

# Ch11. 어댑터뷰2

모바일게임프로그래밍  
김지심교수

# 학습목표



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

02

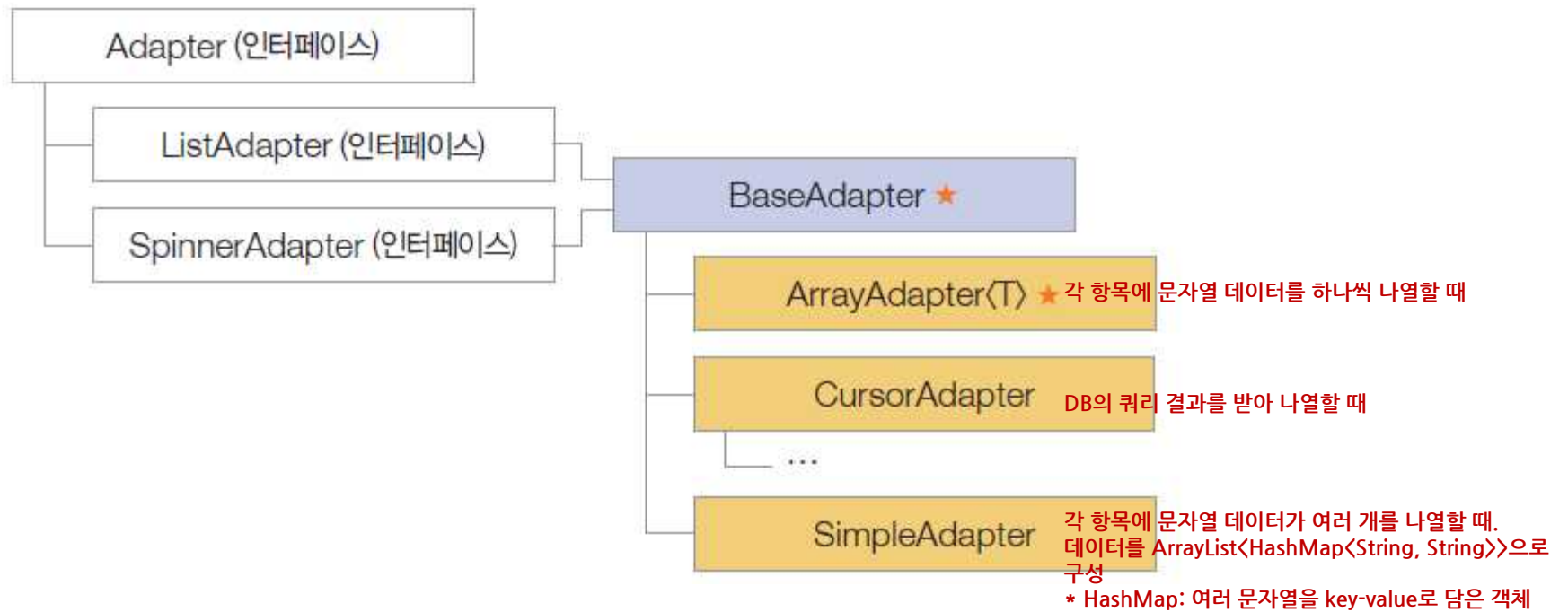


그리드뷰



# 그리드뷰

## \* BaseAdapter 계층도



# 그리드뷰

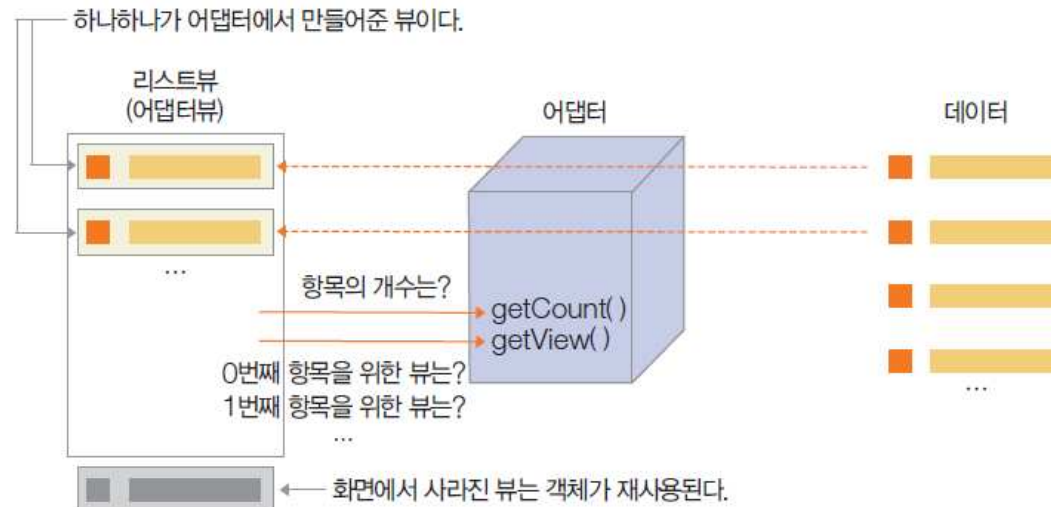
## 커스텀 어댑터뷰 구현

문자열을 나열하는 것 이상의 AdapterView를 만들기 위해서는 개발자가 직접 커스텀 Adapter를 만들어 적용해야 함

BaseAdapter를 상속받아 Adapter에서

- ① 데이터를 관리
- ② getView() 등을 오버라이딩하여 원하는
  - ②-1 항목 뷰를 화면에 출력하고,
  - ②-2 뷰의 이벤트를 구현

→ GridView 뿐 아니라 어떤 어댑터뷰에서든 사용



# 그리드뷰

## 어댑터 구현

CLASS	METHOD/DEFINITION
BaseAdapter 기본 어댑터 클래스	
BaseAdapter()	생성자, 주로 Context를 가짐(액티비티)
public int getCount()	데이터 항목의 전체 개수를 리턴
public Object getItem(int position)	position에 해당하는 항목의 데이터를 리턴 (객체 데이터를 사용할 때 등)
public long getItemId(int position)	position에 해당하는 항목의 식별자를 리턴
public View getView(int position, View convertView, ViewGroup parent)	position: 항목의 위치(from 0) convertView: 현재 위치에 해당하는 뷰 객체 parent: 부모 컨테이너 객체 ※ ViewHolder 패턴: convertView가 null이면 뷰를 새로 만들고 null이 아니면 재사용

→ //TODO 1) getCount()에서 데이터 개수를 리턴  
2) getView()에서 항목별 뷰(레이아웃 등)을 정의하여 리턴  
