모바일프로그래밍
- 안드로이드(Android) -

- 어댑터(Adapter)

- 안드로이드 기능/실습 – 13
  - ListView 구성

- 안드로이드 기능/실습 – 14
  - Spinner(XML)

❑ 어댑터(ADAPTER)

❑ Adapter

- 어댑터(Adapter)
  - 외부 데이터 소스와 어댑터뷰(AdapterView) 사이의 연결 수단을 의미
  - 어댑터는 여러 계층으로 구성
    - ❖ 리스트어댑터, 스피너어댑터, 베이스어댑터…
  - * 리사이클러뷰 관련 내용은 별도 진행

- 데이터 소스
  - DB, XML, 배열 등…

- 어댑터뷰
  - ViewGroup의 서브클래스
  - GridView, ListView, Gallery, Spinner …
  - 어댑터를 이용하여 데이터로 레이아웃을 구성
  - 사용자에 의한 항목 선택 이벤트 처리

❑ ArrayAdapter

- 선택 위젯에서 사용할 값들의 목록을 설정하기 위한 공통 인터페이스로 어댑터를 제공

- 어댑터는 어댑터뷰와 데이터 소스 사이를 연결하고 데이터 항목을 관리

- 어댑터는 여러 계층으로 구성 (리스트어댑터, 스피너어댑터, 베이스어댑터…)

- 데이터소스가 배열인 경우 ArrayAdapter
- 데이터베이스인 경우 CursorAdapter
- XML을 포함한 정적 데이터인 경우 SimpleAdapter 사용

❑ ArrayAdapter

- ArrayAdapter는 자바의 배열이나 java.util.List의 인스턴스를 주는 것으로 어댑터를 사용하기 위한 준비가 끝남

```
String[] items = {“item 1”, “item 2”, “item 3”};

ArrayAdapter<String> adapter = new ArrayAdapter<String>(
this,
android.R.layout.simple_list_item_1,
items
);
```

- 어댑터가 포함될 인스턴스 지정, 항목을 표시하기 위해 사용할 뷰의 리소스 ID(내장된 리소스 ID), 화면에 표시할 실제 내용이 들어있는 배열 지정

