7. 다양한 위젯 활용

1. 웹브라우저(WebView) 사용하기
2. 애니메이션 사용하기
3. 페이지 슬라이딩
4. 뷰 플리퍼 사용
5. 프로그래스바 사용
6. 시크바 사용

1. 웹브라우저 사용하기

- **롤리팝 버전 이후 웹브라우저 특징**
  - 크롬 브라우저 내장
  - HTML5표준 태그를 이용한 기능이 지속적으로 추가될 예정
  - 웹 브라우즈를 애플리케이션 안에 넣고 싶은 경우에 웹뷰(WebView)를 사용, 레이아웃에서 <WebView> 태그 정의함
- **Intent 사용한 웹페이지 접속**
  - 웹페이지가 전체화면을 차지

1. 웹브라우저 사용하기

- **WebView 정의와 퍼미션 설정**
  - XML 레이아웃 파일에 정의
  - 매니페스트에 INTERNET 퍼미션 설정

```
<WebView
    ...
/>
```

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

1. 웹 브라우저 사용하기

- **WebView 위젯 사용**
  - xml 레이아웃 파일에 WebView 정의
  - 객체참조

```
webview = (WebView) findViewById(R.id.webview);
```

  - WebView 설정

```
WebSettings webSettings = webview.getSettings();
webSettings.setJavaScriptEnabled(true);
```

  - 페이지 로딩

```
webview.loadUrl("http://m.naver.com");
```

  - 매니페스트에 인터넷 사용 퍼미션 지정

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

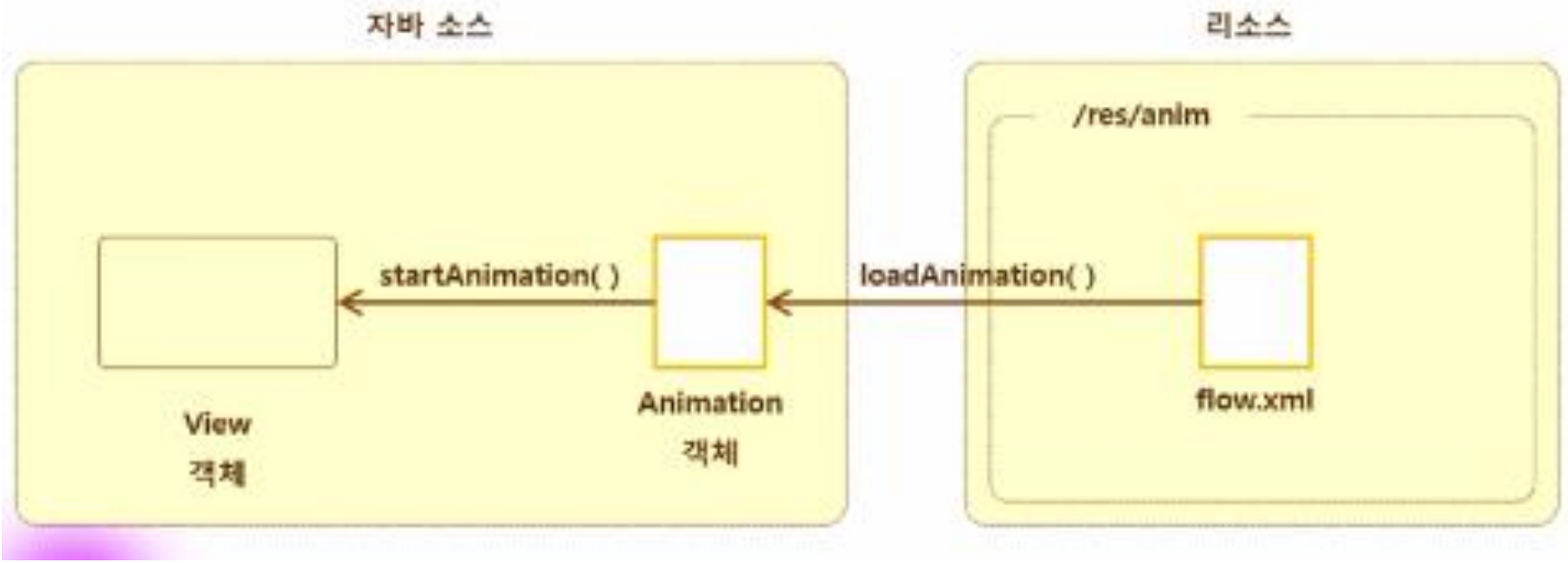
1. 웹브라우저 사용하기

- **실습**
  - 웹 페이지 띄우기(http://m.naver.com)
  - 앱페이지에 HTML 파일 포함하기(p320)
    - assets 폴더 작성
    - HTML5 파일 작성(assets 폴더에 작성)