3) getView()에서 이벤트 처리

# 그리드뷰

## 커스텀 어댑터뷰 구현 기본 구조

```
private class MyGridAdapter extends BaseAdapter {  
  
    Context context;  
  
    public MyGridAdapter(Context context) {  
        this.context = context;  
    }  
  
    @Override  
    public int getCount() {  
        //TODO 데이터 개수 리턴  
    }  
  
    @Override  
    public Object getItem(int position) {  
        //TODO  
    }  
  
    @Override  
    public long getItemId(int position) {  
        //TODO  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
  
        //TODO: 항목을 구성할 뷰를 설정  
  
        //TODO: 항목 이벤트 처리  
  
        //TODO: 항목 리턴  
    }  
}
```

# 그리드뷰

## 그리드뷰(GridView)

그리드뷰는 사진이나 그림을 격자 모양으로 배치

XML 파일에는 간단히 <GridView>를 넣고, Java 코드에는 필요한 내용을 코딩

<GridView>의 속성 중 열 개수를 지정하는 **numColumns**는 필수 지정



# 그리드뷰

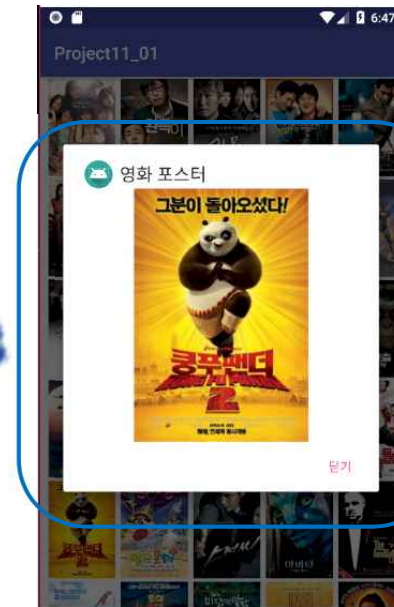
## 예제11-1: Project11\_01

그리드뷰에 영화 포스터들을 배치하고 클릭하면 대화상자에 포스터를 출력

GridView  
(각 항목은  
MyGridAdapter를  
구현하여 개발)



dialog.xml



GridView의 각 그리드에  
(안드로이드에서 제공하는 레이아웃이 아닌)  
ImageView에 이미지 출력 & 이벤트 처리  
→ getView()에서 구현!!

