

05. 레이아웃 익히기

IT CookBook, Android Studio를 활용한 안드로이드 프로그래밍

© 이 강의노트는 한빛아카데미 출판사에서 제공한 강의노트를 기초로 만들어졌습니다.

학습목표

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

-
- 01 레이아웃의 개요
 - 02 리니어레이아웃
 - 03 기타 레이아웃

Page • 3

1. 레이아웃 개요 ▶ 레이아웃 기본 개념[1/2]

❖ 레이아웃

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

```
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

Page • 4

1. 레이아웃 개요 ▶ 레이아웃 기본 개념[2/2]

❖ 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

Page • 5

1. 레이아웃 개요 ▶ 레이아웃의 종류

❖ 레이아웃의 종류

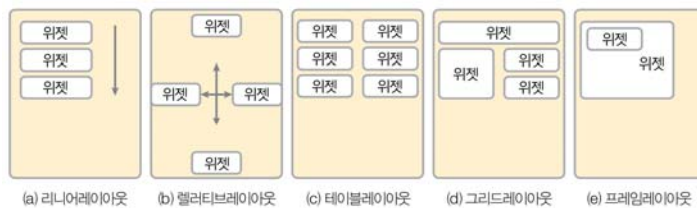


그림 5-1 레이아웃의 종류

- ✓ **리니어레이아웃** : 왼쪽 위부터 아래쪽 또는 오른쪽으로 차례로 배치
- ✓ **렐라티브레이아웃** : 위젯 자신이 속한 레이아웃의 상하좌우의 위치를 지정하여 배치
- ✓ **테이블레이아웃** : 위젯을 행과 열의 개수를 지정한 테이블 형태로 배열
- ✓ **그리드레이아웃** : 테이블레이아웃과 비슷하지만, 행 또는 열을 확장하여 다양하게 배치할 때 더 편리
- ✓ **프레임레이아웃** : 위젯들을 왼쪽 위에 일률적으로 겹쳐서 배치하여 중복해서 보이는 효과를 냄

Page • 6

2. 리니어레이아웃 ▶ 기본 리니어레이아웃 형태[1/7]

❖ 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" >

    ~~~ 여기에 위젯 배치 ~~~

</LinearLayout>
```

Page • 7

2. 리니어레이아웃 ▶ 기본 리니어레이아웃 형태[2/7]

❖ orientation 속성이 vertical 값인 XML 코드

예제 5-1 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>
```



Page • 8

2. 리니어레이아웃 ▶ 기본 리니어레이아웃 형태[3/7]

❖ orientation 속성이 horizontal 값인 XML 코드

예제 5-2 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>
```



Page • 9

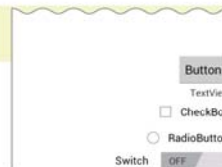
2. 리니어레이아웃 ▶ 기본 리니어레이아웃 형태[4/7]

❖ gravity 속성

- ✓ gravity 속성은 레이아웃 안의 위젯을 어디에 배치할 것인지를 결정

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   ~~~ 중간 생략 ~~~
11 </LinearLayout>
```



Page • 10

2. 리니어레이아웃 ▶ 기본 리니어레이아웃 형태[5/7]

❖ 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>
```



Page • 11

2. 리니어레이아웃 ▶ 기본 리니어레이아웃 형태[7/7]

❖ baselineAligned 속성

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



그림 5-3 baselineAligned 속성

Page • 12

2. 리니어레이아웃 ▶ 중복 리니어레이아웃 형태[1/5]

❖ 중복 리니어레이아웃 형태



그림 5-4 다양한 배치의 레이아웃

Page • 13

2. 리니어레이아웃 ▶ 중복 리니어레이아웃 형태[2/5]

❖ layout_weight 속성

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

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="#00FFFF"
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>
    
```

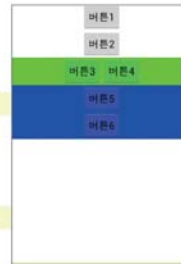
Page • 14

2. 리니어레이아웃 ► 중복 리니어레이아웃 형태[3/5]

❖ layout_height를 wrap_content로 변경

예제 5-6 layout_height를 wrap_content로 변경

