

# 2017 LINC+ 사업단 사회맞춤형 JAVA 실무인력양성과정 안드로이드 앱 개발 기초

컴퓨터공학부  
강승우

# 3. 이벤트 처리

2018.02.05

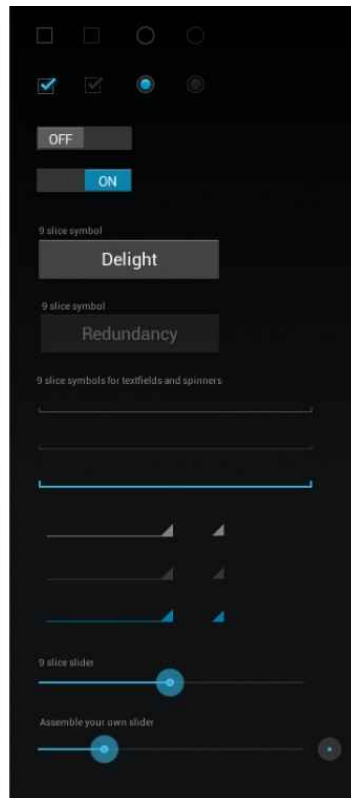
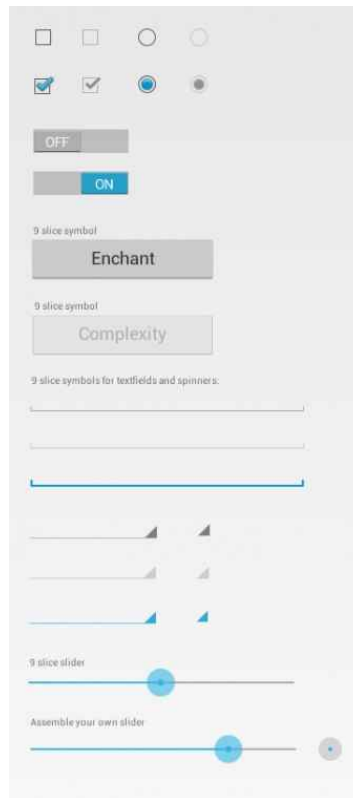
# Topic

- 입력 위젯
- 사용자 입력 처리 방식
  - Event-driven approach
- 이벤트 처리 방법
  - XML 파일에 이벤트 처리 메소드를 등록하는 방법
  - 이벤트를 처리하는 객체를 생성하는 방법
- 다양한 위젯 사례

