

Ch05. 레이아웃 익히기1

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



01



레이아웃 개요



1. 레이아웃 개요

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
```

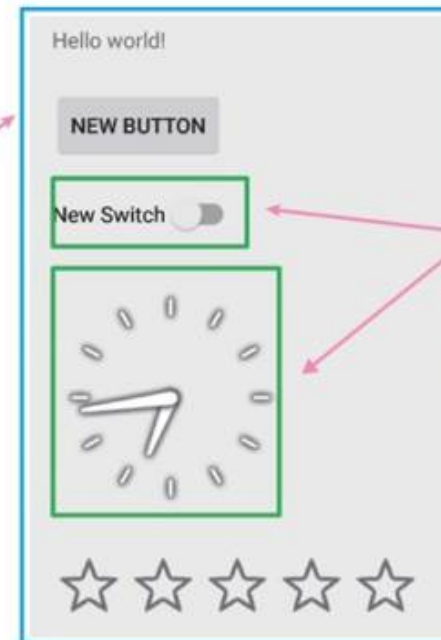
```
└ android.widget.RelativeLayout
```

```
└ android.widget.FrameLayout
```

```
└ android.widget.GridLayout
```

Layout 계층도

뷰그룹



뷰

1. 레이아웃 개요

레이아웃의 종류



LinearLayout : 왼쪽 위부터 아래쪽 또는 오른쪽으로 차례로 배치

RelativeLayout : 위젯 자신이 속한 레이아웃의 상하좌우의 위치를 지정하여 배치

TableLayout : 위젯을 행과 열의 개수를 지정한 테이블 형태로 배열

GridLayout : TableLayout과 비슷하지만, 행 또는 열을 확장하여 다양하게 배치할 때 더 편리

FrameLayout : 위젯들을 왼쪽 위에 일률적으로 겹쳐서 배치하여 중복해서 보이는 효과를 냄
상황에 따라 필요한 위젯을 보일 때

1. 레이아웃 개요

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 가중 값을 설정

여러 개의 레이아웃이 중복될 때 주로 사용

baselineAligned : 레이아웃 안에 배치할 위젯들을 보기 좋게 정렬

02



리니어레이아웃



1. 레이아웃 개요

orientation 속성

리니어레이아웃의 가장 기본적인 속성

vertical : 리니어레이아웃 안에 포함될 위젯의 배치를 수직방향으로 쌓음

horizontal : 수평방향으로 쌓음

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

~~~ 여기에 위젯들 배치 ~~~

```
</LinearLayout>
```

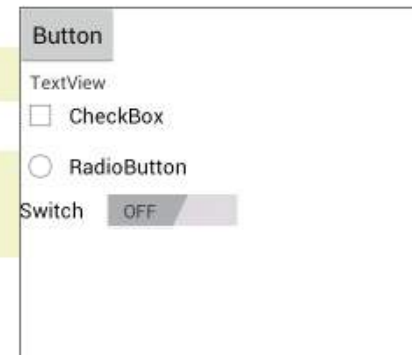
# 1. 레이아웃 개요

예제5-1: Ex05\_01

android:orientation = vertical

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>
```



## 2. 리니어레이아웃 | 일반

예제5-2: Ex05\_02

android:orientation = horizontal(생략시 기본으로 설정됨)

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>
```



