

# Ch11. 어댑터뷰1

모바일게임프로그래밍  
김지심교수

# 학습목표



리스트뷰와 그리드뷰를 활용할 수 있다.  
갤러리와 스피너를 활용할 수 있다.

01



리스트뷰

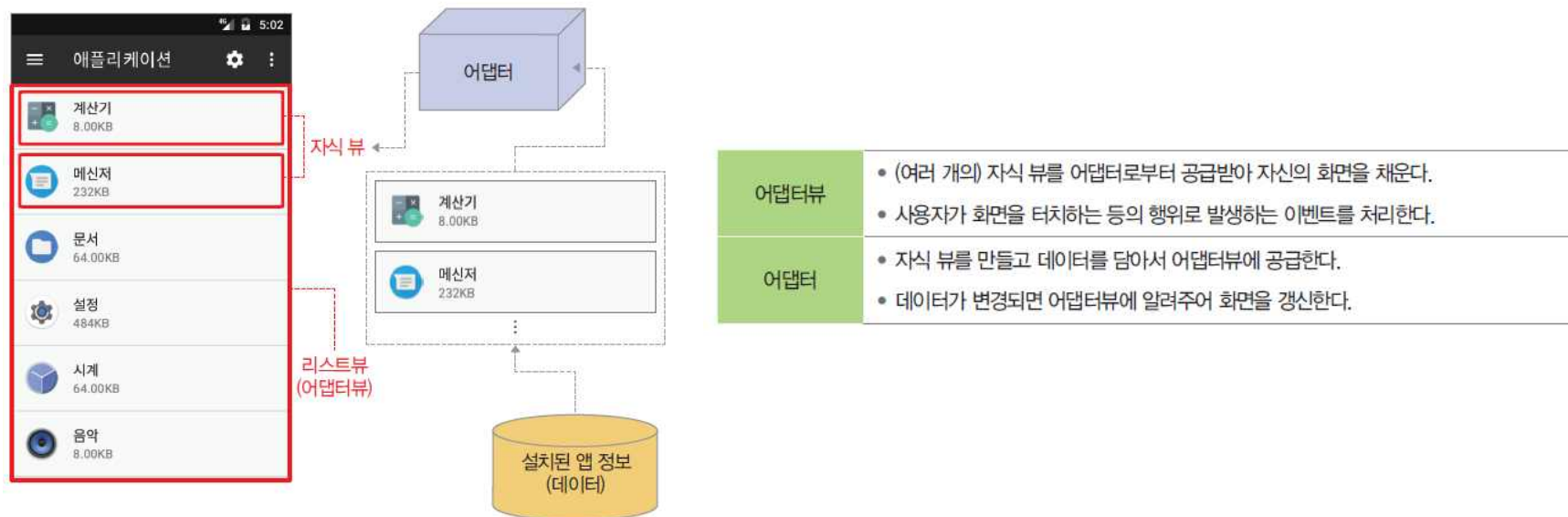


# 리스트뷰

## AdapterView

어댑터: 사용자가 정의한 데이터를 뷰에 출력하기 위해 데이터를 읽어 어댑터뷰에 공급  
ListView, ExpandableListView, GridView, Spinner, Gallery 등의 선택 위젯에 사용

어댑터뷰를 사용할 때 어댑터뷰의 모양을 설정하고 데이터를 채워주는  
타입별 어댑터 클래스를 함께 사용(ex. ArrayAdapter, SimpleCursorAdapter 등)



# 리스트뷰

## AdapterView

클래스 계층도

```
java.lang.Object
└ android.view.View
  └ android.widget.ViewGroup
    └ android.widget.AdapterView
      └ android.widget.AbsListView
        └ android.widget.GridView
        └ android.widget.ListView
        └ android.widget.ExpandableListView
      └ android.widget.AbsSpinner
        └ android.widget.Spinner
        └ android.widget.Gallery
```