```
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>
```



Page • 15

2. 리니어레이아웃 ► 중복 리니어레이아웃 형태[4/5]

❖ layout_weight를 1로 지정

예제 5-7 layout_weight를 1로 지정

```
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>
```



Page • 16

2. 리니어레이아웃 ▶ 중복 리니어레이아웃 형태[5/5]

▶ 작업 물어보기 5-2

리니어레이아웃으로 다음 화면을 구성하는 XML을 작성하라. 단, 레이아웃이 구분되어 보이도록 서로 다른 색으로 지정한다.



그림 5-5 중복 리니어레이아웃

HINT 레이아웃 안에 레이아웃을 여러 번 중첩해도 된다.

2. 리니어레이아웃 ▶ Java 코드로 화면 만들기[1/5]

❖ XML과 Java 코드 동작

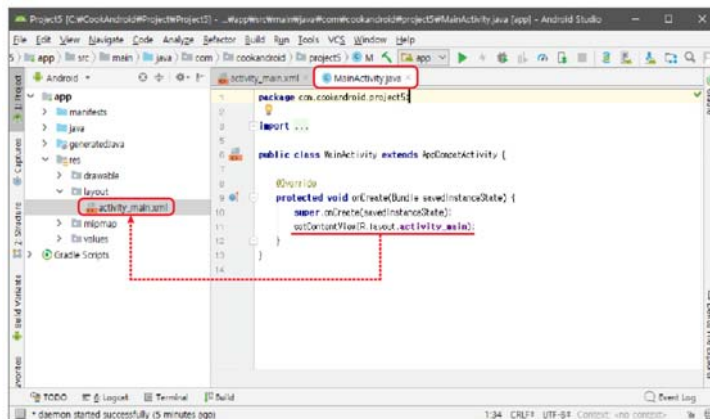


그림 5-6 XML과 Java 코드 동작

Coding Exercise XML Code 1

```
<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" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="horizontal" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#ff0000"
            android:orientation="vertical" >
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

Page • 19

Coding Exercise XML Code 2

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#00ff00"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#ffff00"
        android:orientation="vertical" >
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#000000"
        android:orientation="vertical" >
    </LinearLayout>
</LinearLayout>
```

Page • 20

Coding Exercise XML Code 3

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#0000ff"
    android:orientation="vertical" >

</LinearLayout>

</LinearLayout>
```

Page • 21

Coding Exercise Java Code

```
package com.cookandroid.self5_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Page • 22

2. 리니어레이아웃 ▶ Java 코드로 화면 만들기[2/5]

실습 5-1 XML 없이 화면 코딩하기

- ❖ 버튼을 클릭하면 토스트 메시지가 출력되는 화면을 Java로 코딩해보자.
- ❖ 안드로이드 프로젝트 생성
 - ✓ 프로젝트 이름 : Project5_1
 - ✓ 패키지 이름 : com.cookandroid.project5_1
- ❖ 화면 디자인 및 편집
 - ✓ activity_main.xml 삭제
- ❖ Java 코드 작성 및 수정
 - ✓ activity_main.xml 삭제했기 때문에 오류가 발생함
 - ✓ 오류 행 앞에 //를 붙여 주석으로 처리한 후 진행



그림 5-7 Java 코드만 사용한 프로젝트

Page • 23

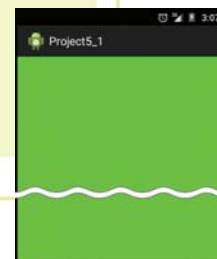
2. 리니어레이아웃 ▶ Java 코드로 화면 만들기[3/5]

실습 5-1 XML 없이 화면 코딩하기

- ❖ 리니어레이아웃을 생성하는 코드를 작성하고 실행

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 }
```



Page • 24

2. 리니어레이아웃 ▶ Java 코드로 화면 만들기[4/5]

