

모바일 프로그래밍

03 입력 위젯과 이벤트 처리 2

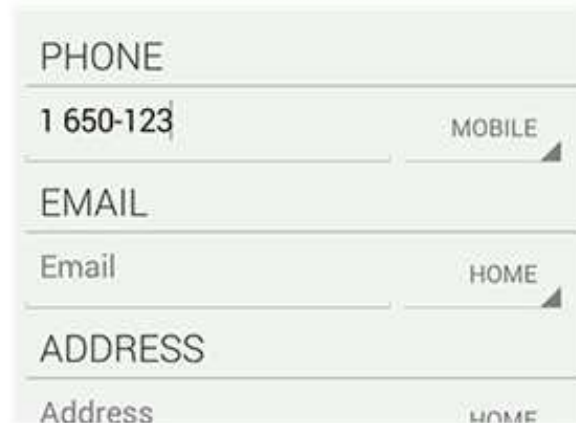
2017 2학기

강승우

03-3 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

- 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"

/>

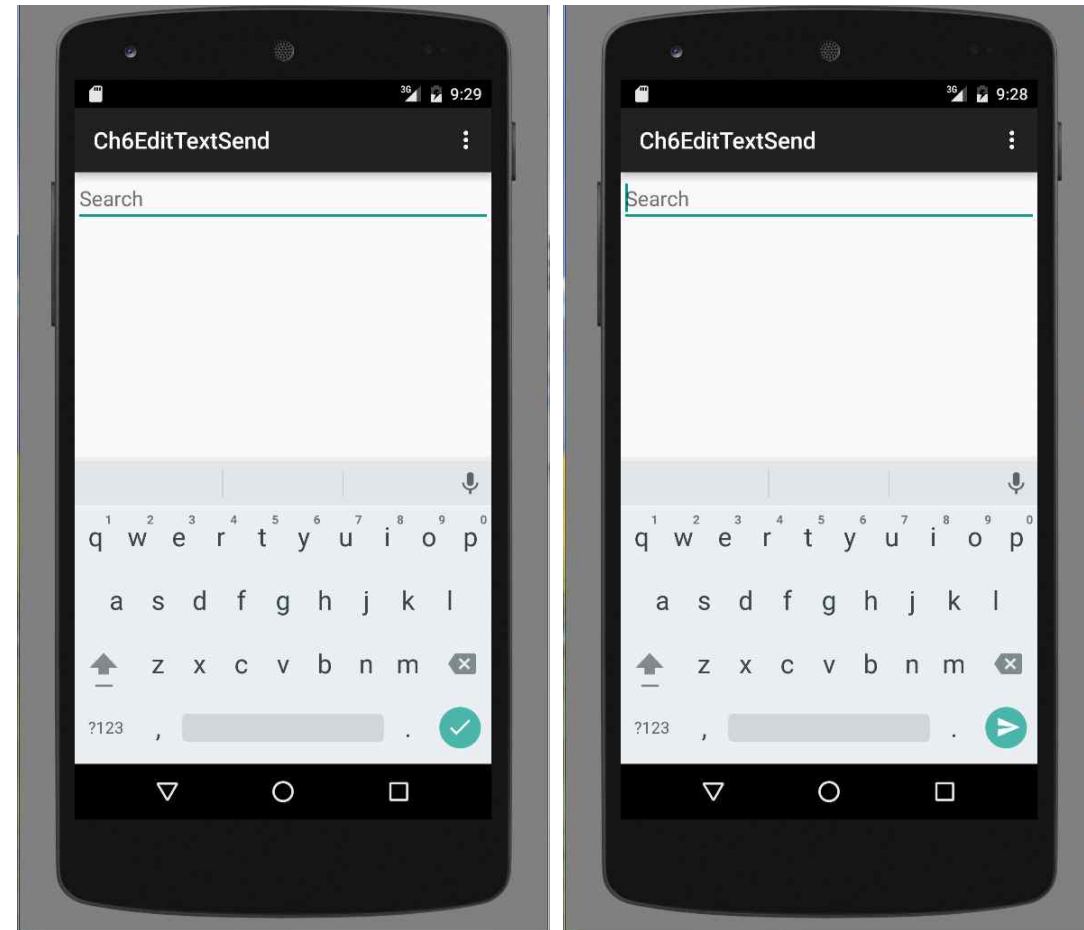
- 키보드 동작 지정
android:imeOptions 속성의 값

- http://developer.android.com/reference/android/widget/TextView.html#attr_android:imeOptions