## 2. 리니어레이아웃 | 일반

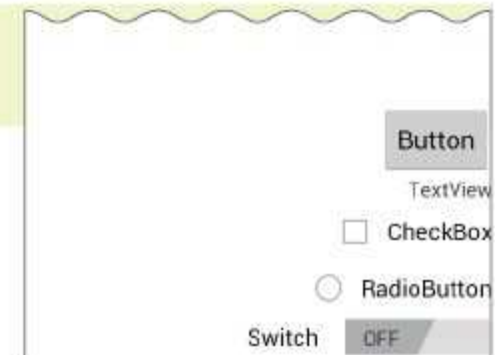
예제 5-3: Ex05\_03

gravity 속성

레이아웃 안의 위젯들을 어디에 배치할 것인지를 결정

예제 5-3 gravity 속성 XML 코드

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



## 2. 리니어레이아웃 | 일반

예제 5-4: Ex05\_04

layout\_gravity 속성

자신의 위치를 부모의 어디에 위치시킬지를 결정

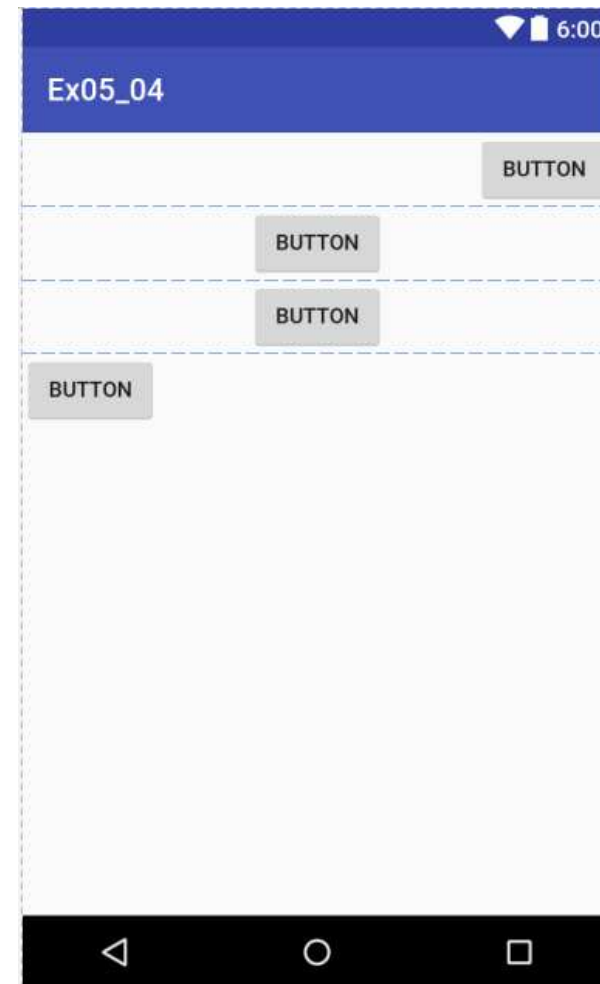
예제 5-4 layout\_gravity 속성 XML 코드

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

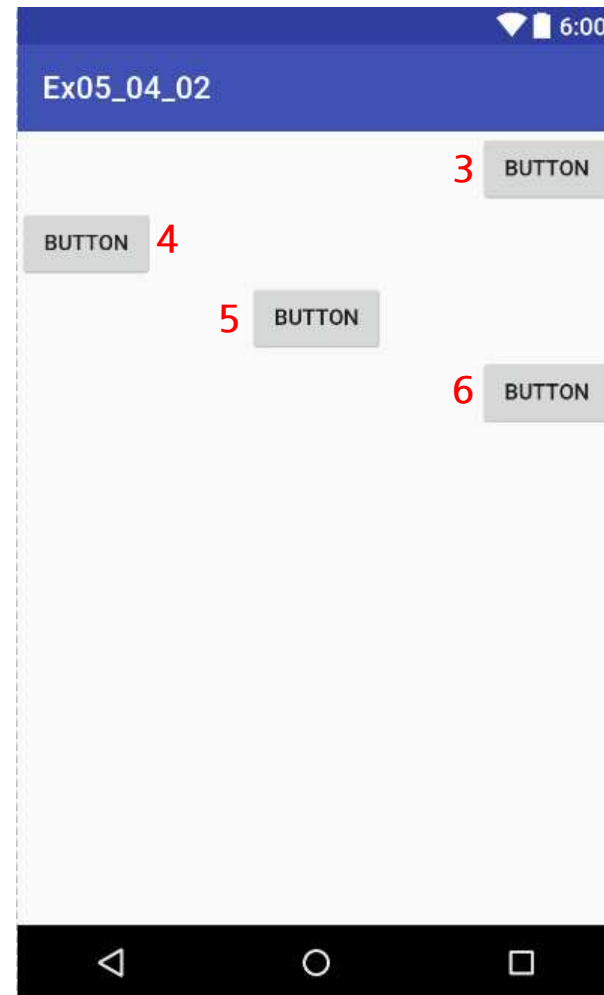




