

Ch06-고급 위젯 익히기



2025-01-01(분반)

AI 오픈소스 전문프로젝트

Fri 09:00~ (S4-1-201)

학습목표

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

목차

01 고급 위젯

02 뷰 컨테이너

01 고급 위젯

1. 날짜/시간 관련 위젯

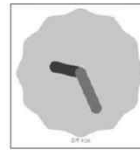
■ 아날로그 시계, 디지털 시계

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

아날로그시계, 디지털시계 계층도

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>
```



5 / 63

1. 날짜/시간 관련 위젯

■ 크로노미터(Chronometer)

- 타이머 형식의 위젯
- 일반적으로 시간을 측정할 때 많이 사용함

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

크로노미터 계층도

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

6 / 63

1. 날짜/시간 관련 위젯

■ 타임피커, 데이트피커, 캘린더뷰

- 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

7 / 63

1. 날짜/시간 관련 위젯

■ 타임피커, 데이트피커, 캘린더뷰

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>
```



8 / 63

1. 날짜/시간 관련 위젯

■ 실습 6-1 날짜/시간 예약 앱 만들기

- 타이머 기능: <예약 시작>과 <예약완료>를 클릭하면 크로노미터가 타이머로 동작함
- 날짜/시간 설정: <날짜 설정>, <시간 설정>을 클릭하면 예약 날짜와 시간을 변경 가능
- <예약완료>를 클릭하면 설정한 날짜와 시간이 결정됨

■ 1 안드로이드 프로젝트 생성

- (1) 프로젝트 이름 : Project6_1
- (2) 패키지 이름 : com.cookandroid.project6_1



그림 6-1 날짜/시간 예약 앱 결과 화면

9 / 63

1. 날짜/시간 관련 위젯

■ 실습 6-1 날짜/시간 예약 앱 만들기

■ 2 화면 디자인 및 편집

- (1) [app]-[res]-[layout]-[activity_main.xml] - 아래쪽 [text] 탭에서 코딩
- (2) 바깥의 LinearLayout 안에 다음과 같이 화면을 구성
 - 리니어레이아웃
 - 크로노미터 1개, 버튼 1개 생성 / 위젯 아이디는 chronometer1, btnStart
 - 라디오그룹
 - 라디오버튼 2개 생성 / 위젯 아이디는 rdoCal, rdoTime
 - 리니어레이아웃
 - 리니어레이아웃의 layout_weight를 1로 설정
 - 프레임레이아웃을 두고 안에 캘린더뷰 1개, 타임피커 1개를 생성
 - 위젯의 아이디는 calendarView1, timePicker1
 - 리니어레이아웃
 - 버튼 1개, 텍스트뷰 10개 생성
 - 버튼의 아이디는 btnEnd로, 텍스트뷰는 홀수 차례에만 아이디를 tvYear, tvMonth, tvDay, tvHour, tvMinute로 설정

10 / 63

1. 날짜/시간 관련 위젯

예제 6-4 activity_main.xml

```

1 <LinearLayout>
2   <LinearLayout
3     android:orientation="vertical" >
4     <Chronometer
5       android:id="@+id/chronometer1"
6       android:format="@string/예약에 걸린 시간 %s"
7       android:gravity="center"
8       android:textSize="20dp" />
9     <Button
10      android:id="@+id/btnStart"
11      android:text="@string/예약 시작" />
12   </LinearLayout>
13   <RadioGroup>
14     <RadioButton
15       android:id="@+id/rdoCal"
16       android:text="@string/날짜 설정 (캘린더뷰)" />
17     <RadioButton
18       android:id="@+id/rdoTime"
19       android:text="@string/시간 설정" />
20   </RadioGroup>

