

## 06. 고급 위젯 다루기

Android Studio를 활용한 안드로이드 프로그래밍

### 학습목표

---

- 고급 위젯을 다루는 방법을 익힌다.
- 뷰 컨테이너와 그 응용법을 이해한다.
- 매니페스트 파일 설정법을 익힌다.

## 차례

---

01 고급 위젯

02 뷰 컨테이너

Page • 3

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [1/15]

---

❖ 아날로그 시계, 디지털 시계

```
java.lang.Object
└─ android.view.View
    └─ android.widget.AnalogClock
        └─ android.widget.TextView
            └─ android.widget.DigitalClock
```

아날로그시계, 디지털시계 계층도

Page • 4

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [2/15]

### ❖ 아날로그 시계, 디지털 시계 예제

예제 6-1 시계 관련 XML 코드

```
1 <LinearLayout>
2     <AnalogClock
3         android:layout_width="match_parent"
4         android:layout_height="wrap_content" />
5     <DigitalClock
6         android:layout_width="match_parent"
7         android:layout_height="wrap_content"
8         android:gravity="center" />
9 </LinearLayout>
```



Page • 5

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [3/15]

### ❖ 크로노미터(Chronometer)

- ✓ 타이머 형식의 위젯
- ✓ 일반적으로 시간을 측정할 때 많이 사용

```
java.lang.Object
└─ android.view.View
    └─ android.widget.TextView
        └─ android.widget.Chronometer
```

크로노미터 계층도

Page • 6

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [4/15]

### ❖ 크로노미터 예제

예제 6-2 크로노미터의 XML 코드

```
1 <LinearLayout>
2     <Chronometer
3         android:id="@+id/chronometer1"
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content"
6         android:format="시간 측정 : %s"
7         android:gravity="center"
8         android:textSize="30dp" />
9 </LinearLayout>
```

시간 측정 : 00:00

Page • 7

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [5/15]

### ❖ 타임피커, 데이트피커, 캘린더뷰

- ✓ 시간 입력 시 사용
- ✓ 타임피커(TimePicker) : 시간을 표시, 조절
- ✓ 데이트피커(DatePicker)와 캘린더뷰(CalendarView) : 날짜를 표시, 조절

```
java.lang.Object
└─ android.view.View
    └─ android.view.ViewGroup
        └─ android.widget.FrameLayout
            └─ android.widget.TimePicker
                └─ android.widget.DatePicker
                    └─ android.widget.CalendarView
```

타임피커, 데이트피커, 캘린더뷰 계층도

Page • 8

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [6/15]

### ❖ 타임피커, 데이트피커 예제

예제 6-3 타임피커와 데이트피커의 XML 코드

```

1 <LinearLayout>
2   <TimePicker
3       android:timePickerMode="spinner"
4       android:layout_width="match_parent"
5       android:layout_height="wrap_content" />
6   <DatePicker
7       android:datePickerMode="spinner"
8       android:layout_width="match_parent"
9       android:layout_height="wrap_content" />
10 </LinearLayout>

```



Page • 9

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [12/15]

### 실습 6-1 날짜/시간 예약 앱 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ 라디오버튼 클릭 시 캘린더뷰와 타임피커 중 하나씩만 보이게 클릭 이벤트 리스너 작성

예제 6-7 Java 코드 3

```

1 rdoCal.setOnClickListener(new View.
  OnClickListener() {
2   public void onClick(View v) {
3       tPicker.setVisibility(View.INVISIBLE);
4       calView.setVisibility(View.VISIBLE);
5   }
6 });
7
8 rdoTime.setOnClickListener(new View.
  OnClickListener() {
9   public void onClick(View v) {
10      tPicker.setVisibility(View.VISIBLE);
11      calView.setVisibility(View.INVISIBLE);
12  }
13 });

```



Page • 10

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [13/15]

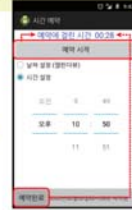
### 실습 6-1 날짜/시간 예약 앱 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ <예약 시작>을 클릭하면 크로노미터가 시작하고 <예약 완료>를 클릭하면 정지하도록 클릭이벤트 리스너를 작성

예제 6-8 Java 코드 4

```
1 btnStart.setOnClickListener(new View.OnClickListener() {
2     public void onClick(View v) {
3         chrono.setBase(SystemClock.elapsedRealtime());
4         chrono.start();
5         chrono.setTextColor(Color.RED);
6     }
7 });
8
9 btnEnd.setOnClickListener(new View.
10     OnClickListener() {
11         public void onClick(View v) {
12             chrono.stop();
13             chrono.setTextColor(Color.BLUE);
14         }
15 });
16
17 calView.setOnDateChangeListener(new
18     CalendarView.OnDateChangeListener() {
19         @Override
20         public void onSelectedDayChange(CalendarView view, int year, int
21             month, int dayOfMonth) {
22             selectYear=year;
23             selectMonth=month+1;
24             selectDay=dayOfMonth;
25         }
26     });
```



Page • 11

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [14/15]

### 실습 6-1 날짜/시간 예약 앱 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ <예약 완료>를 클릭 시 캘린더뷰에서 설정한 연, 월, 일과 타임피커에서 설정한 시, 분이 맨아래 텍스트뷰에 채워지도록 함

예제 6-9 Java 코드 5

```
1 tvYear.setText(Integer.toString(selectYear));
2 tvMonth.setText(Integer.toString(selectMonth));
3 tvDay.setText(Integer.toString(selectDay));
4
5 tvHour.setText(Integer.toString(tpicker.getCurrentHour()));
6 tvMinute.setText(Integer.toString(tpicker.getCurrentMinute()));
```

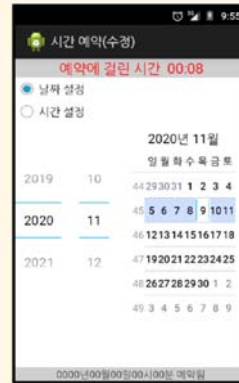
Page • 12

## 1. 고급 위젯 ▶ 날짜/시간 관련 위젯 [15/15]

### ▶ 작업 들어보기 6-1

[실습 6-1]을 다음과 같이 수정하라.

