

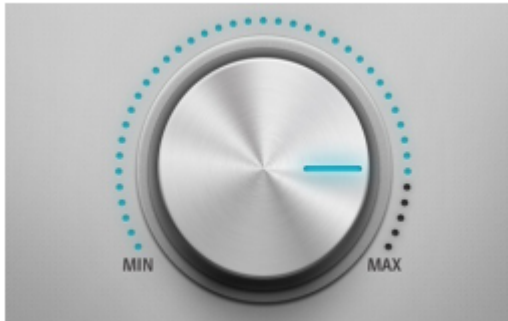


## CHAP 6. 이벤트 처리(2)

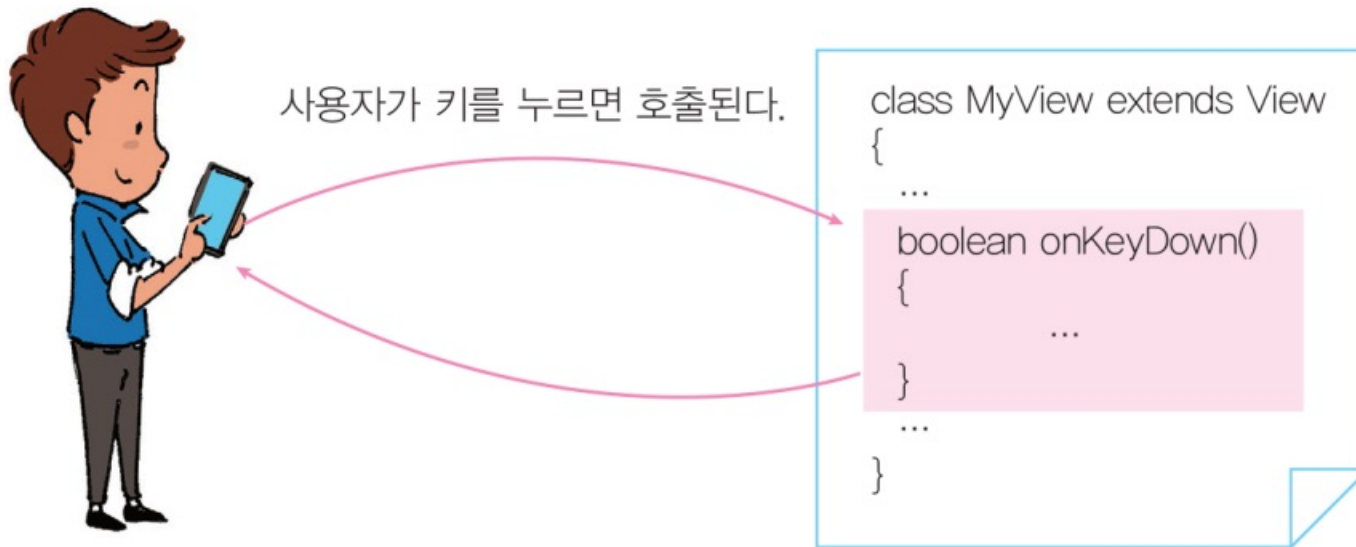
1

# 커스텀 컴포넌트

- 개발자가 직접 **View** 클래스를 상속받아서 필요한 위젯을 개발



# 이벤트 처리 메소드 재정의



# 재정의할 수 있는 콜백 메소드

- onKeyDown(int, KeyEvent)
- onKeyUp(int, KeyEvent)
- onTrackballEvent(MotionEvent)
- onTouchEvent(MotionEvent)
- onFocusChanged(boolean, int, Rect)

# 예제



우선 View 클래스를 상속받아 MyView 클래스를 정의한다. 여기에서 두 개의 메소드를 재정의한다. 하나는 터치 이벤트가 발생하면 이벤트를 처리하기 위하여 onTouchEvent() 메소드를 재정의한다. 또 하나는 화면에 글자를 그리기 위하여 onDraw() 메소드를 재정의한다. onDraw() 메소드는 화면을 다시 그릴 필요가 있을 때 안드로이드에 의하여 호출된다. 여기에다가 그림을 그리는 코드를 추가하면 화면에 그림이 그려진다.

