



[한국ICT인재개발원] 안드로이드

2. 안드로이드 프로젝트 생성해보기

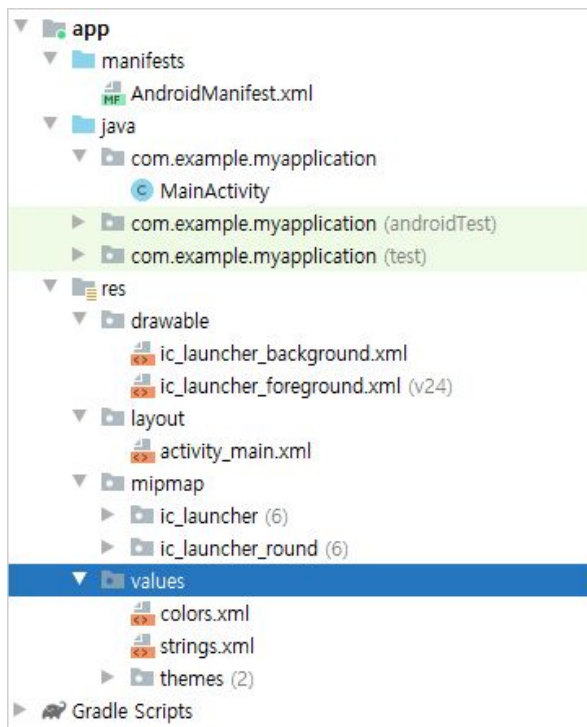
前) 광고데이터 분석 1년

前) IT강의 경력 2년 6개월

前) 머신러닝을 활용한 데이터 분석 프로젝트반 운영 1년

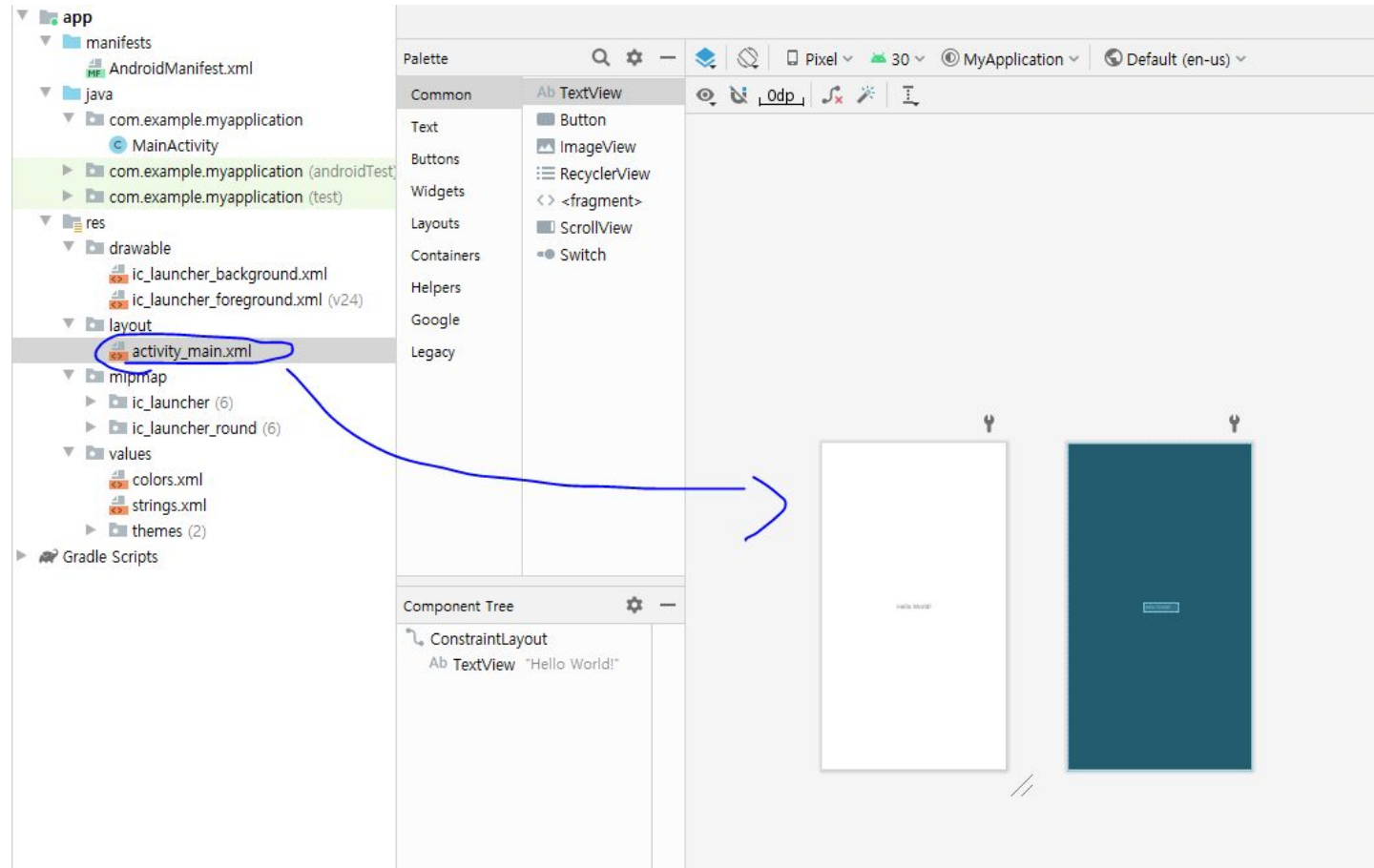
前) 리그오브 레전드 데이터 분석 등...

現) 국비반 강의 진행중

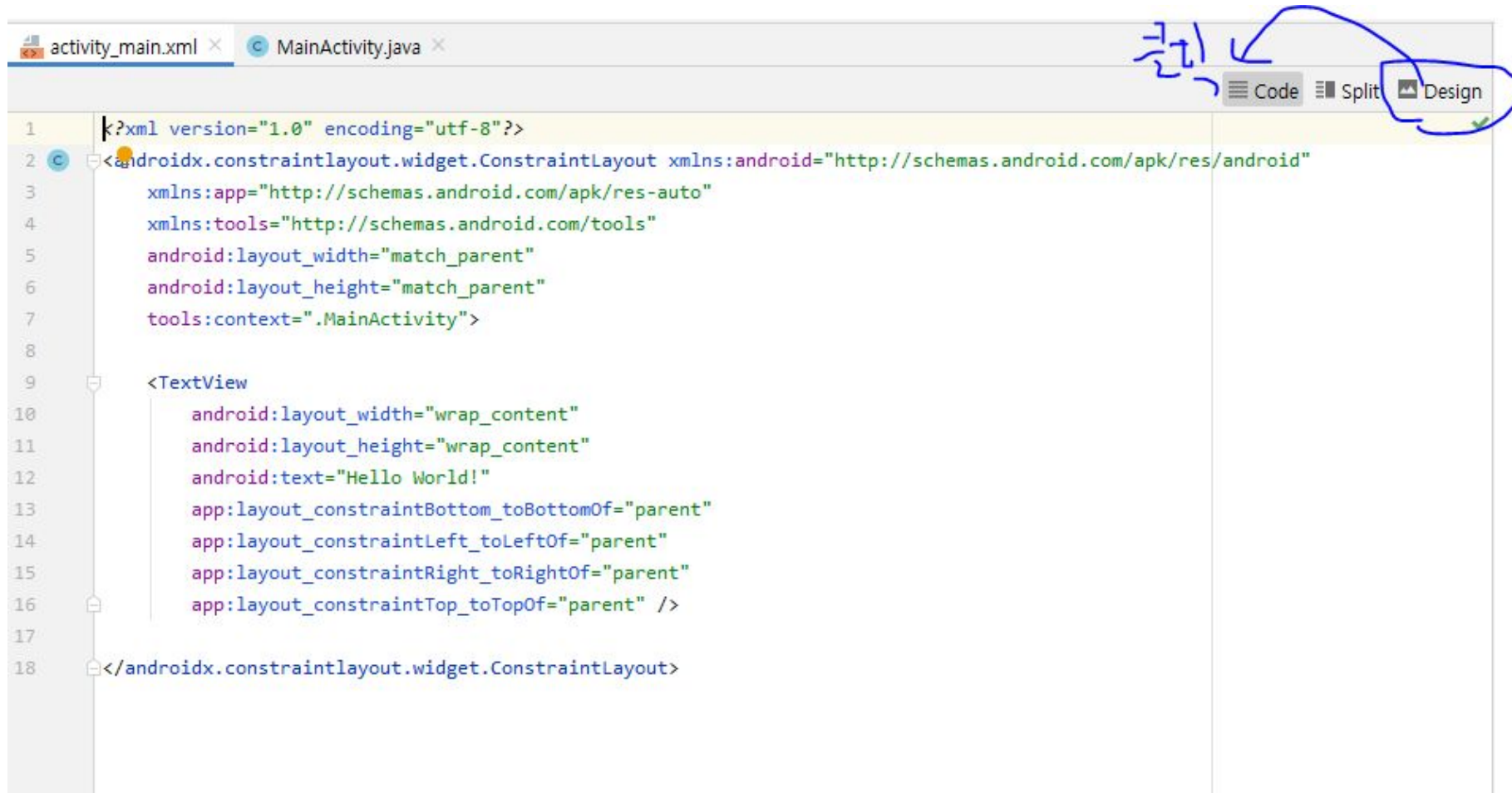


이제 만들어진 **My_Application.app**를 이용해 프로젝트를 간단히 체험해보겠습니다.

