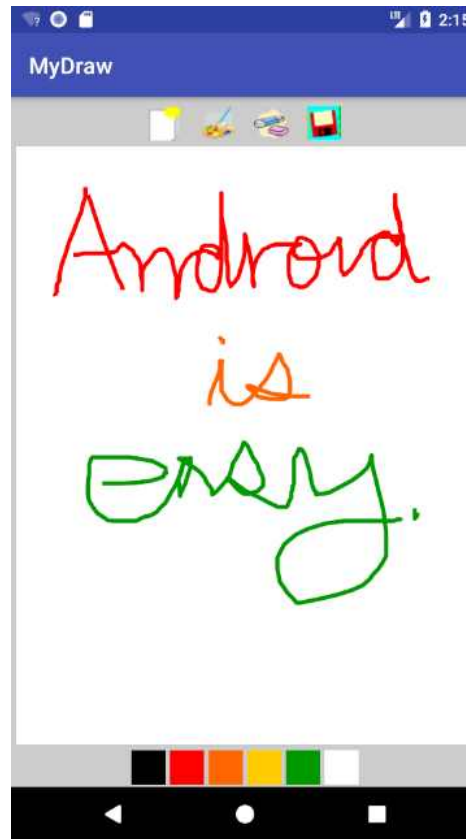


CHAP 4. 이벤트 처리

목표

□ 그림판 만들기

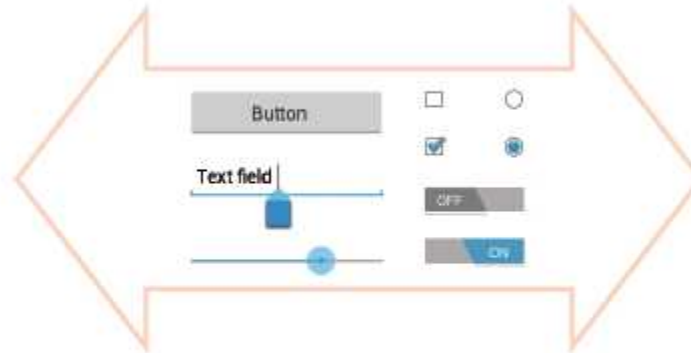


입력 위젯

- 버튼, 텍스트 필드, 시크 바, 체크 박스, 줌 버튼, 토글 버튼



사용자



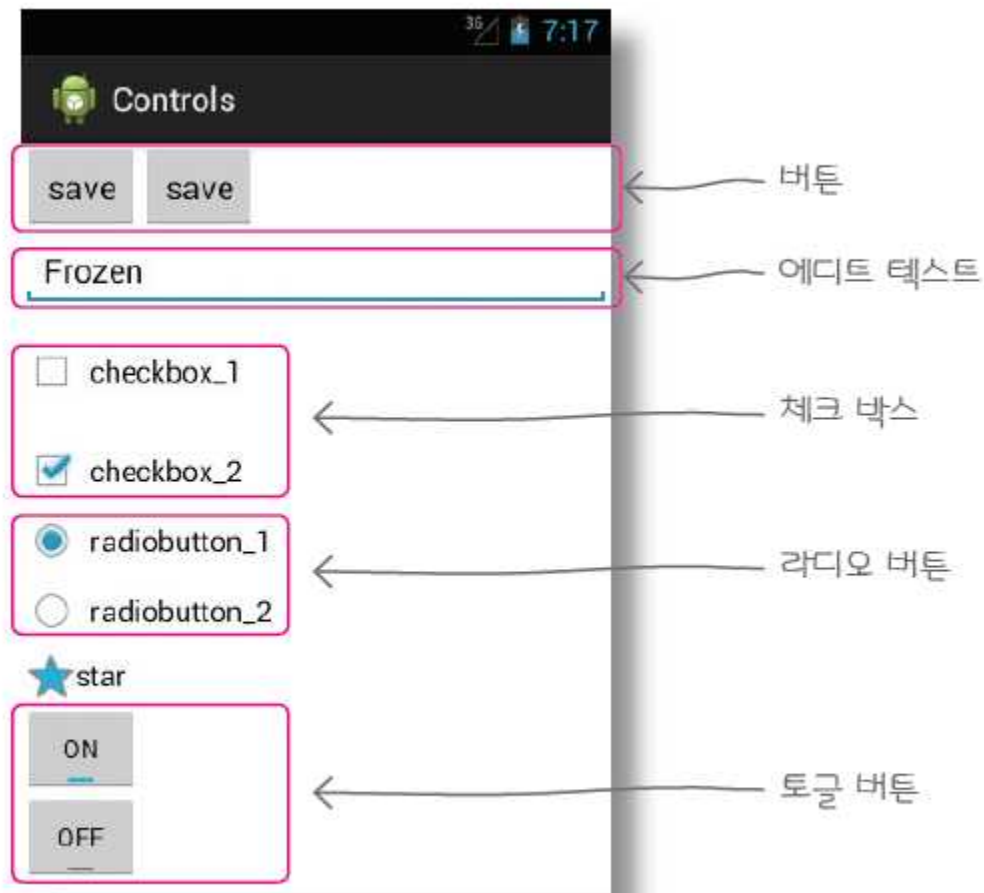
입력 위젯



입력 위젯의 종류

위젯	설명	관련 클래스
Button	어떤 동작을 수행하기 위하여 사용자가 누를 수 있고 클릭할 수 있는 푸시 버튼	Button
Text field	편집이 가능한 텍스트 필드. 자동 완성 기능을 제공하려면 AutoCompleteTextView를 사용한다.	EditText, AutoCompleteTextView
Checkbox	사용자에 의하여 토글될 수 있는 on/off 스위치. 사용자가 그룹에서 여러 가지 옵션을 동시에 선택할 수 있게 하려면 체크 박스를 사용한다.	CheckBox
Radio button	체크 박스와 비슷하지만 그룹에서 하나의 옵션만 선택할 수 있다.	RadioGroup RadioButton
Toggle button	라이트 인디케이터가 있는 on/off 버튼	ToggleButton
Spinner	사용자가 여러 값 중에서 하나를 선택할 수 있는 드롭 다운 리스트	Spinner
Pickers	up/down 버튼이나 스와이프 제스처를 통하여 하나의 값을 선택하는 대화 상자. 날짜를 선택하려면 DatePicker를 사용한다. 시간을 선택하려면 TimePicker를 사용한다.	DatePicker TimePicker

입력 위젯



버튼

□ 텍스트 버튼

```
<Button
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_text"
```

```
... />
```



□ 이미지 버튼

```
<ImageButton
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/button_icon"
```

