

# 21

## CHAPTER

# 입력 컨트롤과 데이터 전송

### 21.1 입력 컨트롤과 데이터 전송 원리

#### 21.1.1 입력 컨트롤 활용 앱의 예

#### 21.1.2 메뉴의 유형 개요

#### 21.1.3 입력 컨트롤 유형

#### 21.1.4 데이터 전송 원리

### 21.2 입력 컨트롤을 이용한 화면 구성: Contents Mgt 프로젝트(갤러리)

#### 21.2.1 프로젝트 개요

#### 21.2.2 프로젝트 개발

#### 21.2.3 홈 화면 구성

#### 21.2.4 액션 바 메뉴와 '사진 등록' 액티비티 실행

#### 21.2.5 '사진 등록' 액티비티의 화면 구성

#### 21.2.6 사진 등록을 위한 카메라 사진 선택

#### 21.2.7 입력 컨트롤들의 입력 값 추출과 알림창 띄우기

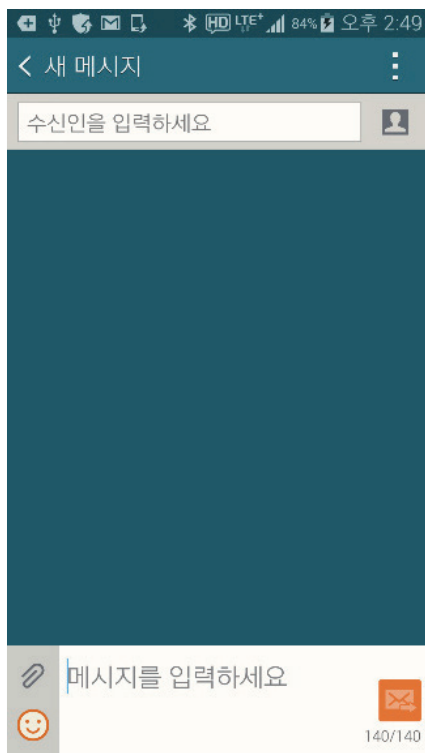
#### 21.2.8 입력 컨트롤들의 입력 값 추출과 정보전달

# 21 1

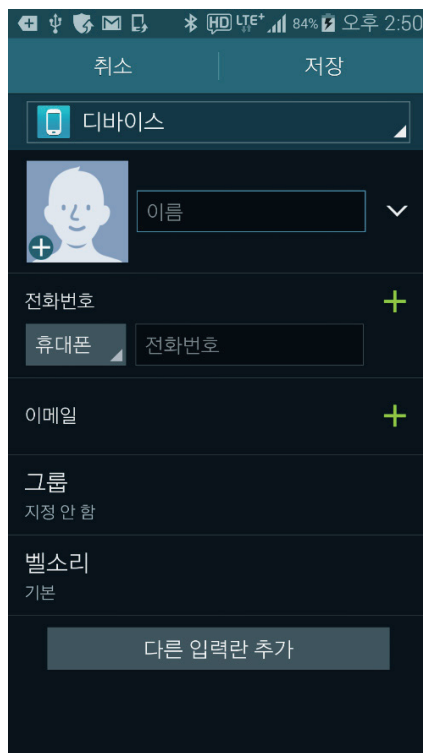
## 입력 컨트롤과 데이터 전송 원리

### 21.1.1 입력 컨트롤 활용 앱의 예

스마트폰이 널리 보편화되고 있는 이유 중의 하나는 사람들과 상호작용하는 유용한 컴포넌트를 가지고 있기 때문이다. 입력 컨트롤(input control)은 앱의 UI에서 스마트폰과 상호작용하는 컴포넌트로서, **버튼**, **텍스트 입력과 출력**, **체크 박스**, **줌 버튼** 등 많은 기능을 가지고 있다. 우리는 입력 컨트롤을 이용하여 다른 사람들과 정보를 원활하게 공유하고 실생활에 밀접한 정보도 편리하게 관리한다. 아래 예는 우리가 수시로 주고 받는 메시지를 작성하거나 지인들의 연락처를 저장할 때 사용하는 앱으로 정보 입력과 버튼 등으로 이루어진 입력 컨트롤을 사용하는 예를 보여준다.



(a) 메시지 작성 화면


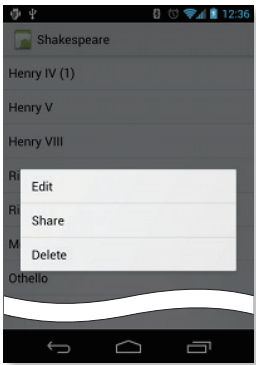


(b) 연락처 등록 화면

- 입력 컨트롤을 이용하는 앱


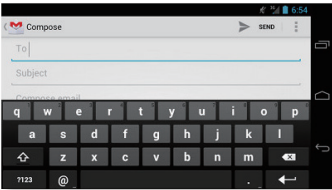
### 21.1.2 메뉴의 유형 개요

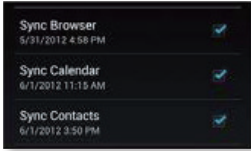
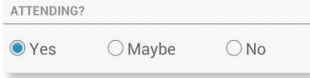
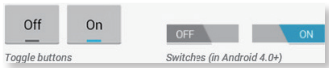
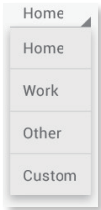
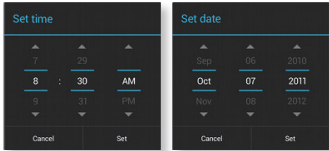
메뉴는 여러 유형의 어플리케이션에서 일상적인 UI 컴포넌트이다. 메뉴 유형은 다음과 같다.

메뉴 유형	설명
Options menu와 action bar	 <p>Options menu는 액티비티를 위한 기본적인 메뉴 아이템들임. 메뉴 아이템은 <b>액션 바(action bar)</b>에 나타남. 액션 바는 [1] app icon, [2] action item, [3] action overflow 들을 포함함.</p>
Context menu	 <p>context menu는 사용자가 엘리먼트를 길게 클릭하면 나타나는 floating menu임</p>
Popup menu	메뉴를 작동시킨 뷰에 고정된 수직 리스트

### 21.1.3 입력 컨트롤 유형

우리가 사용할 수 있는 일반적인 입력 컨트롤의 유형은 다음과 같으며(developer.android.com), 각각 입력 이벤트를 처리하는 기능을 가지고 있다.

컨트롤 유형	설명	관련 클래스
Button	<p>사용자에 의해 눌러지거나 클릭되는 푸쉬 버튼</p> 	Button
Text field	<p>편집 가능한 입력박스. 한 글자만 입력하면 자동으로 단어가 완성되는 기능은 AutoCompleteTextView 를 사용하면 됨</p> 	EditText, AutoCompleteTextView

Checkbox	<p>사용자에 의해 토글되는 on/off 스위치. 여러 아이템을 그룹으로 묶어 다중 선택이 가능할 때 사용함,</p> 	CheckBox
Radio button	<p>그룹에서 하나의 아이템만 선택 가능하도록 할 때 사용함</p> 	RadioGroup, RadioButton
Toggle button	<p>on/off 버튼</p> 	ToggleButton
Spinner	<p>목록에서 한 아이템을 선택하도록 할 때 사용하는 드롭다운(drop-down) 리스트</p> 	Spinner
Picker	<p>up/down 버튼 등을 사용하여 한 가지 값을 선택할 때 사용하는 대화상자. 날짜(연, 월, 일)에 대한 값을 선택하는 DatePicker, 시간(시, 분, 오전/오후)을 선택할 때 사용하는 TimePicker 등이 있음</p> 	DatePicker, TimePicker

#### 21.1.4 데이터 전송 원리

액티비티 간 데이터는 **인텐트**를 통해 전달한다. 데이터 전달은 호출할 액티비티 자바 클래스에 관한 인텐트 객체를 만들고, 인텐트 객체에 전달할 데이터를 저장 후에 액티비티 자바 클래스를 호출한다. 호출된 액티비티 자바 클래스는 인텐트 객체를 찾아 데이터를 추출한다. 액티비티 간 인텐트를 통한 데이터 전송에 관한 자세한 내용은 **8장**을 참고하기 바란다.

# 21 ②

## 입력 컨트롤을 이용한 화면 구성: Contents Mgt 프로젝트(갤러리)

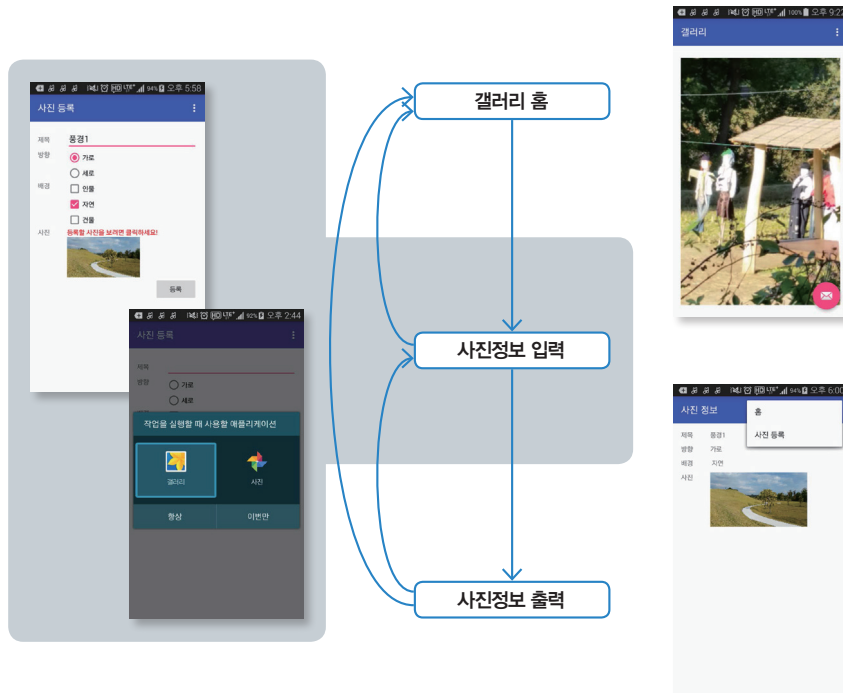
### 21.2.1 프로젝트 개요

입력 컨트롤 중에서 기본적인 컴포넌트를 이용해 카메라로 찍은 사진정보를 입력하면 입력한 결과를 출력하는 갤러리 앱을 만들어 보자. 아래 그림은 갤러리 홈 화면에서 사진정보 입력을 거쳐 사진정보를 출력하는 과정의 흐름을 나타낸 화면 흐름도이다. 사진정보 입력 화면에서 입력 컨트롤을 다루는 방법을 단계적으로 살펴보기로 한다.