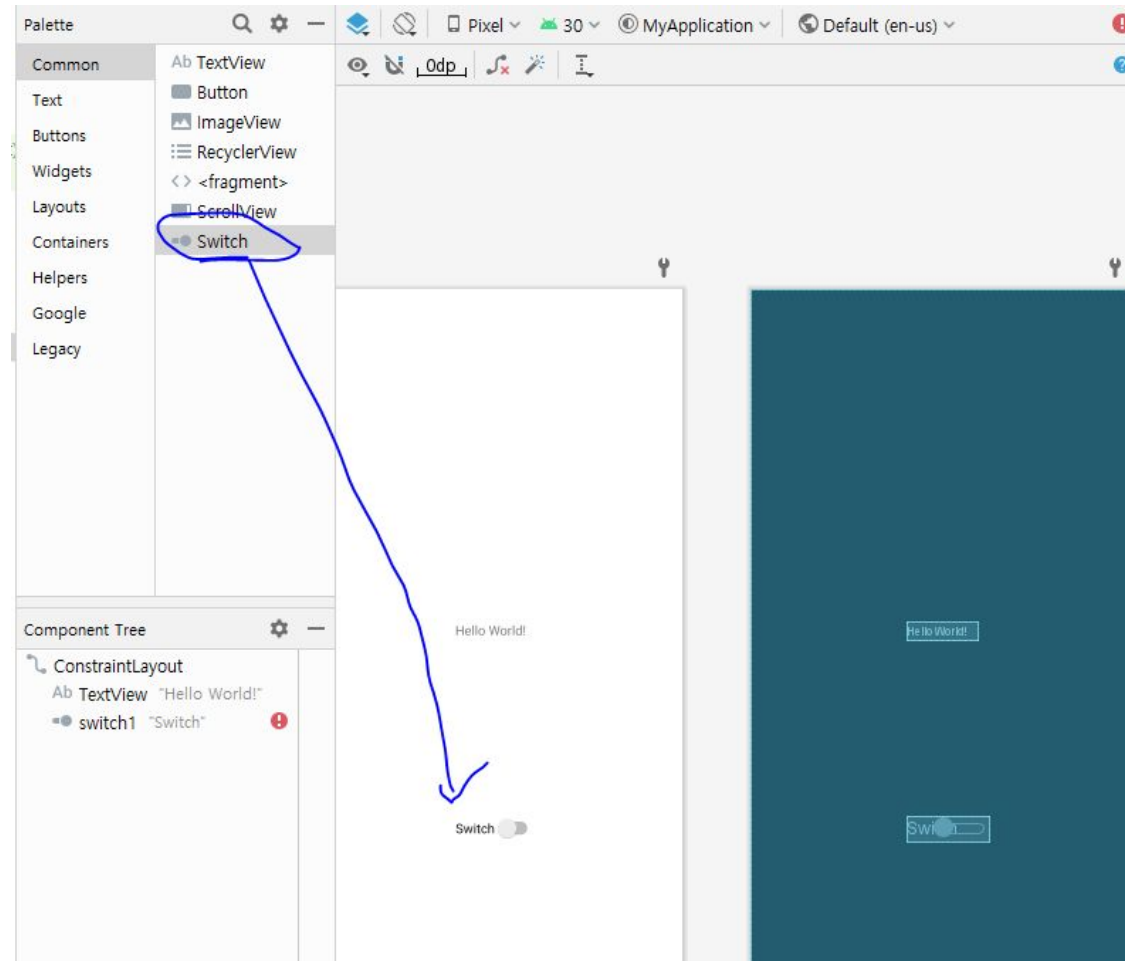
먼저, 화면 디자인 및 레이아웃을 **.xml**파일들이 담당하고
웹으로 치면 컨트롤러와 모델 역할을 자바 클래스가 담당합니다.



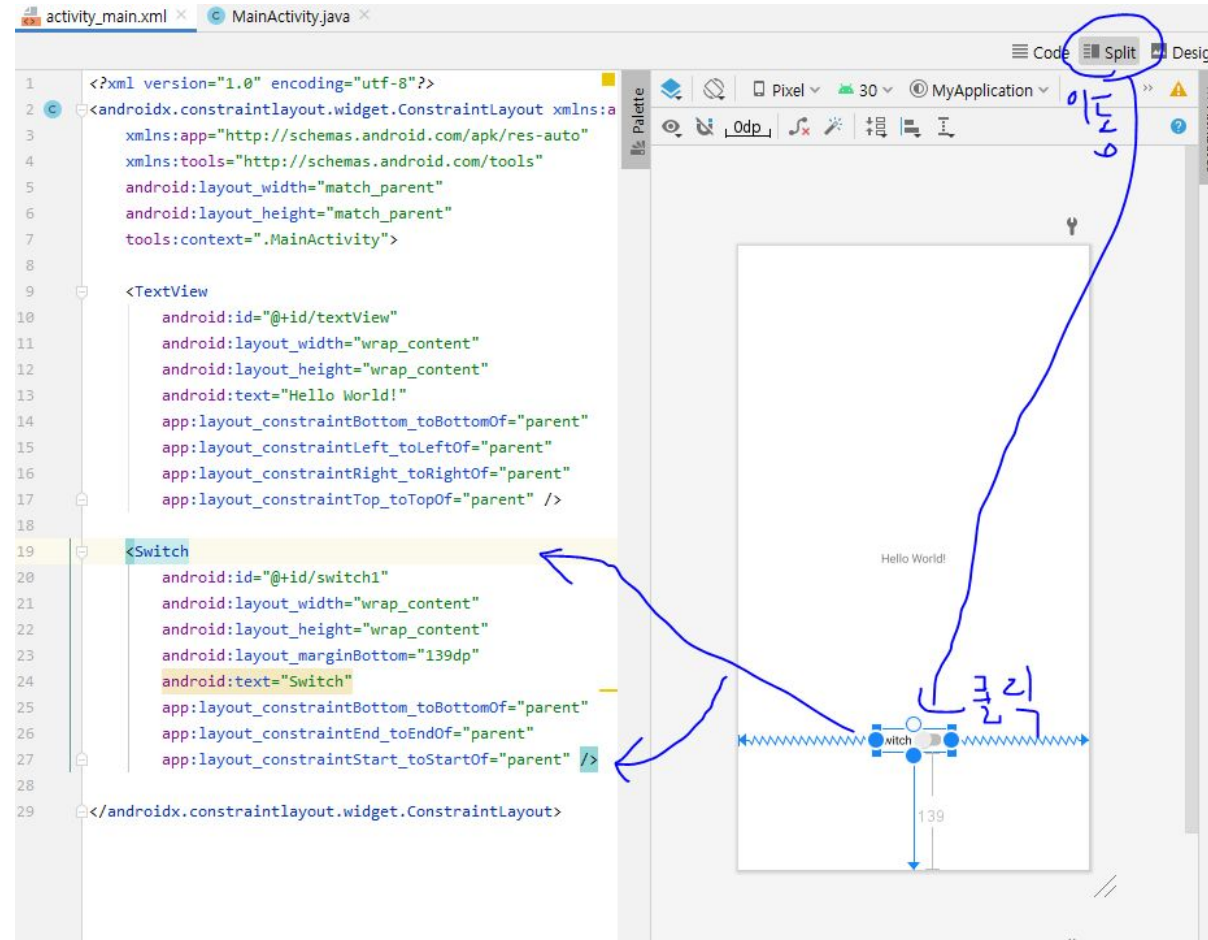
가장 먼저, **res**폴더의 **layout** 하위 **activity_main.xml**을 열어줍니다.
그러면 화면 오른쪽에 레이아웃이 표시됩니다.