```



11 / 63

1. 날짜/시간 관련 위젯

```

21 <LinearLayout
22   android:layout_weight="1" >
23   <FrameLayout>
24     <CalendarView
25       android:id="@+id/calendarView1"
26       android:showWeekNumber="false" />
27     <TimePicker
28       android:id="@+id/timePicker1" />
29   </FrameLayout>
30 </LinearLayout>
31 <LinearLayout
32   android:background="@color/CCCCCC" >
33   <Button
34     android:id="@+id/btnEnd"
35     android:text="@string/예약완료" />
36   <TextView
37     android:id="@+id/tvYear"
38     android:text="@string/0000" />
39   <TextView
40     android:text="@string/년" />
41   <TextView
42     android:text="@string/생략(텍스트뷰 8개)" />
43 </LinearLayout>

```

12 / 63

1. 날짜/시간 관련 위젯

■ 실습 6-1 날짜/시간 예약 앱 만들기

■ 3 Java 코드 작성 및 수정

▪ (1) 전역변수 선언하기

- activity_main.xml에서 id를 부여한 12개 위젯에 대응할 위젯 변수 12개

예제 6-5 Java 코드 1

```
1  ~~~~ 생략(import문) ~~~~
2  public class MainActivity extends Activity {
3      Chronometer chrono;
4      Button btnStart, btnEnd;
5      RadioButton rdoCal, rdoTime;
6      CalendarView calView;
7      TimePicker tPicker;
8      TextView tvYear, tvMonth, tvDay, tvHour, tvMinute;
9      int selectYear, selectMonth, selectDay;
10
11     @Override
12     public void onCreate(Bundle savedInstanceState) {
13         ~~~~ 생략 ~~~~
```

13 / 63

1. 날짜/시간 관련 위젯

■ 실습 6-1 날짜/시간 예약 앱 만들기

■ 3 Java 코드 작성 및 수정

▪ (2) onCreate() 내부에 코딩

- 위젯 변수 12개에 위젯 대입 - 처음에는 캘린더뷰와 타임피커가 보이지 않도록 설정
- 프로젝트 실행 하여 그림과 같은지 확인

14 / 63

1. 날짜/시간 관련 위젯

예제 6-6 Java 코드 2

```

1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4     setTitle("시간 예약");
5
6     btnStart = (Button) findViewById(R.id.btnStart);
7     btnEnd = (Button) findViewById(R.id.btnEnd);
8
9     chrono = (Chronometer) findViewById(R.id.chronometer1);
10
11     rdoCal = (RadioButton) findViewById(R.id.rdoCal);
12     rdoTime = (RadioButton) findViewById(R.id.rdoTime);
13
14     tPicker = (TimePicker) findViewById(R.id.timePicker1);
15     calView = (CalendarView) findViewById(R.id.calendarView1);
16
17     tvYear = (TextView) findViewById(R.id.tvYear);
18     tvMonth = (TextView) findViewById(R.id.tvMonth);
19     tvDay = (TextView) findViewById(R.id.tvDay);
20     tvHour = (TextView) findViewById(R.id.tvHour);
21     tvMinute = (TextView) findViewById(R.id.tvMinute);
22
23     tPicker.setVisibility(View.INVISIBLE);
24     calView.setVisibility(View.INVISIBLE);
25 }

```



15 / 63

1. 날짜/시간 관련 위젯

■ 실습 6-1 날짜/시간 예약 앱 만들기

■ 3 Java 코드 작성 및 수정

- (3) onCreate() 내부에 계속 코딩([예제 6-6]의 24행과 25행 사이)

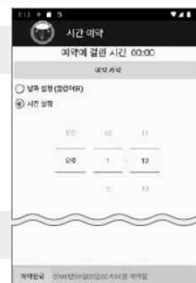
- 클릭 이벤트 리스너 작성 : 라디오버튼을 클릭하면 캘린더뷰와 타임피커 중 하나씩만 보임