프로젝트 개요: 입력 컨트롤을 이용한 화면 구성

Application Name: Contents Mgt

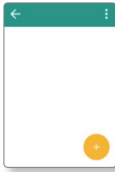
어플리케이션 라벨: 갤러리



## 21.2.2 프로젝트 개발

### STEP 1 프로젝트 생성

다음의 5단계 절차에 따라 프로젝트를 실행한다.

절차	내용
❶ 프로젝트 시작	메뉴에서 'File → New Project' 클릭
❷ 프로젝트 구성	Application Name: <b>Contents Mgt</b> Company Domain: <b>yschang.example.com</b>
❸ 제품형태	Phone and Tablet
❹ 액티비티 유형	 Blank Activity
❺ 액티비티 구성	디폴트 값으로 설정 Activity Name: <b>MainActivity</b> Layout Name: <b>activity_main</b> Title: <b>MainActivity</b> Menu Resource Name: <b>menu_main</b>

### STEP 2 파일 편집

#### ● 파일 구조와 기능

모듈	폴더	소스 파일	편집 내용	관련 액티비티
<b>manifests</b>		AndroidManifest.xml	<ul style="list-style-type: none"> <li>ContentsRegForm 클래스 등록</li> <li>ContentsDetail 클래스 등록</li> </ul>	갤러리홈, 사진정보 입력, 사진정보 출력
java	com.example.yschang.contentsmgt	MainActivity.java	<ul style="list-style-type: none"> <li>이미지 출력</li> <li>메뉴 아이템 구성</li> </ul>	갤러리홈
		ContentsRegForm.java	<ul style="list-style-type: none"> <li>사진정보 입력화면</li> <li>사진정보 전송</li> </ul>	사진정보 입력
		ContentsDetail.java	<ul style="list-style-type: none"> <li>사진정보 수신</li> <li>사진정보 출력</li> </ul>	사진정보 출력

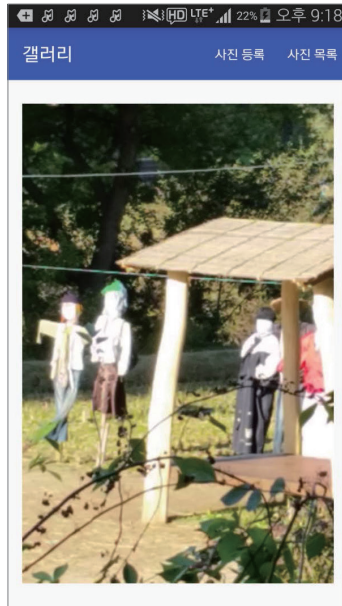
res	drawable	home.png	• 메인 화면 이미지	갤러리홈
	layout	activity_main.xml	• 메인 화면 구성 틀	갤러리홈
		content_main.xml	• 메인 화면의 배경 이미지 배치	갤러리홈
		activity_reg_form.xml	• 사진정보 입력화면 구성 틀	사진정보 입력
		contents_reg_form.xml	• 사진정보 입력 컨트롤의 배치	사진정보 입력
		activity_detail.xml	• 사진정보 출력화면 구성 틀	사진정보 출력
		content_detail.xml	• 사진정보 출력화면 배치	사진정보 출력
	menu	menu_main.xml	• 메인 화면 액션 바의 메뉴 아이템	갤러리홈
		menu_reg_form.xml	• 사진정보 입력화면 액션 바의 메뉴 아이템	사진정보 입력
		menu_detail.xml	• 사진정보 출력화면 액션 바의 메뉴 아이템	사진정보 출력
	values	colors.xml		
		dimens.xml		
		strings.xml	<ul style="list-style-type: none"> <li>• 어플리케이션 라벨 수정</li> <li>• 화면 구성을 위한 문자 자원 정의</li> <li>• 메뉴 아이템을 위한 문자 자원 정의</li> </ul>	갤러리홈, 사진정보 입력, 사진정보 출력
		styles.xml		

■ 수정 ■ 추가

### 21.2.3 홈 화면 구성

갤러리 홈 화면은 **홈 이미지**를 배치하고 다른 화면 간의 이동에 사용할 우측 상단의 **액션바**는 그대로 둔다.

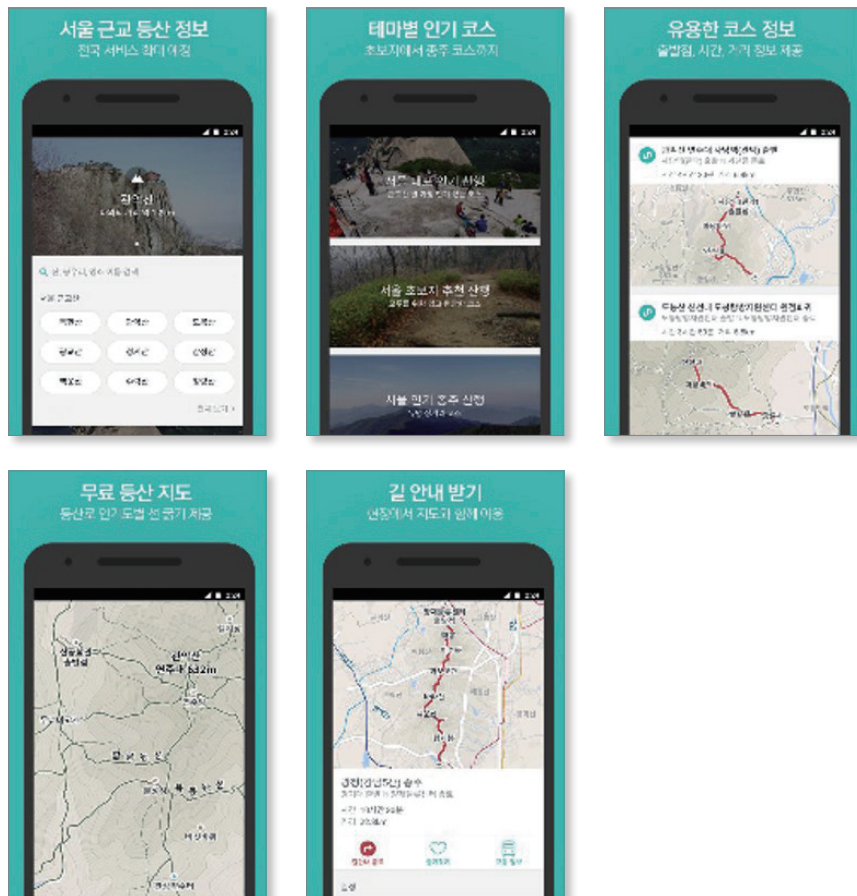
#### 프로젝트



홈 화면



「등산 지도와 인기 코스」 앱은 산림청의 등산로 DB를 활용하여 등산 정보를 제공하며, 서울 근교 35개 산의 인기코스와 맞춤형 코스 정보, 교통·소요시간 등 상세 정보를 제공한다. 주요 기능은 무료 등산 지도, 테마별 인기 코스, 길안내 받기, 카톡으로 코스 공유 등이다. 활용 API는 “[http://www.data.go.kr/comm/file/download.do?atchFileId=FILE\\_000000001287620&fileDetailSn=1](http://www.data.go.kr/comm/file/download.do?atchFileId=FILE_000000001287620&fileDetailSn=1)” 이다.



- 「등산 지도와 인기 코스」 앱의 화면

## STEP 2 > 파일 편집

### ● 파일 간의 연관관계

**strings.xml**에 액티비티 라벨과 메뉴 문자열을 추가한다

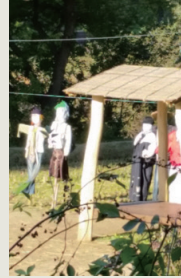
**menu\_main.xml**에는 메뉴 아이템의 id와 title을 수정한다.

DB와 테이블 생성 및 데이터 조작을 위한 **DBManager.java**를 작성하여 관련 클래스에서 사용할 수 있도록 한다.

**MainActivity.java**에는 '사진 등록' 메뉴 클릭 시, 'ContentReg' 클래스를 호출한다.

이미지

추가



home.png (drawable)

메뉴 아이템 리소스

수정

```
menu
    item
        id @+id/action_reg
        title @string/action_reg
    item
        id @+id/action_list
        title @string/action_list
```

menu\_main.xml (menu)

메뉴  
아이템

텍스트 리소스

수정

```
string
    app_name 갤러리
    act_reg_form 사진 등록
    ...
    action_home 홈
    action_reg 사진 등록
    action_list 사진 목록
    ...
    title 제목
    orientation 방향
    ...
```

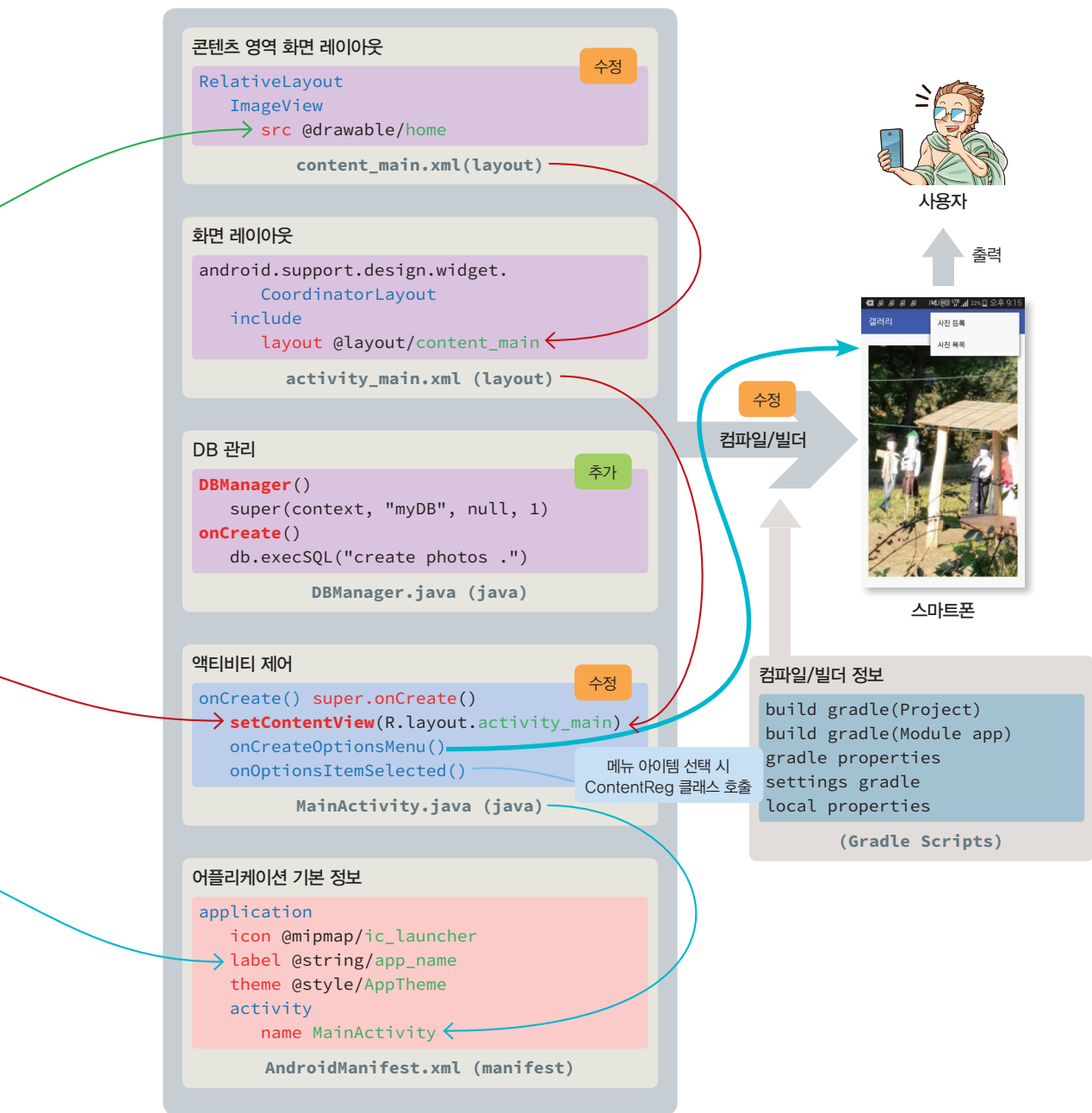
strings.xml (values)