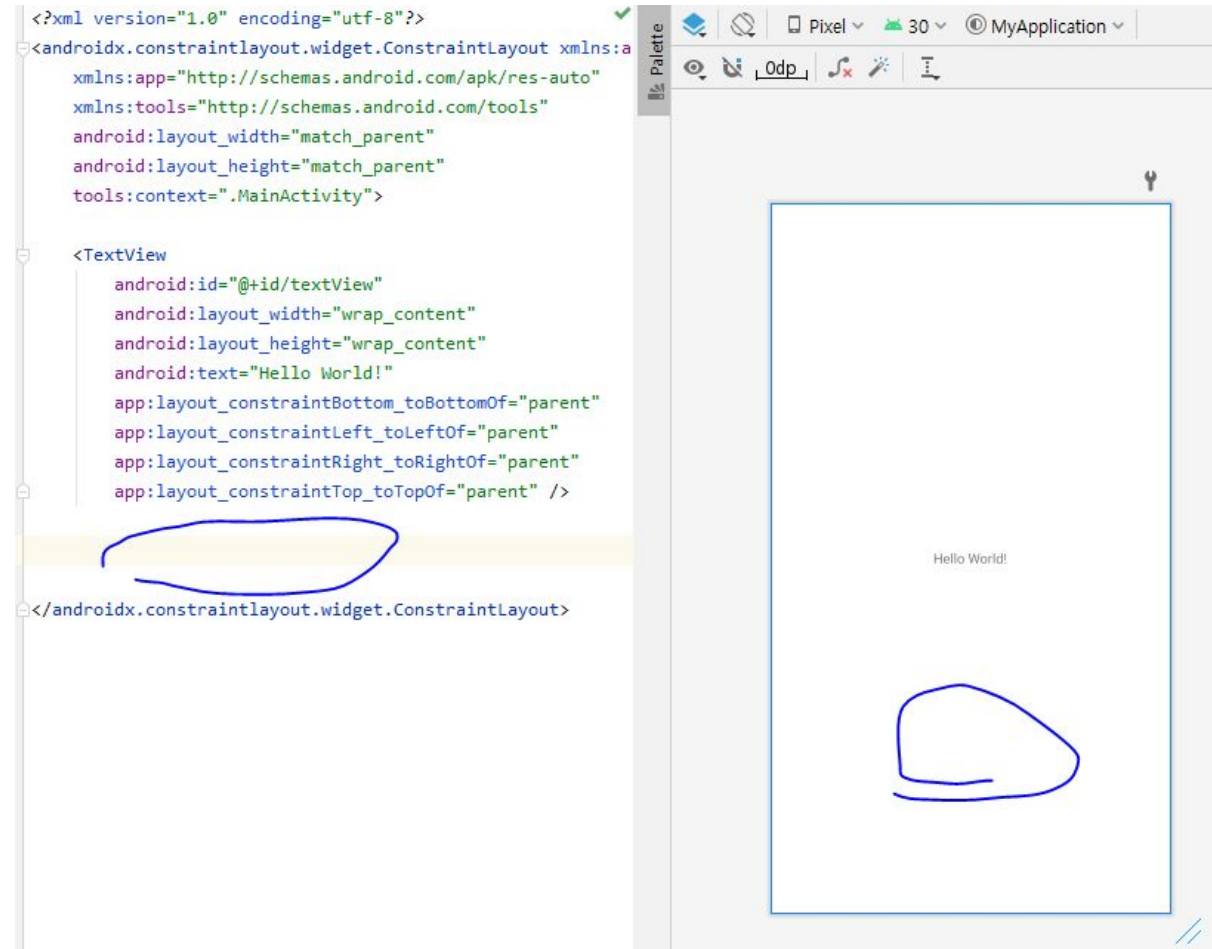
내부코드는 우상단의 Code탭으로 이동시 확인가능합니다.



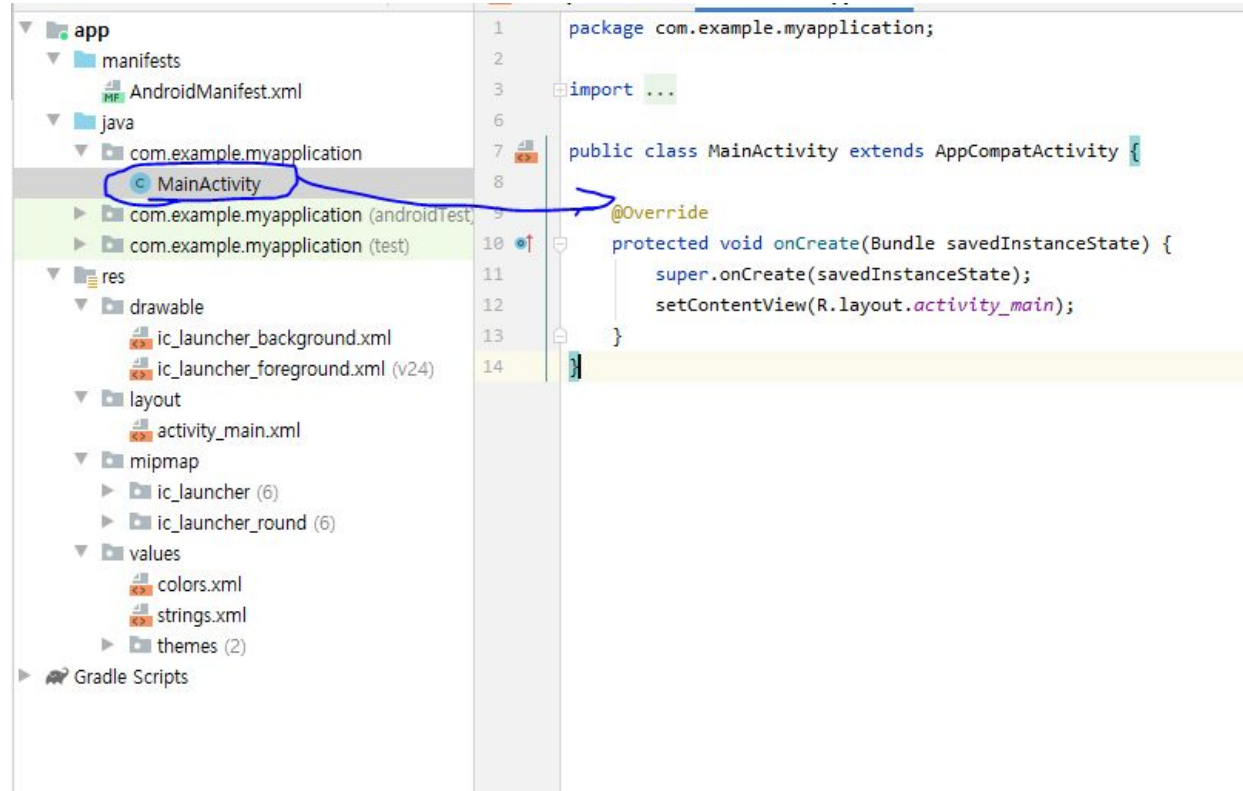
웹 개발과 가장 큰 차이점은 **Design**탭에서 창부품이라고 불리는 여러 요소를 클릭해 드래그 앤 드롭으로 배치할 수 있다는 점입니다.



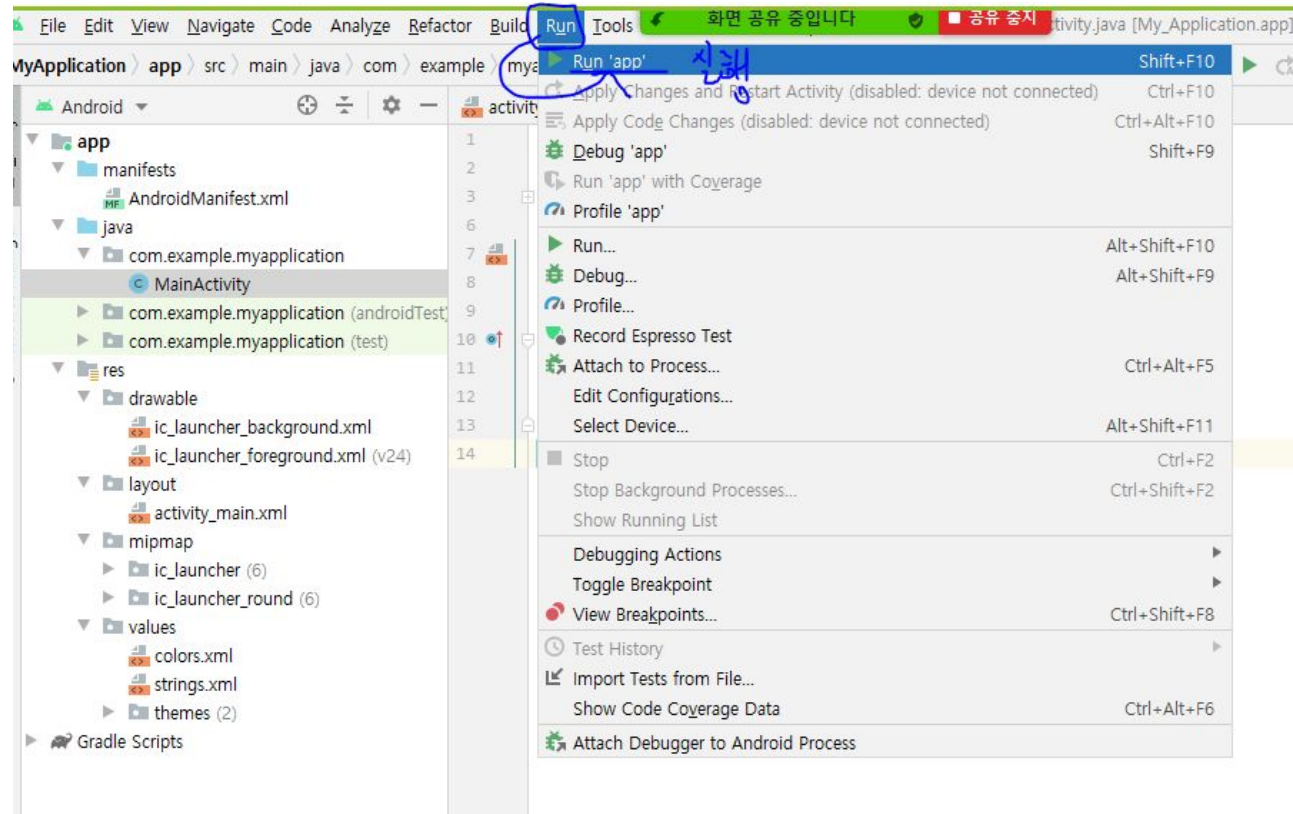
다음 Split탭으로 이동해서 레이아웃 내부의 창부품을 클릭하면 해당 창부품이 코드적으로 어떻게 표시되는지도 확인할 수 있습니다.