# 그리드뷰

## 예제 11-1: Project11\_01 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.project11_01.MainActivity">
9
10    <GridView
11        android:id="@+id/gvMovie"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:numColumns="5"
15        app:layout_constraintBottom_toBottomOf="parent"
16        app:layout_constraintLeft_toLeftOf="parent"
17        app:layout_constraintRight_toRightOf="parent"
18        app:layout_constraintTop_toTopOf="parent" />
19
20 </android.support.constraint.ConstraintLayout>
21
```



# 그리드뷰

## 예제 11-1: Project11\_01

영화 이미지를 res/drawable에 복사

### dialog.xml

포스터를 출력할 대화상자 XML을 추가하여 작성



New Resource File

File name: dialog

Root element: android.support.constraint.ConstraintLayout

Source set: main

Directory name: layout

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode

Chosen qualifiers:

Nothing

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

# 그리드뷰

## 예제11-1: Project11\_01

### MainActivity.java

그리드뷰에 MyGridAdapter 객체를 설정  
(BaseAdapter를 상속받아 MyGridAdapter를 구현할 예정)

```
7 public class MainActivity extends AppCompatActivity {  
8  
9     Integer[] posterID = {R.drawable.mov01, R.drawable.mov02, R.drawable.mov03, R.drawable.mov04, R.drawable.mov05,  
10        R.drawable.mov06, R.drawable.mov07, R.drawable.mov08, R.drawable.mov09, R.drawable.mov10};  
11  
12     @Override  
13     protected void onCreate(Bundle savedInstanceState) {  
14         super.onCreate(savedInstanceState);  
15         setContentView(R.layout.activity_main);  
16  
17         GridView gvMovie = findViewById(R.id.gvMovie);  
18  
19         MyGridAdapter gAdapter = new MyGridAdapter(this);  
20         gvMovie.setAdapter(gAdapter);  
21     }  
22 }
```



# 그리드뷰

## 예제11-1: Project11\_01

### MainActivity.java

MyGridAdapter를 내부 클래스로 구현

**Alt + Enter**

1. Add dependency on module 'Project11\_01\_01'  
2. Create class 'MyGridAdapter'  
3. Create enum 'MyGridAdapter'  
4. Create inner class 'MyGridAdapter'  
5. Create interface 'MyGridAdapter'  
6. Move 'kjs.mjc.project11\_01.MyGridAdapter' from module 'Project11\_01\_01' to 'Project11\_01'  
7. Split into declaration and assignment

2. private class MyGridAdapter extends BaseAdapter{

3. Context context;

4. 0

5. Generate... Alt+Insert

6. Generate  
Constructor  
Getter  
Setter  
Getter and equals() and toString()  
Override Method  
Implement  
Delegate Method  
Copyright

7. Choose Fields to Initialize by Constructor  
kjs.mjc.project11\_01.MainActivity.MyGridAdapter  
context:Context

8. public MyGridAdapter(Context context) {  
this.context = context;  
}

MainActivity를 Context로 전달하여 생성자 정의

```
29 MyGridAdapter gAdapter = new MyGridAdapter(this);  
30 gvMovie.setAdapter(gAdapter);  
31 }  
32  
33 private class MyGridAdapter extends BaseAdapter{  
34  
35     ③ Context context;  
36  
37     ④ 0  
38 }  
39  
40
```

# 그리드뷰

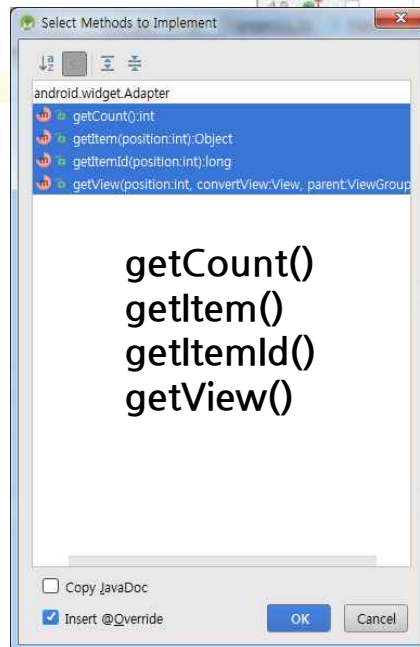
## 예제11-1: Project11\_01

### MainActivity.java

MyGridAdapter를 내부 클래스로 구현  
필수 메소드들을 오버라이딩