앱 라벨

액티비티 라벨

메뉴 아이템

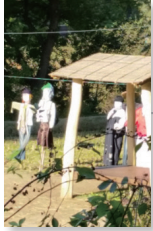
콘텐츠 구성 항목



## ● 편집

### ① 이미지 파일의 복사(res/drawable 폴더)

**drawable** 폴더에 화면에 출력할 이미지를 저장한다.

모듈	폴더	소스 파일	이미지
res	drawable	home.png	

A

### ② 텍스트 자원

**strings.xml**의 **app\_name**을 '갤러리'로 수정하고 액션 바의 메뉴 아이템 자원으로 사용할 **action\_settings**는 그대로 둔다.

소스 | strings.xml

```
01 <resources>
02   <string name="app_name">갤러리</string>
03   <string name="action_settings">Settings</string>
04 </resources>
```

app\_name의 데이터를  
'갤러리'로 수정

B

C

### ③ 메뉴 구성 자원

액션 바의 메뉴 아이템을 구성하는 **menu\_main.xml**의 내용은 그대로 둔다.

소스 | menu\_main.xml

```
01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_settings"
07         android:orderInCategory="100"
08         android:title="@string/action_settings"
09         app:showAsAction="never" />
10 </menu>
```

메뉴 목록

메뉴 아이템

E

C

## 클래스와 속성/메소드

### 클래스

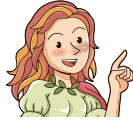
클래스/인터페이스	설명
menu	메뉴 아이템을 포함하는 <b>Menu</b> 클래스를 정의함. 루트 노드로 구성함
item	메뉴 내에 하나의 메뉴 아이템을 표현하는 <b>MenuItem</b> 클래스를 만들

### XML 속성

클래스	속성	설명
menu	tools:context	그래픽 레이아웃의 테마를 사용할 액티비티 클래스
item	android:orderInCategory	메뉴아이템간 우선순위, 낮은 수 일수록 왼쪽에 나타남
	android:showAsAction	메뉴가 액션 바에 언제 어떻게 나타나는지를 기술함
		값
		설명
		ifRoom
		withText
		never
		always
	android:title	메뉴에 표시될 제목

#### 4 화면 설계

프로젝트를 **Blank Activity**로 생성하면 **activity\_main.xml**는 우측 상단에 표시되는 **액션 바**, **콘텐츠 영역**, 콘텐츠 영역 우측 하단에 나타나는 **플로팅 액션 버튼**으로 구성된다.



이번 프로젝트에서 사용하지 않는 **플로팅 액션 버튼**은 삭제한다.

소스 | activity\_main.xml → F

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <android.support.design.widget.CoordinatorLayout
03     xmlns:android="http://schemas.android.com/apk/res/android"
04     xmlns:app="http://schemas.android.com/apk/res-auto"
05     xmlns:tools="http://schemas.android.com/tools"
06     android:layout_width="match_parent"
07     android:layout_height="match_parent"
08     android:fitsSystemWindows="true"
09     tools:context="com.example.contentsmgmt.MainActivity">
10
11     <android.support.design.widget.AppBarLayout
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:theme="@style/AppTheme.AppBarOverlay">
15
16         <android.support.v7.widget.Toolbar
17             android:id="@+id/toolbar"
18             android:layout_width="match_parent"
19             android:layout_height="?attr/actionBarSize"
20             android:background="?attr/colorPrimary"
21             app:popupTheme="@style/AppTheme.PopupOverlay" />
22
23     </android.support.design.widget.AppBarLayout>
24
25     <include layout="@layout/content_main" />
26
27     <android.support.design.widget.FloatingActionButton
28         android:id="@+id/fab"
29         android:layout_width="wrap_content"
30         android:layout_height="wrap_content"
31         android:layout_gravity="bottom|end"
32         android:layout_margin="@dimen/fab_margin"
33         android:src="@android:drawable/ic_dialog_email" />
34
35 </android.support.design.widget.CoordinatorLayout>
```

액션바의 레이아웃

툴바 설정

플로팅 액션 버튼  
설정(▶ 삭제)

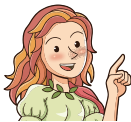
`activity_main.xml`에서 콘텐츠 영역의 화면 배치에 관한 `content_main.xml`에 화면에 출력할 **이미지뷰**를 배치한다.

소스 | `content_main.xml`

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
03     xmlns:app="http://schemas.android.com/apk/res-auto"
04     xmlns:tools="http://schemas.android.com/tools"
05     android:layout_width="match_parent"
06     android:layout_height="match_parent"
07     android:paddingBottom="@dimen/activity_vertical_margin"
08     android:paddingLeft="@dimen/activity_horizontal_margin"
09     android:paddingRight="@dimen/activity_horizontal_margin"
10     android:paddingTop="@dimen/activity_vertical_margin"
11     app:layout_behavior="@string/appbar_scrolling_view_behavior"
12     tools:context="com.example.contentsmgmt.MainActivity"
13     tools:showIn="@layout/activity_main">
14
15     <ImageView
16         android:layout_width="match_parent"
17         android:layout_height="match_parent"
18         android:src="@drawable/home" />
19 </RelativeLayout>
```

초기화면에 사용될 메인 이미지

## 5 액티비티 제어



플로팅 액션 버튼에 관한 부분을 삭제한다. 액션 바는 메뉴가 펼쳐지기는 하지만 메뉴 아이템을 클릭하면 아직 화면의 변화는 없는 상태이다.

소스 | `MainActivity.java`

```
01 package com.example.yschang.contentsmgmt;
02
03 import android.os.Bundle;
04 import android.support.design.widget.FloatingActionButton;
05 import android.support.design.widget.Snackbar;
06 import android.support.v7.app.AppCompatActivity;
07 import android.support.v7.widget.Toolbar;
08 import android.view.View;
09 import android.view.Menu;
10 import android.view.MenuItem;
11
12 public class MainActivity extends AppCompatActivity {
```

```

13
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_main);
18     Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
19     setSupportActionBar(toolbar);
20 }
21
22 @Override
23 public boolean onCreateOptionsMenu(Menu menu) {
24     getMenuInflater().inflate(R.menu.menu_main, menu);
25     return true;
26 }
27
28 @Override
29 public boolean onOptionsItemSelected(MenuItem item) {
30     int id = item.getItemId();
31     if(id == R.id.action_settings) {
32         return true;
33     }
34
35     return super.onOptionsItemSelected(item);
36 }
37
38 }

```

툴바 인식 (F)

액션바에 툴바 설정 (G)

메뉴 아이템 초기화 (D)

메뉴 아이템 선택 시 실행

선택된 메뉴 아이템의 id 인식

## 클래스와 속성/메소드

### 클래스

클래스/인터페이스	설명
Toolbar	18행. 어플리케이션 콘텐츠 내에 사용하기 위한 표준 툴바
Menu	23행. 메뉴 아이템을 관리하는 인터페이스
MenuInflater	XML 메뉴 파일을 메뉴 오브젝트로 만드는데 사용됨
MenuItem	29행. 이전에 만들어진 메뉴 아이템에 대한 직접적인 접근을 위한 인터페이스



## ● 메소드

클래스	메소드	설명
AppCompatActivity	void <b>setSupportActionBar</b> (Toolbar toolbar)	19행. 액티비티 창에 액션 바가 나타나도록 ToolBar를 설정함
Activity	boolean <b>onCreateOptionsMenu</b> (Menu menu)	23행. 액티비티의 표준 옵션 메뉴의 콘텐츠를 초기화함
	boolean <b>onOptionsItemSelected</b> (MenuItem item)	29행. 옵션 메뉴의 아이템이 선택될 때 호출됨
	MenuInflater <b>getMenuInflater()</b>	24행. menu XML 파일을 메뉴 객체화하는 MenuInflater를 반환함
MenuInflater	void <b>inflate</b> (int menuRes, Menu menu)	24행. XML 리소스로부터 메뉴 계층을 만들
MenuItem	abstract int <b>getItemId()</b>	30행. 메뉴 아이템의 id를 반환함

### STEP 3 ▶ 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭

#### 21.2.4 액션 바 메뉴와 '사진 등록' 액티비티 실행



액션 바가 어떻게 반응하는지 살펴보자.

액션 바의 메뉴 아이템인 '사진 등록'을 클릭하면 '사진 등록' 액티비티로 이동하고, '사진 등록' 액티비티에서는 다시 '홈'으로 돌아오는 메뉴를 작성한다.

## 프로젝트



### STEP 2 파일 편집: 홈 액티비티

#### ● 편집

#### ① 텍스트 자원

**strings.xml**에 사진 등록 액티비티 라벨로 사용할 텍스트와 메뉴 아이템으로 사용할 메뉴들의 텍스트 자원들을 구성한다.

#### 소스 | strings.xml

```

01 <resources>
02   <string name="app_name">갤러리</string>
03   <string name="act_reg_form">사진 등록</string>
04   <string name="action_settings1">사진 등록</string>
05   <string name="action_settings2">홈</string>
06 </resources>

```

액티비티 라벨 추가 (I)

액티비티 메뉴 아이템 (J, K)

## 2 메뉴 구성 자원

홈 화면에서 사용할 메뉴 구성을 위한 menu\_main.xml에서 **id**와 **title** 부분의 **action\_settings**를 **action\_settings1**로 수정한다.