2. 애니메이션 사용하기

애니메이션 사용 방식

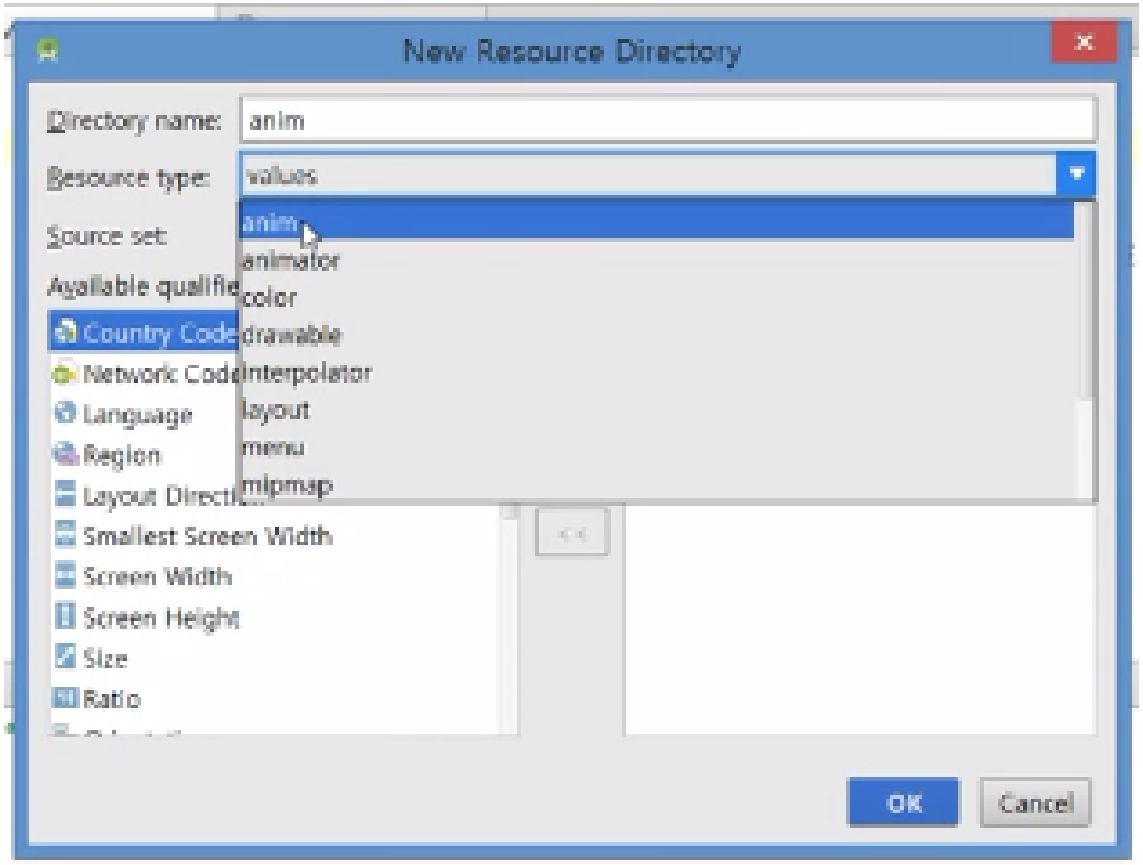
- 전형적인 애니메이션 사용방식은 애니메이션 액션 정보를 xml 파일로 정의한 후 사용
- Animation 객체로 만든 후 뷰의 startAnimation() 메소드를 사용하여 간단한 애니메이션 동작



