

## 모바일 프로그래밍 실습 보고서

학번	201411203	이름	박소영
순서	2	날짜	2018.03.15 목요일
강의자료	01 - 기본위젯		
주제	라디오 버튼, 토글 버튼 이벤트		

### <클릭 이벤트를 처리하는 3가지 방법>

#### 1. xml에서 onClick을 지정해줘서 전용 함수를 만들어준다.

+) OnClickListener : 묵시적인 객체

인터페이스이지만 인터페이스가 가지고 있는 가상 함수들을 abstract 함수들을 구현한 후 객체화 지정한 Button에만 붙는다. onClick 함수를 포함하고 있음.

#### 2. 공용 클래스에 onClick을 재정의한다.

- onClick은 공용이다. 모든 버튼의 클릭 이벤트를 받는다.
- view의 getId() 함수를 통해 누른 객체의 아이디 정보를 받아와 구분한다.

#### 3. 내부 클래스를 만들어서 onClick 함수를 지정해줌.

- RadioButton : 여러가지 옵션 중에서 하나만 선택한다.

- 여러가지의 옵션을 그룹으로 묶어줘야 선택이 된다. 안그러면 각자 논다!

**※ 반드시 RadioGroup으로 묶어줘야 한다!**

- onCheckedChangeListener

클릭이 아닌, 체크 상태가 변동되었는지만 감지함.

파라미터로 체크된 id가 들어온다. -1일 경우 아무것도 선택되지 않은 것.

- onClick만 쓰면 체크된 상태에서 클릭이 된건지, 아닌지 등등의 상태 정보를 알 수가 없다. 리 다양한 Listener를 적절하게 쓰는 것이 중요.

- selector

- 클릭 이벤트가 사용되는 View의 스타일을 미리 지정해 두자.
- drawable에 custom, 사용 가능한지, 선택됐는지 등 상태에 대한 스타일을 적용시킬 수가 있음!
- 실제로 버튼에게 background 속성을 통해 지정해주면 완성!

## Layout : activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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="moblieprogramming.a20180315.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
    </LinearLayout>

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="20dp">
        <RadioButton
            android:id="@+id/radioButton1"
            android:text="IMAGE 1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"/>
        <RadioButton
            android:id="@+id/radioButton2"
            android:text="IMAGE 2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <RadioButton
            android:id="@+id/radioButton3"
            android:text="IMAGE 3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </RadioGroup>

    <ToggleButton
        android:id="@+id/toggleButton"
```

```

        android:textOn=""
        android:textOff=""
        android:background="@drawable/toggle_button_design"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </ToggleButton>

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitCenter"
        android:src="@drawable/img1"/>

</LinearLayout>

```

### Layout : activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<selector
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:state_checked="true"
        android:drawable="@drawable/on"/>
    <item
        android:state_checked="false"
        android:drawable="@drawable/off"/>
</selector>

```

## java 소스코드 : MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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




        init();
    }

    public void init() {
        RadioGroup radioGroup = (RadioGroup) findViewById(R.id.radioGroup);
        // xml에서 만들어놓은 view를 자바 코드에서 끌어옴.
        // 클래스의 각종 함수를 사용할 수 있게 된다.
        radioGroup.setOnCheckedChangeListener(
            new RadioGroup.OnCheckedChangeListener() {
                // OnCheckedChangeListener를 묵시적으로 객체화하는 방법

                @Override
                public void onCheckedChanged(RadioGroup group,
                    @IdRes int checkedId) {
                    if (checkedId != -1) {
                        ImageView imageView = (ImageView)
                            findViewById(R.id.imageView);

                        switch (checkedId) {
                            case R.id.radioButton1:
                                imageView.setImageResource(R.drawable.img1);
                                // imageView 객체에 이미지를 배치함.
                                break;
                            case R.id.radioButton2:
                                imageView.setImageResource(R.drawable.img2);
                                break;
                            case R.id.radioButton3:
                                imageView.setImageResource(R.drawable.img3);
                                break;
                        }
                    }
                }
            }
        ));
    }
}
```

실행 결과

<div><div>20180315</div><div><div><input checked="" type="radio"/> IMAGE 1</div><div><input type="radio"/> IMAGE 2</div><div><input type="radio"/> IMAGE 3</div></div><div><div>OFF</div></div><div></div></div>	<div><div>20180315</div><div><div><input type="radio"/> IMAGE 1</div><div><input checked="" type="radio"/> IMAGE 2</div><div><input type="radio"/> IMAGE 3</div></div><div><div>OFF</div></div><div></div></div>	<div><div>20180315</div><div><div><input type="radio"/> IMAGE 1</div><div><input checked="" type="radio"/> IMAGE 2</div><div><input type="radio"/> IMAGE 3</div></div><div><div>ON</div></div><div></div></div>
---	--	---