

# **모바일프로그래밍**

## **- 안드로이드(Android) -**

## ○안드로이드 기능/실습 - 15

- GridView
- BaseAdapter 사용

## ○안드로이드 기능/실습 - 16

- GridView + ImageView

# **□ 안드로이드 기능/실습 - 15**

## **-그리드뷰(GRIDVIEW)**

## □ GridView

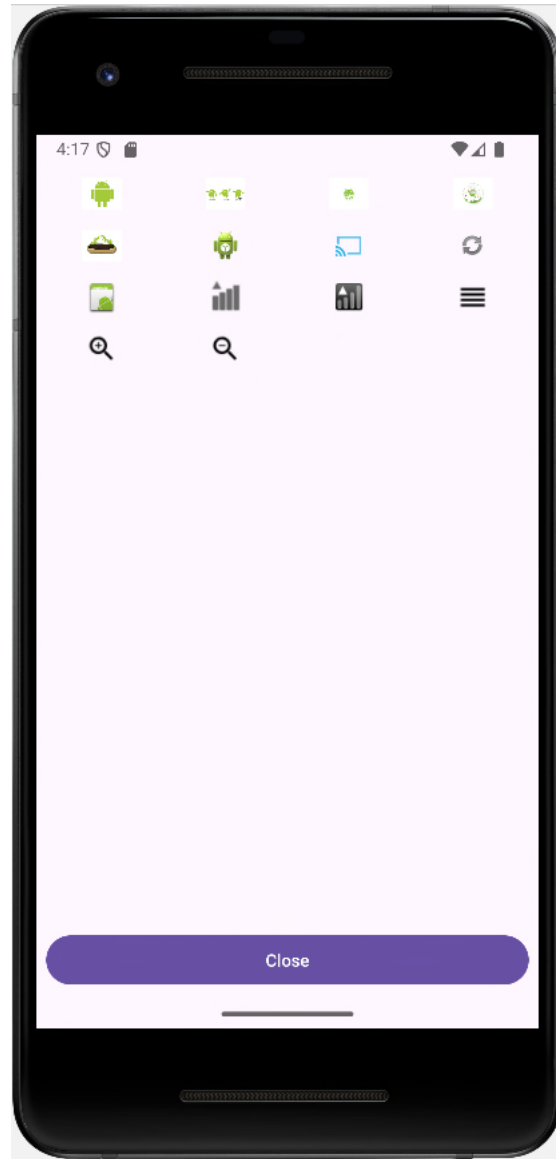
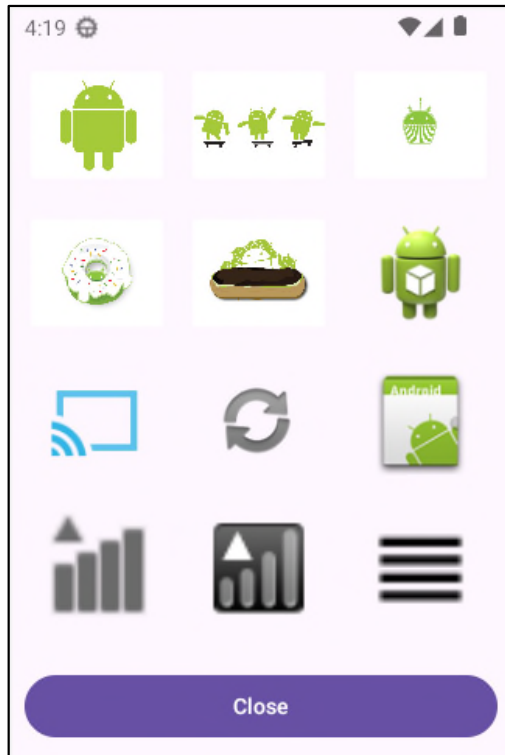
### ○ 목적

- GridView 사용방법을 습득하고 Adapter 가운데 BaseAdapter 사용법을 습득
- BaseAdapter를 이용하여 커스텀어댑터를 정의하여 이미지 표현

