

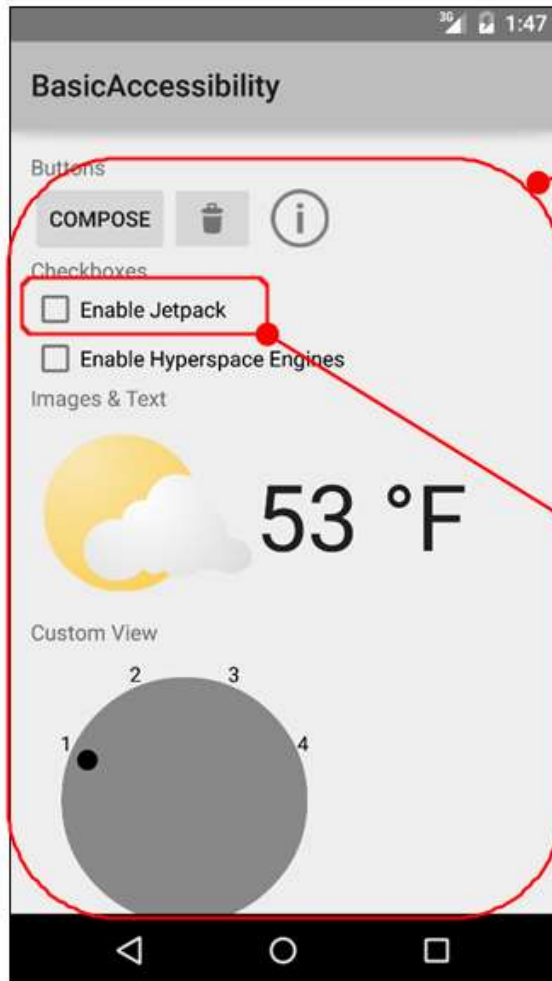
Layout

Mobile Software
2019 Fall

What to do next?

- **ViewGroup과 View**
- UI를 구현하는 3가지 coding style
- Layout
 - ConstraintLayout
 - LinearLayout
 - RelativeLayout
 - FrameLayout
 - TableLayout
 - GridLayout

ViewGroup과 View (1/3)



ViewGroup = Layout

- View 및 ViewGroup을 담을 수 있는 container 역할
- View 및 ViewGroup을 자신의 공간에 배치

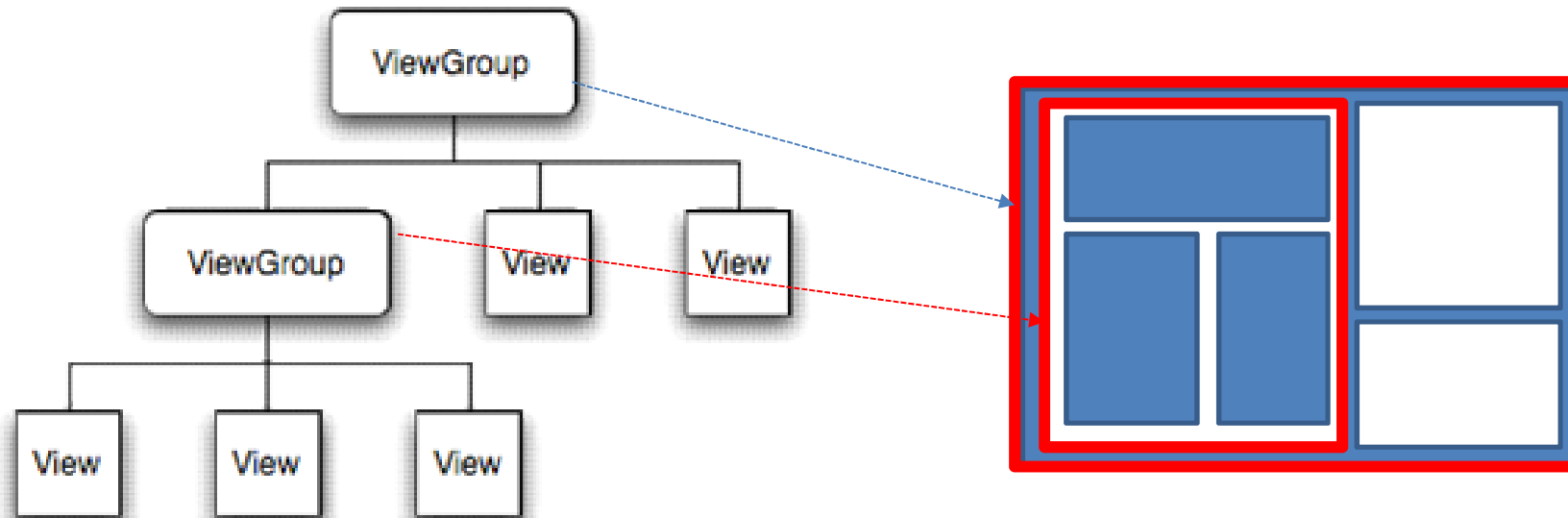
LinearLayout, RelativeLayout
ContainerLayout, FrameLayout
GridLayout, TableLayout
ConstraintLayout

View = Widget

- basic building block

Button, TextView, EditView,
RadioButton, CheckBox, ...

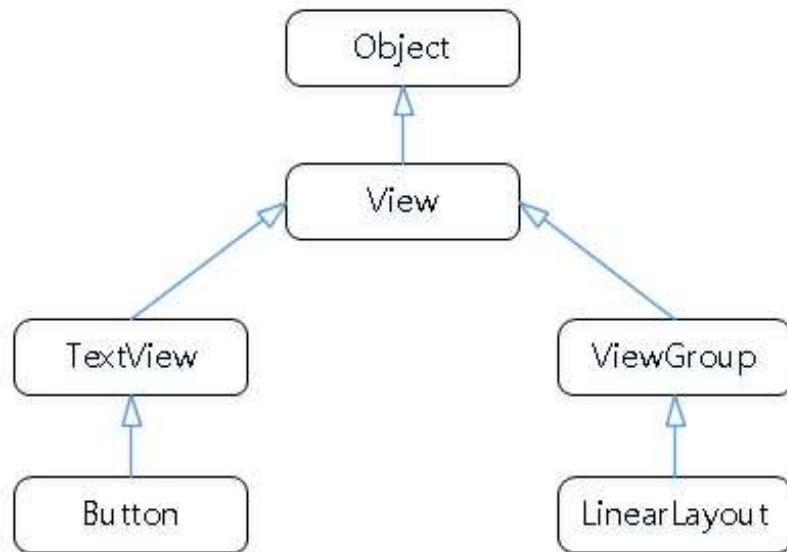
ViewGroup과 View (2/3)



Android 화면 = **UI(User interface)**
user로부터 입력을 받고 처리 결과를 보여준다

Android 화면 = **계층 구조 (hierarchical structure)**
view는 자신을 관리하는 ViewGroup에 포함
→ ViewGroup은 child node에 해당하는 view를 갖고 있다.

