

# 08장 파일 처리



# 학습목표

---

- ❖ 파일을 처리하는 방식을 이해한다.
- ❖ SD 카드의 파일을 처리하는 방식을 이해한다.
- ❖ 커스텀뷰 작성법을 익힌다.

# 차례

---

1. 파일 처리의 기본
2. 파일 처리의 응용

# 01 내장 메모리 파일 처리

- 내장 메모리
  - 앱을 종료했다가 다음에 다시 실행할 때 사용했던 곳부터 이어서 작업하고 싶은 경우 내장 메모리에 파일을 저장하고 읽어오는 방식을 활용함
  - 내장 메모리의 저장 위치 : /data/data/패키지명/files 폴더임
  - 파일 읽기 : Context 클래스의 `openFileInput( )` 메소드 사용
    - `FileInputStream`을 반환함
  - 파일 쓰기 : `openFileOutput( )` 메소드 사용
    - `FileOutputStream`을 반환함
  - JDK에서 제공하는 파일을 읽거나 쓰는 `java.io.FileInputStream` 클래스와 `java.io.FileOutputStream`의 `read( )`, `write( )` 메소드를 사용하여 파일 처리

# 01 내장 메모리 파일 처리

- 내장 메모리에서 파일을 읽거나 쓰는 일반적인 절차

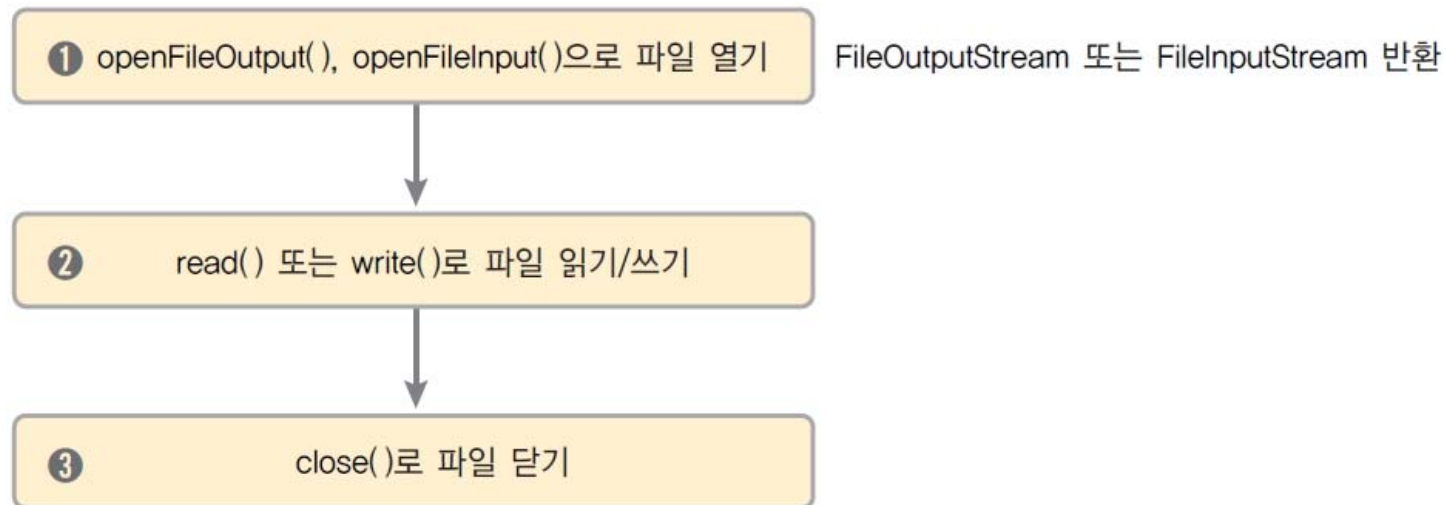


그림 8-1 내장 메모리에서의 파일 처리

# 01 내장 메모리 파일 처리

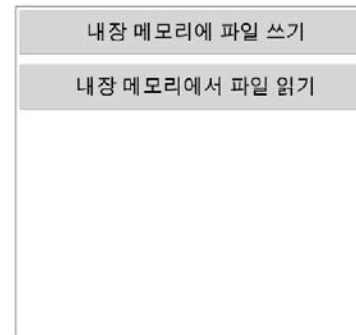
## – 파일을 쓰고 읽는 기본 예제

예제 8-1 파일 처리의 기본 activity\_main.xml

```

1 <LinearLayout>
2     <Button
3         android:id="@+id/btnWrite"
4         android:text="내장 메모리에 파일 쓰기" />
5     <Button
6         android:id="@+id/btnRead"
7         android:text="내장 메모리에서 파일 읽기" />
8 </LinearLayout>

```



예제 8-2 파일 처리의 기본 Kotlin 코드

```

1 public override fun onCreate(savedInstanceState: Bundle?) {
2     super.onCreate(savedInstanceState)
3     setContentView(R.layout.activity_main)
4
5     var btnRead: Button = findViewById<Button>(R.id.btnRead)
6     var btnWrite: Button = findViewById<Button>(R.id.btnWrite)
7
8     btnWrite.setOnClickListener {
9         var outFs : FileOutputStream =
10             openFileOutput("file.txt", Context.MODE_PRIVATE)
11         var str = "북북 안드로이드"
12         outFs.write(str.toByteArray())

```

# 01 내장 메모리 파일 처리

```
13      outFs.close()
14      Toast.makeText(applicationContext,
15          "file.txt가 생성됨", Toast.LENGTH_SHORT).show()
16  }
17
18  btnRead.setOnClickListener {
19      try {
20          var inFs : FileInputStream = openFileInput("file.txt")
21          var txt = ByteArray(30)
22          inFs.read(txt)
23          var str = txt.toString(Charsets.UTF_8)
24          Toast.makeText(applicationContext, str, Toast.LENGTH_SHORT).show()
25          inFs.close()
26      } catch (e : IOException) {
27          Toast.makeText(applicationContext,
28              "파일 없음", Toast.LENGTH_SHORT).show()
29      }
30  }
31 }
```

# 01 내장 메모리 파일 처리

## ■ <실습 8-1> 간단 일기장 앱 만들기

- 날짜를 선택했을 때 그 날짜의 일기가 없으면 일기를 쓸 수 있고, 그 날짜의 일기가 있으면 일기 내용을 보여주도록 만들기

## ■ 안드로이드 프로젝트 생성

- (1) 새 프로젝트 만들기
  - 프로젝트 이름 : 'Project8\_1'
  - 패키지 이름 : 'com.cookandroid.project8\_1'
  - 그 외 규칙은 [실습 2-4]의 (1)~(4)를 따름