어댑터뷰 계층도

# 리스트뷰

## ListView

데이터를 세로 방향의 목록 형태로 보여주고 하나를 선택하는 용도로 사용  
자동 스크롤링됨

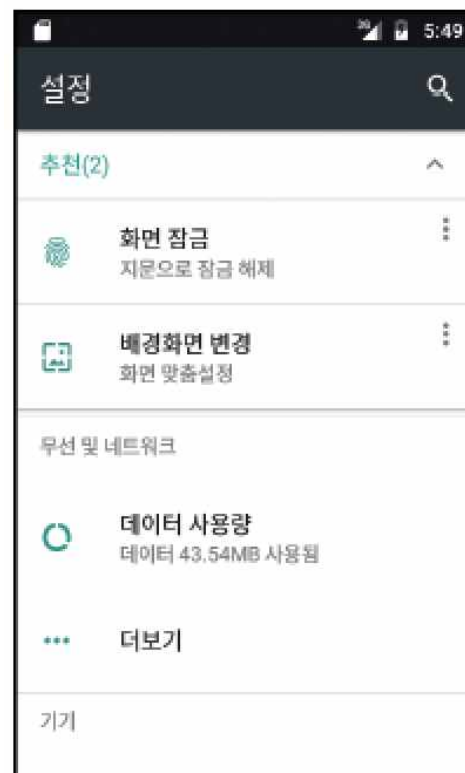


그림 11-1 안드로이드의 [설정] 기능

# 리스트뷰

## ArrayAdapter

CLASS	METHOD/DEFINITION
ArrayAdapter 선택위젯에 배열 데이터를 처리할 때	
ArrayAdapter(Context context, int resource, T[] objects)	ArrayAdapter 생성자 Context: 해당 클래스 resource: 항목에 사용하는 레이아웃의 리소스ID objects: 어댑터뷰에 공급할 데이터(Generics)

## ListView

METHOD	DEFINITION
void setAdapter(ListAdapter adapter)	리스트뷰에 adapter를 출력
void setChoiceMode(int choiceMode)	선택위젯(라디오버튼, 체크박스 등)에 따른 선택모드 지정
void setOnItemClickListener (AdapterView.OnItemClickListener listener)	항목 클릭 이벤트를 처리하는 리스너 등록
void onItemClick(AdapterView<?> parent, View view, int position, long id)	항목클릭 이벤트를 처리하는 콜백메소드 parent: 어댑터뷰 객체(부모) view: 클릭된 항목 뷰 position: 클릭된 항목 위치(from 0) id: 클릭된 항목 위치(long type)

# 리스트뷰

## \* 어댑터 생성시 적용하는 resource

안드로이드가 미리 만들어 제공하는 레이아웃의 id로서 뷰에 따라 다음 레이아웃들을 적용  
(개발자가 직접 정의한 R.layout.\*도 가능)

ID	적용되는 뷰	내용
android.R.simple_dropdown_item_1line	AutoComplete TextView	자동완성시 나타나는 드롭다운리스트 항목(1개씩 1줄)의 텍스트뷰 사용
android.R.layout.simple_list_item_1	ListView	목록 출력 시 1개의 텍스트뷰 사용
android.R.layout.simple_list_item_2	ListView	목록 출력 시 2개의 텍스트뷰 사용
android.R.simple_spinner_item	Spinner	현재 선택된 항목을 스피너뷰가 보여주는 레이아웃



# 리스트뷰

## XML을 이용한 리스트뷰 만들기

### 리스트뷰 기본 형식

1) 리스트뷰에 나열할 데이터를 String 배열로 정의

2) 데이터와 형식을 정의하여 어댑터 객체 생성

```
new ArrayAdapter<String>(Context context, int resource, T[] objects);  
                                android.R.layout.simple_list_item_1
```

3) 리스트뷰에 어댑터 설정

```
list.setAdapter(adapter);
```

4) 리스트뷰 항목의 클릭 이벤트 처리

```
list.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        //TODO  
    }  
});
```

# 리스트뷰

## 예제 11-1: Ex11\_01 리스트뷰 기본 예제

activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context="kjs.mjc.ex11_01.MainActivity">
9
10     <ListView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:id="@+id/listView1"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </android.support.constraint.ConstraintLayout>
```



# 리스트뷰

## 예제 11-1: Ex11\_01

### MainActivity.java

완성 후 실행해보자.