- 캘린더뷰 대신에 데이트피커를 사용하여 날짜를 설정한다.
- <예약 시작>과 <예약완료>를 없앤다. 대신 <예약 시작> 기능은 크로노미터를 클릭하면 동작하게 하고, <예약완료> 기능은 아래쪽 연도를 뿔클릭하면 동작하게 한다.
- 크로노미터를 클릭하기 전에는 라디오버튼, 데이트피커, 타임피커가 안 보이도록 설정하고, 크로노미터를 클릭하면 라디오버튼이 나타나게 한다. 그리고 아래쪽 연도를 뿔클릭하면 라디오버튼, 데이트피커, 타임피커가 다시 사라지게 한다.



Page • 13

### Coding Exercise 실습 6-1 XML Code

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:baselineAligned="false"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <Chronometer
            android:id="@+id/chronometer1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:format="예약에 걸린 시간 %s"
            android:gravity="center"
            android:textSize="20dp" />
    </LinearLayout>
</LinearLayout>
```

Page • 14

## Coding Exercise

```
<Button
    android:id="@+id/btnStart"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="예약 시작" />
</LinearLayout>
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/rdoCal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="날짜 설정 (캘린더뷰)" />

    <RadioButton
        android:id="@+id/rdoTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="시간 설정" />

</RadioGroup>
```

Page • 15

## Coding Exercise

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dip"
    android:layout_weight="1" >

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center" >
        <CalendarView
            android:id="@+id/calendarView1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:showWeekNumber="false" />
        <TimePicker
            android:timePickerMode="spinner"
            android:id="@+id/timePicker1"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
        </FrameLayout>
    </LinearLayout>
```

Page • 16



## Coding Exercise

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#CCCCCC" >

    <Button
        android:id="@+id/btnEnd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="예약완료" />

    <TextView
        android:id="@+id/tvYear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0000" />
```

Page • 17

## Coding Exercise

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="년" />

    <TextView
        android:id="@+id/tvMonth"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="00" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="월" />

    <TextView
        android:id="@+id/tvDay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="00" />
```

Page • 18

## Coding Exercise

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="일" />

<TextView
    android:id="@+id/tvHour"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="00" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="시" />
```

Page • 19

## Coding Exercise

```
<TextView
    android:id="@+id/tvMinute"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="00" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="분 예약됨" />
</LinearLayout>

</LinearLayout>
```

Page • 20

## Coding Exercise 실습 6-1 Java Code

```
package com.cookandroid.project6_1;

import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.graphics.Color;
import android.os.Bundle;
import android.os.SystemClock;
import android.view.View;
import android.widget.Button;
import android.widget.CalendarView;
import android.widget.Chronometer;
import android.widget.RadioButton;
import android.widget.TextView;
import android.widget.TimePicker;

import java.util.Calendar;
```

Page • 21

## Coding Exercise

```
public class MainActivity extends AppCompatActivity {

    Chronometer chrono;
    Button btnStart, btnEnd;
    RadioButton rdoCal, rdoTime;
    CalendarView calView;
    TimePicker tPicker;
    TextView tvYear, tvMonth, tvDay, tvHour, tvMinute;
    int selectYear, selectMonth, selectDay;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        setTitle("시간 예약");
```

Page • 22

## Coding Exercise

```
// 버튼
btnStart = (Button) findViewById(R.id.btnStart);
btnEnd = (Button) findViewById(R.id.btnEnd);

// 크로노미터
chrono = (Chronometer) findViewById(R.id.chronometer1);

// 라디오버튼 2개
rdoCal = (RadioButton) findViewById(R.id.rdoCal);
rdoTime = (RadioButton) findViewById(R.id.rdoTime);

// FrameLayout의 2개 위젯
tPicker = (TimePicker) findViewById(R.id.timePicker1);
calView = (CalendarView) findViewById(R.id.calendarView1);
```

Page • 23

## Coding Exercise

```
// 텍스트뷰 중에서 연,월,일,시,분 숫자
tvYear = (TextView) findViewById(R.id.tvYear);
tvMonth = (TextView) findViewById(R.id.tvMonth);
tvDay = (TextView) findViewById(R.id.tvDay);
tvHour = (TextView) findViewById(R.id.tvHour);
tvMinute = (TextView) findViewById(R.id.tvMinute);

// 처음에는 2개를 안보이게 설정
tPicker.setVisibility(View.INVISIBLE);
calView.setVisibility(View.INVISIBLE);

rdoCal.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        tPicker.setVisibility(View.INVISIBLE);
        calView.setVisibility(View.VISIBLE);
    }
});
```

Page • 24

## Coding Exercise

```
rdoTime.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        tPicker.setVisibility(View.VISIBLE);
        calView.setVisibility(View.INVISIBLE);
    }
});

// 타이머 설정
btnStart.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        chrono.setBase(SystemClock.elapsedRealtime());
        chrono.start();
        chrono.setTextColor(Color.RED);
    }
});
```

Page • 25

## Coding Exercise

```
// 버튼을 클릭하면 날짜,시간을 가져온다.
btnEnd.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        chrono.stop();
        chrono.setTextColor(Color.BLUE);

        tvYear.setText(Integer.toString(selectYear));
        tvMonth.setText(Integer.toString(selectMonth));
        tvDay.setText(Integer.toString(selectDay));
        tvHour.setText(Integer.toString(tPicker.getCurrentHour()));
        tvMinute.setText(Integer.toString(tPicker.getCurrentMinute()));
    }
});
```

Page • 26

## Coding Exercise

```
calView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {  
    @Override  
    public void onSelectedDayChange(@NonNull CalendarView view,  
                                    int year, int month, int dayOfMonth) {  
        selectYear = year;  
        selectMonth = month + 1;  
        selectDay = dayOfMonth;  
    }  
});  
  
}
```

Page • 27

## 1. 고급 위젯 ▶ 기타 위젯[1/5]

### ❖ 자동완성텍스트뷰와 멀티자동완성텍스트뷰

- ✓ 사용자가 단어의 일부만 입력해도 자동완성됨
- ✓ 자동완성텍스트뷰는 1개, 멀티자동완성텍스트뷰는 쉼표로 구분하여 자동완성

```
java.lang.Object  
└─ android.view.View  
    └─ android.widget.TextView  
        └─ android.widget.EditText  
            └─ android.widget.EditText.AutoCompleteTextView  
                └─ android.widget.EditText.MultiAutoCompleteTextView
```

자동완성텍스트뷰 계층도

Page • 28

## 1. 고급 위젯 ▶ 기타 위젯[2/5]

### ❖ 자동완성텍스트뷰와 멀티자동완성텍스트뷰 예제

예제 6-10 자동완성텍스트뷰의 XML 코드