소스 | menu\_main.xml

```
01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_settings1"
07         android:orderInCategory="100"
08         android:title="@string/action_settings1"
09         app:showAsAction="never" />
10 </menu>
```

메뉴 아이템 설정

## 3 액티비티 제어

MainActivity.java에서 액션 바의 메뉴 아이템 클릭 시, '사진등록' 액티비티(Content-RegForm 클래스)를 호출하는 기능을 추가한다.

소스 | MainActivity.java

```
01 package com.example.yschang.contentsmgmt;
02
03 import android.content.Intent;
04 import android.os.Bundle;
05 import android.support.design.widget.FloatingActionButton;
06 import android.support.design.widget.Snackbar;
07 import android.support.v7.app.AppCompatActivity;
08 import android.support.v7.widget.Toolbar;
09 import android.view.Menu;
10 import android.view.MenuItem;
11 import android.view.View;
12
13 public class MainActivity extends AppCompatActivity {
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19         Toolbar toolbar =(Toolbar) findViewById(R.id.toolbar);
```

```

20     setSupportActionBar(toolbar);
21 }
22
23 @Override
24 public boolean onCreateOptionsMenu(Menu menu) {
25     getMenuInflater().inflate(R.menu.menu_main, menu);
26     return true;
27 }
28
29 @Override
30 public boolean onOptionsItemSelected(MenuItem item) {
31     int id = item.getItemId();
32
33     if(id == R.id.action_settings1) {
34         Intent it = new Intent(this, ContentsReg.class);
35         startActivity(it);
36         finish();
37         return true;
38     }
39
40     return super.onOptionsItemSelected(item);
41 }
42 }

```

메뉴 아이템 선택 시 실행

선택된 메뉴 아이템의 id 인식

선택된 메뉴 아이템의 id가 action\_settings1인 경우

ContentsReg 클래스 호출

ContentsReg 클래스를 호출할 인텐트 생성

현재 액티비티 종료

## STEP 2 파일 편집: '사진 등록' 액티비티

### ● 편집

#### ① 메뉴 구성 자원

menu\_main.xml을 복사하여 menu\_reg\_form.xml을 만들고 action\_settings1을 action\_settings2로 수정한다.

소스 | menu\_reg\_form.xml → N

```

01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_settings2"
07         android:orderInCategory="100"
08         android:title="@string/action_settings2"
09         app:showAsAction="never" />
10 </menu>


```

O


K

## 2 화면 설계


activity\_main.xml을 복사하여 **activity\_reg\_form.xml**을 만들고, 콘텐츠 레이아웃 파일을 이용하여 화면을구성하도록 한다.

소스 | activity\_reg\_form.xml 

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
03     xmlns:app="http://schemas.android.com/apk/res-auto"
04     xmlns:tools="http://schemas.android.com/tools"
05     android:layout_width="match_parent"
06     android:layout_height="match_parent"
07     android:fitsSystemWindows="true"
08     tools:context="com.example.contentsmgmt.MainActivity">
09
10     <android.support.design.widget.AppBarLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:theme="@style/AppTheme.AppBarOverlay">
14
15         <android.support.v7.widget.Toolbar
16             android:id="@+id/toolbar"
17             android:layout_width="match_parent"
18             android:layout_height="?attr/actionBarSize"
19             android:background="?attr/colorPrimary"
20             app:popupTheme="@style/AppTheme.PopupOverlay" />
21
22     </android.support.design.widget.AppBarLayout>
23
24     <include layout="@layout/contents_reg_form" />
25
26 </android.support.design.widget.CoordinatorLayout>
```

content\_main.xml을 복사하여 **contents\_reg\_form.xml**을 만들고, 입력 컨트롤들을 배치하기 전에 ImageViw 대신 임시로 사용할 **TextView**를 배치해서 strings.xml에 정의된 **app\_name**을 출력하기로 한다.

소스 | content\_reg\_form.xml 

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
03     xmlns:app="http://schemas.android.com/apk/res-auto"
```

```

04   xmlns:tools="http://schemas.android.com/tools"
05   android:layout_width="match_parent"
06   android:layout_height="match_parent"
07   android:paddingBottom="@dimen/activity_vertical_margin"
08   android:paddingLeft="@dimen/activity_horizontal_margin"
09   android:paddingRight="@dimen/activity_horizontal_margin"
10   android:paddingTop="@dimen/activity_vertical_margin"
11   app:layout_behavior="@string/appbar_scrolling_view_behavior"
12   tools:context="com.example.contentsmgt.MainActivity"
13   tools:showIn="@layout/activity_main">
14
15   <TextView
16       android:layout_width="match_parent"
17       android:layout_height="match_parent"
18       android:text="@string/app_name" />
19 </RelativeLayout>

```

액티비티 호출 결과를 확인하기 위해 테스트용으로  
설정된 app\_name 속성 값 설정

B

### ③ 액티비티 제어

MainActivity.java를 복사하여 **ContentsRegForm.java**를 만들고 메뉴를 클릭하면 ‘**홈**’(MainActivity 클래스)으로 가도록 설정한다.

소스 | ContentsRegForm.java

S

```

01 package com.example.yschang.contentsmgt;
02
03 import android.content.Intent;
04 import android.os.Bundle;
05 import android.support.v7.app.AppCompatActivity;
06 import android.support.v7.widget.Toolbar;
07 import android.view.Menu;
08 import android.view.MenuItem;
09
10 public class ContentsRegForm extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_reg_form);
16         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
17         setSupportActionBar(toolbar);
18     }
19
20     @Override
21     public boolean onCreateOptionsMenu(Menu menu) {

```

R

Q

```

21     getMenuInflater().inflate(R.menu.menu_reg_form, menu);
22     return true;
23 }
24
25 @Override
26 public boolean onOptionsItemSelected(MenuItem item) {
27     int id = item.getItemId();
28
29     if(id == R.id.action_settings2) {
30         Intent it = new Intent(this, MainActivity.class);
31         startActivity(it);
32         finish();
33         return true;
34     }
35
36     return super.onOptionsItemSelected(item);
37 }
38 }

```

메뉴 아이템 선택 시 실행

선택된 메뉴 아이템의 id 인식

선택된 메뉴 아이템의 id가 action\_settings인 경우

MainActivity 클래스 호출

초기화면으로 돌아가는 MainActivity 클래스를 호출할 인텐트 생성

현재 액티비티 종료

#### 4 환경 설정

**AndroidManifest.xml**에 새로 만든 액티비티 클래스인 **ContentsRegForm**을 등록한다.

소스 | AndroidManifest.xml

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.example.yschang.contentsmgmt">
04
05     <application
06         android:allowBackup="true"
07         android:icon="@mipmap/ic_launcher"
08         android:label="@string/app_name"
09         android:supportRtl="true"
10         android:theme="@style/AppTheme">
11         <activity
12             android:name=".MainActivity"
13             android:label="@string/app_name"
14             android:theme="@style/AppTheme.NoActionBar">
15             <intent-filter>
16                 <action android:name="android.intent.action.MAIN" />
17
18                 <category android:name="android.intent.category.LAUNCHER" />
19             </intent-filter>
20         </activity>

```

```

21 <activity
22     android:name=".ContentsRegForm"
23     android:label="@string/act_reg_form"
24     android:theme="@style/AppTheme.NoActionBar">
25 </activity>
26 </application>
27
28 </manifest>

```

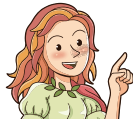
ContentsRegForm 클래스의 액티비티 등록

### STEP 3 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭

#### 21.2.5 '사진 등록' 액티비티의 화면 구성



입력 컨트롤들로 카메라로 찍은 사진을 등록하는 화면을 구성해 보자.

입력사항은 사진 제목, 사진 방향, 배경 구성요소, 그리고 카메라 사진이다. 우선 등록 버튼을 클릭하면 토스트로 “클릭했습니다!”라는 문자열을 출력한다.



## 프로젝트

### 초기 화면

### 등록할 사진 조회를 위한 클릭 설정

## STEP 2 파일 편집: '사진 등록' 액티비티

### ● 편집

#### ① 텍스트 자원

strings.xml에 사진 등록 액티비티의 화면구성 아이템으로 사용될 속성값들을 추가한다.

### 소스 | strings.xml

```
01 <resources>
02     <string name="app_name">갤러리</string>
03     <string name="act_reg_form">사진 등록</string>
04
05     <string name="action_settings1">사진 등록</string>
06     <string name="action_settings2">홈</string>
07
08     <string name="title">제목</string>
09     <string name="orientation">방향</string>
10     <string name="background">배경</string>
11     <string name="photo">사진</string>
12 </resources>
```

화면구성 아이템으로  
사용될 속성 값 추가

## 2 화면 설계

`contents_reg_form.xml`에는 입력제어 구성요소들로 ‘사진 등록’을 위한 화면을 구성한다.

소스 | `content_reg_form.xml`

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
03     android:layout_width="match_parent"
04     android:layout_height="match_parent"
05     android:orientation="vertical"
06     android:paddingBottom="@dimen/activity_vertical_margin"
07     android:paddingLeft="@dimen/activity_horizontal_margin"
08     android:paddingRight="@dimen/activity_horizontal_margin"
09     android:paddingTop="70dp" >
```

```
11 <LinearLayout
12     android:orientation="horizontal"
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content" >
15     <TextView
16         android:layout_width="60sp"
17         android:layout_height="wrap_content"
18         android:text="@string/title" />
19     <EditText
20         android:id="@+id/title"
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content" />
23 </LinearLayout>
```

사진 제목

“제목” 출력을 위한 텍스트뷰

사진 제목 입력을 위한 편집박스

```
25 <LinearLayout
26     android:orientation="horizontal"
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content" >
29     <TextView
30         android:layout_width="60sp"
31         android:layout_height="wrap_content"
32         android:text="@string/orientation" />
33     <RadioGroup
34         android:id="@+id/orientation"
35         android:layout_width="match_parent"
36         android:layout_height="wrap_content"
37         android:orientation="vertical" >
38         <RadioButton
```

사진 방향

“방향” 출력을 위한 텍스트뷰

사진 방향 선택을 위한 라디오 그룹

“가로” 방향 선택을 위한 라디오 버튼

“세로” 방향 선택을  
위한 라디오 버튼

```
39         android:id="@+id/landscape"
40         android:layout_width="wrap_content"
41         android:layout_height="wrap_content"
42         android:text="가로" />
43     <RadioButton
44         android:id="@+id/portrait"
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:text="세로" />
48 </RadioGroup>
49 </LinearLayout>
```

AA

AB

“배경” 출력력을  
위한 텍스트뷰

```
51 <LinearLayout
52     android:orientation="horizontal"
53     android:layout_width="match_parent"
54     android:layout_height="wrap_content" >
55     <TextView
56         android:layout_width="60sp"
57         android:layout_height="wrap_content"
58         android:text="@string/background" />
```