```
11 public class MainActivity extends AppCompatActivity {  
12  
13     @Override  
14     protected void onCreate(Bundle savedInstanceState) {  
15         super.onCreate(savedInstanceState);  
16         setContentView(R.layout.activity_main);  
17  
18         setTitle("리스트뷰 테스트");  
19  
20         final String[] mid = {"히어로즈", "24시", "로스트", "로스트룸", "스몰빌", "탐정몽크",  
21                               "빅뱅이론", "프렌즈", "덕스터", "글리", "가습걸", "테이큰", "슈퍼내추럴", "브이"};  
22  
23         ListView list = findViewById(R.id.listView1);  
24  
25         ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, mid);  
26         list.setAdapter(adapter);  
27  
28         list.setOnItemClickListener((adapterView, view, i, l) -> {  
31             Toast.makeText(MainActivity.this, mid[i], Toast.LENGTH_SHORT).show();  
32         });  
34     }  
35 }
```

# 리스트뷰

## 예제 11-1: Ex11\_01

아래와 같이 수정

라디오버튼 : simple\_list\_item\_single\_choice, CHOICE\_MODE\_SINGLE

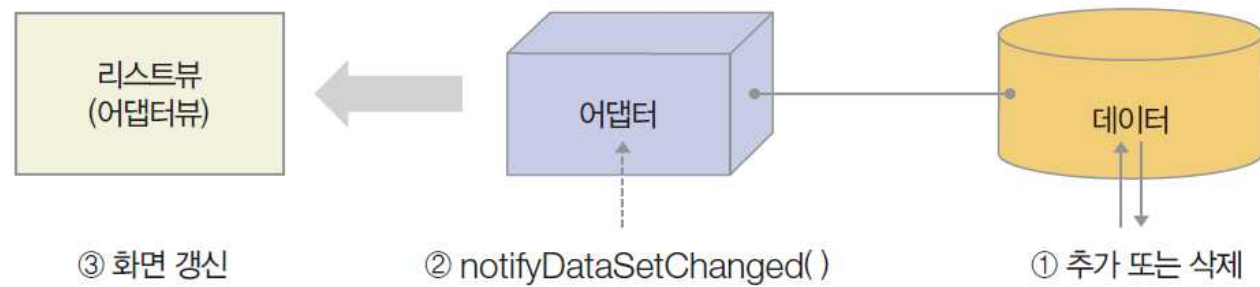
체크박스 : simple\_list\_item\_multiple\_choice, CHOICE\_MODE\_MULTIPLE

