

Ch06. 고급 위젯 다루기2

모바일앱프로그래밍
김지심교수

학습목표

고급 위젯을 다루는 방법을 이해한다.

뷰컨테이너와 응용법을 익혀 프로그래밍할 수 있다.

매니페스트 파일을 설정할 수 있다.

Chronometer

TimePicker

CalendarView

DatePicker

AutoCompleteTextView

MultiAutoCompleteTextView

ProgressBar

SeekBar

RatingBar

ScrollView

SlidingDrawer

ViewFlipper

TabHost

WebView

ActionBar

목차

1. 고급 위젯
2. 뷰컨테이너



01



고급 위젯



1. 고급 위젯 | 자동완성텍스트뷰

자동완성텍스트뷰와 멀티자동완성텍스트뷰

EditText에 자동완성 기능을 추가한 뷰

자동완성텍스트뷰는 1개, 멀티자동완성텍스트뷰는 자동완성한 문자열을 2개 이상(쉼표로 구분) 텍스트뷰에 출력

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

자동완성텍스트뷰 계층도

1. 고급 위젯 | 자동완성텍스트뷰

Reference

AutoCompleteTextView

1) ArrayAdapter<T> (Context context, int resource, T[] objects) 객체를 생성하여

resource: 자동완성 목록에서 항목을 출력할 뷰(TextView)가 정의된 리소스의 id

T[] objects: Generics(모든 종류의 타입을 다룰 수 있도록 일반화된 타입 매개변수) p.148
자동완성 목록의 항목에 출력할 객체 타입의 데이터(T Type)

2) 자동완성텍스트뷰에 adapter를 설정

ex. auto.setAdapter(adapter)

→ 자동완성 목록에 설정된 형식으로 데이터를 출력하라!

1. 고급 위젯 | 자동완성텍스트뷰

Reference

MultiAutoCompleteTextView

1) 자동완성텍스트뷰에 adapter를 설정

ex. multi.setAdapter(adapter)

2) MultiAutoCompleteTextView.CommaTokenizer 객체를 생성하여

자동완성된 항목을 콤마로 구분하여 멀티오토컴플릿텍스트뷰에 출력할 수 있도록 객체 생성

3) MultiAutoCompleteTextView에 설정하고

ex. multi.setTokenizer(token);

1. 고급 위젯 | 자동완성텍스트뷰

예제6-10: Ex06_10

자동완성텍스트뷰와 멀티자동완성텍스트뷰 예제
activity_main.xml

예제 6-10 자동완성텍스트뷰 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>
```



1. 고급 위젯 | 자동완성텍스트뷰

MainActivity.java

