모바일 프로그래밍

06 리스트 뷰

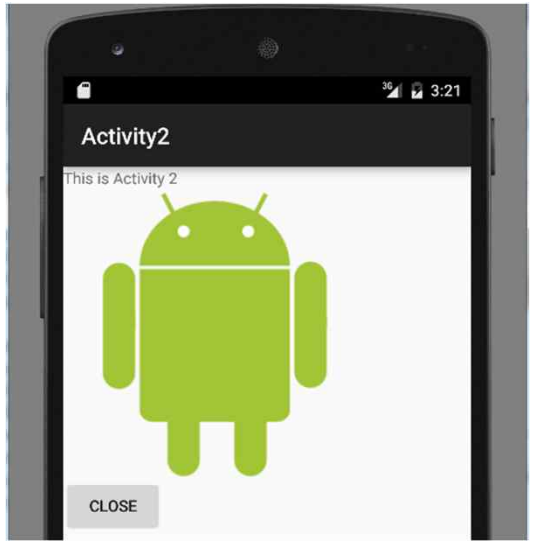
2017 2학기

강승우

Computer Science and Engineering, KOREATECH

지금까지 사용해 온 레이아웃

- 레이아웃에 들어가는 내용(뷰 항목)이 고정되어 있는 것
  - TextView, Button, EditText, ImageView 등
  - 텍스트 내용을 바꾸거나 이미지를 변경하는 것은 가능



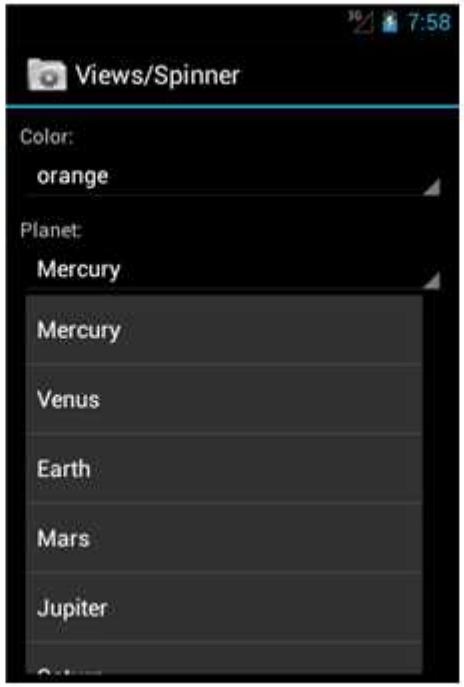
- 하지만, 새로운 TextView 항목을 추가하는 등의 동적인 변경을 하고자 할 때는 어떻게 해야 할까?

어댑터 뷰 (AdapterView)