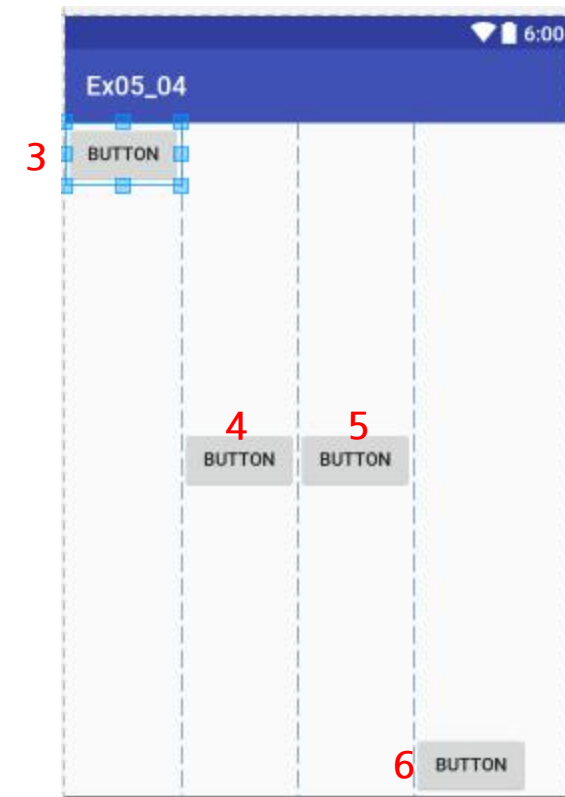
## 2. 리니어레이아웃 | 일반



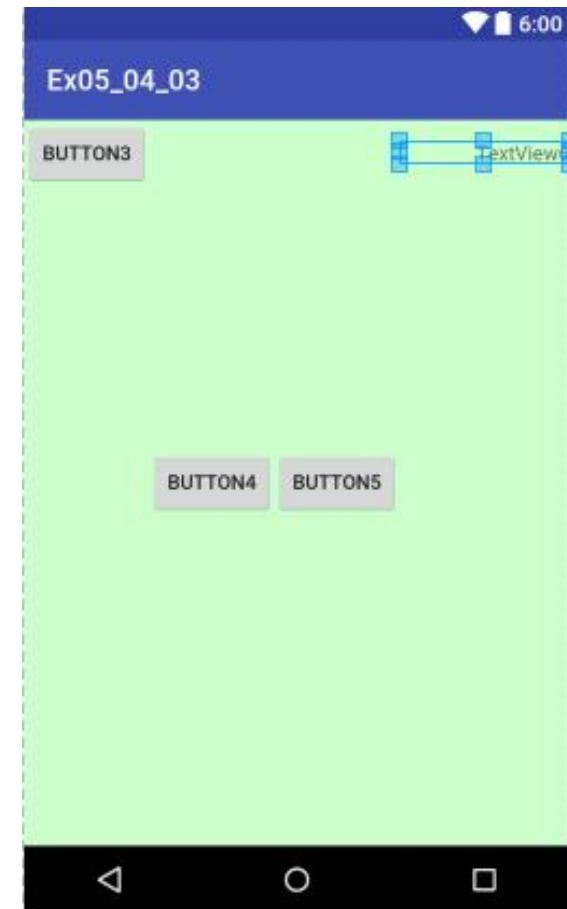
## 2. 리니어레이아웃 | 일반



## 2. 리니어레이아웃 | 일반



## 2. 리니어레이아웃 | 일반



## 2. 리니어레이아웃 | 일반

### 실습(직접 풀어보기): Test05\_01

리니어레이아웃으로 다음 화면을 구성하는 XML을 작성하라.



그림 5-2 리니어레이아웃 활용

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

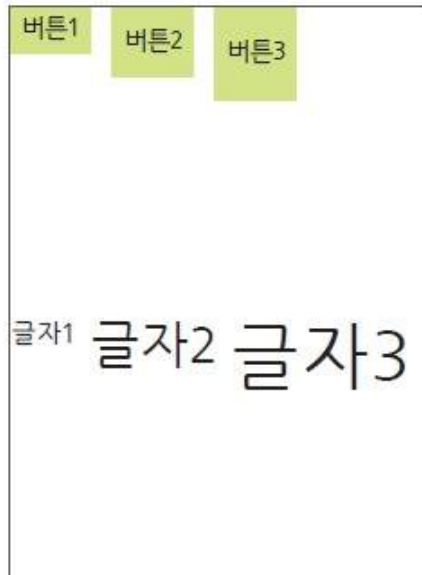
**NOTE** 버튼에 gravity와 layout\_gravity를 모두 설정해야 한다.