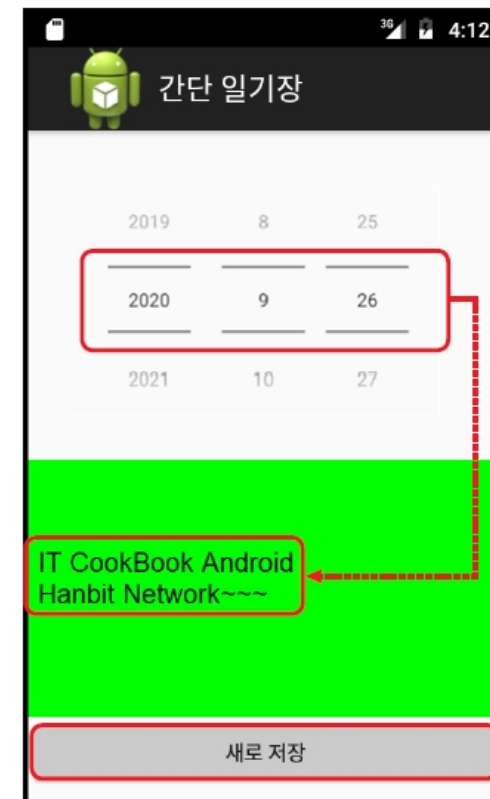


그림 8-2 간단 일기장 앱 결과 화면



# 01 내장 메모리 파일 처리

- 화면 디자인 및 편집
  - (2) activity\_main.xml을 코딩함. 다음과 같이 화면을 구성함
    - 데이트피커, 에디트텍스트, 버튼을 1개씩 생성
    - 각 위젯의 id : datePicker1, edtDiary, btnWrite

예제 8-3 activity\_main.xml

```

1 <LinearLayout>
2     <DatePicker
3         android:id="@+id/datePicker1"
4         android:calendarViewShown="false" />
5     <EditText
6         android:id="@+id/edtDiary"
7         android:background="#00ff00"
8         android:lines="8" />
9     <Button
10        android:id="@+id/btnWrite"
11        android:enabled="false"
12        android:text="Button" />
13 </LinearLayout>

```



# 01 내장 메모리 파일 처리

- Kotlin 코드 작성 및 수정
  - (3) MainActivity.kt를 코딩함
    - 다음과 같은 변수를 전역변수로 선언함
      - activity\_main.xml의 3개 위젯에 대응할 위젯 변수 3개 선언
      - 파일 이름을 지정할 문자열 변수 1개 선언
        - » 파일 이름 : '연\_월\_일.txt'
      - 위젯 변수에 activity\_main.xml의 위젯 대입

# 01 내장 메모리 파일 처리

예제 8-4 Kotlin 코드 1

```
1  ~~~ 생략(import 문) ~~~
2  class MainActivity : AppCompatActivity() {
3      lateinit var dp : DatePicker
4      lateinit var edtDiary : EditText
5      lateinit var btnWrite : Button
6      lateinit var fileName : String
7
8      override fun onCreate(savedInstanceState: Bundle?) {
9          super.onCreate(savedInstanceState)
10         setContentView(R.layout.activity_main)
11         title = "간단 일기장"
12
13         dp = findViewById<DatePicker>(R.id.datePicker1)
14         edtDiary = findViewById<EditText>(R.id.edtDiary)
15         btnWrite = findViewById<Button>(R.id.btnWrite)
16
17     }
18 }
```

# 01 내장 메모리 파일 처리

- (4) onCreate() 내부에 계속 코딩함(데이트피커 설정)
  - Calendar 클래스 이용 - 현재 날짜의 연, 월, 일을 구한 후 데이트피커 초기화
  - 데이트피커의 날짜가 변경되면 변경된 날짜에 해당하는 일기 파일(연\_월\_일.txt)의 내용을 에디트텍스트에 보여줌

예제 8-5 Kotlin 코드 2

```

1      ~~~ 생략([예제 8-4]와 동일) ~~~
2      btnWrite = findViewById<Button>(R.id.btnWrite)
3
4      var cal = Calendar.getInstance()
5      var cYear = cal.get(Calendar.YEAR)
6      var cMonth = cal.get(Calendar.MONTH)
7      var cDay = cal.get(Calendar.DAY_OF_MONTH)
8
9      dp.init(cYear, cMonth, cDay) { view, year, monthOfYear, dayOfMonth ->
10         fileName = (Integer.toString(year) + "_"
11             + Integer.toString(monthOfYear + 1) + "_"
12             + Integer.toString(dayOfMonth) + ".txt")
13         var str = readDiary(fileName)
14         edtDiary.setText(str)
15         btnWrite.isEnabled = true
16     }
17 }
18
19 fun readDiary(fName: String) : String? {
20     return null
21 }
22 }
```

# 01 내장 메모리 파일 처리

- (5) 현재 날짜의 파일(연\_월\_일.txt)을 읽어 일기의 내용을 반환하는 readDiary() 메소드 완성

예제 8-6 Kotlin 코드 3

```
1 fun readDiary(fName: String) : String? {
2     var diaryStr : String? = null
3     var inFs : FileInputStream
4     try {
5         inFs = openFileInput(fName)
6         var txt = ByteArray(500)
7         inFs.read(txt)
8         inFs.close()
9         diaryStr = txt.toString(Charsets.UTF_8).trim()
10        btnWrite.text = "수정하기"
11    } catch (e : IOException) {
12        edtDiary.hint = "일기 없음"
13        btnWrite.text = "새로 저장"
14    }
15    return diaryStr
16 }
```

# 01 내장 메모리 파일 처리

- (6) 맨 아래 버튼을 클릭했을 때 동작하는 내용을 onCreate() 안에 코딩
  - readDiary( )에 의해 <수정하기> 또는 <새로 저장>으로 나타남
  - 에디트텍스트의 내용을 새로 입력하거나 기존의 내용을 수정한 후에 클릭하면 내용이 현재 날짜의 파일(연\_월\_일.txt)로 저장됨