실습 5-1 XML 없이 화면 코딩하기

❖ 버튼을 만들고, 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 });
```

Page • 25

Coding Exercise Java Code 1

```
package com.cookandroid.project5_1;

import android.support.v7.app.AppCompatActivity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    public 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);
```

Page • 26

Coding Exercise XML Code 2

```
LinearLayout baseLayout = new LinearLayout(this);
baseLayout.setOrientation(LinearLayout.VERTICAL);
baseLayout.setBackgroundColor(Color.rgb(0, 255, 0));
setContentView(baseLayout, params);

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

btn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        Toast.makeText(getApplicationContext(), "코드로 생성한 버튼입니다",
            Toast.LENGTH_SHORT).show();
    }
});
}
```

Page • 27

3. 기타 레이아웃 ▶ 렐러티브레이아웃[1/6]

❖ 렐러티브레이아웃(상대 레이아웃)

- ✓ 렐러티브레이아웃은 레이아웃 내부에 포함된 위젯을 상대적인 위치로 배치

❖ 렐러티브레이아웃의 상하좌우에 배치

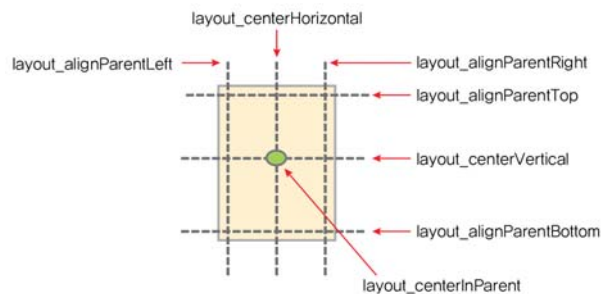


그림 5-9 부모(레이아웃)의 위치를 적용할 때 속성

Page • 28

3. 기타 레이아웃 ▶ 렐라티브레이아웃[2/6]

❖ 렐라티브레이아웃 예제

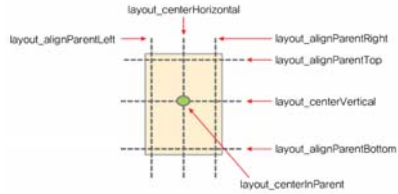


그림 5-9 부모레이아웃의 위치를 적용할 때 속성

예제 5-10 렐라티브레이아웃의 XML 코드 1

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
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>

```



3. 기타 레이아웃 ▶ 렐라티브레이아웃[3/6]

❖ 다른 위젯의 상대 위치에 배치

- ✓ 각 속성의 값은 다른 위젯의 id를 지정하면 됨
- ✓ "@+id/기준 위젯의 아이디"와 같은 형식으로 사용

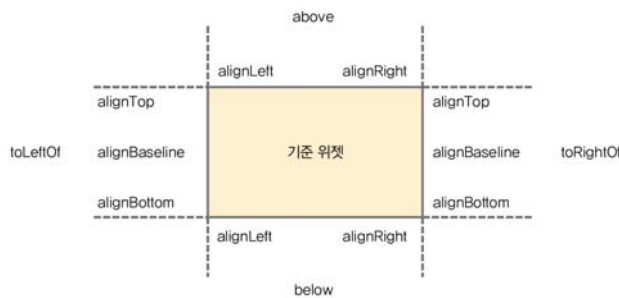


그림 5-10 다른 위젯의 상대적인 위치를 적용할 때의 속성

간략하게 표현하기 위해
속성 이름의 layout_은 생략

3. 기타 레이아웃 ▶ 렐러티브레이아웃[4/6]