- 화면에 동적으로 변경되는 컨텐츠를 채울 때 사용하는 뷰
  - 배열, 파일, 데이터베이스에 저장된 데이터를 화면에 표시할 때 유용

ListView

Gallery



Spinner

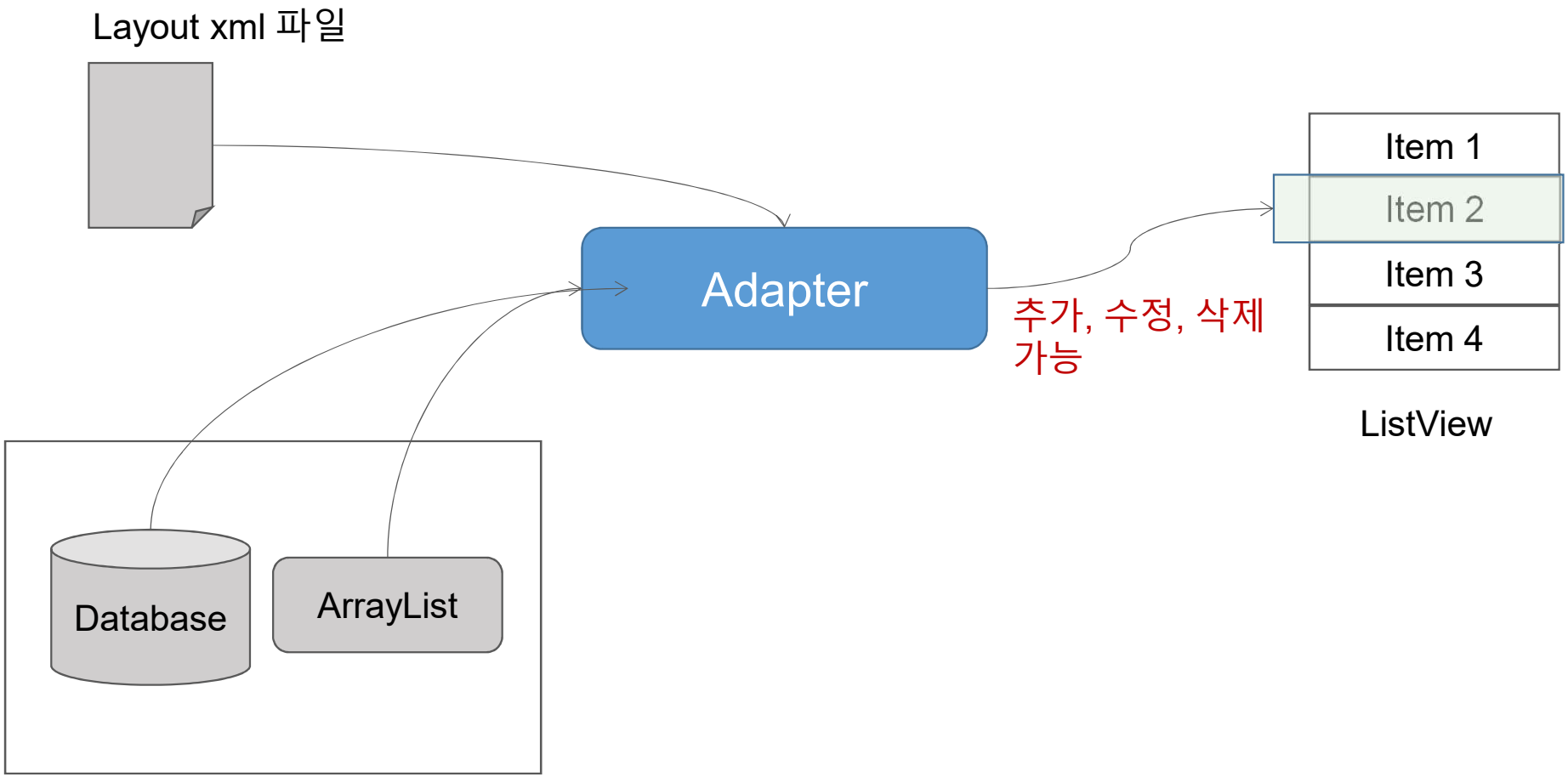
GridView

어댑터 (Adapter)

- Adapter를 사용하여 데이터를 어댑터 뷰에 제공
  - 어댑터는 데이터 소스와 어댑터 뷰 중간에 위치하여 데이터 소스에서 데이터를 읽어서 어댑터 뷰에 공급
  - https://developer.android.com/reference/android/widget/Adapter.html
- ArrayAdapter
  - 배열에서 데이터를 가져오는 어댑터
  - https://developer.android.com/reference/android/widget/ArrayAdapter.html
- SimpleCursorAdapter
  - 데이터베이스에서 데이터를 가져오는 어댑터

➢ TextView와 같은 기본 위젯은 뷰에 직접 데이터를 설정
  - setText() 메소드를 이용했음

어댑터


Layout xml 파일
Adapter
추가, 수정, 삭제 가능
Item 1
Item 2
Item 3
Item 4
ListView
Database
ArrayList