예제 8-7 Kotlin 코드 4

```
1 btnWrite.setOnClickListener {
2     var outFs = openFileOutput(fileName, Context.MODE_PRIVATE)
3     var str = edtDiary.text.toString()
4     outFs.write(str.toByteArray())
5     outFs.close()
6     Toast.makeText(applicationContext, "$fileName 이 저장됨",
7         Toast.LENGTH_SHORT).show()
8 }
```

# 01 내장 메모리 파일 처리

- 프로젝트 실행 및 결과 확인
  - (7) 완성된 코드를 실행해 봄
    - 1) 선택한 날짜에 쓴 일기가 있는 경우
      - 일기 내용이 보이고 버튼이 <수정하기>로 바뀜
    - 2) 선택한 날짜의 일기가 없는 경우
      - 에디트텍스트에 '일기 없음' 힌트가 보이고 버튼이 <새로 저장>으로 바뀜

# 01 내장 메모리 파일 처리

- 안드로이드 응용 프로그램 개발 완료
  - (8) AVD의 초기화면으로 돌아감
  - (9) Android Studio에서 제공하는 툴을 사용하여 일기가 쓰인 파일을 직접 확인해보기
    - Android Studio 화면 오른쪽 아래의 [Device File Explorer] 탭을 클릭하거나, [View]-[Tool Windows]-[Device File Explorer]를 선택함

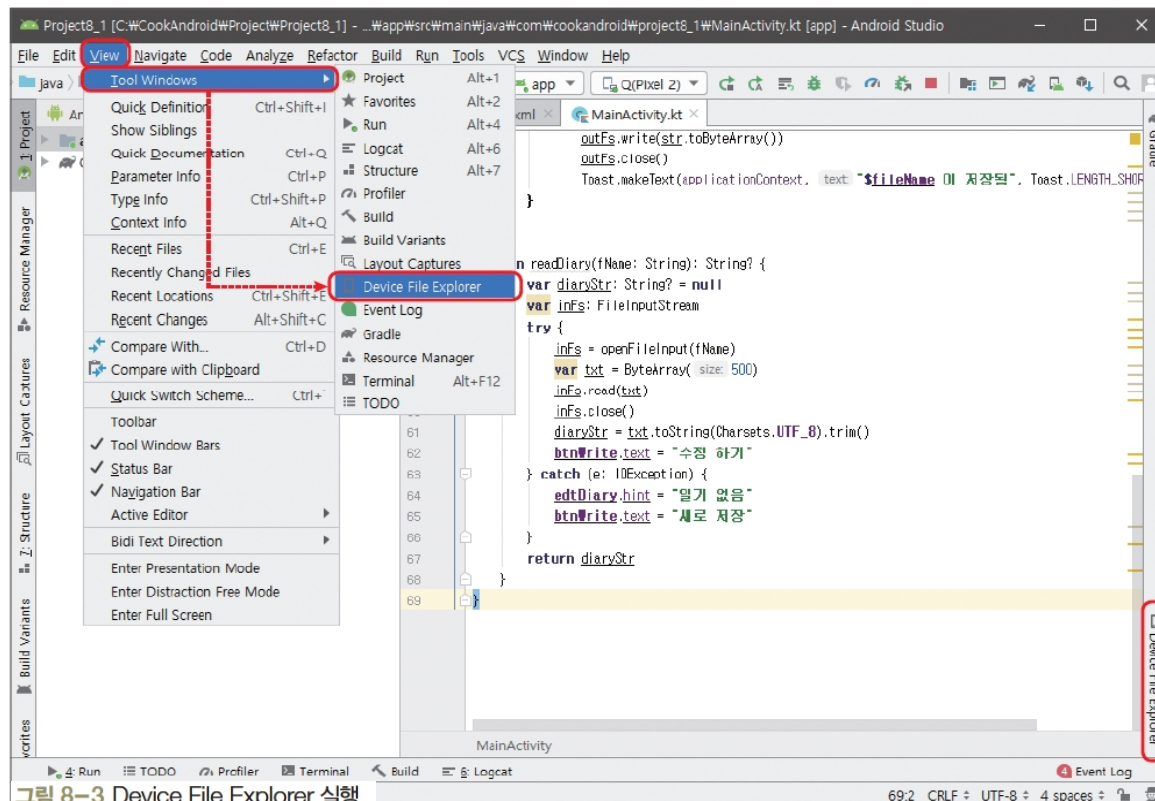


그림 8-3 Device File Explorer 실행



# 01 내장 메모리 파일 처리

- (10) [Device File Explorer] 창의 위쪽에는 AVD가 보이고 아래쪽에는 AVD 내부의 폴더와 파일 목록이 보임
- [data]-[data]-[com.cookandroid.project8\_1]-[files]를 확장하면 생성한 일기 파일 목록이 보임