```
1 <LinearLayout>
2     <AutoCompleteTextView
3         android:id="@+id/autoCompleteTextView1"
4         android:completionHint="선택하세요"
5         android:completionThreshold="2"
6         android:hint="자동완성텍스트뷰" >
7     </AutoCompleteTextView>
8     <MultiAutoCompleteTextView
9         android:id="@+id/multiAutoCompleteTextView1"
10        android:completionHint="선택하세요"
11        android:completionThreshold="2"
12        android:hint="멀티자동완성텍스트뷰" />
13 </LinearLayout>
```



Page • 29

### Coding Exercise

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <AutoCompleteTextView
        android:id="@+id/autoCompleteTextView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:completionHint="선택하세요"
        android:completionThreshold="2"
        android:hint="자동완성텍스트뷰" >
    </AutoCompleteTextView>
    <MultiAutoCompleteTextView
        android:id="@+id/multiAutoCompleteTextView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:completionHint="선택하세요"
        android:completionThreshold="2"
        android:hint="멀티자동완성텍스트뷰" />

</LinearLayout>
```

Page • 30

## 1. 고급 위젯 ▶ 기타 위젯[3/5]

### ❖ 자동완성텍스트뷰와 멀티자동완성텍스트뷰 Java 코드

예제 6-11 자동완성텍스트뷰의 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4     String[] items = { "CSI-뉴욕", "CSI-라스베가스", "CSI-마이애미", "Friends",
5                       "Fringe", "Lost" };
6
7     AutoCompleteTextView auto = (AutoCompleteTextView) findViewById
8         (R.id.autoCompleteTextView1);
9     ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
10         android.R.layout.simple_dropdown_item_1line, items);
11     auto.setAdapter(adapter);
12
13     MultiAutoCompleteTextView multi = (MultiAutoCompleteTextView)
14         findViewById(R.id.multiAutoCompleteTextView1);
15     CommaTokenizer token = new MultiAutoCompleteTextView.CommaTokenizer();
16     multi.setTokenizer(token);
17     multi.setAdapter(adapter);
18 }
```



Page • 31

### Coding Exercise 예제 6-11

```
package com.cookandroid.project6;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.MultiAutoCompleteTextView;
import android.widget.MultiAutoCompleteTextView.CommaTokenizer;

public class MainActivity extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Page • 32



## Coding Exercise

```
String[] items = { "CSI-뉴욕", "CSI-라스베이가스", "CSI-마이애미", "Friends",  
                  "Fringe", "Lost" };  
  
AutoCompleteTextView auto =  
    (AutoCompleteTextView) findViewById(R.id.autoCompleteTextView1);  
ArrayAdapter<String> adapter =  
    new ArrayAdapter<String>(this,  
        android.R.layout.simple_dropdown_item_1line, items);  
auto.setAdapter(adapter);  
  
MultiAutoCompleteTextView multi = (MultiAutoCompleteTextView)  
findViewById(R.id.multiAutoCompleteTextView1);  
CommaTokenizer token = new MultiAutoCompleteTextView.CommaTokenizer();  
multi.setTokenizer(token);  
multi.setAdapter(adapter);  
}  
}
```

Page • 33

## 1. 고급 위젯 ▶ 기타 위젯[4/5]

### ❖ 프로그레스바, 시크바, 레이팅바

```
java.lang.Object  
└ android.view.View  
    └ android.widget.ProgressBar  
        └ android.widget.AbsSeekBar  
            └ android.widget.RatingBar  
                └ android.widget.SeekBar
```

프로그레스바, 시크바, 레이팅바 계층도

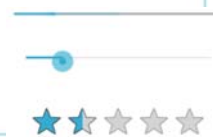


그림 6-2 프로그레스바, 시크바, 레이팅바

- ✓ 프로그레스바(ProgressBar) : 작업의 진행 상황을 바(Bar)나 원 형태로 제공
- ✓ 시크바(SeekBar) : 프로그레스바와 대부분 비슷, 사용자 터치로 임의 조절 가능
- ✓ 레이팅바(RatingBar) : 진행 상황을 별 모양으로 표시

Page • 34

## 1. 고급 위젯 ▶ 기타 위젯[5/5]

### ❖ 프로그레스바, 시크바, 레이팅바 예제

예제 6-12 프로그레스바, 시크바, 레이팅바의 XML 코드

```
1 <LinearLayout>
2     <ProgressBar
3         style="?android:attr/progressBarStyleHorizontal"
4         android:max="100"
5         android:progress="20"
6         android:secondaryProgress="50" />
7     <SeekBar
8         android:progress="20" />
9     <RatingBar
10        android:numStars="5"
11        android:rating="1.5"
12        android:stepSize="0.5" />
13 </LinearLayout>
```

Page • 35

### Coding Exercise 예제 6-12

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <ProgressBar
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:max="100"
        android:progress="20"
        android:secondaryProgress="50" />
```

Page • 36

## Coding Exercise

```
<SeekBar
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:progress="20" />

<RatingBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:numStars="5"
    android:rating="1.5"
    android:stepSize="0.5" />

</LinearLayout>
```

Page • 37

## 2. 뷰 컨테이너 ▶ 간단한 기능의 뷰 컨테이너[1/4]

### ❖ 스크롤뷰(ScrollView)

- ✓ 스크롤뷰는 수직(위아래)으로 스크롤하는 기능
- ✓ 수평(좌우)으로 스크롤하는 수평스크롤뷰(HorizontalScrollView)는 따로 있음
- ✓ 주의할 점은 스크롤뷰에는 단 하나의 위젯만 넣을 수 있음

```
java.lang.Object
└─ android.view.View
    └─ android.widget.ViewGroup
        └─ android.widget.FrameLayout
            └─ android.widget.ScrollView
```

스크롤뷰 계층도

Page • 38

## 2. 뷰 컨테이너 ▶ 간단한 기능의 뷰 컨테이너[2/4]

### ❖ 스크롤뷰 예제

예제 6-13 스크롤뷰의 XML 코드

