

모바일 프로그래밍 실습 보고서

학번	201411203	이름	박소영
순서	5	날짜	2018.03.27 화요일
강의자료	03 - 기본위젯 2		
주제	AdapterView 사용하기		

1. AdapterView : 배열이나 파일, DB에 저장된 데이터를 화면에 보여줄 때 좋음.

- 각 칸도 view처럼 볼 수 있다.
- ListView, Gallery, Spinner, GridView 등등 다양하게 파생된다.

2. ListView : 항목을 수직으로 보여주는 AdapterView

- 스크롤 기능을 지원함.
- 구성 요소
 - **ListView**
 - **Data** : Static, Dynamic 종류는 다 상관 없고, Adapter를 잘 연결해주면 된다!
 - **Adapter** : 데이터와 리스트 뷰를 연결시켜 줌.
 - **OnItemClickListener** : 항목을 선택할 때 처리해 주는 이벤트를 붙여준다.
클릭한 거 자체가 목적이면 onClick을 달면 되지만, **뭘 클릭했는지 알 수 없다.**

3. ListView의 Adapter에서 사용하는 레이아웃의 종류

- 안드로이드에서 이미 지정한 레이아웃이 있음.
- 선택 모드는 ListView에서도 setChoiceMode를 이용해 따로 바꿔줘야 한다.
 - **simple_list_item_1** : 하나의 텍스트뷰로 구성.
 - **simple_list_item_2** : 두개의 텍스트뷰로 구성. (두 줄로 표현)
 - **simple_list_single_choice** : 여러 항목 중 한 개만 선택 가능. 라디오 버튼 생성.
 - **simple_list_multiple_choice** : 여러개 선택 가능. 체크박스 생성
- SimpleAdapter : Map (key, value 쌍) 형태의 데이터를 XML 리소스에 보여주기 위함.
- 가져올 때는 Map으로 가져오기 때문에, 한 번 더 get 함수를 통해 value를 끌어와야 함.
- **getItemAtPosition** 함수를 이용해 특정 index에 위치한 값을 끌어올 수도 있음.

4. ArrayAdapter : 데이터 정보가 배열일 경우에 사용

- 레이아웃 정보와 데이터만 주면 된다!
- key를 어느 필드에 매핑시켜줄 건지 알려주지 않아도 됨!

Layout : activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
    tools:context="moblieprogramming.a20180327.MainActivity">

    <ListView
        android:id="@+id/listView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        tools:layout_constraintTop_creator="1"
        tools:layout_constraintRight_creator="1"
        tools:layout_constraintBottom_creator="1"
        android:layout_marginStart="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginEnd="8dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginTop="8dp"
        tools:layout_constraintLeft_creator="1"
        android:layout_marginBottom="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginLeft="8dp"
        app:layout_constraintVertical_bias="0.0"
        android:layout_marginRight="8dp" />
</android.support.constraint.ConstraintLayout>
<!--android:entries="@array/gamer"-->
```

java 소스코드 : MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    ListView listView;
    ArrayList<HashMap<String, String>> array;
    HashMap<String, String> words;
    ArrayList<String> keySet; // 단어 keySet을 가져와서 배열에 넣어줌
    SimpleAdapter adapter;
    ArrayAdapter<String> arrayAdapter;

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

        init();
    }

    public void init() {
        listView = (ListView) findViewById(R.id.listView);

        // 1. Adapter 이용하기
        array = new ArrayList<>();
        HashMap<String, String> map = new HashMap<>();
        map.put("item1", "FAKER");
        map.put("item2", "SKT T1");
        array.add(map);

        map = new HashMap<>();
        map.put("item1", "BENGI");
        map.put("item2", "SKT T1");
        array.add(map);

        map = new HashMap<>();
        map.put("item1", "SMEB");
        map.put("item2", "KT Rolster");
        array.add(map);
        adapter = new SimpleAdapter(this, array,
            android.R.layout.simple_list_item_2, // 레이아웃 지정
            new String[]{"item1", "item2"}, // key 값 지정
            new int[]{android.R.id.text1, android.R.id.text2});
        // 연결할 레이아웃 지정
    }
}
```

```

        listView.setChoiceMode(AbsListView.CHOICE_MODE_MULTIPLE);
        // 이거 붙여줘야 실제로 선택됨!
        listView.setAdapter(adapter); // 어댑터로 리스트뷰와 데이터 연결

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener(){
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
                int position, long id) {
                String str = array.get(position).get("item1").toString();
            }
        });

        // 2. entries 이용하기
        // xml으로 Resource에 데이터를 등록해놓으면
        // 이 귀찮은 과정을 생략해도 된다.

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener(){
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
                int position, long id) {
                String str = parent.getItemAtPosition(position).toString();
                // entry로 구현한 경우 배열을 가지고 있는 것이 아니기 때문에
                // AdapterView(parent)의 몇 번째 인덱스 > 이런 식으로 접근해야
함!
            }
        });

        listView.setDivider(new ColorDrawable(Color.RED));
        listView.setDividerHeight(3);

        // 3. 단어장 만들기
        words = new HashMap<>();
        readFile();
        keySet = new ArrayList<>(words.keySet());

        // 데이터 세팅이 끝났으니 어댑터로 연결해준다!
        arrayAdapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, keySet);
        listView.setAdapter(arrayAdapter);
    }

    public void readFile() {

```

```

Scanner scanner = new Scanner(
    getResources().openRawResource(R.raw.words));

while (scanner.hasNextLine()){
    String eng = scanner.nextLine();
    String kor = scanner.nextLine();

    words.put(eng, kor);
}

scanner.close();
}
}

```

실행 결과

