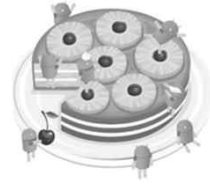


Ch05-레이아웃 익히기



2025-01-01(분반)

# AI 오픈소스 전문프로젝트

Fri 09:00~ (S4-1-201)

## 학습목표

- 레이아웃의 개념을 이해한다.
- 화면을 다양한 레이아웃으로 구성한다.
- Java 코드만으로 화면을 작성해본다.

## 목차

01 레이아웃의 개요

02 리니어레이아웃

03 기타 레이아웃

## 01

### 레이아웃의 개요

## 1. 레이아웃 기본 개념

### ■ 레이아웃

```
java.lang.Object
└ android.view.View
    └ android.widget.ViewGroup
        └ android.widget.LinearLayout
            └ android.widget.TableLayout
        └ android.widget.RelativeLayout
        └ android.widget.FrameLayout
        └ android.widget.GridLayout
```

레이아웃 계층도

- ViewGroup 클래스로부터 상속받으며 내부에 무엇을 담는 용도로 사용
- 레이아웃 중에서 가장 많이 사용되는 것은 리니어레이아웃(LinearLayout)

5 / 57

## 1. 레이아웃 기본 개념

### ■ 레이아웃에서 자주 사용되는 속성

- **orientation** : 레이아웃 안에 배치할 위젯의 수직 또는 수평 방향을 설정
- **gravity** : 레이아웃 안에 배치할 위젯의 정렬 방향을 좌측, 우측, 중앙으로 설정
- **padding** : 레이아웃 안에 배치할 위젯의 여백을 설정
- **layout\_weight** : 레이아웃이 전체 화면에서 차지하는 공간의 가중값을 설정, 여러 개의 레이아웃이 중복될 때 주로 사용
- **baselineAligned** : 레이아웃 안에 배치할 위젯을 보기 좋게 정렬

6 / 57

## 2. 레이아웃의 종류

### ■ 리니어레이아웃 (선형 레이아웃)

- 왼쪽 위부터 아래쪽 또는 오른쪽으로 차례로 배치

### ■ 렐러티브레이아웃 (상대 레이아웃)

- 위젯 자신이 속한 레이아웃의 상하좌우의 위치를 지정하여 배치
- 다른 위젯으로부터 상대적인 위치 지정

### ■ 테이블레이아웃

- 위젯을 행과 열의 개수를 지정한 테이블 형태로 배열

7 / 57

## 2. 레이아웃의 종류

### ■ 그리드레이아웃

- 테이블레이아웃과 비슷하지만, 행 또는 열을 확장하여 다양하게 배치할 때 더 편리함

### ■ 프레임레이아웃

- 위젯들을 왼쪽 위에 일률적으로 겹쳐서 배치하여 중복해서 보이는 효과를 냄

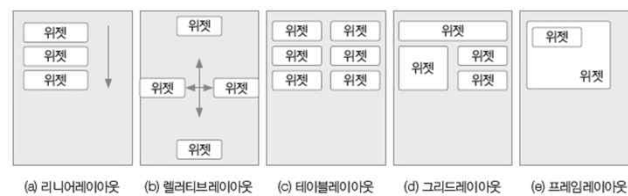


그림 5-1 레이아웃의 종류

8 / 57

## 02

# 리니어레이아웃

### 1. 기본 리니어레이아웃의 형태

#### ■ 리니어레이아웃

- 안드로이드 프로젝트 처음 생성 시 activity.xml 파일 생성
- activity.xml 안에 텍스트뷰 1개가 기본 생성

#### ■ orientation 속성

- 리니어레이아웃의 가장 기본적인 속성
- **vertical** : 리니어레이아웃 안에 포함될 위젯의 배치를 수직방향으로 쌓음
- **horizontal** : 수평 방향으로 쌓겠다는 의미

```
<LinearLayout 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:orientation="vertical" >


    ~~~~ 이곳에 위젯 배치 ~~~~

</LinearLayout>
```