### ○ GridView

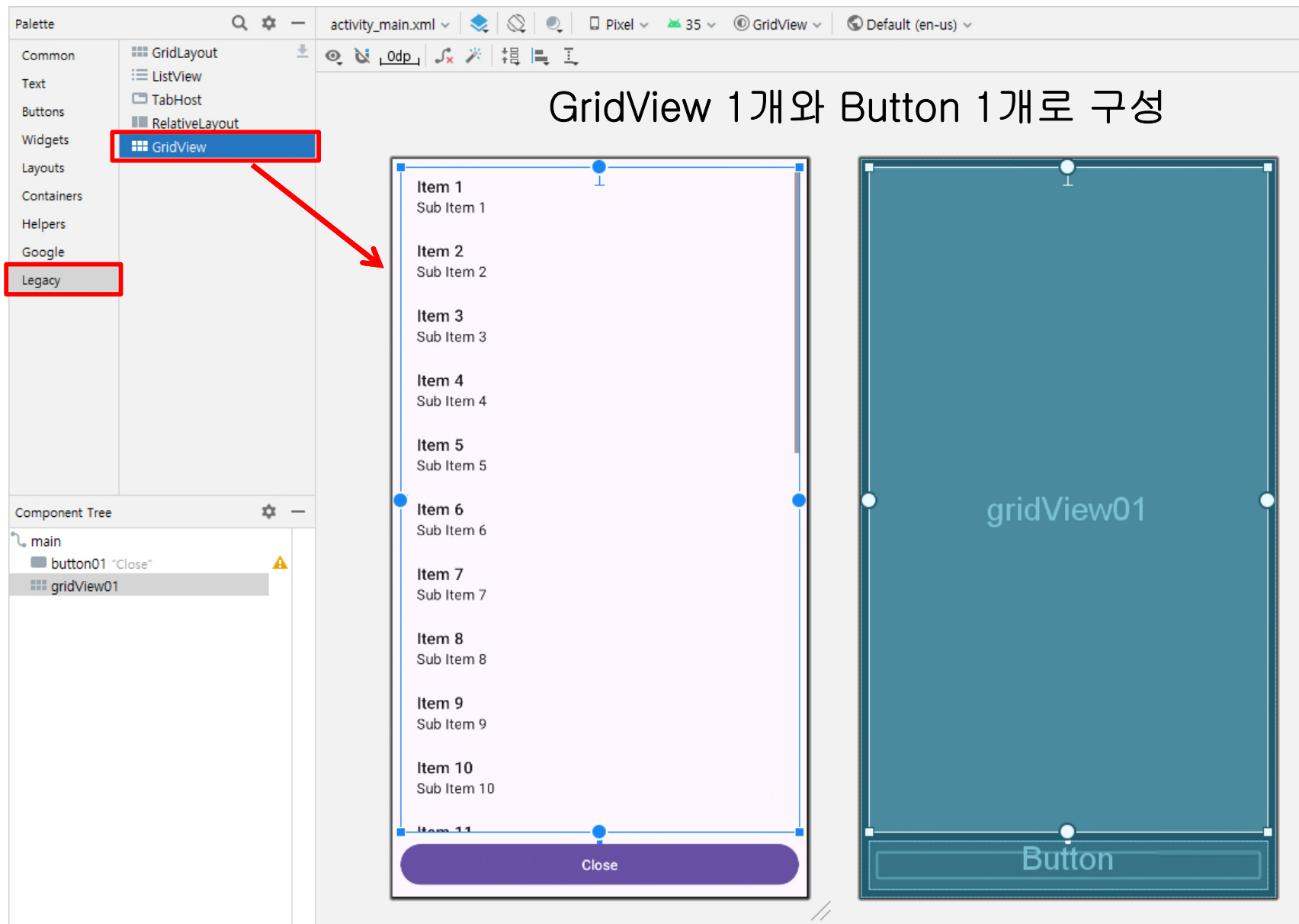
- 2차원 격자 모양으로 필요한 항목을 배치하고 항목을 선택하게 해 주는 기능
- 속성
  - ❖ android:verticalSpacing
    - 항목 간에 상하 여백을 얼마나 줄 것인가를 결정
  - ❖ android:horizontalSpacing
    - 항목 간의 좌우 여백을 얼마나 줄 것인가를 결정
  - ❖ android:columnWidth
    - 컬럼의 너비 지정
  - ❖ android:numColumns
    - 한 줄에 표시되는 컬럼 수, auto\_fit으로 하면 자동 조절

## □ 실행 모습



문제점은?  
- 해결 방법은?