ViewGroup과 View (3/3)



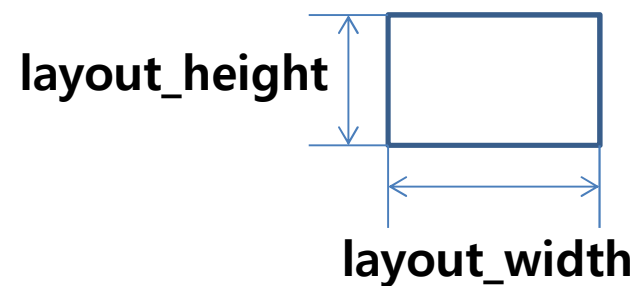
상속 (inheritance)
View ← ViewGroup
superclass *subclass*

ViewGroup은 **View**로부터 상속받았다.
→ View 는 ViewGroup을 포함한다.

Button은 **TextView**로부터 상속받았다.
→ TextView는 Button을 포함한다.



공통 속성을 갖는다.



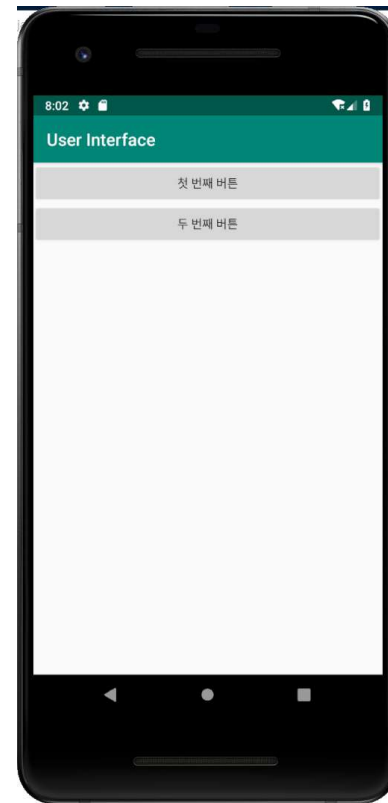
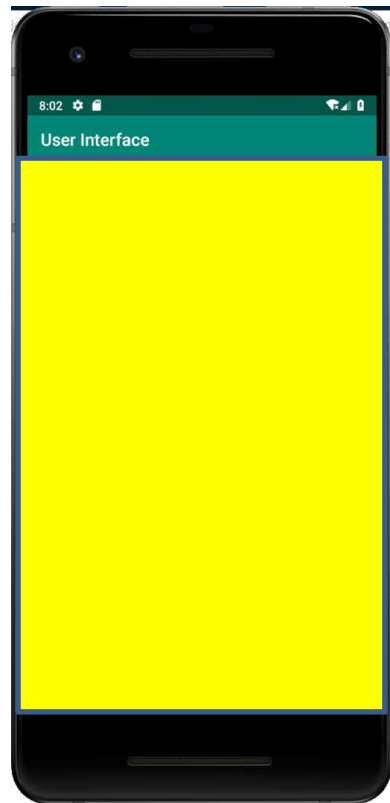
What to do next?

- ViewGroup과 View
- **UI를 구현하는 3가지 coding style**
- Layout
 - ConstraintLayout
 - LinearLayout
 - RelativeLayout
 - FrameLayout
 - TableLayout
 - GridLayout

UI 구현

1. ViewGroup 생성
2. View를 ViewGroup에 추가
3. Activity에 나타나도록 설정 : **setContentView**

root =
LinearLayout



UI를 구현하는 3가지 coding style

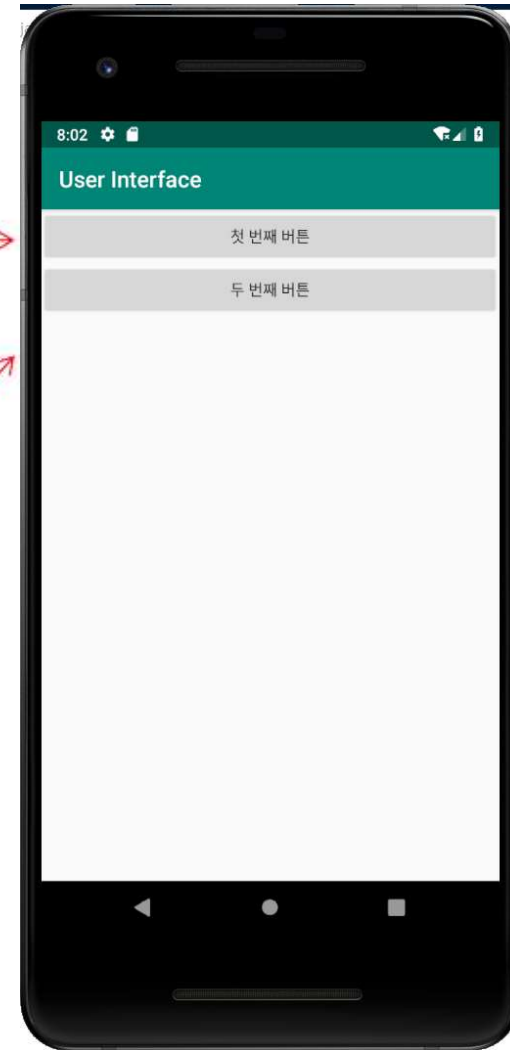
1번째 방법 : XML layout

```
...  
<Button  
  android:text= "첫 번째 버튼"  
  android:id="@+id/button1"  
  android:layout_width="match_parent"  
  android:layout_height="wrap_content">  
</Button>  
...
```

2번째 방법 : kotlin 코드

```
...  
val b1 = Button(this)  
b1.text = "첫 번째 버튼"  
linearLayout.addView(b1)  
...
```

3번째 방법 : XML layout + kotlin 코드



XML layout .vs. kotlin code

- **XML layout**

- Android studio layout editor 사용
 - XML 코드 자동 생성
- UI를 변경할 경우 XML 파일만 수정
 - Java 코드를 recompile할 필요가 없음
 - preview 기능 → UI를 수정할 때마다 즉시 확인 가능
- 정적 UI 구현에 유리

- **kotlin 코드**

- UI를 확인하려면 → `compile>run`
- 동적 UI 구현
 - 동적 UI란: Activity 실행 중에 UI를 바꿀 수 있음.