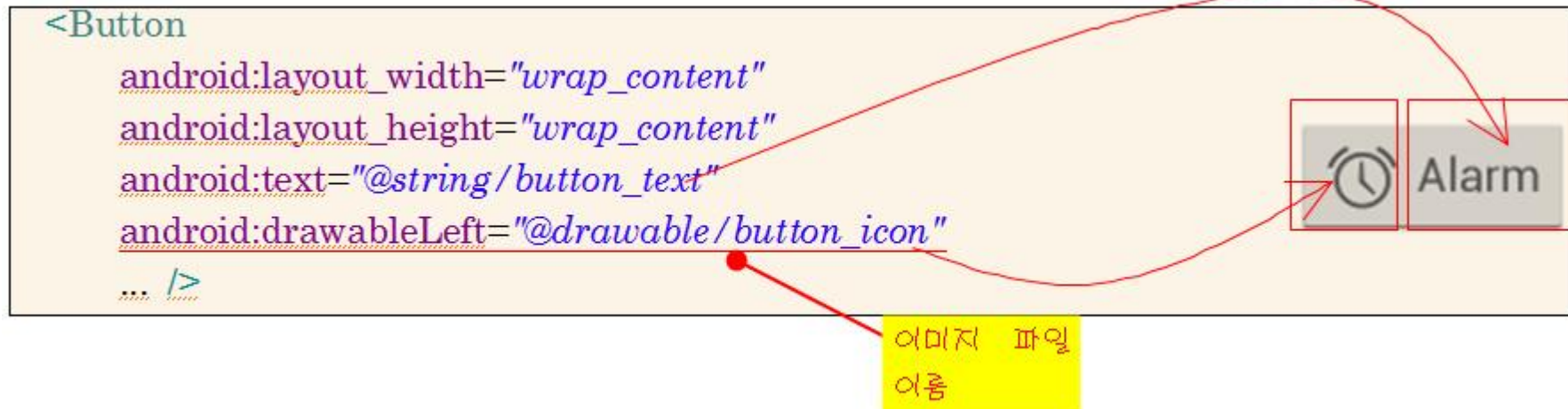
```
... />
```

이미지 파일
이름



버튼

- 텍스트와 이미지를 동시에 가지는 버튼



버튼의 이벤트 처리

사용자가 클릭하면 호출된다.



```
<?xml version="1.0" encoding="utf-8"?>
<Button
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="sendMessage"
    android:text="@string/button_send" />
```

```
public class MyActivity extends Activity {
    @Override
    public void onCreate(...) {
        //...
    }
    public void sendMessage(View view)
    {
    }
}
```


이벤트를 처리하는 가장 간단한 방법

- 레이아웃 안의 **<Button>** 요소에 **onClick** 속성을 추가

```
<?xml version="1.0" encoding="utf-8"?>
<Button xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onClick"
    android:text="@string/button_send" />
```

onClick 속성에 이벤트를
처리하는 메소드 이름을
적는다.

<ImageButton> 태그를 사용한 이미지 버튼

activity_main.xml

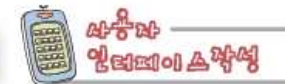
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:src="@drawable/mybutton" />
```

← 이미지 버튼을 정의한다.

← XML로 이벤트 처리 메소드를 지정하자. 이것이 가장 간단하다.

```
</LinearLayout>
```



이벤트 처리 코드

MainActivity.java



```
package kr.co.company.imagebutton1;  
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

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

```
@Override
```

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

이미지 버튼이 눌러지면 토스트 메시지를 출력한다.

```
public void onClick(View target) {  
    Toast.makeText(getApplicationContext(), "버튼이 눌러졌습니다",  
        Toast.LENGTH_SHORT).show();  
}
```

```
}
```

레이아웃 파일

- 레이아웃 파일에 버튼을 정의한다.

/res/layout/activity_main.xml

```
<?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" >

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/android_button"
        android:padding="10dp" />

</LinearLayout>
```

커스텀 버튼



MainActivity.java

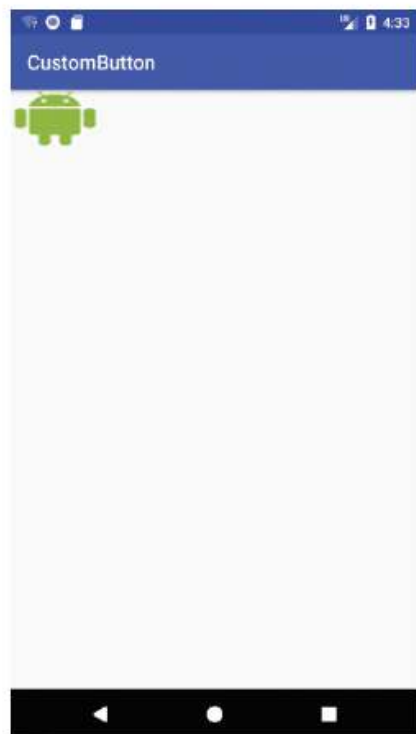
```
package kr.co.company.custombutton;
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick(View target) {
        Toast.makeText(getApplicationContext(), "Beep Bop",
            Toast.LENGTH_SHORT).show();
    }
}
```

버튼이 클릭되면 토스트 메시지가 출력된다.

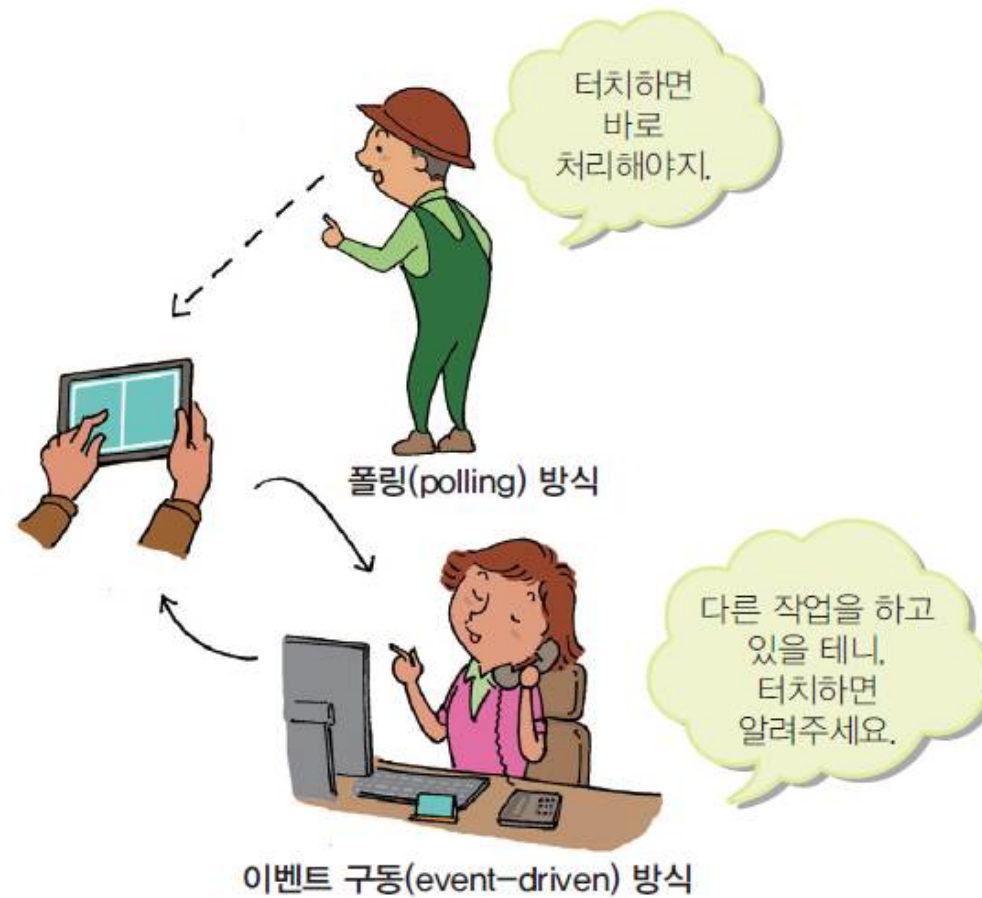
실행 결과



버튼의 상태에 따라서 이
미지가 변경되는 것을 확
인한다.