```
34 private class MyGridAdapter extends BaseAdapter{  
35  
36     Context context;  
37  
38     public MyGridAdapter(Context context) {  
39         this.context = context;  
40     }  
41  
42  
43 }
```

기본 메소드들 오버라이딩  
(Ctrl + I)



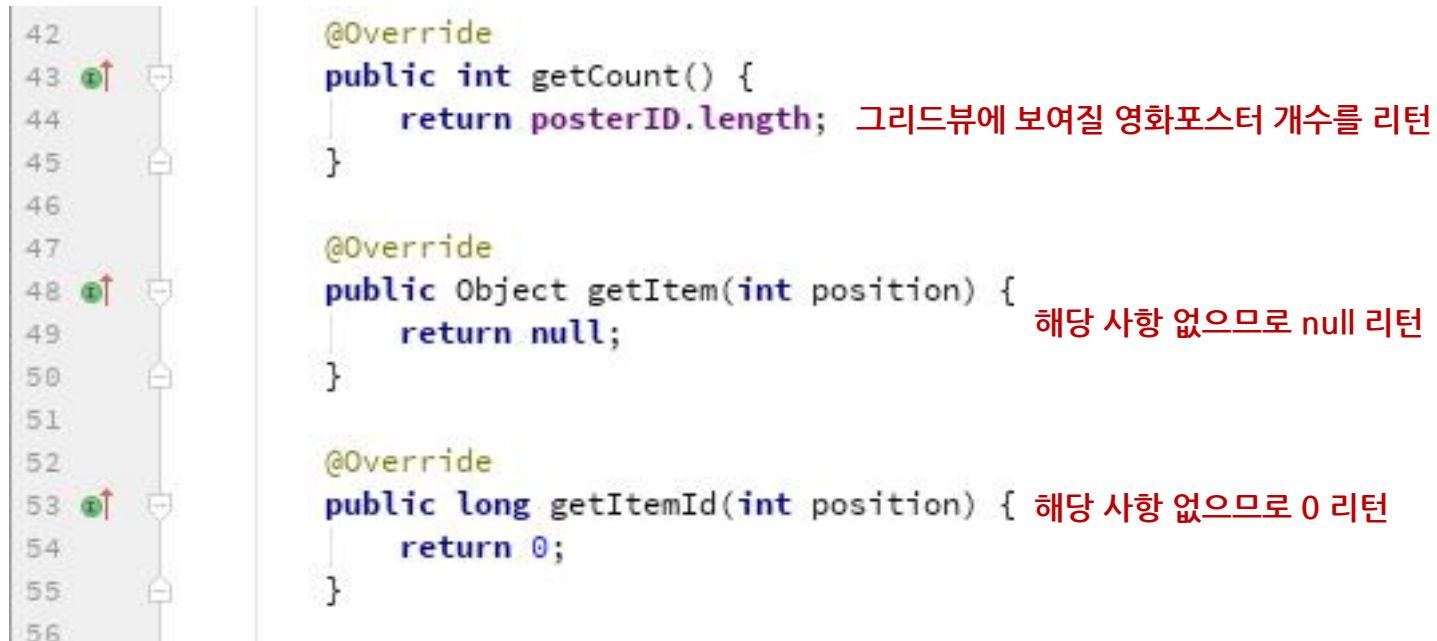
```
38 public MyGridAdapter(Context context) {  
39     this.context = context;  
40 }  
41  
42 @Override  
43 public int getCount() {  
44     return 0;  
45 }  
46  
47 @Override  
48 public Object getItem(int i) {  
49     return null;  
50 }  
51  
52 @Override  
53 public long getItemId(int i) {  
54     return 0;  
55 }  
56  
57 @Override  
58 public View getView(int i, View view, ViewGroup viewGroup)  
59     return null;  
60 }
```

# 그리드뷰

## 예제11-1: Project11\_01

### MainActivity.java

메소드들 오버라이딩하여 완성한 후 실행해보기!