## □ 레이아웃 구성



## □ 레이아웃

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button01"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="Close"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

## □ 레이아웃

```
<GridView
    android:id="@+id/gridView01"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:columnWidth="90dp"
    android:gravity="center"
    android:horizontalSpacing="10sp"
    android:numColumns="auto_fit"
    android:stretchMode="columnWidth"
    android:verticalSpacing="10dp"
    app:layout_constraintBottom_toTopOf="@+id/button01"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```



```
</androidx.constraintlayout.widget.ConstraintLayout>
```

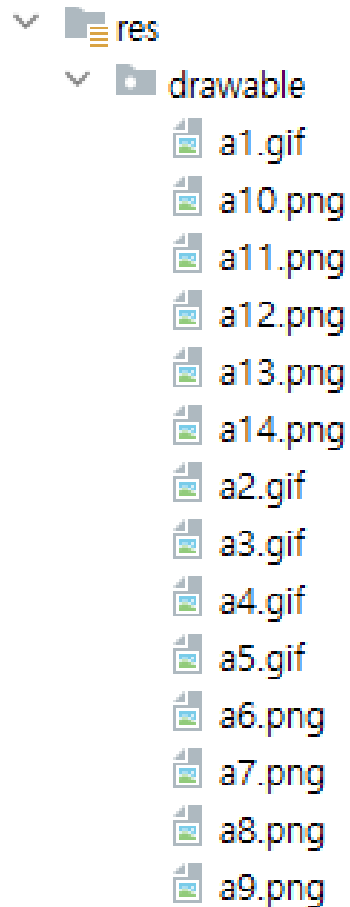
반드시 설정이 필요  
- 레이아웃 또는  
코드 상에서 설정 필요  
(코드 설정 권장)

- 각 열의 너비
- 가운데 정렬
- 항목간 좌우 여백
- 열 개수
- 남은 공간 배정
- 항목간 상하 여백



## □ Resource 사전 작업

- 표시될 이미지를 res/drawable에 복사
- 이미지는 수업자료 게시판에 업로드



## □ 코드 분석

```
package com.practice.ex.gridview;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.GridView;
import android.widget.ImageView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    public static Integer[] mThumbIds = {R.drawable.a1, R.drawable.a2, R.drawable.a3, 3 usages
        R.drawable.a4, R.drawable.a5, R.drawable.a6, R.drawable.a7,
        R.drawable.a8, R.drawable.a9, R.drawable.a10, R.drawable.a11,
        R.drawable.a12, R.drawable.a13, R.drawable.a14};

    GridView grid01; 2 usages
    Button btn01; 2 usages
```

표시할 이미지 정보를 배열에 정의

## □ 코드 분석

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {...});

    grid01 = (GridView) findViewById(R.id.gridView01);

    grid01.setAdapter(new ImageAdapter( context: this));

    btn01 = (Button) findViewById(R.id.button01);
    btn01.setOnClickListener(this);
}
```

```
@Override
public void onClick(View view) {
    finish();
}
```

- 그리드뷰 객체를 사용자가 정의한 커스텀어댑터를 이용하여 연결
- 정의한 커스텀 어댑터인 ImageAdapter와 연결
- 커스텀어댑터 정의는 다음페이지의 BaseAdapter 상속 코드 참조 (내부 클래스로 생성)

## □ 코드 분석

```
private class ImageAdapter extends BaseAdapter { 1 usage
```

BaseAdapter를 상속받는  
ImageAdapter 클래스

```
private Context mContext; 2 usages
```

```
public ImageAdapter(Context context) { 1 usage  
    mContext = context;  
}
```

생성자로 Context 초기화  
Context : 애플리케이션 실행을 위해  
필요한 기본 정보들과 환경

```
@Override  
public int getCount() {  
    return mThumbIds.length;  
}
```

데이터의 총 개수

```
@Override  
public Object getItem(int i) {  
    return null;  
}
```