❖ 렐러티브레이아웃 안에서 다른 위젯의 특정한 곳 배치 예제

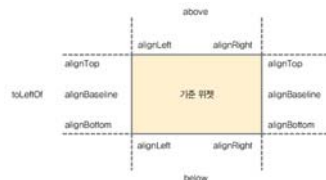


그림 5-10 다른 위젯의 상대적인 위치를 적용할 때의 속성

예제 5-11 렐러티브레이아웃의 XML 코드 2

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent" >
4     <Button
5         android:id="@+id/baseBtn"
6         android:layout_width="158dp"
7         android:layout_height="158dp"
8         android:layout_centerHorizontal="true"
9         android:layout_centerVertical="true"
10        android:text="기준 위젯" />
11
12     <Button
13         android:layout_alignTop="@id/baseBtn"
14         android:layout_toLeftOf="@id/baseBtn"
15         android:text="1번" />
16
17     <Button
18         android:layout_alignTop="@id/baseBtn"
19         android:layout_alignLeft="@id/baseBtn"
20         android:text="4번" />
21
22     <Button
23         android:layout_alignRight="@id/baseBtn"
24         android:layout_below="@id/baseBtn"
25         android:text="5번" />
26
27     <Button
28         android:layout_alignRight="@id/baseBtn"
29         android:layout_alignBottom="@id/baseBtn"
30         android:text="6번" />
31 </RelativeLayout>

```



3. 기타 레이아웃 ▶ 렐러티브레이아웃[5/6]

❖ 여러 위젯에서 상대 위치를 지정한 예제



그림 5-10 다른 위젯의 상대적인 위치를 적용할 때의 속성

예제 5-12 렐러티브레이아웃 속성을 조합한 XML 코드

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
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
10    <Button
11        android:id="@+id/baseBtn2"
12        android:layout_alignParentRight="true"
13        android:layout_centerVertical="true"
14        android:text="기준2" />
15
16    <Button
17        android:layout_alignTop="@id/baseBtn2"
18        android:layout_toRightOf="@id/baseBtn1"
19        android:text="1번" />
20
21    <Button
22        android:layout_alignParentRight="true"
23        android:layout_below="@id/baseBtn1"
24        android:text="2번" />
25 </RelativeLayout>

```



3. 기타 레이아웃 ▶ 테이블레이아웃[1/8]

❖ 테이블레이아웃(TableLayout)

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

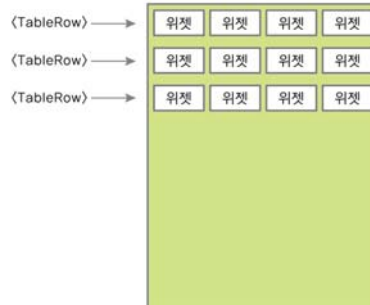


그림 5-12 테이블레이아웃의 개요

Page • 33

3. 기타 레이아웃 ▶ 테이블레이아웃[2/8]

❖ 테이블레이아웃의 속성

- ✓ layout_column : 지정된 열에 현재 위젯을 표시하라는 의미
- ✓ stretchColumns : 지정된 열의 폭을 늘리라는 의미
- ✓ stretchColumns = "*" : 각 셀을 같은 크기로 확장, 전체 화면이 꽉 차는 효과

Page • 34

3. 기타 레이아웃 ▶ 테이블레이아웃[3/8]

❖ 테이블레이아웃 예제

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>

```



Page • 35

3. 기타 레이아웃 ▶ 테이블레이아웃[4/8]

실습 5-2 테이블레이아웃을 활용한 계산기 앱 만들기

❖ 테이블레이아웃을 활용하여 숫자 버튼까지 있는 계산기를 만들어보자.

❖ 안드로이드 프로젝트 생성

- ✓ 프로젝트 이름 : Project5_2
- ✓ 패키지 이름 : com.cookandroid.project5_2

❖ 화면 디자인 및 편집

- ✓ 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 20dp
- ✓ 각 위젯의 id는 위에서부터 Edit1, Edit2, BtnNum0~9, BtnAdd, BtnSub, BtnMul, BtnDiv, TextResult



그림 5-13 테이블레이아웃을 활용한 계산기 결과 화면

Page • 36

3. 기타 레이아웃 ▶ 테이블레이아웃[5/8]

실습 5-2 테이블레이아웃을 활용한 계산기 앱 만들기

❖ 화면 디자인 및 편집

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="숫자 입력" />
10   </TableRow>
11   ----- 중간 행 (TableRow 1개, EditText 1개) -----
12
13   <TableRow>
14       <Button
15           android:id="@+id/btnNum0"
16           android:text="0" />
17       ----- 중간 행 (숫자 Button 4개) -----
18   </TableRow>
19   <TableRow>
20       ----- 중간 행 (숫자 Button 5개) -----
21   </TableRow>
22
23 </TableLayout>
24
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 ----- 중간 행 (TableRow 3개, 연산 Button 3개) -----
33
34 <TableRow>
35     <TextView
36         android:id="@+id/textResult"
37         android:layout_margin="5dp"
38         android:layout_span="5"
39         android:text="계산 결과 : "
40         android:textColor="#FF0000"
41         android:textSize="20dp" />
42 </TableRow>
43
44 </TableLayout>