```
12 public class MainActivity extends AppCompatActivity {
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         String[] items = {"CSI-뉴욕", "CSI-라스베가스", "CSI-마미에미", "Friends", "Fringe", "Lost", ""};
20
21         //AutoCompleteTextView
22         AutoCompleteTextView auto = (AutoCompleteTextView) findViewById(R.id.autoCompleteTextView1);
23         MultiAutoCompleteTextView multi = (MultiAutoCompleteTextView) findViewById(R.id.multiAutoCompleteTextView1);
24
25         //adapter 객체 생성
26         ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line, items);
27         auto.setAdapter(adapter);
28
29
30         //MultiAutoCompleteTextView
31         multi.setAdapter(adapter);
32         MultiAutoCompleteTextView.CommaTokenizer token = new MultiAutoCompleteTextView.CommaTokenizer();
33         multi.setTokenizer(token);
34
35     }
36 }
```

- * 'FR'까지 입력하여 자동완성 → 'LO'까지 입력하여 자동완성
- * 자동완성 가능한 글자수 변경해보기

1. 고급 위젯 | 진행/척도 위젯

프로그레스바, 시크바, 레이팅바

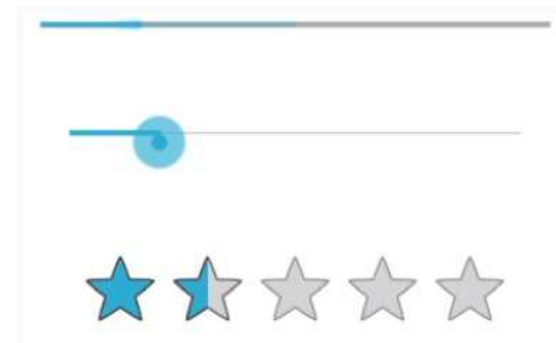
```
java.lang.Object
├─ android.view.View
│   └─ android.widget.ProgressBar
│       └─ android.widget.AbsSeekBar
│           ├── android.widget.RatingBar
│           └─ android.widget.SeekBar
```

프로그레스바, 시크바, 레이팅바 계층도

프로그레스바(ProgressBar) : 작업의 진행 상황을 바(Bar)나 원 형태로 제공

시크바(SeekBar) : 프로그레스바와 대부분 비슷, 사용자 터치로 임의 조절 가능
음량 조절, 동영상 재생위치 지정 등

레이팅바(RatingBar) : 진행 상황을 별 모양으로 표시, 척도 선택



1. 고급 위젯 | 진행/척도 위젯

디자인 모드에서 각 위젯 배치하고 AVD에서 실행해보기

ProgressBar 속성

max: 최대 범위

progress: 초기값

secondaryProgress: 보조값

SeekBar

progress

RatingBar

numStars: 별 갯수

rating: 초기값

stepSize: 스텝 크기

1. 고급 위젯 | 진행/척도 위젯

예제6-12: Ex06_12

프로그레스바, 시크바, 레이팅바 예제

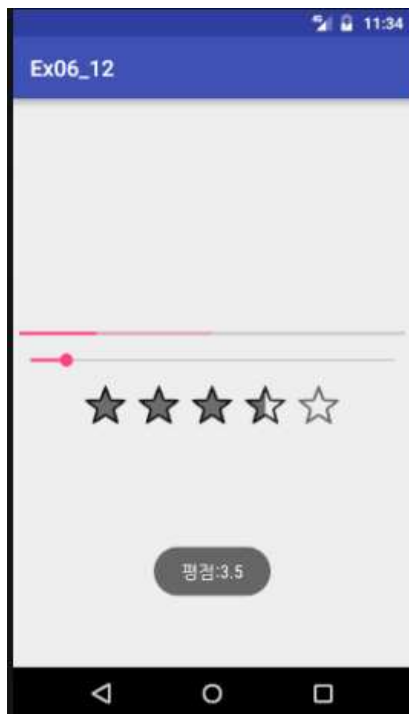
예제 6-12 프로그레스바, 시크바, 레이팅바 XML 코드

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

1. 고급 위젯 | 진행/척도 위젯

MainActivity.java

Rating 결과 출력: setOnRatingBarChangeListener 리스너 사용