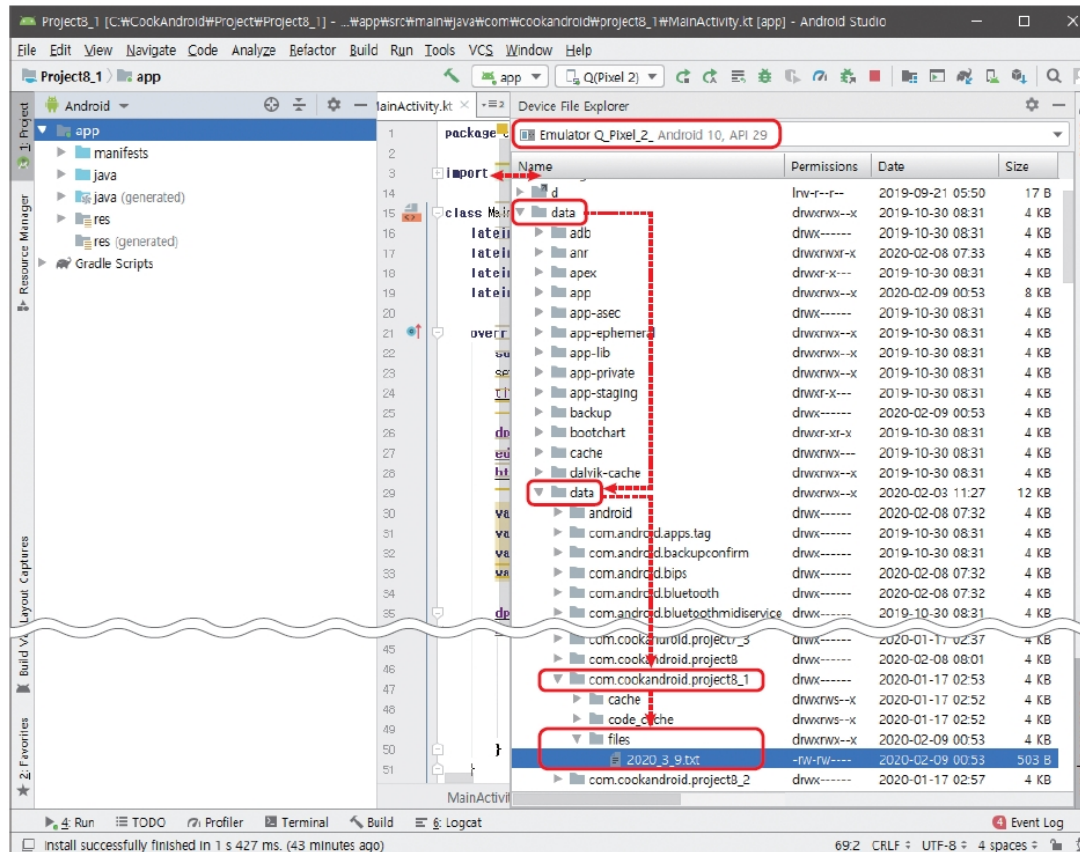


그림 8-4 일기 파일 확인

# 01 내장 메모리 파일 처리

- (11) 파일을 PC로 가져올 수도 있음
  - PC로 가져올 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 [Save As]를 선택하고 PC의 폴더를 지정

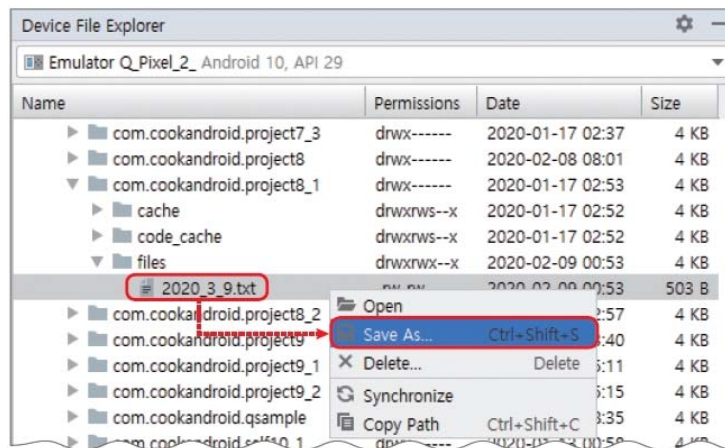


그림 8-5 파일 가져오기

- (12) 가져온 일기는 \*.txt 파일로 메모장에서 읽을 수 있음
- (13) 오른쪽 아래의 [Device File Explorer]를 클릭하여 창을 닫음

# 01 내장 메모리 파일 처리

## ▶ 직접 풀어보기 8-1

[실습 8-1]을 처음 실행하면 해당 날짜의 일기가 있어도 나오지 않는다. 처음 실행할 때부터 그날의 일기가 있으면 에디트텍스트에 일기 내용을 출력하고 버튼이 <수정하기>가 되게 한다. 그리고 일기가 없으면 에디트텍스트에 '일기 없음'이라는 힌트가 나오고 버튼은 <새로 저장>이 되게 한다.

## 02 raw 폴더 파일 처리

- /res/raw 폴더에 필요한 파일을 저장하여 사용하는 방법
  - 기본적으로 /res 아래에 raw 폴더를 생성하고 프로젝트에서 사용할 파일을 넣어둠
    - Kotlin 코드에서 openRawResource() 메소드를 사용하여 접근 할 수 있음
    - FileInputStream 클래스 대신 InputStream 클래스를 사용함
  - /res/raw는 프로젝트에 포함된 폴더이므로 읽기 전용으로만 사용 가능
    - 쓰기는 내장 메모리나 SD 카드를 사용해야 함

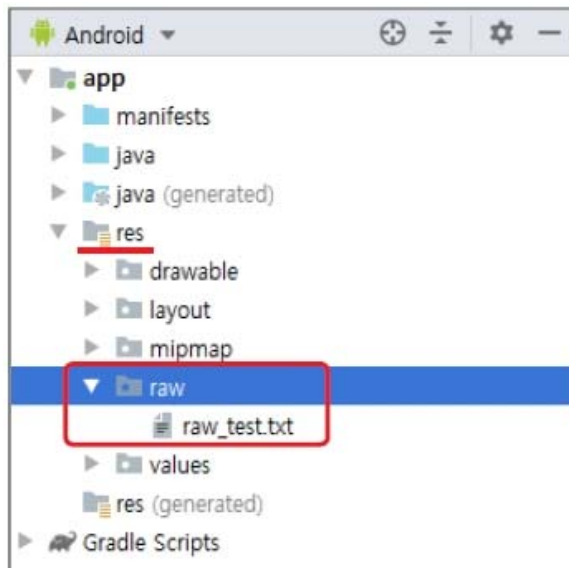


그림 8-6 /res/raw 폴더 생성

## 02 raw 폴더 파일 처리

- raw 폴더 파일 처리의 간단한 예제
  - 미리 프로젝트의 res 폴더에 raw 폴더를 생성하고, 임의의 내용을 입력한 \*.txt 파일을 하나 복사해 놓기

예제 8-8 /res/raw 폴더의 파일 읽기 activity\_main.xml

```
1 <LinearLayout>
2     <Button
3         android:id="@+id/btnRead"
4         android:text="/res/raw에서 파일 읽기" />
5     <EditText
6         android:id="@+id/edtRaw"
7         android:lines="10" />
8 </LinearLayout>
```

## 02 raw 폴더 파일 처리

예제 8-9 /res/raw 폴더의 파일 읽기 Kotlin 코드

```
1 public override fun onCreate(savedInstanceState: Bundle?) {
2     super.onCreate(savedInstanceState)
3     setContentView(R.layout.activity_main)
4
5     var btnRead: Button
6     var edtRaw: EditText
7     btnRead = findViewById<Button>(R.id.btnRead)
8     edtRaw = findViewById<EditText>(R.id.edtRaw)
9
10    btnRead.setOnClickListener {
11        var inputS = resources.openRawResource(R.raw.raw_test)
12        var txt = ByteArray(inputS.available())
13        inputS.read(txt)
14        edtRaw.setText(txt.toString(Charsets.UTF_8))
15        inputS.close()
16    }
17 }
```

/RES/RAW에서 파일 읽기

IT Cookbook. 안드로이드를 학습하고 있습니다.