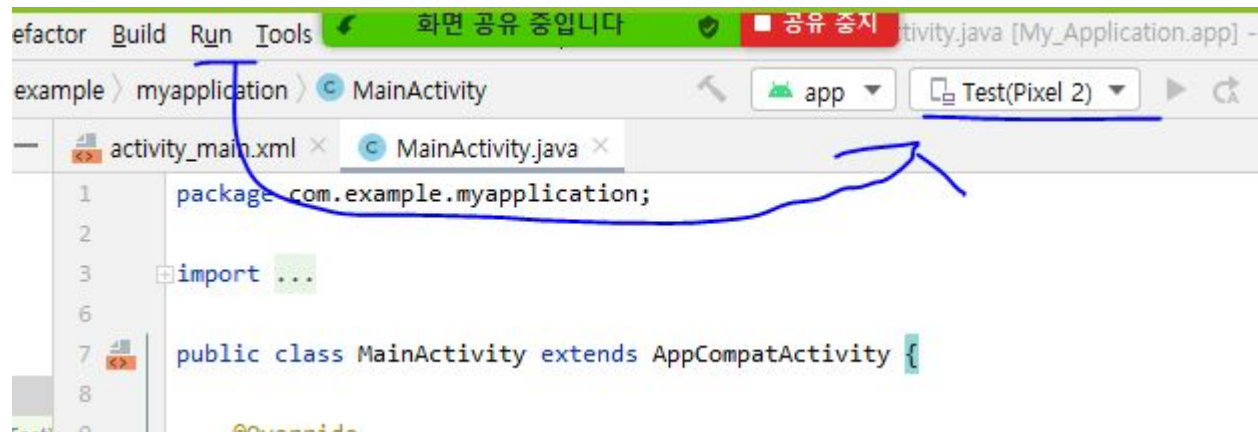
좌측 코드 삭제시 우측 화면에서도 역시 삭제됩니다.



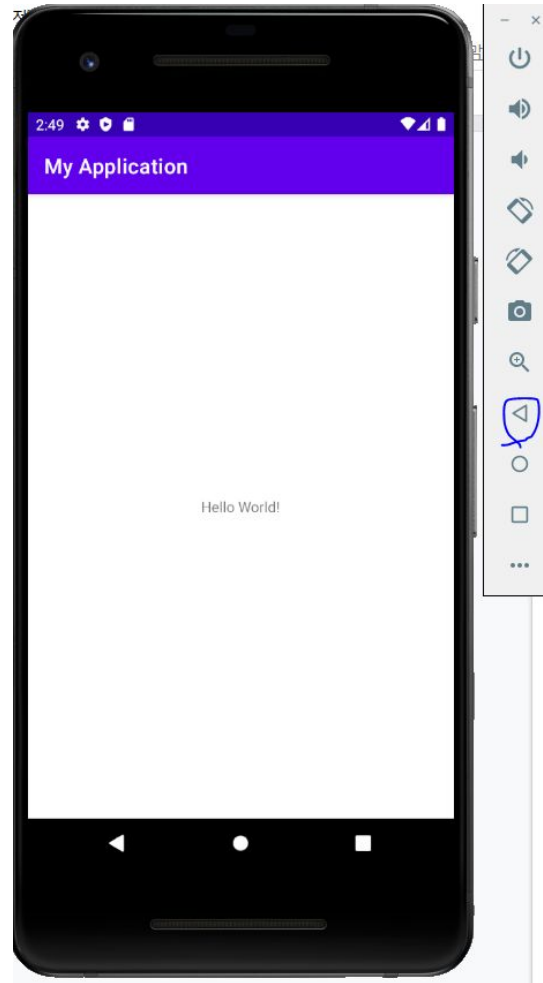
이번엔 자바 코드에 대해서 보겠습니다.
좌측의 java 폴더 하단에 있으며, 익숙한 자바 문법이 보입니다.
기본적으로는 onCreate() 메서드로부터 모든 호출이 시작됩니다.
setContentView(R.layout.activity_main); 에 의해 activity_main.xml의 레이아웃이 실행시 화면으로 표시되는 것입니다.



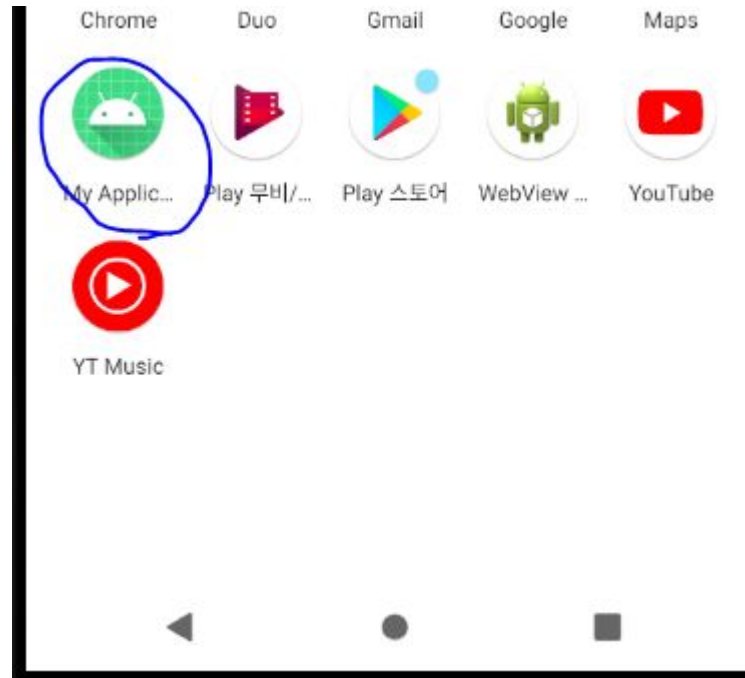
상단 Run메뉴에서 Run_\'app\'를 선택하면 AVD에서 실행됩니다.



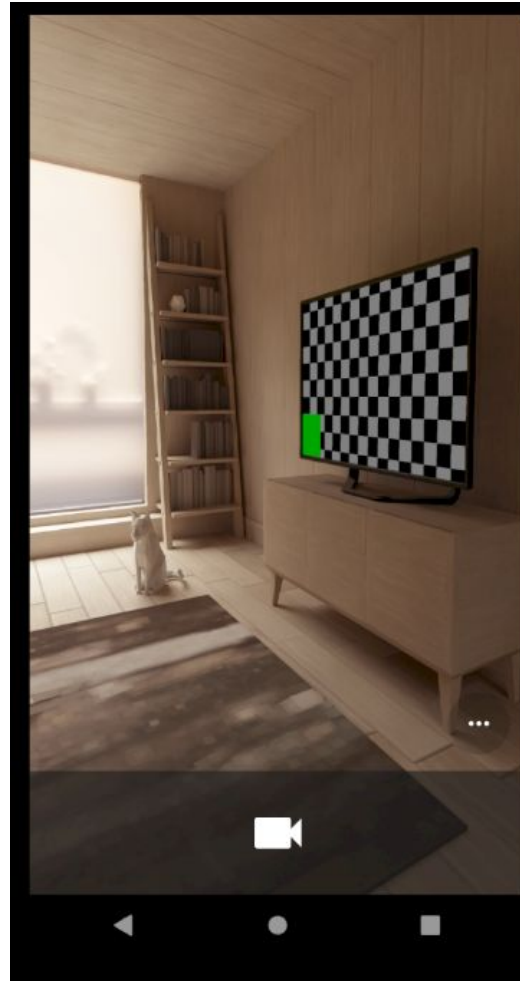
폰 기종을 바꾸고 싶다면 위의 Test(Pixel 2) 가 적힌 탭을 클릭해 다른 폰으로 바꾸면 됩니다.



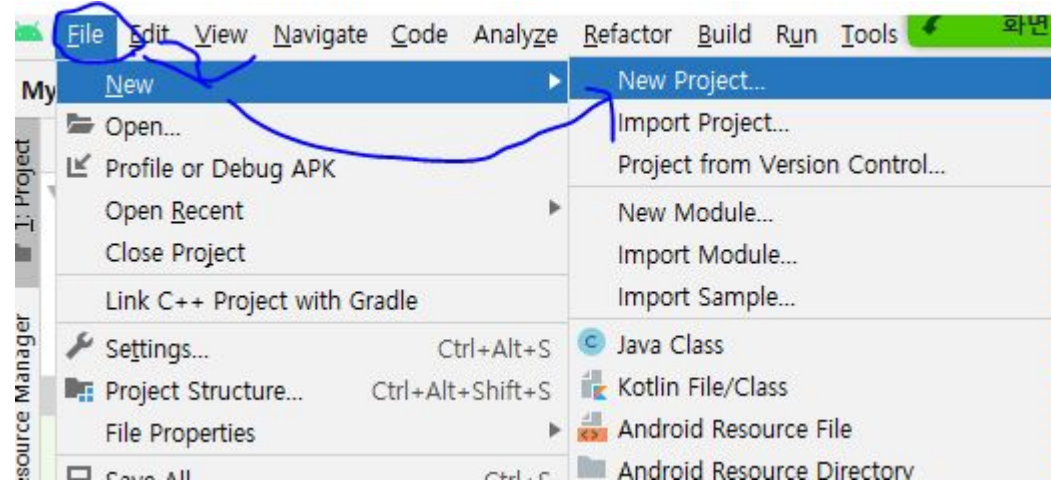
빌드 완료시 위와 같이 아까의 레이아웃이 **AVD**에 나오는것을 볼 수 있습니다.
향후 기능과 버튼을 더 배치해 우리가 원하는 프로그램을 만들면 됩니다.
◁ 버튼을 누르면 **AVD** 초기 화면으로 돌아갈 수있습니다.



