

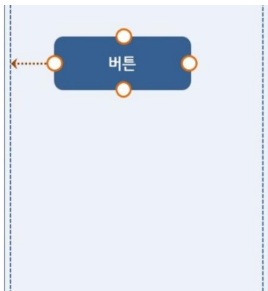
Layout?

Layout?

- Layout은 View 위젯들을 화면에 배치하는 과정에서 위젯의 위치를 **정렬**, **그룹화** 역할
- Layout은 View 위젯들을 포함하는 컨테이너 역할

어떤 작업을 하는지에 따라 용도에 맞는
Layout을 선택해서 사용하는 것이 중요!!

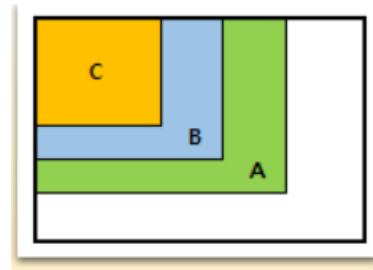
Layout 의 종류



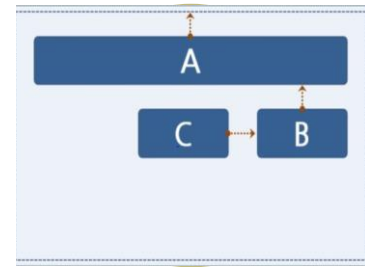
CONSTRAINTLAYOUT
(제약 레이아웃)



LINEARLAYOUT
(리니어 레이아웃)



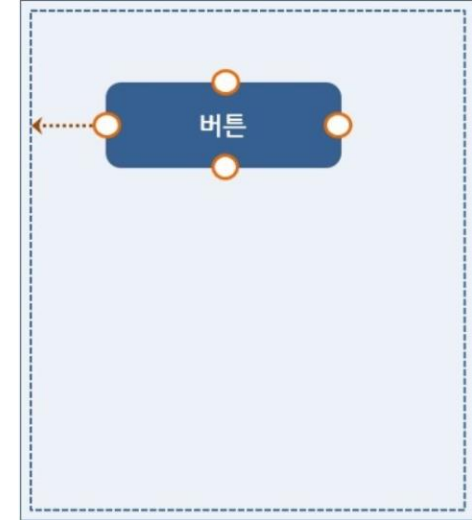
FRAMELAYOUT
(프레임 레이아웃)



RELATIVELAYOUT
(상대 레이아웃)