```
11 public class MainActivity extends AppCompatActivity {  
12  
13     @Override  
14     protected void onCreate(Bundle savedInstanceState) {  
15         super.onCreate(savedInstanceState);  
16         setContentView(R.layout.activity_main);  
17  
18         setTitle("리스트뷰 테스트");  
19  
20         final String[] mid = {"히어로즈", "24시", "로스트", "로스트룸", "스몰빌", "탐정몽크",  
21             "빅뱅이론", "프렌즈", "덕스터", "글리", "가습걸", "테이큰", "슈퍼내추럴", "브이"};  
22  
23         ListView list = findViewById(R.id.listView1);  
24  
25         ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_single_choice, mid);  
26         list.setChoiceMode(ListView.CHOICE_MODE_SINGLE);  
27         list.setAdapter(adapter);  
28  
29         list.setOnItemClickListener((adapterView, view, i, l) -> {  
32             Toast.makeText(MainActivity.this, mid[i], Toast.LENGTH_SHORT).show();  
33         });  
35     }  
36 }
```

# 리스트뷰

## 리스트뷰의 추가 · 삭제

리스트뷰의 항목을 추가·삭제하려면 `ArrayList<T>`를 정의한 후  
`add()`와 `remove()` 메소드를 사용  
변경사항을 업데이트하기 위해 `notifyDataSetChanged()`



# 리스트뷰

## 예제 11-4: Ex11\_04

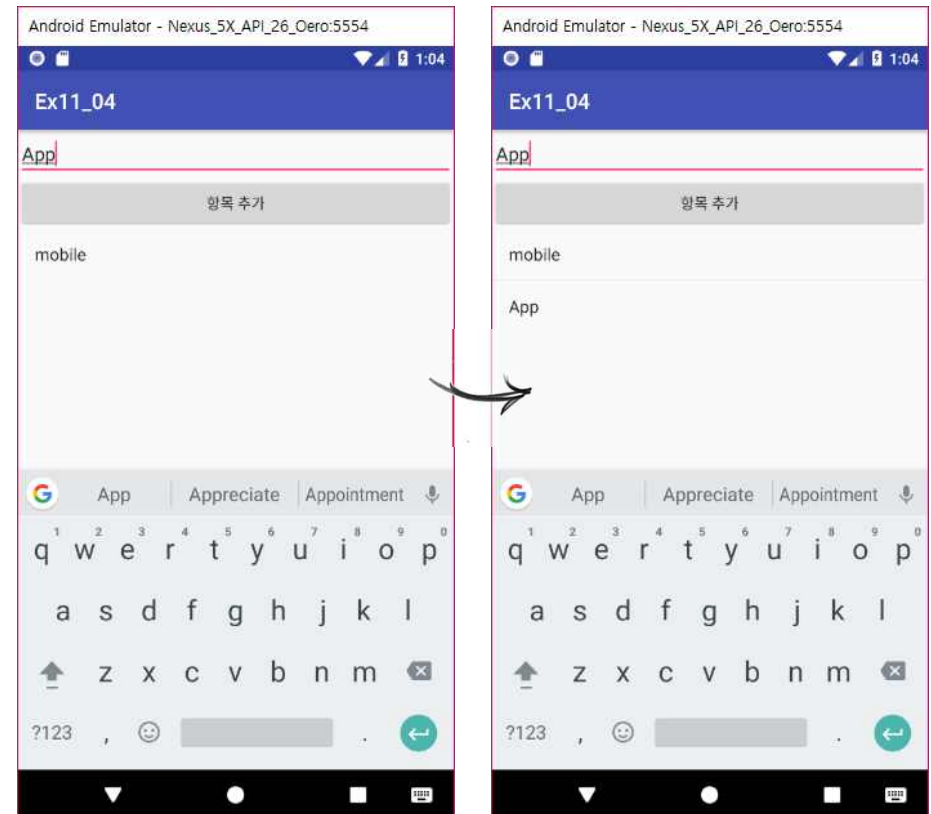
자바 코딩으로 EditText에 입력받은 항목을 리스트뷰에 추가하고  
항목을 롱클릭하여 삭제(추가·삭제를 동적으로 구현)

ArrayList 이용

추가: add()

삭제: remove()

업데이트: notifyDataSetChanged()



# 리스트뷰

## 예제 11-4: Ex11\_04 activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="kjs.mjc.ex11_04.MainActivity">
9
10     <EditText
11         android:layout_width="0dp"
12         android:layout_height="wrap_content"
13         android:id="@+id/editItem"
14         android:hint="추가할 항목 입력"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent"
18         app:layout_constraintBottom_toBottomOf="@+id/btnAdd"/>
19
20     <Button
21         android:layout_width="0dp"
22         android:layout_height="wrap_content"
23         android:id="@+id/btnAdd"
24         android:text="항목 추가"
25         app:layout_constraintLeft_toLeftOf="parent"
26         app:layout_constraintRight_toRightOf="parent"
27         app:layout_constraintTop_toBottomOf="@+id/editItem"/>
28
29     <ListView
30         android:layout_width="0dp"
31         android:layout_height="wrap_content"
32         android:id="@+id/lvItem"
33         app:layout_constraintLeft_toLeftOf="parent"
34         app:layout_constraintRight_toRightOf="parent"
35         app:layout_constraintTop_toBottomOf="@+id/btnAdd"/>
36 </android.support.constraint.ConstraintLayout>
```



15 ※ 밀줄 색상 변경: 테마에 따라 res\values\colors.xml의 colorAccent가 밀줄 색상이므로 이를 다른 색상으로 변경 가능