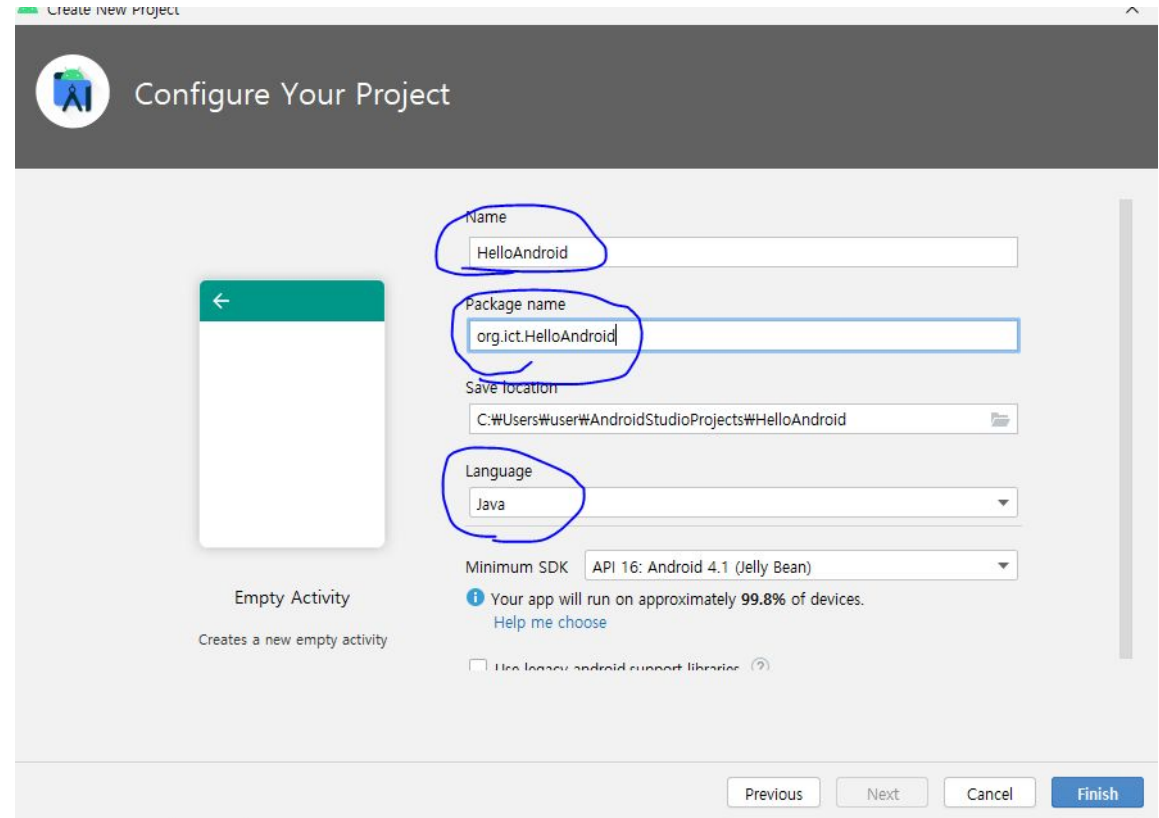
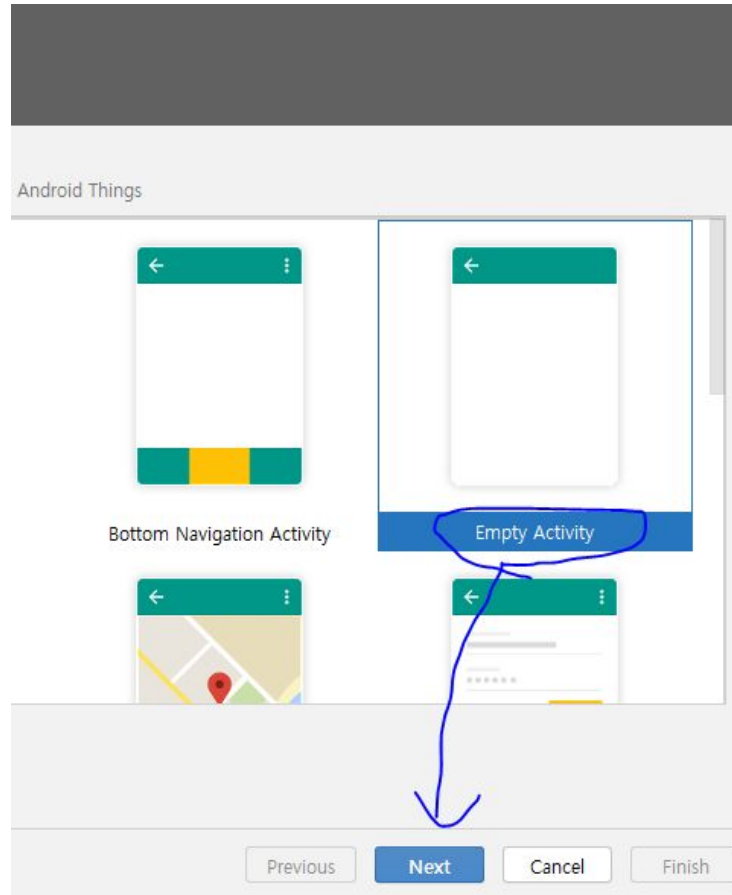
초기 아이콘은 저렇게 안드로이드 머리가 반 노출된 아이콘이며
추후 96x96 픽셀 정도 크기의 png 아이콘으로 교체 가능합니다.



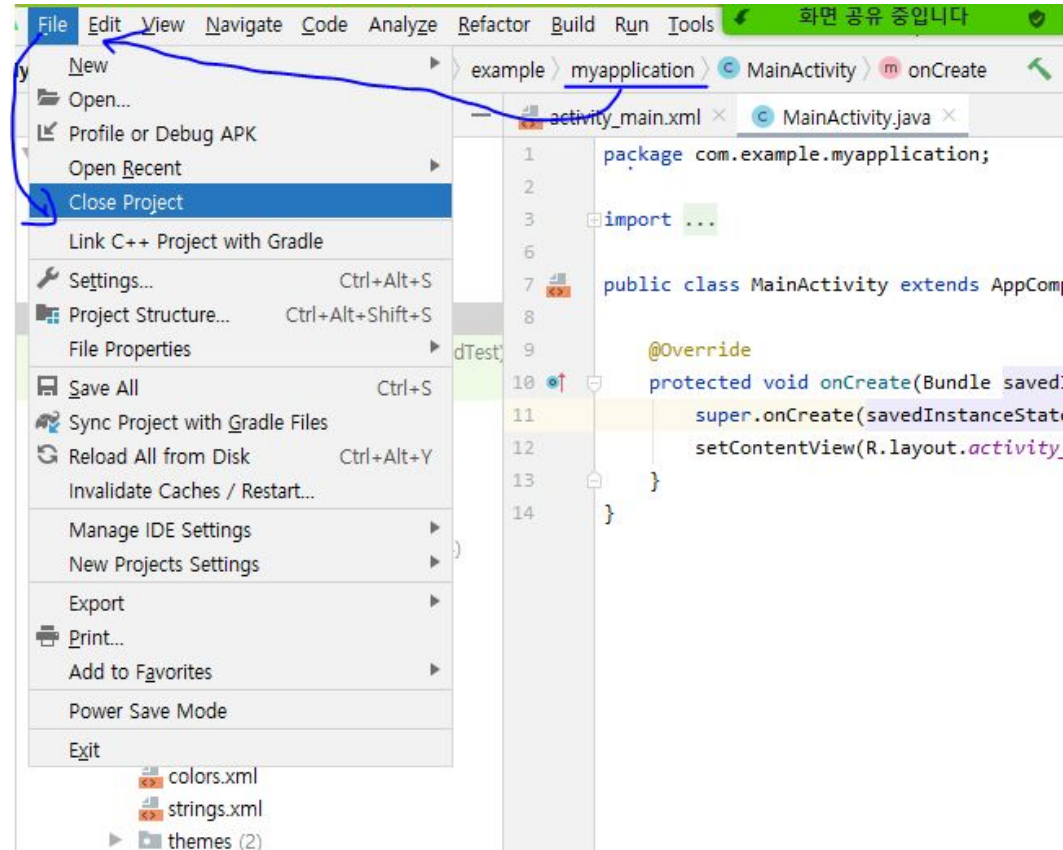
카메라 어플등을 켜면 가상의 화면이 나와 사진촬영이나 동영상 촬영도 가능합니다.



