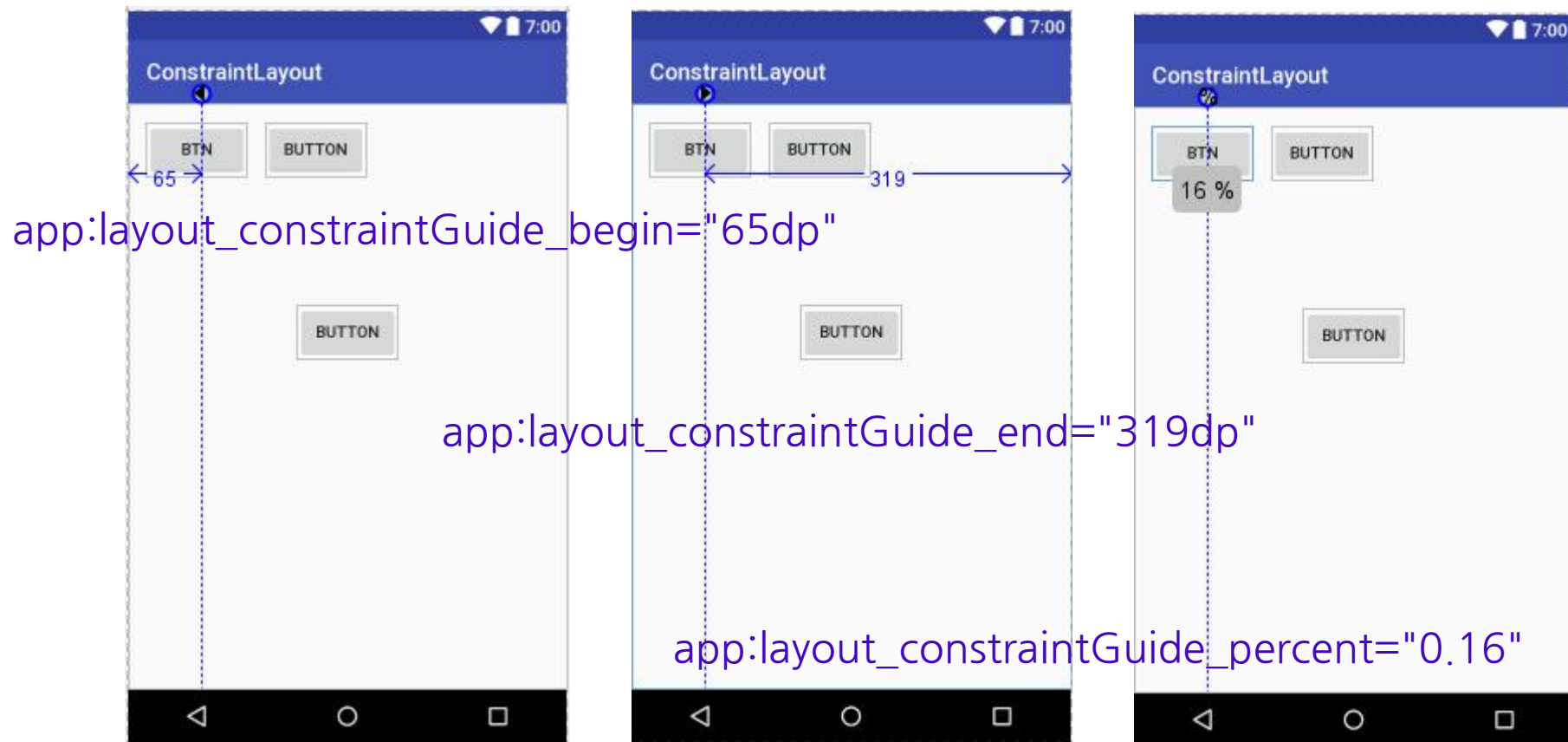
가이드라인: 일정 기준선으로 뷰를 정렬할 때나 기준선에 맞추어 추가할 때 사용됨

세로방향 가이드라인



ConstraintLayout

ConstraintLayout 기본 가이드라인의 단위 조정

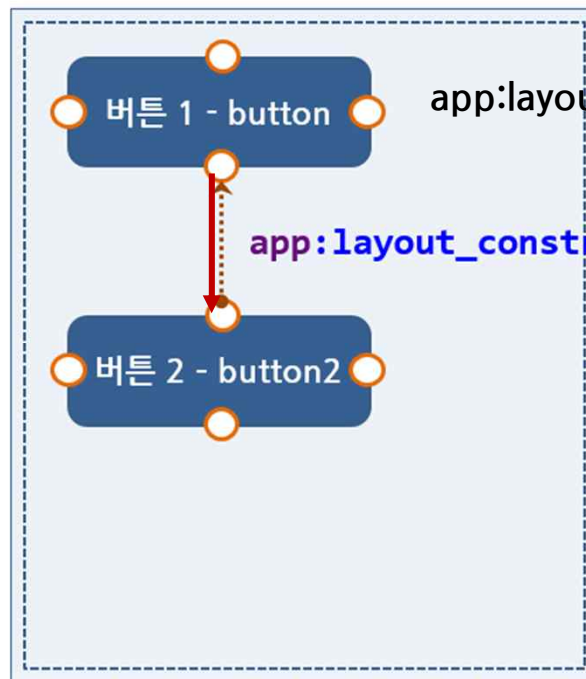


ConstraintLayout

xml 코드

app: support 라이브러리에서 지원

```
layout_constraint[소스 뷰의 연결점]_[타겟 뷰의 연결점]="[타겟 뷰의 id]"
```



[cross-chaining]

app:layout_constraintBottom_toTopOf="@+id/button2"

app:layout_constraintTop_toBottomOf="@+id/button"

ConstraintLayout

실습: MyLogin

1개의 가이드라인을 이용
상단엔 LinearLayout을 사용
EditText, Button, TextView로 정의

TextView에 아이콘 추가
android:drawableStart=
"@drawable/ic_person_black_24dp"