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.
9     OnClickListener() {
10     public void onClick(View v) {
11         tPicker.setVisibility(View.VISIBLE);
12         calView.setVisibility(View.INVISIBLE);
13 });

```



16 / 63

1. 날짜/시간 관련 위젯

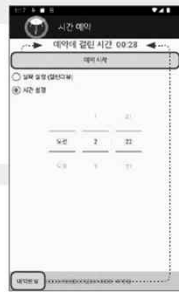
- 실습 6-1 날짜/시간 예약 앱 만들기
- 3 Java 코드 작성 및 수정
 - (4) onCreate() 내부에 계속 코딩
 - 클릭 이벤트 리스너 작성
 - <예약 시작> 클릭 → 크로노미터 시작
 - <예약완료> 클릭 → 크로노미터 정지

17 / 63

1. 날짜/시간 관련 위젯

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 calView.setOnDateChangeListener(new
17     CalendarView.OnDateChangeListener() {
18
19     @Override
20     public void onSelectedDayChange(CalendarView view, int year, int month,
21         int dayOfMonth) {
22
23         selectYear=year;
24         selectMonth=month+1;
25         selectDay=dayOfMonth;
26     }
27 });
```



18 / 63

1. 날짜/시간 관련 위젯

■ 실습 6-1 날짜/시간 예약 앱 만들기

■ 3 Java 코드 작성 및 수정

- (5) [예제 6-8]의 12행과 13행 사이[btnEnd의 onClick() 메소드 안]에 다음 예제를 코딩
 - <예약완료> 클릭
 - 캘린더뷰에서 설정한 연, 월, 일과 타임피커에서 설정한 시, 분이 맨 아래 텍스트뷰에 채워짐

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()));
```

19 / 63

1. 날짜/시간 관련 위젯

▶ 직접 알아보기 6-1

[실습 6-1]을 다음과 같이 수정하라.

- 캘린더뷰 대신에 데이트피커를 사용하여 날짜를 설정한다.
- <예약 시작>과 <예약 완료>를 없앤다. 대신 예약 시작 기능은 크로노미터를 클릭하면 동작하게 하고, 예약 완료 기능은 화면 하단의 연도(0000년)를 톱클릭하면 동작하게 한다.
- 크로노미터를 클릭하기 전에는 라디오버튼, 데이트피커, 타임피커가 안 보이도록 설정하고, 크로노미터를 클릭하면 라디오버튼이 나타나게 한다. 그리고 화면 하단의 연도(0000년)를 톱클릭하면 라디오버튼, 데이트피커, 타임피커가 다시 사라지게 한다.

그림 6-2 수정된 날짜/시간 예약 앱



20 / 63

2. 기타 위젯

■ 자동완성텍스트뷰와 멀티자동완성텍스트뷰

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

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

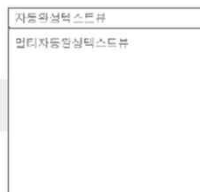
자동완성텍스트뷰, 멀티자동완성텍스트뷰 계층도

21 / 63

2. 기타 위젯

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>
```



22 / 63

2. 기타 위젯

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 }

```



23 / 63

2. 기타 위젯

■ 프로그레스바, 시크바, 레이팅바

- 진행 상태를 표시하는 기능을 함

```

java.lang.Object
├── android.view.View
│   ├── android.widget.ProgressBar
│   ├── android.widget.AbsSeekBar
│   │   ├── android.widget.RatingBar
│   │   └── android.widget.SeekBar

```

프로그레스바, 시크바, 레이팅바 계층도

24 / 63

2. 기타 위젯

■ 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



그림 6-3 프로그레스바, 시크바, 레이팅바

25 / 63

2. 기타 위젯

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>
```

26 / 63

02 뷰 컨테이너

1. 간단한 기능의 뷰 컨테이너

■ 스크롤뷰(ScrollView)

- 수직(위아래)으로 스크롤하는 기능
- 수평(좌우)으로 스크롤하는 수평스크롤뷰(HorizontalScrollView)는 따로 있음
- 주의 : 스크롤뷰에는 단 하나의 위젯만 넣을 수 있음