2. 애니메이션 사용하기

트윈 애니메이션 절차

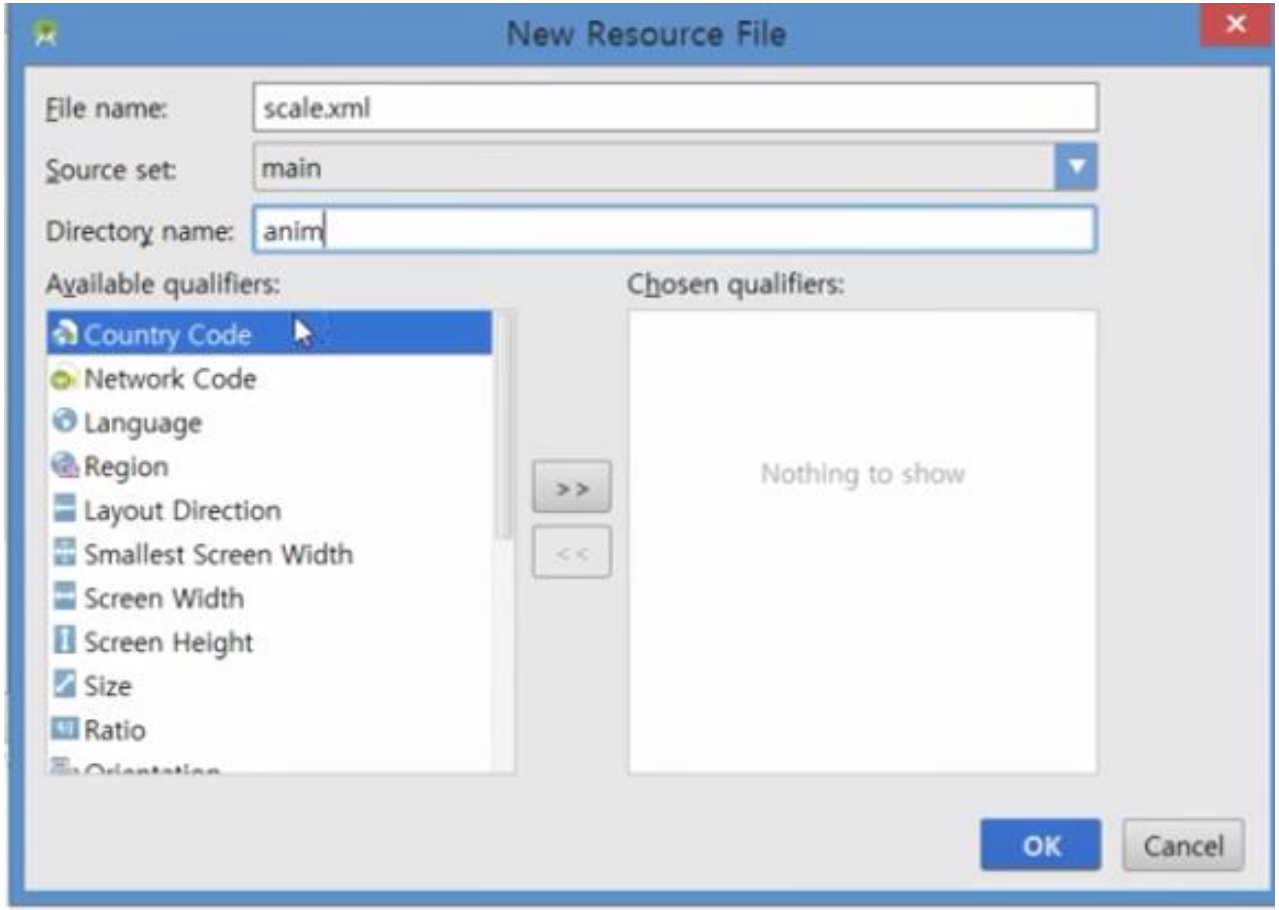
- res에 anim 폴더 작성
  - res 선택-> new Resource directory 선택



2. 애니메이션 사용하기

트윈 애니메이션 작성 절차

- anim 폴더에 트윈 애니메이션 xml 파일(translate.xml) 작성
  - anim 폴터-> new -> Animation Resource File 선택



2. 애니메이션 사용하기

- translate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="100%p"
        android:toXDelta="0%p"
        android:duration="6000"
        android:repeatCount="3"/>

    <alpha
        android:fromAlpha="0.5"
        android:toAlpha="1"
        android:duration="6000"
        android:repeatCount="3"
        />
</set>
```

2. 애니메이션 사용하기

- anim 폴더에 트윈 애니메이션 xml 파일(translate.xml) 작성
- 애니메이션 xml 파일 로드