```
1
2 <ScrollView xmlns:android="http://www."
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical" >
6
7   <LinearLayout
8     android:layout_width="fill_parent"
9     android:layout_height="fill_parent"
10    android:orientation="vertical" >
11
12     <Button
13       android:layout_width="match_parent"
14       android:layout_height="100dp"
15       android:text="버튼 1" />
16     ~~~~~ 중간 생략(버튼 7개) ~~~~~
17   </LinearLayout>
18
19 </ScrollView>
```



Page • 39

### Coding Exercise 예제 6-13

```
<ScrollView xmlns:android="~~~"
  xmlns:tools="~~~"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >

  <LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
      android:layout_width="match_parent"
      android:layout_height="100dp"
      android:text="버튼 1" />

    <Button
      android:layout_width="match_parent"
      android:layout_height="100dp"
      android:text="버튼 2" />
```

Page • 40

## Coding Exercise

```
<Button
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:text="버튼 3" />

<Button
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:text="버튼 4" />

<Button
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:text="버튼 5" />

<Button
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:text="버튼 6" />
```

Page • 41

## Coding Exercise

```
<Button
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:text="버튼 7" />

<Button
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:text="버튼 8" />

</LinearLayout>

</ScrollView>
```

Page • 42

## 2. 뷰 컨테이너 ▶ 간단한 기능의 뷰 컨테이너[3/4]

### ❖ 슬라이딩드로어(SlidingDrawer)

- ✓ 슬라이딩드로어(SlidingDrawer)의 사전적 의미는 서랍을 뜻함
- ✓ 위젯들을 서랍처럼 열어서 보여주거나 닫아서 감춤

```
java.lang.Object
└─ android.view.View
    └─ android.widget.ViewGroup
        └─ android.widget.SlidingDrawer
```

슬라이딩드로어 계층도

- ✓ 슬라이딩드로어의 일반적 형태

```
<슬라이딩드로어 handle="핸들명" content="콘텐츠명" >
  <버튼 id="핸들명" /> // 서랍 손잡이 역할
  <리니어레이아웃 id="콘텐츠명">

    // 이곳이 서랍 내부이며 필요한 위젯을 넣으면 됨

  </리니어레이아웃>
</슬라이딩드로어>
```

Page • 43

## 2. 뷰 컨테이너 ▶ 간단한 기능의 뷰 컨테이너[4/4]

### ❖ 슬라이딩드로어 예제

예제 6-14 슬라이딩드로어의 XML 코드

```
1 <LinearLayout>
2   <TextView
3     android:text="여기는 서랍 밖입니다." />
4   <SlidingDrawer
5     android:content="@id/content"
6     android:handle="@id/handle" >
7
8     <Button
9       android:id="@id/handle"
10      android:text="서랍 손잡이" />
11
12   <LinearLayout
13     android:id="@id/content"
14     android:background="#00FF00"
15     android:gravity="center" >
16
17     <TextView
18       android:text="여기는 서랍 안입니다." />
19   </LinearLayout>
20
21 </SlidingDrawer>
22 </LinearLayout>
```



Page • 44

## Coding Exercise

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="여기는 서랍 밖입니다."
        android:textSize="20dp" />

    <SlidingDrawer
        android:id="@+id/slidingDrawer1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:content="@+id/content"
        android:handle="@+id/handle" >
```

Page • 45

## Coding Exercise

```
        <Button
            android:id="@+id/handle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="서랍 손잡이" />
        <LinearLayout
            android:id="@+id/content"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#00ff00"
            android:gravity="center" >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="여기는 서랍 안입니다."
                android:textSize="20dp" />
        </LinearLayout>
    </SlidingDrawer>

</LinearLayout>
```

Page • 46

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[1/28]

### ❖ 뷰플리퍼(ViewFlipper)

- ✓ 안에 여러 개의 위젯을 배치한 후, 필요에 따라서 화면을 왼쪽과 오른쪽으로 밀어서 하나의 위젯씩 화면에 보여주는 방식의 뷰 컨테이너

```
java.lang.Object
└─ android.view.View
    └─ android.widget.ViewGroup
        └─ android.widget.FrameLayout
            └─ android.widget.ViewAnimator
                └─ android.widget.ViewFlipper
```

뷰플리퍼 계층도

- ✓ 뷰플리퍼의 일반적 형태

```
<리니애레이아웃>
<리니애레이아웃>
    // 왼쪽/오른쪽으로 전환할 버튼 또는 이미지뷰
</리니애레이아웃>
<뷰플리퍼>

    // 이곳에 한 번에 하나씩 보여줄 위젯들 넣음

</뷰플리퍼>
</리니애레이아웃>
```

Page • 47

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[2/28]

### ❖ 뷰플리퍼 예제

예제 6-15 뷰플리퍼의 XML 코드

```
1 <LinearLayout>
2   <LinearLayout
3     android:orientation="horizontal" >
4     <Button
5       android:id="@+id/btnPrev"
6       android:text="이전화면" />
7     <Button
8       android:id="@+id/btnNext"
9       android:text="다음화면" />
10  </LinearLayout>
11  <ViewFlipper
12    android:id="@+id/viewflipper1">
13    <LinearLayout
14      android:background="@color/white" >
15      ----- 이곳에 필요한 위젯 삽입 -----
16    </LinearLayout>
17    <LinearLayout
18      android:background="@color/black" >
19      ----- 이곳에 필요한 위젯 삽입 -----
20    </LinearLayout>
21    <LinearLayout
22      android:background="@color/red" >
23      ----- 이곳에 필요한 위젯 삽입 -----
24    </LinearLayout>
25  </ViewFlipper>
26 </LinearLayout>
```



Page • 48



## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[3/28]

### ❖ 뷰플리퍼 Java 코드

예제 6-16 뷰플리퍼의 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4
5     Button btnPrev, btnNext;
6     final ViewFlipper vFlipper;
7
8     btnPrev = (Button) findViewById(R.id.btnPrev);
9     btnNext = (Button) findViewById(R.id.btnNext);
10    vFlipper = (ViewFlipper) findViewById(R.id.viewFlipper1);
11
12    btnPrev.setOnClickListener(new View.OnClickListener() {
13        public void onClick(View v) {
14            vFlipper.showPrevious();
15        }
16    });
17    ----- 중간 생략(다음화면 버튼 1개 코딩) -----
18 }
```

Page • 49

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[4/28]

### ▶ 직접 풀어보기 6-2

뷰플리퍼를 이용하여 자동 사진 보기 앱을 작성하라.

- 적절한 이미지 여러 장이 자동으로 넘어가는 앱을 만든다.
- <사진보기 시작>과 <사진보기 중지>를 만들고, <사진보기 시작>을 클릭하면 1초 단위로 화면이 자동으로 넘어간다.
- 뷰플리퍼 안에 리니어레이아웃을 배치할 필요는 없고 직접 이미지 뷰가 나오면 된다.

■ 화면 넘김 시작 메소드로 startFlipping() 중지 메소드로 stopFlipping() 화면 넘김 간격 메소드로 setFlipInterval(밀리초)를 사용한다.



Page • 50

## Coding Exercise 직접 풀어보기 6-2 XML Code

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:id="@+id/btnStart"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text=" 사진보기 시작 " />
        <Button
            android:id="@+id/btnStop"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text=" 사진보기 정지 " />
    </LinearLayout>
</LinearLayout>
```

