

모바일 프로그래밍

03 입력 위젯과 이벤트 처리 1

2017 2학기

강승우

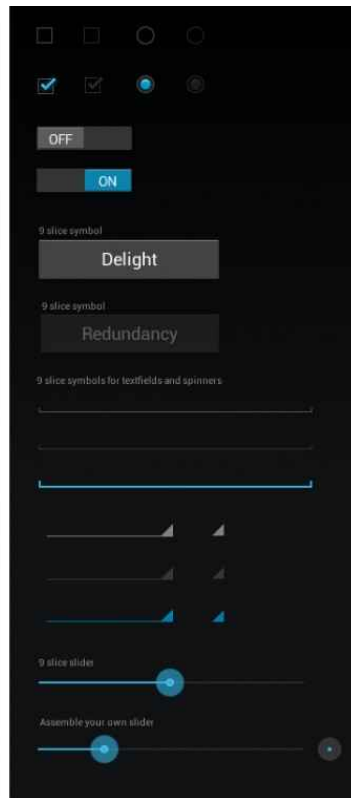
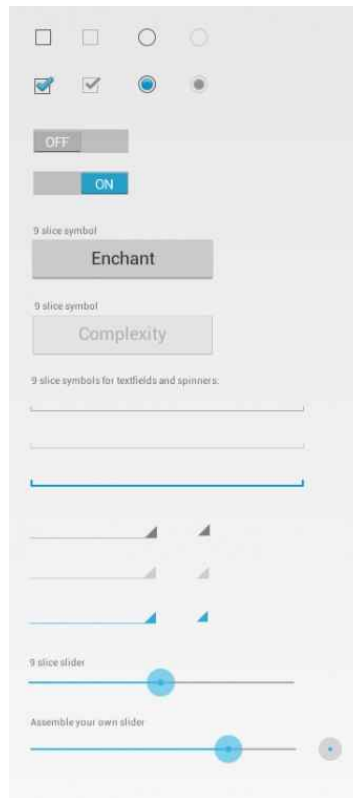
Topic

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

03-1 입력 위젯 / 사용자 입력 처리

입력 위젯

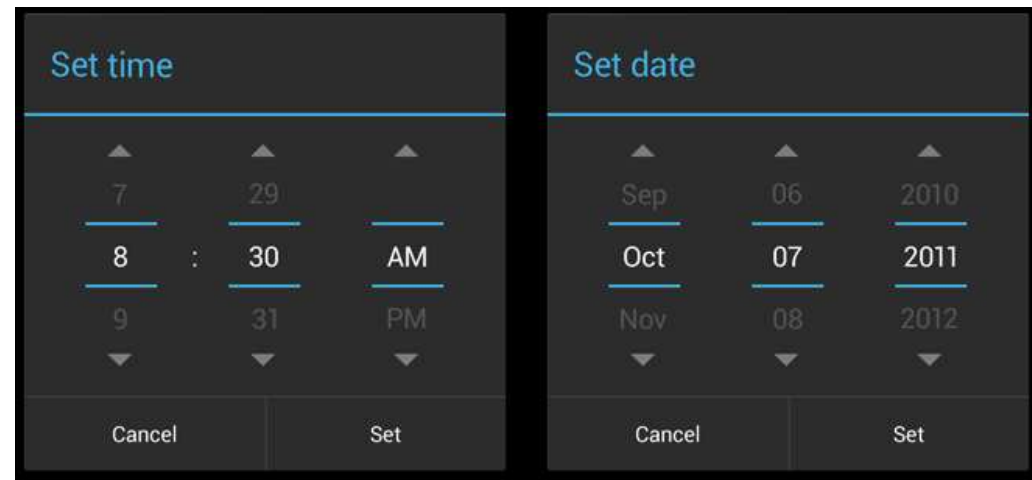
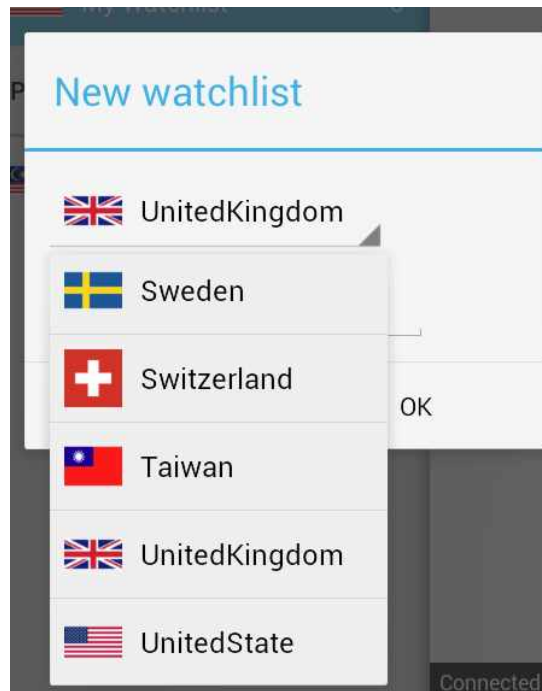
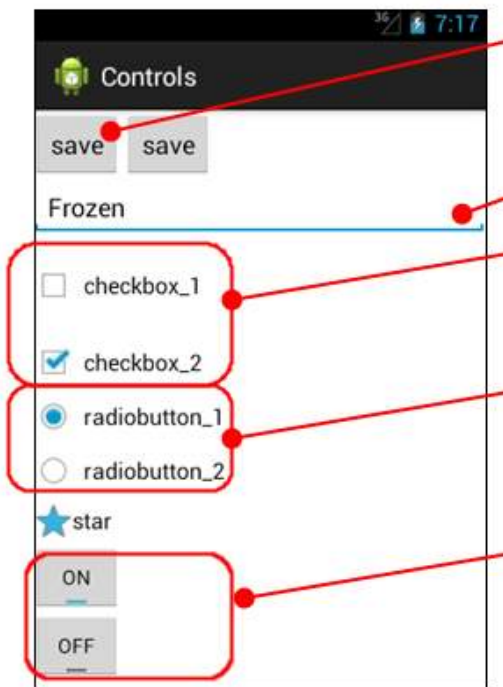
- 사용자 인터페이스에서 사용자의 입력을 받는 component



입력 위젯

- 종류

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



사용자 입력 처리



- 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

03-2 이벤트 처리 방법

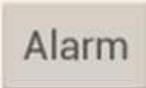
이벤트 처리 방법 3가지

- XML 파일에 이벤트 처리 메소드를 등록하는 방법
 - 가장 쉬운 방법
- 이벤트 처리 객체를 생성하여 컴포넌트(위젯 객체)에 등록
 - 일반적인 방법
- 뷰 클래스의 이벤트 처리 메소드를 재정의
 - 커스텀 뷰를 작성하는 경우: (예) 게임


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
어플

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



이미지 파일
어플

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:
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 위젯에 등록
 - 이벤트 처리 객체는 이벤트 처리 메소드를 가지고 있음
- 이벤트 처리 객체
 - 이벤트 리스너(event listener)를 구현한 클래스의 객체
 - 이벤트 리스너에는 이벤트 처리 메소드(callback method)가 정의되어 있음

구조

```
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
- 내부 클래스(inner class)로 정의됨

이벤트 리스너 객체 생성

이벤트 리스너 객체를 위젯에 등록

이벤트 리스너

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	onClick(View v)
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	View: 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 클래스에 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:
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:
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:
Ch6ButtonEventInface