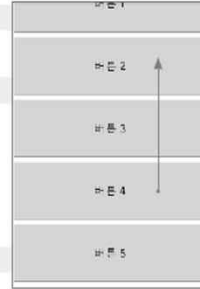
```
java.lang.Object
└ android.view.View
  └ android.widget.ViewGroup
    └ android.widget.FrameLayout
      └ android.widget.ScrollView
```

스크롤뷰 계층도

1. 간단한 기능의 뷰 컨테이너

예제 6-13 스크롤뷰의 XML 코드

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



29 / 63

1. 간단한 기능의 뷰 컨테이너

■ 슬라이딩드로어(SlidingDrawer)

- 위젯들을 서랍처럼 열어서 보여주거나 닫아서 감춤

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

슬라이딩드로어 계층도

30 / 63

1. 간단한 기능의 뷰 컨테이너

■ 슬라이딩드로어의 일반적 형태

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

    // 이곳이 필요한 위젯을 넣는 서랍 내부

  </리니어레이아웃>
</슬라이딩드로어>
```

■ 규칙

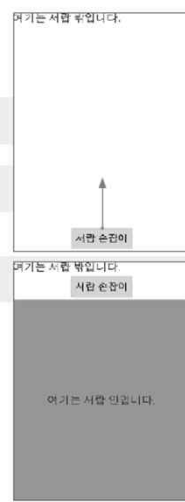
- 1. 슬라이딩드로어의 handle 이름과 슬라이딩드로어의 손잡이 역할을 하는 버튼 id가 동일해야 함
- 2. 슬라이딩드로어의 content 이름과 리니어레이아웃의 id가 동일해야 함

31 / 63

1. 간단한 기능의 뷰 컨테이너

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 </SlidingDrawer>
21 </LinearLayout>
```



32 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 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 개요도

■ 뷰플리퍼의 일반적 형태

```
<리니어레이아웃>
<리니어레이아웃>
    // 왼쪽 또는 오른쪽으로 전환할 버튼이나 이미지뷰
</리니어레이아웃>
<뷰플리퍼>

    // 한 번에 하나씩 보여줄 위젯 삽입

</뷰플리퍼>
</리니어레이아웃>
```

33 / 63

2. 복잡한 기능의 뷰 컨테이너

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="#ff0000" >
15        ~~~~ 필요한 위젯 삽입 ~~~~
16      </LinearLayout>
17      <LinearLayout
18        android:background="#00ff00" >
19        ~~~~ 필요한 위젯 삽입 ~~~~
20      </LinearLayout>
21      <LinearLayout
22        android:background="#0000ff" >
23        ~~~~ 필요한 위젯 삽입 ~~~~
24      </LinearLayout>
25    </ViewFlipper>
26  </LinearLayout>
```



34 / 63

2. 복잡한 기능의 뷰 컨테이너

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    ~~~~ 생략(다음 화면 버튼 1개) ~~~~
18 }

```

35 / 63

2. 복잡한 기능의 뷰 컨테이너

▶ 직접 알아보기 6-2

뷰플리퍼를 이용하여 자동 사진 보기 앱을 작성하라.

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

힌트 화면 넘김 시작 메소드로 startFlipping(), 중지 메소드로 stopFlipping(), 화면 넘김 간격 메소드로 setFlipInterval(밀리초)을 사용한다.

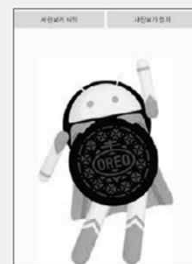


그림 6-4 자동 사진 보기 앱

36 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 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 계층도