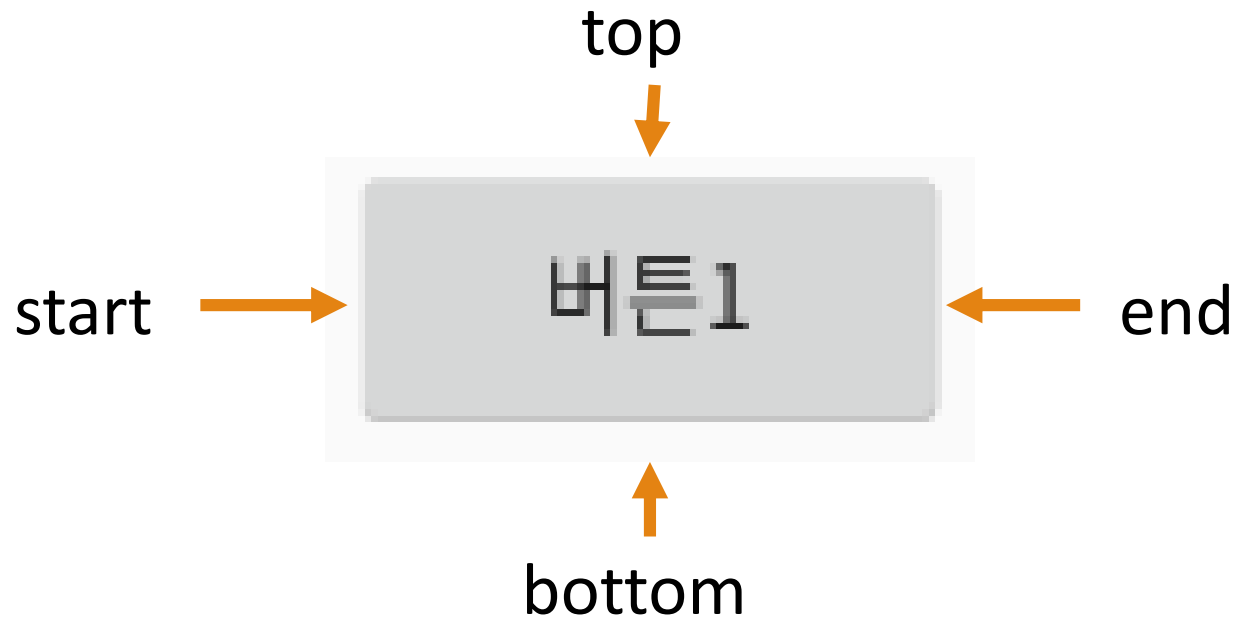
ConstraintLayout

- 프로젝트를 처음 만들었을 때 자동으로 만들어 주는 레이아웃
- RelativeLayout 과 비슷하지만 더 다양하고 강력한 기능 제공
- 자신이 정한 기준에 맞춰서 정렬 가능
 - 최소 두개의 제약을 걸어주야함.

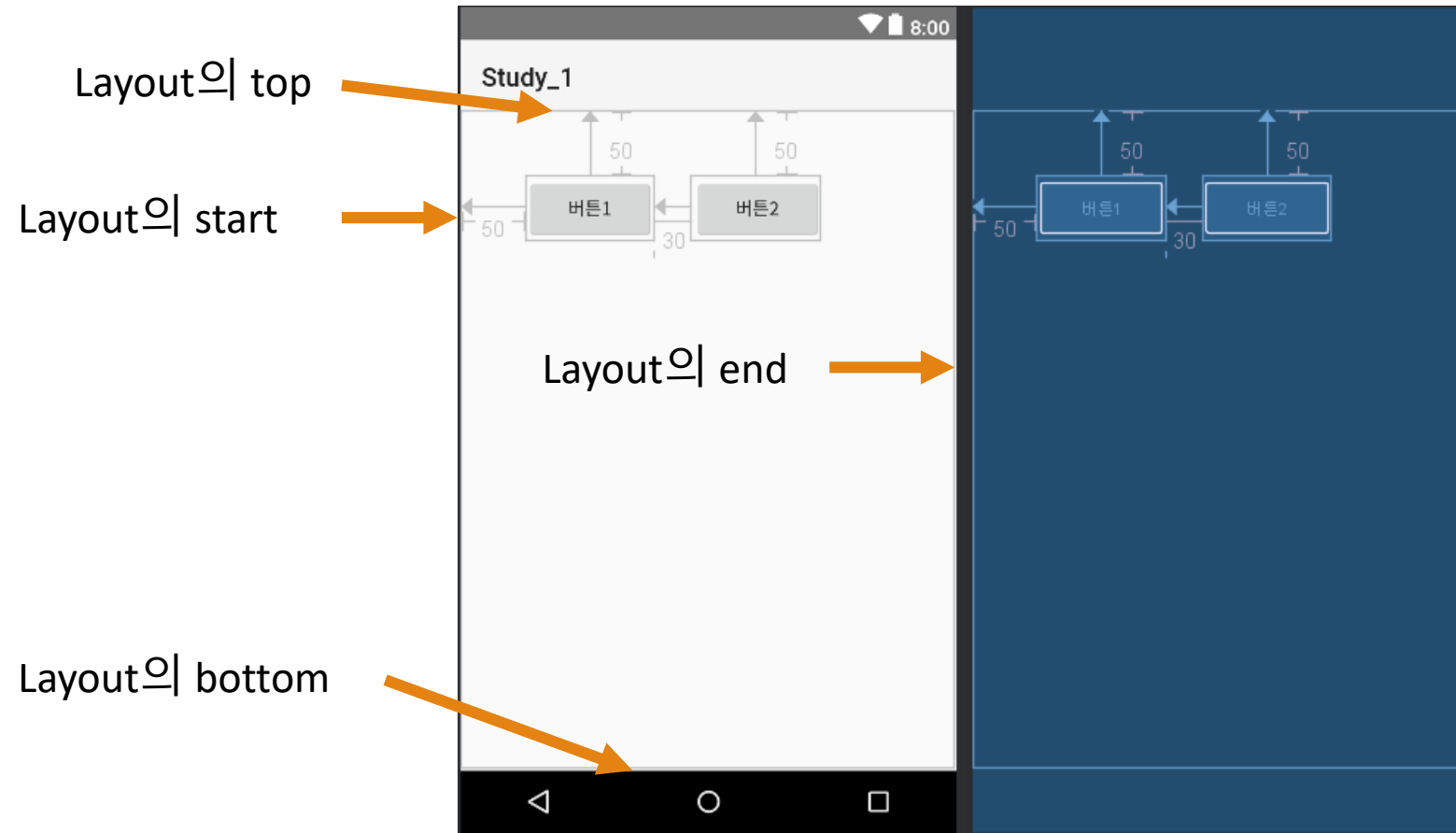


- ConstraintLayout (제약 레이아웃)