실습 준비

- 새 프로젝트 생성
 - Activity : **Empty Activity**
 - Application name : **User Interface**
 - Package name : **edu.ourincheon.userinterface**
 - Minimum API level : **API 24** (Nougat)
 - Activity name : **MainActivity.kt**
 - Layout name : **activity_main.xml**
- 자동 생성된 **레이아웃 파일**은 기본 layout으로 **ConstraintLayout** 이 지정되어 있음.
 - **ConstraintLayout** 대신 **LinearLayout** 으로 변경
 - **TextView** 도 삭제

Root Layout 변경

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</android.support.constraint.ConstraintLayout>
```



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

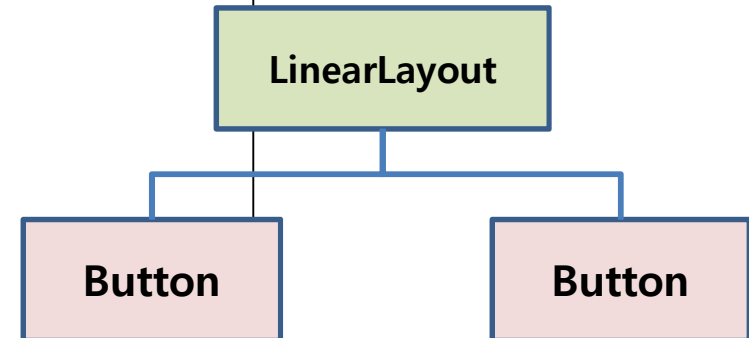
</LinearLayout>
```

실습 1: XML 파일로 UI 구현(1/2)

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/first_button"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/second_button"/>
</LinearLayout>
```



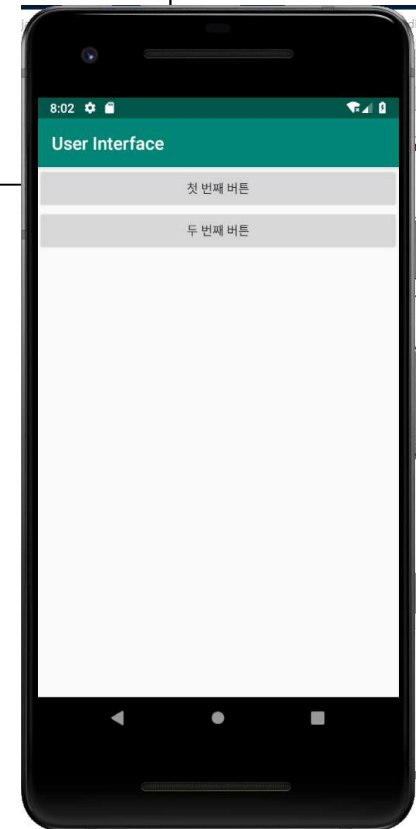
strings.xml

```
<resources>
    <string name="app_name">User Interface</string>
    <string name="first_button">첫 번째 버튼</string>
    <string name="second_button">두 번째 버튼</string>
</resources>
```

실습 1: XML 파일로 UI 구현(2/2)

MainActivity.kt

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```



실습 2: 코드로 UI 구현 (1/2)

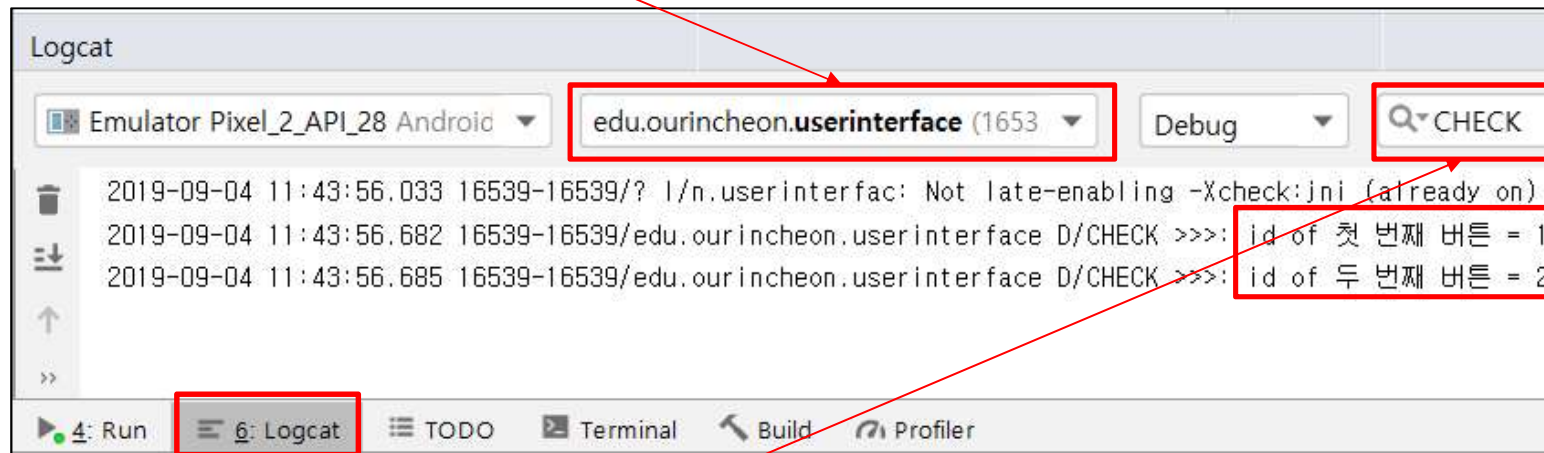
MainActivity.kt

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    // setContentView(R.layout.activity_main)  
  
    val linearLayout = LinearLayout(this)  
    linearLayout.orientation = LinearLayout.VERTICAL  
  
    val param = LinearLayout.LayoutParams(  
        LinearLayout.LayoutParams.MATCH_PARENT,  
        LinearLayout.LayoutParams.MATCH_PARENT  
    )  
    linearLayout.layoutParams = param  
  
    val b1 = Button(this)  
    b1.text = resources.getString(R.string.first_button)  
    b1.id = View.generateViewId()  
    Log.d("CHECK >>>", "id of ${b1.text} = ${b1.id}")  
  
    val b2 = Button(this)  
    b2.text = resources.getString(R.string.second_button)  
    b2.id = View.generateViewId()  
    Log.d("CHECK >>>", "id of ${b2.text} = ${b2.id}")  
  
    val buttonParam = LinearLayout.LayoutParams(  
        LinearLayout.LayoutParams.MATCH_PARENT,  
        LinearLayout.LayoutParams.WRAP_CONTENT  
    )  
    b1.layoutParams = buttonParam  
    b2.layoutParams = buttonParam  
    linearLayout.addView(b1)  
    linearLayout.addView(b2)  
  
    setContentView(linearLayout)  
}
```

```
linearLayout.addView(b1, buttonParam)  
linearLayout.addView(b2, buttonParam)
```


실습 2: 코드로 UI 구현 (2/2)

```
package edu.ourincheon.userinterface
```



```
Log.d("CHECK >>>", "id of ${b1.text} = ${b1.id}")
```