안드로이드 프로그래밍은 생각보다 쉽습니다. ^^

# 파일 처리의 응용

## 가상 SD 카드 확인

- MP3 파일처럼 여러 응용 프로그램에서 사용되는 경우 SD 카드에 저장해 활용
- 안드로이드는 SD 카드에 저장된 데이터에 특별한 인증 절차 없이 접근 가능
- AVD 이름을 선택하고 [Edit]를 클릭하면 사용하는 SD 카드를 확인 가능

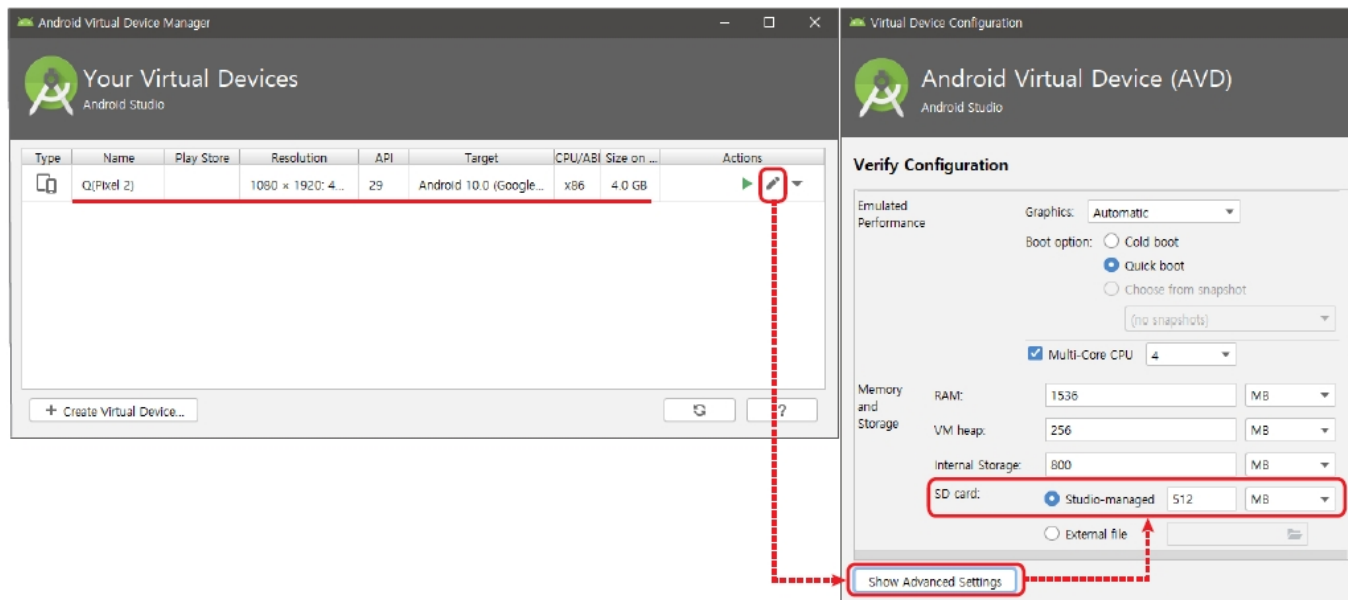


그림 8-7 가상의 SD 카드 확인(Pixel 2 AVD)

# 01 SD 카드에서 파일 읽기

- SD 카드에서 파일 읽기
  - (1) Device File Explorer에서 /sdcard 폴더 또는/storage/emulated/0 폴더에 텍스트 파일 올리기

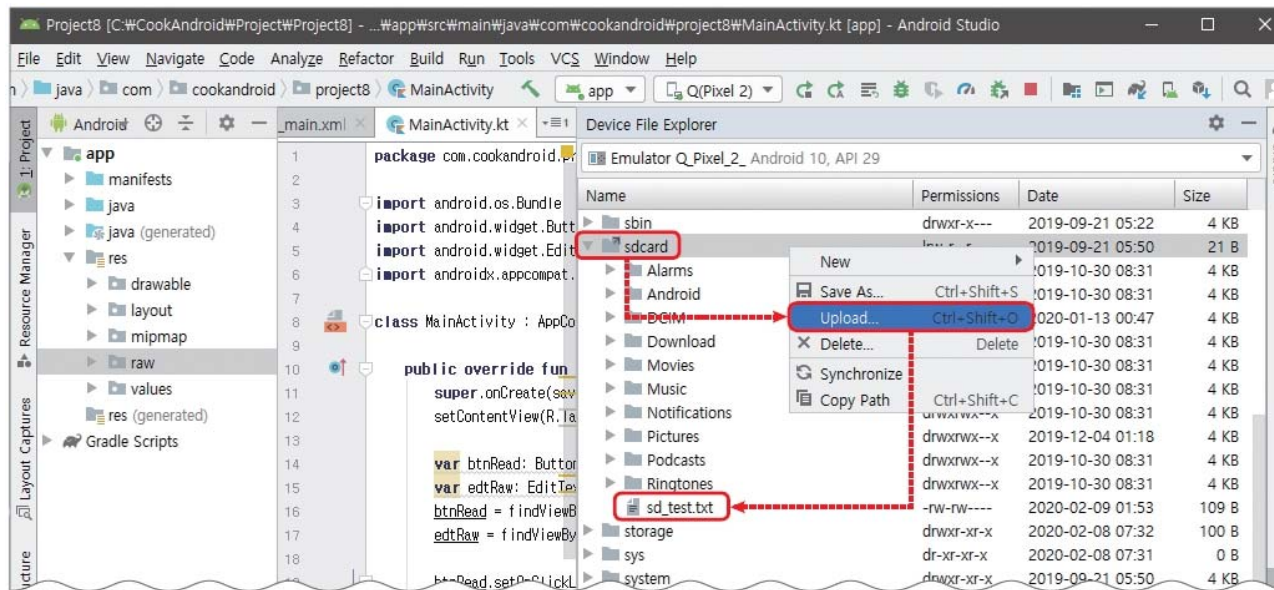


그림 8-8 SD 카드에 파일 넣기(Upload)