## MyViewEventActivity.java

```
package kr.co.company.myviewevent;
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

class MyView extends View {
    int key;
    String str;
    int x, y;

    public MyView(Context context) {
        super(context);
        setBackgroundColor(Color.YELLOW);
    }
}
```

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    x = (int) event.getX();
    y = (int) event.getY();
    invalidate();
    return super.onTouchEvent(event);
}

```

이벤트 처리기의 매개 변수인 event를 통하여 많은 정보들이 전달된다. 이 중에서 자신에게 필요한 것만을 골라서 사용하면 된다. 이벤트 처리기에서 이벤트를 처리하였으면 true를 반환하고 처리하지 않았으면 false를 반환한다.

```

@Override
protected void onDraw(Canvas canvas) {
    Paint paint = new Paint();
    paint.setTextSize(60);
    canvas.drawText("(" + x + " ", "+y+") 에서 터치 이벤트가 발생하였음", x, y, paint);
}

```

```

}

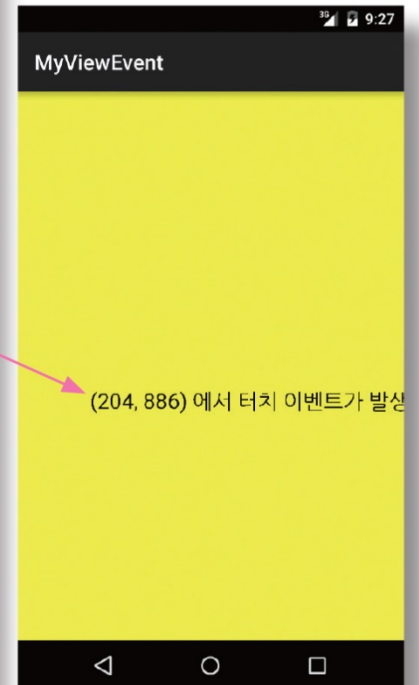
```

```

public class MyViewEventActivity extends ActionBarActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        MyView w = new MyView(this);
        setContentView(w);
    }
}

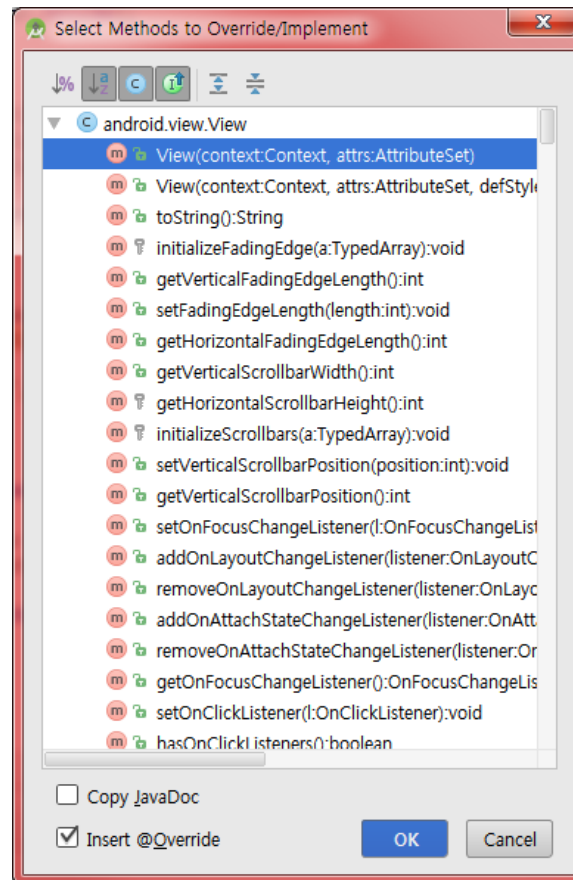
```



(204, 886) 에서 터치 이벤트가 발생

# 콜백 메소드를 재정의할 때 편리한 기능

- [Code]->[Override Methods...] 사용!



## 예제: 볼륨 컨트롤 작성

- 이미지 “knob.png” 를 res/drawable 폴더에 저장한다.





# VOLUMECONTROLVIEW.JAVA

```
public class VolumeControlView extends ImageView implements View.OnTouchListener
{

    private double angle = 0.0;
    private KnobListener listener;
    float x, y;
    float mx, my;

    public interface KnobListener {
        public void onChanged(double angle);
    }

    public void setKnobListener(KnobListener lis) {
        listener = lis;
    }

    public VolumeControlView(Context context) {
        super(context);
        this.setImageResource(R.drawable.knob);
        this.setOnTouchListener(this);
    }

    public VolumeControlView(Context context, AttributeSet attrs) {
        super(context, attrs);
        this.setImageResource(R.drawable.knob);
        this.setOnTouchListener(this);
    }
}
```

# VOLUMECONTROLVIEW.JAVA

