자바로 레이아웃 구성하기

## UI를 작성하는 3가지 방법

- □ XML 파일로 사용자 인터페이스를 작성하는 방법
- □ 코드로 사용자 인터페이스를 작성하는 방법
- □ XML과 코드를 동시에 사용하는 방법

## 액티비티

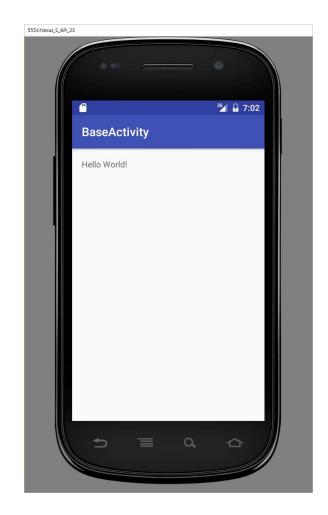
```
public class MainActivity extends AppCompatActivity
{
    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_main );
    }
}
```

```
res/layout/activity_main.xml

<RelativeLayout xmlns:android="http://..."
    android:layout_width="match_parent"
    android:layout_height="match_parent">

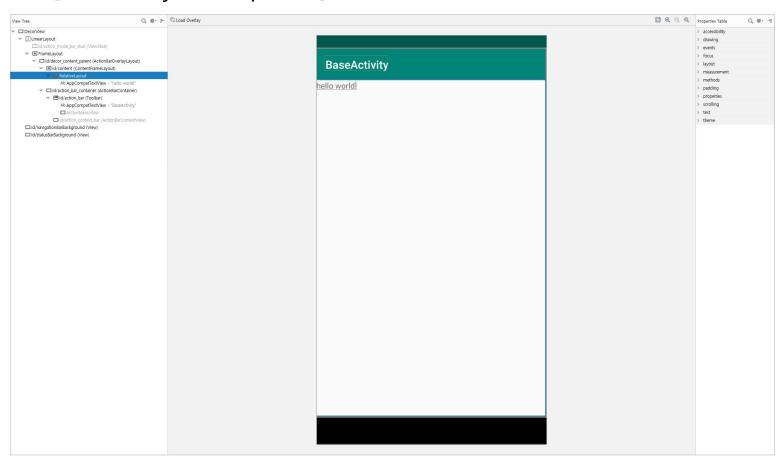
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="hello world!" />

</RelativeLayout>
```

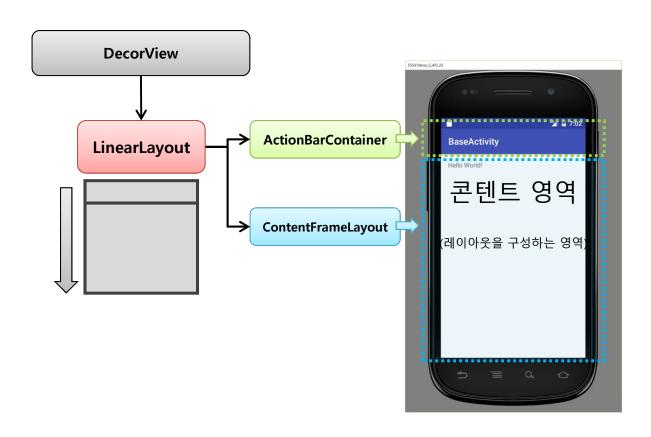


# 액티비티 기본 화면 구성 분석

- Android Studio
  - [Tools→Layout Inspector]



# 액티비티 기본 화면 구성 분석



# 콘텐트 영역을 채워보자

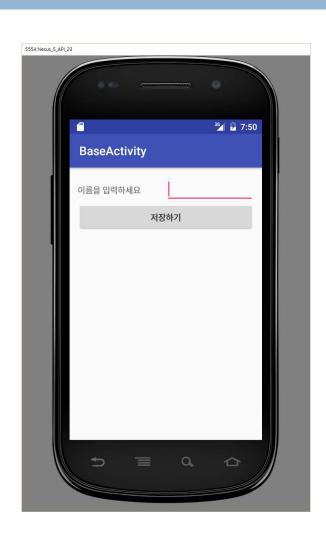
```
res/layout/activity main.xml
<LinearLayout _____
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout height="match parent"
    android:layout margin="20px">
    <LinearLayout android:orientation="horizontal"</pre>
        android:layout width="match parent"
       android:layout height="wrap content">
       <TextView android:text="이름을 입력하세요"
            android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
        <EditText android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
    </LinearLayout>
    <Button android:text="저장하기"
       android:layout width="match parent"
       android:layout height="wrap content"/>
</LinearLayout>
```