```
42
43 ①↑
44
45
46
47
48 ①↑
49
50
51
52
53 ①↑
54
55
56
```

```
@Override
public int getCount() {
    return posterID.length; 그리드뷰에 보여질 영화포스터 개수를 리턴
}

@Override
public Object getItem(int position) {
    return null; 해당 사항 없으므로 null 리턴
}

@Override
public long getItemId(int position) { 해당 사항 없으므로 0 리턴
    return 0;
}
```

# 그리드뷰

## 예제11-1: Project11\_01 MainActivity.java

```
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
```

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    //GridView에 출력할 ImageView를 생성하여 이미지 출력
    ImageView ivG = new ImageView(context);
    ivG.setLayoutParams(new GridView.LayoutParams(250, 300));
    ivG.setScaleType(ImageView.ScaleType.FIT_CENTER);
    ivG.setPadding(5, 5, 5, 5);

    ivG.setImageResource.posterID[position]);

    final int pos = position;
    ivG.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            View dlgView = View.inflate(MainActivity.this, R.layout.dialog, null);
            AlertDialog.Builder dlg = new AlertDialog.Builder(MainActivity.this);
            ImageView ivPoster = (ImageView) dlgView.findViewById(R.id.ivPoster);
            ivPoster.setImageResource.posterID[pos]);

            dlg.setTitle("영화 포스터");
            dlg.setIcon(R.mipmap.ic_launcher);
            dlg.setView(dlgView);
            dlg.setNegativeButton("닫기", null);
            dlg.show();
        }
    });

    return ivG;
}
```

각 그리드에 이미지뷰를 생성하여  
포스터를 출력하도록 설정

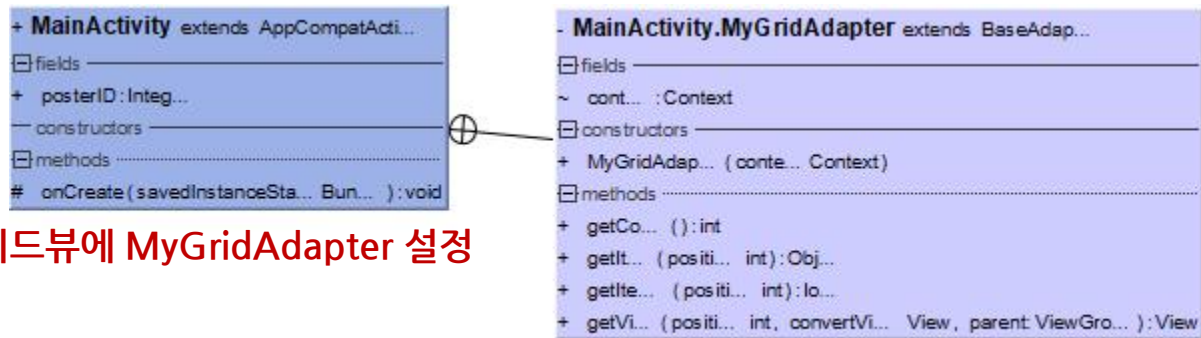
각 그리드의 클릭 이벤트 처리  
(대화상자로 포스터 출력)

return ivG; 그리드의 뷰 리턴(포스터를 담은 이미지뷰 리턴)



# 그리드뷰

## Summary



그리드뷰에 MyGridAdapter 설정

컨텍스트를 받아 그리드뷰의 항목 생성 및 이벤트 처리  
- 복습시 중요한 상세 로직을 순서대로 작성해보자.