## 1. 기본 리니어레이아웃의 형태

예제 5-1 orientation 속성이 vertical인 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>
```



11 / 57

## 1. 기본 리니어레이아웃의 형태

예제 5-2 orientation 속성이 horizontal인 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   ~~~~ 생략 ~~~~
10 </LinearLayout>
```



12 / 57

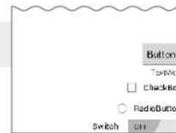
## 1. 기본 리니어레이아웃의 형태

### ■ gravity 속성

- gravity 속성은 레이아웃 안의 위젯을 어디에 배치할 것인지를 결정함

예제 5-3 gravity 속성의 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   ~~~ 생략 ~~~
11 </LinearLayout>
```



13 / 57

## 1. 기본 리니어레이아웃의 형태

### ■ layout\_gravity 속성

- layout\_gravity는 자신의 위치를 부모의 어디쯤에 위치시킬지를 결정

예제 5-4 layout\_gravity 속성의 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>
```



14 / 57

## 1. 기본 리니어레이아웃의 형태

### ▶ 직접 알아보기 5-1

리니어레이아웃으로 오른쪽과 같은 화면을 구성하는 XML을 작성하라.

- 리니어레이아웃의 orientation 속성을 vertical로 설정한다.
- 버튼 3개를 생성하고 버튼의 layout\_width는 110dp로, layout\_height는 100dp로 한다.

**힌트** 버튼에 gravity와 layout\_gravity 속성을 모두 지정해야 한다.



그림 5-2 리니어레이아웃 활용

15 / 57

## 1. 기본 리니어레이아웃의 형태

### ■ baselineAligned 속성

- baselineAligned 속성은 크기가 다른 위젯들을 보기 좋게 정렬함
- true와 false 값을 가질 수 있음



(a) false로 지정

(b) 생략하거나 true로 지정

그림 5-3 baselineAligned 속성

16 / 57



## 2. 중복 리니어레이아웃의 형태

### ■ 중복 리니어레이아웃

- 리니어레이아웃 안에 리니어레이아웃을 생성하는 방식



17 / 57

## 2. 중복 리니어레이아웃의 형태

### ■ layout\_weight 속성

- 리니어레이아웃을 여러 개 사용할 경우 각 레이아웃의 크기를 지정할 때 사용
- 주로 전체 화면에 대한 비율(%)로 지정

18 / 57

## 2. 중복 리니어레이아웃의 형태

### ■ layout\_weight 속성

- 큰 리니어레이아웃 안에 3개의 리니어레이아웃을 사용하고 각 레이아웃에 버튼을 2개씩 넣은 화면 구성하기

예제 5-5 3개의 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" >
```



19 / 57

## 2. 중복 리니어레이아웃의 형태