사진 배경

사진 배경 구성

```
59 <LinearLayout
60     android:orientation="vertical"
61     android:layout_width="match_parent"
62     android:layout_height="wrap_content" >
```

V

“인물” 선택을  
위한 체크 박스

```
63 <CheckBox
64     android:id="@+id/background1"
65     android:layout_width="wrap_content"
66     android:layout_height="wrap_content"
67     android:text="인물" />
```

AC

“자연” 선택을  
위한 체크 박스

```
68 <CheckBox
69     android:id="@+id/background2"
70     android:layout_width="wrap_content"
71     android:layout_height="wrap_content"
72     android:text="자연" />
```

AD

“건물” 선택을  
위한 체크 박스

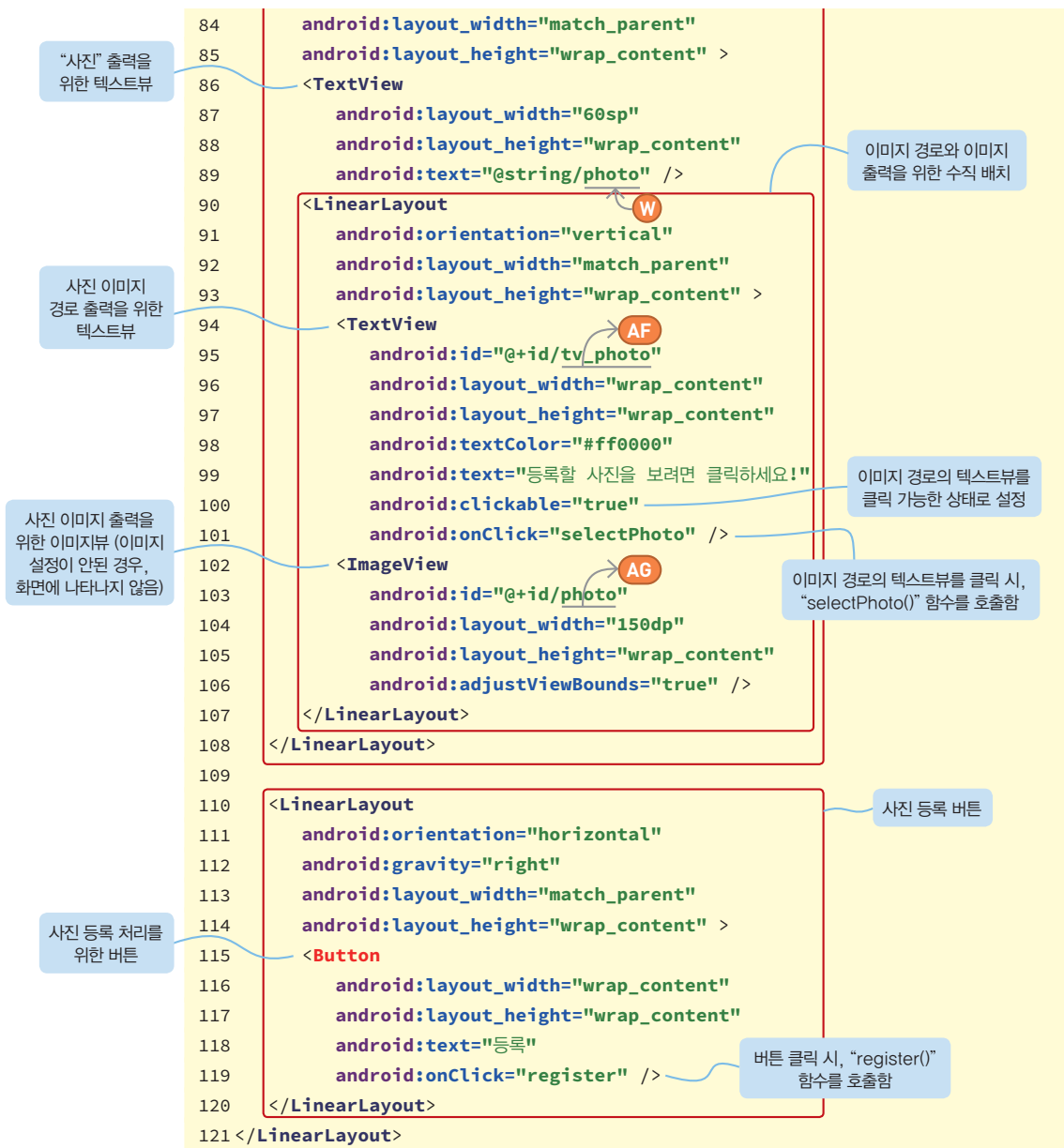
```
73 <CheckBox
74     android:id="@+id/background3"
75     android:layout_width="wrap_content"
76     android:layout_height="wrap_content"
77     android:text="건물" />
```

AE

```
79 </LinearLayout>
80 </LinearLayout>
```

사진

```
81
82 <LinearLayout
83     android:orientation="horizontal"
```



## 클래스와 속성/메소드

### ● 클래스

클래스/인터페이스	설명
Button	115행. 푸쉬 버튼 위젯을 나타냄
CheckBox	64,69,74행. '체크' 또는 '체크 없음'의 두 상태 중 하나로 될 수 있는 버튼임
RadioButton	55,60행. '체크' 또는 '체크 없음'의 두 상태 중 하나를 표시하는 버튼임
RadioGroup	50행. 라디오버튼들의 집합에 대하여 하나의 라디오버튼만 선택이 가능하도록 만드는데 사용됨

### ③ 액티비티 제어

**ContentsRegForm.java**는 사진이 보여질 부분을 클릭하면 실행할 `selectPhoto()` 콜백 메소드와 “등록” 버튼을 클릭하면 실행할 `register()` 메소드를 만들고, **Toast**로 “클릭했습니다!”라는 문자를 출력하도록 설정한다.

소스 | ContentsRegForm.java → AH

```
01 package com.example.yschang.contentsmgt;
02
03 import android.content.Context;
04 import android.content.Intent;
05 import android.database.Cursor;
06 import android.graphics.Bitmap;
07 import android.graphics.BitmapFactory;
08 import android.net.Uri;
09 import android.os.Bundle;
10 import android.provider.MediaStore;
11 import android.support.v7.app.AppCompatActivity;
12 import android.support.v7.widget.Toolbar;
13 import android.view.Menu;
14 import android.view.MenuItem;
15 import android.view.View;
16 import android.widget.ImageView;
17 import android.widget.TextView;
18 import android.widget.Toast;
19
20 public class ContentsRegForm extends AppCompatActivity {
21
22     @Override
```

```

23  protected void onCreate(Bundle savedInstanceState) {
24      super.onCreate(savedInstanceState);
25      setContentView(R.layout.activity_reg_form);
26      Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
27      setSupportActionBar(toolbar);
28  }
29
30  public void selectPhoto(View v) {
31      Toast.makeText(this, "클릭했습니다!", Toast.LENGTH_SHORT).show();
32  }
33
34  public void register(View v) {
35      Toast.makeText(this, "클릭했습니다!", Toast.LENGTH_SHORT).show();
36  }
37
38  @Override
39  public boolean onCreateOptionsMenu(Menu menu) {
40      getMenuInflater().inflate(R.menu.menu_reg_form, menu);
41      return true;
42  }
43
44  @Override
45  public boolean onOptionsItemSelected(MenuItem item) {
46      int id = item.getItemId();
47
48      if(id == R.id.action_settings2) {
49          Intent it = new Intent(this, MainActivity.class);
50          startActivity(it);
51          finish();
52          return true;
53      }
54
55      return super.onOptionsItemSelected(item);
56  }
57  }

```

P 이미지 경로의 텍스트뷰를 클릭 시 실행되는 콜백 메소드

Q

이미지 경로의 텍스트뷰를 클릭 시 실행되는 콜백 메소드

버튼 클릭 시 실행되는 콜백 메소드

메뉴 아이템 설정

메뉴 아이템 선택 시 실행

선택된 메뉴 아이템의 id 인식

선택된 메뉴 아이템의 id가 action\_settings인 경우, MainActivity 클래스를 호출하고 현재 액티비티는 종료

### STEP 3 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭

## 21.2.6 사진 등록을 위한 카메라 사진 선택

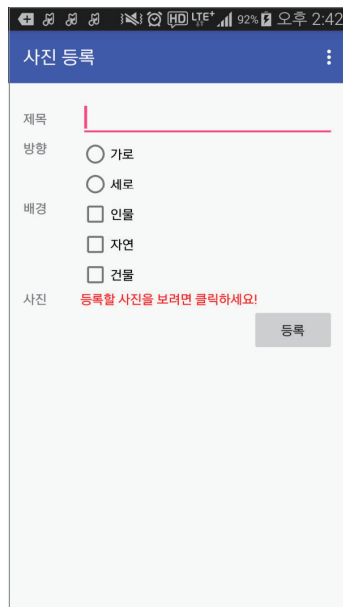


이제 사진을 선택하는 `selectPhoto()` 메소드를 구현하도록 하자.

“등록할 사진을 보려면 클릭하세요!”라는 문자열을 클릭하면 카메라로 찍은 사진들이 있는 곳으로 이동하고, 사진을 선택하면 사진 경로를 파악하여 변수에 저장하고 사진을 이미지뷰에 출력한다.