## 3-1 입력 위젯 / 사용자 입력 처리 방식

# 입력 위젯

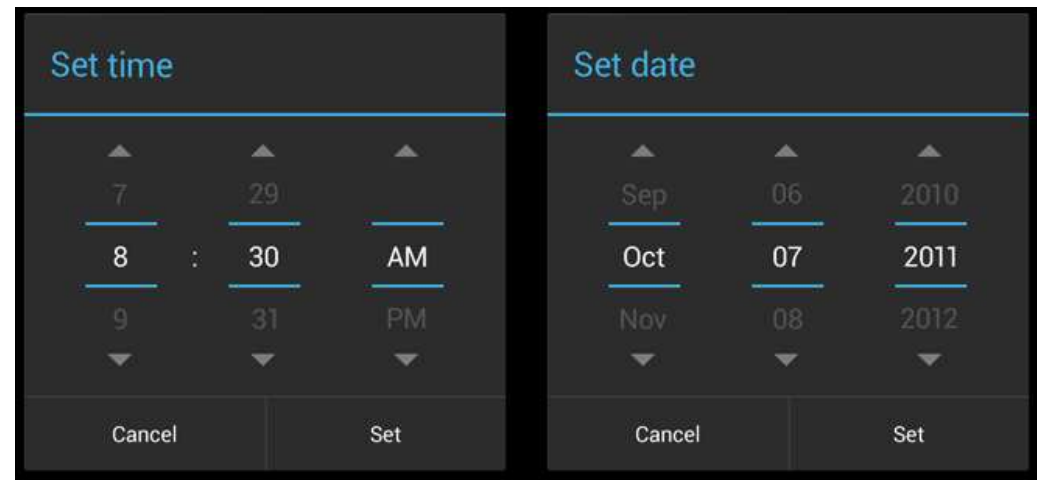
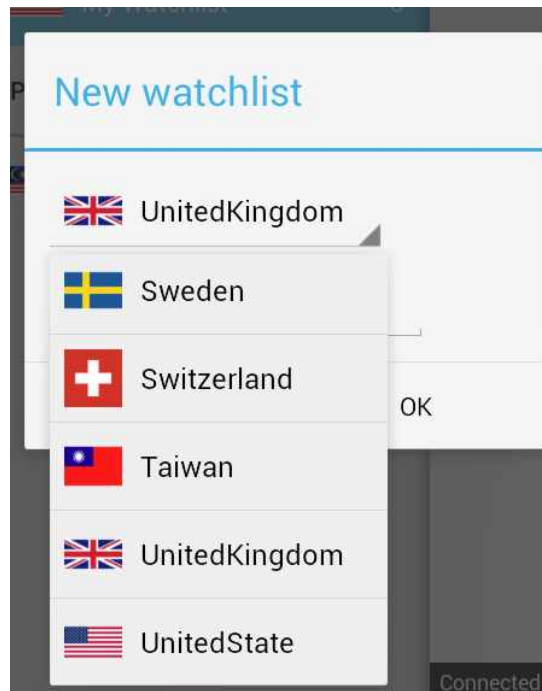
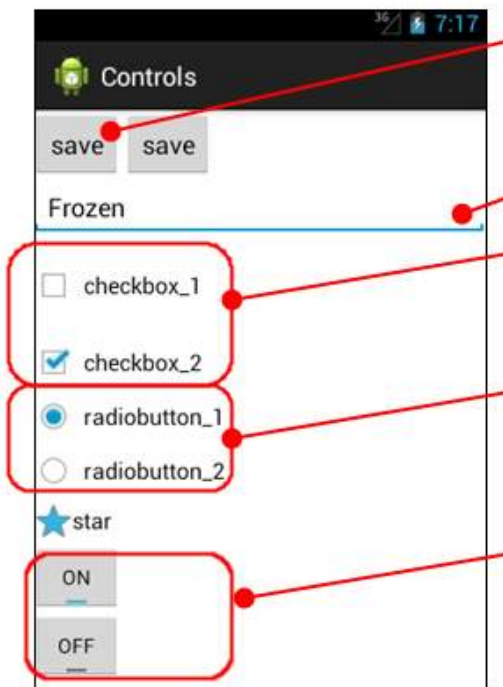
- 사용자 인터페이스에서 사용자의 입력을 받는 component



# 입력 위젯

- 종류

- Button / EditText / CheckBox / RadioButton / ToggleButton
- Spinner / DatePicker/TimePicker



# 사용자 입력 처리 방식



- Polling: 애플리케이션에서 무한 루프를 돌면서 입력을 기다리다가 입력이 들어오면 처리
- Event-driven: 애플리케이션은 이벤트 리스너를 등록해 놓고 다른 작업을 하다가 입력이 들어오면 이벤트가 발생하여 해당 이벤트 리스너가 동작하여 처리

## 3-2 이벤트 처리 방법



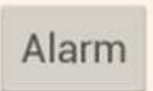
# 이벤트 처리 방법

- XML 파일에 이벤트 처리 메소드를 등록하는 방법
  - 가장 쉬운 방법
- 이벤트 처리 객체를 생성하여 입력 위젯 객체에 등록하는 방법
  - 일반적인 방법

# Button 이벤트 처리

- 버튼
  - 텍스트 버튼
  - 이미지 버튼
  - 텍스트+이미지 버튼

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_text"  
    ... />
```



```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/button_icon"  
    ... />
```

이미지 파일  
어름