```
8 public class MainActivity extends AppCompatActivity {
9     RatingBar rB;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         rB = (RatingBar) findViewById(R.id.rB);
17
18         rB.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
19             @Override
20             public void onRatingChanged(RatingBar ratingBar, float v, boolean b) {
21                 Toast.makeText(getApplicationContext(), "평점:" + v +
22                     "상태: " + String.valueOf(b), Toast.LENGTH_SHORT).show();
23             }
24         });
25     }
26 }
27
```

02



뷰컨테이너



2. 뷰컨테이너

뷰컨테이너

4장에서 배운 5개 레이아웃을 제외한, ViewGroup에서 상속받은 위젯들

간단한 뷰컨테이너: ScrollView, SlidingDrawer

복잡한 뷰컨테이너: ViewFlipper, TabHost, WebView, ActionBar

2. 뷰컨테이너 | 스크롤뷰

스크롤뷰(ScrollView)

스크롤뷰: 수직(위아래)으로 스크롤

기본적으로 수직 스크롤 지원

* 수평 스크롤: HorizontalScrollView

child 위젯들은 주로 컨테이너(ViewGroup)가 됨

* 스크롤뷰에는 단 하나의 뷰그룹만 넣을 수 있음
(1개 Layout을 넣고 그 안에 여러 개 위젯을 배치)

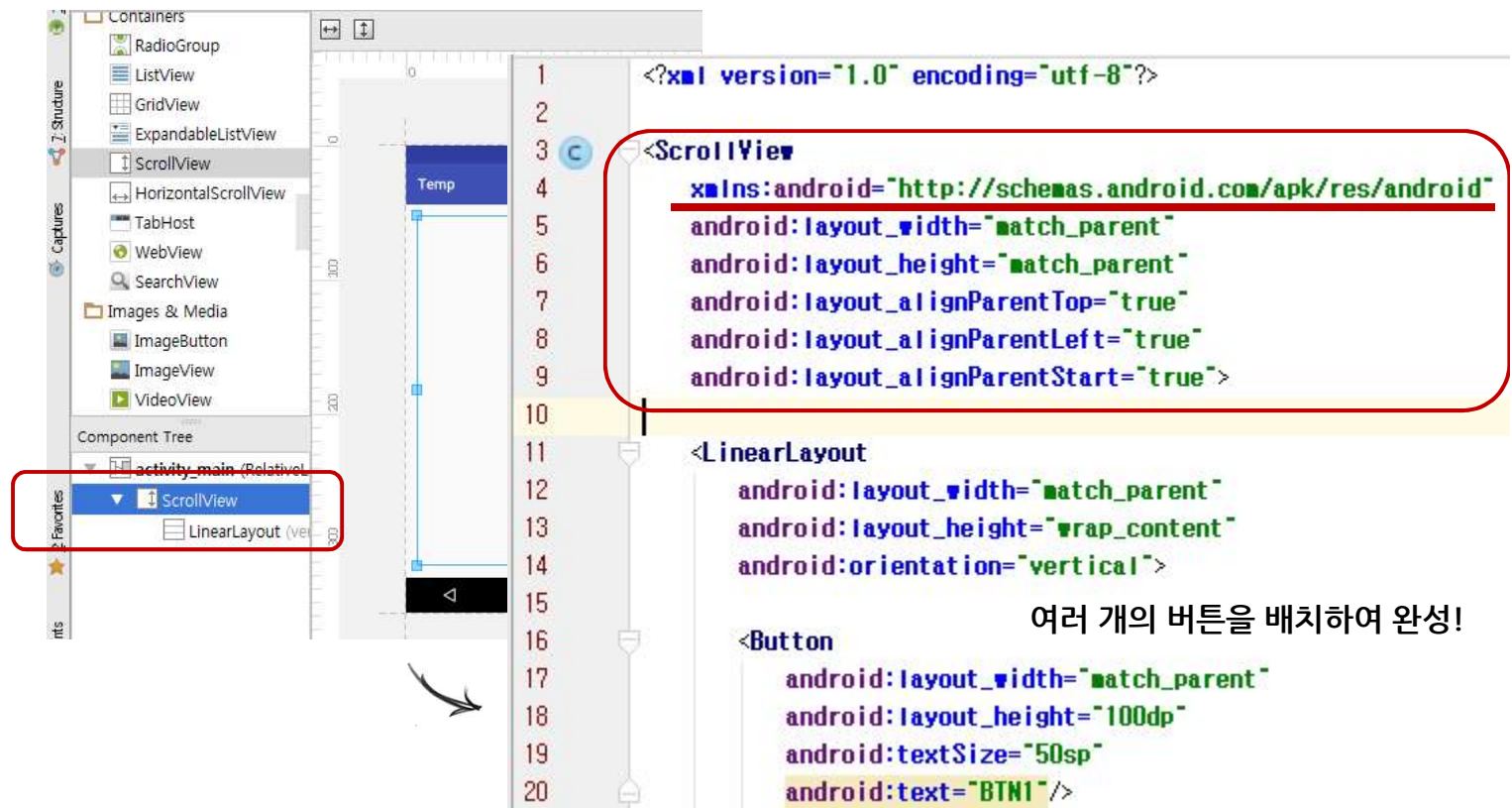
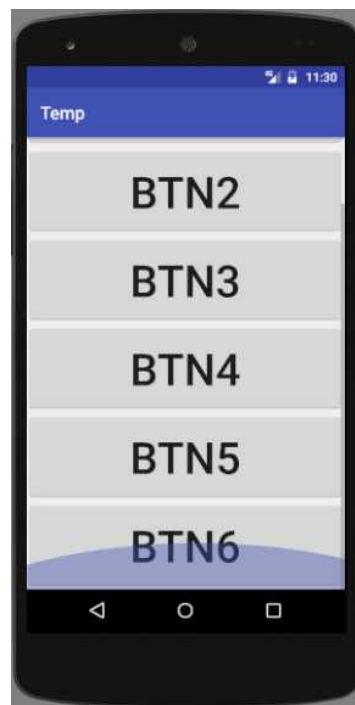
```
java.lang.Object
    ↳ android.view.View
        ↳ android.widget.ViewGroup
            ↳ android.widget.FrameLayout
                ↳ android.widget.ScrollView
```

스크롤뷰 계층도

예제6-13: Ex06_13

디자인 모드에서 스크롤뷰 배치→ LinearLayout을 포함한 스크롤뷰가 생성됨

XML에서 루트 엘리먼트는 1개여야 하므로, 스크롤뷰로 대체하기 (namespace 명시)



2. 뷰컨테이너 | 슬라이딩드로어

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

- * handle: 손잡이, handle의 속성값 = 손잡이 역할의 버튼 id
- * content: 내부 뷰와 연결, content의 속성값 = 레이아웃의 id

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

    // 이곳이 서랍 내부이며, 필요한 위젯들을 넣으면 됨


  </리니어레이아웃>
</슬라이딩드로어>
```