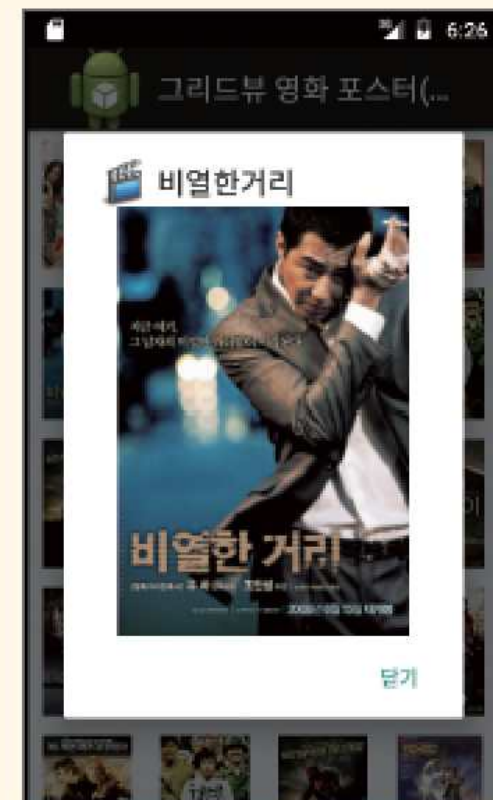
# 그리드뷰

## 실습(직접 풀어보기) 11-1: Test11\_01

### ▶ 직접 풀어보기 11-1

[실습 11-1]에서 영화 포스터를 클릭하면 나오는 대화상자의 제목창에 영화의 제목이 보이도록 수정하자. 대화상자의 아이콘도 수정한다.

**HINT 1** posterID와 동일한 개수의 문자열 배열을 정의하고 영화 제목을 넣는다.



03



갤러리



# 갤러리

## 갤러리(Gallery)

사진이나 이미지를 배치하고 수평으로 스크롤링

# 갤러리

## 실습11-2: Project11\_02 갤러리 및 이미지 터치 시 출력



그림 11-3 갤러리 영화 포스터 보기 실행 화면

# 갤러리

## 실습11-2: Project11\_02

영화 이미지를 res/drawable에 복사

`activity_main.xml`

갤러리와 이미지뷰를 추가

# 갤러리

## 실습11-2: Project11\_02

### MainActivity.java

BaseAdapter를 상속받는 내부 클래스 MyGalleryAdapter를 정의하고  
그리드뷰에 MyGalleryAdapter 적용

```
15 public class MainActivity extends AppCompatActivity {
16
17     @Override
18     public void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21         setTitle("갤러리 영화 포스터(개선)");
22
23         Gallery gallery = findViewById(R.id.gallery1);
24
25         MyGalleryAdapter galAdapter = new MyGalleryAdapter(this);
26         gallery.setAdapter(galAdapter);
27     }
28
29     public class MyGalleryAdapter extends BaseAdapter {
30
31         Context context;
32         Integer[] posterID = { R.drawable.mov11, R.drawable.mov12,
33                                R.drawable.mov13, R.drawable.mov14, R.drawable.mov15,
34                                R.drawable.mov16, R.drawable.mov17, R.drawable.mov18,
35                                R.drawable.mov19, R.drawable.mov20 };
36
37         String[] posterTitle = { "여인의 향기", "주라기 공원", "포레스트 검프", "사랑의 불학살",
38                                "흑성탈출", "마른다운비행", "내 이름은 칸", "해리포터", "마더", "킹콩을 들다" };
39
40         public MyGalleryAdapter(Context c) { context = c; }
41
42         public int getCount() { return posterID.length; }
43
44         public Object getItem(int arg0) { return null; }
45
46         public long getItemId(int position) { return 0; }
47     }
48
49     51
52     53
54     55
```



# 갤러리

## 실습11-2: Project11\_02

### MainActivity.java

갤러리의 각 이미지 터치 시 아래쪽 이미지뷰에 포스터 출력

```
56 public View getView(int position, final View convertView, ViewGroup parent) {
57     ImageView imageview = new ImageView(context);
58     imageview.setLayoutParams(new Gallery.LayoutParams(100, 150));
59     imageview.setScaleType(ImageView.ScaleType.FIT_CENTER);
60     imageview.setPadding(5, 5, 5, 5);
61     imageview.setImageResource(posters[position]);
62
63     final int pos = position;
64     imageview.setOnTouchListener(new View.OnTouchListener() {
65         public boolean onTouch(View v, MotionEvent event) {
66             ImageView ivPoster = (ImageView) findViewById(R.id.ivPoster);
67             ivPoster.setScaleType(ImageView.ScaleType.FIT_CENTER);
68             ivPoster.setImageResource(posters[pos]);
69
70             Toast toast = new Toast(getApplicationContext());
71             View toastView = (View) View.inflate(
72                 getApplicationContext(), R.layout.toast, null);
73             TextView toastText = (TextView) toastView
74                 .findViewById(R.id.tvMsg);
75             toastText.setText(posters[pos]);
76             toast.setView(toastView);
77             toast.show();
78
79             return false;
80         }
81     });
82
83     return imageview;
84 }
```