Page • 51

## Coding Exercise

```
<ViewFlipper
    android:id="@+id/viewFlipper1"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:src="@drawable/cupcake" />
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:src="@drawable/donut" />
    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:src="@drawable/eclair" />
```

Page • 52

## Coding Exercise

```
<ImageView
    android:id="@+id/imageView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/froyo" />

<ImageView
    android:id="@+id/imageView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/gingerbread" />

<ImageView
    android:id="@+id/imageView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/honeycomb" />

<ImageView
    android:id="@+id/imageView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/icecream" />
```

Page • 53

## Coding Exercise

```
<ImageView
    android:id="@+id/imageView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/jellybean" />

<ImageView
    android:id="@+id/imageView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/kitkat" />

<ImageView
    android:id="@+id/imageView10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/lollipop" />

<ImageView
    android:id="@+id/imageView11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/marshmallow" />
```

Page • 54

## Coding Exercise

```
<ImageView
    android:id="@+id/imageView12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/nougat" />

<ImageView
    android:id="@+id/imageView13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/oreo" />

<ImageView
    android:id="@+id/imageView14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/pie" />

</ViewFlipper>

</LinearLayout>
```

Page • 55

## Coding Exercise 직접 풀어보기 6-2 Java Code

```
package com.cookandroid.self6_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ViewFlipper;

public class MainActivity extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnStart, btnStop;
        final ViewFlipper vFlipper;

        btnStart = (Button) findViewById(R.id.btnStart);
        btnStop = (Button) findViewById(R.id.btnStop);
```

Page • 56

## Coding Exercise

```
vFlipper = (ViewFlipper) findViewById(R.id.viewFlipper1);

vFlipper.setFlipInterval(1000);

btnStart.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        vFlipper.startFlipping();
    }
});

btnStop.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        vFlipper.stopFlipping();
    }
});
}
```

Page • 57

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[5/28]

### ❖ 탭호스트(TabHost)

- ✓ 여러 탭을 두고 각 탭을 클릭할 때마다 해당 화면이 나오도록 설정하는 뷰 컨테이너

```
java.lang.Object
└ android.view.View
    └ android.widget.ViewGroup
        └ android.widget.FrameLayout
            └ android.widget.TabHost
```

탭호스트 개충도

- ✓ 탭 호스트의 구성

```
<탭호스트 id="@android:id/tabhost">
  <리니어레이아웃>
    <탭위젯 id="@android:id/tabs" />
    <프레임레이아웃 id="@android:id/tabcontent">

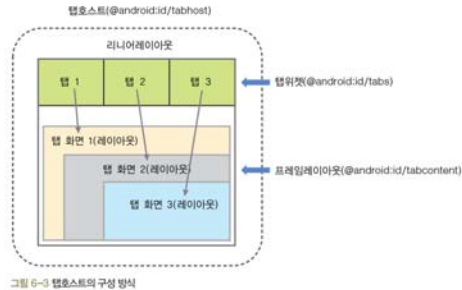
        // 여기에 각 탭스펙에 대응할 탭 화면(레이아웃)을 3개 넣음

    </프레임레이아웃>
  </리니어레이아웃>
</탭호스트>
```

Page • 58

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[6/28]

### ❖ 탭호스트



✓ 탭을 생성하고 탭화면을 연결하기 위한 Java 코드 형식

```
TabHost tabHost = getTabHost(); // 탭호스트 변수 생성
// 탭스펙 생성
TabSpec tabSpec1 = tabHost.newTabSpec("TAG1").setIndicator("탭에 출력될 글자");
tabSpec1.setContent(R.id.tab1); // 탭스펙을 탭과 연결
tabHost.addTab(tabSpec1); // 탭을 탭호스트에 부착
```

**탭스펙(TabSpec) : 탭을 구성하는 요소들의 집합**

Page • 59

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[7/28]

### ❖ 탭호스트 예제

예제 6-17 탭호스트의 XML 코드

```
1 <TabHost xmlns:android="http://schemas.android.com/apk/res/android"
2   android:id="@android:id/tabhost">
3   <LinearLayout>
4     <TabWidget
5       android:id="@android:id/tabs">
6     </TabWidget>
7     <FrameLayout
8       android:id="@android:id/tabcontent">
9       <LinearLayout
10        android:id="@id/tabSong"
11        android:background="#f00000">
12        <LinearLayout>
13        <LinearLayout
14          android:id="@id/tabArtist"
15          android:background="#f0f000">
16        </LinearLayout>
17        <LinearLayout
18          android:id="@id/tabAlbum"
19          android:background="#f000ff">
20        </LinearLayout>
21      </FrameLayout>
22    </LinearLayout>
23  </TabHost>
```



Page • 60

## Coding Exercise

```
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
        <TabWidget
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
        </TabWidget>
        <FrameLayout
            android:id="@+id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >
```

Page • 61

## Coding Exercise

```
        <LinearLayout
            android:id="@+id/tabSong"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#f00000"
            android:orientation="vertical">
        </LinearLayout>
        <LinearLayout
            android:id="@+id/tabArtist"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#f0f000"
            android:orientation="vertical">
        </LinearLayout>
        <LinearLayout
            android:id="@+id/tabAlbum"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#f000ff"
            android:orientation="vertical">
        </LinearLayout>
    </FrameLayout>
</LinearLayout>
</TabHost>
```

Page • 62

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[8/28]