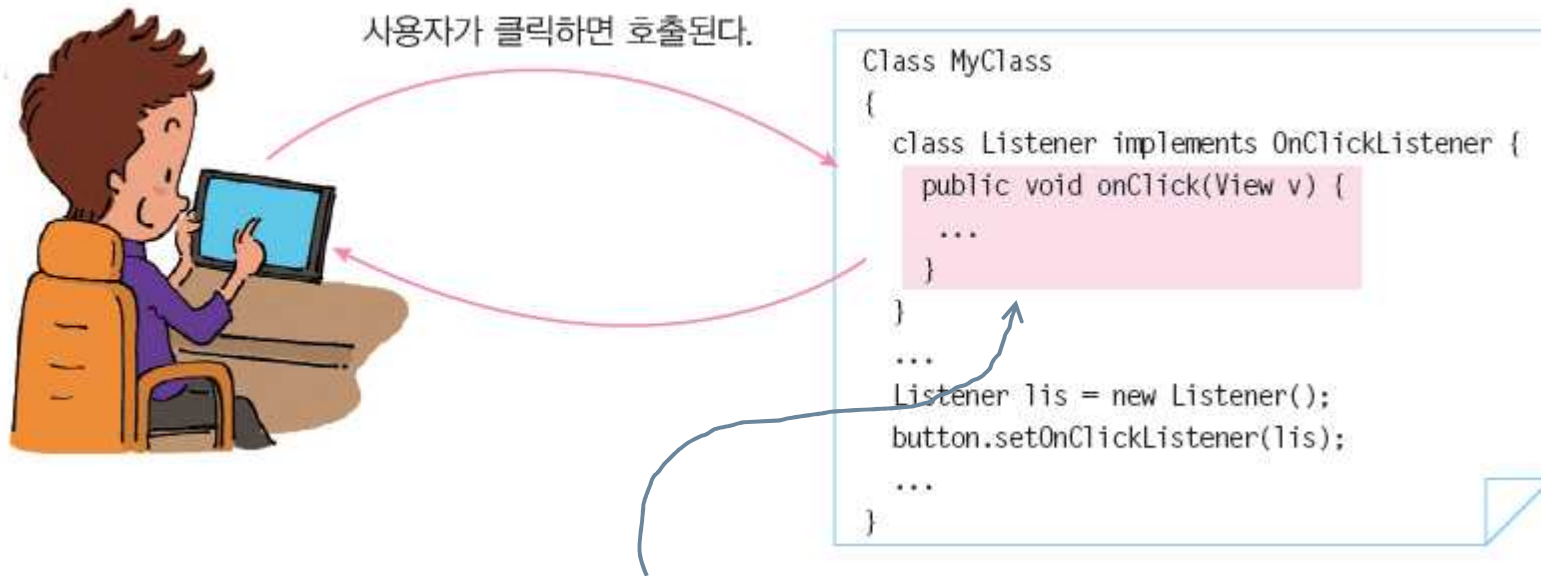
폴링과 이벤트 구동 방식



안드로이드에서의 이벤트 처리 방법

- XML 파일에 이벤트 처리 메소드를 등록하는 방법
 - ▣ 가장 쉬운 방법 <- 권장
 - ▣ 앞에서 살펴보았음!
- 이벤트 처리 객체를 생성하여 컴포넌트에 등록
 - ▣ 일반적인 방법
- 뷰 클래스의 이벤트 처리 메소드를 재정의
 - ▣ 커스텀 뷰를 작성하는 경우: (예) 게임

이벤트 처리 객체 사용



버튼에 붙은 리스너 객체가 이벤트를 처리한다.

이벤트 리스너



전체적인 구조

```
class MyClass
```

```
{
```

```
    class Listener implements OnClickListener {  
        public void onClick(View v){
```

```
            ...
```

```
        }
```

```
    }
```

```
    ...
```

```
    Listener lis = new Listener();
```

```
    button.setOnClickListener(lis);
```

```
    ...
```

```
}
```

인터페이스를 구현한 클래스 정의

이벤트 리스너 객체 생성

버튼에 이벤트 리스너 객체를 등록

리스너의 종류

리스너	콜백 메소드	설명
View.OnClickListener	<code>onClick()</code>	사용자가 어떤 항목을 터치하거나 내비게이션 키나 트랙볼로 항목으로 이동한 후에 엔터키를 눌러서 선택하면 호출된다.
View.OnLongClickListener	<code>onLongClick()</code>	사용자가 항목을 터치하여서 일정 시간 동안 그대로 누르고 있으면 발생한다.
View.OnFocusChangeListener	<code>onFocusChange()</code>	사용자가 하나의 항목에서 다른 항목으로 포커스를 이동할 때 호출된다.
View.OnKeyListener	<code>onKey()</code>	포커스를 가지고 있는 항목 위에서 키를 눌렀다가 놓았을 때 호출된다.
View.OnTouchListener	<code>onTouch()</code>	사용자가 터치 이벤트로 간주되는 동작을 한 경우에 호출된다.

리스너 객체를 생성하는 방법

- 리스너 클래스를 내부 클래스로 정의한다.
- 리스너 클래스를 무명 클래스로 정의한다.
- 리스너 인터페이스를 액티비티 클래스에 구현한다.

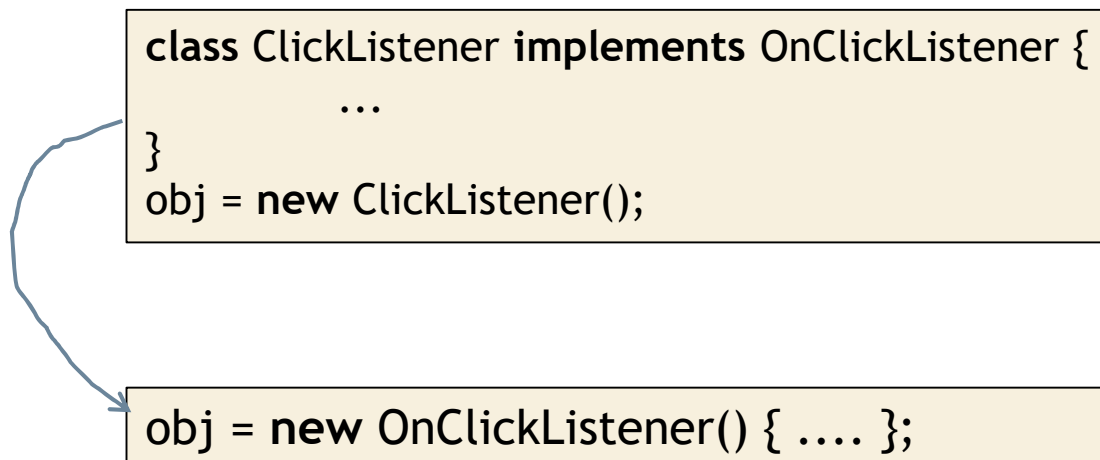
가장 많이 사용되는 방법!

무명 클래스로 이벤트를 처리하는 예제

```
...
public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(),
                    "버튼이 눌러졌습니다", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

무명 클래스

- 클래스 몸체는 정의되지만 이름이 없는 클래스이다.
- 무명 클래스는 클래스를 정의하면서 동시에 객체를 생성하게 된다.



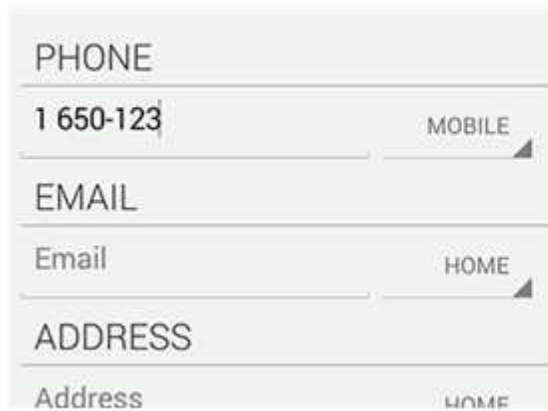
코드 분석

무명 클래스 사용

```
...  
public class MainActivity extends AppCompatActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Button button = (Button) findViewById(R.id.button);  
        button.setOnClickListener(new OnClickListener() {  
            public void onClick(View v) {  
                Toast.makeText(getApplicationContext(), "버튼  
                눌러졌습니다", Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

텍스트 필드

- 텍스트 필드(**text field**)를 사용하면 사용자가 앱에 텍스트를 타이핑하여 입력할 수 있다.
- 단일 라인이거나 멀티 라인일 수 있다.