# 01 SD 카드에서 파일 읽기

- SD 카드에서 파일 읽기
  - (2) AndroidManifest.xml 파일에 SD 카드를 사용할 수 있도록 퍼미션 및 application에 관련 속성 추가

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<application
    android:requestLegacyExternalStorage="true"
```

## 예제 8-10 SD 카드에서 파일 읽기 activity\_main.xml

```
1 <LinearLayout>
2     <Button
3         android:id="@+id/btnRead"
4         android:text="SD 카드에서 파일 읽기" />
5     <EditText
6         android:id="@+id/edtSD"
7         android:lines="10" />
8 </LinearLayout>
```

# 01 SD 카드에서 파일 읽기

예제 8-11 SD 카드에서 파일 읽기 Kotlin 코드

```

1  public override fun onCreate(savedInstanceState: Bundle?) {
2      super.onCreate(savedInstanceState)
3      setContentView(R.layout.activity_main)
4      ActivityCompat.requestPermissions(this, arrayOf(android.Manifest.
5          permission.WRITE_EXTERNAL_STORAGE), Context.MODE_PRIVATE)
6
7      var btnRead : Button
8      var edtSD : EditText
9      btnRead = findViewById<Button>(R.id.btnRead)
10     edtSD = findViewById<EditText>(R.id.edtSD)
11
12     btnRead.setOnClickListener {
13         var inFs = FileInputStream("/storage/emulated/0/sd_test.txt")
14         var txt = ByteArray(inFs.available())
15         inFs.read(txt)
16         edtSD.setText(txt.toString(Charsets.UTF_8))
17         inFs.close()
18     }
19 }

```

## SD 카드에서 파일 읽기

IT Cookbook. 안드로이드를 학습하고 있습니다.

이 파일은 SD카드 테스트용입니다. ^^

## 02 SD 카드에 폴더 및 파일 생성

- SD 카드에 폴더 및 파일 생성 예제
  - Environment 클래스의 정적 메소드를 이용해 SD카드의 동작 여부 및 관련 폴더 경로 구함

예제 8-12 SD 카드에 폴더 및 파일 생성 activity\_main.xml

```
1 <LinearLayout>
2     <Button
3         android:id="@+id/btnMkdir"
4         android:text="SD 카드에 디렉터리 생성" />
5     <Button
6         android:id="@+id/btnRmdir"
7         android:text="SD 카드에서 디렉터리 삭제" />
8 </LinearLayout>
```

## 02 SD 카드에 폴더 및 파일 생성

예제 8-13 SD 카드에 폴더 및 파일 생성 Kotlin 코드

```

1  public override fun onCreate(savedInstanceState: Bundle?) {
2      super.onCreate(savedInstanceState)
3      setContentView(R.layout.activity_main)
4      ActivityCompat.requestPermissions(this, arrayOf(android.Manifest.
5          permission.WRITE_EXTERNAL_STORAGE), Context.MODE_PRIVATE)
6
7      var btnMkdir : Button
8      var btnRmdir : Button
9      btnMkdir = findViewById<Button>(R.id.btnMkdir)
10     btnRmdir = findViewById<Button>(R.id.btnRmdir)
11
12     var strSDpath = Environment.getExternalStorageDirectory().absolutePath
13     var myDir = File("$strSDpath/mydir")
14
15     btnMkdir.setOnClickListener {
16         myDir.mkdir()
17     }
18
19     btnRmdir.setOnClickListener {
20         myDir.delete()
21     }
22 }

```

SD 카드에 디렉터리 생성

SD 카드에서 디렉터리 삭제

Device File Explorer

Emulator Q\_Pixel\_2\_ Android 10, API 29

Name	Permissions	Date	Size
res	drwxr-xr-x	2019-09-21 05:39	4 KB
sbin	drwxr-x---	2019-09-21 05:22	4 KB
sdcard	lrwxr--r--	2019-09-21 05:50	21 B
Alarms	drwxrwx--x	2019-10-16 01:02	4 KB
Android	drwxrwx--x	2019-10-16 01:02	4 KB
DCIM	drwxrwx--x	2019-10-17 07:10	4 KB
Download	drwxrwx--x	2019-10-16 01:02	4 KB
Movies	drwxrwx--x	2019-10-16 01:02	4 KB
Music	drwxrwx--x	2019-10-16 01:02	4 KB
mydir	drwxrwx--x	2019-10-25 03:21	4 KB
Notifications	drwxrwx--x	2019-10-16 01:02	4 KB
Pictures	drwxrwx--x	2019-10-16 01:02	4 KB
Podcasts	drwxrwx--x	2019-10-16 01:02	4 KB
Ringtones	drwxrwx--x	2019-10-16 01:02	4 KB
sd_test.txt	-rw-rw----	2019-10-25 02:22	109 B
storage	drwxr-xr-x	2019-10-25 00:34	100 B
sys	dr-xr-xr-x	2019-10-25 00:34	0 B
system	drwxr-xr-x	2019-09-21 05:50	4 KB

## 03 특정 폴더의 하위 폴더 및 파일 목록

- File.listFiles() 메소드
  - 지정한 폴더의 하위 폴더 및 파일 목록에 접근함
  - 반환값 : File[ ] 형

예제 8-14 시스템 폴더의 폴더/파일 목록 activity\_main.xml

```
1 <LinearLayout>
2     <Button
3         android:id="@+id/btnFilelist"
4         android:text="시스템 폴더의 폴더/파일 목록" />
5     <EditText
6         android:id="@+id/edtFilelist" />
7 </LinearLayout>
```

## 03 특정 폴더의 하위 폴더 및 파일 목록

예제 8-15 시스템 폴더의 폴더/파일 목록 Kotlin 코드

```

1 public override fun onCreate(savedInstanceState: Bundle?) {
2     super.onCreate(savedInstanceState)
3     setContentView(R.layout.activity_main)
4
5     var btnFilelist : Button
6     var edtFilelist : EditText
7     btnFilelist = findViewById<Button>(R.id.btnFilelist)
8     edtFilelist = findViewById<EditText>(R.id.edtFilelist)
9
10    btnFilelist.setOnClickListener {
11        var sysDir = Environment.getRootDirectory().absolutePath
12        var sysFiles = File(sysDir).listFiles()
13
14        var strFname: String
15        for (i in sysFiles.indices) {
16            if (sysFiles[i].isDirectory == true)
17                strFname = "<폴더> " + sysFiles[i].toString()
18            else
19                strFname = "<파일> " + sysFiles[i].toString()
20
21            edtFilelist.setText(edtFilelist.text.toString() + "\n" + strFname)
22        }
23    }
24 }

```

시스템 폴더의 폴더/파일 목록