■ 탭 호스트의 구성

```

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

        // 이곳에 각 탭스택에 대응할 탭 화면(레이아웃)을 3개 삽입

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

37 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 탭호스트(TabHost)

- 탭호스트, 탭위젯, 프레임레이아웃은 지정된 id를 변경하지 않아야 안드로이드가 탭호스트의 구성임을 인식함

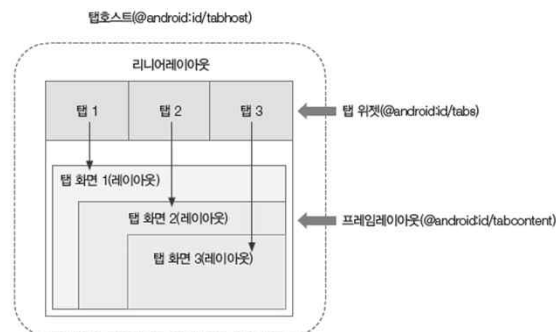


그림 6-5 탭호스트의 구성 방식

38 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 탭호스트(TabHost)

- 탭을 생성하고 탭 화면을 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
```

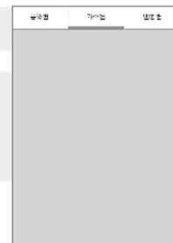
- 탭스펙(TabSpec) : 탭을 구성하는 요소들의 집합

39 / 63

2. 복잡한 기능의 뷰 컨테이너

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>
```



40 / 63

2. 복잡한 기능의 뷰 컨테이너

예제 6-18 탭호스트의 Java 코드

```

1  ~~~~ 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 }

```

41 / 63

2. 복잡한 기능의 뷰 컨테이너

▶ 직접 풀어보기 6-3

탭호스트를 이용하여 동물 선택 앱을 작성하라.

- 탭위젯을 아래쪽에 배치하고 탭 4개가 나오게 한다.
- 프레임레이아웃 안의 리니어레이아웃 3개를 제거하고 4개의 이미지뷰를 배치한다.

힌트 프레임레이아웃의 layout_weight 속성을 1로 한다.

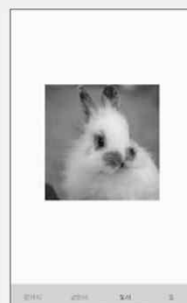


그림 6-6 동물 선택 앱

42 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 액션바(Action Bar)

- 액션바(Action Bar)는 허니콤(Android 3.0, API 11)에서 태블릿과 같은 대형 화면에서 여러 화면을 사용하기 위해서 고안됨
- 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 활용됨

43 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 액션바와 프래그먼트를 활용한 화면 구현

- 액션바를 생성한 후, 탭(Tab) 위젯을 액션바에 등록
- XML 사용하지 않고 모두 Java로 구현

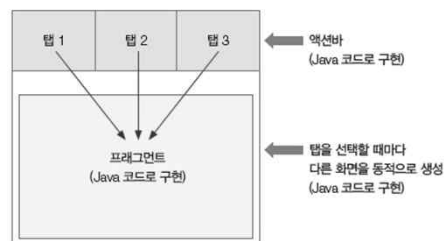


그림 6-7 액션바와 프래그먼트의 구성 방식

44 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 액션바와 프래그먼트를 활용하여 화면 구현

- 액션바에 탭을 추가하는 Java 코드 형식

```
ActionBar bar = getActionBar(); // 상단에 표시할 액션바 준비
// 탭호스트와 같이 탭의 모양이 되도록 설정
bar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
tabSong = bar.newTab(); // 액션바에 탭 생성
tabSong.setText("음악별"); // 탭의 글자 설정
tabSong.setTabListener(this); // 탭을 터치하면 작동하는 리스너 지정
bar.addTab(tabSong); // 액션바에 탭 추가
```

45 / 63

2. 복잡한 기능의 뷰 컨테이너

예제 6-19 액션바의 Java 코드 1

```
1  ~~~~ 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         ~~~~ 생략(tabArtist, tabAlbum) ~~~~
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 }
```



46 / 63

2. 복잡한 기능의 뷰 컨테이너

- Fragment를 상속받는 MyTabFragment 클래스를 내부 클래스로 생성

예제 6-20 액션바의 Java 코드 2

```

1 public static class MyTabFragment extends androidx.fragment.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 == "음악별") baseLayout.setBackgroundColor(Color.RED);
17         if (tabName == "가수별") baseLayout.setBackgroundColor(Color.GREEN);
18         if (tabName == "앨범별") baseLayout.setBackgroundColor(Color.BLUE);
19
20         return baseLayout;
21     }
22 }

```

47 / 63

2. 복잡한 기능의 뷰 컨테이너

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

예제 6-21 액션바의 Java 코드 3