PHONE

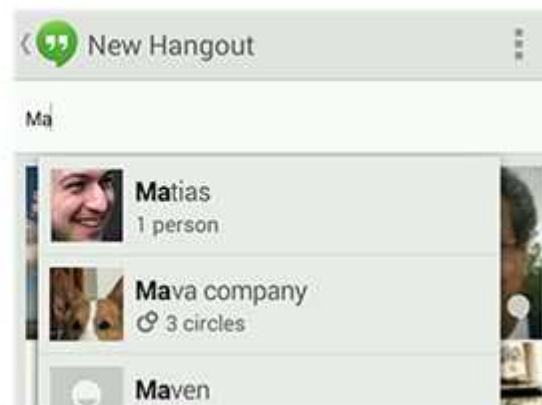
1 650-123 MOBILE

EMAIL

Email HOME

ADDRESS

Address HOME



키보드 종류 지정

```
<EditText  
    android:id="@+id/email_address"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:hint="@string/email_hint"  
    android:inputType="textEmailAddress" />
```

이메일 형태의 입력을 받는다.



(textEmailAddress 입력 타입)



(phone 입력 타입)

에디트 텍스트

activity_main.xml

```
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">
```

```
    <EditText
        android:id="@+id/edittext"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="여기에 텍스트를 입력하십시오."
        android:inputType="text" />
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="텍스트 보이기" />
```

```
</LinearLayout>
```

에디트 텍스트



에디트 텍스트의 이벤트 처리

MainActivity.java

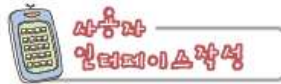
```
package kr.co.company.edittest;  
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        final EditText editText;  
        Button btn;  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        editText = (EditText) findViewById(R.id.edittext);  
        btn = (Button) findViewById(R.id.button);  
        btn.setOnClickListener(new View.OnClickListener() {  
            public void onClick(View v) {  
                String str = editText.getText().toString();  
                Toast.makeText(getApplicationContext(), str, Toast.LENGTH_LONG).show();  
            }  
        });  
    }  
}
```

여기에 final이 붙는 이유는 접근사항에 있음!

사용자가 "Send" 버튼을 누르면 화면에 토스트 메시지를 표시한다.

체크 박스



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

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

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent" >
```

```
    <CheckBox android:id="@+id/checkbox_meat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="고기"
        android:onClick="onCheckboxClicked" />
```

← 체크 박스

```
    <CheckBox android:id="@+id/checkbox_cheese"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="치즈"
```

```
        android:onClick="onCheckboxClicked" />
```

```
</LinearLayout>
```

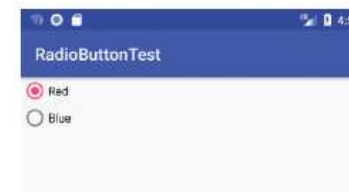
체크박스의 이벤트 처리

```
public void onCheckboxClicked(View view) {  
    boolean checked = ((CheckBox) view).isChecked();  
  
    switch(view.getId()) {  
        case R.id.checkbox_meat:  
            if (checked)  
                Toast.makeText(getApplicationContext(), "고기 선택",  
                    Toast.LENGTH_SHORT).show();  
            else  
                Toast.makeText(getApplicationContext(), "고기 선택 해제",  
                    Toast.LENGTH_SHORT).show();  
            break;  
        case R.id.checkbox_cheese:  
            if (checked)  
                Toast.makeText(getApplicationContext(), "치즈 선택",  
                    Toast.LENGTH_SHORT).show();  
            else  
                Toast.makeText(getApplicationContext(), "치즈 선택 해제",  
                    Toast.LENGTH_SHORT).show();  
            break;  
    }  
}
```


라디오 버튼

activity_main.xml

```
<?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" >
    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >
        <RadioButton
            android:id="@+id/radio_red"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="onRadioButtonClicked"
            android:text="Red" />
        <RadioButton
            android:id="@+id/radio_blue"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="onRadioButtonClicked"
            android:text="Blue" />
    </RadioGroup>
</LinearLayout>
```



라디오 그룹 안에 라디오 버튼을 2개 정의한다.

라디오 버튼의 이벤트 처리

MainActivity.java



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

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

```
    @Override
```

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

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

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

```
    }
```

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

```
        boolean checked = ((RadioButton) view).isChecked();
```

```
        switch(view.getId()) {
```

```
            case R.id.radio_red:
```

라디오 버튼의 이벤트를 처리한다.

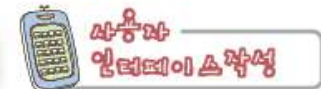
버튼의 체크 여부를 읽는다.

```
        if (checked)
            Toast.makeText(getApplicationContext(),
                ((RadioButton) view).getText(),
                Toast.LENGTH_SHORT).show();
        break;
    case R.id.radio_blue:
        if (checked)
            Toast.makeText(getApplicationContext(),
                ((RadioButton) view).getText(),
                Toast.LENGTH_SHORT).show();
        break;
    }
}
```


토글 버튼

- XML로 라디오 버튼을 정의한다.

activity_main.xml



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

```
<ToggleButton
    android:id="@+id/togglebutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Vibrate on"
    android:textOff="Vibrate off"
    android:onClick="onToggleClicked" />
```



← 토글 버튼

```
</LinearLayout>
```

MainActivity.java

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

```
public class MainActivity extends AppCompatActivity {

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

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

```
        boolean on = ((ToggleButton) view).isChecked();

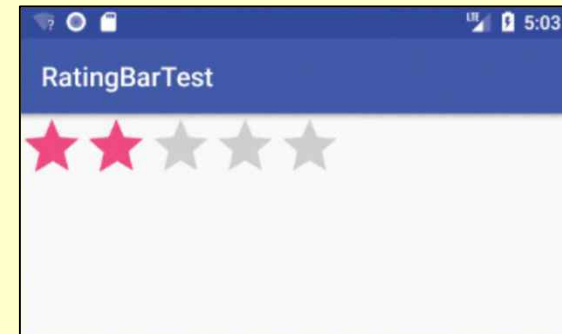
        if (on) {
            Toast.makeText(getApplicationContext(), "Checked",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Not checked",
                Toast.LENGTH_SHORT).show();
        }
    }
```

```
}
```

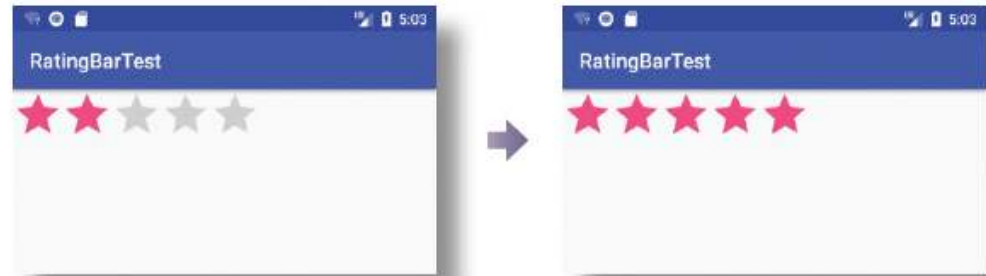
레이팅바

- XML로 레이팅 바를 정의한다.

```
...  
<RatingBar  
    android:id="@+id/ratingbar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:numStars="5"  
    android:stepSize="1.0"  
</>
```



레이팅바



...