리스트 뷰 (ListView)

- 항목들을 수직 방향의 목록 형태로 보여주는 어댑터 뷰
  - 상하 스크롤이 가능
  - 일반적으로 목록의 한 항목을 선택하여 일정한 작업 수행

리스트 뷰 생성 방법

- 레이아웃 파일에 <ListView> element 선언하기
  - 예제 프로젝트 이름: Ch9ListView

- ListActivity를 상속받는 액티비티로 만들기
  - 리스트 뷰가 레이아웃 화면으로 미리 설정되어 있는 ListActivity 사용
    - https://developer.android.com/reference/android/app/ListActivity.html
  - 예제 프로젝트 이름: Ch9ListView2

리스트 뷰 예제 1 – ListView element

```
public class MainActivity extends AppCompatActivity {
    private ListView m_ListView;
    private ArrayAdapter<String> m_Adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        String[] values = {"하스스톤", "몬스터 헌터", "디아블로", "와우", "리니지", "안드로이드", "아이폰"};

        // Android에서 제공하는 String 문자열 하나를 출력하는 layout으로 어댑터 생성
        m_Adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, values);
        // Xml에서 추가한 ListView의 객체
        m_ListView = (ListView) findViewById(R.id.list);
        // ListView에 어댑터 연결
        m_ListView.setAdapter(m_Adapter);
        // ListView 아이템 터치 시 이벤트를 처리할 리스너 설정
        m_ListView.setOnItemClickListener(onClickListItem);
    }
```

리스트 뷰 예제 – ListView element

- 리스트 뷰 항목을 선택 이벤트 처리를 위한 AdapterView.OnItemClickListener 를 구현해야 함

```
// 아이템 터치 이벤트 리스너 구현
private AdapterView.OnItemClickListener onClickListItem = new AdapterView.OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        // 이벤트 발생 시 해당 아이템 위치를 텍스트로 출력
        Toast.makeText(getApplicationContext(), m_Adapter.getItem(position), Toast.LENGTH_SHORT).show();
    }
};
```

리스트 뷰 예제 2 – ListActivity 상속

```
public class MainActivity extends ListActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_main);
        String[] values = {"하스스톤", "몬스터 헌터", "디아블로", "와우", "리니지", "안드로이드", "아이폰"};

        // Android에서 제공하는 String 문자열 하나를 출력하는 layout으로 어댑터 생성
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, values);
        // ListView에 어댑터 연결
        setListAdapter(adapter);
    }

    @Override
    protected void onListItemClick(ListView l, View view, int position, long id) {
        String item = (String)getListAdapter().getItem(position);
        Toast.makeText(getApplicationContext(), item + " selected", Toast.LENGTH_SHORT).show();
    }
}
```

- 리스트 뷰 항목을 선택 이벤트 처리를 위해 구현해야 하는 메소드

리스트 뷰의 표준 레이아웃

android.R.layout.

레이아웃 ID	설명
simple_list_item_1	하나의 텍스트 뷰 사용
simple_list_item_2	두개의 텍스트 뷰 사용
simple_list_item_checked	항목당 체크 표시
simple_list_item_single_choice	한 개의 항목만 선택
simple_list_item_multiple_choice	여러 개의 항목 선택 가능