### ❖ 탭호스트 Java 코드

예제 6-18 탭호스트의 Java 코드

```
1  ----- 중간 생략(import문) -----
2  @SuppressWarnings("deprecation")
3  public class MainActivity extends TabActivity {
4
5      @Override
6      public void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_main);
9
10         TabHost tabHost = getTabHost();
11
12         TabSpec tabSpecSong = tabHost.newTabSpec("SONG").setIndicator("음악별");
13         tabSpecSong.setContent(R.id.tabSong);
14         tabHost.addTab(tabSpecSong);
15
16         TabSpec tabSpecArtist = tabHost.newTabSpec("ARTIST").setIndicator("가수별");
17         tabSpecArtist.setContent(R.id.tabArtist);
18         tabHost.addTab(tabSpecArtist);
19
20         TabSpec tabSpecAlbum = tabHost.newTabSpec("ALBUM").setIndicator("앨범별");
21         tabSpecAlbum.setContent(R.id.tabAlbum);
22         tabHost.addTab(tabSpecAlbum);
23
24         tabHost.setCurrentTab(0);
25     }
26 }
```

Page • 63

### Coding Exercise 예제 6-18

```
package com.cookandroid.project6;

import android.app.TabActivity;
import android.os.Bundle;
import android.widget.TabHost;
import android.widget.TabHost.TabSpec;

@SuppressWarnings("deprecation")
public class MainActivity extends TabActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TabHost tabHost = getTabHost();

        TabSpec tabSpecSong = tabHost.newTabSpec("SONG").setIndicator("음악별");
        tabSpecSong.setContent(R.id.tabSong);
        tabHost.addTab(tabSpecSong);
```

Page • 64



## Coding Exercise

```
TabSpec tabSpecArtist = tabHost.newTabSpec("ARTIST")
    .setIndicator("가수별");
tabSpecArtist.setContent(R.id.tabArtist);
tabHost.addTab(tabSpecArtist);

TabSpec tabSpecAlbum =
    tabHost.newTabSpec("ALBUM").setIndicator("앨범별");
tabSpecAlbum.setContent(R.id.tabAlbum);
tabHost.addTab(tabSpecAlbum);

tabHost.setCurrentTab(0);
    }
}
```

Page • 65

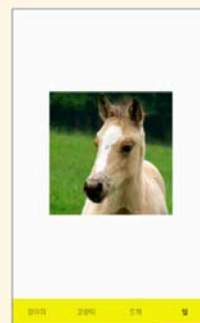
## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[9/28]

### ▶ 직접 풀어보기 6-3

탭호스트를 이용하여 동물 선택 앱을 작성하라.

- 탭위젯을 아래쪽에 배치하고 탭은 4개가 나오게 한다.
- 프레임레이아웃 안의 리니어레이아웃 3개를 제거하고 4개의 이미지 뷰로 배치한다.

 프레임레이아웃의 layout\_weight 속성을 1로 한다.



Page • 66

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[10/28]

### ❖ 액션바(Action Bar)

- ✓ 액션바(Action Bar)는 허니콤(Android 3.0, API 11)에서 태블릿과 같은 대형 화면에서 여러 화면을 사용하기 위해서 고안
- ✓ 태블릿, 스마트폰 등 다양한 크기의 화면을 디자인하는 데 활용

### ❖ 프래그먼트(Fragment)

- ✓ 액티비티보다 작은 단위의 화면
- ✓ 프래그먼트를 사용하면 대형 화면에서 액티비티 화면을 분할해서 표현 가능
- ✓ 스마트폰과 같은 소형 화면에서는 화면의 분할보다는 실행 중에 화면을 동적으로 추가, 제거하는 데 더 많이 활용

Page • 67

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[11/28]

### ❖ 액션바와 프래그먼트를 활용하여 화면 구현

- ✓ 액션바를 생성한 후, 탭(Tab) 위젯을 액션바에 등록



그림 6-4 액션바와 프래그먼트의 구성 방식

Page • 68

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[12/28]

### ❖ 액션바와 프래그먼트를 활용하여 화면 구현

#### ✓ 액션바에 탭을 추가하는 Java 코드 형식

```
ActionBar bar = getActionBar(); // 상단에 표시할 액션바 준비
// 탭호스트와 같이 탭의 모양을 가지도록 설정
bar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
tabSong = bar.newTab(); // 탭을 액션바에 생성
tabSong.setText("음악별"); // 탭의 글자를 설정
tabSong.setTabListener(this); // 탭을 터치하면 작동하는 리스너를 지정
bar.addTab(tabSong); // 탭을 액션바에 추가
```

Page • 69

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[13/28]

### ❖ 액션바와 프래그먼트를 활용하여 화면 구현

#### ✓ 액션바 Java 코드

예제 6-19 액션바의 Java 코드 1

```
1  ----- 중간 생략(import문) -----
2  public class MainActivity extends AppCompatActivity implements ActionBar.
   TabListener {
3      ActionBar.Tab tabSong, tabArtist, tabAlbum;
4
5      protected void onCreate(Bundle savedInstanceState) {
6          super.onCreate(savedInstanceState);
7          ActionBar bar = getSupportActionBar();
8          bar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
9
10         tabSong = bar.newTab();
11         tabSong.setText("음악별");
12         tabSong.setTabListener(this);
13         bar.addTab(tabSong);
14         ----- 중간 생략(Artist 탭, Album 탭) -----
15     }
16     public void onTabSelected(ActionBar.Tab tab, FragmentTransaction ft) {
17     }
18     public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction ft) {
19     }
20     public void onTabReselected(ActionBar.Tab tab, FragmentTransaction ft) {
21     }
22
23 }
```

Page • 70

## Coding Exercise 예제 6-19

```
package com.cookandroid.project6;

import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

@SuppressWarnings("deprecation")
public class MainActivity extends AppCompatActivity implements ActionBar.TabListener{

    ActionBar.Tab tabSong, tabArtist, tabAlbum;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActionBar bar = getSupportActionBar();
        bar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        tabSong = bar.newTab();
        tabSong.setText("음악별");
        tabSong.setTabListener(this);
        bar.addTab(tabSong);
```

Page • 71

## Coding Exercise

```
        tabArtist = bar.newTab();
        tabArtist.setText("가수별");
        tabArtist.setTabListener(this);
        bar.addTab(tabArtist);

        tabAlbum = bar.newTab();
        tabAlbum.setText("앨범별");
        tabAlbum.setTabListener(this);
        bar.addTab(tabAlbum);
    }

    @Override
    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction ft) {
    }

    @Override
    public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction ft) {
    }

    @Override
    public void onTabReselected(ActionBar.Tab tab, FragmentTransaction ft) {
    }
}
```