2. 뷰컨테이너 | 슬라이딩드로어

예제6-14: Ex06_14

서랍 안 리니어레이아웃 보여주는 슬라이딩드로어 예제

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context="com.example.kjs.ex06_14.MainActivity">
9   <TextView
10     android:layout_width="wrap_content"
11     android:layout_height="wrap_content"
12     android:text="서랍밖" />
13
14   <SlidingDrawer
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:handle="@+id/handle"
18     android:content="@+id/content">
19     <Button
20       android:layout_width="wrap_content"
21       android:layout_height="wrap_content"
22       android:id="@+id/handle"
23       android:text="손잡이" />
24
25     <LinearLayout
26       android:layout_width="match_parent"
27       android:layout_height="wrap_content"
28       android:id="@+id/content"
29       android:orientation="vertical"
30       android:gravity="center"
31       android:background="#00ff00">
32       <TextView
33         android:layout_width="wrap_content"
34         android:layout_height="wrap_content"
35         android:text="서랍안" />
36     </LinearLayout>
37   </SlidingDrawer>
38 </LinearLayout>
```



2. 뷰컨테이너 | 뷰플리퍼

뷰플리퍼(ViewFlipper)

안에 여러 개의 위젯을 배치한 후, 필요에 따라서 화면을 왼쪽과 오른쪽으로 밀어서 하나의 위젯씩 화면에 보여주는 방식의 뷰컨테이너

뷰플리퍼의 형식

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

뷰플리퍼 계층도

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

    // 여기에 한번에 하나씩 보여줄 위젯들을 넣음

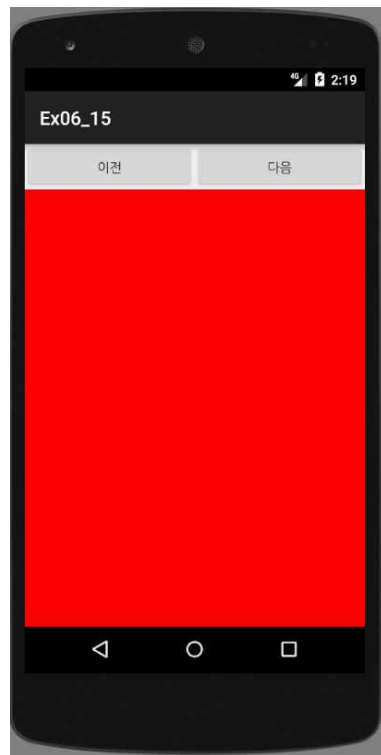
</뷰플리퍼 >
</리니어레이아웃>
```


2. 뷰컨테이너 | 뷰플리퍼

예제6-15: Ex06_15

배경색이 다른 리니어레이아웃들을
보여주는 예제

activity_main.xml



```
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context="com.android.kjs.ex06_15.MainActivity"
8   android:orientation="vertical">
9
10  <LinearLayout
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content">
13    <Button
14      android:layout_width="wrap_content"
15      android:layout_height="wrap_content"
16      android:id="@+id/btnPrev"
17      android:layout_weight="1"
18      android:text="이전"
19      android:onClick="Clicked"/>
20    <Button
21      android:layout_width="wrap_content"
22      android:layout_height="wrap_content"
23      android:id="@+id/btnNext"
24      android:layout_weight="1"
25      android:text="다음"
26      android:onClick="Clicked"/>
27  </LinearLayout>
28
29  <ViewFlipper
30    android:layout_width="match_parent"
31    android:layout_height="match_parent"
32    android:id="@+id/vf1">
33    <LinearLayout
34      android:layout_width="match_parent"
35      android:layout_height="match_parent"
36      android:background="#ff0000"/>
37    <LinearLayout
38      android:layout_width="match_parent"
39      android:layout_height="match_parent"
40      android:background="#00ff00"/>
41    <LinearLayout
42      android:layout_width="match_parent"
43      android:layout_height="match_parent"
44      android:background="#0000ff"/>
45  </ViewFlipper>
46 </LinearLayout>
```