실습 3: XML layout + 코드 = UI

MainActivity.kt

XML Layout에서
정의한 view의 id를
참조하기 위해
import 해야 하는 클래스

button1, button2는
XML Layout 에서
정의한 해당 view의 id

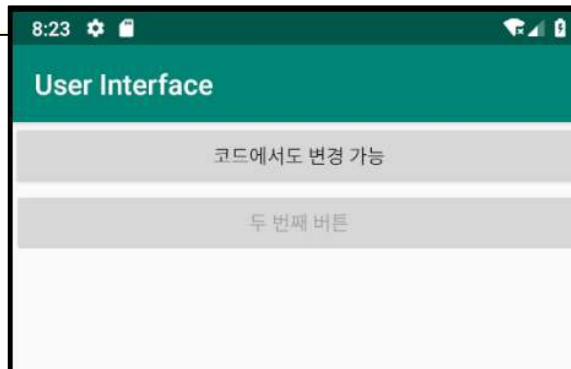
```
package edu.ourincheon.userinterface

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        button1.text = "코드에서도 변경 가능"
        button2.isEnabled = false
    }
}
```

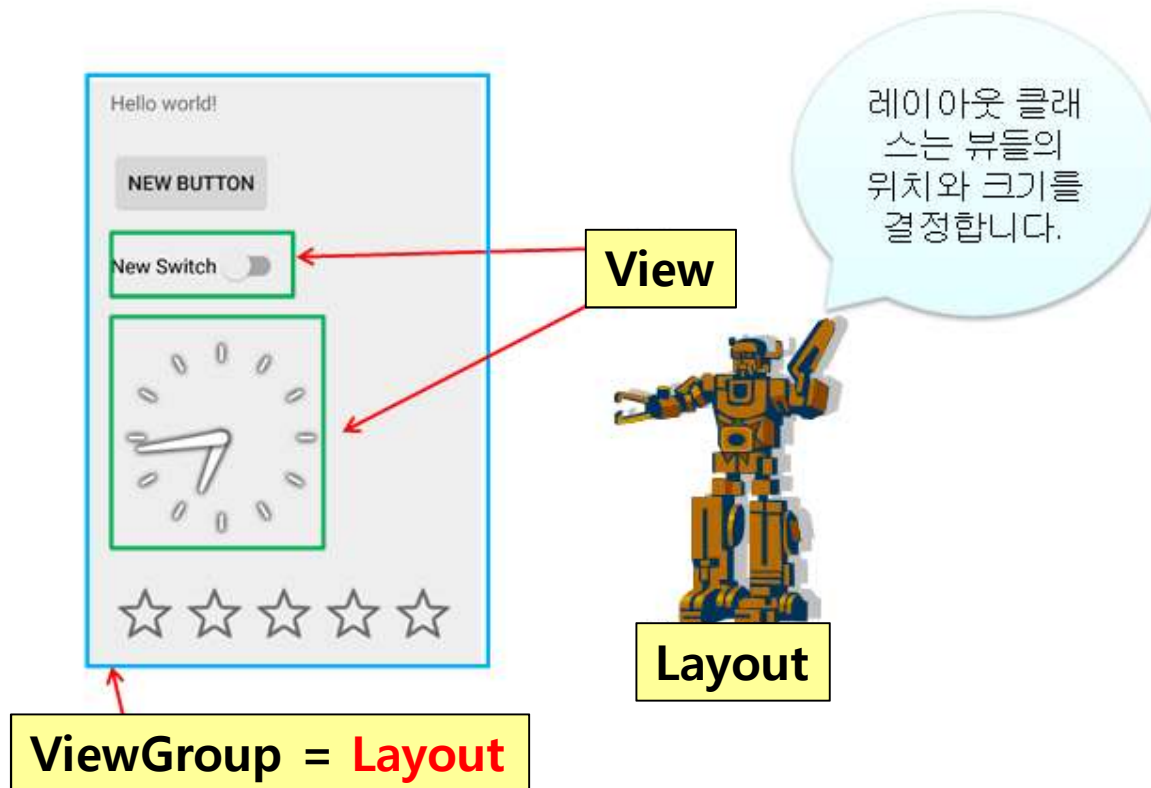


What to do next?

- ViewGroup과 View
- UI를 구현하는 3가지 coding style
- **Layout**
 - ConstraintLayout
 - **LinearLayout**
 - **RelativeLayout**
 - **FrameLayout**
 - **TableLayout**
 - **GridLayout**

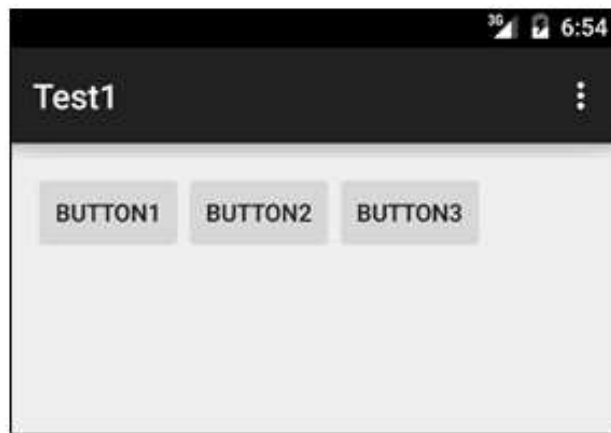
Layout

- Layout이란?
 - 화면에 View 를 배치 → View의 **위치**와 **크기**를 지정

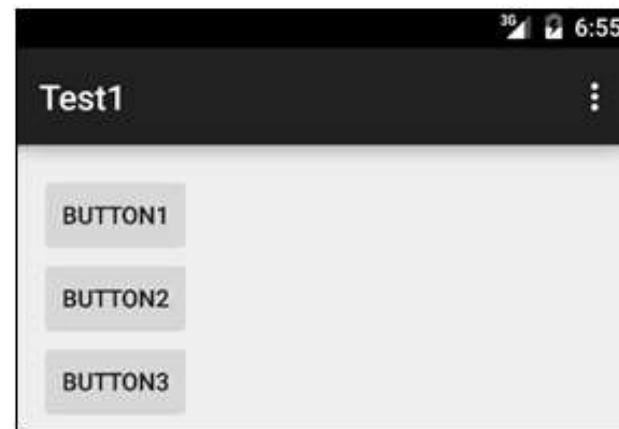


LinearLayout

- 속성
 - orientation
 - layout_gravity 와 gravity
 - layout_weight
 - baselineAligned



orientation="horizontal"



orientation="vertical"