- “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:
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:
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>
```

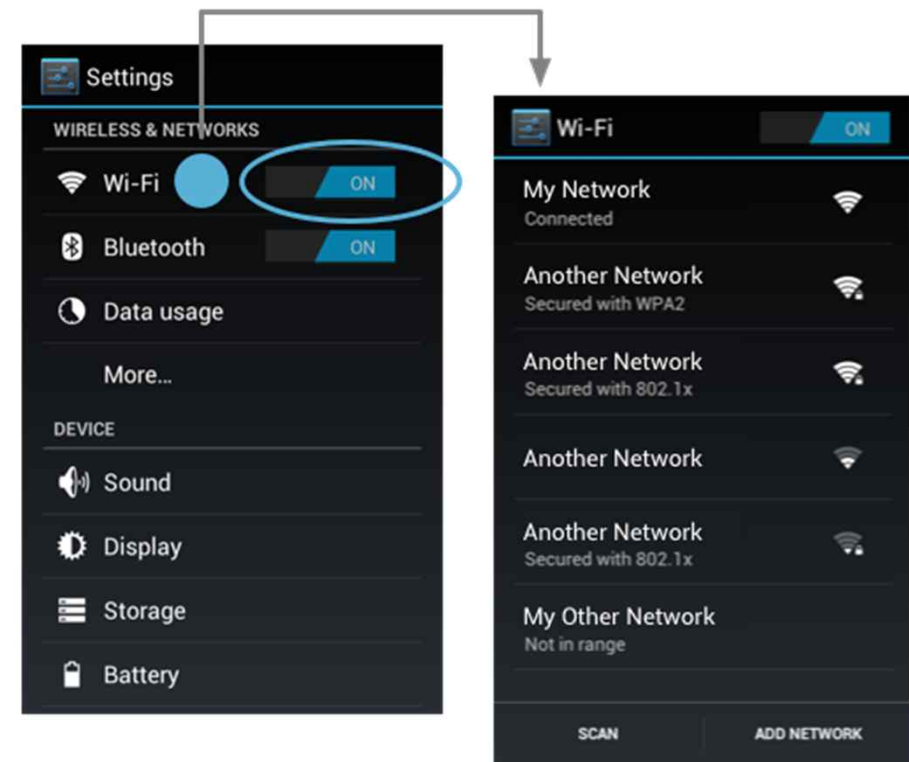
예제 프로젝트 이름:
Ch6RadioButton

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
        }
    }
});
```

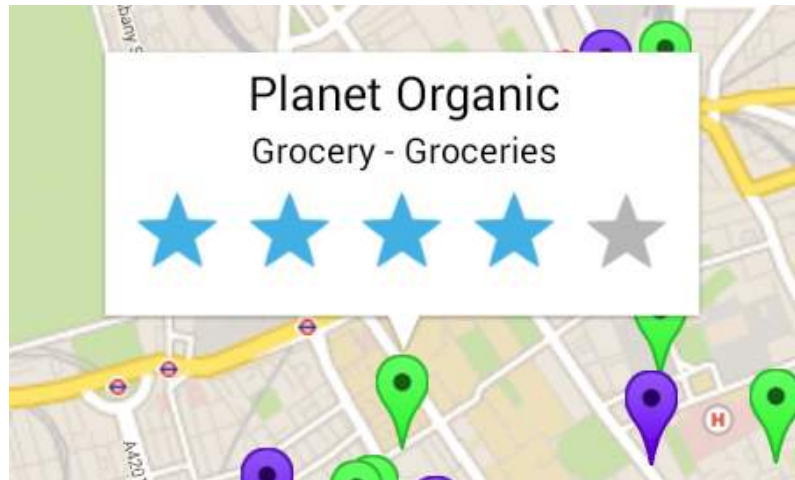
<https://developer.android.com/guide/topics/ui/controls/togglebutton.html>

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

RatingBar

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

- 별점 매기는 인터페이스를 만드는데 사용할 수 있는 위젯
 - 예: 매장, 맛집, 영화 평가



```
<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">

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

</LinearLayout>
```

예제 프로젝트 이름:
Ch6RatingBar

RatingBar 이벤트 처리

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

```
    final RatingBar ratingBar= (RatingBar) findViewById(R.id.ratingbar);
    ratingBar.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
        @Override
        public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
            Toast.makeText(getApplicationContext(), "New Rating: " + rating, Toast.LENGTH_SHORT).show();
        }
    });
}
```