C:₩Users₩iamjw₩AppData₩Local₩Android₩sdk₩platforms₩android-24₩data₩res₩values₩attrs.xml

```
public class MainActivity extends Activity
{
    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_main );
    }
}
```

activity\_main.xml의 id

# 콘텐트 영역을 채워보자



## setContentView 함수의 역할



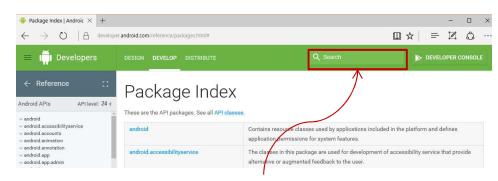
- 인플레이션(Inflation)
  - XML 레이아웃에 정의된 정보를 메모리 상에서 객체로 만드는 객체화 과정

- setContentView 함수는 레이아웃 XML의 내용을 파싱하여 뷰들을 생성하고, 뷰에 정의된 속성들을 설정한다. 또한 생성된 뷰들을 상하관계에 맞춰 배치한다. 참고로 setContentView 함수 내에 이러한 처리는 모두 LayoutInflater라는 클래스에 의존한다.
- 레이아웃 XML의 처리 결과로 생성된 뷰들을 콘텐트 영역에 추가한다.

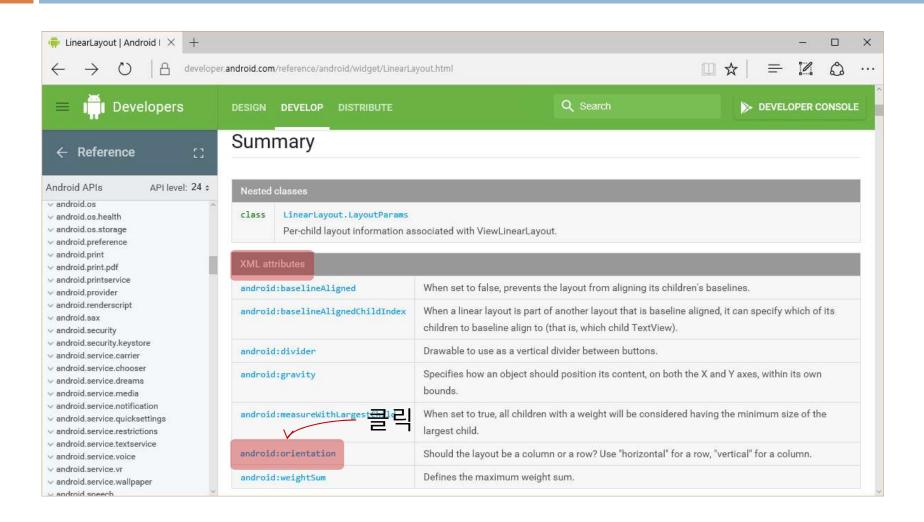
저장하기

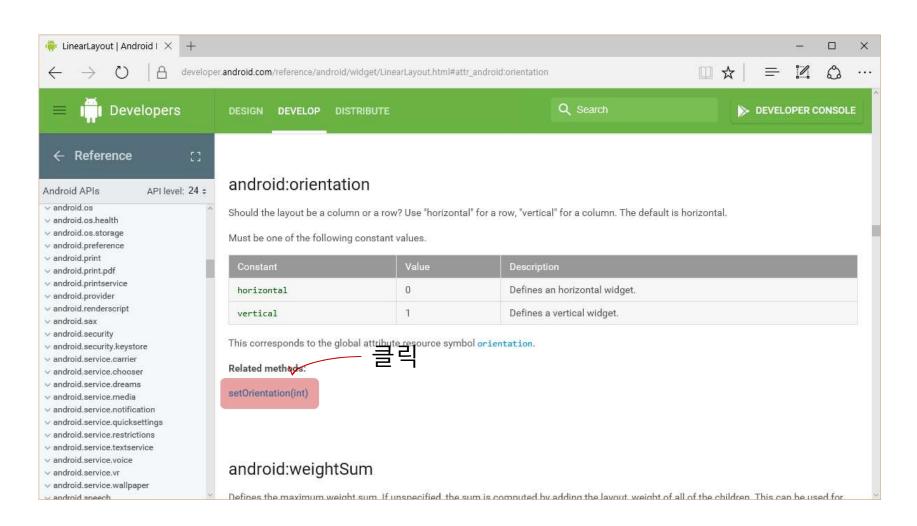
```
res/layout/activity_main.xml
<LinearLayout xmlns:android="http://...</pre>
    android:orientation="vertical"