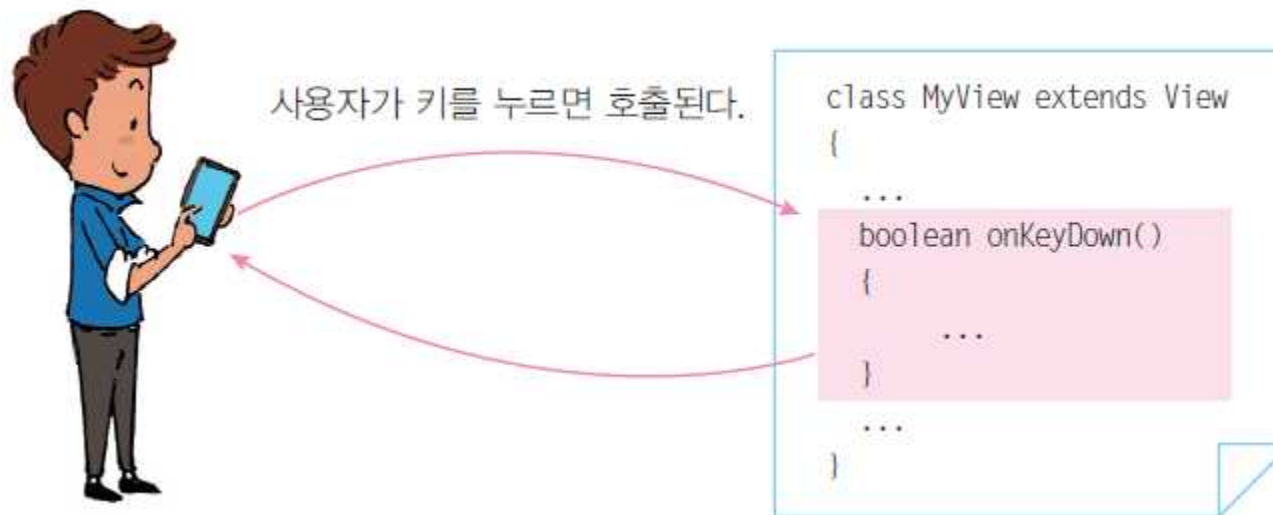
```
public class MainActivity extends AppCompatActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        final RatingBar ratingbar = (RatingBar) findViewById(R.id.ratingbar);  
        ratingbar.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {  
            public void onRatingChanged(RatingBar ratingBar, float rating,  
                boolean fromUser) {  
                Toast.makeText(getApplicationContext(),  
                    "New Rating: " + rating, Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

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

MainActivity.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(0);
```

```
        y = (int) event.getY(0);
```

```
        invalidate();
```

```
        return super.onTouchEvent(event);
```

```
    }
```

```
    @Override
```

```
    protected void onDraw(Canvas canvas) {
```

```
        Paint paint = new Paint();
```

```
        paint.setTextSize(50);
```

```
        canvas.drawCircle(x, y, 30, paint);
```

```
        canvas.drawText("(" + x + ", " + y + ") 에서 터치 이벤트가 발생하였음",
```

```
        paint);
```

```
    }
```

```
}
```



예제

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        MyView w = new MyView(this);  
        setContentView(w);  
    }  
}
```

커스텀 뷰를 액티비티의 화면으로 설정한다.

예제: 보름 컨트롤 작성

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



VolumeControlView.java

VolumeControlView.java

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

```
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

```
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);  
    }
```

ImageView를 상속받아서 자신만의
커스텀 컴포넌트를 정의한다.

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

```
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

activity_main.xml

```
<?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" >

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

    <RatingBar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/volume"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="13dp"
        android:layout_marginStart="13dp"
        android:layout_marginTop="36dp"
        android:id="@+id/ratingBar" />

</LinearLayout>
```


MainActivity.java

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

// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

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

```
    @Override
```

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

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

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

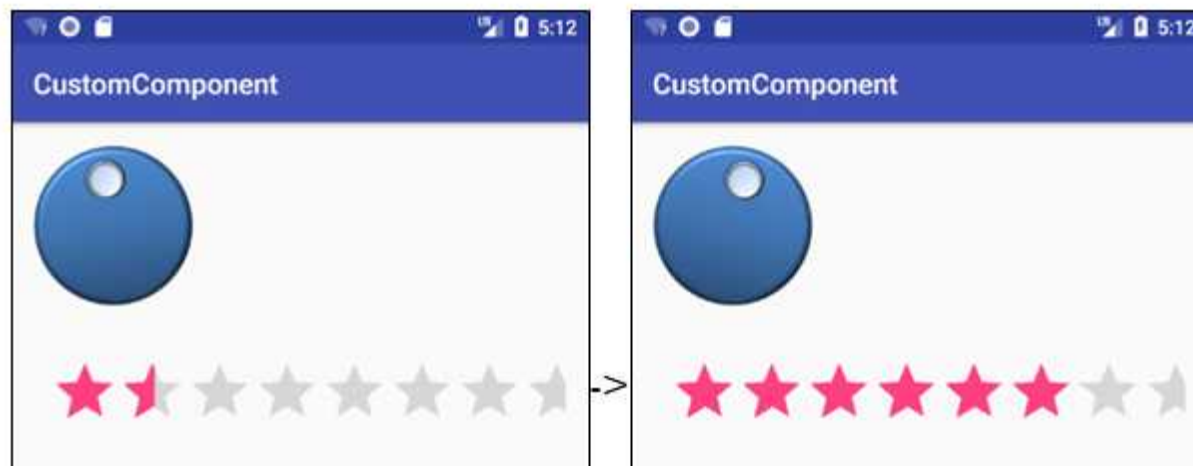
```
        final RatingBar ratingbar = (RatingBar)
            findViewById(R.id.ratingBar);
```

```
        VolumeControlView view = (VolumeControlView) findViewById(R.id.volume);
        view.setKnobListener(new VolumeControlView.KnobListener() {
```

```
            @Override
```

```
            public void onChanged(double angle) {
                float rating = ratingbar.getRating();
                if (angle > 0 && rating < 7.0)
                    // 오른쪽으로 회전
                    ratingbar.setRating(rating+1.0f);
                else if (rating > 0.0)
                    // 왼쪽으로 회전
                    ratingbar.setRating(rating-1.0f);
            }
        });
    }
}
```

실행 결과



터치 이벤트

- 일반적으로 커스텀 뷰를 정의하고 onTouchEvent() 재정의

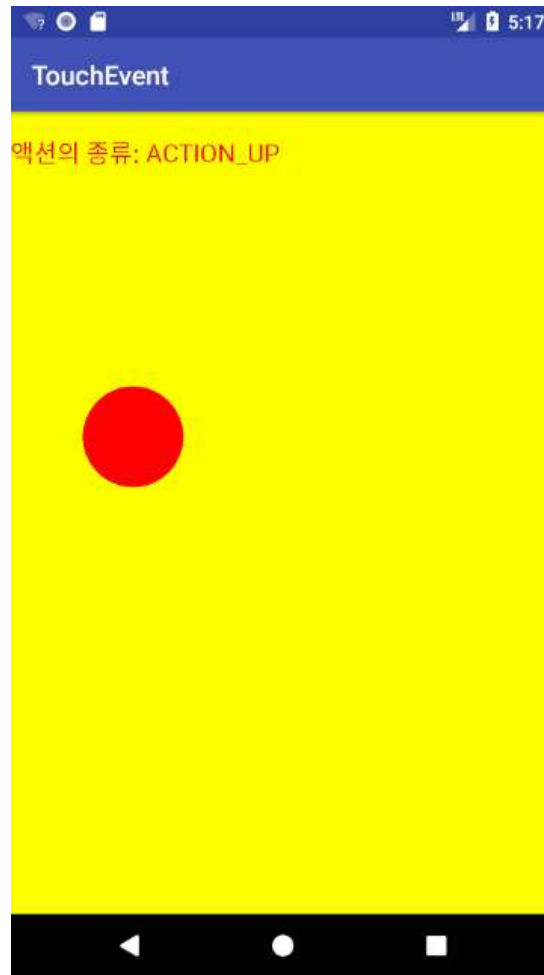
```
class MyView extends View { ...
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        x = (int) event.getX();
        y = (int) event.getY();
        ...
    }
}
```


터치 이벤트의 종류

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



터치로 얻은 움직임



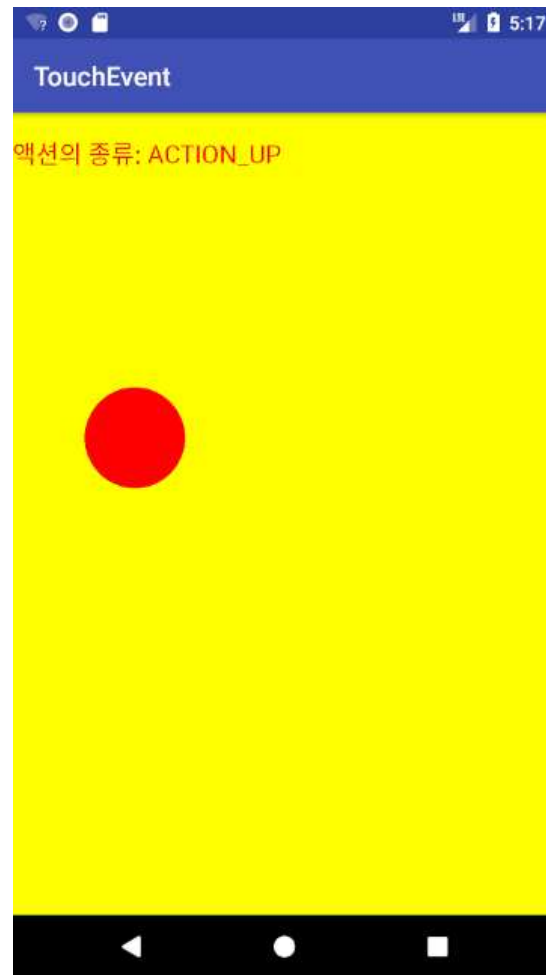
터치 이벤트 예제