- 앞의 예제에서 사용한 레이아웃

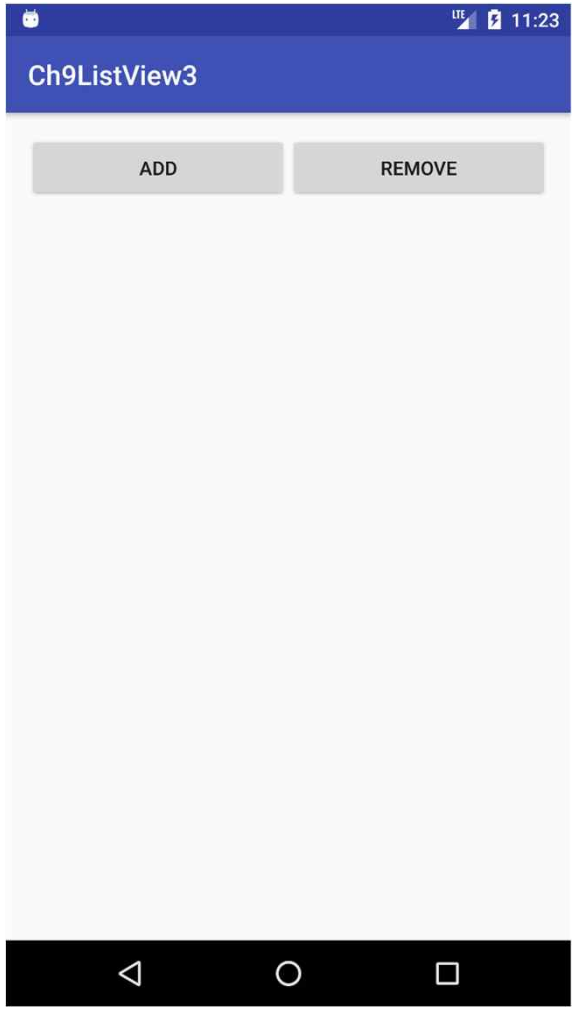
리스트 뷰 항목 추가/삭제 예제

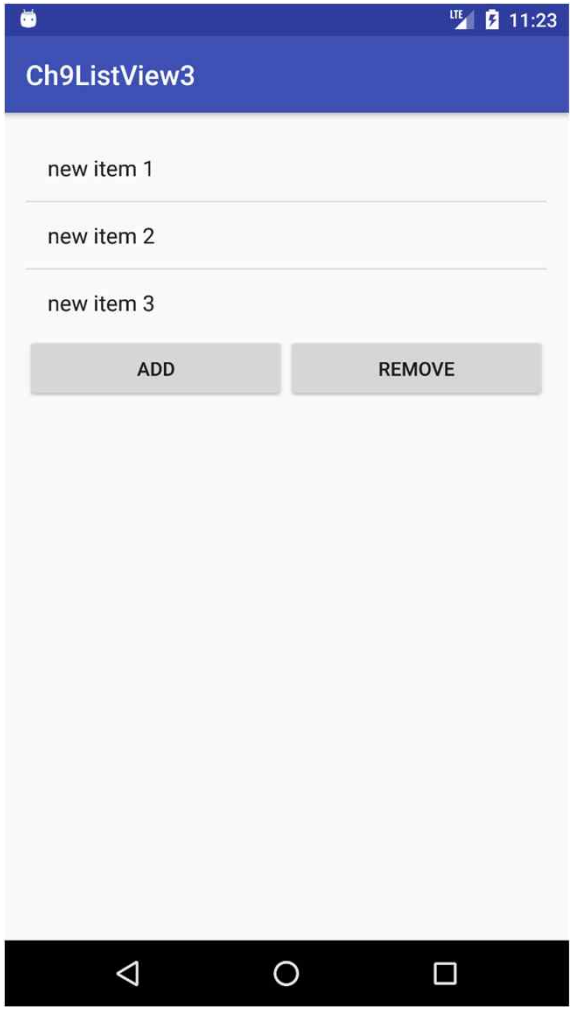
- 예제 프로젝트 이름
  - Ch9ListView3
  - Add, Remove 버튼을 이용하여 항목 추가, 삭제
- 관련 API
  - ArrayAdapter의 add, remove 메소드 사용
  - void add(T object)
  - void remove(T object)