```

<폴더> /system/fonts
<폴더> /system/product
<폴더> /system/priv-app
<파일> /system/build.prop
<폴더> /system/etc
<폴더> /system/vendor
<폴더> /system/lib
<폴더> /system/app
<폴더> /system/bin
<폴더> /system/sbin
<폴더> /system/usr
<폴더> /system/framework
<폴더> /system/apex

```

## 03 특정 폴더의 하위 폴더 및 파일 목록

### ■ <실습 8-2> 간단 이미지 뷰어 만들기

- SD 카드 특정 폴더의 이미지 파일을 보여주는 간단한 이미지 뷰어 앱을 만들어봄

### ■ 안드로이드 프로젝트 생성

- (1) 새 프로젝트 만들기
  - 프로젝트 이름 : 'Project8\_2'
  - 패키지 이름 : 'com.cookandroid.project8\_2'
  - 그 외 규칙은 [실습 2-4]의 (1)~(4)를 따름

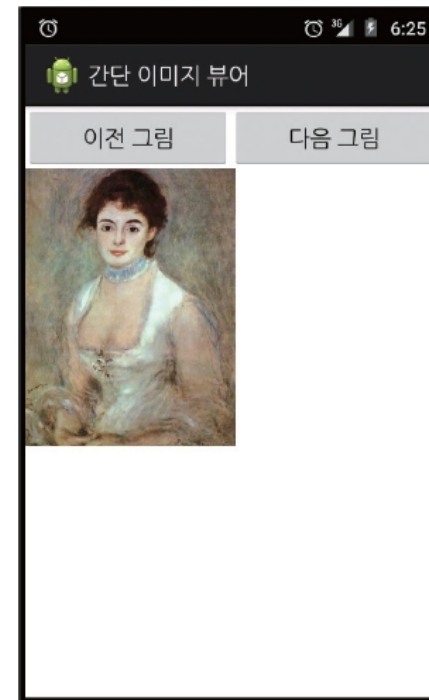


그림 8-9 간단 이미지 뷰어 앱 결과 화면

## 03 특정 폴더의 하위 폴더 및 파일 목록

- 화면 디자인 및 편집
  - (2) 이번 실습은 커스텀 위젯 (Custom Widget, Custom View)을 직접 만들어 activity\_main.xml에 넣어 사용해야 함
    - 이번에 만들 커스텀 위젯은 지정된 이미지 파일을 출력하는 역할을 함
    - [app]-[java]-[패키지 이름]에서 마우스 오른쪽 버튼을 클릭
    - [New]-[Kotlin File/ Class]를 선택하고 아래 정보 입력
      - Name : 'myPictureView'
      - Kind : 'Class'

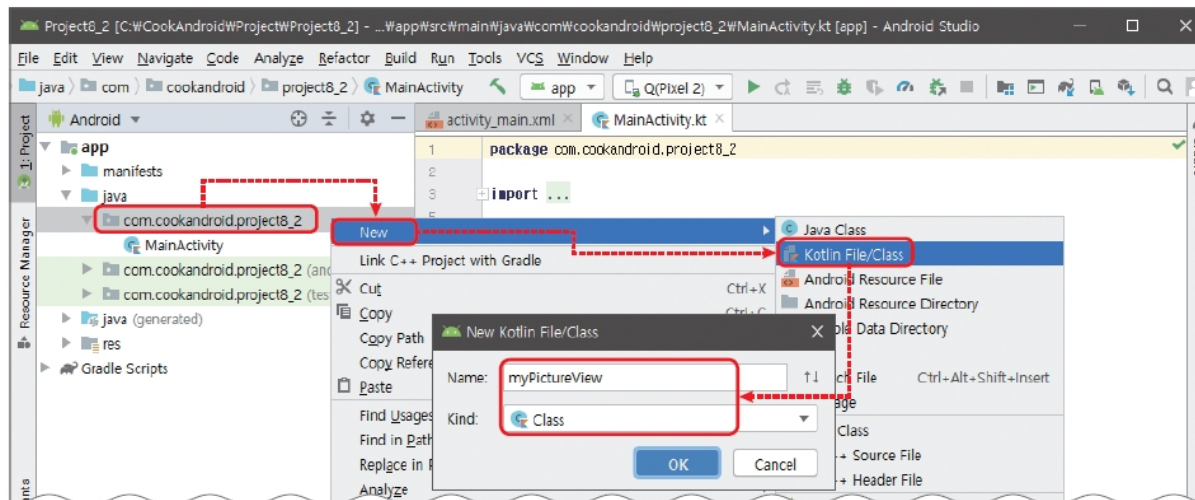


그림 8-10 클래스 생성



## 03 특정 폴더의 하위 폴더 및 파일 목록

- (3) 생성된 코드를 다음과 같이 수정함

예제 8-16 myPictureView 클래스의 Kotlin 코드 1

```
1  ~~~ 생략(import 문) ~~~
2  class myPictureView(context: Context, attrs: AttributeSet?) :
3      View(context, attrs) {
4  }
```

- (4) onDraw( ) 메소드를 오버라이딩함
  - 자동 완성된 코드 외에 나머지를 완성하고 저장함

## 03 특정 폴더의 하위 폴더 및 파일 목록

예제 8-17 myPictureView 클래스의 Kotlin 코드 2

```
1  ~~~ 생략(import 문) ~~~
2  class myPictureView(context: Context, attrs: AttributeSet?) :
3      View(context, attrs) {
4      var imagePath : String? = null
5
6      override fun onDraw(canvas: Canvas) {
7          super.onDraw(canvas)
8          try {
9              if (imagePath != null) {
10                  var bitmap = BitmapFactory.decodeFile(imagePath)
11                  canvas.scale(2f, 2f, 0f, 0f)
12                  canvas.drawBitmap(bitmap!!, 0f, 0f, null)
13                  bitmap!!.recycle()
14              }
15          } catch (e : Exception) {
16          }
17      }
18 }
```

## 03 특정 폴더의 하위 폴더 및 파일 목록

- (5) activity\_main.xml을 코딩함. 다음과 같이 화면을 구성함.
  - 가로 레이아웃에 버튼 2개 생성
  - 앞에서 생성한 커스텀 위젯인 myPictureView 생성
  - 위젯의 이름 : btnPrev, btnNext, myPictureView1

예제 8-18 activity\_main.xml