위치의 항목 리턴

```
@Override  
public long getItemId(int i) {  
    return mThumbIds[i];  
}
```

position 위치의 ID 리턴

## □ 코드 분석

@Override

public View getView(int i, View view, ViewGroup viewGroup) {

ImageView imageView01;      화면에 표시할 항목의 번호, 해당 위치에 사용중인

if(view == null) {      화면 표시 위젯, 항목 표시를 위한 부모 위젯

imageView01 = new ImageView(mContext);

imageView01.setLayoutParams(new GridView.LayoutParams( w: 85, h: 85));

imageView01.setScaleType(ImageView.ScaleType.CENTER\_CROP);      보여줄 이미지의

imageView01.setPadding( left: 8, top: 8, right: 8, bottom: 8);      폭과 너비를

} else {      이미지 패딩 사이즈      85 픽셀로...

imageView01 = (ImageView)view;

}

imageView01.setImageResource(mThumbIds[i]);

return imageView01;

실제 출력할 이미지 셋팅

}

}

}

## □ *BaseAdapter*

### ○ getCount()

- Adapter에 몇 개의 항목이 있는지 확인

### ○ getItem()

- 위치의 항목을 조사하여 배열의 몇 번째 요소인지를 리턴

### ○ getItemId()

- Position 위치의 ID 리턴, 첨자 자체가 고유 ID이므로 첨자를 그대로 리턴

### ○ getView()

- 항목 하나를 출력하기 위한 뷰를 생성하여 리턴
- 커스텀 어댑터를 정의할 때, 이 메소드를 재정의하여 커스텀 레이아웃을 생성
- 항목 뷰를 생성하는 가장 핵심적인 메소드

## □ 테마 변경이 필요한 경우

### ○ 여러 속성값이 존재, 필요에 따라 사용

- 여기서는 사용법만 설명
- 변경 테마는 Theme.AppCompat.Light.NoActionBar 사용

### ○ 준비

- 리소스의 themes.xml(or styles.xml)에 사용할 스타일 추가 지정
  - ❖ 색상 설정, 테마별 설정 등 추가적인 작업을 해야 함
- AndroidManifest.xml 수정





## □ *AndroidManifest.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/NoActionBarTheme"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

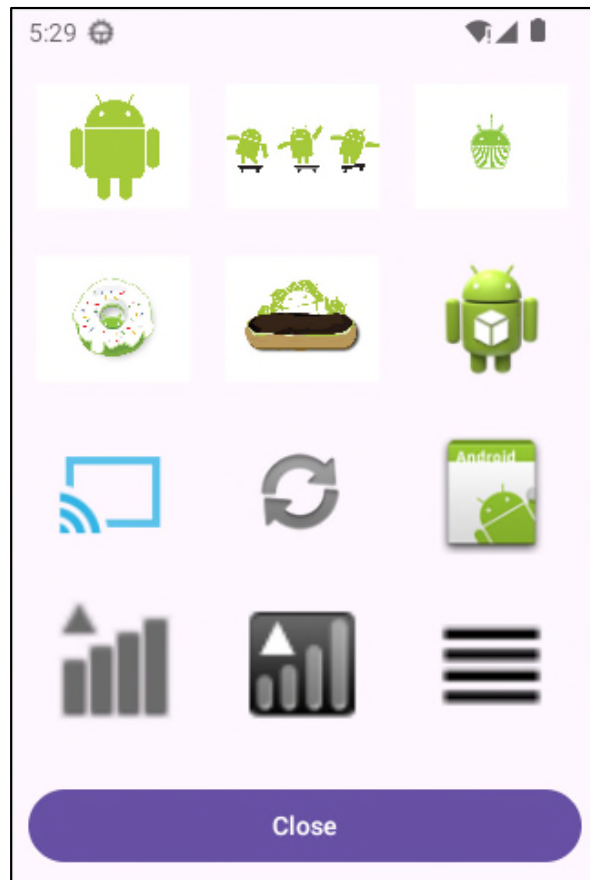
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

