

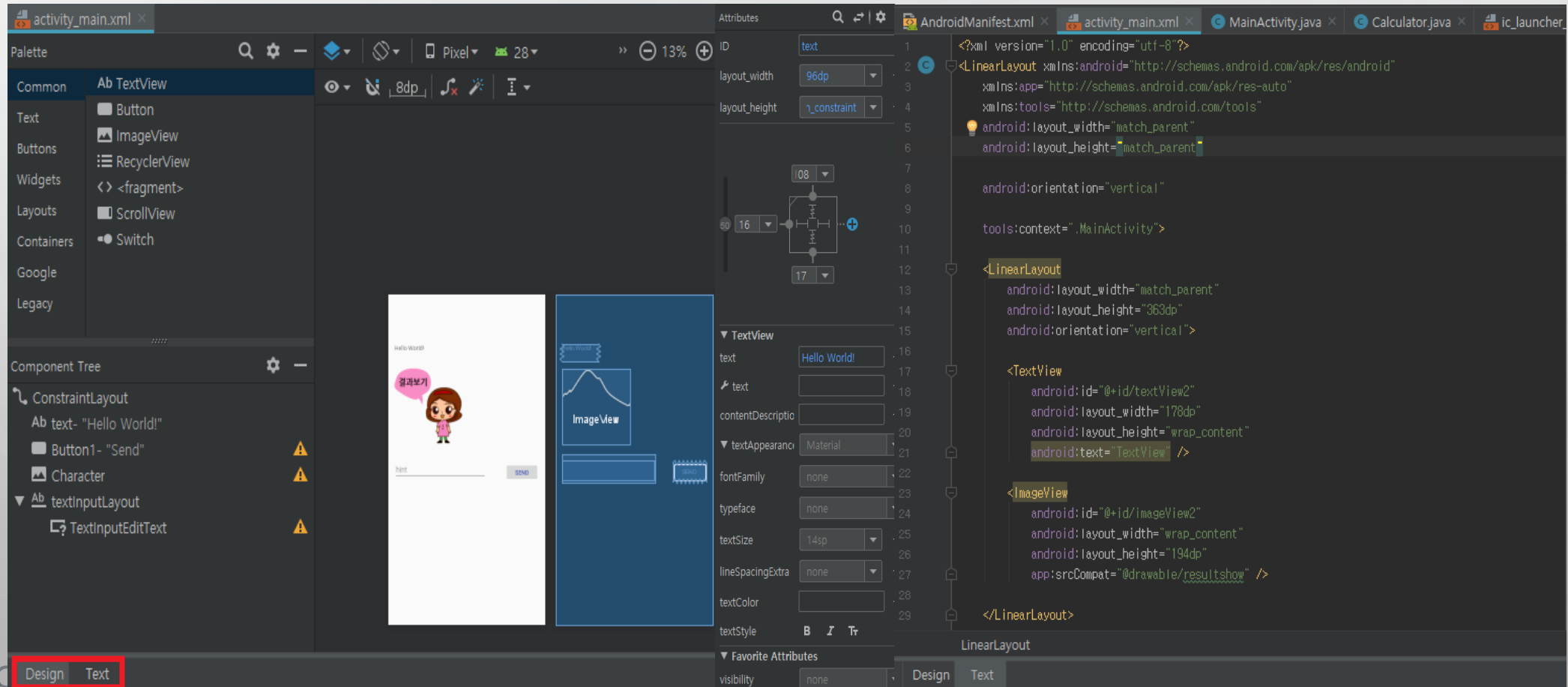


ANDROIDSTUDIO -LAYOUT-

SOULSEEK

LAYOUT

- **View(위젯)**
 - **Text View, Button View, Image View** 등을 말한다
 - **Layout**의 구성요소이다.
 - **Design**과 **XML**코드로 설정 할 수 있다.



LAYOUT

- **LinearLayout**
 - **horizontal** 또는 **Vertical** 중심으로 이어서 배치하는 형태
- **RelativeLayout**
 - 이미 존재하는 뷰와의 관계로 배치하는 형태
- **FrameLayout**
 - 왼쪽 상단을 기준으로 뷰들을 겹쳐서 쌓는 형태
- **TableLayout**
 - 열과 행을 가지는 테이블 배치 형태
- **ConstraintLayout**
 - **UI**설계를 손쉽게 설계할 수 있으며 관계상태를 설정 할 수 있는 형태

LINEARLAYOUT

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
```

```
    tools:context=".MainActivity">
```

```
</LinearLayout>
```

속성	속성값	기능
orientation	vertical	뷰들을 수직으로 배열한다.
	horizontal	뷰들을 수평으로 배열한다.
layout_width layout_height	wrap_content	뷰 내용물의 크기에 맞추고 내용물의 크기에 맞추어 가로세로크기 결정.
	match_content	크기를 꽉 채운다.

LINEARLAYOUT

<LinearLayout

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:orientation="horizontal">
```

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:hint="첫번째 숫자" />
```

<EditText

```
android:id="@+id/e1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginLeft="10dp"  
android:hint="여기에 입력하세요."/>
```

</LinearLayout>

- id를 지정해서 바꿀 때

- layout_margin
 - 모든 방향으로 공백을 준다.
- Layout_marginTop
 - 위쪽 방향으로 공백을 준다.
- Layout_marginLeft
 - 왼쪽 방향으로 공백을 준다.
- Layout_marginRight
 - 오른쪽 방향으로 공백을 준다.

VIEW, ONCLICK, LISTVIEW CODE

```
public class MainActivity extends AppCompatActivity {
```

```
    EditText e1, e2;  
    TextView t1;  
    Button b1;  
    String input1, input2;  
    int sum;  
    static final String[] LIST_MENU = {"LIST1", "LIST2", "LIST3", "LIST4", "LIST5"};
```

해당 view들의 변수를 선언해서
사용 할 수 있는 상태로 만들어
준다.

ListView에 등록할 item배열

```
@Override
```

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

```
    e1 = findViewById(R.id.e1);  
    e2 = findViewById(R.id.e2);  
    t1 = findViewById(R.id.t1);  
    b1 = findViewById(R.id.b1);
```

설정한 id로 검색해서 Layout상의
view들을 연결해 준다.

```
    ArrayAdapter adapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1  
                                            , LIST_MENU);
```

```
    ListView listview = findViewById(R.id.listview1);  
    listview.setAdapter(adapter);
```

아이템과 연결할 adapter를 생성하고 listview를
사용할 수 있도록 layout에서 가져온 후 생성한
adapter를 선언해 준다.

```
}
```

```
}
```

BUTTON ONCLICK

```
b1.setOnClickListener(new View.OnClickListener(){  
    @Override  
    public void onClick(View v){  
        input1 = e1.getText().toString();  
        input2 = e2.getText().toString();  
        sum = Integer.parseInt(input1) + Integer.parseInt(input2);  
        t1.setText(Integer.toString(sum));  
    }  
});
```

무명 클래스로 버튼 클릭을 체크하고
숫자들의 합을 구해서 표시하는 부분

```
{  
    b1.setOnClickListener(ml);  
}
```

```
class MyListener implements View.OnClickListener{  
    public void onClick(View v){  
        TwoSum(e1, e2, t1);  
    }  
  
    public void TwoSum(EditText edi1, EditText edi2, TextView text)  
    {  
        String input1 = edi1.getText().toString();  
        String input2 = edi2.getText().toString();  
  
        int sum = Integer.parseInt(input1) + Integer.parseInt(input2);  
  
        text.setText(Integer.toString(sum));  
    }  
}
```

클래스를 따로 빼서 클래스와 함수를
생성하여 처리