```

Page • 37

3. 기타 레이아웃 ▶ 테이블레이아웃[6/8]

실습 5-2 테이블레이아웃을 활용한 계산기 앱 만들기

❖ Java 코드 작성 및 수정

✓ 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

예제 5-15 Java 코드 1

```

1 ----- 중간 행 -----
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         ----- 중간 행 -----

```

Page • 38

3. 기타 레이아웃 ▶ 테이블레이아웃[7/8]

실습 5-2 테이블레이아웃을 활용한 계산기 앱 만들기

❖ Java 코드 작성 및 수정

- ✓ 숫자 버튼이 없다고 가정하고 연산 버튼을 터치했을 때의 내용을 코딩

예제 5-16 Java 코드 2

```
1  ----- 중간 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     ----- 중간 생략(연산 버튼 3개 대입) -----
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     ----- 중간 생략(연산 버튼 3개 터치 이벤트 리스너) -----
24 }
```

Page • 39

3. 기타 레이아웃 ▶ 테이블레이아웃[8/8]

실습 5-2 테이블레이아웃을 활용한 계산기 앱 만들기

❖ Java 코드 작성 및 수정

- ✓ 숫자 버튼 10개를 배열 변수에 대입한 후에 각 버튼의 클릭 이벤트 리스너를 만들

예제 5-17 Java 코드 3

```
1  for (i = 0; i < numBtnDs.length; i++) {
2      numButtons[i] = (Button) findViewById(R.id.numBtnDs[i]);
3  }
4
5  for (i = 0; i < numBtnDs.length; i++) {
6      final int index; // 주석: 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].setText(num1);
15             } else if (edit2.isFocused() == true) {
16                 num2 = edit2.getText().toString();
17                 numButtons[index].setText(num2);
18             } else {
19                 Toast.makeText(getApplicationContext(),
20                     "먼저 에디트텍스트를 선택하세요", Toast.LENGTH_SHORT).show();
21             }
22         }
23     });
24 }
25 }
26
27 }
```

Page • 40

Coding Exercise XML Code 1

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/tableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <TableRow>
        <EditText
            android:layout_weight = "1"
            android:id="@+id/Edit1"
            android:layout_span="5"
            android:hint="숫자1 입력" />
    </TableRow>
    <TableRow>
        <EditText
            android:layout_weight = "1"
            android:id="@+id/Edit2"
            android:layout_span="5"
            android:hint="숫자2 입력" />
    </TableRow>
```

Page * 41

Coding Exercise XML Code 2

```
<TableRow>
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum0"
        android:text="0" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum1"
        android:text="1" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum2"
        android:text="2" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum3"
        android:text="3" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum4"
        android:text="4" />
</TableRow>
```

Page * 42

Coding Exercise XML Code 3

```
<TableRow>
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum5"
        android:text="5" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum6"
        android:text="6" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum7"
        android:text="7" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum8"
        android:text="8" />
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnNum9"
        android:text="9" />
</TableRow>
```

Page * 43

Coding Exercise XML Code 4