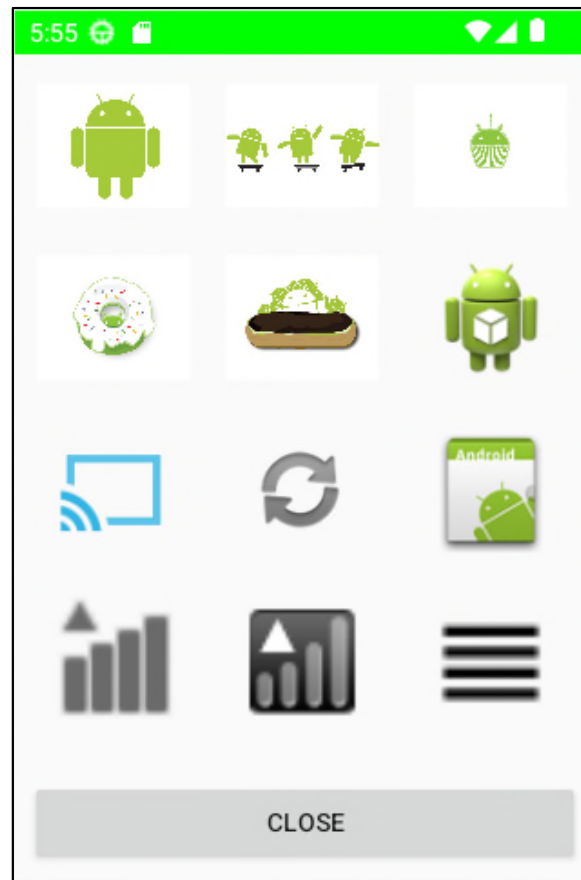
지정한 스타일 적용



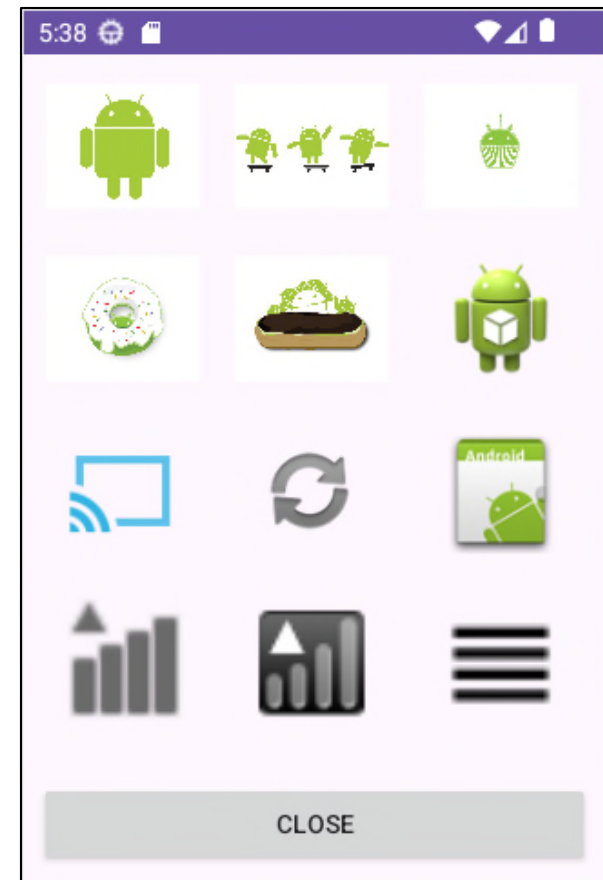
## □ 테마 변경 스타일 적용 전,후 수행 모습



AppCompatActivity 상속시  
EdgeToEdge 사용 O



Theme 변경 예  
EdgeToEdge 사용 X



Activity 상속시  
EdgeToEdge 기능 사용 X

색상 설정 등, 필요한 사항은 별도 설정