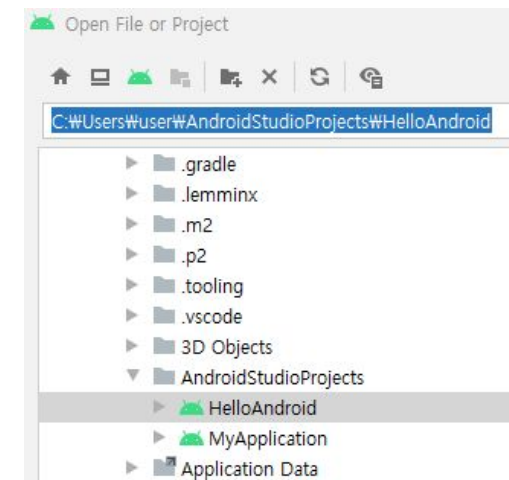
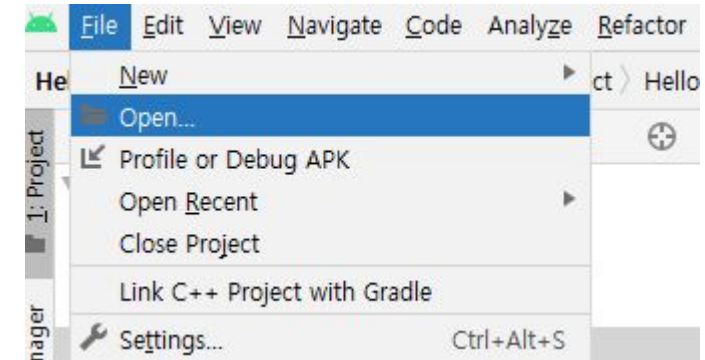
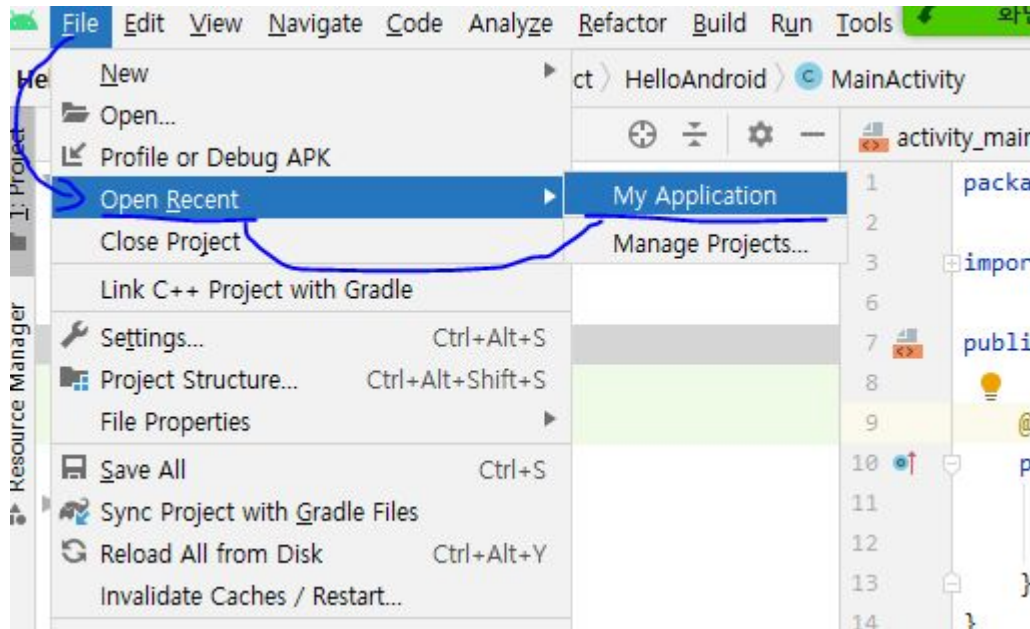
이제 프로젝트를 체크하기 위해 상단 **File -> New -> New Project**를 선택해서 새로운 프로젝트를 만듭니다.



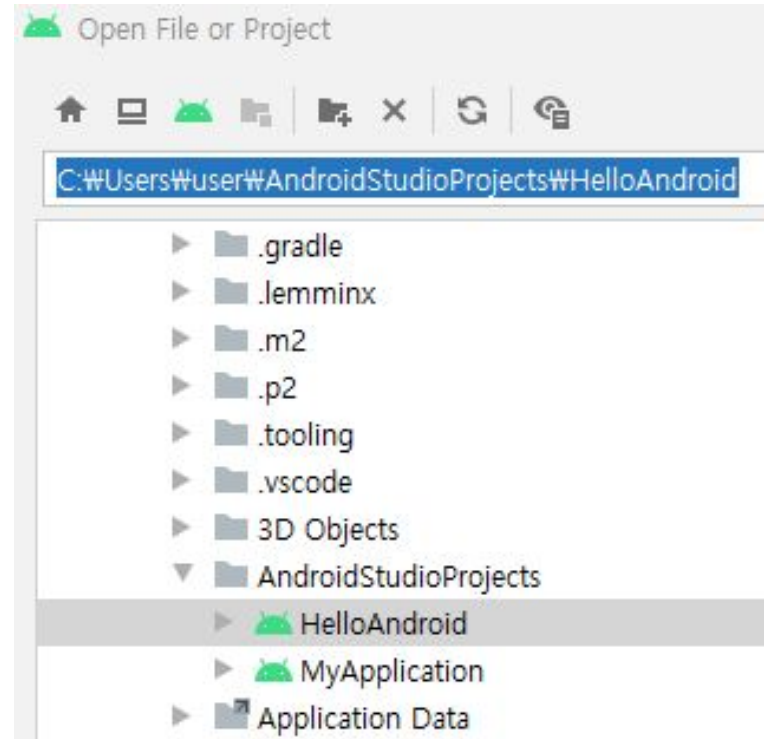
Empty Activity를 골라주시고 프로젝트 이름은 **HelloAndroid**,
패키지는 **org.ict.HelloAndroid**로 스프링과 같이 서비스 서버의 주소를
뒤집어서 표기합니다. 언어는 **java**를 골라주세요.



새 창이 열리는데 기본적으로 안드로이드 스튜디오는 여러 프로젝트를 동시에 띄우고 작업할 수 있습니다. 이는 장점이 되기도 하지만 작업중이지 않은 프로젝트가 리소스를 잡아먹으므로 **File -> Close Project**로 첫 프로젝트인 **myapplication**을 닫아줍니다.



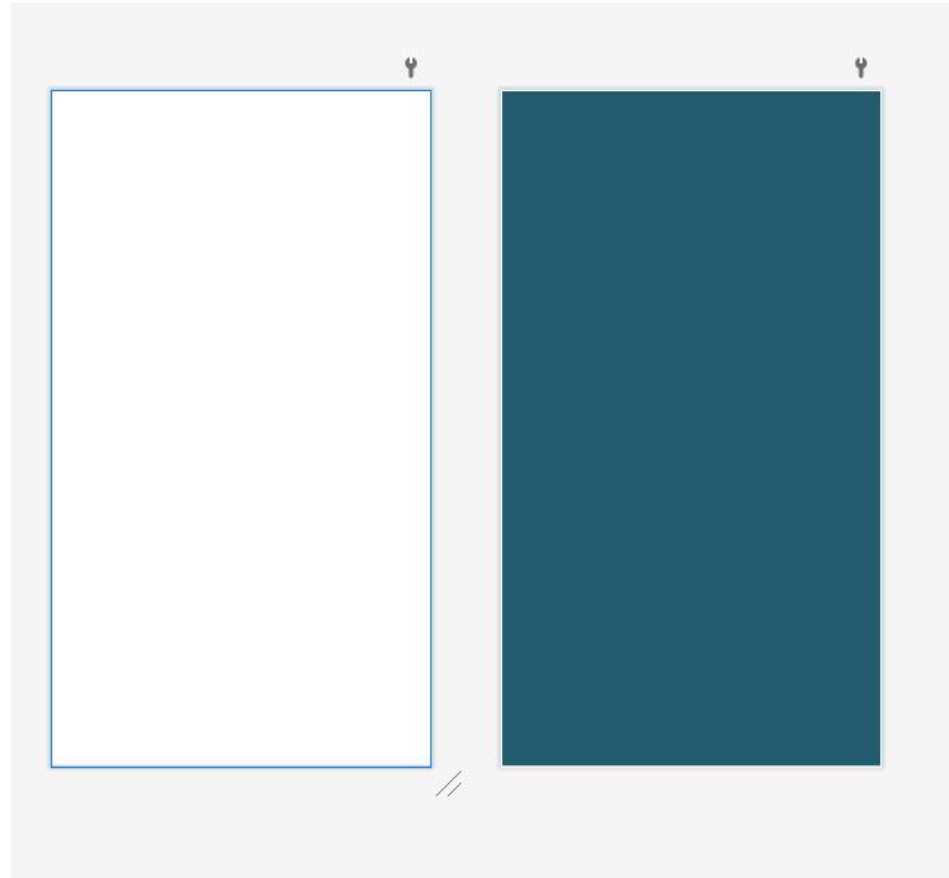
다시 다른 프로젝트 오픈시는 **Open Recent**에서 고르거나
혹은 **Open**을 눌러 나오는 창에서 선택합니다.



안드로이드 프로젝트는 그냥 위의 **Open**의 경로로 가서 폴더째로 복사하거나 압축해서 전달해 열면 그대로 다른 PC에서 쓸 수 있습니다.