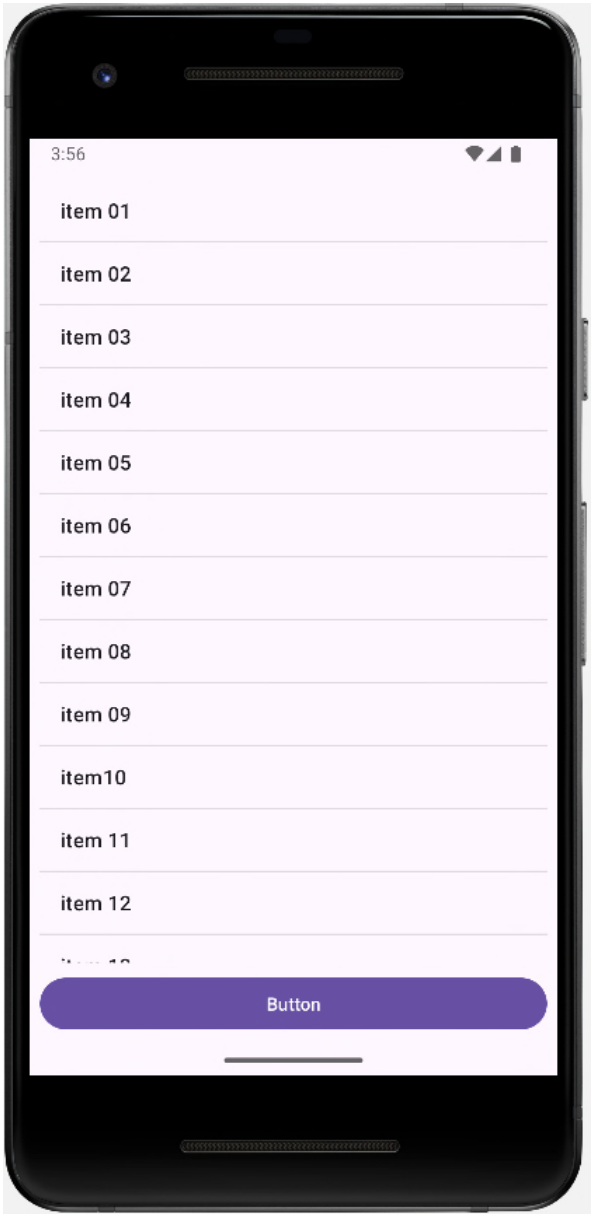
❑ 레이아웃 리소스 ID

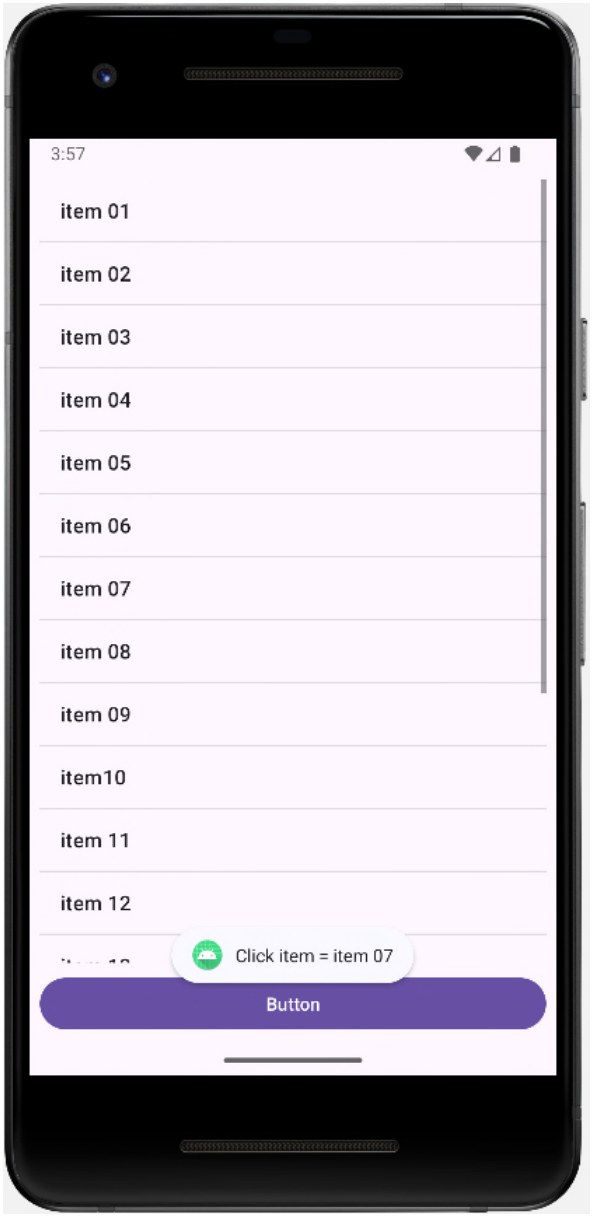
- simple_list_item_1
  - 하나의 텍스트 뷰로 구성된 레이아웃
- simple_list_item_checked
  - 체크가 표시되는 레이아웃
- simple_list_item_single_choice
  - 라디오 버튼이 표시되는 레이아웃
- simple_list_item_multiple_choice
  - 체크 버튼이 표시되는 레이아웃
- …

❑ 안드로이드 기능/실습 - 13
LISTVIEW 구성

- 목적
  - 어댑터 사용 방법 습득 및 어댑터뷰(리스트뷰, 그리드뷰, 스피너, 등)의 사용법을 습득
- ListView
  - 안드로이드에서 리스트박스를 구현하기 위하여 제공해주는 클래스
- 선택위젯
  - 어댑터(Adapter)를 이용하여 데이터 연결
  - 특징
    - ❖ 배열 리스트, 데이터베이스 내의 데이터 등 여러 개의 데이터 항목들을 나타내 줌