# 리스트뷰

## 예제 11-4: Ex11\_04 MainActivity.java

```
14 public class MainActivity extends AppCompatActivity {  
15  
16     @Override  
17     protected void onCreate(Bundle savedInstanceState) {  
18         super.onCreate(savedInstanceState);  
19         setContentView(R.layout.activity_main);  
20  
21         final EditText editItem = findViewById(R.id.editItem);  
22         Button btnAdd = findViewById(R.id.btnAdd);  
23         ListView lvItem = findViewById(R.id.lvItem);  
24  
25         final ArrayList<String> midList = new ArrayList<>();  
26  
27         final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, midList);  
28         lvItem.setAdapter(adapter);  
29     }
```



# 리스트뷰

## 예제 11-4: Ex11\_04 MainActivity.java

```
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47
```

```
btnAdd.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        midList.  
        adapter.  
    }  
});  
  
lvItem.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public boolean onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
        midList.  
        adapter.  
        return false;  
    }  
});  
}
```

완성해보기!

완성해보기!

# 리스트뷰

simple\_list\_item\_1는 어디에?

Ctrl + B

안드로이드 스튜디오가 설치된 하위 경로 상에서 platforms의 하위에 존재

C:\Android\sdk\platforms\android-27\data\res\layout

# 리스트뷰

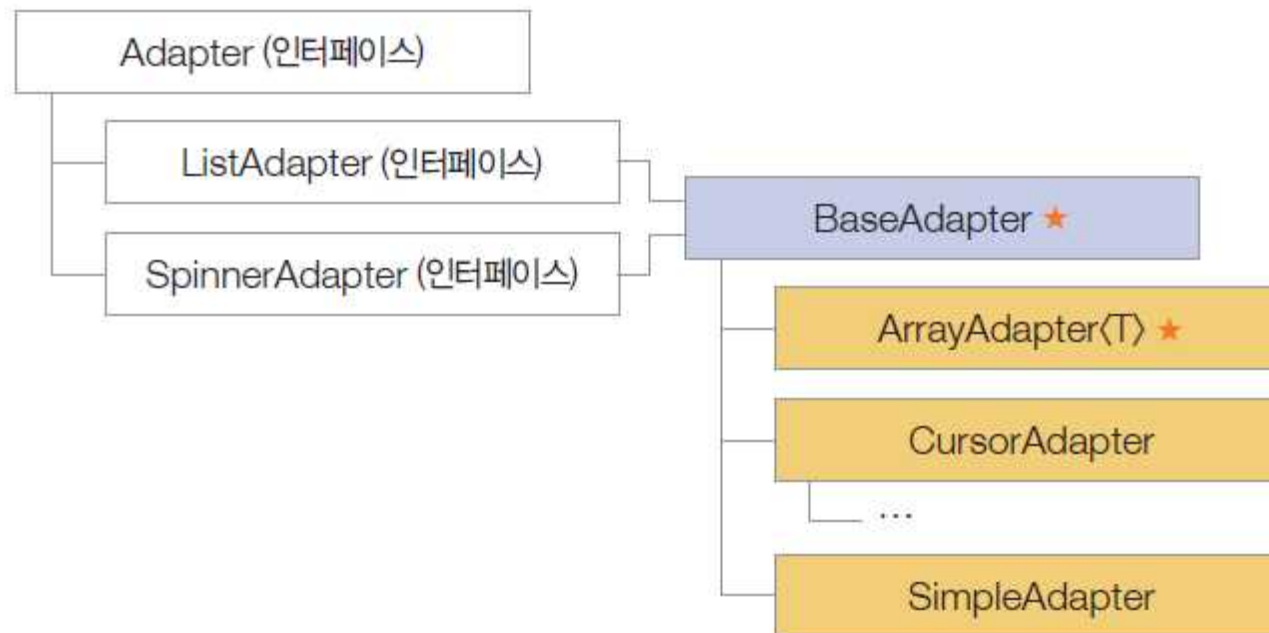
## ※ 두 개의 텍스트뷰로 항목 구성

HashMap 데이터를 여러 개 나열할 수 있는 SimpleAdatper 클래스 이용  
'simple\_list\_item\_2' 형식으로 출력

```
15 public class MainActivity extends AppCompatActivity {
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         ListView list = findViewById(R.id.listView1);
23
24         ArrayList<HashMap<String, String>> tellList = new ArrayList<>();
25         HashMap<String, String> item;
26
27         for (int i = 0; i < 10; i++) {
28             item = new HashMap<>();
29             item.put("name", "홍길동" + i);
30             item.put("tel", "010-0000-000" + i);
31             tellList.add(item);
32         }
33
34         String[] vauleArray = new String[]{"name", "tel"};
35         int[] viewForm = new int[]{android.R.id.text1, android.R.id.text2};
36
37         SimpleAdapter adapter = new SimpleAdapter(this, tellList, android.R.layout.simple_list_item_2, vauleArray, viewForm);
38         list.setAdapter(adapter);
39     }
40 }
41
```

# 리스트뷰

## \* SimpleAdapter 계층도



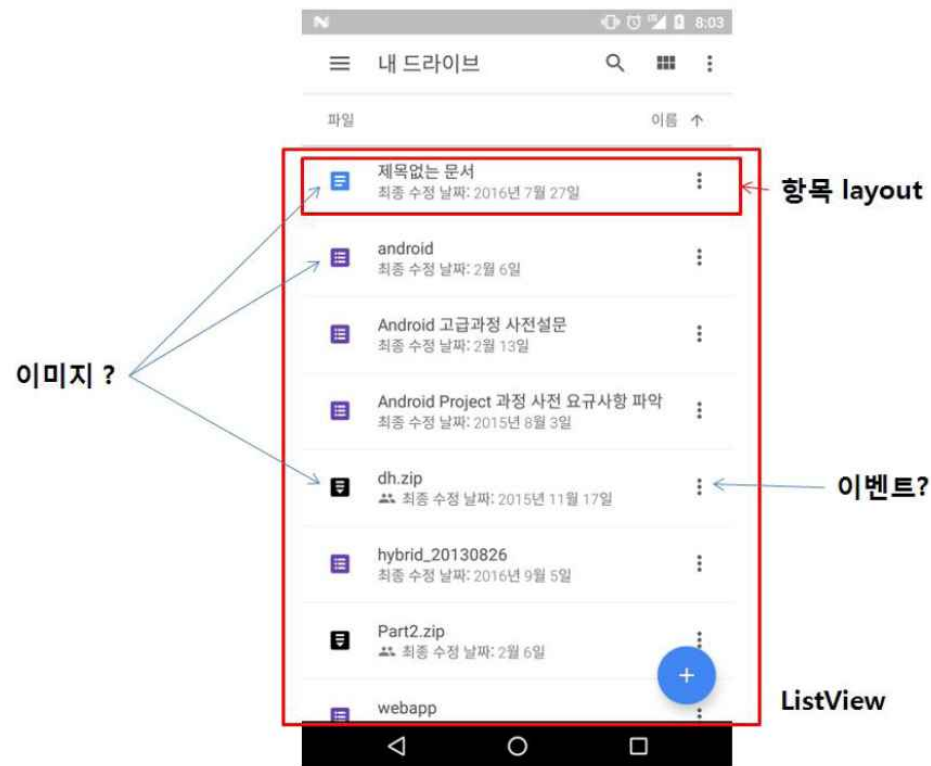
# 리스트뷰

## ※ Custom Adapter가 필요한 예시

개발자 알고리즘대로 항목의 데이터가 설정되어야 할 때

개발자 알고리즘대로 항목별 뷰의 이벤트를 각기 처리해야 할 때

개발자 알고리즘대로 항목별 레이아웃을 다르게 적용해야 할 때



# 참고문헌

## 참고문헌

강성윤(2017). 깡샘의 안드로이드 프로그래밍, 루비페이퍼.

김선우(2017). 단계별로 배우는 안드로이드 프로그래밍, 한빛아카데미