실습 4 : LinearLayout – orientation



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:orientation="horizontal"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <Button
11        android:id="@+id/button1"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="버튼 1" />
15
16    <Button
17        android:id="@+id/button2"
18        android:layout_width="wrap_content"
19        android:layout_height="wrap_content"
20        android:text="버튼 2" />
21
22    <Button
23        android:id="@+id/button3"
24        android:layout_width="wrap_content"
25        android:layout_height="wrap_content"
26        android:text="버튼 3" />
27
28 </LinearLayout>
29
30
```

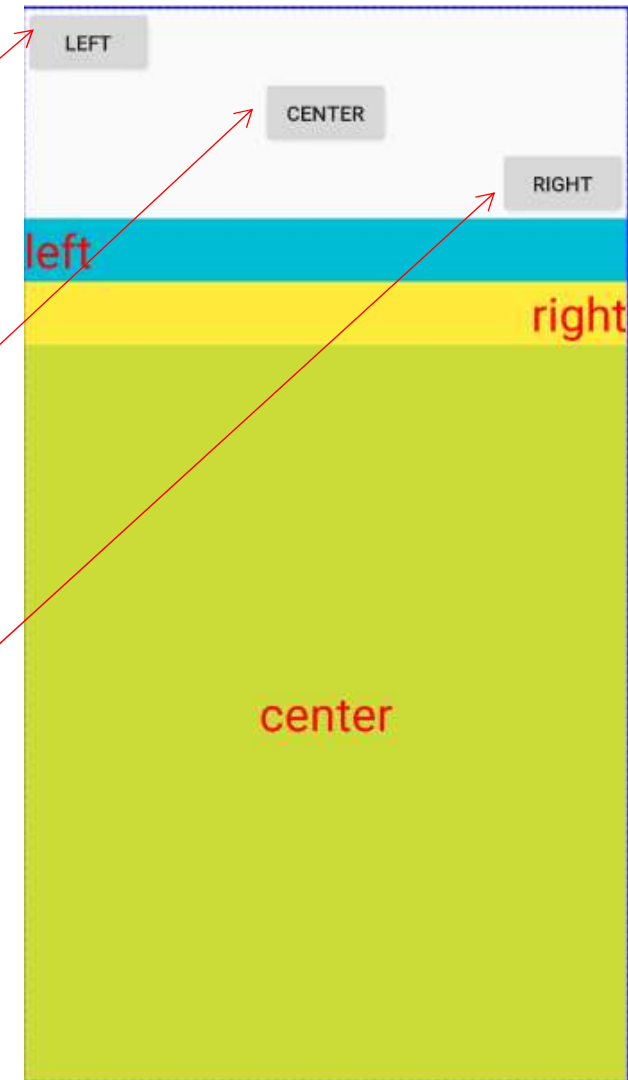


실습 5(a) : LinearLayout – **layout_gravity**

```
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:text="left" />

<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="center" />

<Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="right" />
```

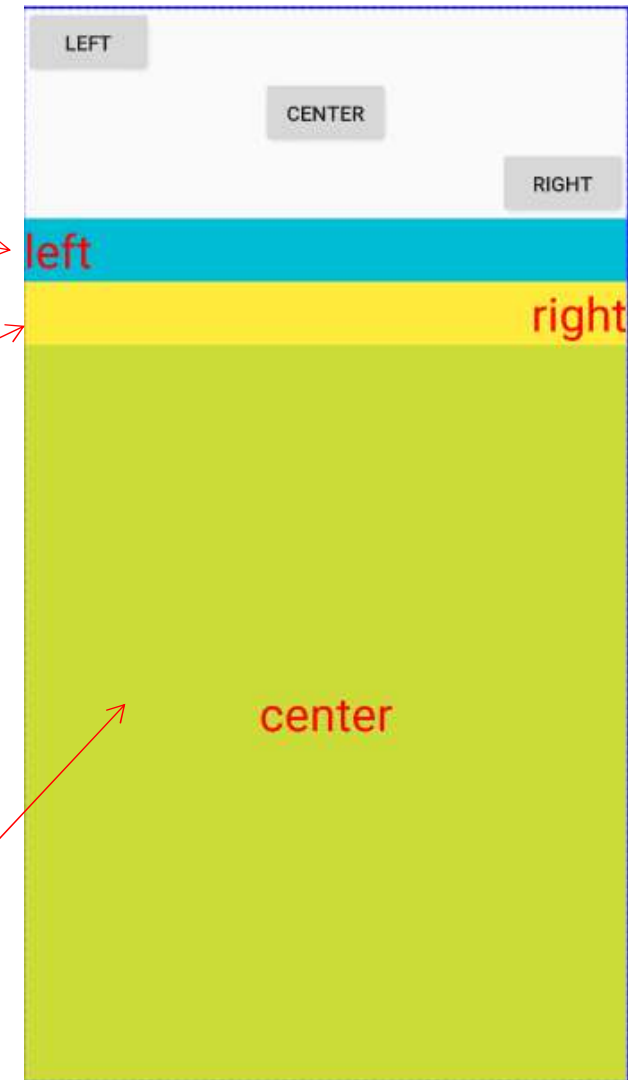


실습 5(b) : LinearLayout – gravity

```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:textColor="#ffff0000"
    android:background="#00BCD4"
    android:textSize="32dp"
    android:text="left" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="right"
    android:textColor="#ffff0000"
    android:background="#FFEB3B"
    android:textSize="32dp"
    android:text="right" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal|center_vertical"
    android:textColor="#ffff0000"
    android:background="#CDDC39"
    android:textSize="32dp"
    android:text="center" />
```



실습 6 : Linear Layout - **layout_weight**

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ffffff00"
        android:text="텍스트"
        android:textColor="#ffff0000"
        android:textSize="24dp"
        android:layout_weight="1" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ff00ffff"
        android:text="텍스트"
        android:textColor="#ffff0000"
        android:textSize="24dp"
        android:layout_weight="1" />
</LinearLayout>
```