❑ 실행 모습

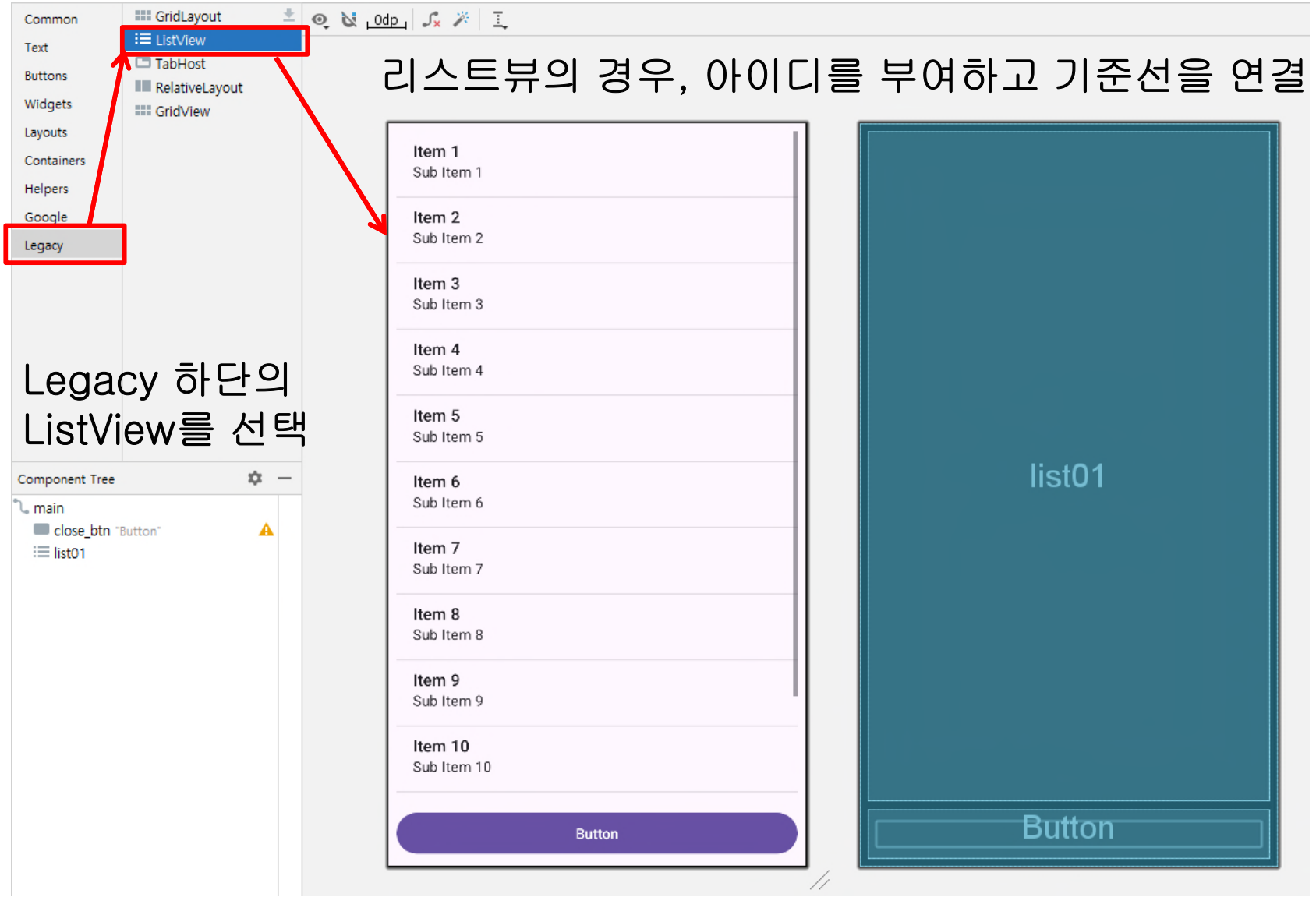

3:56
item 01
item 02
item 03
item 04
item 05
item 06
item 07
item 08
item 09
item10
item 11
item 12
Button


3:57
item 01
item 02
item 03
item 04
item 05
item 06
item 07
item 08
item 09
item10
item 11
item 12
Click item = item 07
Button

❑ 진행 순서

- 레이아웃에서 ListView 추가
- ListView 객체를 얻어 옴
- ListView 객체로 setAdapter() 메소드의 매개변수로 화면에 출력하고자 하는 데이터를 지정하여 호출
- 리스너와 연결
  - setOnItemClickListener() 메소드로 리스너와 연결
  - 선택한 항목을 확인