```
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>
```

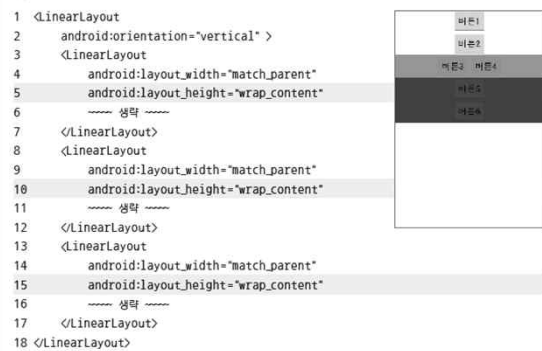
20 / 57

## 2. 중복 리니어레이아웃의 형태

### ■ layout\_weight 속성

- 첫 번째 레이아웃의 버튼만 보이는 문제 해결하기
  - android:layout\_height="match\_parent"를 android:layout\_height="wrap\_content"로 변경
  - 레이아웃마다 구분되어 보이도록 내부에 있는 3개의 레이아웃에 background 속성 지정

예제 5-6 layout\_height를 wrap\_content로 변경



21 / 57

## 2. 중복 리니어레이아웃의 형태

### ■ layout\_weight 속성

- 레이아웃이 화면을 꽉 채우도록 수정하기
  - layout\_weight 속성 사용

예제 5-7 layout\_weight를 1로 지정



22 / 57

## 2. 중복 리니어레이아웃의 형태

### ▶ 직접 알아보기 5-2

리니어레이아웃으로 오른쪽과 같은 화면을 구성하는 XML을 작성하라. 단, 레이아웃이 구분되어 보이도록 각각 다른 색으로 지정한다.

**힌트** 레이아웃 안에 레이아웃을 여러 번 중첩해도 된다



그림 5-5 중첩 리니어레이아웃

23 / 57

## 3. Java 코드로 화면 만들기

### ■ 일반적인 안드로이드 프로젝트의 구성

- XML로 화면을 구성하고 Java 코드로 화면에 출력하는 형태

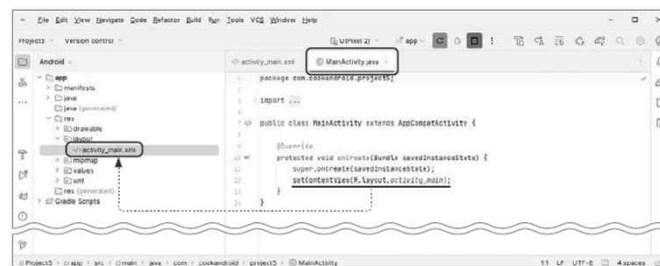


그림 5-6 XML과 Java 코드의 동작

24 / 57

### 3. Java 코드로 화면 만들기

- 실습 5-1 XML 파일 없이 화면 코딩하기
- 1 안드로이드 프로젝트 생성
  - (1) 프로젝트 이름 : Project5\_1
  - (2) 패키지 이름 : com.cookandroid.project5\_1
- 2 화면 디자인 및 편집
  - (1) activity\_main.xml 삭제



그림 5-7 Java 코드로만 사용한 프로젝트

25 / 57

### 3. Java 코드로 화면 만들기

- 실습 5-1 XML 파일 없이 화면 코딩하기
- 3 Java 코드 작성 및 수정
  - (1) activity\_main.xml 삭제했기 때문에 오류가 발생함
  - (2) 오류 행 앞에 //를 붙여 주석으로 처리한 후 진행
  - (3) 리니어레이아웃을 생성하는 코드를 작성하고 실행

26 / 57

### 3. Java 코드로 화면 만들기

예제 5-8 리니어레이아웃을 생성하는 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     // setContentView(R.layout.activity_main);
4
5     LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
6         LinearLayout.LayoutParams.MATCH_PARENT,
7         LinearLayout.LayoutParams.MATCH_PARENT);
8
9     LinearLayout baseLayout = new LinearLayout(this);
10    baseLayout.setOrientation(LinearLayout.VERTICAL);
11    baseLayout.setBackgroundColor(Color.rgb(0, 255, 0));
12    setContentView(baseLayout, params);
13 }
```



27 / 57

### 3. Java 코드로 화면 만들기

- 실습 5-1 XML 파일 없이 화면 코딩하기
- 3 Java 코드 작성 및 수정
  - (4) 버튼 만들기
  - (5) 버튼을 클릭했을 때 토스트 메시지를 작성 (onCreate( ) 안에 이어서 코딩)

예제 5-9 버튼을 생성하는 Java 코드

```
1 Button btn = new Button(this);
2 btn.setText("버튼입니다");
3 btn.setBackgroundColor(Color.MAGENTA);
4 baseLayout.addView(btn);
5
6 btn.setOnClickListener(new View.OnClickListener() {
7     public void onClick(View arg0) {
8         Toast.makeText(getApplicationContext(),
9             "코드로 생성한 버튼입니다", Toast.LENGTH_SHORT).show();
10    }
11 });
```

28 / 57

### 3. Java 코드로 화면 만들기

#### ▶ 직접 알아보기 5-3

오른쪽과 같은 화면을 XML 파일 없이 Java 코드로 작성하라.