    android:layout width="match parent"
    android:layout height="match parent"
    android:layout margin="20px">
    <LinearLayout android:orientation="horizontal"</pre>
        android:layout width="match parent"
        android:layout height="wrap content">
        <TextView android:text="이름을 입력하세요"
            android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
        <EditText android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
    </LinearLayout>
    <Button android:text="저장하기"
        android:layout width="match parent"
        android:layout height="wrap content"/>
</LinearLayout>
```

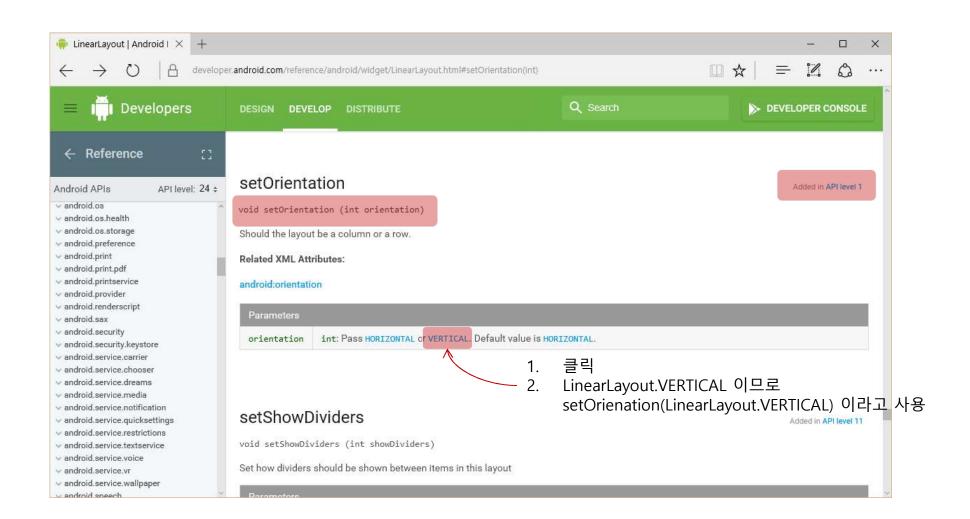
#### https://developer.android.com/reference 방문



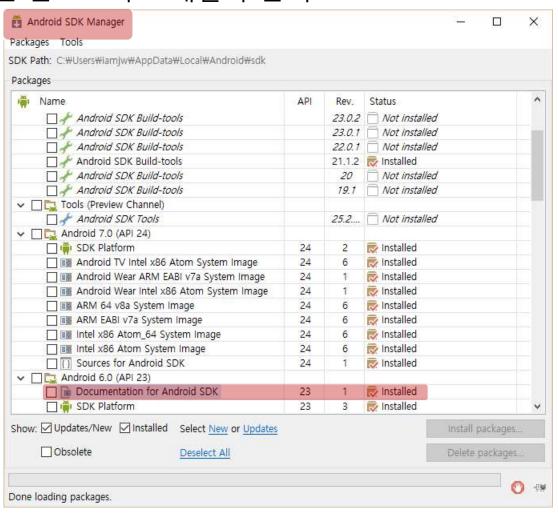
- 1. LinearLayout 입력
- 2. 추천어들 중에서 android.widget.LinearLayout 선택