❑ 레이아웃 구성


Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
GridLayout
ListView
TabHost
RelativeLayout
GridView
0dp
리스트뷰의 경우, 아이디를 부여하고 기준선을 연결
Legacy 하단의
ListView를 선택
Component Tree
main
close_btn "Button"
list01
Item 1
Sub Item 1
Item 2
Sub Item 2
Item 3
Sub Item 3
Item 4
Sub Item 4
Item 5
Sub Item 5
Item 6
Sub Item 6
Item 7
Sub Item 7
Item 8
Sub Item 8
Item 9
Sub Item 9
Item 10
Sub Item 10
Button
list01
Button

❑ 레이아웃 구성

```xml
<?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/close_btn"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

❑ 레이아웃 구성

```
    <ListView
        android:id="@+id/list01"
        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_toTopOf="@+id/close_btn"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

❑ 코드 분석

```
package com.practice.ex.listview;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

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,
                                                               AdapterView.OnItemClickListener {

    private String[] list_items = {"item 01", "item 02", "item 03", "item 04", "item 05",   2 usages
            "item 06", "item 07", "item 08", "item 09", "item10", "item 11", "item 12",
            "item 13", "item 14", "item 15", "item 16", "item17", "item 18", "item 19"};

    Button btn01;  2 usages
    ListView lv;  3 usages
```

선택된 아이템 처리

화면에 표시할 데이터를 배열로 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) -> {...});

    btn01 = (Button) findViewById(R.id.close_btn);
    btn01.setOnClickListener(this);

    lv = (ListView) findViewById(R.id.list01);

    ArrayAdapter<String> adapter = new ArrayAdapter<String>
            ( context: this, android.R.layout.simple_list_item_1, list_items);

    lv.setAdapter(adapter);

    lv.setOnItemClickListener(this);
}

@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
    Toast.makeText( context: this,  text: "Click item = " + list_items[i], Toast.LENGTH_LONG).show();
}