```
Animation translate=AnimationUtils.loadAnimation(this, R.id.translate);
```

- 애니메이션 시작

```
startAnimation(translate);
```

- 애니메이션 리스너 설정
  - Animation 리스너의 onAnimationEnd(), onAnimationStart() 메소드에 애니메이션 시작과 종료시 해야할 일을 설정

2. 애니메이션 사용하기

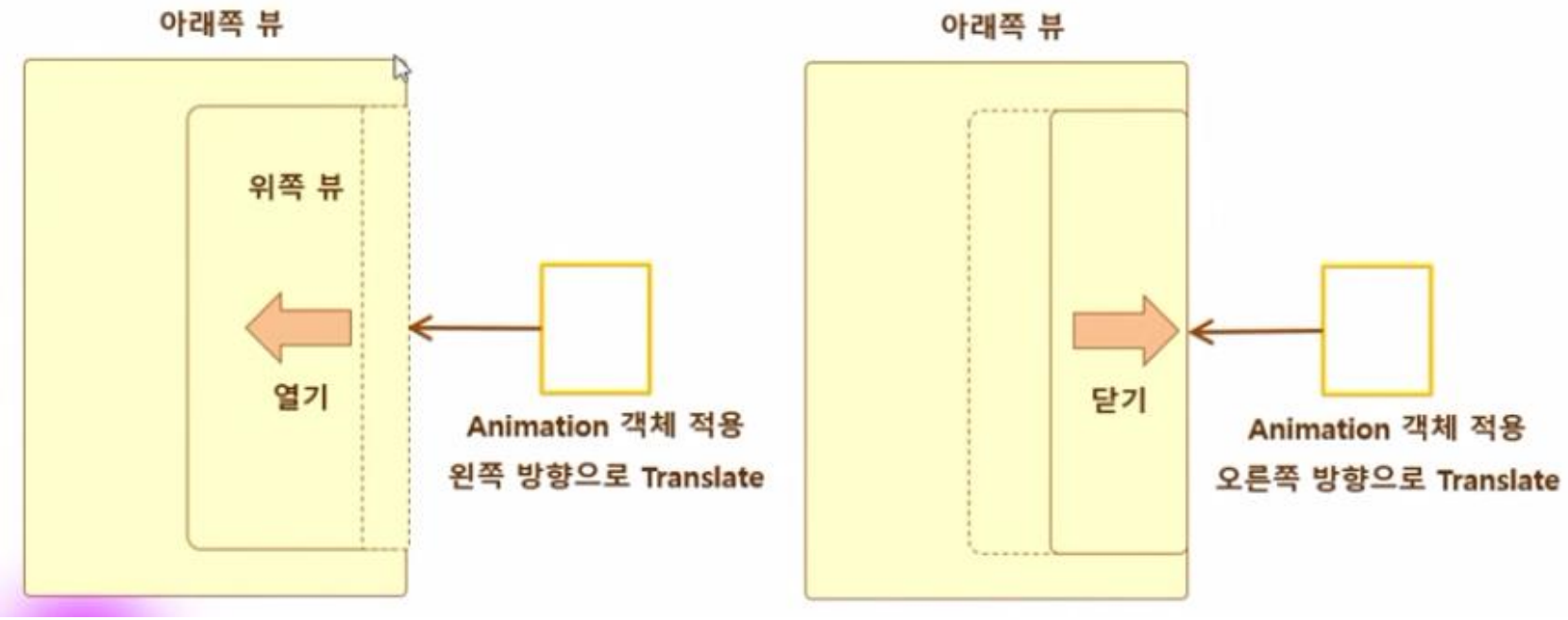
- **실습**
  - TextView 이동 및 알파, 크기 애니메이션 지정
  - 화면 전체 애니메이션
  - 애니메이션 리스너 설정하여 2개의 애니메이션이 연속으로 작동되도록 한다.