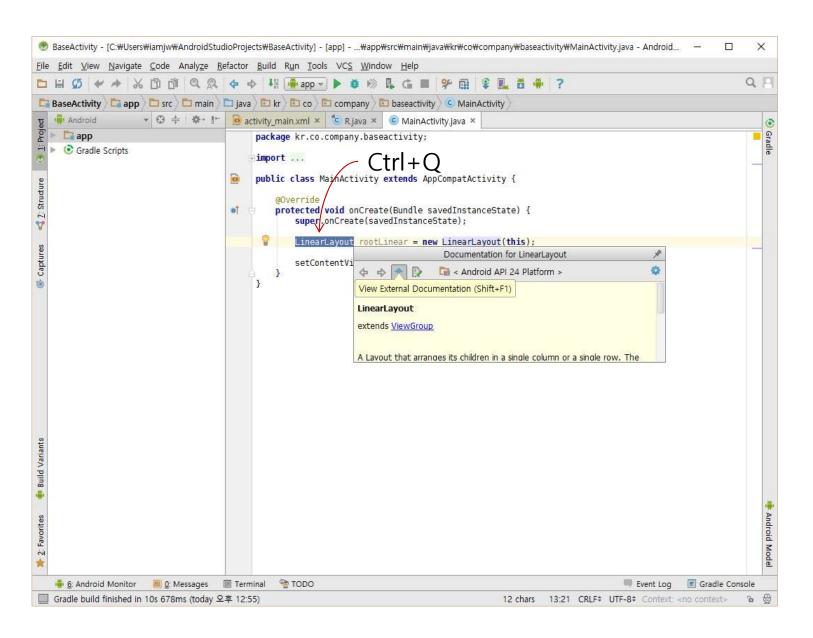




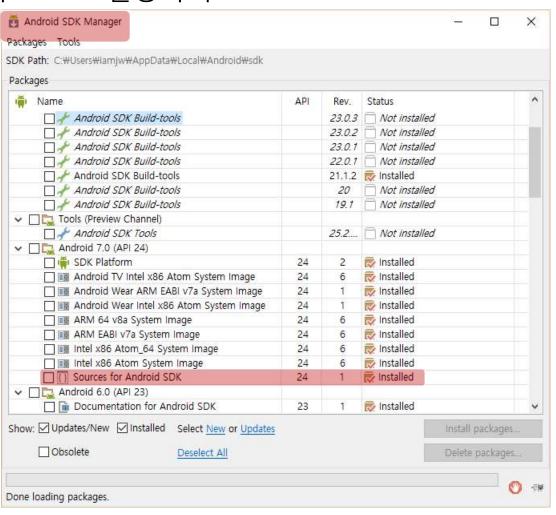


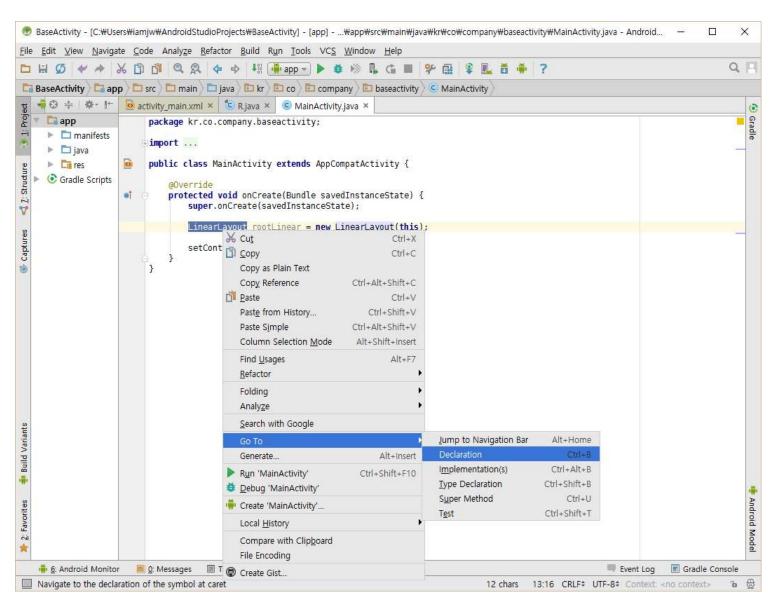
#### □ 오프라인 안드로이드 개발자 문서

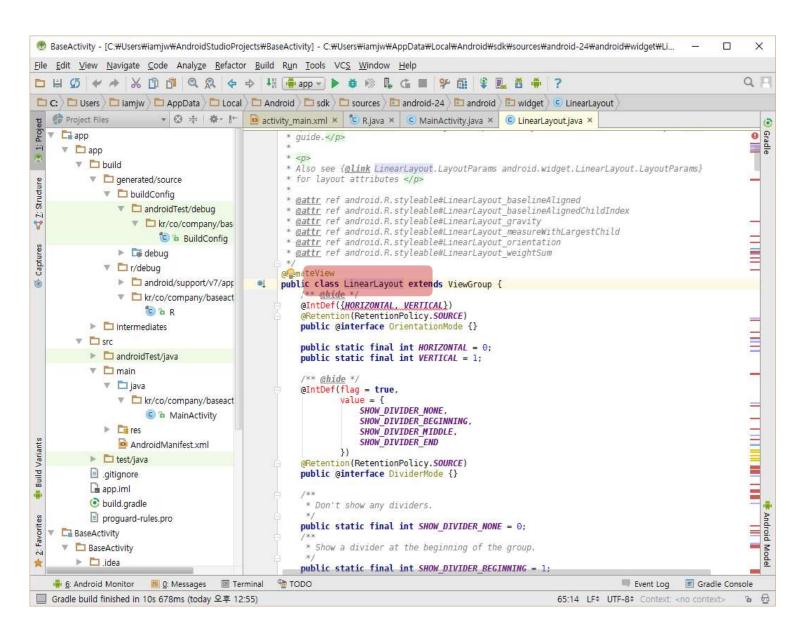




#### □ 프레임워크 소스 활용하기







첫 번째 과정: 전체 뷰를 감싸고 있는 수직 배치 LinearLayout을 자바 소스로 옮긴다.

#### res/layout/activity main.xml <LinearLayout xmlns:android="http://..."</pre> android:orientation="vertical" android:layout width="match parent" android:layout height="match parent" android:layout margin="20px"> <LinearLayout android:orientation="horizontal"</pre> android:layout width="match parent" android:layout height="wrap content"> <TextView android:text="이름을 입력하세요" android:layout width="0dp" android:layout height="wrap content" android:layout weight="1"/> <EditText android:layout width="0dp" android:layout height="wrap content" android:layout weight="1"/> </LinearLayout> <Button android:text="저장하기" android:layout\_width="match parent" android:layout height="wrap content"/> </LinearLayout>

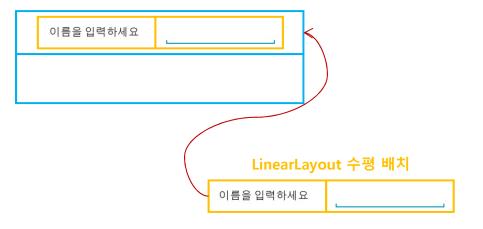
#### LinearLayout 수직 배치

이름을 입력하세요 전 장하기

첫 번째 과정: 전체 뷰를 감싸고 있는 수직 배치 LinearLayout을 자바 소스로 옮긴다.

두 번째 과정: 이름 입력 부분의 레이아웃을 자바 소스로 옮긴다.

```
res/layout/activity main.xml
<LinearLayout xmlns:android="http://..."</pre>
   android:orientation="vertical"
   android:layout width="match parent"
   android:layout height="match parent"
   android:layout_margin="20px">
   <LinearLayout android:orientation="horizontal"</pre>
       android:layout width="match parent"
       android:layout height="wrap content">
       <TextView android:text="이름을 입력하세요"
            android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
        <EditText android:layout width="0dp"
            android:layout_height="wrap_content"
            android:layout weight="1"/>
   </LinearLayout>
   <Button android:text="저장하기"
       android:layout width="match parent"
       android:layout height="wrap content"/>
</LinearLayout>
```



두 번째 과정: 이름 입력 부분의 레이아웃을 자바 소스로 옮긴다.

