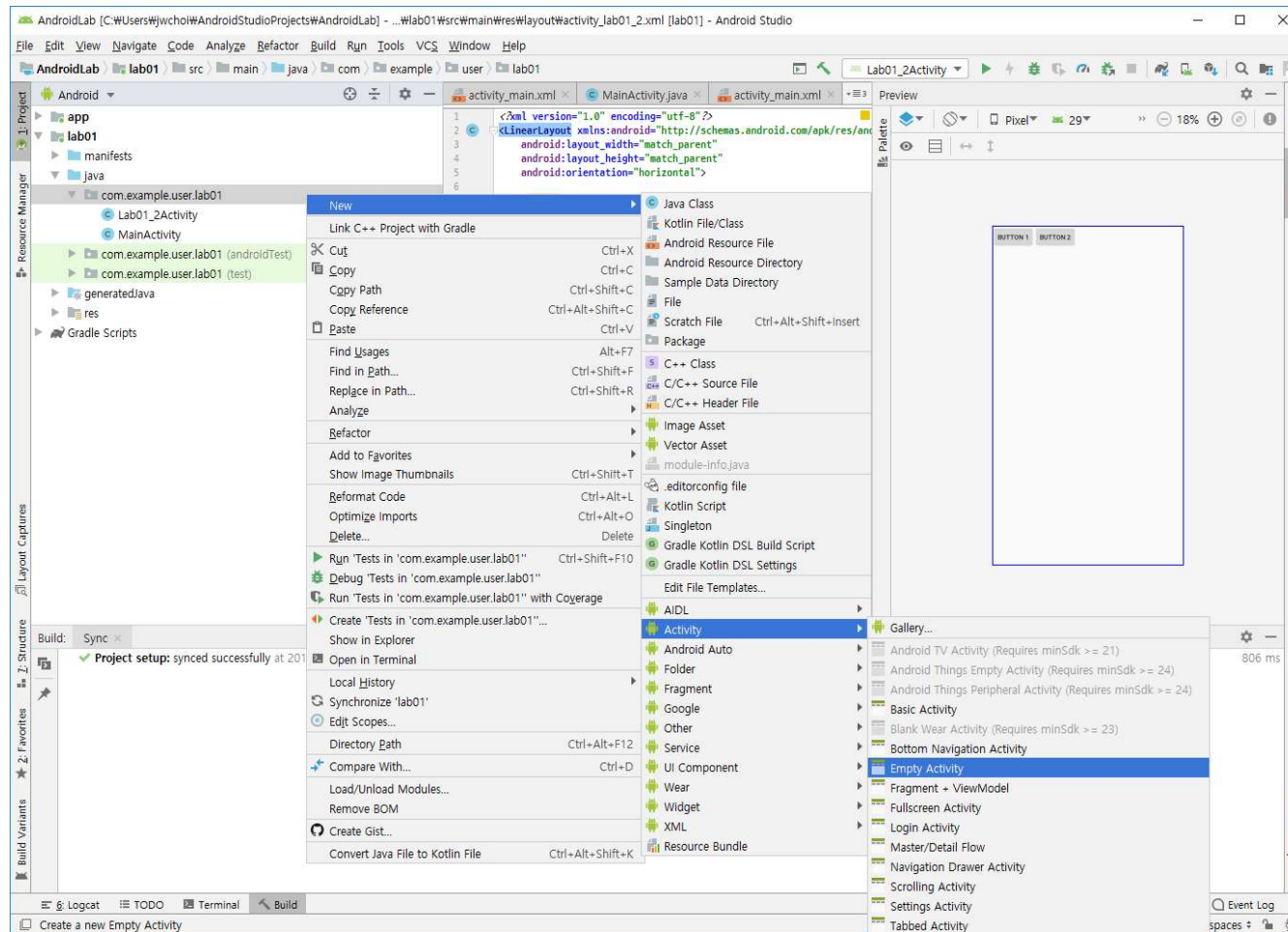




뷰 기초 속성 활용

Step 1 _ 액티비티 추가




New Android Activity

 **Configure Activity**
Android Studio



Creates a new empty activity



Activity Name:

Lab01_3Activity

☒ Generate Layout File

Layout Name:

activity_lab01_3

☒ Launcher Activity

Package name:

com.example.user.lab01

Source Language:

Java

If true, this activity will have a CATEGORY_LAUNCHER intent filter, making it visible in the launcher

Previous

Next

Cancel

Finish

Step 2 _ activity_lab01_3.xml 작성

Default, "Horizontal"

```
<?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">
```

가짜

```
<Button
    android:id="@+id/btn_visible_true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="visible true" />
```

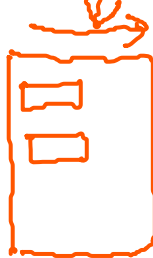
가짜

```
<TextView
    android:id="@+id/text_visible_target"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="hello world"
    android:background="#FF0000"
    android:textColor="#FFFFFF"
    android:padding="16dp"
    android:visibility="invisible" />
```

가짜

```
<Button
    android:id="@+id/btn_visible_false"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="visible false" />
```

```
</LinearLayout>
```



resource + type / name

↑

추가해달라
(R.java에!)

예전에는
자식은
view를 return ~> 형 변환 필요!
T (Generic) 으로 return ~> 형 변환 불 필요!

Step 3 _ Lab01_3Activity.java 작성

```
public class Lab01_3Activity extends AppCompatActivity implements View.OnClickListener {
```

```
    Button trueBtn;  
    TextView targetTextView;  
    Button falseBtn;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_lab01_3);
```

```
    // View 객체 획득
```

```
    trueBtn = (Button) findViewById(R.id.btn_visible_true);
```

```
    targetTextView = (TextView) findViewById(R.id.text_visible_target);
```

```
    falseBtn = (Button) findViewById(R.id.btn_visible_false);
```

```
    // Button 이벤트 등록
```

```
    trueBtn.setOnClickListener(this);
```

```
    falseBtn.setOnClickListener(this);
```

```
}
```

```
// 버튼 이벤트 콜백함수
```

```
@Override
```

```
public void onClick(View view) {
```

```
    if(view == trueBtn) {
```

```
        // trueBtn이 눌리면 targetTextView를 visible 상태로 변경
```

```
        targetTextView.setVisibility(View.VISIBLE);
```

```
    } else if(view == falseBtn) {
```

```
        // falseBtn이 눌리면 targetTextView를 invisible 상태로 변경
```

```
        targetTextView.setVisibility(View.INVISIBLE);
```

```
    }
```

```
}
```

```
}
```

interface를 구현하여 다른 곳에서 정의한
함수를 호출하도록 함.

자신을 제어하겠다. Xml을 읽어
객체를 return (생성됨)