```

1 MyTabFragment myFragments[] = new MyTabFragment[3];
2
3 public void onTabSelected(Tab tab,
4                           FragmentTransaction ft) {
5     MyTabFragment myTabFrag = null;
6
7     if (myFragments[tab.getPosition()] == null) {
8         myTabFrag = new MyTabFragment();
9         Bundle data = new Bundle();
10        data.putString("tabName", tab.getText().toString());
11        myTabFrag.setArguments(data);
12        myFragments[tab.getPosition()] = myTabFrag;
13    }
14    else
15        myTabFrag = myFragments[tab.getPosition()];
16
17    ft.replace(android.R.id.content, myTabFrag);
18 }

```



48 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 웹뷰(WebView)

- 사용자가 웹브라우저 기능을 앱 안에 직접 포함시킬 수 있는 위젯



49 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

- 에디트텍스트에 URL을 입력하면 해당 링크로 이동하는 앱
- 화면의 로고와 프로그램의 아이콘 변경

■ 1 안드로이드 프로젝트 생성

- (1) 프로젝트 이름 : Project6_2
- (2) 패키지 이름 : com.cookandroid.project6_2

■ 2 화면 디자인 및 편집

- (1) 아래 규칙에 맞는 화면 코딩
 - 리니어레이아웃을 하나 더 만들고,
그 안에 에디트텍스트 1개와 버튼 2개로 구성
 - 하단에는 웹뷰를 만들
 - 각 위젯의 id는 edtUrl, btnGo, btnBack, webView1



그림 6-8 간단 웹브라우저 앱 결과 화면

50 / 63

2. 복잡한 기능의 뷰 컨테이너



51 / 63

2. 복잡한 기능의 뷰 컨테이너

- 실습 6-2 간단 웹브라우저 만들기
- 2 화면 디자인 및 편집
 - (2) [app]-[manifests]-[AndroidManifest.xml] 파일을 연다.
 - AndroidManifest.xml : 보통 매니페스트라고 부름
 - 매니페스트 파일 : 프로젝트의 전반적인 환경 설정하는 파일

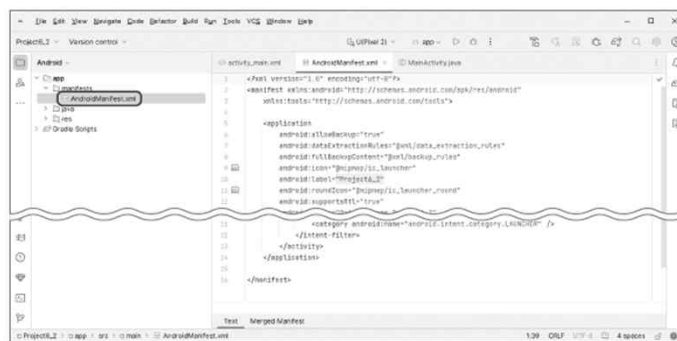


그림 6-9 AndroidManifest.xml 파일 열기

52 / 63

2. 복잡한 기능의 뷰 컨테이너

- 실습 6-2 간단 웹브라우저 만들기
- 2 화면 디자인 및 편집
 - (3) 메니페스트 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_name"
8         android:logo="@drawable/web"
9         android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
10
11         <activity
12             android:name=".MainActivity"
13             android:label="@string/app_name"
14             android:exported="true">
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     </application>
22 </manifest>
    
```

53 / 63

2. 복잡한 기능의 뷰 컨테이너

- 실습 6-2 간단 웹브라우저 만들기
- 2 화면 디자인 및 편집
 - (4) 실행
 - 화면 상단에 아이콘과 타이틀이 변경된 것 확인



그림 6-10 메니페스트에서 설정한 내용 확인

54 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

■ 3 Java 코드 작성 및 수정

- (1) activity_main.xml의 4개 위젯에 대응할 위젯 변수 4개를 전역변수로 선언
- (2) onCreate() 메소드 안의 각 변수에 위젯을 대입