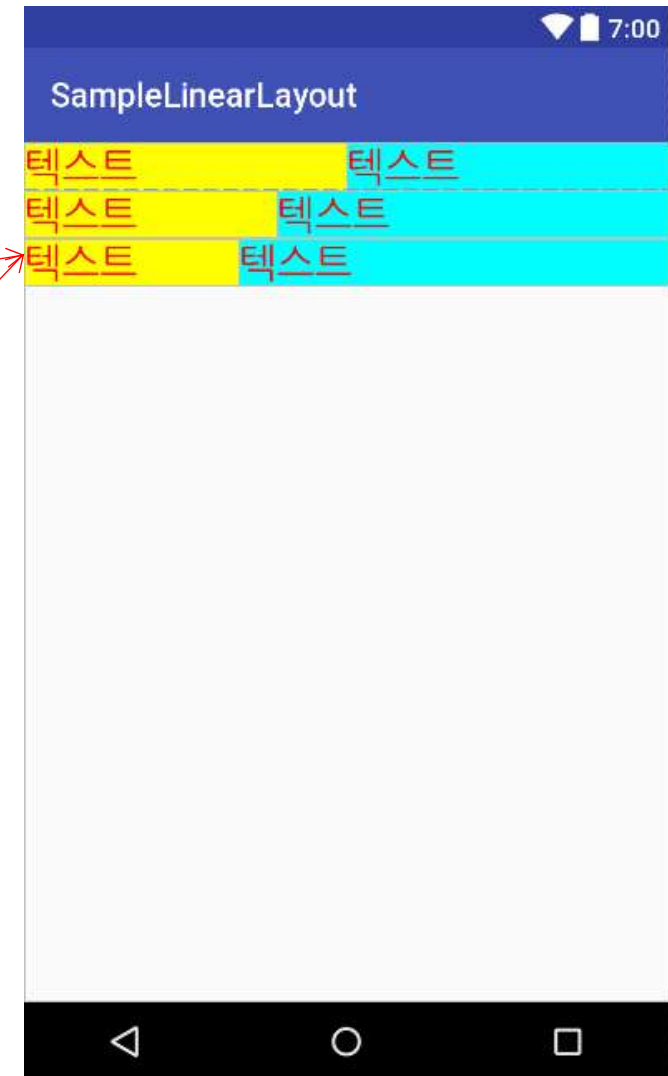
이 값을 2로 바꾸면



실습 6 : Linear Layout - **layout_weight**

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    >
    <TextView
        android:id="@+id/textView5"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="#ffffff00"
        android:text="텍스트"
        android:textColor="#ffff0000"
        android:textSize="24dp"
        android:layout_weight="1" />
    <TextView
        android:id="@+id/textView6"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="#ff00ffff"
        android:text="텍스트"
        android:textColor="#ffff0000"
        android:textSize="24dp"
        android:layout_weight="2" />
</LinearLayout>
```

0dp의 의미는 ?



실습 7 : Linear Layout - **baselineAligned**

- 실습 4에서 **main.xml** 클릭
 - **Design** 탭 선택
 - 3개의 **Button** 모두 선택 > delete
 - **LinearLayout** 의 orientation="**horizontal**"
 - Palette 창 > Widgets > **Plain TextView**
 - 3개의 **TextView** 순서대로 삽입
 - Component tree > textView > Properties 창
 - text="**The**", textColor="#ff0000", textSize="**20sp**"
 - text="**World of**", textColor="#00ff00", textSize="**30sp**"
 - text = "**Android**", textColor = "#0000ff", textSize="**40sp**"
 - Component tree > **LinearLayout** > Properties 창
 - **gravity** 속성 없음
 - **baselineAligned** > check-box 클릭 > 체크 표시

실습 7 : Linear Layout - **baselineAligned**

```
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="horizontal"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:baselineAligned="true">
```

Component Tree

- LinearLayout (horizontal)
 - Ab textView4- "The"
 - Ab textView5- "World of"
 - Ab textView6- "Android"

All Attributes

accessibilityLiveRe		
accessibilityTraver		
accessibilityTraver		
actionBarNavMod		
addStatesFromChi		
alpha		
alwaysDrawnWithC		
animateLayoutCha		
animationCache		
background		
backgroundTint		
backgroundTintMc		
baselineAligned	<input checked="" type="checkbox"/>	true

TheWorld ofAndroid

TheWorld ofAndroid

RelativeLayout

- 부모 container나 다른 view의 위치를 기준으로 자신의 위치를 결정
 - 부모 container나 다른 view의 위치는 어떻게 알 수 있을까?
 - RelativeLayout에서는 모든 view에 대해 id를 정의
 - “@+id/*identifier*”
 - 다른 view의 id를 참조 → “@+id/ *다른 view의 id*”
 - 상대 위치를 지정하는 속성
 - layout_alignParentTop, layout_alignParentBottom
 - layout_alignParentLeft, layout_alignParentRight
 - layout_above, layout_below
 - layout_toLeftOf, layout_toRightOf
 - layout_alignTop, layout_alignBottom, ...
 - layout_alignBaseline

실습 8 : RelativeLayout

```
<TextView
    android:id="@+id/address"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:text="주소를 입력하세요" />

<EditText
    android:id="@+id/input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/address"
    android:inputType="text" />

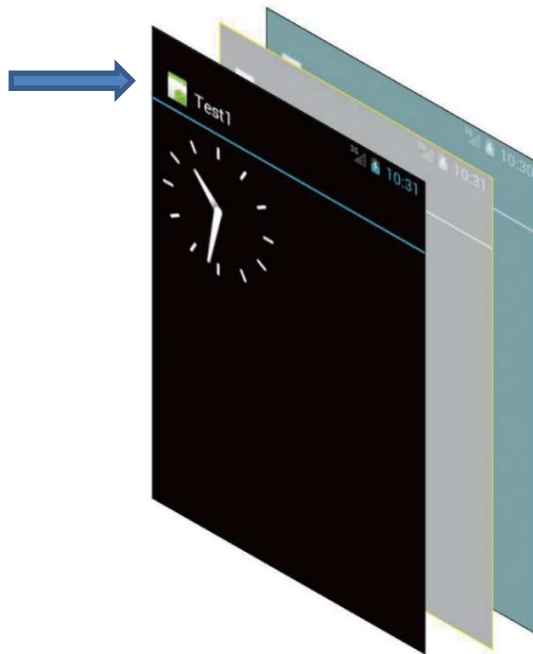
<Button
    android:id="@+id/cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_below="@+id/input"
    android:text="취소" />