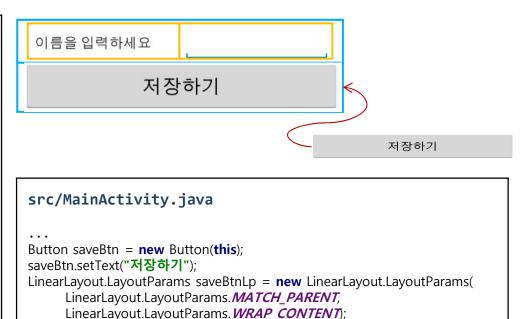
```
src/MainActivity.java
LinearLayout nameInputLinear = new LinearLayout(this);
nameInputLinear.setOrientation(LinearLayout. HORIZONTAL);
LinearLayout.LayoutParams nameInputLp =
     new LinearLayout.LayoutParams(
           LinearLayout.LayoutParams.MATCH PARENT,
           LinearLayout.LayoutParams. WRAP_CONTENT);
TextView nameTv = new TextView(this);
nameTv.setText("이름을 입력하세요");
LinearLayout.LayoutParams nameTextLp = new LinearLayout.LayoutParams(
     LinearLayout.LayoutParams. WRAP_CONTENT);
nameTextLp.weight = 1;
EditText nameEt = new EditText(this);
LinearLayout.LayoutParams nameEditLp = new LinearLayout.LayoutParams(
     LinearLayout.LayoutParams. WRAP CONTENT);
nameEditLp.weight = 1;
nameInputLinear.addView(nameTv, nameTextLp);
nameInputLinear.addView(nameEt, nameEditLp);
rootLinear.addView(nameInputLinear, nameInputLp);
```

</LinearLayout>

## 자바 소스로 구현하기

세 번째 과정: 저장하기 버튼을 자바 소스로 옮긴다.

```
res/layout/activity main.xml
<LinearLayout xmlns:android="http://..."</pre>
   android:orientation="vertical"
    android:layout width="match parent"
    android:layout height="match parent"
    android:layout margin="20px">
   <LinearLayout android:orientation="horizontal"</pre>
        android:layout width="match parent"
        android:layout height="wrap content">
        <TextView android:text="이름을 입력하세요 "
            android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
        <EditText android:layout width="0dp"
            android:layout height="wrap content"
            android:layout weight="1"/>
    </LinearLayout>
    <Button android:text="저장하기"
        android:layout width="match parent"
        android:layout height="wrap content"/>
```



rootLinear.addView(saveBtn, saveBtnLp);

### 자바 소스로 구현하기 네 번째 과정: 전체 레이아웃을 콘텐트 영역에 추가한다.

src/MainActivity.java
...
setContentView(rootLinear, rootLp);





# XML과 코드를 동시에 사용하는 방법

- 1. 프로젝트 UserInterface3를 생성한다.
- activity\_main.xml을 다음과 같이 수정한다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
 android:layout_width="match_parent"
  android:layout height="match parent"
  android:orientation="vertical">
  <Button
    android:layout width="match parent"
    android:layout height="wrap content"
    android:text="첫 번째 버튼"
    android:id="@+id/button1"/>
  <Button
                                            버튼에 식별자를 부여한다.
    android:layout width="match parent"
    android:layout_height="wrap_content/"
    android:text="두 번째 버튼"
    android:id="@+id/button2"/>
</LinearLayout>
```



# XML과 코드를 동시에 사용하는 방법

3. MainActivity.java 파일을 다음과 같이 수정한다.

```
package kr.co.company.userinterface3;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

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

    Button b1 = (Button) findViewById(R.id.button1);
        b1.setText("코드에서도 변경 가능");

    Button b2 = (Button) findViewById(R.id.button2);
        b2.setEnabled(false);
    }
}

id7h button1인 버튼을 찾는다.
```