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

어댑터 설정
– ArrayAdapter 설명 참조

리스트뷰 화면에 출력하고자 하는 대상 지정

리스트에서 선택된 항목을 알아내기 위하여
아이템 처리 리스너와 연결

선택된 아이템 처리

position, id – 위치를 나타냄

- 클릭된 항목의 부모 뷰인 어댑터 뷰
- 사용자가 클릭한 항목에 해당되는 뷰
- 선택된 항목의 위치
- Position과 동일

❑ 안드로이드 기능/실습 - 14 SPINNER

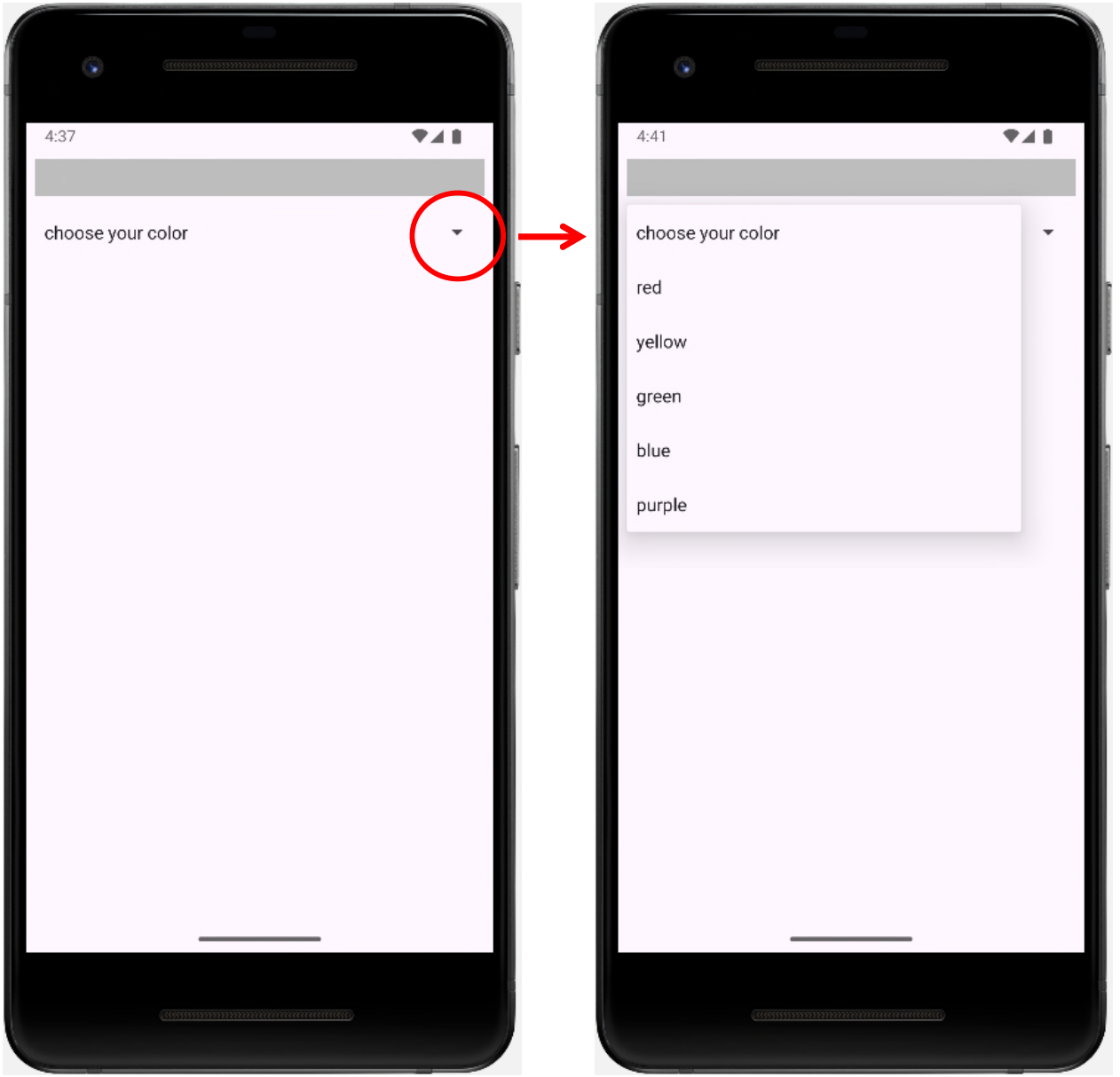
❑ Spinner

- 목적
  - AdapterView 가운데 스피너(Spinner) 사용법을 습득하고, 배열 리소스를 지정하여 사용하는 방법을 습득

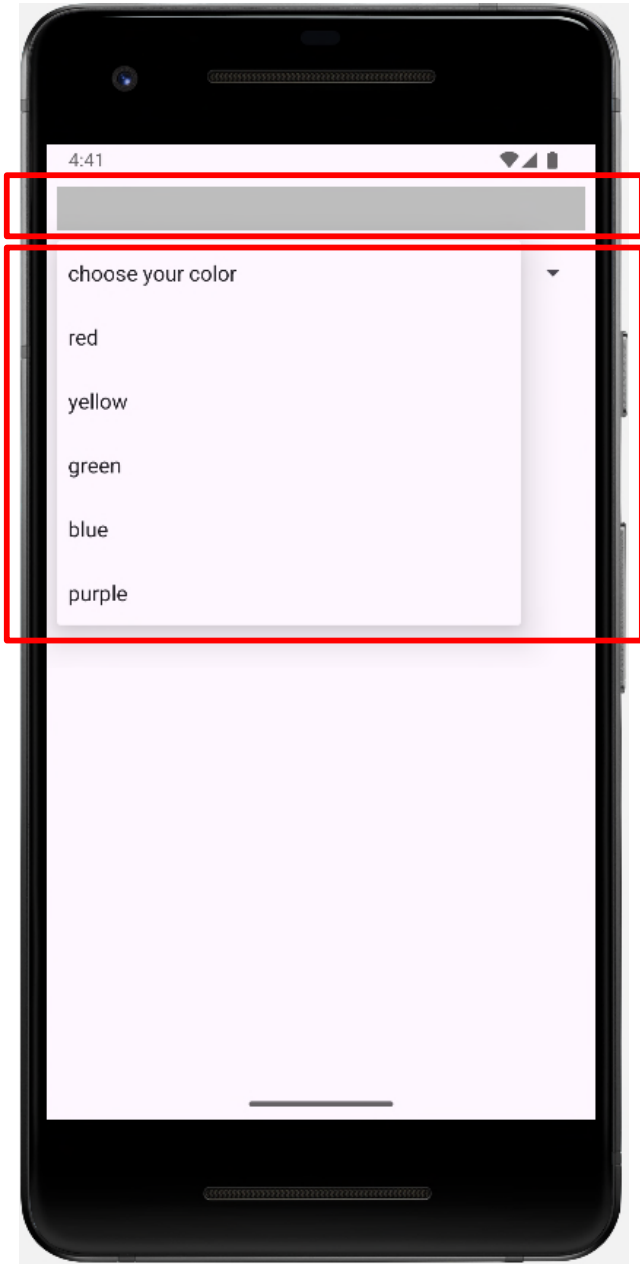
- Spinner
  - 안드로이드에서 드롭다운 선택 기능을 구현하기 위하여 제공하는 위젯
  - ListView와 마찬가지로 setAdapter() 메소드의 매개변수로 화면에 출력하고자 하는 데이터 지정이 가능