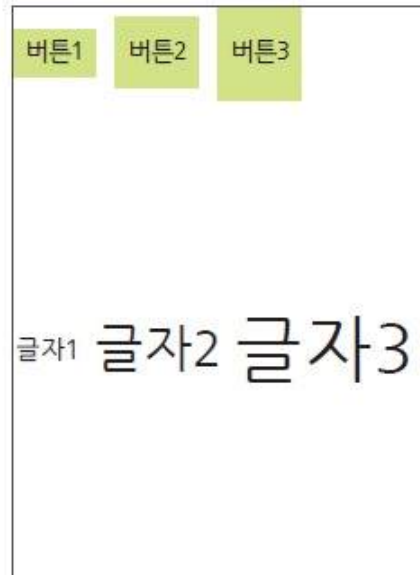
## 2. 리니어레이아웃 | 일반

### baselineAligned 속성

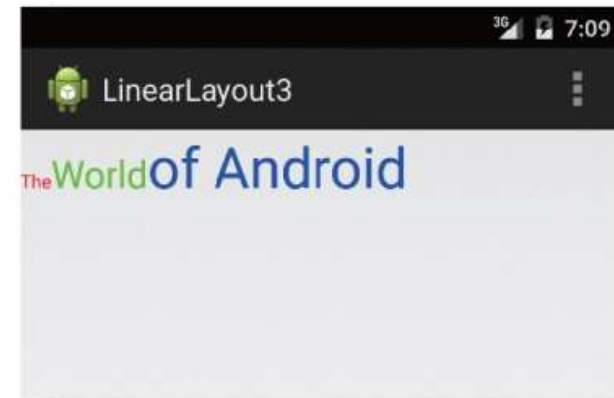
baselineAligned 속성은 크기가 다른 위젯들을 보기 좋게 정렬함  
true와 false 값을 가질 수 있음



(a) false로 지정



(b) 생략하거나 true로 지정



## 2. 리니어레이아웃 | 중복 리니어 레이아웃

### 중복 리니어레이아웃 형태

```
<LinearLayout>

    <LinearLayout>
        위젯들...
    </LinearLayout>

    <LinearLayout>
        위젯들...
    </LinearLayout>

    <LinearLayout>
        위젯들...
    </LinearLayout>

</LinearLayout>
```



### layout\_weight 속성

리니어레이아웃을 여러 개 사용할 경우 각 레이아웃이 차지하는 비율 지정

## 2. 리니어레이아웃 | 중복 리니어 레이아웃

예제5-5(~5-7): Ex05\_05



## 2. 리니어레이아웃 | 중복 리니어 레이아웃

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <LinearLayout
8     android:layout_width="match_parent"
9     android:layout_height="match_parent"
10    android:gravity="center"
11    android:orientation="vertical" >
12
13    <Button
14      android:layout_width="wrap_content"
15      android:layout_height="wrap_content"
16      android:text="버튼1" />
17
18    <Button
19      android:layout_width="wrap_content"
20      android:layout_height="wrap_content"
21      android:text="버튼2" />
22  </LinearLayout>
23
24  <LinearLayout
25    android:layout_width="match_parent"
26    android:layout_height="match_parent"
27    android:background="#00FF00"
28    android:gravity="center"
29    android:orientation="horizontal" >
30
31    <Button
32      android:layout_width="wrap_content"
33      android:layout_height="wrap_content"
34      android:text="버튼3" />
35
36    <Button
37      android:layout_width="wrap_content"
38      android:layout_height="wrap_content"
39      android:text="버튼4" />
40  </LinearLayout>
41
```