```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_text"  
    android:drawableLeft="@drawable/button_icon"  
    ... />
```

이미지 파일  
어름



# 1) XML 파일에 이벤트 처리 메소드 등록

→ 레이아웃 XML에 정의된 <Button> 요소에 onClick 속성을 추가

```
<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" tools:context=".MainActivity">
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Press a button"
    />
```

```
</RelativeLayout>
```

예제 프로젝트 이름:  
02\_Event\_Handling/Ch6ButtonEvent

# 이벤트 처리 메소드 코드

- 자바 코드에 onClick 메소드 구현

@Override

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

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

```
//
```

```
// 이벤트가 발생했을 때 실행할 코드를 작성한다.
```

```
//
```

```
// 이 예제에서는 토스트 메시지를 출력한다.
```

```
Toast.makeText(getApplicationContext(), "Button pressed!", Toast.LENGTH_SHORT).show();
```

```
}
```

## 2) 이벤트 처리 객체를 이용하여 처리

- 이벤트를 처리하는 객체를 생성하여 입력 위젯에 등록
  - 이벤트 처리 객체는 이벤트 처리 메소드를 가지고 있음
- 이벤트 처리 객체
  - 이벤트 리스너(event listener)를 구현한 클래스의 객체임
  - 이벤트 리스너에는 이벤트 처리 메소드(callback method)가 정의되어 있음

# 이벤트 리스너와 이벤트 처리 객체 예제

```
class MyClass
```

```
{
```

```
    class Listener implements OnClickListener {
```

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

```
            ...
```

```
        }
```

```
    }
```

```
    ...
```

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

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

```
    ...
```

```
}
```

이벤트 리스너(OnClickListener라는 인터페이스)를 구현한 클래스 정의  
- 내부 클래스(Inner class)로 정의됨

이벤트 리스너 객체 생성

이벤트 리스너 객체를 위젯에 등록

# 이벤트 리스너 종류

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

[http://developer.android.com/  
reference/android/view/View.  
OnClickListener.html](http://developer.android.com/reference/android/view/View.OnClickListener.html)

# View.OnClickListener

Added in API level 1  
Summary: Methods | [\[Expand All\]](#)

```
public static interface View.OnClickListener
```

```
android.view.View.OnClickListener
```

▼ Known Indirect Subclasses

[CharacterPickerDialog](#), [KeyboardView](#), [LeanbackListPreferenceDialogFragment.ViewHolder](#), [QuickContactBadge](#), [SearchOrbView](#), [SpeechOrbView](#)

Interface definition for a callback to be invoked when a view is clicked.

## Summary

### Public methods

abstract	<a href="#">onClick(View v)</a>
void	Called when a view has been clicked.

## Public methods

### onClick

Added in API level 1

```
void onClick (View v)
```

Called when a view has been clicked.

### Parameters

v	<b>View:</b> The view that was clicked.
---	---



# 이벤트 리스너 객체 생성 방법

1. 리스너 클래스를 내부 클래스(inner class)로 정의
  - 이벤트 리스너를 구현하는 내부 클래스
2. 리스너 클래스를 무명(익명) 클래스(anonymous class)로 정의
3. 리스너 인터페이스를 액티비티 클래스에서 구현

# 내부 클래스로 정의

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

내부 클래스를 정의하여 이벤트 리스너 인터페이스를 구현

```
class MyListenerClass implements View.OnClickListener {  
    public void onClick(View view) {  
        Toast.makeText(getApplicationContext(), "Button pressed!", Toast.LENGTH_SHORT).show();  
    }  
}
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Button button = (Button) findViewById(R.id.button);  
    MyListenerClass buttonListener = new MyListenerClass();  
    button.setOnClickListener(buttonListener);  
}
```

이벤트 리스너 클래스의 객체를 생성하고 이를 Button 객체에 등록

예제 프로젝트 이름:  
02\_Event\_Handling/Ch6ButtonEventInner

# 무명/익명 클래스로 정의

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

```
    @Override
```

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

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

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