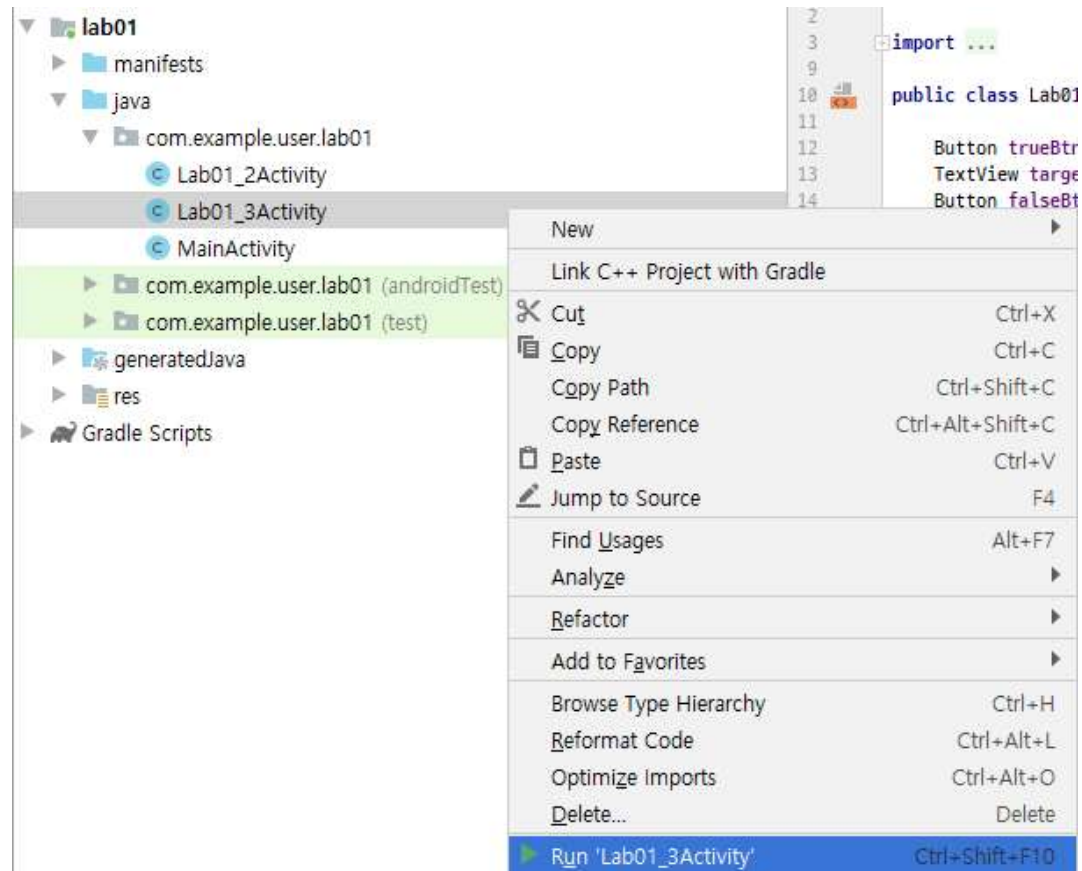
from API
"화면 객체"
view = widget
객체

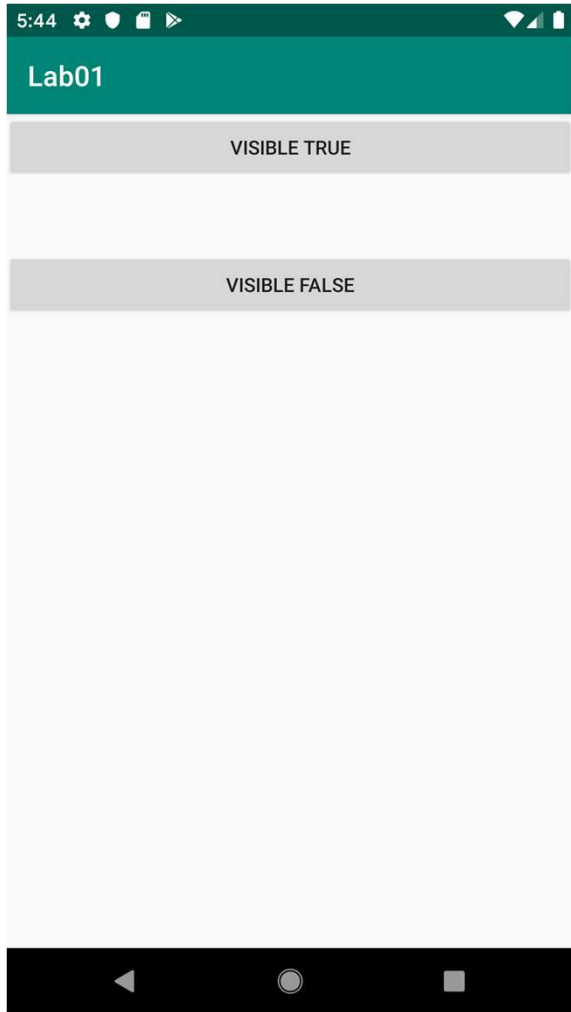
Event Handler -> find View By Id (resId)
(Java에서는 Listener) L> id를 get

"set On name Listener ()"