- 여기서는 setAdapter() 대신 레이아웃에서 android:entries 속성에 배열 리소스를 지정하여 사용 하는 방법을 이용 (XML 이용법)
  - setAdapter()를 이용하는 방법은 기능/실습 13 참조

❑ 실행 모습


4:37
choose your color
4:41
choose your color
red
yellow
green
blue
purple

❑ 레이아웃 구성



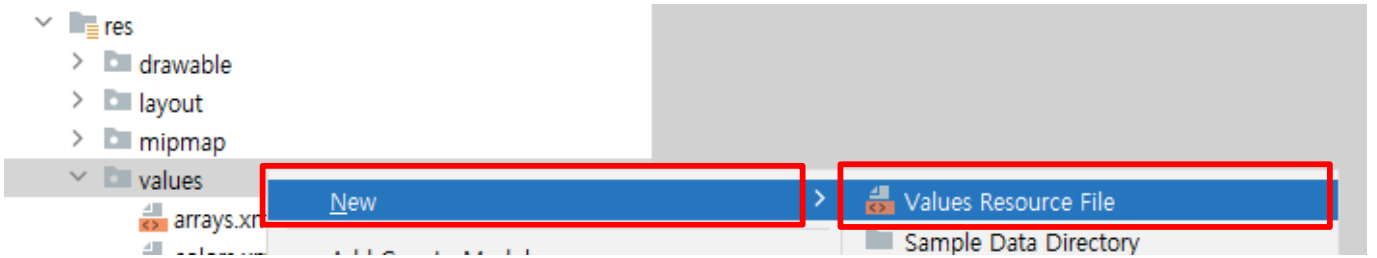
TextView

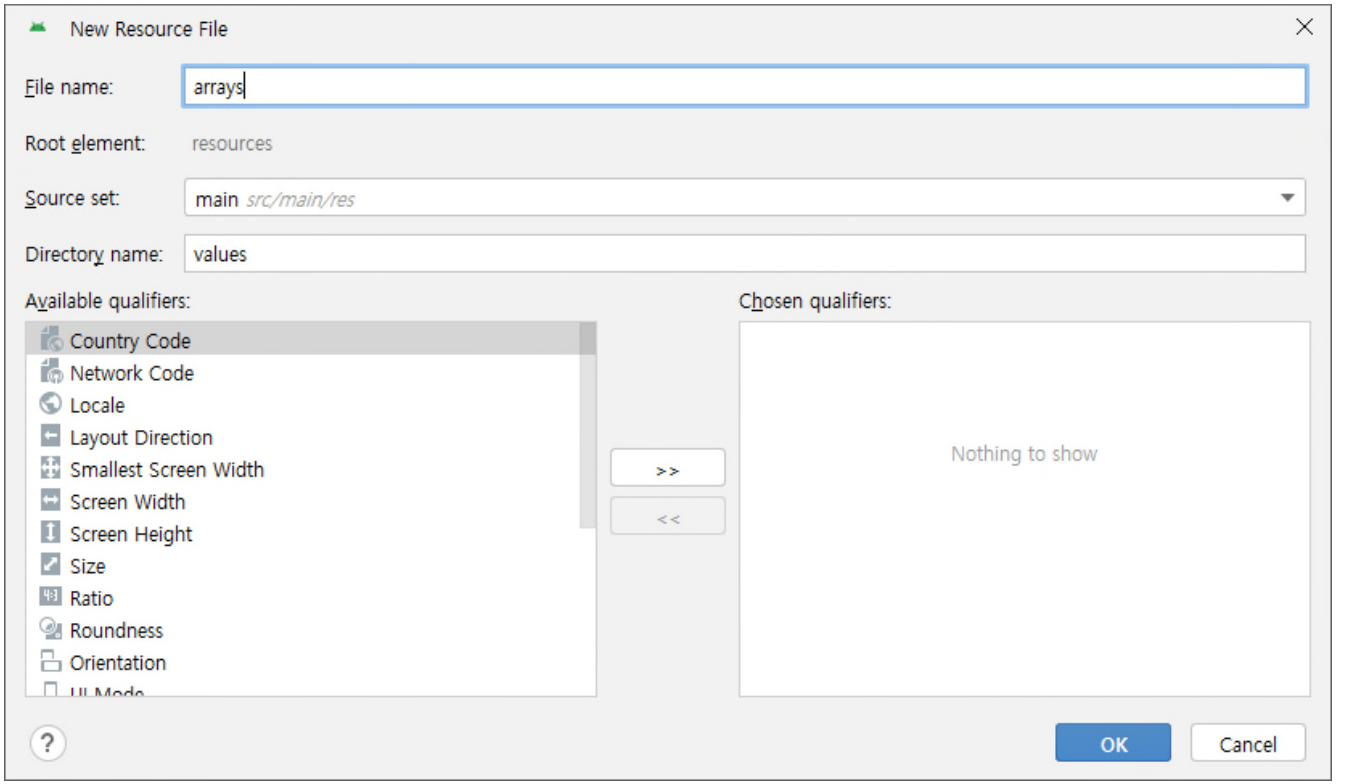
Spinner

스피너에서 선택한 항목의 이름이 상단의 텍스트뷰에 출력

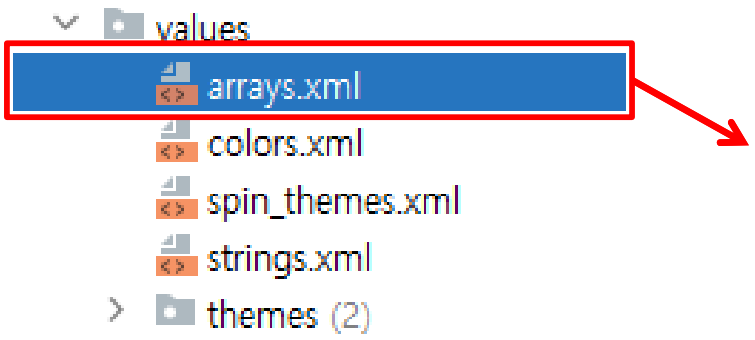
❍ 배열 자료를 준비하기 위해서 arrays.xml 파일 추가

❑ Values Resource File 선택





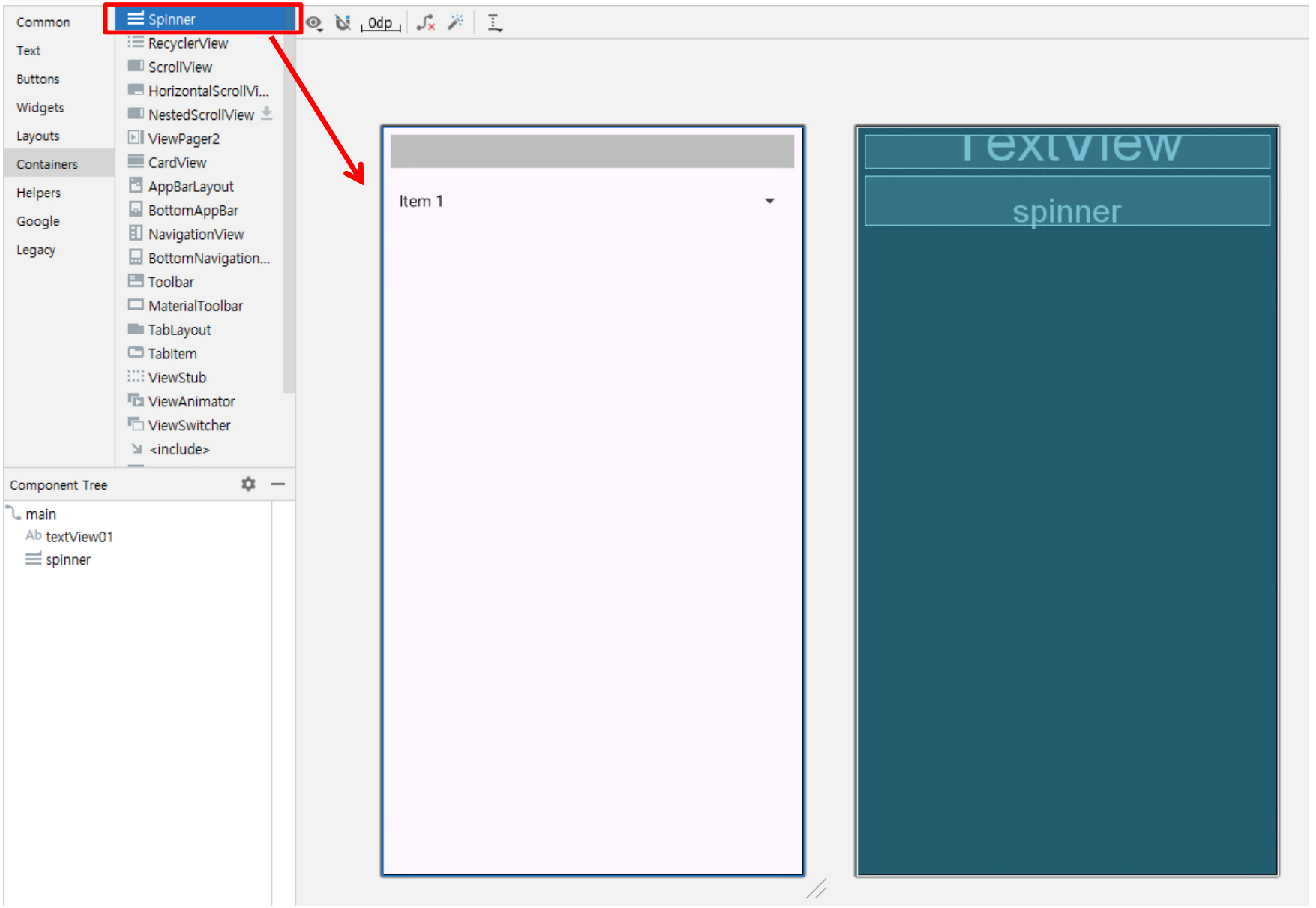
❍배열 자료를 준비하기 위해서 arrays.xml 파일 추가



arrays.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="colors">
        <item>choose your color</item>
        <item>red</item>
        <item>yellow</item>
        <item>green</item>
        <item>blue</item>
        <item>purple</item>
    </string-array>
</resources>
```

❑ 레이아웃 구성


Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Spinner
RecyclerView
ScrollView