```
<TableRow>
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnAdd"
        android:layout_margin="5dp"
        android:layout_span="5"
        android:text="더하기" />
</TableRow>
<TableRow>
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnSub"
        android:layout_margin="5dp"
        android:layout_span="5"
        android:text="빼기" />
</TableRow>
<TableRow>
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnMul"
        android:layout_margin="5dp"
        android:layout_span="5"
        android:text="곱하기" />
</TableRow>
```

Page * 44

Coding Exercise XML Code 5

```
<TableRow>
    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnMul"
        android:layout_margin="5dp"
        android:layout_span="5"
        android:text="곱하기" />
</TableRow>

<TableRow>

    <Button
        android:layout_weight = "1"
        android:id="@+id/BtnDiv"
        android:layout_margin="5dp"
        android:layout_span="5"
        android:text="나누기" />
</TableRow>
```

Page • 45

Coding Exercise XML Code 6

```
<TableRow>

    <TextView
        android:id="@+id/TextResult"
        android:layout_margin="5dp"
        android:layout_span="5"
        android:text="계산 결과 : "
        android:textColor="#FF0000"
        android:textSize="20dp" />
</TableRow>

</TableLayout>
```

Page • 46

Coding Exercise Java Code 1

```
package com.cookandroid.project5_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText edit1, edit2;
    Button btnAdd, btnSub, btnMul, btnDiv;
    TextView textResult;
    String num1, num2;
    Integer result;
```

Page * 47

Coding Exercise Java Code 2

```
// 10개 숫자 버튼 배열
Button[] numButtons = new Button[10];
// 10개 숫자 버튼의 id 값 배열
Integer[] numBtnIDs = { R.id.BtnNum0, R.id.BtnNum1, R.id.BtnNum2,
    R.id.BtnNum3, R.id.BtnNum4, R.id.BtnNum5, R.id.BtnNum6,
    R.id.BtnNum7, R.id.BtnNum8, R.id.BtnNum9 };
int i; // 증가값 용도

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    setTitle("데이블레이아웃 계산기");
    edit1 = (EditText) findViewById(R.id.Edit1);
    edit2 = (EditText) findViewById(R.id.Edit2);

    btnAdd = (Button) findViewById(R.id.BtnAdd);
    btnSub = (Button) findViewById(R.id.BtnSub);
    btnMul = (Button) findViewById(R.id.BtnMul);
    btnDiv = (Button) findViewById(R.id.BtnDiv);
```

Page * 48

Coding Exercise Java Code 3

```
textResult = (TextView) findViewById(R.id.TextResult);

btnAdd.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) + Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});

btnSub.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) - Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});
```

Page • 49

Coding Exercise Java Code 4

```
btnMul.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) * Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});

btnDiv.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) / Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});
```

Page • 50

Coding Exercise Java Code 5

```
// 숫자 버튼 10개를 대입
for (i = 0; i < numBtnIDs.length; i++) {
    numButtons[i] = (Button) findViewById(numBtnIDs[i]);
}
// 숫자 버튼 10개에 대해서 클릭이벤트 처리
for (i = 0; i < numBtnIDs.length; i++) {

    final int index; // 주의! 꼭 필요함..
    index = i;
```

Page • 51

Coding Exercise Java Code 6