각 항목(이미지뷰)을 생성하고  
이미지 출력토록 설정

갤러리 항목 클릭 이벤트 처리  
(하단의 이미지뷰에 이미지 출력)



# 갤러리

## 실습(직접 풀어보기) 11-2: Test11\_02

### ▶ 작업 풀어보기 11-2

[실습 11-2]에서 영화 포스터를 클릭하면 영화의 제목이 토스트 메시지에 나타나도록 수정하자. 토스트는 아이콘과 함께 보인다.

**NOTE** 고급 토스트는 [실습 7-3]을 참조한다.



04



스피너



# 스피너

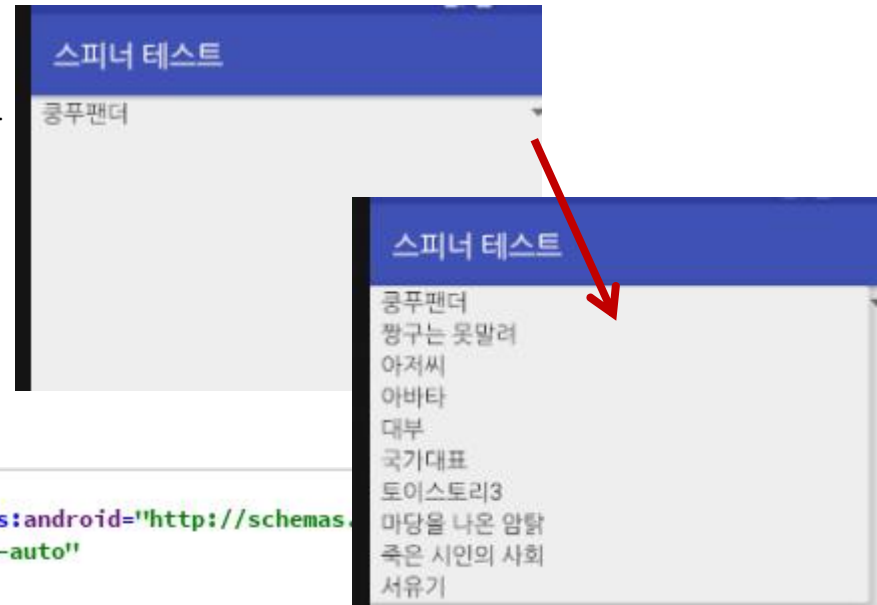
## 스피너(Spinner)

PC의 드롭다운(Drop Down) 박스에 해당

### 예제 11-16: Ex11\_16

스피너 기본 예제

화면 디자인



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      android:id="@+id/activity_main"
5      android:orientation="vertical"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent">
8
9      <Spinner
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:id="@+id/spinner1"
13         app:layout_constraintTop_toTopOf="parent"/>
14  </android.support.constraint.ConstraintLayout>
15
```

# 스피너

## 예제 11-16: Ex11\_16 MainActivity.java

```
8 public class MainActivity extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         setTitle("스피너 테스트");
15
16         final String[] movie = { "쿵푸팬더", "짱구는 못말려", "아저씨", "아바타", "대부", "국가대표",
17                                 "토이스토리3", "마당을 나온 암탉", "죽은 시인의 사회", "서유기" };
18
19         Spinner spinner = (Spinner) findViewById(R.id.spinner1);
20
21         ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_item, movie);
22         spinner.setAdapter(adapter);
23     }
24 }
```

# 스피너

## 실습(직접 풀어보기) 11-3: Test11\_03

### ▶ 직접 풀어보기 11-3

[예제 11-17]의 스피너에서 영화 제목을 선택하면 해당 영화 포스터가 아래의 이미지뷰에 나오게 하자.

**HINT 1** Spinner의 `setOnItemSelectedListener()`를 사용한다.



# 참고문헌

## 참고문헌

강성윤(2017). 깡샘의 안드로이드 프로그래밍, 루비페이퍼.

김선우(2017). 단계별로 배우는 안드로이드 프로그래밍, 한빛아카데미