```

1 <LinearLayout>
2     <LinearLayout
3         android:orientation="horizontal" >
4         <Button
5             android:id="@+id/btnPrev"
6             android:layout_weight="1"
7             android:text=" 이전 그림 " />
8         <Button
9             android:id="@+id/btnNext"
10            android:layout_weight="1"
11            android:text=" 다음 그림 " />
12     </LinearLayout>
13     <com.cookandroid.project8_2.myPictureView
14         android:id="@+id/myPictureView1" />
15 </LinearLayout>

```



## 03 특정 폴더의 하위 폴더 및 파일 목록

- (6) 그림 파일을 SD 카드에 미리 넣어둠
  - [Device File Explorer]에서 /storage/emulated/0/Pictures(또는 /sdcard/Pictures) 폴더에 적당한 이미지 파일을 몇 개 넣어둠
    - 만약 Pictures 폴더가 없다면 /storage/emulated/0(또는 /sdcard) 폴더에서 마우스 오른쪽 버튼을 클릭하고 [New]-[Directory]를 선택하여 Pictures 폴더를 만들기
  - Android Manifest.xml 파일에 SD 카드를 사용할 수 있도록 퍼미션을 지정

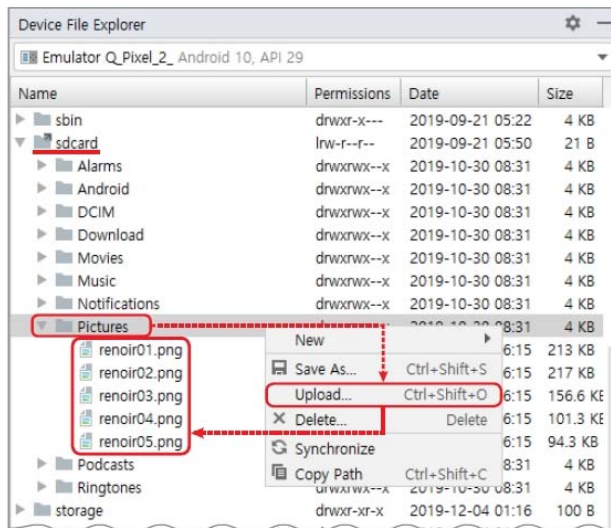


그림 8-11 SD 카드에 이미지 파일 복사

## 03 특정 폴더의 하위 폴더 및 파일 목록

- Kotlin 코드 작성 및 수정
  - (7) MainActivity.kt를 코딩함. 다음과 같은 변수를 전역변수로 선언
    - activity\_main.xml의 위젯 3개에 대응할 위젯 변수 3개 선언
    - SD 카드에서 읽어올 이미지 파일의 배열과 파일명을 저장한 문자열 변수 선언
    - 위젯 변수에 activity\_main.xml의 위젯을 대입

예제 8-19 Kotlin 코드 1

```

1  ~~~ 생략(import 문) ~~~
2  class MainActivity : AppCompatActivity() {
3      lateinit var btnPrev : Button
4      lateinit var btnNext : Button
5      lateinit var myPicture : myPictureView
6      var curNum : Int = 0
7      var imageFiles : Array<File>? = null
8      lateinit var imageFname : String
9
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13         title = "간단 이미지 뷰어"
14         ActivityCompat.requestPermissions(this, arrayOf(android.Manifest.
15             permission.WRITE_EXTERNAL_STORAGE), Context.MODE_PRIVATE)
16
17         btnPrev = findViewById<Button>(R.id.btnPrev)
18         btnNext = findViewById<Button>(R.id.btnNext)
19         myPicture = findViewById<myPictureView>(R.id.myPictureView1)
20
21     }
22 }

```

## 03 특정 폴더의 하위 폴더 및 파일 목록

- (8) onCreate( ) 안에 코딩함
- SD 카드의 /storage/emulated/0/Pictures(또는 /sdcard/Pictures) 폴더에서 파일을 읽어오고, 첫 번째 파일을 커스텀 위젯에 출력함

예제 8-20 Kotlin 코드 2

```

1      ~~~ 생략([예제 8-19]와 동일) ~~~
2      myPicture = findViewById<myPictureView>(R.id.myPictureView1)
3
4      imageFiles = File(Environment.getExternalStorageDirectory().
5                          absolutePath + "/Pictures").listFiles()
6      imageFname = imageFiles!![0].toString()
7      myPicture.imagePath=imageFname
8
9      }
10 }
```



## 03 특정 폴더의 하위 폴더 및 파일 목록

- (9) 버튼을 클릭하면 동작하는 람다식을 정의함

예제 8-21 Kotlin 코드 3

```

1  btnPrev.setOnClickListener {
2      if (curNum <= 0) {
3          Toast.makeText(applicationContext, "첫번째 그림입니다",
4              Toast.LENGTH_SHORT).show()
5      } else {
6          curNum--
7          imageFname = imageFiles!![curNum].toString()
8          myPicture.imagePath=imageFname
9          myPicture.invalidate()
10     }
11 }
12
13 btnNext.setOnClickListener {
14     if (curNum >= imageFiles!!.size - 1) {
15         Toast.makeText(applicationContext,
16             "마지막 그림입니다", Toast.LENGTH_SHORT).show()
17     } else {
18         curNum++
19         ~~~ 생략 ~~~

```



## 03 특정 폴더의 하위 폴더 및 파일 목록

- 프로젝트 실행 및 결과 확인
  - (10) 완성된 코드를 실행해봄. SD 카드에 복사한 이미지가 차례로 나타남

### ▶ 작업 풀어보기 8-2

[실습 8-2]를 다음과 같이 수정하라.

- 버튼 사이에 '현재 그림 번호/전체 그림 개수' 텍스트뷰가 나타나게 한다.
- 토스트 메시지를 없애고, 첫 번째 그림에서 <이전 그림>을 클릭하면 마지막 그림이, 마지막 그림에서 <다음 그림>을 클릭하면 첫 번째 그림이 나오게 한다.



그림 8-12 수정된 간단 이미지 뷰어 앱