# **□ 안드로이드 기능/실습 - 16**

## **- 이미지뷰(IMAGEVIEW)**

## □ *ImageView*

### ○ 목적

- GridView와 BaseAdpater 사용법 연습 및 ImageView 사용법 습득

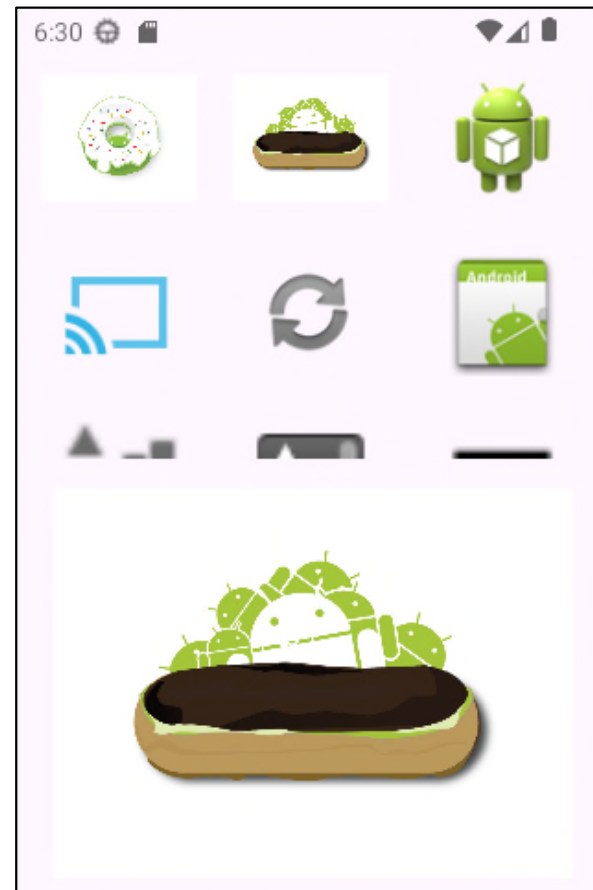
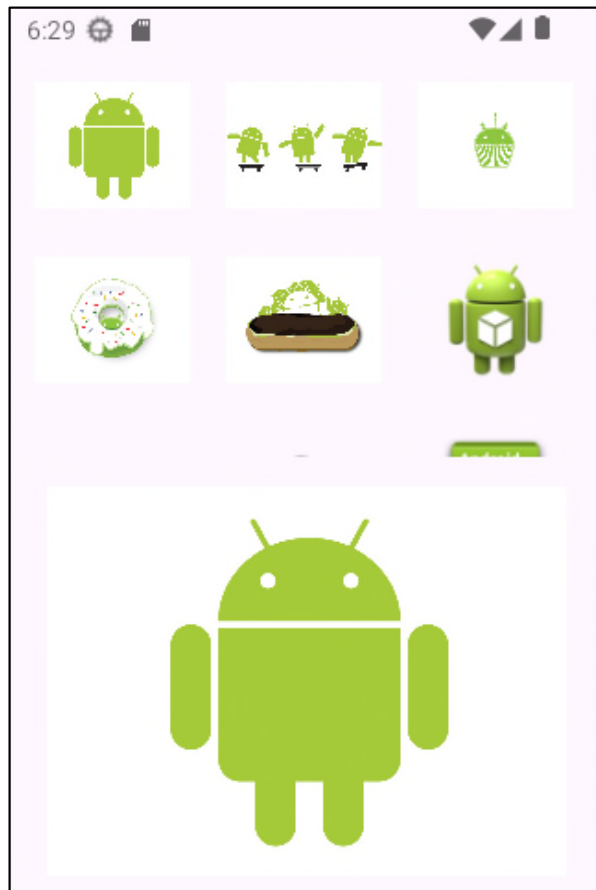
### ○ ImageView

- 아이콘 등과 같은 임의의 이미지를 출력함
- TextView 클래스를 확장한 것으로 이미지를 표시할 수 있는 TextView라고 생각할 수 있음