2. 뷰컨테이너 | 뷰플리퍼

MainActivity.java

버튼 클릭시, showPrevious(),
showNext() 사용하여 전환

→ XML에서 버튼의 onClick 속성과
자바에서 onClick 메소드를 사용하여
완성하시오!

```
1 package com.android.kjs.ex06_15;
2
3 import ...
4
5
6
7
8
9 public class MainActivity extends AppCompatActivity {
10
11     ViewPager vF1;
12     Button btnPrev, btnNext;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         btnPrev= (Button) findViewById(R.id.btnPrev);
20         btnNext= (Button) findViewById(R.id.btnNext);
21         vF1 = (ViewPager) findViewById(R.id.vF1);
22
23         btnPrev.setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View v) {
26                 vF1.showPrevious();
27             }
28         });
29
30         btnNext.setOnClickListener(new View.OnClickListener() {
31             @Override
32             public void onClick(View v) {
33                 vF1.showNext();
34             }
35         });
36     }
37 }
38
```

2. 뷰컨테이너 | 뷰플리퍼

실습(직접 풀어보기) 6-2: Test06_02

▶ 직접 풀어보기 6-2

뷰플리퍼를 이용해서 자동 사진 보기 앱을 작성하자.

- 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.

HINT 화면 넘김 시작 메소드로 `startFlipping()`, 중지 메소드로 `stopFlipping()`, 화면 넘김 간격 메소드로 `setFlipInterval(밀리초)`를 사용한다.



2. 뷰컨테이너 | 탭호스트

탭호스트(TabHost)

여러 탭을 배치하고 각 탭을 클릭할 때마다 해당 화면이 나오도록 설정하는 뷰컨테이너

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

탭호스트 계층도

2. 뷰컨테이너 | 탭호스트

탭호스트



* 지정된 id를 그대로 사용해야 함

```
<탭호스트 id="@android:id/tabhost">  
  <리니어레이아웃>  
    <탭위젯 id="@android:id/tabs" />  
    <프레임레이아웃 id="@android:id/tabcontent">  
  
        // 여기에 각 탭스펙에 대응할 탭화면(레이아웃)을 3개 넣음  
  
    </프레임레이아웃>  
  </리니어레이아웃>  
</탭호스트>
```

2. 뷰컨테이너 | 탭호스트

[Referece]

TabActivity를 상속하여 구현

탭스펙(탭을 구성하는 요소들의 집합)을 인플레이션하고 탭호스트에 설정하는 방식으로 구현

```
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
```

1) 탭을 구현하는 TabHost 객체를 얻어옴

2) TabWidget을 구현하는 TabSpec 객체를 생성하고 indicator(텍스트) 설정

```
= TabHost.TabSpec tabSpecSong = tabHost.newTabSpec("SONG");
tabSpecSong.setIndicator("음악별");
```