예제 6-24 Java 코드 1

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

55 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

■ 3 Java 코드 작성 및 수정

- (3) WebViewClient의 상속을 받는 웹뷰클라이언트 클래스 정의
 - [예제 6-24]의 18행에 다음 코드를 입력
 - onCreate() 메소드 밖, 메인 클래스 안에 위치

예제 6-25 Java 코드 2

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

56 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

■ 3 Java 코드 작성 및 수정

- (4) CookWebViewClient 클래스 안에 커서를 두고 Android Studio 메뉴의 [Code]-[Override Methods]를 선택
- (5) 부모 클래스(여기서는 WebViewClient)에서 오버라이딩 또는 임플리먼트가 가능한 메소드 목록이 나옴
- (6) 메소드 중 shouldOverrideUrlLoading()를 선택하고 <OK> 클릭

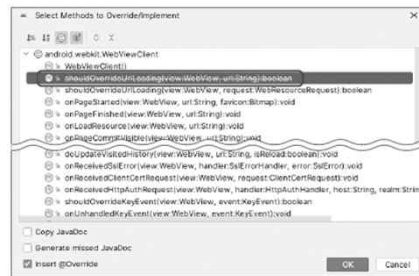


그림 6-11 메소드의 자동 오버라이딩

57 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

■ 3 Java 코드 작성 및 수정

- (4)~(6)을 통해 자동으로 오버라이딩된 메소드

```

예제 6-26   Java 코드 3
1      ~~~~ 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 }
    
```

58 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

■ 3 Java 코드 작성 및 수정

- (7) onCreate() 메소드 안에 계속 코딩([예제 6-26]의 2행과 3행 사이)
 - 생성한 CookWebViewClient 클래스를 웹뷰에 설정
 - 화면 확대/축소 아이콘이 보이도록 설정
 - 두 버튼을 클릭하면 입력한 URL로 이동하거나 이전 화면으로 돌아가도록 함

59 / 63

2. 복잡한 기능의 뷰 컨테이너

예제 6-27 Java 코드 4

```
1  ~~~~ 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  webSet.setJavaScriptEnabled(true);
9  btnGo.setOnClickListener(new View.OnClickListener() {
10     public void onClick(View v) {
11         web.loadUrl(edtUrl.getText().toString());
12     }
13 });
14
15 btnBack.setOnClickListener(new View.OnClickListener() {
16     public void onClick(View v) {
17         web.goBack();
18     }
19 });
20 }
21 ~~~~ 생략 ~~~~
```

60 / 63

2. 복잡한 기능의 뷰 컨테이너

■ 실습 6-2 간단 웹브라우저 만들기

■ 4 프로젝트 실행 및 결과 확인

- 에디트텍스트에 'http://m.daum.net'을 입력하고 <이동> 클릭
 - 웹페이지 열리지 않는 이유 : 프로젝트에 인터넷을 사용할 수 있는 퍼미션을 주지 않음
 - 퍼미션은 AndroidManifest.xml에서 지정



그림 6-12 웹페이지가 열리지 않음

61 / 63

2. 뷰 컨테이너 ▶ 복잡한 기능의 뷰 컨테이너

■ [반복] 2 화면 디자인 및 편집

- AndroidManifest.xml을 열고 <application> 행 위에 다음 한 행을 추가
 - 인터넷을 사용하는 프로젝트마다 AndroidManifest.xml에 이 행을 추가

```
예제 6-28 AndroidManifest.xml
1  ~~~~ 생략 ~~~~
2
3
4  <uses-permission android:name="android.permission.INTERNET" />
5
6  <application
7      android:icon="@drawable/emo_im_cool"
8  ~~~~ 생략 ~~~~
```

62 / 63



감사합니다.