### ○ 속성

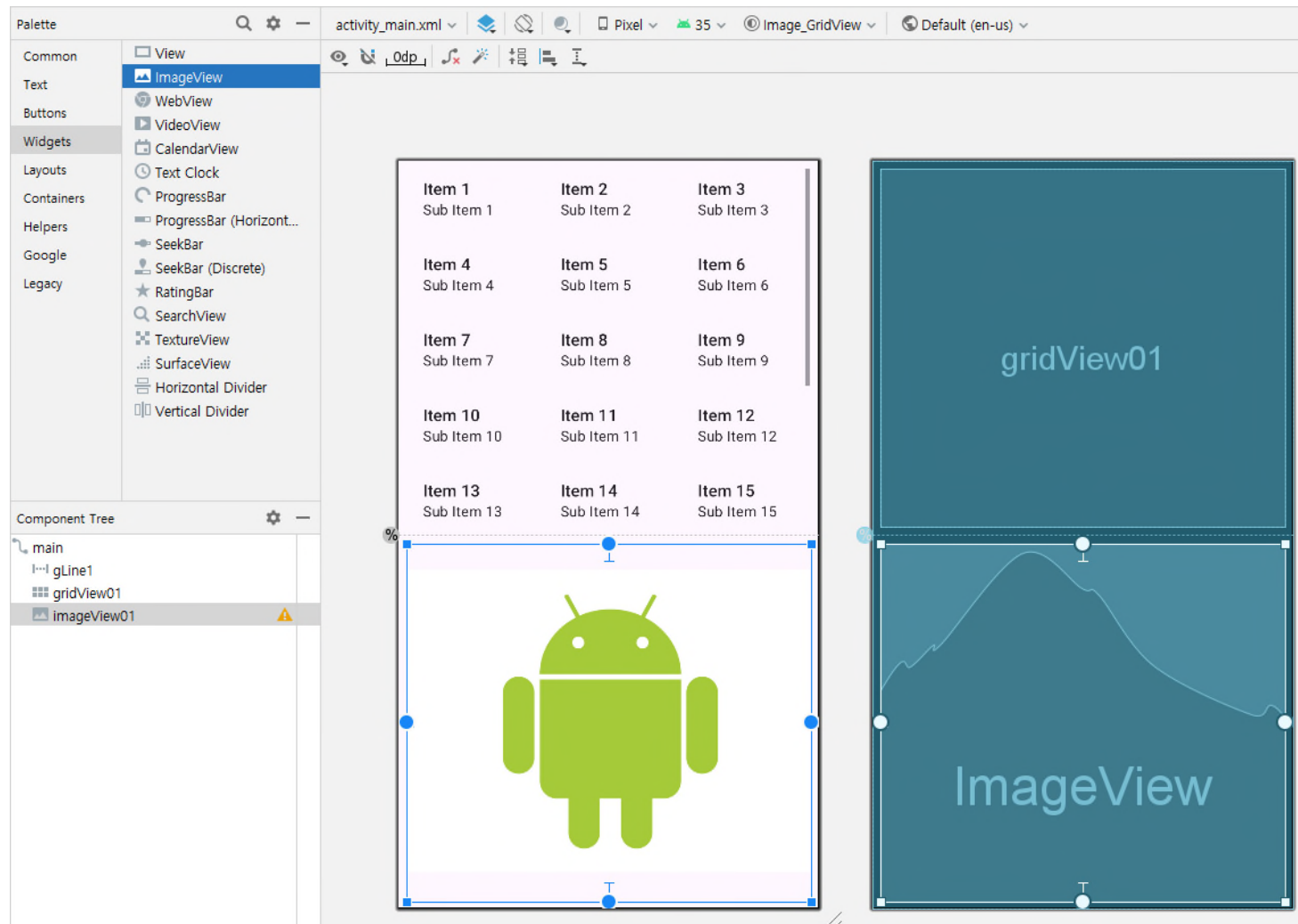
- android:src , app:srcCompat
  - ❖ 출력 대상이 되는 이미지 명 입력
  - ❖ drawable 폴더에 있는 이미지를 지정함
- android:tint
  - ❖ 이미지 배경 색상
- ...

## □ 실행 모습



## □ 레이아웃 구성

- 1개의 GridView와 1개의 ImageView로 구성
- Guideline을 이용하여 50% 씩 공간 배정



## □ 레이아웃

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/gLine1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.50" />
```

## □ 레이아웃

```
1 <GridView
    android:id="@+id/gridView01"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:columnWidth="90dp"
    android:gravity="center"
    android:horizontalSpacing="10dp"
    android:numColumns="3"
    android:stretchMode="columnWidth"
    android:verticalSpacing="10dp"
    app:layout_constraintBottom_toTopOf="@+id/gLine1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

기능/실습 15번 및  
해보기/실습 참조

## □ 레이아웃

```
<ImageView
    android:id="@+id/imageView01"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/gLine1"
    app:srcCompat="@drawable/a1" />
```