3) 탭별 화면을 설정하고 TabHost에 add

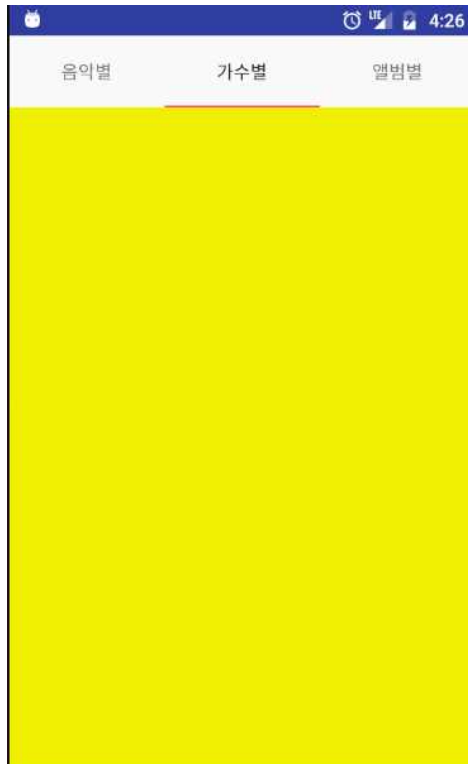
2. 뷰컨테이너 | 탭호스트

예제6-17: Ex06_17

activity_main.xml

탭호스트 틀과 프레임레이아웃 디자인

* TabHost가 루트 엘리먼트여야 함



```
2 <TabHost xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:id="@android:id/tabhost">
6
7     <LinearLayout
8         android:layout_width="match_parent"
9         android:layout_height="match_parent"
10        android:orientation="vertical">
11
12        <TabWidget
13            android:id="@android:id/tabs"
14            android:layout_width="match_parent"
15            android:layout_height="wrap_content" />
16
17        <FrameLayout
18            android:id="@android:id/tabcontent"
19            android:layout_width="match_parent"
20            android:layout_height="match_parent">
21            <LinearLayout
22                android:id="@+id/tabSong"
23                android:layout_width="match_parent"
24                android:layout_height="match_parent"
25                android:background="#DAA520"
26                android:orientation="vertical" />
27            <LinearLayout
28                android:id="@+id/tabArtist"
29                android:layout_width="match_parent"
30                android:layout_height="match_parent"
31                android:background="#f0f000"
32                android:orientation="vertical" />
33            <LinearLayout
34                android:id="@+id/tabAlbum"
35                android:layout_width="match_parent"
36                android:layout_height="match_parent"
37                android:background="#FFD700"
38                android:orientation="vertical" />
39        </FrameLayout>
40    </LinearLayout>
41</TabHost>
```

2. 뷰컨테이너 | 탭호스트

MainActivity.java

```
7 public class MainActivity extends TabActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13
14        TabHost tabHost = getTabHost();
15
16        TabHost.TabSpec tabSpecSong = tabHost.newTabSpec("SONG").setIndicator("음악별");
17        tabSpecSong.setContent(R.id.tabSong);
18        tabHost.addTab(tabSpecSong);
19
20
21
22
23
24
25
26
27
28
29
30    }
31 }
```

완성해보기!

2. 뷰컨테이너 | 탭호스트

실습(직접 풀어보기): Test06_03

▶ 직접 풀어보기 6-3

탭호스트를 이용해서 동물 선택 앱을 작성하자.

- 탭위젯을 아래쪽에 배치하고, 탭은 4개가 나오도록 한다.
- 프레임레이아웃 안의 3개의 리니어레이아웃을 제거하고, 4개의 이미지뷰로 배치한다.

HINT 프레임레이아웃의 layout_weight 속성을 1로 준다.



2. 뷰컨테이너 | 웹뷰

웹뷰(WebView)

사용자가 웹브라우저 기능을 앱 안에 직접 포함시킬 수 있는 위젯
WebViewClient를 상속받아 웹 관련 속성을 설정하고 구현

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

웹뷰 계층도

[Reference]

WebView 클래스: 웹뷰 실행 구현 (필수)

setWebViewClient

ex.web.loadUrl(edtUrl.getText().toString());

WebSettings 클래스: 웹뷰 관련 설정

ex. setBuiltInZoomControls(true)

2. 뷰컨테이너 | 웹뷰

[Reference]

1) WebView에 WebViewClient 설정

외부 브라우저앱이나 새로운 창을 사용하지 않고 웹뷰에서 웹사이트 열도록 설정

2) WebSettings로 상세 속성들을 설정 (생략 가능)

ex. `setBuiltInZoomControls(true)`

3) 웹사이트 로딩

`loadUrl(String url);`

ex. `web.loadUrl edtUrl.getText().toString());`

* 매니페스트에서 권한을 설정해야 함

2. 뷰컨테이너 | 웹뷰

실습6-2: Project06_02

액션바에 아이콘, 제목을 출력하고
웹뷰로 인터넷 접속

activity_main.xml



```
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/activity_main"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7     <LinearLayout
8         android:orientation="horizontal"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content">
11        <EditText
12            android:layout_width="wrap_content"
13            android:layout_height="wrap_content"
14            android:id="@+id/edtURL1"
15            android:layout_weight="1"
16            android:maxLines="1"
17            android:hint="URL을 입력하세요" />
18        <Button
19            android:text="이동"
20            android:layout_width="wrap_content"
21            android:layout_height="wrap_content"
22            android:id="@+id/btnGo"
23            android:layout_weight="1" />
24        <Button
25            android:text="이전"
26            android:layout_width="wrap_content"
27            android:layout_height="wrap_content"
28            android:id="@+id/btnBack"
29            android:layout_weight="1" />
30    </LinearLayout>
31    <WebView
32        android:layout_width="match_parent"
33        android:layout_height="match_parent"
34        android:id="@+id/webView1"/>
35 </LinearLayout>
```