- 레이아웃에 에디트텍스트 1개, 버튼 1개, 텍스트뷰 1개를 생성한다.
- 버튼을 클릭하면 에디트텍스트에 쓰인 문자열이 텍스트 뷰에 나타나게 한다.

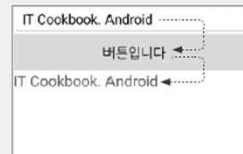


그림 5-8 XML 파일 없이 프로젝트 생성

29 / 57

# 03

## 기타 레이아웃

## 1. 렐러티브레이아웃

- 렐러티브레이아웃에 있는 위젯의 위치 속성
  - 렐러티브레이아웃의 상하좌우에 배치
  - 렐러티브레이아웃의 다른 위젯의 상대 위치에 배치
- 렐러티브레이아웃의 상하좌우에 배치
  - 각 속성값은 true 또는 false로 지정

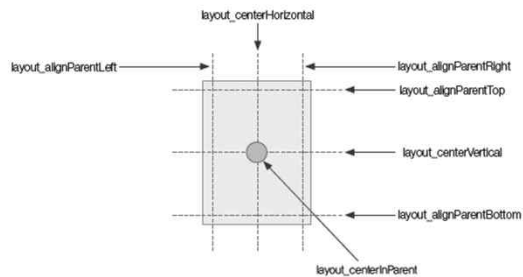


그림 5-9 부모(레이아웃)의 위치를 적용할 때의 속성

31 / 57

## 1. 렐러티브레이아웃

예제 5-10 렐러티브레이아웃의 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>
    
```



32 / 57



## 1. 렐러티브레이아웃

### ■ 다른 위젯의 상대 위치에 배치

- 각 속성의 값은 다른 위젯의 id를 지정
- “@+id/기준 위젯의 아이디”와 같은 형식으로 사용

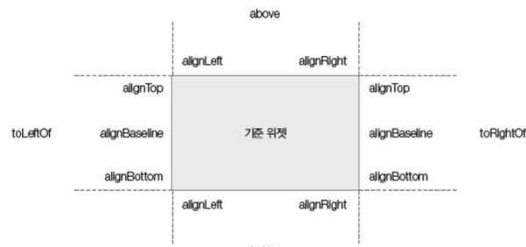


그림 5-10 다른 위젯의 상대적인 위치를 적용할 때의 속성

33 / 57

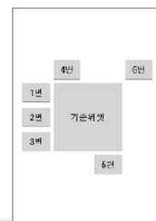
## 1. 렐러티브레이아웃

예제 5-11 렐러티브레이아웃의 XML 코드 2

```

1 <RelativeLayout xmlns:android="http://~"
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="기준위젯" />
11   <Button
12       android:layout_alignTop="@+id/baseBtn"
13       android:layout_toLeftOf="@+id/baseBtn"
14       android:text="1번" />
15   ----- 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>

```



34 / 57

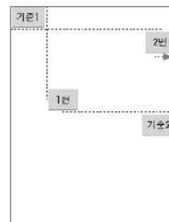
## 1. 렐러티브레이아웃

예제 5-12 렐러티브레이아웃 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>

```



35 / 57

## 1. 렐러티브레이아웃

### ▶ 직접 알아보기 5-4

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



그림 5-11 레이아웃 비교

36 / 57

## 2. 테이블레이아웃

### ■ 테이블레이아웃(TableLayout)

- 주로 위젯을 표 형태로 배치할 때 사용함
- <TableRow>와 함께 사용되는데 <TableRow>의 개수가 바로 행의 개수가 됨
- 열의 개수는 <TableRow> 안에 포함된 위젯의 수로 결정. 3행 4열의 테이블레이아웃

