

모바일 프로그래밍 실습 보고서

학번	201411203	이름	박소영
순서	3	날짜	2018.03.20 화요일
강의자료	01 - 기본 위젯		
주제	사용자에게 입력 받는 EditText		

1. EditText : 사용자에게 입력을 받을 수 있는 위젯

- 이벤트 핸들링 : 입력을 받자마자 이벤트를 발생시키고, 해당 이벤트가 발생하면 버튼을 활성화하거나... 이렇게 처리해줄 수 있다.
- 값을 가지고 오는 것이 주 기능인데, getText()으로 가져온 후 toString()으로 변환해줘야 함.

2. autoCompleteText : 자동완성

- 혼자 쓰일 수 없는, AdapterView이다. 데이터 셋이 필요하다. 어디서 찾아줄 건지를 찾아줘야 한다. 또, 뷰랑 연결해줄 수 있는 Adapter가 필요하고, 마지막으로 View도 있어야 한다.
- 어댑터에는 Watcher 객체를 붙여줘야 한다.
- TextWatcher 3개의 함수가 오버라이딩 된다. 인터페이스이기 때문에 3개의 함수를 다 구현해줘야 한다.

3. MultiAutoCompleteTextView

- MultiAutoCompleteTextView.CommaTokenizer();
- 콤마를 토큰라이저 (토큰을 끊는 단위)로 사용하겠다는 뜻.

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line, COUNTRIES);
AutoCompleteTextView textView = (AutoCompleteTextView)
findViewById(R.id.edit);
textView.setAdapter(adapter);
```

▲simple 이 view를 쓰겠다. 이 레이아웃에 보여주겠다! COUNTRIES에서 데이터를 찾겠다는 뜻.

< Layout >

- gravity : 내 안에 있는 것들에 대한 것
- layout_gravity : 내 자신의 위치
- 텍스트뷰에서 글자가 가운데로 갔으면 한다면 gravity를 써야 한다. (내 컨텐츠니까)

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.a20180320.MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btn"
        android:onClick="btnClick"
        android:text="입력"
        android:enabled="false"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <AutoCompleteTextView
        android:id="@+id/autoEdit"
        android:completionThreshold="1"
        android:completionHint="음식 이름을 입력하세요."
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

java 소스코드 : MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText editText;
    AutoCompleteTextView autoTextView;
    Button button;
    ArrayList<String> arr;
    // 스트링 데이터가 들어가 있는 배열과 뷰를 연결시켜주는 어댑터 생성
    ArrayAdapter<String> adapter; // <String> : 배열에 들어가 있는 데이터의
    자료형

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

    public void init() {
        autoTextView = (AutoCompleteTextView)findViewById(R.id.autoEdit);
        arr = new ArrayList<>();
        adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_dropdown_item_1line, arr);
        // 첫 번째 인자 : context.this가 안될 경우 getApplicationContext();
        // 두 번째 인자 : 한 칸에 대한 레이아웃을 어떻게 사용할 것인지.
        // 안드로이드에 등록된 레이아웃을 사용했다.
        // 세 번째 인자 : 연결 시켜줄 데이터셋

        autoTextView.setAdapter(adapter);
        // 데이터 정보를 가지고 있는 어댑터를 AutoCompleteTextView에 적용시켜준다.

        editText = (EditText)findViewById(R.id.editText);
        button = (Button)findViewById(R.id.btn);

        // 리스너 객체를 묵시적으로 생성해서 쓴다.
        editText.addTextChangedListener(new TextWatcher() {
            public void beforeTextChanged(
                CharSequence s, int start, int count, int after) {}
            public void onTextChanged(
                CharSequence s, int start, int before, int count) {}

            public void afterTextChanged(Editable s) {
                String str = s.toString();
            }
        });
    }
}
```

```

        if (str.length() > 0) {
            button.setEnabled(true);
        } else {
            button.setEnabled(false);
        }
    }
});
}

public void btnClick(View view) {
    // 데이터는 버튼을 클릭했을 때 저장한다.
    String str = editText.getText().toString();
    adapter.add(str);
    // string은 adapter에 add한다. 그러면 어댑터에 데이터가 추가되는 것.
    // 추가된 데이터는 어댑터가 가지고 있는 저장 구조 (여기서는 arr 배열)에
    저장된다.
    adapter.notifyDataSetChanged();
    // 데이터값 변경되었다는 것을 체크해야 뷰 갱신이 일어난다.
    editText.setText("");
    // 초기화
}
}

```

실행 결과