```
private double getAngle(float x, float y) {  
    mx = x - (getWidth() / 2.0f);  
    my = (getHeight() / 2.0f) - y;  
  
    double degree = Math.atan2(mx, my) * 180.0 / 3.141592;  
    return degree;  
}
```

```
@Override  
public boolean onTouch(View v, MotionEvent event) {  
    x = event.getX(0);  
    y = event.getY(0);  
    angle = getAngle(x, y);  
    invalidate();  
    listener.onChanged(angle);  
  
    return true;  
}
```

```
protected void onDraw(Canvas c) {  
    Paint paint = new Paint();  
    c.save();  
    c.rotate((float) angle, getWidth() / 2, getHeight() / 2);  
    super.onDraw(c);  
    c.restore();  
}
```

```
}
```

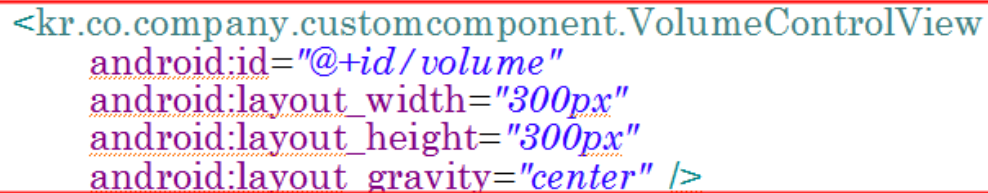
# MAIN.XML

main.xml

---

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".CustomComponentActivity" >
```

```
    <kr.co.company.customcomponent.VolumeControlView
        android:id="@+id/volume"
        android:layout_width="300px"
        android:layout_height="300px"
        android:layout_gravity="center" />
```



```
</RelativeLayout>
```

---

# CUSTOMCOMPONENTACTIVITY.JAVA

*CustomComponentActivity.java*

---

```
package kr.co.company.customcomponent;
```

```
import android.os.Bundle;
```

```
import android.support.v7.app.ActionBarActivity;
```

```
public class CustomComponentActivity extends ActionBarActivity {
```

```
    /** Called when the activity is first created. */
```

```
    @Override
```

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

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

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

```
        VolumeControlView view = (VolumeControlView) findViewById(R.id.volume);
```

```
        view.setKnobListener(new VolumeControlView.KnobListener() {
```

```
            @Override
```

```
            public void onChanged(double angle) {
```

```
                if (angle > 0)
```

```
                    ; // 오른쪽으로 회전
```

```
            else
```

```
                ; // 왼쪽으로 회전
```

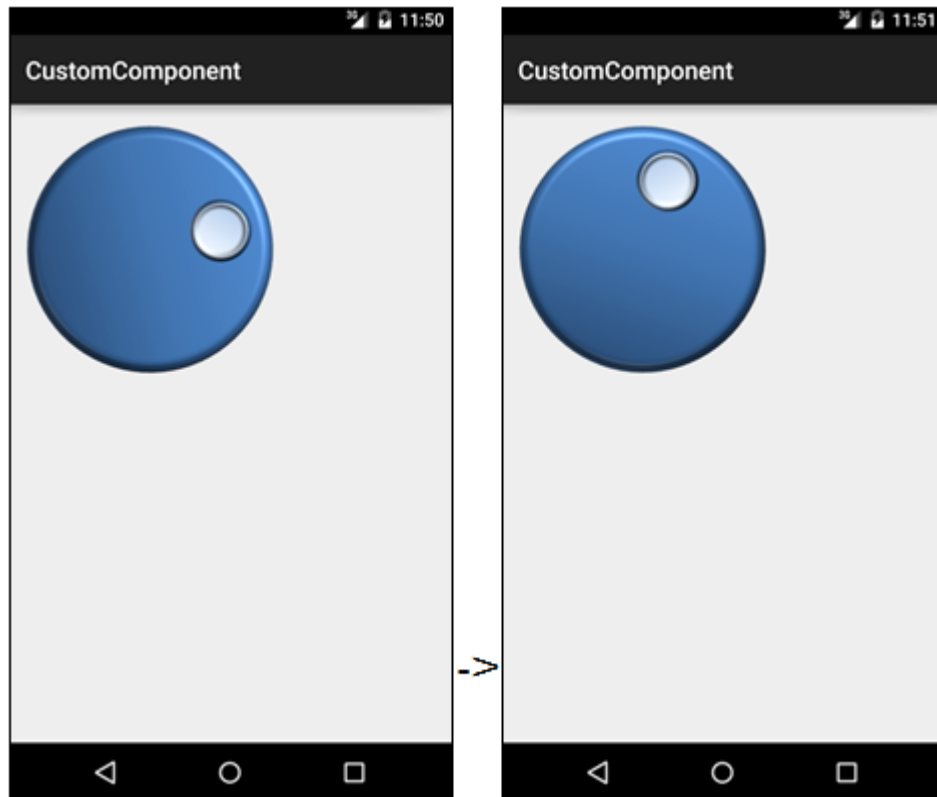
```
            }
```

```
        });
```

```
    }
```

```
}
```