ConstraintLayout



ConstraintLayout



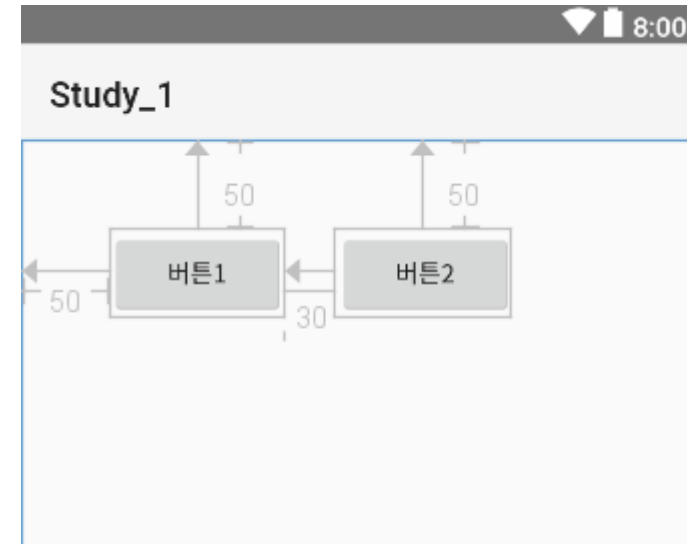
ConstraintLayout

버튼1

```
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
android:layout_marginStart="50dp"  
android:layout_marginTop="50dp"
```

버튼2

```
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintStart_toEndOf="@+id/button1"  
android:layout_marginStart="30dp"  
android:layout_marginTop="50dp"
```

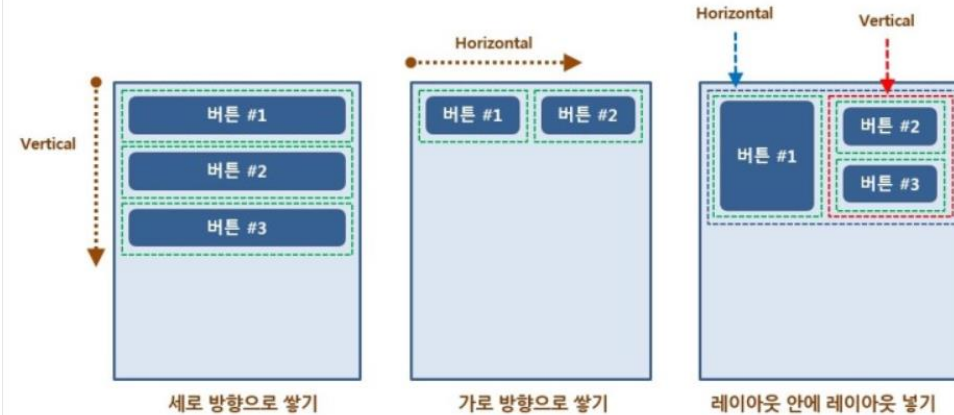


LinearLayout

- Horizontal(수평),
- Vertical(수직) 두가지
 - Orientation 속성을 꼭 넣어야함

```
android:orientation="horizontal"
```

```
android:orientation="vertical"
```



- LinearLayout (리니어 레이아웃)

RelativeLayout

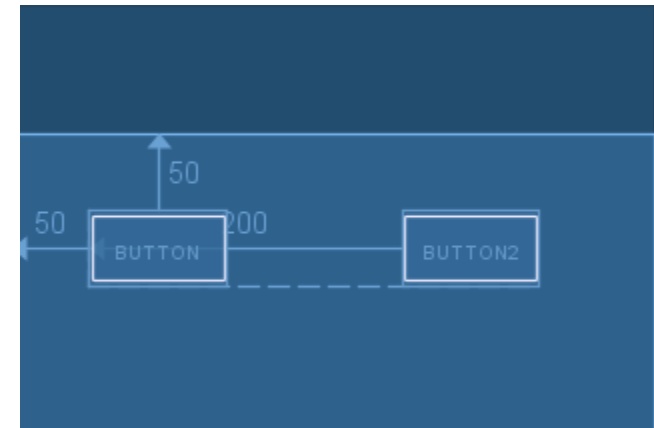
- 자신의 위치를 상대적으로 정함
 - 2가지 방식

Parent에 맞추는 것(button)

```
android:layout_alignParentTop="true"
android:layout_alignParentStart="true"
android:layout_marginStart="50dp"
android:layout_marginTop="50dp"
```