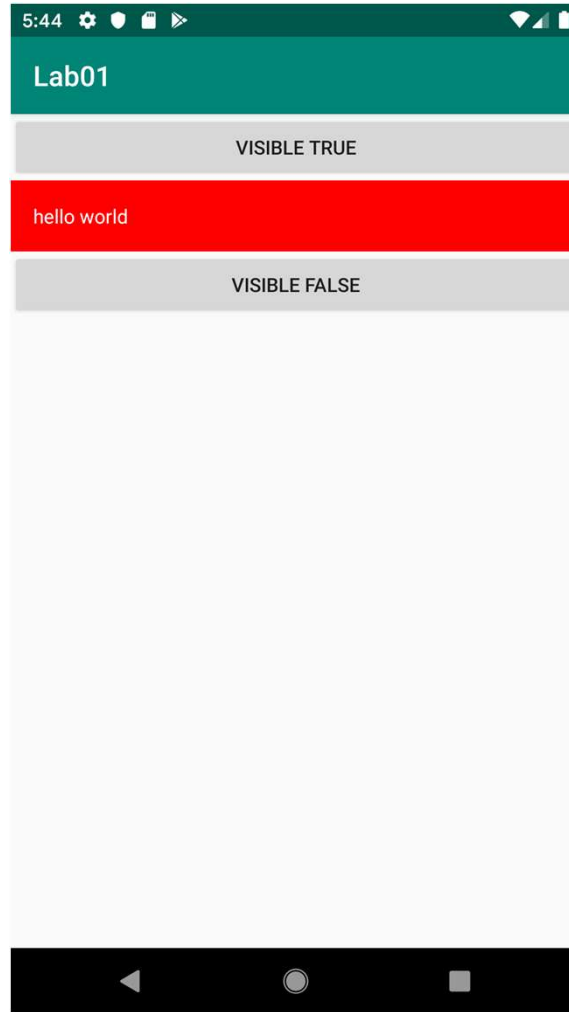
setVisibility (boolean)
L> 가시성

Step 4 _ 앱 실행

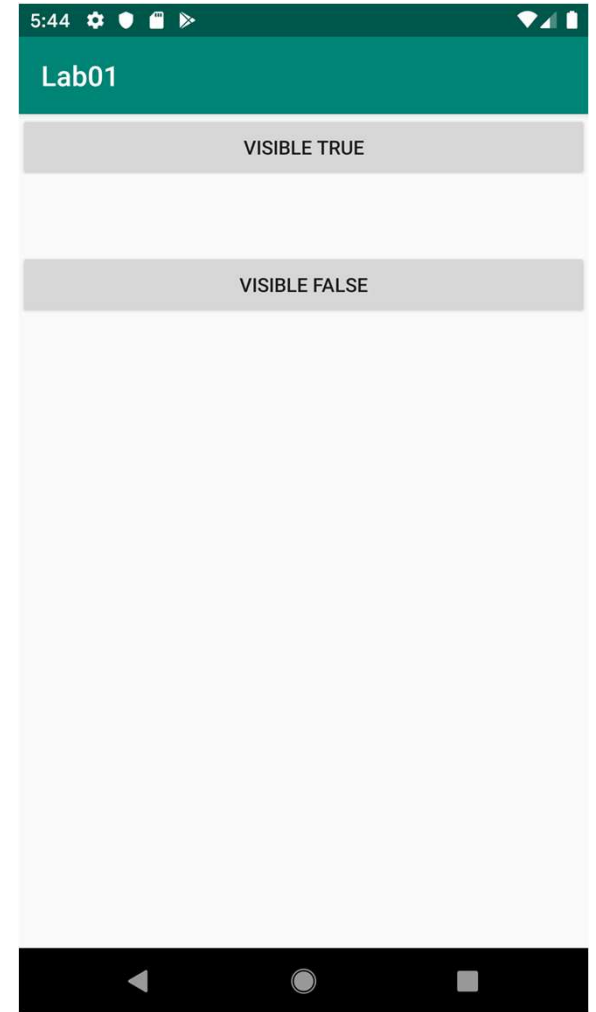




초기 상태



visible



invisible