```
        Button button = (Button) findViewById(R.id.button);
```

```
        button.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

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

```
                Toast.makeText(getApplicationContext(), "Button pressed!", Toast.LENGTH_SHORT).show();
```

```
            }
```

```
        });
```

```
    }
```

Anonymous class

- 클래스 몸체는 정의되지만 이름이 없는 클래스
- 클래스를 정의하면서 동시에 객체를 생성
- 한번만 사용 가능

```
new ParentClass() { // 클래스 몸체 }
```

예제 프로젝트 이름:

02\_Event\_Handling/Ch6ButtonEventAnony

# 이벤트 리스너 인터페이스 구현

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

```
    @Override
```

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

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

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

```
        Button button = (Button) findViewById(R.id.button);
```

```
        button.setOnClickListener(this);
```

```
    }
```

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

```
        Toast.makeText(getApplicationContext(), "Button pressed!", Toast.LENGTH_SHORT).show();
```

```
    }
```

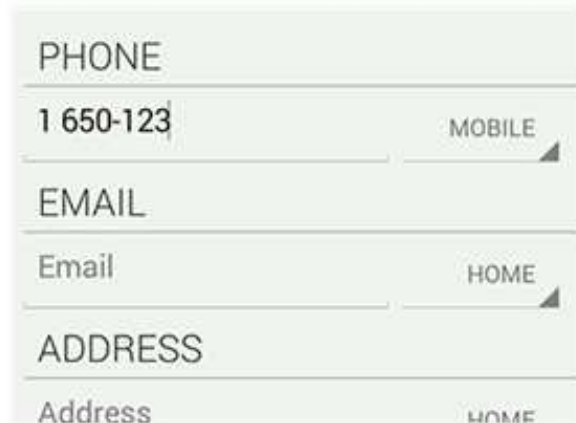
예제 프로젝트 이름:

02\_Event\_Handling/Ch6ButtonEventInterface

## 3-3 다른 위젯 사례

# 텍스트 필드 (Text field)

- 사용자가 앱에 텍스트를 타이핑하여 입력할 수 있음
  - 단일 라인 / 멀티 라인
  - 텍스트 필드를 터치하면 커서가 나타나고 키보드가 화면의 하단에 표시



PHONE

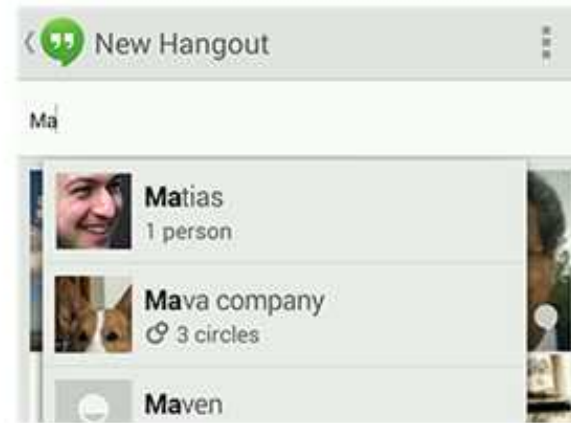
1 650-123 MOBILE

EMAIL

Email HOME

ADDRESS

Address HOME



✓레이아웃 파일에 <EditText> 요소 추가

# EditText 입력 설정

- 키보드 종류 지정
  - 입력의 종류에 따라 키보드 형태를 다르게 할 수 있음
  - 숫자, 이메일 주소, 비밀번호
  - android:inputType 속성
- 키보드 제어
  - 대소문자, 자동 수정, 멀티 라인 등
  - android:inputType 속성
- 키보드 동작 지정
  - 입력을 완료했을 때 수행할 동작을 지정할 수 있음
  - Return 키 위치에 나타나는 버튼과 액션 (Search, Send 등)
  - android:imeOptions 속성

- 키보드 종류 지정 android:inputType 속성의 값 종류
  - “text” : 일반적인 텍스트 입력 키보드
  - “textEmailAddress” : @ 문자를 이용한 이메일 주소 입력 키보드
  - “textUri” : / 문자를 이용한 URL 주소 입력 키보드
  - “number” : 숫자를 입력하는 키패드
  - “phone” : 전화번호를 입력하는 키보드