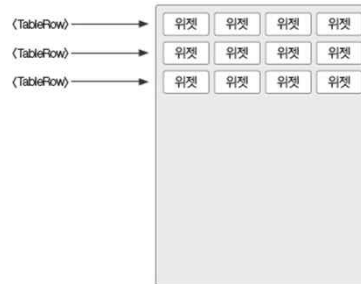


그림 5-12 테이블레이아웃의 개념

37 / 57

## 2. 테이블레이아웃

### ■ 테이블레이아웃의 속성

- `layout_column` : 지정된 열에 현재 위젯을 표시함
- `stretchColumns` : 지정된 열의 폭을 늘림
- `stretchColumns = "*" : 각 셀을 같은 크기로 확장, 전체 화면이 꽉 차는 효과`

38 / 57

## 2. 테이블레이아웃

예제 5-13 테이블레이아웃의 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

39 / 57

## 2. 테이블레이아웃

- 실습 5-2 테이블레이아웃 계산기 앱 만들기
  - 테이블레이아웃을 활용하여 숫자 버튼이 있는 계산기
- 1 안드로이드 프로젝트 생성
  - (1) 프로젝트 이름 : Project5\_2
  - (2) 패키지 이름 : com.cookandroid.project5\_2

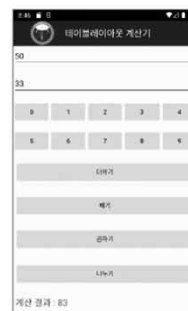


그림 5-13 테이블레이아웃 계산기 앱 결과 화면

40 / 57

## 2. 테이블레이아웃

### ■ 실습 5-2 테이블레이아웃 계산기 앱 만들기

#### ■ 2 화면 디자인 및 편집

- (1) 다음 규칙에 따라 화면 구성하기
  - TableLayout 1개와 TableRow 9개로 구성
  - 에디트텍스트 2개, 숫자 버튼 10개, 연산 버튼 4개, 텍스트뷰 1개 생성
  - 연산 버튼 위젯에는 layout\_margin을 적절히 지정
  - 결과를 보여줄 TextView는 색상을 빨간색, 글자 크기는 20dp
  - 각 위젯의 id는 위에서부터 Edit1, Edit2, BtnNum0~9, BtnAdd, BtnSub, BtnMul, BtnDiv, TextResult

41 / 57

## 3. 기타 레이아웃 ▶ 테이블레이아웃

예제 5-14 activity\_main.xml

```

1 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent" >
4
5     <TableRow>
6         <EditText
7             android:id="@+id/Edit1"
8             android:layout_span="5"
9             android:hint="숫자1 입력" />
10    </TableRow>
11
12    ~~~~ 생략(TableRow 1개, 에디트텍스트 1개) ~~~~
13
14    <TableRow>
15        <Button
16            android:id="@+id/BtnNum0"
17            android:text="0" />
18        ~~~~ 생략(숫자 버튼 4개) ~~~~
19    </TableRow>
20
21    <TableRow>
22        ~~~~ 생략(숫자 버튼 5개) ~~~~
23    </TableRow>
24

```



42 / 57

### 3. 기타 레이아웃 ▶ 테이블레이아웃

```
25 <TableRow>
26     <Button
27         android:id="@+id/BtnAdd"
28         android:layout_margin="5dp"
29         android:layout_span="5"
30         android:text="더하기" />
31 </TableRow>
32
33 ~~~~ 생략(TableRow 3개, 연산 버튼 3개) ~~~~
34
35 <TableRow>
36     <TextView
37         android:id="@+id/TextResult"
38         android:layout_margin="5dp"
39         android:layout_span="5"
40         android:text="계산 결과 : "
41         android:textColor="#F00000"
42         android:textSize="20dp" />
43 </TableRow>
44 </TableLayout>
```

43 / 57

## 2. 테이블레이아웃

### ■ 실습 5-2 테이블레이아웃 계산기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

##### ▪ (1) 전역변수 선언

- 숫자 버튼을 제외한 activity\_main.xml의 7개 위젯에 대응할 위젯 변수 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
- 증가값으로 사용할 정수 변수