```
public class MainActivity extends AppCompatActivity {  
    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.RED);  
            canvas.drawCircle(x, y, 100, paint);  
            paint.setTextSize(50);  
            canvas.drawText("액션의 종류: " + str, 0, 100, paint);  
        }  
    }  
}
```

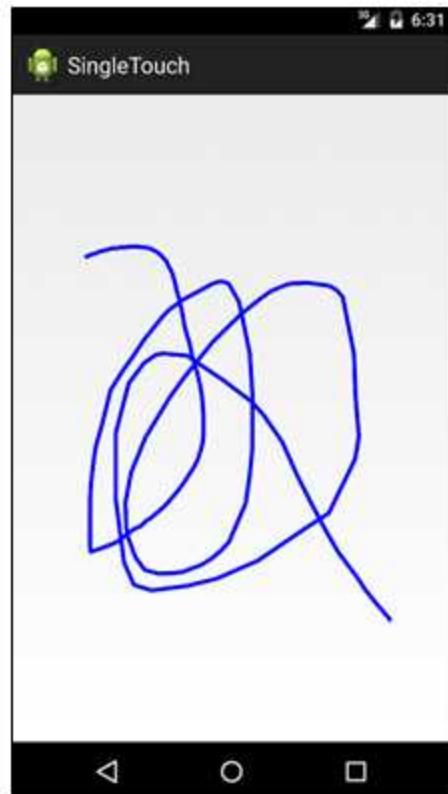
터치 이벤트 예제

```
@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);
}
}
```

실행 결과



터치로 곡선 그리기



터치로 곡선 그리기 #1

MainActivity.java

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

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(new SingleTouchView(this, null));  
    }  
}
```

터치로 곡선 그리기 #2

SingleTouchView.java

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

```
public class SingleTouchView extends View {  
    private Paint paint = new Paint();  
    private Path path = new Path();  
  
    public SingleTouchView(Context context, AttributeSet attrs) {  
        super(context, attrs);  
  
        paint.setAntiAlias(true);  
        paint.setStrokeWidth(10f);  
        paint.setColor(Color.BLUE);  
        paint.setStyle(Paint.Style.STROKE);  
        paint.setStrokeJoin(Paint.Join.ROUND);  
    }  
}
```


터치로 곡선 그리기 #3

```
@Override
protected void onDraw(Canvas canvas) {
    canvas.drawPath(path, paint);
}
```

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    float eventX = event.getX();
    float eventY = event.getY();

    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            path.moveTo(eventX, eventY);
            return true;
        case MotionEvent.ACTION_MOVE:
            path.lineTo(eventX, eventY);
            break;
        case MotionEvent.ACTION_UP:
            break;
        default:
            return false;
    }

    invalidate();
    return true;
}
```

실험결과 글쓰기



멀티 터치

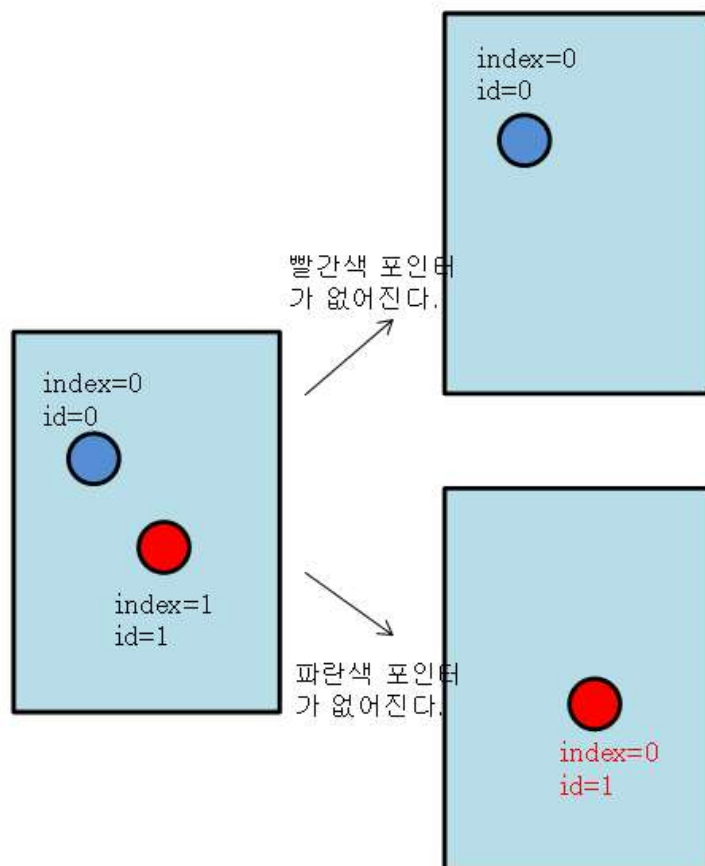
- 여러 개의 손가락을 이용하여 화면을 터치하는 것으로 이미지를 확대/축소할 때 많이 사용된다.



터치 이벤트

- ACTION_DOWN – 화면을 터치하는 첫 번째 포인터에 대하여 발생한다. 제스처 인식이 시작된다. 첫 번째 터치는 항상 MotionEvent에서 인덱스 0번에 저장된다.
- ACTION_POINTER_DOWN – 첫 번째 포인터 이외의 포인터에 대하여 발생된다. 포인터 데이터는 `getActionIndex()`이 반환하는 인덱스에 저장된다.
- ACTION_MOVE – 화면을 누르면서 이동할 때 발생한다.
- ACTION_POINTER_UP – 마지막 포인터가 아닌 다른 포인터가 화면에서 없어지면 발생된다.
- ACTION_UP – 화면을 떠나는 마지막 포인터에 대하여 발생된다.

인덱스와 아이디



터치된 위치에 원을 그리는 예제

MainActivity.java

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

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

```
    @Override
```

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

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

```
        setContentView(new MultiTouchView(this, null));
```

```
    }
```

```
}
```

액티비티의 화면을
MultiTouchView 객체로 설정한다.

MultiTouchView.java

■ **package** kr.co.company.multitouch;
// 소스만 입력하고 **Ctrl-Shift-O**를 눌러서 **import** 문장을 자동으로 생성한다.