(textEmailAddress 입력 타입)



(phone 입력 타입)



- 키보드 제어 android:inputType 속성의 값 종류
  - “textCapSentences” : 각 문장의 첫 번째 글자를 대문자로 한다
  - “textCapWords” : 각 단어의 첫 번째 글자를 대문자로 한다
  - “textAutoCorrect” : 자동으로 잘못된 단어를 수정한다
  - “textPassword” : 입력되는 글자를 점으로 표시한다
  - “textMultiLine” : 멀티 라인으로 입력 받는다
- 여러 값을 조합해서 지정 가능

<EditText

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/postal\_address"

android:hint="postal address"

android:inputType="textPostalAddress|textCapWords|textNoSuggestions"

/>

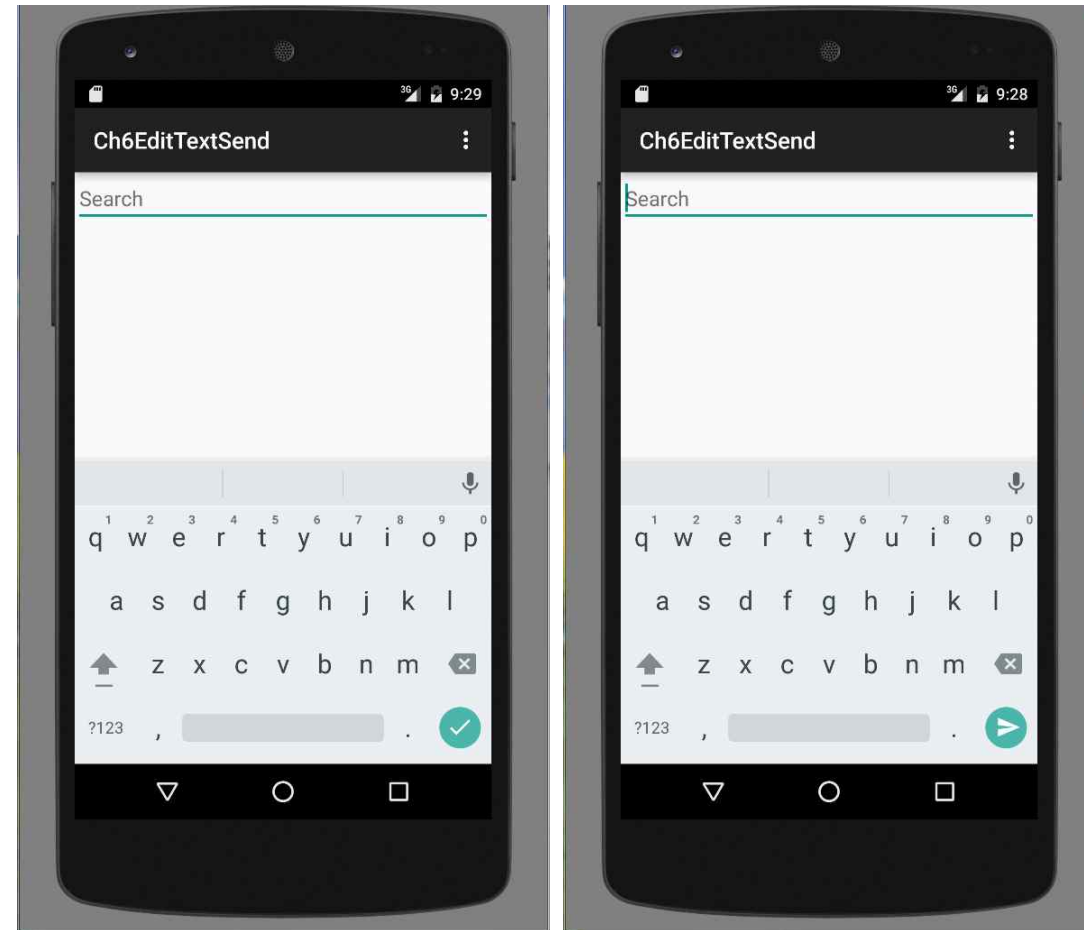
- 키보드 동작 지정  
android:imeOptions 속성의 값

- [http://developer.android.com/reference/android/widget/TextView.html#attr\\_android:imeOptions](http://developer.android.com/reference/android/widget/TextView.html#attr_android:imeOptions)