```
……
animation= AnimationUtils.loadAnimation(this,R.anim.scale);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        imageView.startAnimation(animation);
    }
});
```

3. 페이지 슬라이딩

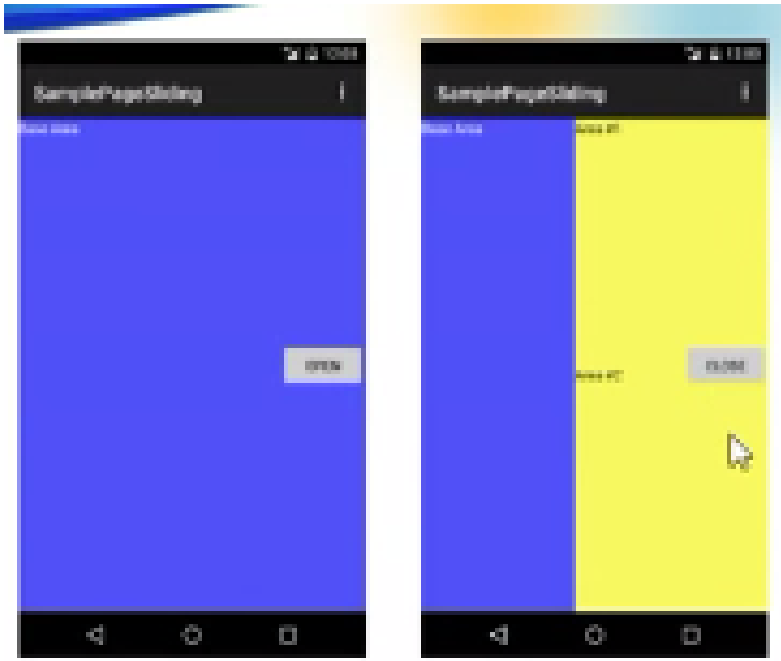
페이지 슬라이딩

- 뷰의 중첩과 애니메션을 접목한 방식
- 하나의 뷰 위에 다른 뷰가 올라가 있을 때 보이거나 감추는 과정을 애니메이션으로 적용



3. 페이지 슬라이딩

- **실습-페이지 슬라이딩 사용하기 예제**
  - 메인 액티비티의 xml 레이아웃 정의
    - 메인 액티비티 레이아웃 Frame 레이아웃 사용
    - frame 레이아웃에 3개의 레이아웃 삽입(바탕 레이아웃, 스라이딩이 보일 때 레이아웃, 버튼 포함 레이아웃)
  - res에 anim 폴더 작성
    - translate_left.xml, translate_right.xml 작성
  - 메인 액티비티 코드 작성
    - 애니메이션 기능을 사용한 슬라이딩 기능 넣기



애니메이션 파일

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="100%p"
        android:toXDelta="0%p"
        android:duration="500"
        android:repeatCount="0"
        android:fillAfter="true" >// 애니메이션 종료 후 현재 위치 유지
</set>
```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="0%p"
        android:toXDelta="100%p"
        android:duration="500"
        android:repeatCount="0"
        android:fillAfter="false" />// 애니메이션 종료 후 원래 위치로 돌아가기
</set>
```

Activity_main.xml, MainActivity

- activity_main.xml 교재 참고(p297)
- 내부 애니메이션 리스너 정의

```
private class SlidingPageAnimationListener implements Animation.AnimationListener{
    @Override
    public void onAnimationStart(Animation animation) { }
    @Override
    public void onAnimationEnd(Animation animation) {
        if(isPageOpen){
            slidingPage01.setVisibility(View.INVISIBLE);
            btn.setText("Open");
            isPageOpen=false;
        }else{
            btn.setText("Close");
            isPageOpen=true;
        }
    }
    @Override
    public void onAnimationRepeat(Animation animation) {}
}
```

애니메이션 객체 생성, 로딩, 리스너 설정

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btn=(Button)findViewById(R.id.button1);
    slidingPage01=(LinearLayout)findViewById(R.id.slidingPage01);

    translateLeftAnim= AnimationUtils.loadAnimation(this, R.anim.translate_left);
    translateRightAnim= AnimationUtils.loadAnimation(this, R.anim.translate_right);

    SlidingPageAnimationListener animationListener=new SlidingPageAnimationListener();

    translateLeftAnim.setAnimationListener(animationListener);
    translateRightAnim.setAnimationListener(animationListener);
}
```

애니메이션 시작

```
public void onButtonCliecked(View v){
    if(isPageOpen){
        slidingPage01.startAnimation(translateRightAnim);
    }else{
        slidingPage01.setVisibility(View.VISIBLE);
        slidingPage01.startAnimation(translateLeftAnim);
    }
}
```