2. 뷰컨테이너 | 웹뷰

AndroidManifest.xml

액션바에 아래 요소들을 출력하기 위해, 우선 매니페스트 설정

* AndroidManifest.xml(매니페스트) : 프로젝트 실행에 필요한 요소를 설정하는 파일

아이콘·로고 리소스, 레이블을 수정

```
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   package="com.example.kjs.project06_02">
4
5   <application
6     android:allowBackup="true"
7     android:icon="@drawable/chrome"
8     android:logo="@drawable/ie"
9     android:label="안드로이드 웹브라우저"
10    android:theme="@style/AppTheme"
11    android:supportRtl="true">
12
13     <activity android:name=".MainActivity"
14       android:label="간단 웹브라우저">
15
16       <intent-filter>
17         <action android:name="android.intent.action.MAIN" />
18
19         <category android:name="android.intent.category.LAUNCHER" />
20       </intent-filter>
21     </activity>
22   </application>
23
24 </manifest>
```

2. 뷰컨테이너 | 웹뷰

MainActivity.java

```
11 public class MainActivity extends AppCompatActivity {
12
13     EditText edtUrl;
14     Button btnGo, btnBack;
15     WebView web;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         setTitle("간단 웹브라우저");
23
24         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
25         getSupportActionBar().setIcon(R.drawable.ie);
26
27
28         edtUrl = (EditText) findViewById(R.id.edtURL1);
29         btnGo = (Button) findViewById(R.id.btnGo);
30         btnBack = (Button) findViewById(R.id.btnBack);
31         web = (WebView) findViewById(R.id.webView1);
32     }
```

setTitle("간단 웹브라우저"); 매니페스트에 설정된 label을 변경하고 싶은 경우 설정

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setIcon(R.drawable.ie);

매니페스트에서 설정된 로고를 액션바에 출력(p.76)
* 매니페스트에만 설정하면 로고가 출력되지 않음
(setIcon 메소드 사용해야 함)

2. 뷰컨테이너 | 웹뷰

MainActivity.java

웹뷰를 설정하고 Url 로딩

```
33 web.setWebViewClient(new WebViewClient()); //웹뷰로 사이트 탐색할 수 있도록 웹뷰에 Client를 연결
34 web.setWebViewClient(new myWebViewClient());
35
36 //
37 //
38
39 btnGo.setOnClickListener(new View.OnClickListener() {
40     @Override
41     public void onClick(View view) {
42         web.loadUrl edtUrl.getText().toString();
43     }
44 });
45
46 btnBack.setOnClickListener(new View.OnClickListener() {
47     @Override
48     public void onClick(View view) {
49         web.goBack();
50     }
51 });
52 }
53
54 // class myWebViewClient extends WebViewClient {
55 //     @Override
56 //     public boolean shouldOverrideUrlLoading(WebView view, String url) {
57 //         view.loadUrl(url);
58 //         return true;
59 //     }
60 // }
61 }
```

주석 처리 후 실행해보자.

WebSettings 클래스를 이용하여 안드로이드 빌트인 줌 컨트롤 사용토록 설정
But, 해당 웹사이트에서 기능 구현하지 않은 경우 사용 불가

새 사이트를 열 경우 외부 앱을 이용하지 않고 웹뷰에서 열도록 설정

2. 뷰컨테이너 | 웹뷰

프로젝트 실행 및 결과 확인

“http://daum.net”을 입력하고 <이동>을 클릭



2. 뷰컨테이너 | 웹뷰

퍼미션 설정

AndroidManifest.xml에 인터넷 사용 퍼미션 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 ~~~~
```

2. 뷰컨테이너 | 웹뷰

실행

아이콘, 로고, 타이틀 등 확인: [설정] > [애플리케이션]