<Button
    android:id="@+id/confirm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_below="@+id/input"
    android:layout_marginEnd="100dp"
    android:text="확인" />
```

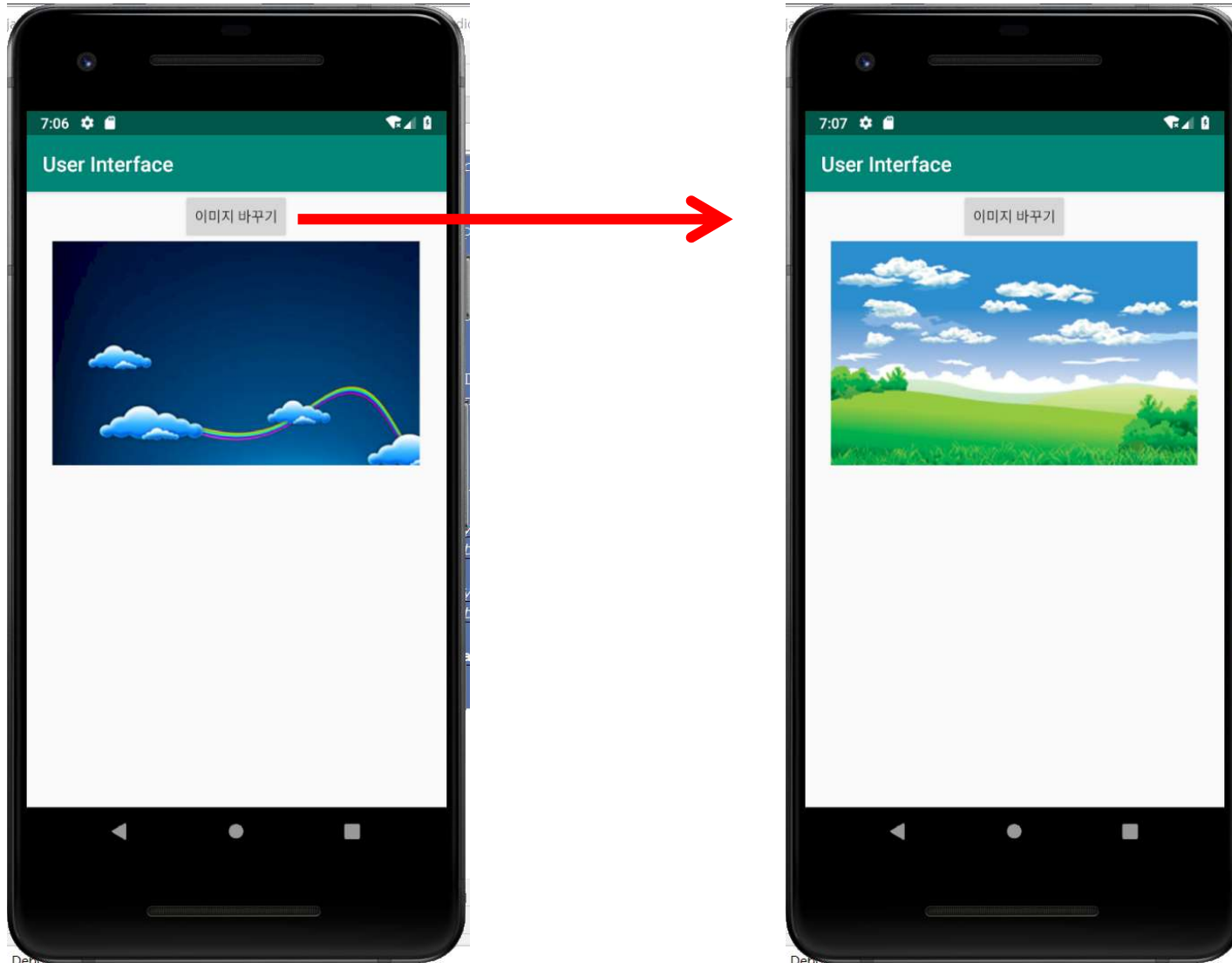


FrameLayout

- 여러 개의 view를 겹쳐서 배치
 - 배치 기준은 맨 위 상단 (*upper left corner*)
 - 선언한 순서대로 배치
 - Child view의 가시성(visibility)은 속성 값을 설정
android: **visibility** = "**visible**"
button1. **setVisibility** (View. **INVISIBLE**);

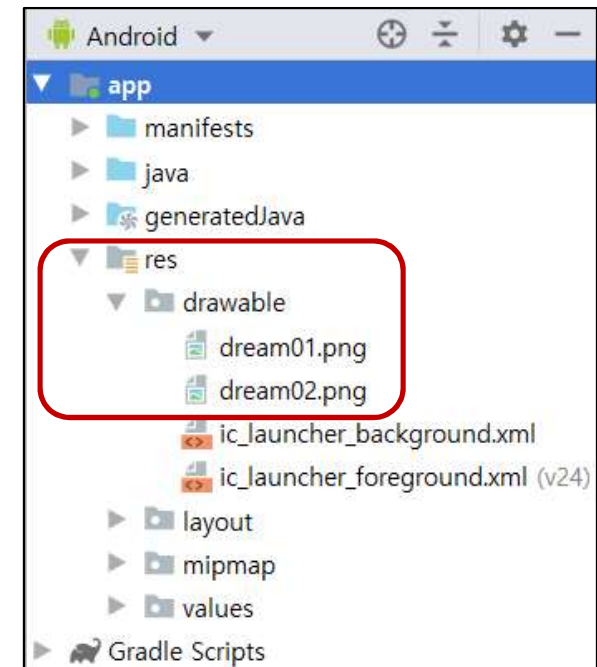
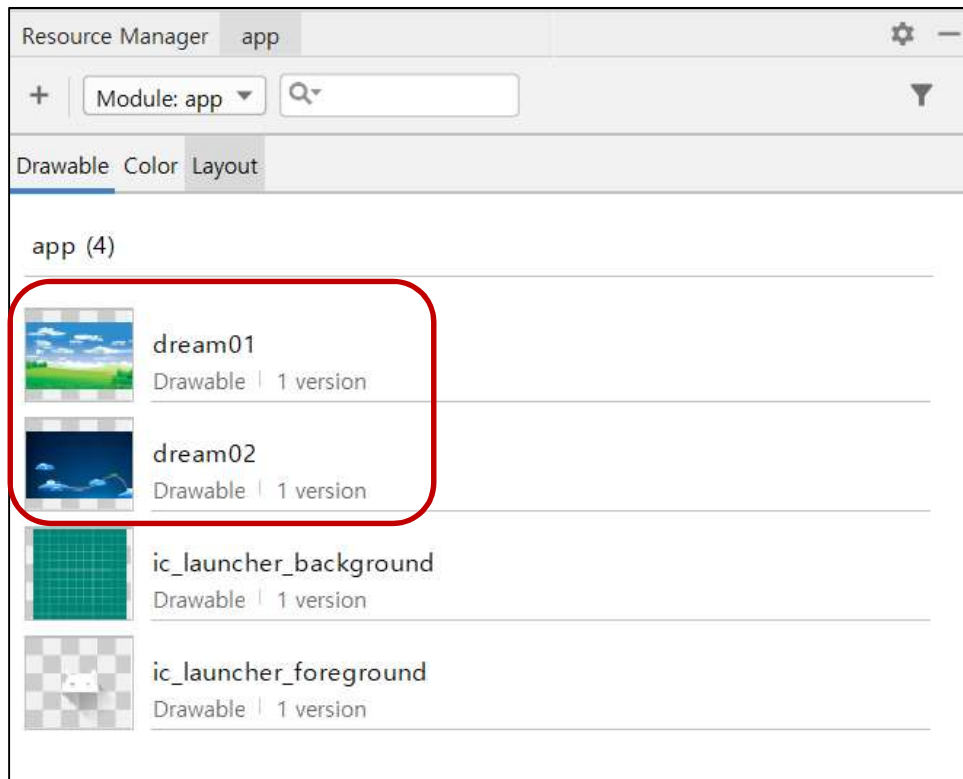


실습 9 : FrameLayout



실습 9 : FrameLayout – drawable res

View > Tool Windows > Resource Manager



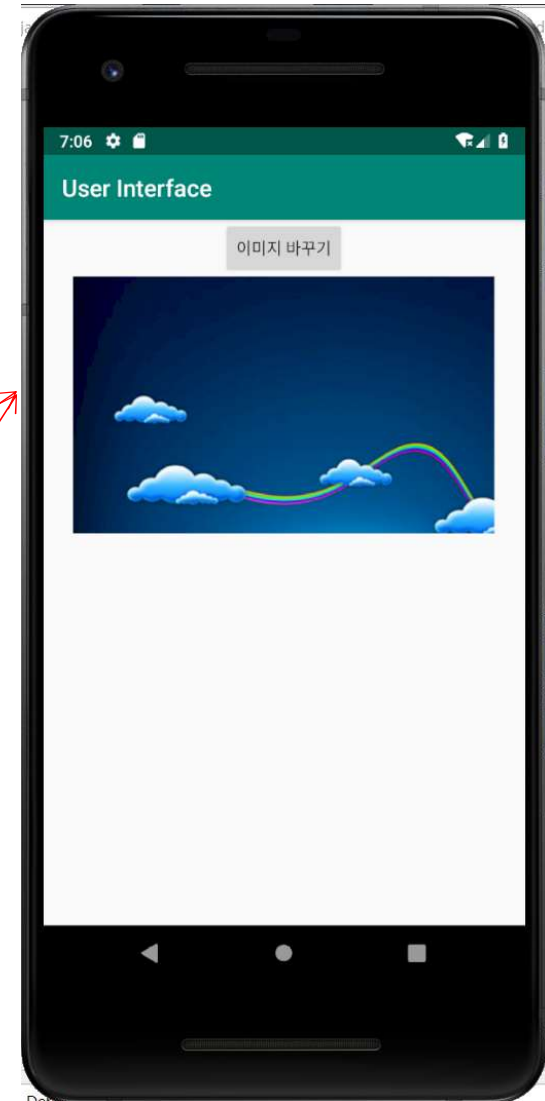
실습 9 : FrameLayout – Layout

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="이미지 바꾸기"
    android:onClick="onButtonClicked" />
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/dream01"
        android:layout_gravity="center"
        android:visibility="invisible" />
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/dream02"
        android:layout_gravity="center"
        android:visibility="visible" />
```