```
public class MultiTouchView extends View {  
  
    private static final int SIZE = 60;  
  
    final int MAX_POINTS = 10;  
    float[] x = new float[MAX_POINTS];  
    float[] y = new float[MAX_POINTS];  
    boolean[] touching = new boolean[MAX_POINTS];  
  
    private Paint mPaint;  
  
    public MultiTouchView(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        initView();  
    }  
  
    private void initView() {  
        mPaint = new Paint(Paint.ANTI_ALIAS_FLAG);  
        mPaint.setColor(Color.BLUE);  
        mPaint.setStyle(Paint.Style.FILL_AND_STROKE);  
    }  
}
```

최대 10개 포인터의 위치와
상태를 저장할 수 있다.

@Override

public boolean onTouchEvent(MotionEvent event) {

int index = event.getActionIndex();
int id = event.getPointerId(index);
int action = event.getActionMasked();

인덱스로부터 포인터의 아이디를 구한다.

switch (action) {

case MotionEvent.ACTION_DOWN:

case MotionEvent.ACTION_POINTER_DOWN:

x[id] = (int) event.getX(index);
y[id] = (int) event.getY(index);
touching[id] = true;

break;

case MotionEvent.ACTION_MOVE:

break;

case MotionEvent.ACTION_UP:

case MotionEvent.ACTION_POINTER_UP:

case MotionEvent.ACTION_CANCEL:

touching[id] = false;

break;

}

invalidate();

return true;

}

화면이 터치되면 위치를 계산하여 배열에 저장한다.
touching[] 배열에 true를 저장하여서 현재 터치가 되어 있다는 것을 표시한다.

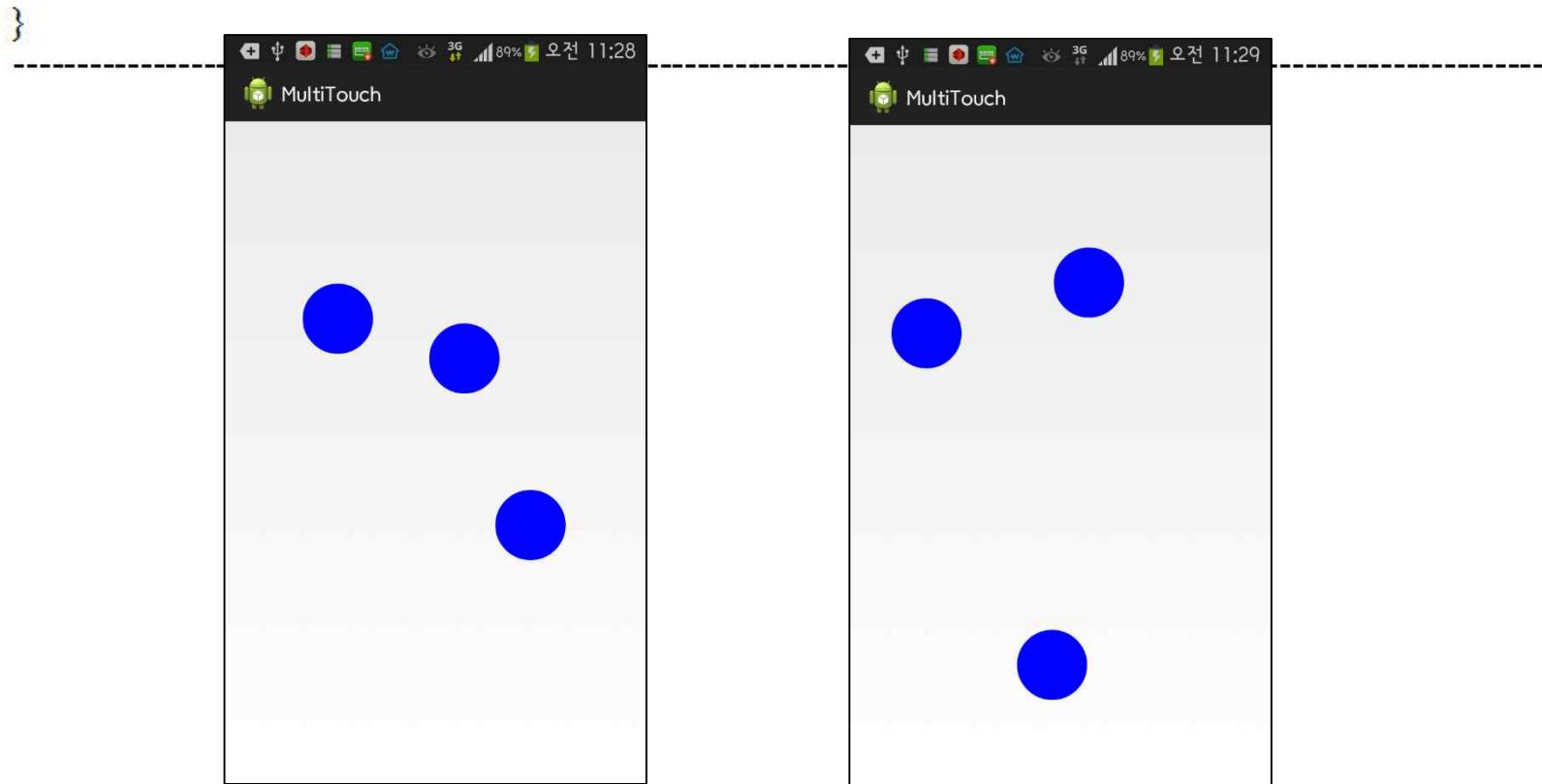
처리가 종료되었음을 저장한다.

@Override

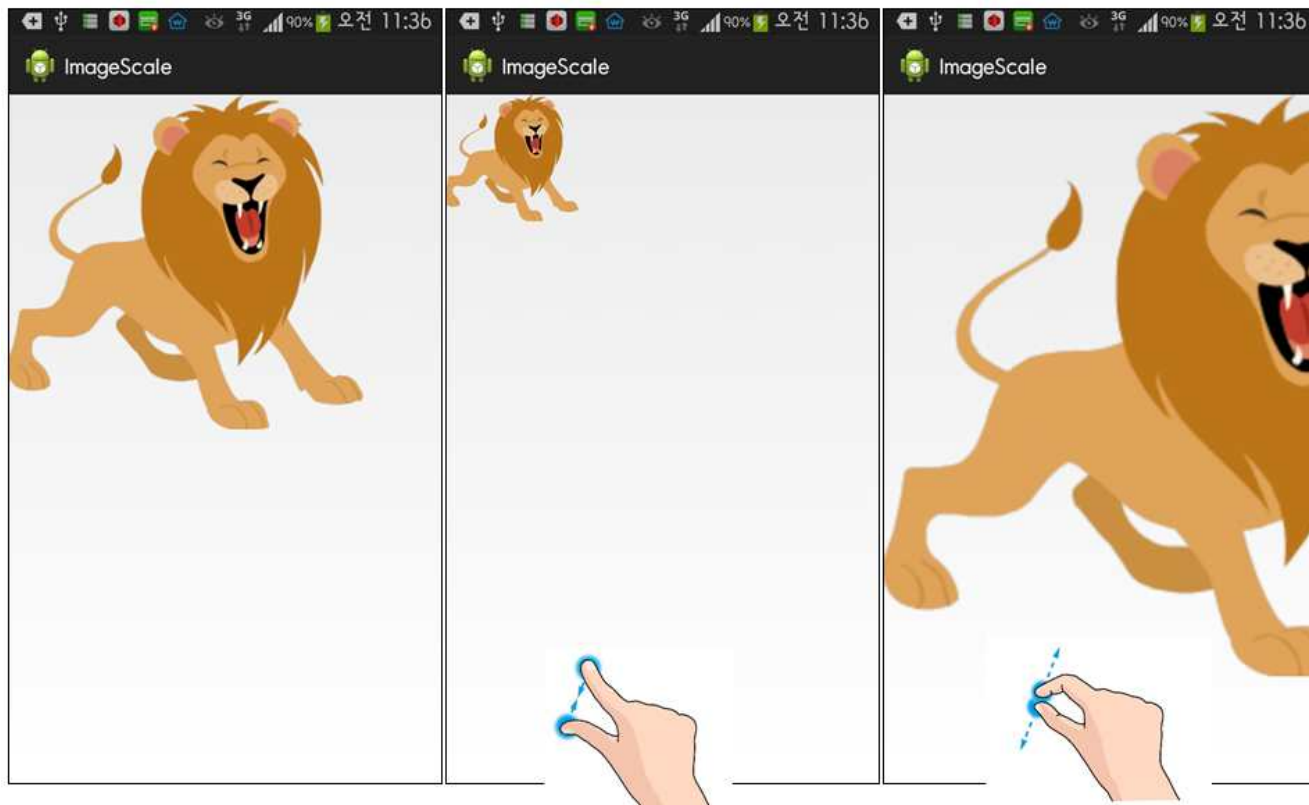
```
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);
```