44 / 57

## 2. 테이블레이아웃

예제 5-15 Java 코드 1

```
1  ~~~~ 생략 ~~~~
2  public class MainActivity extends AppCompatActivity {
3      EditText edit1, edit2;
4      Button btnAdd, btnSub, btnMul, btnDiv;
5      TextView textResult;
6      String num1, num2;
7      Integer result;
8      Button[] numButtons = new Button[10];
9      Integer[] numBtnIDs = { R.id.BtnNum0, R.id.BtnNum1, R.id.BtnNum2, R.id.BtnNum3,
10         R.id.BtnNum4, R.id.BtnNum5, R.id.BtnNum6, R.id.BtnNum7,
11         R.id.BtnNum8, R.id.BtnNum9};
12     int i;
13
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         ~~~~ 생략 ~~~~
```

45 / 57

## 2. 테이블레이아웃

- 실습 5-2 테이블레이아웃 계산기 앱 만들기
- 3 Java 코드 작성 및 수정
  - (2) onCreate( ) 내부 코딩
    - 숫자 버튼이 없다고 가정하고 연산 버튼을 터치했을 때의 내용을 코딩

46 / 57

## 2. 테이블레이아웃

예제 5-16 Java 코드 2

```
1  ~~~~ 생략 ~~~~
2  public void onCreate(Bundle savedInstanceState) {
3      super.onCreate(savedInstanceState);
4      setContentView(R.layout.activity_main);
5
6      setTitle("테이블레이아웃 계산기");
7
8      edit1 = (EditText) findViewById(R.id.Edit1);
9      edit2 = (EditText) findViewById(R.id.Edit2);
10     btnAdd = (Button) findViewById(R.id.BtnAdd);
11     ~~~~ 생략(연산 버튼 3개) ~~~~
12     textResult = (TextView) findViewById(R.id.TextResult);
13
14     btnAdd.setOnClickListener(new View.OnClickListener() {
15         public boolean onTouch(View arg0, MotionEvent arg1) {
16             num1 = edit1.getText().toString();
17             num2 = edit2.getText().toString();
18             result = Integer.parseInt(num1) + Integer.parseInt(num2);
19             textResult.setText("계산 결과 : " + result.toString());
20             return false;
21         }
22     });
23     ~~~~ 생략(연산 버튼 3개 터치 이벤트 리스너) ~~~~
24 }
```

47 / 57

## 2. 테이블레이아웃

### ■ 실습 5-2 테이블레이아웃 계산기 앱 만들기

#### ■ 3 Java 코드 작성 및 수정

- (3) 숫자 버튼 10개를 배열 변수에 대입한 후에 각 버튼의 클릭 이벤트 리스너를 만들기
  - 10개의 버튼을 만들기 위해 배열 사용 → for문 사용
  - onCreate() 안에 작성

48 / 57



## 2. 테이블레이아웃

예제 5-17 Java 코드 3

```

1 for (i = 0; i < numBtnIDs.length; i++) {
2     numButtons[i] = (Button) findViewById(numBtnIDs[i]);
3 }
4
5 for (i = 0; i < numBtnIDs.length; i++) {
6     final int index; // 주의! 꼭 필요함
7     index = i;
8
9     numButtons[index].setOnClickListener(new View.OnClickListener() {
10        public void onClick(View v) {
11
12            if (edit1.isFocused() == true) {
13                num1 = edit1.getText().toString()
14                    + numButtons[index].getText().toString();
15                edit1.setText(num1);
16            } else if (edit2.isFocused() == true) {
17                num2 = edit2.getText().toString()
18                    + numButtons[index].getText().toString();
19                edit2.setText(num2);
20            } else {
21                Toast.makeText(getApplicationContext(),
22                    "먼저 에디트텍스트를 선택하세요", Toast.LENGTH_SHORT).show();
23            }
24        }
25    });
26
27 }