4. 뷰플리퍼(ViewFliper)

사용 목적

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

뷰플리퍼 계층도

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

뷰플리퍼 일반적인 형태

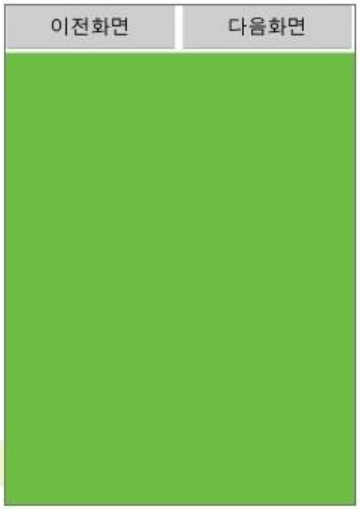
```
<리니어레이아웃>
    <리니어레이아웃>
        // 왼쪽/오른쪽으로 전환할 버튼 또는 이미지뷰
    </리니어레이아웃>
    <뷰플리퍼>

        // 여기에 한번에 하나씩 보여줄 위젯들을 넣음

    </뷰플리퍼 >
</리니어레이아웃>
```

예제

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



예제-2

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

        btnPrev = (Button) findViewById(R.id.btnPrev);
        btnNext = (Button) findViewById(R.id.btnNext);

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

        btnPrev.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                vFlipper.showPrevious();
            }
        });
        btnNext.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                vFlipper.showNext();
            }
        });
    }
```

▶ 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.

5. 프로그래스바 사용

목적

- 여러가지 화면 구성을 하고 그 안에서 프로그램 중간 중간 상태정보를 보여주는 가장 좋은 방법 중 하나

• 막대 모양

- 작업의 진행 정도를 알려줄 수 있도록 막대 모양으로 표시함
- style 속성의 값을 "?android:attr/progressBarStyleHorizontal"로 설정함

• 원 모양

- 작업이 진행 중임을 알려줌
- 원 모양으로 된 프로그레스바가 반복적으로 표시됨

5. 프로그래스바 사용

프로그래스바 사용 메소드

[Code]

void setProgress (int progress)
void incrementProgressBy (int diff)

- 진행률이 변경되면 progress 속성으로 설정되었던 값을 바꿈

[Code]

requestWindowFeature(Window.FEATURE_PROGRESS);

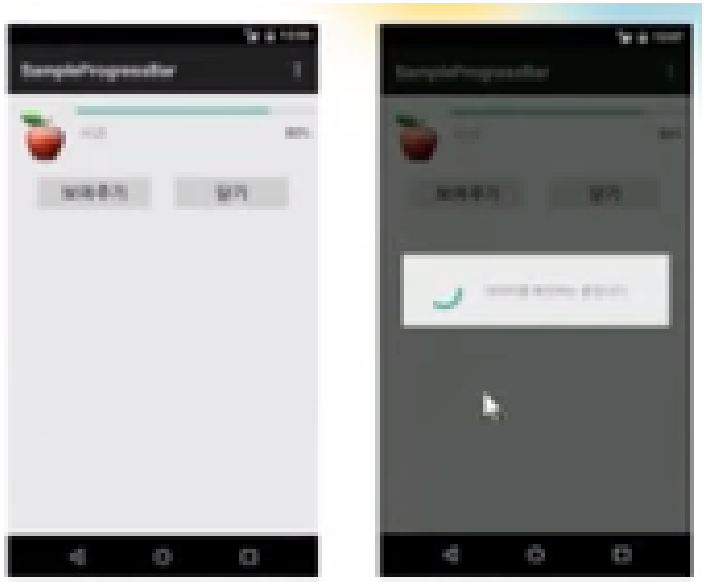
예제

메인 액티비티의
XML 레이아웃 정의

-메인 액티비티 레이아웃 정의

메인 액티비티 코드 작성

-프로그레스바 사용 코드 넣기



6. 시크바 사용

- 시크바(SeekBar)는 프로그레스바를 확장하여 만들어진 것임
- 프로그레스바의 속성을 가지고 있으면서도 사용자가 값을 조정할 수 있도록 해 줌

[Code]

void onStartTrackingTouch (SeekBar seekBar)
void onStopTrackingTouch (SeekBar seekBar)
void onProgressChanged (SeekBar seekBar, int progress,
boolean fromUser)