## 실행 결과 (실습)



# 터치 이벤트

- 일반적으로 커스텀 뷰를 정의하고 onTouchEvent() 재정의

```
class MyView extends View { ...  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        x = (int) event.getX();  
        y = (int) event.getY();  
        ...  
    }  
}
```

# 터치 이벤트의 종류

액션	설명
ACTION_DOWN	누르는 동작이 시작됨
ACTION_UP	누르고 있다가 떼 때 발생함
ACTION_MOVE	누르는 도중에 움직임
ACTION_CANCEL	터치 동작이 취소됨
ACTION_OUTSIDE	터치가 현재의 위젯을 벗어남



ACTION\_DOWN



ACTION\_UP



ACTION\_MOVE

# 터치 이벤트 예제

```
public class TouchEventActivity extends Activity {  
    protected class MyView extends View {  
        int x = 100, y = 100;  
        String str;  
        public MyView(Context context) {  
            super(context);  
            setBackgroundColor(Color.YELLOW);  
        }  
        @Override  
        protected void onDraw(Canvas canvas) {  
            Paint paint = new Paint();  
            paint.setColor(Color.MAGENTA);  
            canvas.drawRect(x, y, x + 50, y + 50, paint);  
            canvas.drawText("액션의 종류: " + str, 0, 20, paint);  
        }  
    }  
}
```



# 터치 이벤트 예제

@Override

```
public boolean onTouchEvent(MotionEvent event) {  
    x = (int) event.getX();  
    y = (int) event.getY();  
    if (event.getAction() == MotionEvent.ACTION_DOWN)  
        str = "ACTION_DOWN";  
    if (event.getAction() == MotionEvent.ACTION_MOVE)  
        str = "ACTION_MOVE";  
    if (event.getAction() == MotionEvent.ACTION_UP)  
        str = "ACTION_UP";  
    invalidate();  
    return true;  
}
```

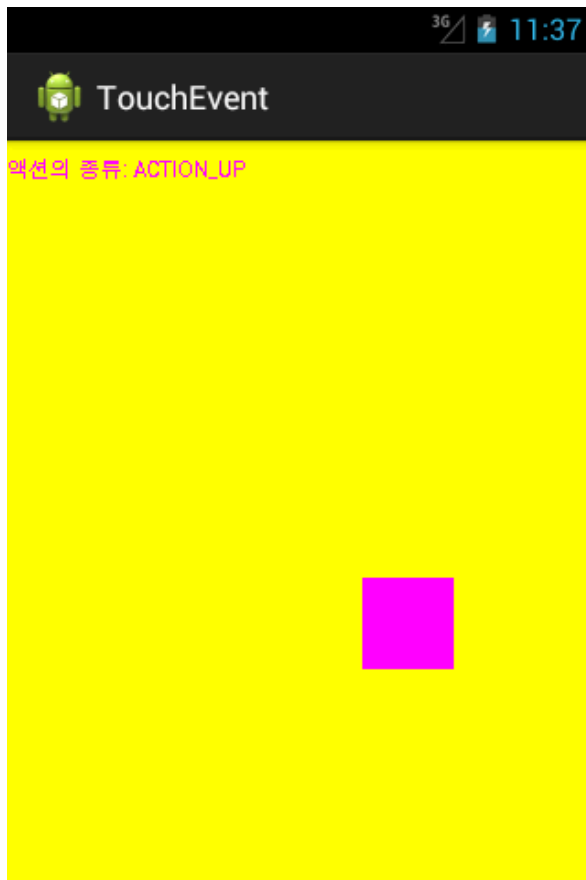
}

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    MyView w = new MyView(this);  
    setContentView(w);  
}
```

}

# 실행결과



## 실습 (1)

- 볼륨 컨트롤 예제를 좀더 자연스럽게 동작하도록 수정하십시오. 휠이 클릭한 위치에서부터 동작하도록 하십시오.