```
numButtons[index].setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // 포커스가 되어 있는 에디트텍스트에 숫자 추가
        if (edit1.isFocused() == true) {
            num1 = edit1.getText().toString()
                + numButtons[index].getText().toString();
            edit1.setText(num1);
        } else if (edit2.isFocused() == true) {
            num2 = edit2.getText().toString()
                + numButtons[index].getText().toString();
            edit2.setText(num2);
        } else {
            Toast.makeText(getApplicationContext(),
                "먼저 에디트텍스트를 선택하세요", Toast.LENGTH_SHORT).show();
        }
    }
});
}
```

Page • 52

3. 기타 레이아웃 ▶ 그리드레이아웃[1/4]

❖ 그리드레이아웃(GridLayout)

- ✓ 테이블레이아웃처럼 위젯을 표 형태로 배치할 때 사용하지만 좀 더 직관적임
- ✓ Android 4.0(아이스크림 샌드위치, API 14)부터 지원



그림 5-14 에디트텍스트에 포커스가 없는 상태에서 클릭

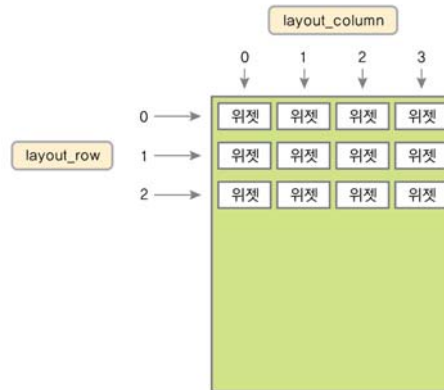


그림 5-15 그리드레이아웃의 개요

Page • 53

3. 기타 레이아웃 ▶ 그리드레이아웃[2/4]

❖ 그리드레이아웃 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 확장시, 위젯을 확장된 셀에 꽉 채우는 효과를 냄

Page • 54

3. 기타 레이아웃 ▶ 그리드레이아웃[3/4]


❖ 그리드레이아웃 예제

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    <!-- 중간 생략 -->
25 </GridLayout>

```



Page • 55

3. 기타 레이아웃 ▶ 그리드레이아웃[4/4]

▶ 작업 들어보기 5-5

[실습 5-2]를 그리드레이아웃으로 변경하여 실행하라.

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



그림 5-16 그리드레이아웃을 활용한 계산기

Page • 56

Coding Exercise XML Code 1

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/tableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="5"
    android:orientation="vertical"
    android:rowCount="9">
    <EditText
        android:id="@+id/Edit1"
        android:layout_columnSpan="5"
        android:layout_gravity="fill_horizontal"
        android:hint="숫자1 입력" />
    <EditText
        android:id="@+id/Edit2"
        android:layout_columnSpan="5"
        android:layout_gravity="fill_horizontal"
        android:hint="숫자2 입력" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

Page • 57

Coding Exercise XML Code 2

```
<Button
    android:id="@+id/BtnNum0"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="0" />
<Button
    android:id="@+id/BtnNum1"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="1" />
<Button
    android:id="@+id/BtnNum2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="2" />
```

Page • 58

Coding Exercise XML Code 3

```
<Button
    android:id="@+id/BtnNum3"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="3" />

<Button
    android:id="@+id/BtnNum4"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="4" />
</LinearLayout>
```

Page • 59

Coding Exercise XML Code 4

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/BtnNum5"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="5" />

    <Button
        android:id="@+id/BtnNum6"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="6" />
```

Page • 60

Coding Exercise XML Code 5

```
<Button
    android:id="@+id/BtnNum7"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="7" />

<Button
    android:id="@+id/BtnNum8"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="8" />

<Button
    android:id="@+id/BtnNum9"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="9" />
</LinearLayout>
```

Page • 61

Coding Exercise XML Code 6

```
<Button
    android:id="@+id/BtnAdd"
    android:layout_column="0"
    android:layout_columnSpan="5"
    android:layout_gravity="fill_horizontal"
    android:layout_margin="5dp"
    android:layout_row="4"
    android:text="더하기" />

<Button
    android:id="@+id/BtnSub"
    android:layout_column="0"
    android:layout_columnSpan="5"
    android:layout_gravity="fill_horizontal"
    android:layout_margin="5dp"
    android:layout_row="5"
    android:text="빼기" />
```

Page • 62

Coding Exercise XML Code 7

```
<Button
    android:id="@+id/BtnMul"
    android:layout_column="0"
    android:layout_columnSpan="5"
    android:layout_gravity="fill_horizontal"
    android:layout_margin="5dp"
    android:layout_row="6"
    android:text="곱하기" />

<Button
    android:id="@+id/BtnDiv"
    android:layout_column="0"
    android:layout_columnSpan="5"
    android:layout_gravity="fill_horizontal"
    android:layout_margin="5dp"
    android:layout_row="7"
    android:text="나누기" />
```

Page • 63

Coding Exercise XML Code 8