다른 뷰에 맞추는 것(button2)

```
android:layout_alignBottom="@+id/button"
android:layout_alignStart="@+id/button"
android:layout_marginStart="200dp"
```



Drawable

- Drawable은 뷰에 설정할 수 있는 객체

- 비트맵 드로어블(BitmapDrawable)

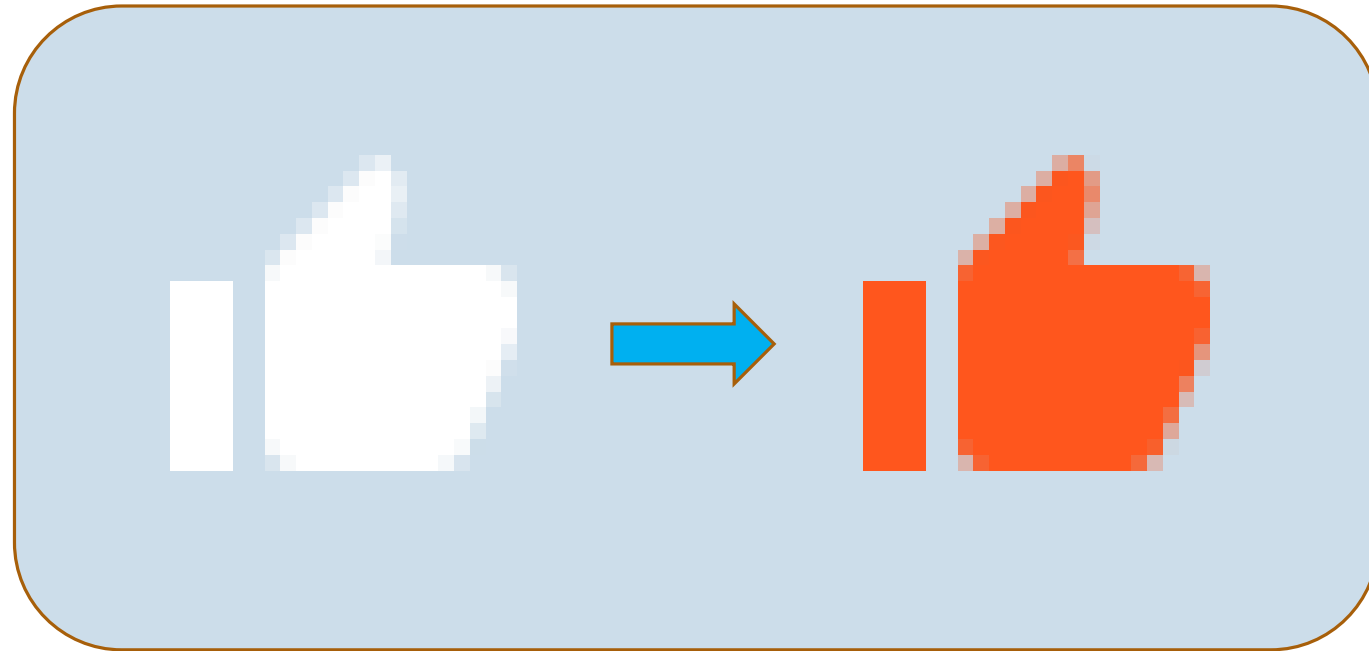
- 상태 드로어블(StateListDrawable)

- 전환 드로어블(TransitionDrawable)

- 셰이프 드로어블(ShapeDrawable)