먼저 프로젝트가 열리면 **activity_main.xml**로 이동해서 **code**탭으로 간 다음, **textView** 태그를 삭제해버립니다.



그러면 Design탭에 위와 같이 Textview 부품이 삭제된것을 볼 수 있습니다.

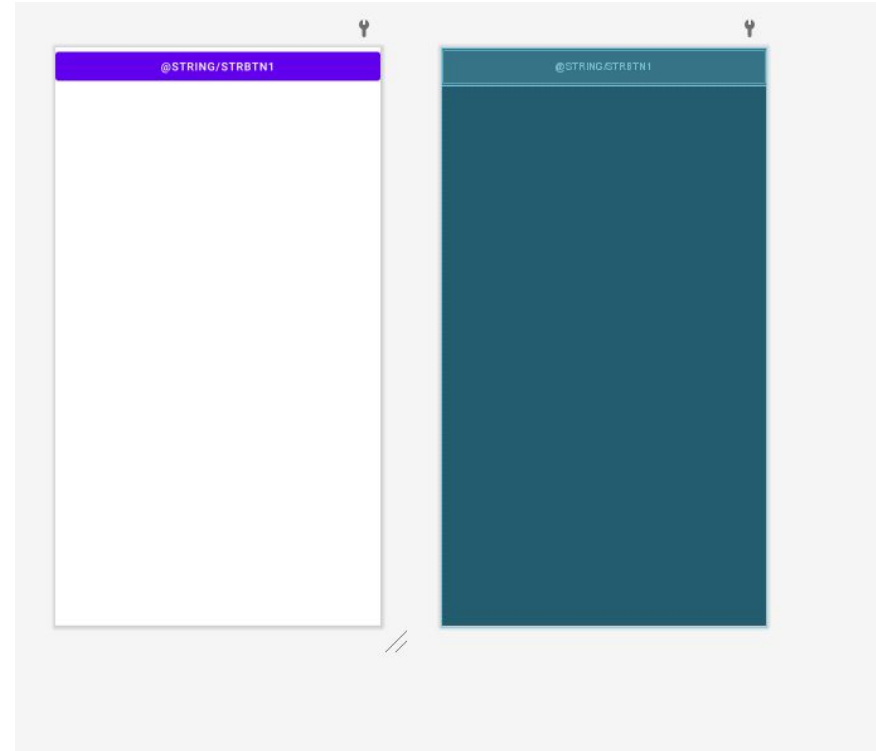


이제 사이에 `<Button>` 부품을 만들고, 내부 설정에 `android`를 적어넣으면 버튼의 크기나 속성값을 지정할 수 있습니다.
위와 같이 필수적으로 요구하는 속성이 있으니 클릭해 추가합니다.

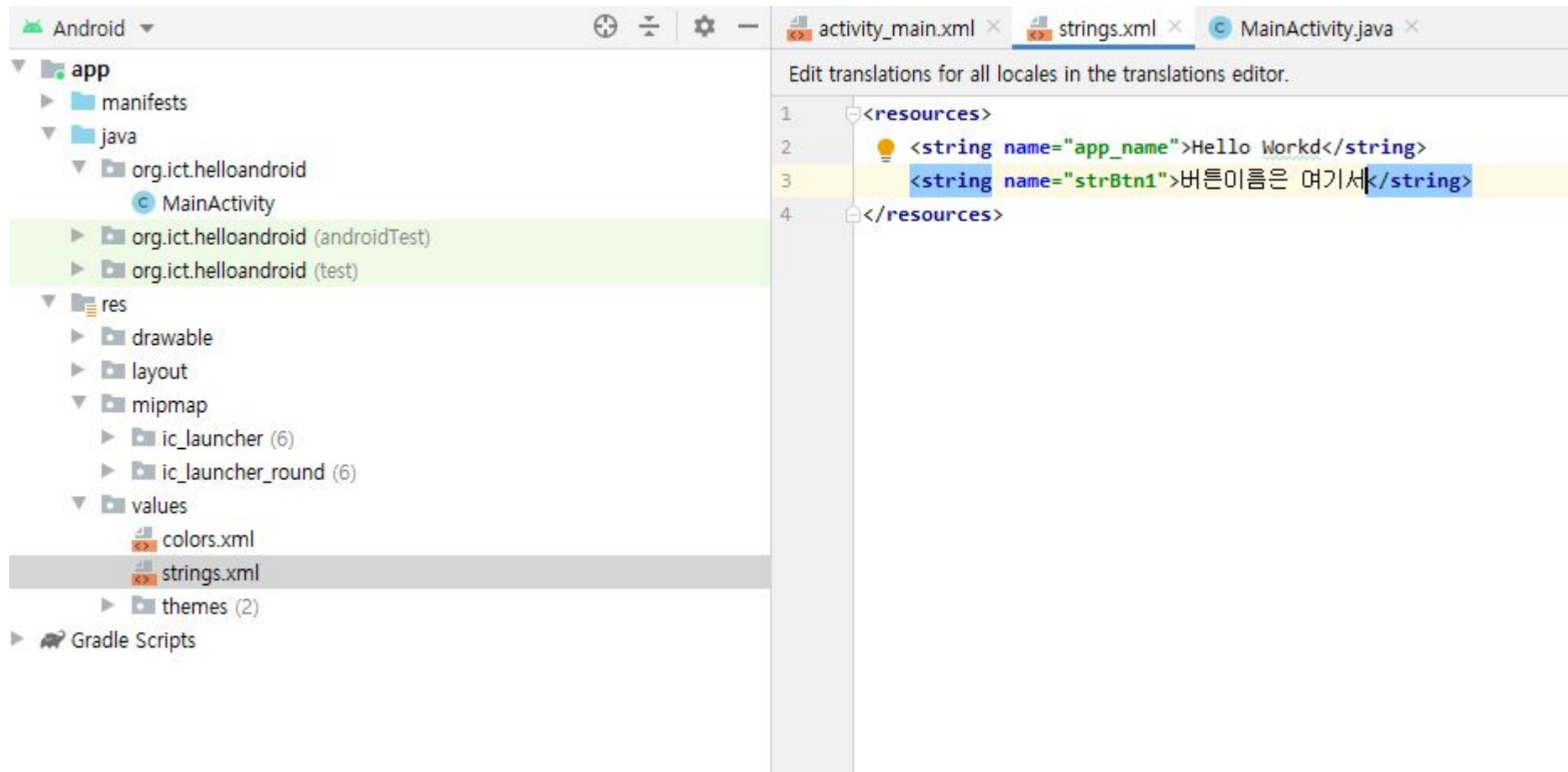
```
<Button  
    android:layout_height="wrap_content"  
></Button>
```

레이아웃 높이 기입하는 부분에 **ctrl+space**를 누르면 입력할 수 있는 옵션들이 자동으로 완성됩니다. **wrap_content**를 넣습니다.

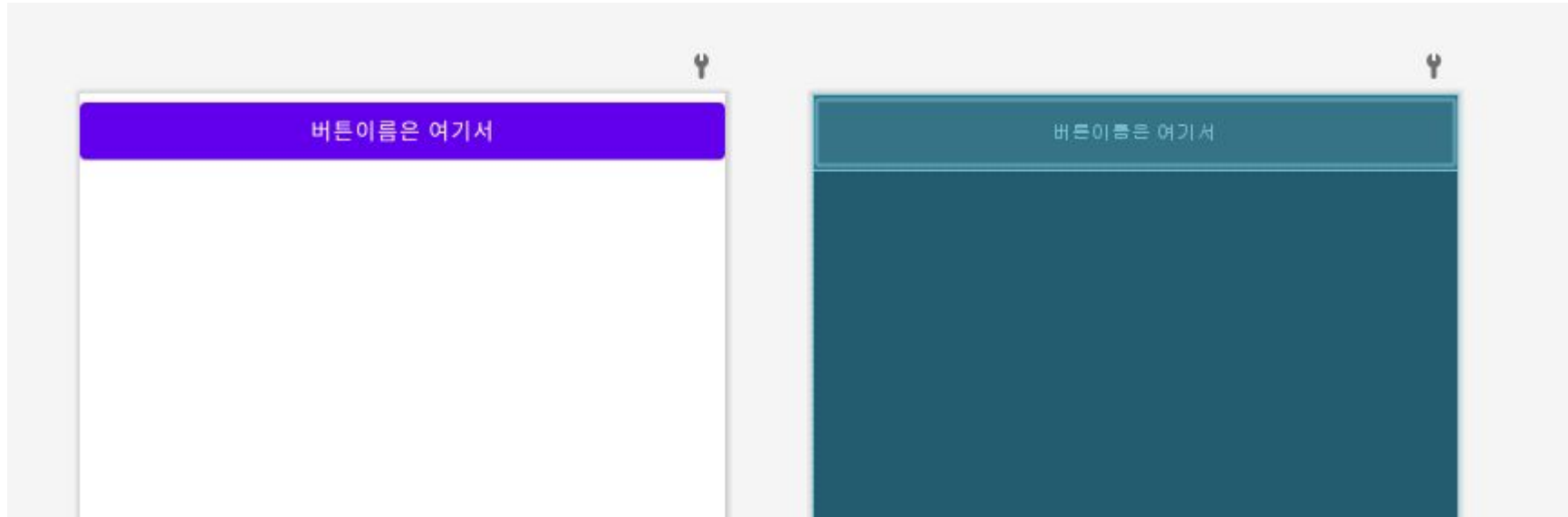
```
<Button  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:id="@+id/button1"  
    android:text="@string/strbtn1"  
></Button>
```