- “actionSend” : 키보드에 “send” 버튼 생성

<EditText

```
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:id="@+id/search"  
android:hint="Search"  
android:imeOptions="actionSend"  
android:inputType="text"  
>
```



# EditText 이벤트 처리

예제 프로젝트 이름:  
02\_Event\_Handling/Ch6EditTextSend

**@Override**

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

```
    final EditText editText = (EditText) findViewById(R.id.search);
```

```
    editText.setOnEditorActionListener(new TextView.OnEditorActionListener() {
```

**@Override**

```
    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
```

```
        boolean handled = false;
```

```
        if (actionId == EditorInfo.IME_ACTION_SEND) {
```

```
            Toast.makeText(getApplicationContext(), editText.getText(), Toast.LENGTH_SHORT).show();
```

```
            handled = true;
```

```
        }
```

```
        return handled;
```

```
    }
```

```
});
```

```
}
```

\* Send 버튼이 눌렸을 때 수행할 루틴을 작성한다. 예를 들면 메시지 전송을 위한 sendMessage() 함수 호출  
\* 여기서는 간단히 toast 메시지 표시하는 예제를 넣은 것임

<http://developer.android.com/reference/android/widget/TextView.OnEditorActionListener.html>

# imeOptions 속성 값 예제

- android:imeOptions="actionGo"
  - 이동 (예: 웹 브라우저에서 URL 입력 후 해당 주소로 이동할 때)
  - EditorInfo.IME\_ACTION\_GO
- android:imeOptions="actionSearch"
  - 검색 (예: 검색 엔진에서 키워드 입력 후 검색을 수행하고자 할 때)
  - EditorInfo.IME\_ACTION\_SEARCH
- android:imeOptions="actionNext"
  - 다음 (예: 입력 필드가 여러 개 있는 경우 다음 항목으로 이동할 때)
  - EditorInfo.IME\_ACTION\_NEXT
- android:imeOptions="actionPrevious"
  - 이전 (예: 입력 필드의 이전 항목으로 이동할 때)
  - EditorInfo.IME\_ACTION\_PREVIOUS

# CheckBox

<http://developer.android.com/reference/android/widget/CheckBox.html>

- 하나의 그룹 안에서 여러 개의 항목을 동시에 선택할 때 사용하는 위젯
  - 보통 수직 리스트에 체크 박스를 부착하여 사용
- 레이아웃에서 <CheckBox> 요소 추가
  - 각각의 체크 박스는 별도로 취급
  - 따라서 각 체크 박스 별로 클릭 리스너를 등록
  - 쉬운 방법은 XML에서 onClick 속성 이용

```
<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">
```

```
<CheckBox
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/checkbox_meat"
    android:text="Meat"
    android:onClick="onCheckboxClicked"
/>
```

```
<CheckBox
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/checkbox_cheese"
    android:text="Cheese"
    android:onClick="onCheckboxClicked"
/>
```

```
</LinearLayout>
```