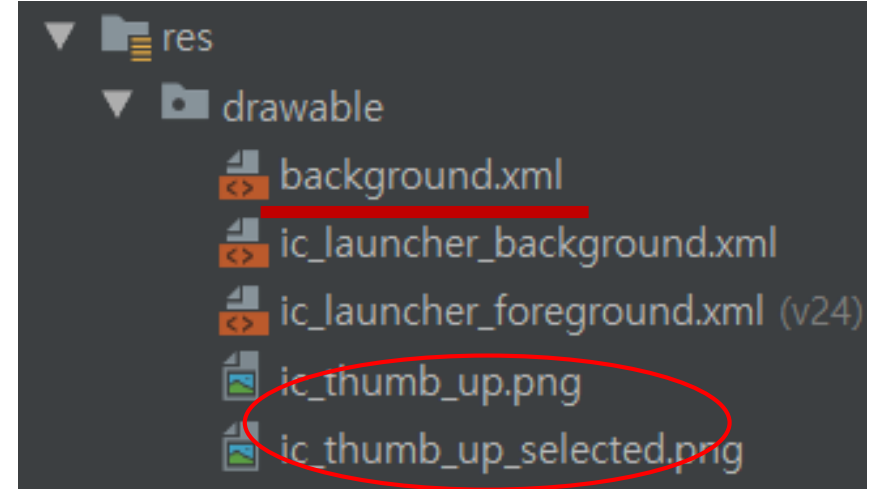
등등....

상태 드로어블(StateListDrawable)

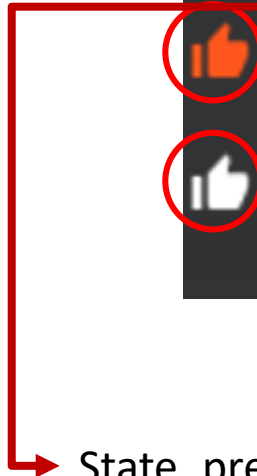


상태 드로어블(StateListDrawable)

- Res/drawable 에 '*****.xml'파일을 만듦.
 - Drawable에서 우클릭 new 클릭!
- 필요한 이미지 파일도!
 - 그냥 이미지 파일 끌어다 놓으면 됨!



상태 드로어블(StateListDrawable)



```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true"
          android:drawable="@drawable/ic_thumb_up_selected"/>
    <item android:drawable="@drawable/ic_thumb_up"/>
</selector>
```

State_pressed : 눌렀을때의 속성

상태 드로어블(StateListDrawable)

<Button

android:id="@+id/button4"

android:background="@drawable/background"

android:layout_width="50dp"

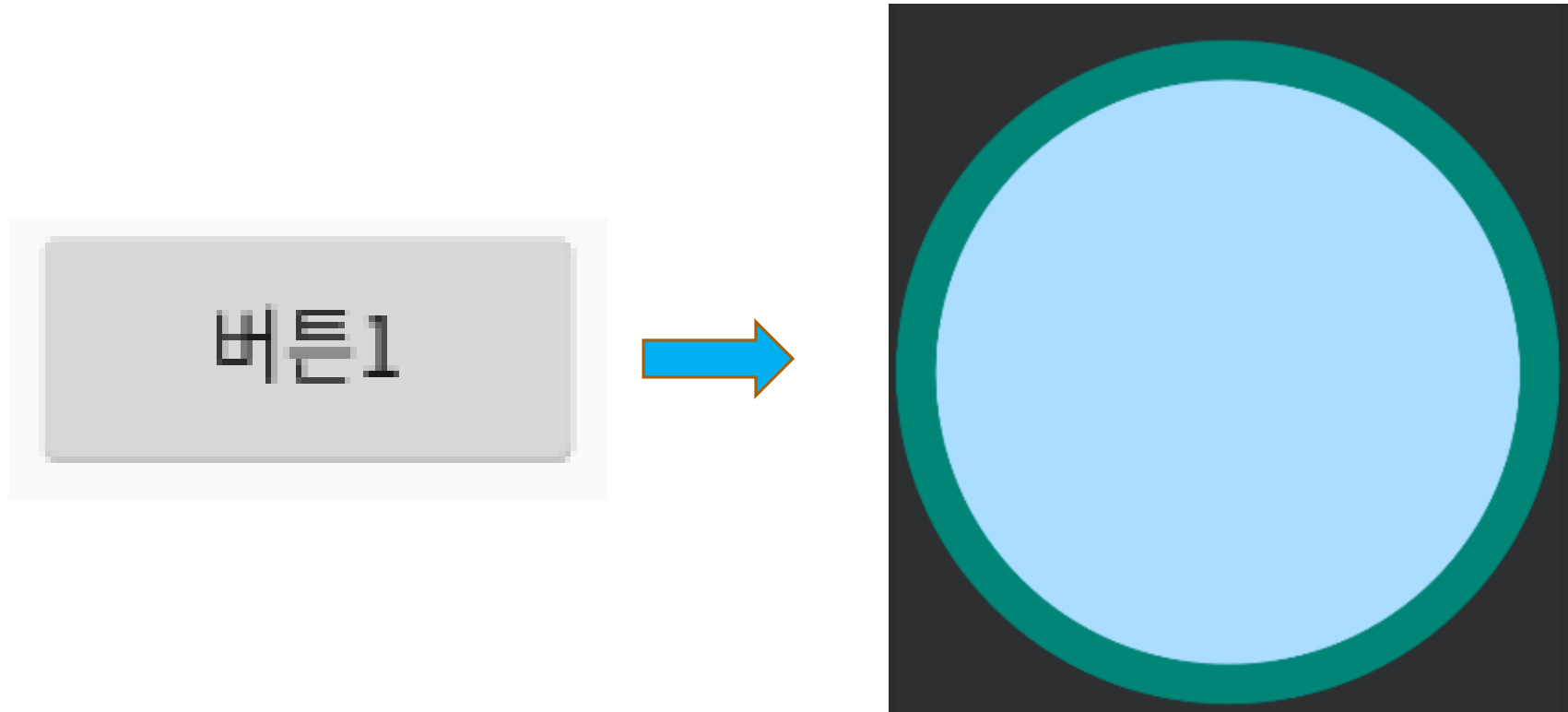
android:layout_height="50dp"