```

49 / 57

## 2. 테이블레이아웃

### ■ 실습 5-2 테이블레이아웃 계산기 앱 만들기

#### ■ 4 프로젝트 실행 및 결과 확인

- (1) 에디트텍스트에 포커스가 없는 상태에서 숫자 버튼을 클릭하면 메시지가 나타남



그림 5-14 에디트텍스트에 포커스가 없는 상태에서 클릭

50 / 57

### 3. 그리드레이아웃

#### ■ 그리드레이아웃(GridLayout)

- 테이블레이아웃처럼 위젯을 표 형태로 배치할 때 사용하지만 좀 더 직관적
  - ex) 2행 3열 (인덱스가 0부터 시작)
    - layout\_row 속성 : 1 / layout\_column 속성 : 2
- Android 4.0(아이스크림 샌드위치, API 14)부터 지원

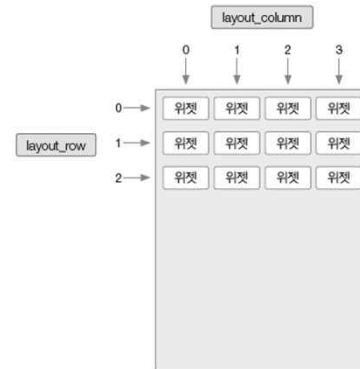


그림 5-15 그리드레이아웃의 개념

51 / 57

### 3. 그리드레이아웃

#### ■ 그리드레이아웃(GridLayout)의 속성

- <GridLayout> 자체에 자주 사용되는 속성
  - rowCount : 행 개수
  - columnCount : 열의 수
  - orientation : 그리드를 수평 방향을 우선할지, 수직 방향을 우선할지를 결정
- 그리드레이아웃 안에 포함될 위젯에서 자주 사용되는 속성
  - layout\_row : 자신이 위치할 행 번호(0번부터 시작)
  - layout\_column : 자신이 위치할 열 번호(0번부터 시작)
  - layout\_rowSpan : 행을 지정된 수만큼 확장
  - layout\_columnSpan : 열을 지정된 수만큼 확장
  - layout\_gravity : 주로 fill, fill\_vertical, fill\_horizontal 등으로 지정 행 또는 열 확장 시, 위젯을 확장된 셀에 꽉 채우는 효과를 냄

52 / 57

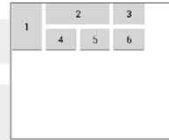
### 3. 그리드레이아웃

예제 5-18 그리드레이아웃의 XML 코드

```

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

```



53 / 57

### 3. 그리드레이아웃

#### ▶ 작업 풀어보기 5-5

[실습 5-2]를 그리드레이아웃으로 변경하여 실행하라.

**힌트1** 그리드레이아웃은 위젯의 높이 조절이 잘 되지 않아 전체 화면이 채워지지 않을 수 있다.

**힌트2** Java 코드는 고칠 필요가 없고 XML만 변경하면 된다. XML 위젯의 아이디도 동일하게 사용한다.

그림 5-16 그리드레이아웃 계산기 앱



54 / 57

## 4. 프레임레이아웃

### ■ 프레임레이아웃(FrameLayout)

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



그림 5-17 프레임레이아웃의 개념

55 / 57

## 4. 프레임레이아웃

### ■ 프레임레이아웃(FrameLayout)의 속성

- <FrameLayout>에서 가끔 사용하는 속성
  - **foreground** : 프레임레이아웃의 전경 이미지를 지정
  - **foregroundGravity** : 전경 이미지의 위치를 지정

예제 5-19 프레임레이아웃의 XML 코드

```
1 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   android:layout_width="match_parent"
3   android:layout_height="match_parent"
4   android:foreground="@drawable/dog"
5   android:foregroundGravity="center|fill_horizontal" >
6   <RatingBar
7     android:id="@+id/ratingBar1" >
8   <ImageView
9     android:src="@drawable/ic_launcher" />
10  <CheckBox
11    android:text="CheckBox" />
12 </FrameLayout>
```



56 / 57



**감사합니다.**