예제 프로젝트 이름:  
02\_Event\_Handling/Ch6CheckBoxEvent

# CheckBox 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(), "Meat checked", Toast.LENGTH_SHORT).show();  
            else  
                Toast.makeText(getApplicationContext(), "Meat not checked", Toast.LENGTH_SHORT).show();  
            break;  
        case R.id.checkbox_cheese:  
            if(checked)  
                Toast.makeText(getApplicationContext(), "Cheese checked", Toast.LENGTH_SHORT).show();  
            else  
                Toast.makeText(getApplicationContext(), "Cheese not checked", Toast.LENGTH_SHORT).show();  
            break;  
    }  
}
```

# RadioButton

<http://developer.android.com/reference/android/widget/RadioButton.html>

- 하나의 그룹 안에서 하나의 버튼만 선택 가능
  - 하나의 라디오 버튼을 클릭하면, 다른 버튼은 자동으로 선택 해제
- 두 개의 클래스 사용
  - RadioButton
  - RadioGroup
    - 라디오 버튼들을 그룹핑 하는 데 사용
    - 하나의 라디오 그룹에 여러 개의 라디오 버튼이 들어갈 수 있고 이 중에 하나만 선택됨



```
<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">

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

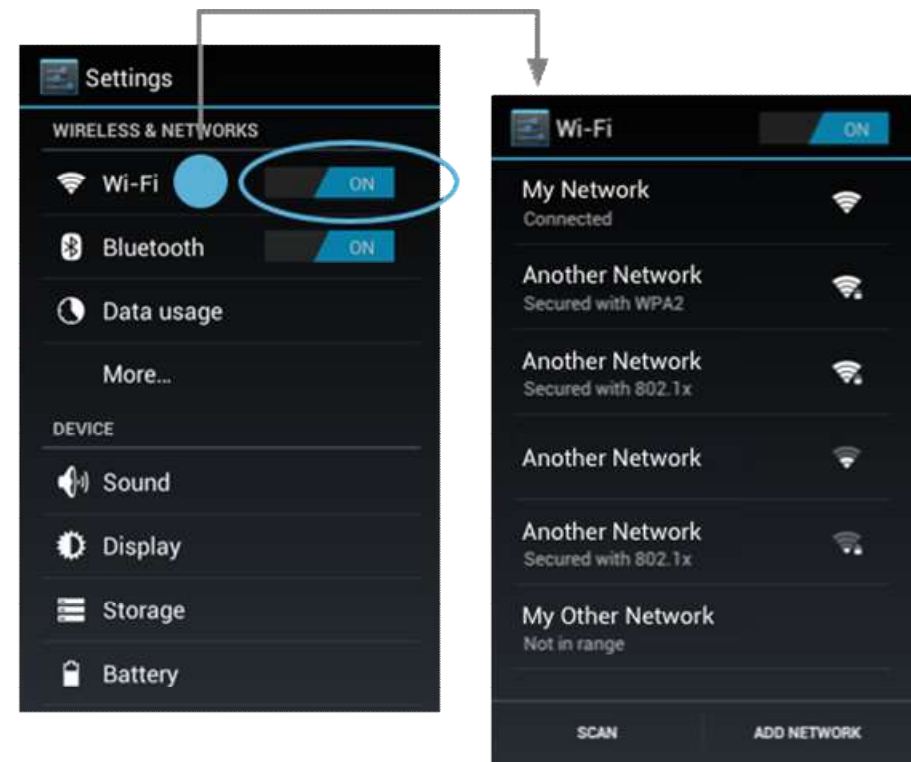
예제 프로젝트 이름:  
02\_Event\_Handling/Ch6RadioButton

# RadioButton 이벤트 처리

```
public void onRadioButtonClicked(View view) {  
    boolean checked = ((RadioButton) view).isChecked();  
  
    switch(view.getId()) {  
        case R.id.radio_red:  
            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;  
    }  
}
```

# ToggleButton

- On, Off 두 가지 상태 중의 하나로 변경되도록 만들어진 버튼
- 이벤트 처리는 CheckBox, RadioButton 등과 유사



# ToggleButton 이벤트 처리

```
ToggleButton toggle = (ToggleButton) findViewById(R.id.togglebutton);
toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            // The toggle is enabled
        } else {
            // The toggle is disabled
        }
    }
});
```

<https://developer.android.com/guide/topics/ui/controls/togglebutton.html>

<https://developer.android.com/reference/android/widget/ToggleButton.html>