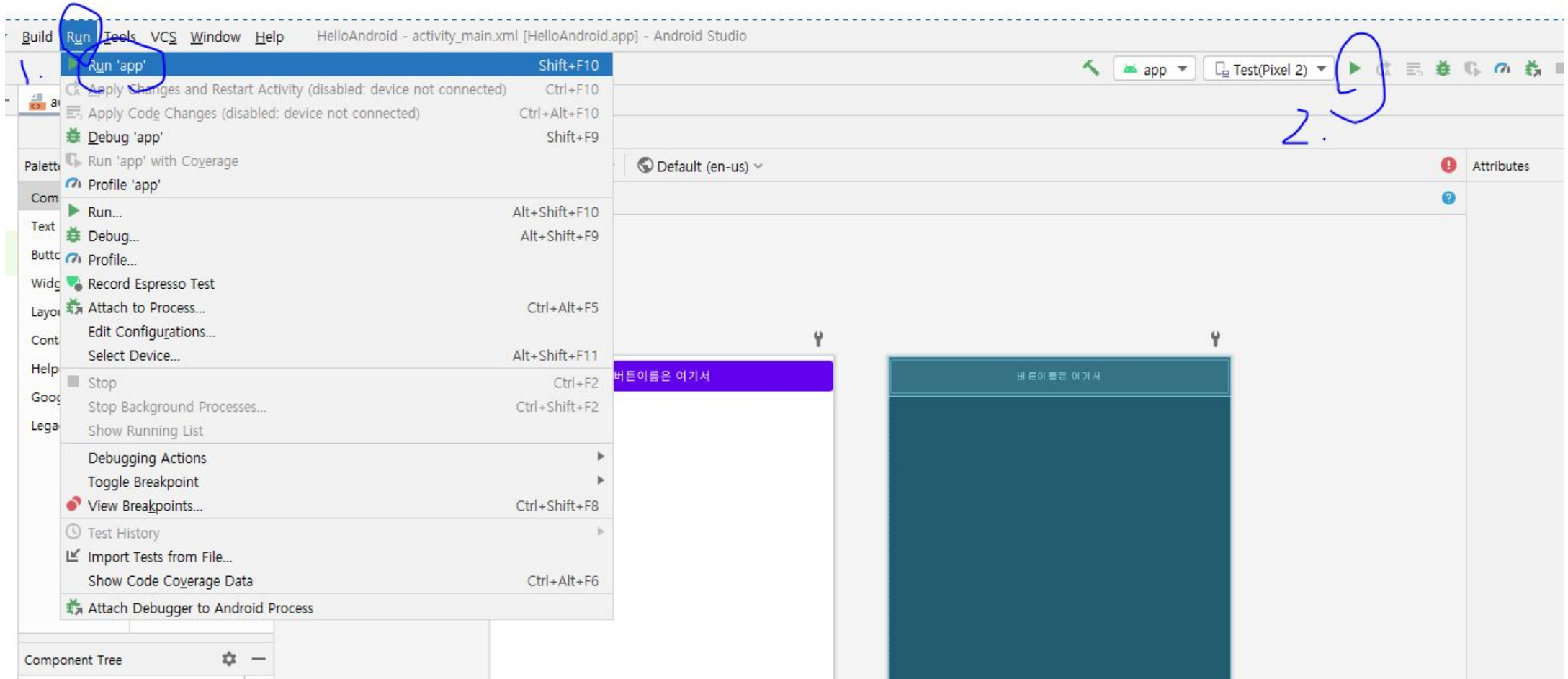
같은 방식으로 위와 같이 만들어주시면 됩니다.
디자인 탭에서는 우측과 같이 보입니다.
단, `text="@string/strbtn1"`은 현재 빨간색으로 경고를 띄우고 있습니다.



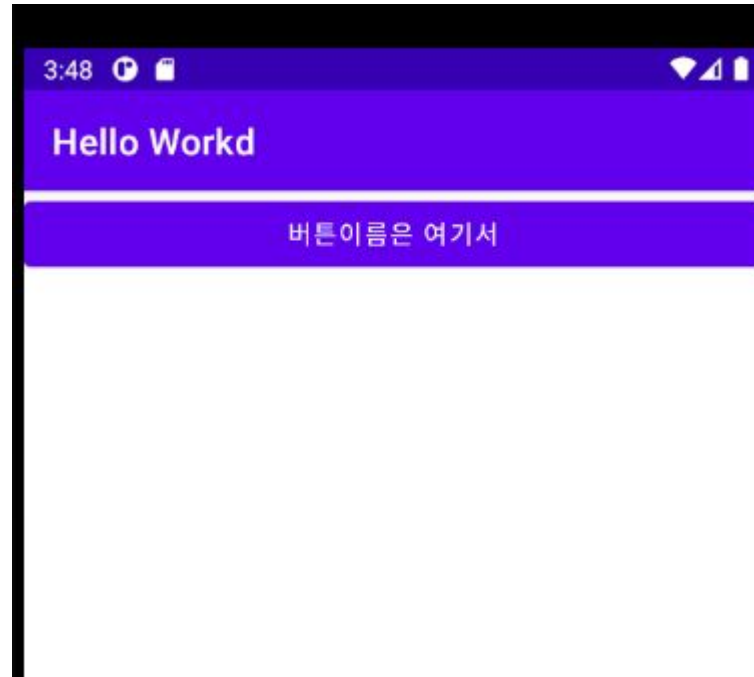
안드로이드에서는 문자열들은 전부 **strings.xml**에 저장하도록 설계되어 있기 때문에 **res/values/strings.xml**에 **strbtn1**에 해당하는 문자열을 작성합니다.



이제 `strBtn1`에 저장해둔 이름이 제대로 표시되는것을 볼 수 있습니다.

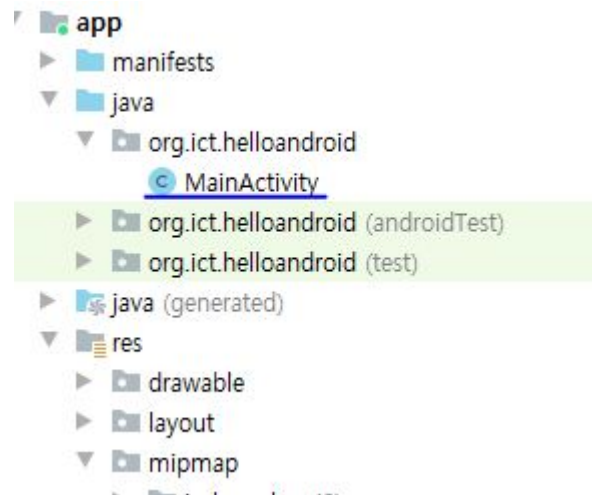


메뉴바의 run- run'app'을 실행하거나, 혹은 상단 재생버튼을 클릭해 어플을 실행해보도록 하겠습니다.



위와 같이 화면이 나옵니다.

물론, 버튼은 클릭하면 클릭이벤트를 연결해두지 않았기 때문에 별다른 일은 일어나지 않습니다만, 클릭 자체에는 반응을 합니다.



이제 이 버튼에 마치 자바스크립트의 **onclick**처럼 이벤트를 연결해보겠습니다.

java/org.ict.helloandroid/MainActivity를 선택합니다.

```
package org.ict.helloandroid;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button button1;

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

다음으로, 이제 이벤트를 처리하기 위해 **Button** 객체를 선언합니다.

```
public class MainActivity extends AppCompatActivity {  
  
    Button button1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        button1 = (Button) findViewById(R.id.button1);  
    }  
}
```

```
activity_main.xml x strings.xml x MainActivity.java x  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="h  
3     xmlns:app="http://schemas.android.com/apk/res-auto"  
4     xmlns:tools="http://schemas.android.com/tools"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     tools:context=".MainActivity">  
8  
9     <Button  
10         android:layout_height="wrap_content"  
11         android:layout_width="match_parent"  
12         android:id="@+id/button1"  
13         android:text="버튼이름은 여기서"  
14     ></Button>  
15  
16 </androidx.constraintlayout.widget.ConstraintLayout>
```

상단에 선언해둔 button1을 onCreate() 내부에서 activity_main.xml의 button1과 연결합니다.
onCreate는 일종의 main메서드처럼 작동하는 메서드입니다.

```
button1 = (Button) findViewById(R.id.button1);

button1.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        Toast.makeText(getApplicationContext(), text: "버튼을 눌렀습니다.", Toast.LENGTH_SHORT).show();
    }
});
```

버튼을 연결했다면, 이제 버튼을 클릭했을때 발생하는 이벤트를 연결해야합니다.

onclick을 안드로이드에서는 위와 같이 표현합니다.

setOnClickListener를 생성한 다음, 내부에 onClickListener를 생성하고 클릭시 실행될 메서드를 그 안에 선언해줍니다.



이제 프로젝트를 실행하면 위와 같이 나옵니다.
Toast는 하단에 “버튼을 눌렀습니다” 라고 출력되는 부품을 의미합니다.