```
42 <LinearLayout
43   android:layout_width="match_parent"
44   android:layout_height="match_parent"
45   android:background="#0000FF"
46   android:gravity="center"
47   android:orientation="vertical" >
48
49   <Button
50     android:layout_width="wrap_content"
51     android:layout_height="wrap_content"
52     android:text="버튼5" />
53
54   <Button
55     android:layout_width="wrap_content"
56     android:layout_height="wrap_content"
57     android:text="버튼6" />
58 </LinearLayout>
59
60 </LinearLayout>
```

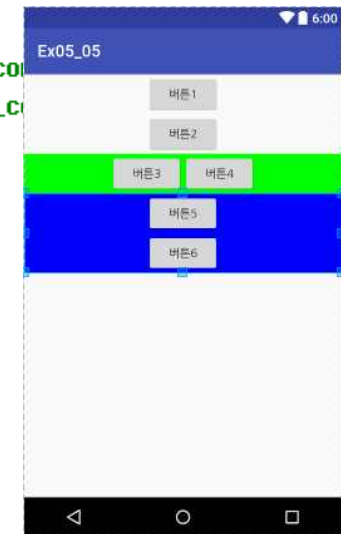


## 2. 리니어레이아웃 | 중복 리니어 레이아웃

layout\_height를 wrap\_content로 변경

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6
7     <LinearLayout
8         android:layout_width="match_parent"
9         android:layout_height="wrap_content"
10        android:gravity="center"
11        android:orientation="vertical" >
12
13        <Button
14            android:layout_width="wrap_content"
15            android:layout_height="wrap_content"
16            android:text="버튼1" />
17
18        <Button
19            android:layout_width="wrap_content"
20            android:layout_height="wrap_content"
21            android:text="버튼2" />
22    </LinearLayout>
23
24    <LinearLayout
25        android:layout_width="match_parent"
26        android:layout_height="wrap_content"
27        android:background="#00FF00"
28        android:gravity="center"
29        android:orientation="horizontal" >
30
31        <Button
32            android:layout_width="wrap_content"
33            android:layout_height="wrap_content"
34            android:text="버튼3" />
35
36        <Button
37            android:layout_width="wrap_content"
38            android:layout_height="wrap_content"
39            android:text="버튼4" />
40    </LinearLayout>
```

```
42 <LinearLayout
43     android:layout_width="match_parent"
44     android:layout_height="wrap_content"
45     android:background="#0000FF"
46     android:gravity="center"
47     android:orientation="vertical" >
48
49     <Button
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:text="버튼5" />
53
54     <Button
55         android:layout_width="wrap_content"
56         android:layout_height="wrap_content"
57         android:text="버튼6" />
58 </LinearLayout>
59
60 </LinearLayout>
```





## 2. 리니어레이아웃 | 중복 리니어 레이아웃

layout\_weight를 1로 지정

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <LinearLayout
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content"
10    android:gravity="center"
11    android:orientation="vertical"
12    android:layout_weight="1">
13
14    <Button
15      android:layout_width="wrap_content"
16      android:layout_height="wrap_content"
17      android:text="버튼1" />
18
19    <Button
20      android:layout_width="wrap_content"
21      android:layout_height="wrap_content"
22      android:text="버튼2" />
23  </LinearLayout>
24
25  <LinearLayout
26    android:layout_width="match_parent"
27    android:layout_height="wrap_content"
28    android:background="#00FF00"
29    android:gravity="center"
30    android:orientation="horizontal"
31    android:layout_weight="1">
32
33    <Button
34      android:layout_width="wrap_content"
35      android:layout_height="wrap_content"
36      android:text="버튼3" />
37
38    <Button
39      android:layout_width="wrap_content"
40      android:layout_height="wrap_content"
41      android:text="버튼4" />
```