### 프로젝트

#### 초기 화면



#### 사용할 어플리케이션 선택

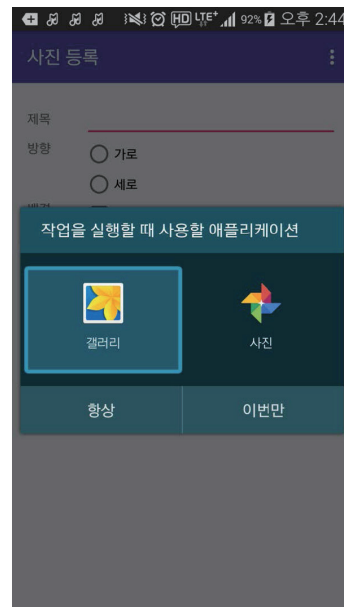


사진 선택



선택한 사진 출력



## STEP 2 파일 편집: '사진 등록' 액티비티

### ● 편집: 액티비티 제어

**ContentsRegForm.java**의 selectPhoto() 메소드에 카메라 사진을 선택하고 경로를 저장하며 선택한 사진을 출력한다.

소스 | ContentsRegForm.java → AI

```
01 package com.example.yschang.contentsmgmt;
02
03 import android.content.Context;
04 import android.content.Intent;
05 import android.database.Cursor;
06 import android.graphics.Bitmap;
07 import android.graphics.BitmapFactory;
08 import android.net.Uri;
09 import android.os.Bundle;
10 import android.provider.MediaStore;
11 import android.support.v7.app.AppCompatActivity;
12 import android.support.v7.widget.Toolbar;
13 import android.view.Menu;
```



```

14 import android.view.MenuItem;
15 import android.view.View;
16 import android.widget.ImageView;
17 import android.widget.TextView;
18
19 public class ContentsRegForm extends AppCompatActivity {
20
21     public final static int PICK_PHOTO_CODE = 1046;
22     ImageView iv_photo;
23     TextView tv_photo;
24     String path;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_reg_form);
30         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
31         setSupportActionBar(toolbar);
32
33         iv_photo = (ImageView) findViewById(R.id.photo);
34         tv_photo = (TextView) findViewById(R.id.tv_photo);
35     }
36
37     public void selectPhoto(View v) {
38         Intent intent = new Intent(Intent.ACTION_PICK,
39             MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
40         startActivityForResult(intent, PICK_PHOTO_CODE);
41     }
42
43     @Override
44     public void onActivityResult(int requestCode, int resultCode,
45                                     Intent data) {
46         super.onActivityResult(requestCode, resultCode, data);
47
48         Uri photoUri = data.getData();
49         path = getRealPathFromURI(this, photoUri);
50
51         try {
52             Bitmap bm = MediaStore.Images.Media.getBitmap(
53                                     this.getContentResolver(), photoUri);
54
55             int WIDTH = 150;
56             int width = WIDTH;
57             float ratio = (float) bm.getHeight() / bm.getWidth();

```

사진을 출력할 이미지뷰 인식  
 사진경로 출력을 위한 텍스트뷰 인식  
 텍스트뷰 클릭 시 실행되는 콜백 메소드  
 시스템의 사진목록 액티비티에서 선택된 사진 처리를 위한 콜백 메소드  
 사진 이미지의 실제 경로 추출  
 이미지의 너비를 150 픽셀로 설정

```

57         int height =(int)(WIDTH * ratio);
58
59         Bitmap thumb = Bitmap.createScaledBitmap(bm, width, height, false);
60         iv_photo.setImageBitmap(thumb);
61     } catch (FileNotFoundException e) {
62         Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
63     } catch (IOException e) {
64         Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
65     }
66 }
67
68 public String getRealPathFromURI(Context context, Uri contentUri) {
69     Cursor cursor = null;
70     try {
71         String[] proj = { MediaStore.Images.Media.DATA };
72         cursor = context.getContentResolver().query(contentUri, proj,
73                                                     null, null, null);
74         int column_index = cursor.getColumnIndexOrThrow(
75                                                     MediaStore.Images.Media.DATA);
76         cursor.moveToFirst();
77         return cursor.getString(column_index);
78     } finally {
79         if(cursor != null) {
80             cursor.close();
81         }
82     }
83 }
84
85 public void register(View v) {
86     Toast.makeText(this, "클릭했습니다!", Toast.LENGTH_SHORT).show();
87 }
88
89 @Override
90 public boolean onCreateOptionsMenu(Menu menu) {
91     getMenuInflater().inflate(R.menu.menu_reg_form, menu);
92     return true;
93 }
94
95 @Override
96 public boolean onOptionsItemSelected(MenuItem item) {
97     int id = item.getItemId();
98
99     if(id == R.id.action_settings2) {
100         Intent it = new Intent(this, MainActivity.class);
101         startActivity(it);

```

이미지의 높이를 너비의  
변화율로 조정

조정된 크기로 썸네일  
이미지 생성

썸네일 이미지 출력

이미지의 실제 경로 반환

AJ

사진정보 저장을 위한 버튼  
클릭 시 실행되는 콜백 메소드

```

100         finish();
101         return true;
102     }
103
104     return super.onOptionsItemSelected(item);
105 }
106 }

```

## 클래스와 속성/메소드

### ● 클래스

클래스/인터페이스	설명
<a href="#">Bitmap</a>	<b>52, 59행.</b> 비트맵(이미지 저장 파일 포맷의 한 형태) 관련 클래스
<a href="#">ContentResolver</a>	애플리케이션이 데이터를 액세스하도록 지원
<a href="#">MediaStore.Images.Media</a>	<b>39, 52, 71, 73행.</b> 외부저장소의 미디어 관리
<a href="#">FileNotFoundException</a>	<b>61행.</b> 지정한 파일이 발견되지 않는 예외사항 처리
<a href="#">Cursor</a>	<b>69행.</b> 데이터베이스 질의에 의한 데이터 집합에 대한 액세스를 제공하는 인터페이스

### ● 상수

클래스	상수	설명
Intent	String <a href="#">ACTION_PICK</a>	<b>38행.</b> 데이터로부터 선택된 아이템을 고름
MediaStore.Images. Media	String DATA	<b>71, 73행.</b> 이미지 파일의 경로
	static final Uri <a href="#">EXTERNAL_CONTENT_URI</a>	<b>39행.</b> 외부저장소에 있는 콘텐츠 Uri

### ● 메소드

클래스	메소드	설명
Bitmap	static Bitmap <a href="#">createScaledBitmap</a> (Bitmap src, int dstWidth, int dstHeight, boolean filter)	<b>59행.</b> 현재 Bitmap 이미지를 조정된 크기의 이미지로 변경
	final int <a href="#">getHeight</a> ()	<b>56행.</b> Bitmap 이미지의 높이를 반환
	final int <a href="#">getWidth</a> ()	<b>56행.</b> Bitmap 이미지의 너비를 반환
Context	abstract ContentResolver <a href="#">getContentResolver</a> ()	<b>52, 72행.</b> 앱의 패키지에 대한 ContentResolver 객체 반환

ContentResolver	final Cursor <b>query</b> (Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)	<b>72행.</b> 주어진 URI에 대한 질의로 결과집합에 대한 커서 반환
Cursor	abstract int <b>getColumnIndexOrThrow</b> (String columnName)	<b>73행.</b> 주어진 항목명에 해당하는 색인번호 반환
	abstract Boolean <b>moveToFirst</b> ()	<b>74행.</b> 커서를 첫 번째 행으로 이동
	abstract String <b>getString</b> (int columnIndex)	<b>75행.</b> 항목의 값을 문자로 반환
	abstract void <b>close</b> ()	<b>78행.</b> 커서를 닫음
FragmentActivity	void <b>onActivityResult</b> (int requestCode, int resultCode, Intent data)	<b>44행.</b> StartActivityResult()에 의해 호출된 액티비티로부터 결과를 받아 처리함(재정의)
	void <b>startActivityForResult</b> (Intent intent, int requestCode)	<b>40행.</b> requestCode(시스템의 사진 목록)에 해당하는 액티비티를 호출. 실행결과를 받고자 할 때 사용
ImageView	void <b>setImageBitmap</b> (Bitmap bm)	<b>60행.</b> 이미지뷰의 내용으로 Bitmap 파일을 설정
Intent	Uri <b>getData</b> ()	<b>48행.</b> 인텐트가 가지고 있는 데이터를 반환함
MediaStore.Images.Media	final static Bitmap <b>getBitmap</b> (ContentResolver cr, Uri url)	<b>52행.</b> Uri에 해당하는 이미지 반환

### STEP 3 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭

## 21.2.7 입력 컨트롤들의 입력 값 추출과 알림창 띄우기



입력 컨트롤에 입력한 값과 선택한 사진 경로를 추출하여 **알림창**에 띄우는 과정을 살펴보자.

### 프로젝트

초기 화면

등록할 사진 조회를 위한 클릭 설정

### STEP 2 파일 편집: '사진 등록' 액티비티

#### ● 편집: 액티비티 제어

입력한 데이터를 추출하여 알림창으로 띄우는 과정을 **ContentsRegForm.java**의 `register()` 메소드에 추가한다.

소스 | ContentsRegForm.java → AK

```
01 package com.example.yschang.contentsmgt;  
02
```

```

03 import android.app.AlertDialog;
04 import android.content.Context;
05 import android.content.Intent;
06 import android.database.Cursor;
07 import android.graphics.Bitmap;
08 import android.graphics.BitmapFactory;
09 import android.net.Uri;
10 import android.os.Bundle;
11 import android.provider.MediaStore;
12 import android.support.v7.app.AppCompatActivity;
13 import android.support.v7.widget.Toolbar;
14 import android.view.Menu;
15 import android.view.MenuItem;
16 import android.view.View;
17 import android.widget.CheckBox;
18 import android.widget.EditText;
19 import android.widget.ImageView;
20 import android.widget.RadioButton;
21 import android.widget.RadioGroup;
22 import android.widget.TextView;
23
24 public class ContentsRegForm extends AppCompatActivity {
25
26     public final static int PICK_PHOTO_CODE = 1046;
27     ImageView iv_photo;
28     TextView tv_photo;
29     String path;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_reg_form);
35         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
36         setSupportActionBar(toolbar);
37
38         iv_photo = (ImageView) findViewById(R.id.photo);
39         tv_photo = (TextView) findViewById(R.id.tv_photo);
40     }
41
42     public void selectPhoto(View v) {
43         Intent intent = new Intent(Intent.ACTION_PICK,
44             MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
45         startActivityForResult(intent, PICK_PHOTO_CODE);
46     }
47

```

```

48     @Override
49     public void onActivityResult(int requestCode, int resultCode,
                                   Intent data) {
50
51         super.onActivityResult(requestCode, resultCode, data);
52
53         Uri photoUri = data.getData();
54         path = getRealPathFromURI(this, photoUri);
55
56         try {
57             Bitmap bm = MediaStore.Images.Media.getBitmap(
58                                     this.getContentResolver(), photoUri);
59
60             int WIDTH = 150;
61             int width = WIDTH;
62             float ratio = (float) bm.getHeight() / bm.getWidth();
63             int height = (int)(WIDTH * ratio);
64
65             Bitmap thumb = Bitmap.createScaledBitmap(bm, width, height, false);
66             iv_photo.setImageBitmap(thumb);
67         } catch (FileNotFoundException e) {
68             Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
69         } catch (IOException e) {
70             Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
71         }
72
73     public String getRealPathFromURI(Context context, Uri contentUri) {
74         Cursor cursor = null;
75         try {
76             String[] proj = { MediaStore.Images.Media.DATA };
77             cursor = context.getContentResolver().query(
78                                     contentUri, proj, null, null, null);
79             int column_index = cursor.getColumnIndexOrThrow(
80                                     MediaStore.Images.Media.DATA);
81             cursor.moveToFirst();
82             return cursor.getString(column_index);
83         } finally {
84             if(cursor != null) {
85                 cursor.close();
86             }
87         }
88     }