```
<TextView
    android:id="@+id/TextResult"
    android:layout_column="0"
    android:layout_columnSpan="5"
    android:layout_gravity="fill_horizontal"
    android:layout_margin="5dp"
    android:layout_row="8"
    android:text="계산 결과 : "
    android:textColor="#FF0000"
    android:textSize="20dp" />

</GridLayout>
```

Page • 64

Coding Exercise Java Code 1

```
package com.cookandroid.self5_5;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText edit1, edit2;
    Button btnAdd, btnSub, btnMul, btnDiv;
    TextView textResult;
    String num1, num2;
    Integer result;
```

Page • 65

Coding Exercise Java Code 2

```
// 10개 숫자 버튼 배열
Button[] numButtons = new Button[10];
// 10개 숫자 버튼의 id 값 배열
Integer[] numBtnIDs = { R.id.BtnNum0, R.id.BtnNum1, R.id.BtnNum2,
    R.id.BtnNum3, R.id.BtnNum4, R.id.BtnNum5, R.id.BtnNum6,
    R.id.BtnNum7, R.id.BtnNum8, R.id.BtnNum9 };
int i; // 증가값 용도

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    setTitle("그리드레이아웃 계산기");

    edit1 = (EditText) findViewById(R.id.Edit1);
    edit2 = (EditText) findViewById(R.id.Edit2);
```

Page • 66

Coding Exercise Java Code 3

```
btnAdd = (Button) findViewById(R.id.BtnAdd);
btnSub = (Button) findViewById(R.id.BtnSub);
btnMul = (Button) findViewById(R.id.BtnMul);
btnDiv = (Button) findViewById(R.id.BtnDiv);

textResult = (TextView) findViewById(R.id.TextResult);

btnAdd.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) + Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});
```

Page • 67

Coding Exercise Java Code 4

```
btnSub.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) - Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});

btnMul.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) * Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});
```

Page • 68

Coding Exercise Java Code 5

```
btnDiv.setOnTouchListener(new View.OnTouchListener() {
    public boolean onTouch(View arg0, MotionEvent arg1) {
        num1 = edit1.getText().toString();
        num2 = edit2.getText().toString();
        result = Integer.parseInt(num1) / Integer.parseInt(num2);
        textResult.setText("계산 결과 : " + result.toString());
        return false;
    }
});

// 숫자 버튼 10개를 대입
for (i = 0; i < numBtnIDs.length; i++) {
    numButtons[i] = (Button) findViewById(numBtnIDs[i]);
}
```

Page • 69

Coding Exercise Java Code 6

```
// 숫자 버튼 10개에 대해서 클릭이벤트 처리
for (i = 0; i < numBtnIDs.length; i++) {
    final int index; // 주의! 꼭 필요함..
    index = i;
    numButtons[index].setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            // 포커스가 되어 있는 에디트텍스트에 숫자 추가
            if (edit1.isFocused() == true) {
                num1 = edit1.getText().toString()
                    + numButtons[index].getText().toString();
                edit1.setText(num1);
            } else if (edit2.isFocused() == true) {
                num2 = edit2.getText().toString()
                    + numButtons[index].getText().toString();
                edit2.setText(num2);
            } else {
                Toast.makeText(getApplicationContext(),
                    "먼저 에디트텍스트를 선택하세요", Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

Page • 70

3. 기타 레이아웃 ▶ 프레임레이아웃[1/3]

❖ 프레임레이아웃(FrameLayout)

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



그림 5-17 프레임레이아웃의 개요

Page • 71

3. 기타 레이아웃 ▶ 프레임레이아웃[2/3]

❖ 프레임레이아웃의 속성

- ✓ foreground : 프레임레이아웃의 전경 이미지를 지정
- ✓ foregroundGravity : 전경 이미지의 위치를 지정

Page • 72

3. 기타 레이아웃 ▶ 프레임레이아웃[3/3]

❖ 프레임레이아웃의 예제

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>
```



Question ?