```
    for (int i = 0; i < MAX_POINTS; i++) {  
        if (touching[i]) {  
            canvas.drawCircle(x[i], y[i], SIZE, mPaint);  
        }  
    }  
}
```

현재 터치되어 있는 포인트
위치에 원을 그린다.



핀치Zoom 구현



액티비티 정의

MainActivity.java

```
package kr.co.company.imagescale;
```

```
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

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

```
    @Override
```

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

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

```
        setContentView(new MyImageView(this));
```

```
    }
```

```
}
```

부 정의

```
public class MyImageView extends View {  
    private Drawable image;  
    private ScaleGestureDetector gestureDetector;  
    private float scale = 1.0f;  
  
    public MyImageView(Context context) {  
        super(context);  
        image = context.getResources().getDrawable(R.drawable.lion);  
        setFocusable(true);  
        image.setBounds(0, 0, image.getIntrinsicWidth(),  
            image.getIntrinsicHeight());  
        gestureDetector = new ScaleGestureDetector(context, new ScaleListener());  
    }  
  
    @Override  
    protected void onDraw(Canvas canvas) {  
        super.onDraw(canvas);  
        canvas.save();  
        canvas.scale(scale, scale);  
        image.draw(canvas);  
        canvas.restore();  
    }  
  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        gestureDetector.onTouchEvent(event);  
        invalidate();  
        return true;  
    }  
}
```

제스처 인식기 객체를 생성
한다.

캔버스에 선축 연산을 적용
한다. 좀 더 자세한 내용은
다음 장을 참조한다.

제스처 인식기의 터치 이벤
트 처리 메소드를 호출해준
다.

정의

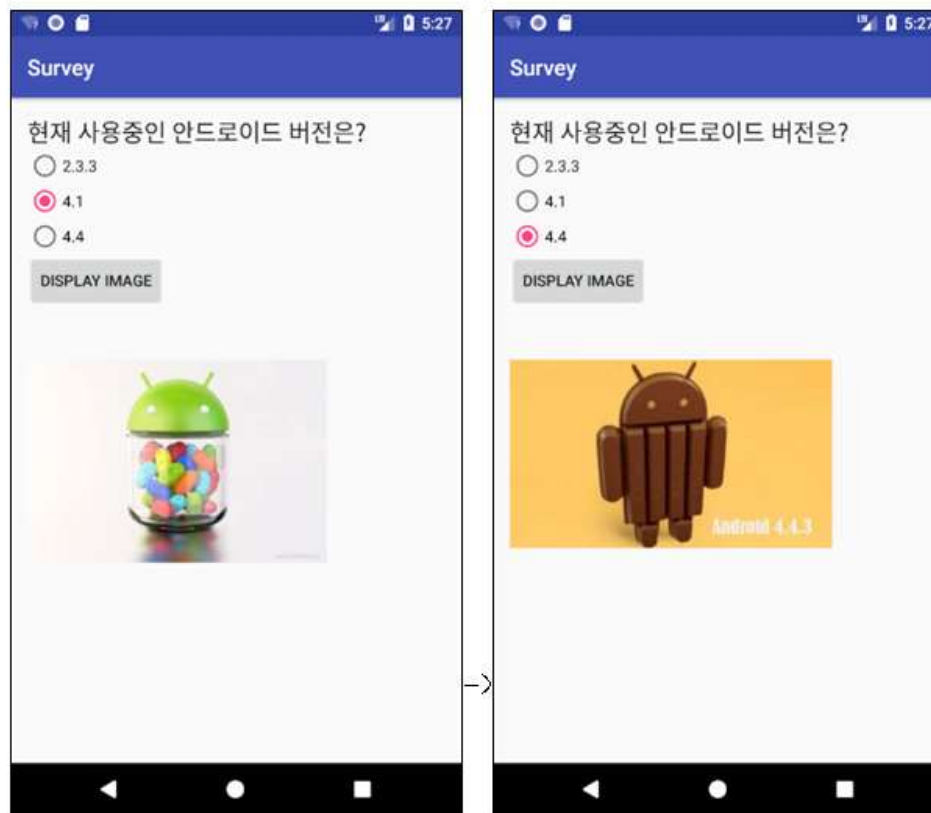
```
private class ScaleListener extends
    ScaleGestureDetector.SimpleOnScaleGestureListener {
    @Override
    public boolean onScale(ScaleGestureDetector detector) {
        scale *= detector.getScaleFactor();

        if (scale < 0.1f)
            scale = 0.1f;
        if (scale > 10.0f)
            scale = 10.0f;

        invalidate();
        return true;
    }
}
```

신축 연산이 감지되었으면 호출된다.

Lab: 여론 조사 앱 작성



이벤트 처리

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

```
public class MainActivity extends AppCompatActivity {  
    RadioGroup group;  
    RadioButton button1, button2, button3;  
    Button button;  
    ImageView image;  
  
    OnClickListener radio_listener = new OnClickListener() {  
        public void onClick(View arg0) {  
            switch (group.getCheckedRadioButtonId()) {  
                case R.id.radio0:  
                    image.setImageResource(R.drawable.image0);  
                    break;  
                case R.id.radio1:  
                    image.setImageResource(R.drawable.image1);  
                    break;  
                case R.id.radio2:  
                    image.setImageResource(R.drawable.image2);  
                    break;  
            }  
        }  
    };  
}
```

이벤트 처리

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

    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    setTitle("Survey");

    group = (RadioGroup) findViewById(R.id.radioGroup1);
    button1 = (RadioButton) findViewById(R.id.radio0);
    button2 = (RadioButton) findViewById(R.id.radio1);
    button3 = (RadioButton) findViewById(R.id.radio2);

    button1.setOnClickListener(radio_listener);
    button2.setOnClickListener(radio_listener);
    button3.setOnClickListener(radio_listener);

    button = (Button) findViewById(R.id.button1);
    image = (ImageView) findViewById(R.id.imageView1);

}
}
```

Lab: 온도 변환기 앱 작성

- 섭씨 온도를 받아서 화씨 온도로 변환하는 앱을 작성



해심 코드

```
public void onClicked(View view) {  
    switch (view.getId()) {  
        case R.id.btn_change:  
            RadioButton celsiusButton = (RadioButton) findViewById(R.id.celsius);  
            RadioButton fahrenheitButton =  
                (RadioButton) findViewById(R.id.fahrenheit);  
            if (text.getText().length() == 0) {  
                Toast.makeText(this, "정확한 값을 입력하십시오.",  
                    Toast.LENGTH_LONG).show();  
                return;  
            }  
  
            float inputValue = Float.parseFloat(text.getText().toString());  
            if (celsiusButton.isChecked()) {  
                text.setText(String  
                    .valueOf(convertFahrenheitToCelsius(inputValue)));  
                celsiusButton.setChecked(false);  
                fahrenheitButton.setChecked(true);  
            } else {  
                text.setText(String  
                    .valueOf(convertCelsiusToFahrenheit(inputValue)));  
                fahrenheitButton.setChecked(false);  
                celsiusButton.setChecked(true);  
            }  
            break;  
        }  
    }  
}
```