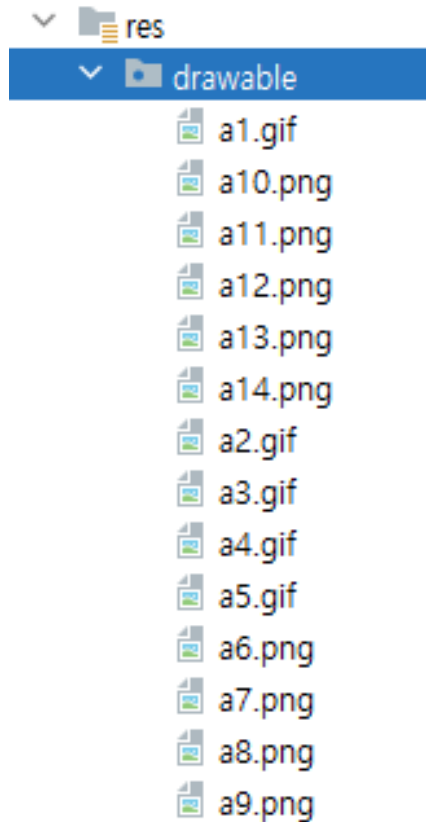
초기 이미지 설정이 반드시 필요

```
</androidx.constraintlayout.widget.ConstraintLayout>
```



## □ Resource 사전 작업

- 표시될 이미지를 res/drawable에 복사
- 이미지는 수업자료 게시판
  - 기능/실습 15와 같은 이미지 사용



## □ 코드 분석

```
package com.practice.ex.image_gridview;
```

```
import android.content.Context;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.view.ViewGroup;
```

```
import android.widget.AdapterView;
```

```
import android.widget.BaseAdapter;
```

```
import android.widget.GridView;
```

```
import android.widget.ImageView;
```

```
import androidx.activity.EdgeToEdge;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.core.graphics.Insets;
```

```
import androidx.core.view.ViewCompat;
```

```
import androidx.core.view.WindowInsetsCompat;
```

선택된 아이템 처리를 위함

```
public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {
```

```
Integer[] mThumbIds = {R.drawable.a1, R.drawable.a2, R.drawable.a3, R.drawable.a4, 4 usages  
    R.drawable.a5, R.drawable.a6, R.drawable.a7, R.drawable.a8,  
    R.drawable.a9, R.drawable.a10, R.drawable.a11, R.drawable.a12,  
    R.drawable.a13, R.drawable.a14};
```

```
GridView gridView01; 3 usages
```

```
ImageView grid_image01; 2 usages
```

표시할 이미지 정보를 배열에 정의

## □ 코드 분석

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {...});

    gridView01 = (GridView)findViewById(R.id.gridView01);
    grid_image01 = (ImageView)findViewById(R.id.imageView01);

    gridView01.setAdapter(new ImageAdapter( context: this));

    gridView01.setOnItemClickListener(this);
}
```

커스텀 어댑터인 ImageAdapter에 연결

아이템이 선택되었을 때 처리를 위해 리스너에 등록

```
@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
    grid_image01.setImageResource(mThumbIds[i]);
}
```

선택된 아이템에 따른 처리

이미지 뷰 객체에 setImageResource()를 이용하여 이미지 배정

## □ 코드 분석

```
} private class ImageAdapter extends BaseAdapter { 1 usage
|
|     private Context mContext; 2 usages
|
|     public ImageAdapter(Context context) { 1 usage
|         mContext = context;
|     }
|
|     @Override
|     public int getCount() {
|         return mThumbIds.length;
|     }
|
|     @Override
|     public Object getItem(int i) {
|         return null;
|     }
|
|     @Override
|     public long getItemId(int i) {
|         return mThumbIds[i];
|     }
| }
```

기능/실습 15 참조

## □ 코드 분석

```
@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    ImageView imageView01;
    if(view == null) {
        imageView01 = new ImageView(mContext);
        imageView01.setLayoutParams(new GridView.LayoutParams(w: 85, h: 85));
        imageView01.setScaleType(ImageView.ScaleType.CENTER_CROP);

        imageView01.setPadding(left: 8, top: 8, right: 8, bottom: 8);
    } else {
        imageView01 = (ImageView)view;
    }
    imageView01.setImageResource(mThumbIds[i]);
    return imageView01;
}
```

기능/실습 15 참조