HorizontalScrollVi...
NestedScrollView
ViewPager2
CardView
AppBarLayout
BottomAppBar
NavigationView
BottomNavigation...
Toolbar
MaterialToolbar
TabLayout
TabItem
ViewStub
ViewAnimator
ViewSwitcher
<include>
0dp
Component Tree
main
textView01
spinner
Item 1
TextView
spinner

❑ 레이아웃 구성

```
<?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">

    <TextView
        android:id="@+id/textView01"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:background="#FFBDBDBD"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

❑ 레이아웃 구성

```
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:entries="@array/colors"
        android:minHeight="48dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView01" />

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

android:entries
화면에 출력할 항목을 지정

❑ 코드 분석

```
package com.practice.ex.spinner;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Spinner;
import android.widget.TextView;

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 AdapterView.OnItemSelectedListener {

    Spinner spin;  4 usages
    TextView text01;  3 usages
    int count = 0;  1 usage

    @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) -> {...});

        spin = (Spinner)findViewById(R.id.spinner);
        text01 = (TextView)findViewById(R.id.textView01);

        spin.setOnItemSelectedListener(this);
    }
```

아이템 선택을 처리하기 위한 리스너
Spinner에서 아이템 처리는
OnItemSelectedListener를 사용

선택한 리스트 항목을
알아내기 위해 리스너에 연결

❑ 코드 분석

항목선택시 호출되는 메소드

```
@Override  1 usage
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
    if(count != spin.getSelectedItemId()) {
        TextView temp = (TextView) spin.getSelectedView();
        text01.setText(temp.getText());
    } else {
        text01.setText("");
    }
}

@Override  1 usage
public void onNothingSelected(AdapterView<?> adapterView) {

}
}
```

선택항목에서 데이터 추출할 때

연결 해제 시 호출되는 메소드