Page • 72

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[14/28]

### ❖ 액션바와 프래그먼트를 활용하여 화면 구현

- ✓ Fragment를 상속받는 MyTabFragment 클래스를 내부 클래스로 생성

예제 6-20 액션바의 Java 코드 2

```
1 public static class MyTabFragment extends android.support.v4.app.Fragment {
2     String tabName;
3     public void onCreate(Bundle savedInstanceState) {
4         super.onCreate(savedInstanceState);
5         Bundle data = getArguments();
6         tabName = data.getString("tabName");
7     }
8     public View onCreateView(LayoutInflater inflater, ViewGroup container,
9         Bundle savedInstanceState) {
10         LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
11             LinearLayout.LayoutParams.MATCH_PARENT,
12             LinearLayout.LayoutParams.MATCH_PARENT);
13         LinearLayout baseLayout = new LinearLayout(super.getActivity());
14         baseLayout.setOrientation(LinearLayout.VERTICAL);
15         baseLayout.setLayoutParams(params);
16         if (tabName == "음악방") baseLayout.setBackgroundColor(Color.RED);
17         if (tabName == "가수방") baseLayout.setBackgroundColor(Color.GREEN);
18         if (tabName == "앨범방") baseLayout.setBackgroundColor(Color.BLUE);
19
20         return baseLayout;
21     }
22 }
```

Page • 73

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[15/28]

### ❖ 액션바와 프래그먼트를 활용하여 화면 구현

- ✓ 멤버변수로 프래그먼트 배열 변수를 추가, onTabSelected( ) 메소드 코딩

예제 6-21 액션바의 Java 코드 3

```
1 MyTabFragment myFrgs[] = new MyTabFragment[3];
2
3 public void onTabSelected(Tab tab,
4     FragmentTransaction ft) {
5     MyTabFragment myTabFrag = null;
6
7     if (myFrgs[tab.getPosition()] == null) {
8         myTabFrag = new MyTabFragment();
9         Bundle data = new Bundle();
10        data.putString("tabName", tab.getText().toString());
11        myTabFrag.setArguments(data);
12        myFrgs[tab.getPosition()] = myTabFrag;
13    }
14    else
15        myTabFrag = myFrgs[tab.getPosition()];
16    ft.replace(android.R.id.content, myTabFrag);
17 }
```



Page • 74

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[16/28]

### ❖ 웹뷰(WebView)

- ✓ 사용자가 웹브라우저 기능을 앱 안에 직접 포함시킬 수 있는 위젯

```
java.lang.Object
└─ android.view.View
    └─ android.widget.ViewGroup
        └─ android.widget.AbsoluteLayout
            └─ android.webkit.WebView
```

웹뷰 계층도

Page • 75

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[17/28]

### 실습 6-2 간단 웹브라우저 만들기

- ❖ 안드로이드 프로젝트 생성
  - ✓ 프로젝트 이름 : Project6\_2
  - ✓ 패키지 이름 : com.cookandroid.project6\_2
- ❖ 화면 디자인 및 편집
  - ✓ 리니어레이아웃을 하나 더 만들고,
  - ✓ 그 안에 에디트텍스트 1개와 버튼 2개로 구성  
하단에는 웹뷰를 만들
  - ✓ 각 위젯의 id는 edtUrl, btnGo, btnBack,  
webView1로 함



그림 6-5 간단 웹브라우저 결과 화면

Page • 76

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[18/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ 화면 디자인 및 편집

예제 6-22 activity\_main.xml

```

1 <LinearLayout>
2   <LinearLayout>
3     <EditText
4       android:id="@+id/edtUrl"
5       android:layout_weight="1"
6       android:singleLine="true" >
7   </EditText>
8   <Button
9     android:id="@+id/btnGo"
10    android:text="이동" />
11  <Button
12    android:id="@+id/btnBack"
13    android:text="이전" />
14 </LinearLayout>
15 <WebView
16   android:id="@+id/webView1" />
17 </LinearLayout>

```



Page • 77

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[19/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ 화면 디자인 및 편집

- ✓ AndroidManifest.xml(매니페스트) : 프로젝트의 전반적인 환경을 설정하는 파일
- ✓ 위치: Project....\app\src\main\AndroidManifest.xml

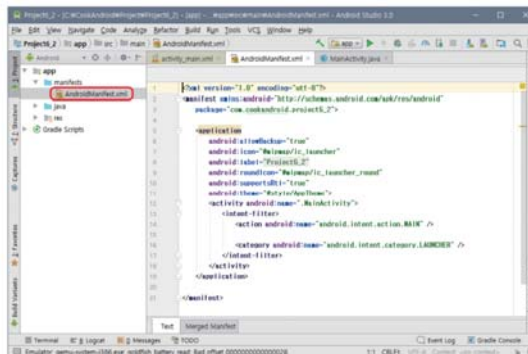


그림 6-6 AndroidManifest.xml 파일 열기

Page • 78

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[20/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ 화면 디자인 및 편집

##### ✓ 파일 수정

예제 6-23 AndroidManifest.xml

```

1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2     package="com.cookandroid.project6_2">
3
4     <application
5         android:allowBackup="true"
6         android:icon="@drawable/emo_im_cool"
7         android:label="@string/app_label"
8         android:logo="@drawable/web"
9         android:theme="@style/AppTheme" >
10         <activity
11             android:name=".MainActivity"
12             android:label="@string/app_label" >
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19     </application>
20
21 </manifest>

```

Page • 79

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[21/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ 실행

##### ✓ 화면 상단에 아이콘과 타이틀이 변경된 것 확인

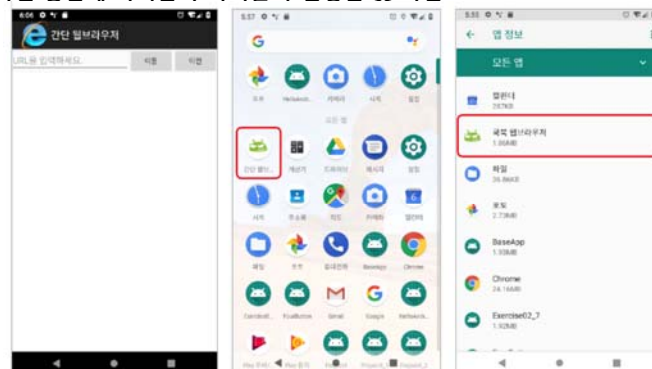


그림 6-7 매니페스트에서 설정한 내용 확인

Page • 80



## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[22/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ activity\_main.xml의 4개 위젯에 대응할 위젯 변수 4개를 전역변수로 선언
- ✓ onCreate( ) 메소드 안의 각 변수에 위젯을 대입