```

사진정보 저장을 위한 버튼 클릭 시 실행되는 콜백 메소드

사진 제목 입력을 위한 EditText 뷰 인식 및 입력한 제목 추출

```
88 public void register(View v) {
89     EditText et_title =(EditText)findViewById(R.id.title);
90     String str_title = et_title.getText().toString();
91
92     RadioGroup rg_orientation =(RadioGroup)findViewById(
93         R.id.orientation);
94     RadioButton rb_orientation;
95     String str_orientation = "";
96     if(rg_orientation.getCheckedRadioButtonId() == R.id.landscape) {
97         rb_orientation =(RadioButton)findViewById(R.id.landscape);
98         str_orientation = rb_orientation.getText().toString();
99     }
100     if(rg_orientation.getCheckedRadioButtonId() == R.id.portrait) {
101         rb_orientation =(RadioButton)findViewById(R.id.portrait);
102         str_orientation = rb_orientation.getText().toString();
103     }
104     CheckBox chk_background;
105     String str_background1 = "";
106     String str_background2 = "";
107     String str_background3 = "";
108
109     chk_background =(CheckBox)findViewById(R.id.background1);
110     if(chk_background.isChecked()) {
111         str_background1 =(String)chk_background.getText();
112     }
113     chk_background =(CheckBox)findViewById(R.id.background2);
114     if(chk_background.isChecked()) {
115         str_background2 =(String)chk_background.getText();
116     }
117     chk_background =(CheckBox)findViewById(R.id.background3);
118     if(chk_background.isChecked()) {
119         str_background3 =(String)chk_background.getText();
120     }
121
122     AlertDialog.Builder alert = new AlertDialog.Builder(this);
123     alert.setTitle("알림창");
124     alert.setMessage("제목: " + str_title +
125         "\n방향: " + str_orientation +
126         "\n배경: " + str_background1 + " " + str_background2 +
127         " " + str_background3 +
128         "\n사진경로: " + path);
129     alert.setIcon(R.mipmap.ic_launcher);
130     alert.setPositiveButton("확인", null);
131     alert.show();
132 }
```

입력한 사진 방향의 값을 추출하기 위한 라디오 그룹

입력한 사진 방향이 '가로(landscape)'이면 '가로' 문자로 표시되는 라디오 버튼 값 추출

id가 background1인 체크박스('인물')가 체크되었으면 체크박스의 값을 추출

입력한 사진 배경 값 추출

알림창에 입력한 사진정보 출력

알림창에 제목 설정

알림창 객체 생성

알림창에 공지할 메시지 설정

알림창에 표시할 아이콘 설정

"확인" 버튼 하나만 출력

알림창을 보이게 함



```

132
133     @Override
134     public boolean onCreateOptionsMenu(Menu menu) {
135         getMenuInflater().inflate(R.menu.menu_reg_form, menu);
136         return true;
137     }
138
139     @Override
140     public boolean onOptionsItemSelected(MenuItem item) {
141         int id = item.getItemId();
142
143         if(id == R.id.action_settings2) {
144             Intent it = new Intent(this, MainActivity.class);
145             startActivity(it);
146             finish();
147             return true;
148         }
149
150         return super.onOptionsItemSelected(item);
151     }
152 }

```

## 클래스와 속성/메소드

### ● 클래스

클래스/인터페이스	설명
<a href="#">AlertDialog</a>	대화상자로 문자열을 출력함
<a href="#">AlertDialog.Builder</a>	122행, AlertDialog의 내부 클래스
<a href="#">CompoundButton</a>	버튼이 체크되었는지 아닌지, 두 상태를 가진 버튼

## ● 메소드

클래스	메소드	설명
EditText	Editable <code>getText()</code>	90,97,101,111,115,119행. 텍스트뷰가 출력하고 있는 텍스트를 반환
RadioGroup	int <code>getCheckedRadioButtonId()</code>	95,99행. 선택된 라디오 버튼의 ID를 반환함
AlertDialog.Builder	AlertDialog.Builder <code>setTitle(CharSequence title)</code>	123행. 대화상자에 표시될 title을 설정함
	AlertDialog.Builder <code>setMessage(CharSequence message)</code>	124행. 출력되는 message를 설정함
	AlertDialog.Builder <code>setIcon(int iconId)</code>	128행. 알림창 제목 내에 아이콘으로 표시될 이미지로, drawable 폴더의 이미지 리소스 id를 설정함
	AlertDialog.Builder <code>setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)</code>	129행. 알림창 버튼에 표시될 text와 클릭 시 호출 될 listener를 설정함
	AlertDialog <code>show()</code>	130행. 알림창을 보여줌
CompoundButton	boolean <code>isChecked()</code>	110,114,118행. 선택되면 true, 그렇지 않으면 false가 반환됨
TextView	CharSequence <code>getText()</code>	90,97,101,111,115,119행. 뷰의 텍스트 추출

## STEP 3 > 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭

## 21.2.8 입력 컨트롤들의 입력 값 추출과 정보 전달



입력한 사진 정보를 전달하고 전달받은 정보를 출력하는 방법을 살펴보자.

### 프로젝트

액티비티2 (등록할 사진정보)

제목: 풍경1

방향: ☒ 가로 ☐ 세로

배경: ☐ 인물 ☒ 자연 ☐ 건물

사진: 등록할 사진을 보려면 클릭하세요!

등록

액티비티3 (등록한 사진정보)

제목: 풍경1

방향: 가로

배경: 자연

사진:

홈

사진 등록

### STEP 2 파일 편집: '사진 등록' 액티비티

#### ● 편집: 액티비티 제어

ContentRegForm.java에서 알림창 대신 입력 값을 추출해서 '사진 정보' 액티비티로 전달하도록 수정한다.

소스 | ContentsRegForm.java → AK

```
01 package com.example.yschang.contentsmgmt;
02
```

```

03 import android.content.Context;
04 import android.content.Intent;
05 import android.database.Cursor;
06 import android.graphics.Bitmap;
07 import android.graphics.BitmapFactory;
08 import android.net.Uri;
09 import android.os.Bundle;
10 import android.provider.MediaStore;
11 import android.support.v7.app.AppCompatActivity;
12 import android.support.v7.widget.Toolbar;
13 import android.view.Menu;
14 import android.view.MenuItem;
15 import android.view.View;
16 import android.widget.CheckBox;
17 import android.widget.EditText;
18 import android.widget.ImageView;
19 import android.widget.RadioButton;
20 import android.widget.RadioGroup;
21 import android.widget.TextView;
22
23 public class ContentsRegForm extends AppCompatActivity {
24
25
26     public final static int PICK_PHOTO_CODE = 1046;
27     ImageView iv_photo;
28     TextView tv_photo;
29     String path;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_reg_form);
35         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
36         setSupportActionBar(toolbar);
37
38         iv_photo = (ImageView) findViewById(R.id.photo);
39         tv_photo = (TextView) findViewById(R.id.tv_photo);
40     }
41
42     public void selectPhoto(View v) {
43         Intent intent = new Intent(Intent.ACTION_PICK,
44             MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
45         startActivityForResult(intent, PICK_PHOTO_CODE);
46     }
47

```

```

48     @Override
49     public void onActivityResult(int requestCode, int resultCode,
                                   Intent data) {
50
51         super.onActivityResult(requestCode, resultCode, data);
52
53         Uri photoUri = data.getData();
54         path = getRealPathFromURI(this, photoUri);
55
56         try {
57             Bitmap bm = MediaStore.Images.Media.getBitmap(
58                                     this.getContentResolver(), photoUri);
59
60             int WIDTH = 150;
61             int width = WIDTH;
62             float ratio =(float) bm.getHeight() / bm.getWidth();
63             int height =(int)(WIDTH * ratio);
64
65             Bitmap thumb = Bitmap.createScaledBitmap(
66                                     bm, width, height, false);
67             iv_photo.setImageBitmap(thumb);
68         } catch(FileNotFoundException e) {
69             Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
70         } catch(IOException e) {
71             Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
72         }
73     }
74
75     public String getRealPathFromURI(Context context, Uri contentUri) {
76         Cursor cursor = null;
77         try {
78             String[] proj = { MediaStore.Images.Media.DATA };
79             cursor = context.getContentResolver().query(contentUri,  proj,
80                                                         null, null, null);
81             int column_index = cursor.getColumnIndexOrThrow(
82                                     MediaStore.Images.Media.DATA);
83             cursor.moveToFirst();
84             return cursor.getString(column_index);
85         } finally {
86             if(cursor != null) {
87                 cursor.close();
88             }
89         }
90     }

```

```

88 public void register(View v) {
89     EditText et_title =(EditText)findViewById(R.id.title);
90     String str_title = et_title.getText().toString();
91
92     RadioGroup rg_orientation =(RadioGroup)findViewById(
93                                     R.id.orientation);
94     RadioButton rb_orientation;
95     String str_orientation = "";
96     if(rg_orientation.getCheckedRadioButtonId() == R.id.landscape) {
97         rb_orientation =(RadioButton)findViewById(R.id.landscape);
98         str_orientation = rb_orientation.getText().toString();
99     }
100     if(rg_orientation.getCheckedRadioButtonId() == R.id.portrait) {
101         rb_orientation =(RadioButton)findViewById(R.id.portrait);
102         str_orientation = rb_orientation.getText().toString();
103     }
104
105     CheckBox chk_background;
106     String str_background1 = "";
107     String str_background2 = "";
108     String str_background3 = "";
109
110     chk_background =(CheckBox)findViewById(R.id.background1);
111     if(chk_background.isChecked()) {
112         str_background1 =(String)chk_background.getText();
113     }
114     chk_background =(CheckBox)findViewById(R.id.background2);
115     if(chk_background.isChecked()) {
116         str_background2 =(String)chk_background.getText();
117     }
118     chk_background =(CheckBox)findViewById(R.id.background3);
119     if(chk_background.isChecked()) {
120         str_background3 =(String)chk_background.getText();
121     }
122
123     Intent it = new Intent(this, ContentsDetail.class);
124     it.putExtra("it_title", str_title);
125     it.putExtra("it_orientation", str_orientation);
126     it.putExtra("it_background1", str_background1);
127     it.putExtra("it_background2", str_background2);
128     it.putExtra("it_background3", str_background3);
129     it.putExtra("it_path", path);
130     startActivity(it);
131     finish();
132 }

```

ContentsDetail 클래스를 호출하는 인텐트를 생성하고, 인텐트 변수들에 해당 값들을 저장 후에 ContentsDetail 액티비티를 호출. 현재 액티비티는 종료

```

132
133     @Override
134     public boolean onCreateOptionsMenu(Menu menu) {
135         getMenuInflater().inflate(R.menu.menu_reg_form, menu);
136         return true;
137     }
138
139     @Override
140     public boolean onOptionsItemSelected(MenuItem item) {
141         int id = item.getItemId();
142
143         if(id == R.id.action_settings2) {
144             Intent it = new Intent(this, MainActivity.class);
145             startActivity(it);
146             finish();
147             return true;
148         }
149
150         return super.onOptionsItemSelected(item);
151     }
152 }

```

## STEP 2 파일 편집: '사진 정보' 액티비티

### ● 편집

#### ① 텍스트 자원

strings.xml에 사진 정보 액티비티의 액티비티 라벨을 추가한다.