android:layout_margin="40dp"

/>

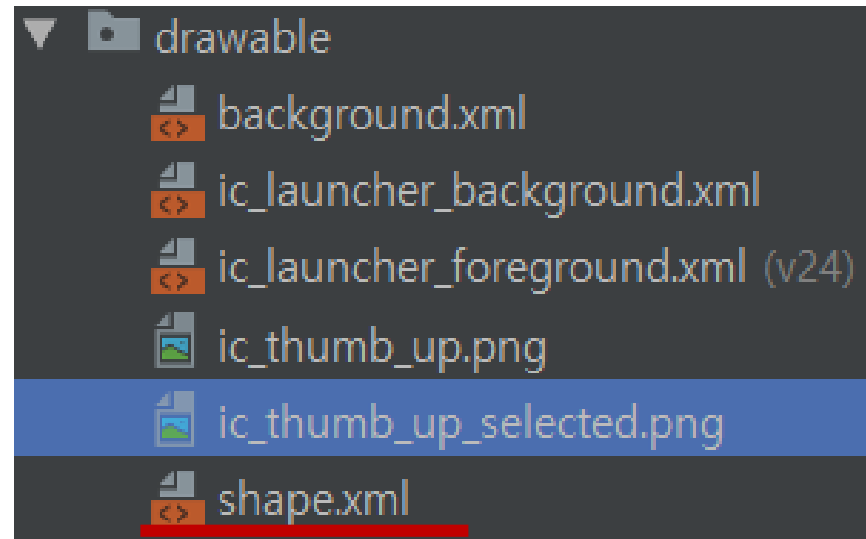
Background 속성에 아까 만든 drawable 넣기!

쉐이프 드로어블(ShapeDrawable)



쉐이프 드로어블(ShapeDrawable)

- Res/drawable 에 '*****.xml'파일을 만듦.
 - Drawable에서 우클릭 new 클릭!
 - 상태 드로어블이랑 같음!



쉐이프 드로어블(ShapeDrawable)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval">
    <size android:width="50dp" android:height="50dp"/>
    <stroke android:width="3dp" android:color="@color/colorPrimary"/>
    <solid android:color="#aaddff"/>
</shape>
```

Selector -> shape 바꾸기

- Size = 사이즈
- Stroke = 테두리 속성
- Solid = 안에 속성
- 등등... 필요할때 검색해서 사용 !

쉐이프 드로어블(ShapeDrawable)

```
<Button
```

```
    android:id="@+id/button5"  
    android:background="@drawable/shape"  
    android:layout_width="30dp"  
    android:layout_height="30dp"  
    android:layout_margin="40dp"  
/>
```

```
<size android:width="50dp" android:height="50dp"/>
```

#주의 shape.xml 에서 size속성을 50씩 주었지만,
activity_main.xml 에서 준 width, height로 적용됨!