예제 6-24 Java 코드 1

```
1  --- 중간 생략(import문) ---
2  public class MainActivity extends AppCompatActivity {
3      EditText edtUrl;
4      Button btnGo, btnBack;
5      WebView web;
6
7
8      @Override
9      public void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         edtUrl = (EditText) findViewById(R.id.edtUrl);
14         btnGo = (Button) findViewById(R.id.btnGo);
15         btnBack = (Button) findViewById(R.id.btnBack);
16         web = (WebView) findViewById(R.id.webView1);
17     }
18
19 }
```

Page • 81

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[23/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ WebViewClient의 상속을 받는 웹뷰클라이언트 클래스를 정의

예제 6-25 Java 코드 2

```
1  class CookWebViewClient extends WebViewClient {
2
3  }
```

Page • 82

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[24/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ 메소드 중 `shouldOverrideUrlLoading()` 메소드를 선택하고 <OK>를 클릭

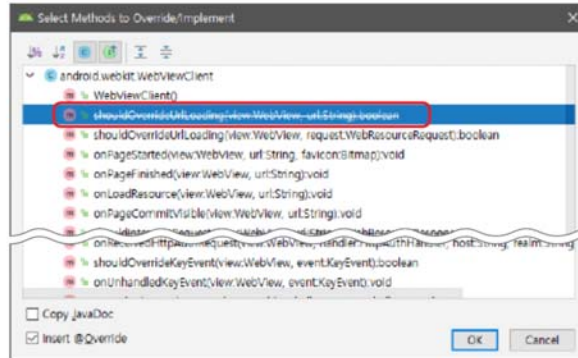


그림 6-8 메소드의 자동 오버라이딩

Page • 83

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[25/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ 자동으로 오버라이드된 메소드

#### 예제 6-26 Java 코드 3

```
1      ~~~~ 중간 생략 ~~~~
2      web = (WebView) findViewById(R.id.webView1);
3  }
4
5  class CookWebViewClient extends WebViewClient {
6
7      @Override
8      public boolean shouldOverrideUrlLoading(WebView view, String url) {
9
10         return super.shouldOverrideUrlLoading(view, url);
11     }
12
13 }
14 }
```

Page • 84

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[26/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ Java 코드 작성 및 수정

- ✓ 생성한 CookWebViewClient 클래스를 웹뷰에 설정
- ✓ 화면 확대/축소 아이콘이 보이도록 설정
- ✓ 두 버튼을 클릭하면 입력한 URL로 이동하거나 이전 화면으로 돌아가도록 함

예제 6-27 Java 코드 4

```

1  ----- 중간 생략 -----
2  web = (WebView) findViewById(R.id.webView1);
3
4  web.setWebViewClient(new CookWebViewClient());
5
6  WebSettings webSet = web.getSettings();
7  webSet.setBuiltInZoomControls(true);
8
9  btnGo.setOnClickListener(new View.OnClickListener() {
10     public void onClick(View v) {
11         web.loadUrl(editUrl.getText().toString());
12     }
13 });
14
15 btnBack.setOnClickListener(new View.OnClickListener() {
16     public void onClick(View v) {
17         web.goBack();
18     }
19 });
20 }
21 ----- 중간 생략 -----

```

Page • 85

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[27/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ 프로젝트 실행 및 결과 확인

- ✓ 에디트텍스트에 "http://m.daum.net"을 입력하고 <이동>을 클릭



그림 6-9 웹페이지가 열리지 않음

Page • 86

## 2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너[28/28]

### 실습 6-2 간단 웹브라우저 만들기

#### ❖ 다시 화면 디자인 및 편집

- ✓ AndroidManifest.xml을 열고 <application 행 위에 다음 한 행을 추가

#### 예제 6-28 AndroidManifest.xml

```
1  ~~~~ 중간 생략 ~~~~
2
3
4  <uses-permission android:name="android.permission.INTERNET" />
5
6  <application
7      android:icon="@drawable/emo_im_cool"
8  ~~~~ 중간 생략 ~~~~
```

Page • 87

### Coding Exercise 실습 6-2 XML Code

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <EditText
            android:id="@+id/edtUrl"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="URL을 입력하세요."
            android:singleLine="true" >

        </EditText>
```

Page • 88

## Coding Exercise

```
<Button
    android:id="@+id/btnGo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="이동" />

<Button
    android:id="@+id/btnBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="이전" />
</LinearLayout>

<WebView
    android:id="@+id/webView1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clickable="true" />

</LinearLayout>
```

Page • 89

## Coding Exercise 실습 6-2 Java Code

```
package com.cookandroid.project6_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebResourceRequest;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;

@SuppressWarnings("deprecation")
public class MainActivity extends AppCompatActivity {

    EditText edtUrl;
    Button btnGo, btnBack;
    WebView web;
```

Page • 90

## Coding Exercise

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setIcon(R.drawable.web);

    edtUrl = (EditText) findViewById(R.id.edtUrl);
    btnGo = (Button) findViewById(R.id.btnGo);
    btnBack = (Button) findViewById(R.id.btnBack);
    web = (WebView) findViewById(R.id.webView1);

    web.setWebViewClient(new CookWebViewClient());

    WebSettings webSet = web.getSettings();
    webSet.setBuiltInZoomControls(true);

    btnGo.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            web.loadUrl(edtUrl.getText().toString());
        }
    });
};
```

Page • 91

## Coding Exercise

```
btnBack.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        web.goBack();
    }
});

}

class CookWebViewClient extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        return super.shouldOverrideUrlLoading(view, url);
    }
}

}
```

Page • 92

## Coding Exercise AndroidManifest.xml 파일

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.cookandroid.project6_2">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/emo_im_cool"
        android:label="실습 웹브라우저"
        android:logo="@drawable/web"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <activity android:name=".MainActivity"
            android:label="간단 웹브라우저" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Page \* 93

## Question ?

IT CookBook, Android Studio를 활용한 안드로이드 프로그래밍