#### 소스 | strings.xml

```

01 <resources>
02     <string name="app_name">갤러리</string>
03     <string name="act_reg_form">사진 등록</string>
04     <string name="act_detail">사진 정보</string>
05
06     <string name="action_settings1">사진 등록</string>
07     <string name="action_settings2">홈</string>
08
09     <string name="title">제목</string>
10     <string name="orientation">방향</string>
11     <string name="background">배경</string>
12     <string name="photo">사진</string>
13 </resources>

```

사진 정보 액티비티 라벨 추가

## ② 메뉴 구성

`menu_detail.xml`을 만들고 ‘사진 정보’ 액티비티에서 사용할 ‘사진 등록’과 ‘홈’ 메뉴를 구성한다.

소스 | menu\_detail.xml → AM

```
01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_settings2"
07         android:orderInCategory="100"
08         android:title="@string/action_settings2"
09         app:showAsAction="never" />
10     <item
11         android:id="@+id/action_settings1"
12         android:orderInCategory="100"
13         android:title="@string/action_settings1"
14         app:showAsAction="never" />
15 </menu>
```

AN → K: '홈' 메뉴 아이템

AO → J: '사진 등록' 메뉴 아이템

## ③ 화면 설계

`activity_detail.xml`을 만들고 콘텐츠 화면으로 `contents_detail.xml`을 설정한다.

소스 | activity\_detail.xml → AP

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <android.support.design.widget.CoordinatorLayout
03     xmlns:android="http://schemas.android.com/apk/res/android"
04     xmlns:app="http://schemas.android.com/apk/res-auto"
05     xmlns:tools="http://schemas.android.com/tools"
06     android:layout_width="match_parent"
07     android:layout_height="match_parent"
08     android:fitsSystemWindows="true"
09     tools:context="com.example.contentsmgmt.MainActivity">
10     <android.support.design.widget.AppBarLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:theme="@style/AppTheme.AppBarOverlay">
14     </AppBarLayout>
15 </CoordinatorLayout>
```



```

15     <android.support.v7.widget.Toolbar
16         android:id="@+id/toolbar"
17         android:layout_width="match_parent"
18         android:layout_height="?attr/actionBarSize"
19         android:background="?attr/colorPrimary"
20         app:popupTheme="@style/AppTheme.PopupOverlay" />
21
22 </android.support.design.widget.AppBarLayout>
23
24 <include layout="@layout/contents_detail" />
25
26 </android.support.design.widget.CoordinatorLayout>

```

화면에 출력할 contents\_detail.xml에는 사진 제목, 방향, 배경, 사진 이미지 등을 배치한다.

소스 | contents\_detail.xml → AR

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
03     android:layout_width="match_parent"
04     android:layout_height="match_parent"
05     android:orientation="vertical"
06     android:paddingBottom="@dimen/activity_vertical_margin"
07     android:paddingLeft="@dimen/activity_horizontal_margin"
08     android:paddingRight="@dimen/activity_horizontal_margin"
09     android:paddingTop="70dp" >
10
11     <LinearLayout
12         android:orientation="horizontal"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content" >
15         <TextView
16             android:layout_width="60sp"
17             android:layout_height="wrap_content"
18             android:text="@string/title" />
19         <TextView
20             android:id="@+id/title"
21             android:layout_width="match_parent"
22             android:layout_height="wrap_content" />
23     </LinearLayout>
24

```

사진 제목 출력을 위한  
리니어 레이아웃

AS T

```

25 <LinearLayout
26     android:orientation="horizontal"
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content" >
29     <TextView
30         android:layout_width="60sp"
31         android:layout_height="wrap_content"
32         android:text="@string/orientation" />
33     <TextView
34         android:id="@+id/orientation"
35         android:layout_width="match_parent"
36         android:layout_height="wrap_content" />
37 </LinearLayout>

```

사진 방향 출력을 위한  
리니어 레이아웃

```

39 <LinearLayout
40     android:orientation="horizontal"
41     android:layout_width="match_parent"
42     android:layout_height="wrap_content" >
43     <TextView
44         android:layout_width="60sp"
45         android:layout_height="wrap_content"
46         android:text="@string/background" />
47     <TextView
48         android:id="@+id/background"
49         android:layout_width="wrap_content"
50         android:layout_height="wrap_content" />
51 </LinearLayout>

```

사진 배경 출력을 위한  
리니어 레이아웃

```

53 <LinearLayout
54     android:orientation="horizontal"
55     android:layout_width="match_parent"
56     android:layout_height="wrap_content" >
57     <TextView
58         android:layout_width="60sp"
59         android:layout_height="wrap_content"
60         android:text="@string/photo" />
61     <ImageView
62         android:id="@+id/photo"
63         android:layout_width="200dp"
64         android:layout_height="wrap_content"
65         android:adjustViewBounds="true" />
66 </LinearLayout>
67
68 </LinearLayout>

```

사진 이미지 출력을 위한  
리니어 레이아웃

#### 4 액티비티 제어

전달받은 정보를 추출하고 출력하며, '홈'과 '사진 등록'으로 링크하는 메뉴를 설정한다.

소스 | ContentsDetail.java

AW

```
01 package com.example.yschang.contentsmgt;
02
03 import android.content.Intent;
04 import android.graphics.Bitmap;
05 import android.graphics.BitmapFactory;
06 import android.os.Bundle;
07 import android.support.v7.app.AppCompatActivity;
08 import android.support.v7.widget.Toolbar;
09 import android.view.Menu;
10 import android.view.MenuItem;
11 import android.widget.ImageView;
12 import android.widget.TextView;
13
14 public class ContentsDetail extends AppCompatActivity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_detail);
20         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
21         setSupportActionBar(toolbar);
22
23         TextView tv_title = (TextView) findViewById(R.id.title);
24         TextView tv_orientation = (TextView) findViewById(R.id.orientation);
25         TextView tv_background = (TextView) findViewById(R.id.background);
26         ImageView iv_photo = (ImageView) findViewById(R.id.photo);
27
28         Intent it = getIntent();
29         String str_title = it.getStringExtra("it_title");
30         String str_orientation = it.getStringExtra("it_orientation");
31         String str_background1 = it.getStringExtra("it_background1");
32         String str_background2 = it.getStringExtra("it_background2");
33         String str_background3 = it.getStringExtra("it_background3");
34         String str_path = it.getStringExtra("it_path");
35
36         tv_title.setText(str_title);
37         tv_orientation.setText(str_orientation);
38         tv_background.setText(str_background1 + " ");
39         tv_background.append(str_background2 + " ");
40         tv_background.append(str_background3);
41     }
42 }
```

사진정보 출력을  
위한 뷰들 인식

인텐트로부터 전달받  
은 사진정보 추출

사진 제목, 방향,  
배경 값 출력

```

41
42 BitmapFactory.Options bmOptions = new BitmapFactory.Options();
43 Bitmap bm = BitmapFactory.decodeFile(str_path, bmOptions);
44
45 int WIDTH = 200;
46 int width = WIDTH;
47 float ratio =(float) bm.getHeight() / bm.getWidth();
48 int height =(int) (WIDTH * ratio);
49
50 Bitmap thumb = Bitmap.createScaledBitmap(bm, width, height, true);
51 iv_photo.setImageBitmap(thumb);
52
53 }
54
55 @Override
56 public boolean onCreateOptionsMenu(Menu menu) {
57     // Inflate the menu; this adds items to the action bar if it is present.
58     getMenuInflater().inflate(R.menu.menu_detail, menu);
59     return true;
60 }
61
62 @Override
63 public boolean onOptionsItemSelected(MenuItem item) {
64     int id = item.getItemId();
65
66     if(id == R.id.action_settings2) {
67         Intent it = new Intent(this, MainActivity.class);
68         startActivity(it);
69         finish();
70         return true;
71     }
72
73     if(id == R.id.action_settings1) {
74         Intent it = new Intent(this, ContentsRegForm.class);
75         startActivity(it);
76         finish();
77         return true;
78     }
79
80     return super.onOptionsItemSelected(item);
81 }
82 }

```

사진 이미지 출력

경로에 해당하는 이미지 추출

이미지 너비 설정

이미지 너비에 맞는 비율로 높이 조정

샘네일 이미지 생성

샘네일 이미지 출력

## 클래스와 속성/메소드

### ● 클래스

클래스/인터페이스	설명
<code>BitmapFactory</code>	파일 또는 바이트 배열을 포함하는 다양한 소스로부터 BitMap 객체를 만들
<code>BitmapFactory.Options</code>	42행. <code>BitmapFactory</code> 의 내부 클래스

### ● 메소드

클래스	메소드	설명
<code>BitmapFactory</code>	static <code>Bitmap decodeFile(String pathName, BitmapFactory.Options opts)</code>	43행. 파일 경로를 bitmap으로 변환함

## 5 환경 설정

`AndroidManifest.xml`에 새로 만든 액티비티 클래스인 `ContentsDetail`을 등록한다.

### 소스 | AndroidManifest.xml

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.example.yschang.contentsmgmt">
04
05     <uses-permission android:name=
06         "android.permission.READ_EXTERNAL_STORAGE" />
07     <uses-permission android:name=
08         "android.permission.WRITE_EXTERNAL_STORAGE" />
09     <uses-permission android:name="android.permission.CAMERA" />
10     <permission android:name="android.permission.FLASHLIGHT" />
11     <uses-feature android:name="android.hardware.camera"/>
12
13     <application
14         android:allowBackup="true"
15         android:icon="@mipmap/ic_launcher"
16         android:label="@string/app_name"
17         android:supportRtl="true"
18         android:theme="@style/AppTheme">
19
20         <activity
21             android:name=".MainActivity"
22             android:label="@string/app_name"
23             android:theme="@style/AppTheme.NoActionBar">

```

앱이 외부 저장소를 읽도록 허용

앱이 외부 저장소를 쓸 수 있도록 허용

카메라 디바이스 접근 허용

카메라 플래시 사용 허용

카메라가 없는 디바이스의 경우에는 앱이 설치되지 않도록 설정

```

21     <intent-filter>
22         <action android:name="android.intent.action.MAIN" />
23
24         <category android:name="android.intent.category.LAUNCHER" />
25     </intent-filter>
26 </activity>
27 <activity
28     android:name=".ContentsRegForm"
29     android:label="@string/act_reg_form"
30     android:theme="@style/AppTheme.NoActionBar">
31 </activity>
32 <activity
33     android:name=".ContentsDetail"
34     android:label="@string/act_detail"
35     android:theme="@style/AppTheme.NoActionBar">
36 </activity>
37 </application>
38
39 </manifest>

```

'사진 정보' 액티비티

'사진 등록' 액티비티

### STEP 3 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