```
44 <LinearLayout
45   android:layout_width="match_parent"
46   android:layout_height="wrap_content"
47   android:background="#0000FF"
48   android:gravity="center"
49   android:orientation="vertical"
50   android:layout_weight="1">
51
52   <Button
53     android:layout_width="wrap_content"
54     android:layout_height="wrap_content"
55     android:text="버튼5" />
56
57   <Button
58     android:layout_width="wrap_
59     android:layout_height="wrap
60     android:text="버튼6" />
61 </LinearLayout>
62
63 </LinearLayout>
```



## 2. 리니어레이아웃 | 중복 리니어 레이아웃

비율을 달리 지정

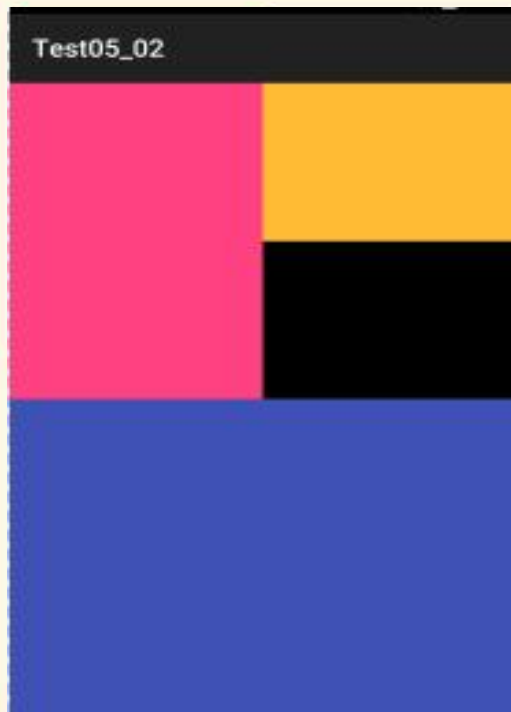
ex. 각 레이아웃의 `layout_height`을 0dp로,  
`layout_weight`는 1, 2,3으로 변경해보는 등 실습

## 2. 리니어레이아웃 | 중복 리니어 레이아웃

### 실습(직접 풀어보기): Test05\_02

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

**HINT** 레이아웃 안에 다시 레이아웃을 여러 번 중첩해도 된다.



## 2. 리니어레이아웃 | 동적생성

실습: Project05\_01

버튼을 클릭하면 토스트 메시지가 출력되는 화면을 Java만으로 코딩해보자.  
(without activity\_main.xml)

화면 디자인

activity\_main.xml 삭제

Java 코드 작성

activity\_main.xml 삭제하여 오류가 발생하면  
해당 행을 주석 처리한 후 진행



## 2. 리니어레이아웃 | 동적생성

```
11 public class MainActivity extends AppCompatActivity {
12
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         // setContentView(R.layout.activity_main);
18
19         LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT, LinearLayout.LayoutParams.MATCH_PARENT);
20
21         LinearLayout baseLayout = new LinearLayout(this);
22         baseLayout.setOrientation(LinearLayout.VERTICAL);
23         baseLayout.setBackgroundColor(Color.rgb(0,0,0));
24         setContentView(baseLayout, params);
25     }
26 }
```

Context c, AttributeSet attrs  
int width, int height  
int width, int height, float weight  
LayoutParams p  
MarginLayoutParams source  
LayoutParams source

파라미터 상에서 Ctrl + p 한 후,  
파라미터 확인

'setContentView(R.layout.activity\_main);' 대신  
생성한 화면 단위(뷰)에 리니어레이아웃을 생성하라



## 2. 리니어레이아웃 | 동적생성

버튼을 만들고, 버튼을 클릭했을 때 토스트 메시지를 작성  
(onCreate() 안에 이어서 코딩)

```
26 Button btn = new Button(this);
27 btn.setText("버튼입니다.");
28 btn.setBackgroundColor(Color.WHITE);
29 baseLayout.addView(btn);
30
31 btn.setOnClickListener(new View.OnClickListener() {
32     @Override
33     public void onClick(View v) {
34         Toast.makeText(getApplicationContext(), "코드로 생성한 버튼입니다.", Toast.LENGTH_LONG).show();
35     }
36 });
37
38 }
39
40 }
```

## 2. 리니어레이아웃 | 동적생성

### 실습(직접풀어보기) 5-3: Test05\_03

#### ▶ 직접 풀어보기 5-3

다음 화면을 XML 파일 없이 Java 코드만 이용하여 완성하라.

- 레이아웃에는 에디트텍스트 1개와 버튼 1개, 텍스트뷰 1개를 생성한다.
- 버튼을 클릭하면 에디트텍스트에 써진 문자열이 텍스트뷰에 나타나도록 한다.