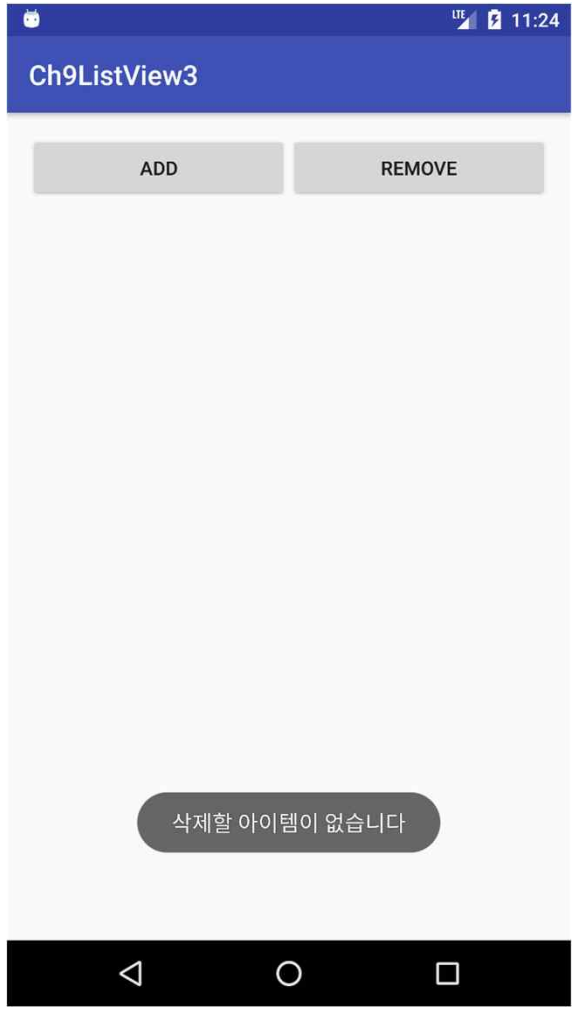
  - 추가, 삭제를 하려면 ListView로 보여지는 아이템은 정적 배열 객체가 아닌 List 객체여야 함
    - ArrayAdapter(Context context, int resource, T[] objects)
    - ArrayAdapter(Context context, int resource, List<T> objects)

https://developer.android.com/reference/android/widget/ArrayAdapter.html

리스트 뷰 항목 추가/삭제 예제







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

    //String[] values = {"하스스톤", "몬스터 헌터", "디아블로", "와우", "리니지", "안드로이드", "아이폰"};
    ArrayList<String> values = new ArrayList<>();
    //values.add("하스스톤");
    //values.add("몬스터 헌터");

    // Android에서 제공하는 String 문자열 하나를 출력하는 layout으로 어댑터 생성
    m_Adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, values);

    // Xml에서 추가한 ListView의 객체
    m_ListView = (ListView) findViewById(R.id.list);

    // ListView에 어댑터 연결
    m_ListView.setAdapter(m_Adapter);

    // ListView 아이템 터치 시 이벤트를 처리할 리스너 설정
    m_ListView.setOnItemClickListener(onClickListItem);
}
```

values를 String 객체를 담는 ArrayList 객체로 선언

이 부분은 앞의 Ch9ListView 예제와 동일

```
public void onClick(View view) {
    int count;
    count = m_Adapter.getCount();

    if(view.getId() == R.id.add) {
        // add 버튼 클릭한 경우 리스트의 마지막에 새 아이템 추가
        // adapter에 아이템 추가
        m_Adapter.add("new item " + Integer.toString(count + 1));
    } else if(view.getId() == R.id.remove) {
        // remove 버튼 클릭한 경우 리스트의 마지막 아이템을 삭제
        // 삭제할 아이템이 없으면 메시지 출력 후 종료
        if (count == 0) {
            Toast.makeText(getApplicationContext(), "삭제할 아이템이 없습니다", Toast.LENGTH_SHORT).show();
            return;
        }
        // 리스트의 마지막 아이템을 얻음
        String item = m_Adapter.getItem(count - 1);
        // 해당 아이템을 adapter에서 삭제
        m_Adapter.remove(item);
    }
}
```