visibility
속성은?

visibility
속성은?

클릭 이벤트
핸들러 선언



실습 9 : **FrameLayout** - **MainActivity**

```
class MainActivity : AppCompatActivity() {  
    var toggleImage = true  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
  
    fun onClicked(view: View) {  
        if (toggleImage) {  
            imageView.visibility = View.VISIBLE  
            imageView2.visibility = View.INVISIBLE  
        } else {  
            imageView.visibility = View.INVISIBLE  
            imageView2.visibility = View.VISIBLE  
        }  
        toggleImage = !toggleImage  
    }  
}
```

실습 10(a): TableLayout

- <TableRow> 태그 = table의 한 개의 행
- 속성
 - stretchColumns, shrinkColumns
 - layout_column, layout_span

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <TextView...>
        <EditText...>
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <TextView...>
        <EditText...>
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Button...>
        <Button...>
    </TableRow>
</TableLayout>
```

stretchColumns="0,1" 을 추가하면?

주소	인천시 연수구 송도동
이름	인천대학교
저장	취소

실습 10(b) : TableLayout

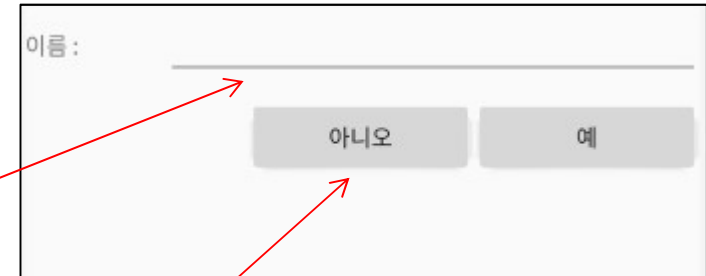
```
<TableLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:stretchColumns="0,1,2,3">
```

```
<EditText  
    android:id="@+id/editText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_span="3" />
```

layout_span의 의미는?

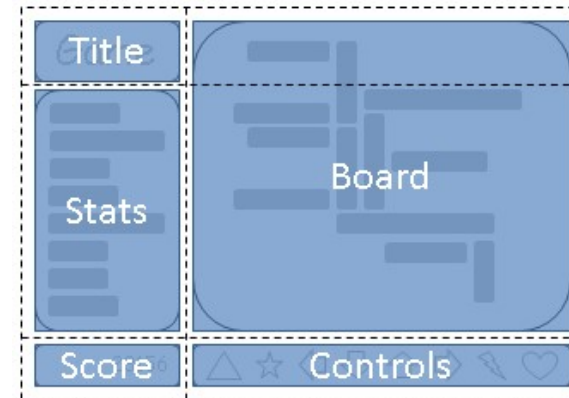
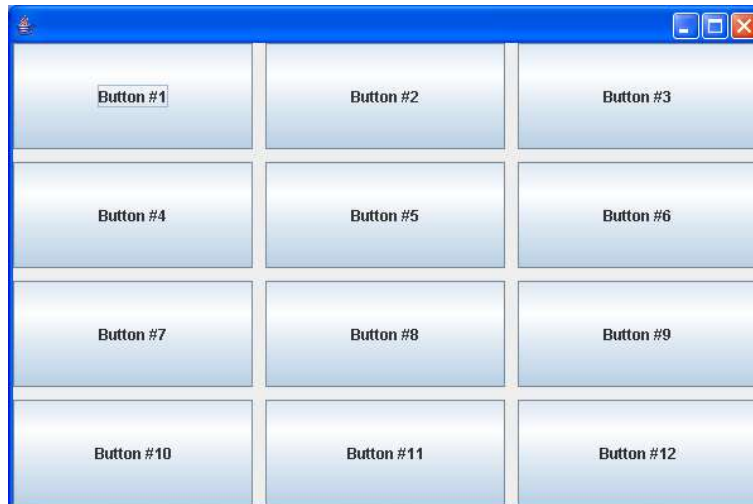
```
<TableRow  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginTop="10dp">  
  
    <Button  
        android:id="@+id/button"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_column="2"  
        android:text="아니오" />
```

layout_column의 의미는?



GridLayout

- 자식 view를 바둑판 모양 grid 에 배치 : Android 4.0부터 도입
- 용어
 - 행(row), 열(column), 셀(cell)
 - 첫 번째 행의 index는 0, 첫 번째 열의 index는 0
 - (2,0) : 세 번째 행 첫 번째 열의 셀을 가리킴
- 속성
 - **rowCount, columnCount**
 - **rowSpan, columnSpan**
 - **layout_low, layout_column**



실습 11: GridLayout (1)

```
<GridLayout
    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:columnCount="2"
    android:rowCount="2"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <Button
        android:text="Button 1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button1"/>

    <Button
        android:text="Button 4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button4"/>

</GridLayout>
```



실습 12: GridLayout (2)

columnCount, rowCount 속성 값 없앰
Button의 위치 및 차지하는 크기를 직접 지정

```
<GridLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:layout_column="0"
        android:layout_row="0" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 3"
        android:layout_column="2"
        android:layout_gravity="fill_vertical"
        android:layout_row="0"
        android:layout_rowSpan="2" />
```

