

SECTION 12 뷰플리퍼

뷰플리퍼(ViewFlipper)는 여러 화면을 한 화면에서 볼 수 있게 해주는 컨테이너다. 안드로이드 2.3까지는 작은 화면에서 여러 개의 화면을 효율적으로 다룰 수 있는 방법이라 많이 유용했지만 안드로이드 3.0부터는 프래그먼트(Fragment)로 인해 예전만큼 활용도가 높지는 않다.

한 화면에서 여러 화면을 이동시키기

참고 프로젝트 ViewFilpperDemoA1

뷰플리퍼(ViewFlipper)를 사용하면 한 화면에서 여러 개의 뷰를 붙이고 이를 자유롭게 이동시킬 수 있다. 이 예제는 뷰플리퍼에 추가된 RED, GREEN, BLUE 화면을 “이전”과 “다음” 버튼으로 전환하는 방법을 보여준다.

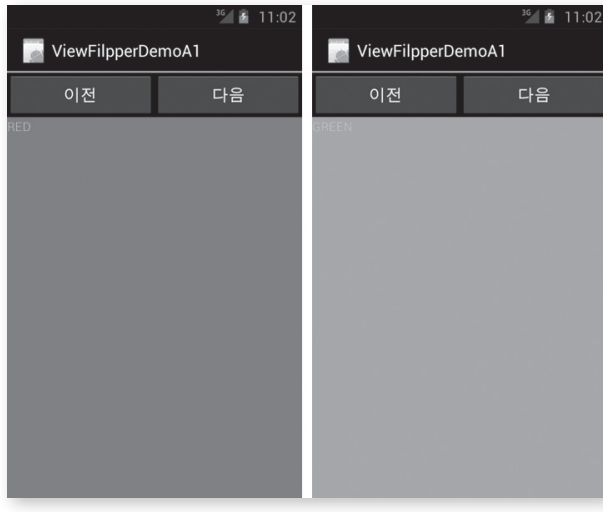


그림 4-69

이렇게 만들기 위해서는 가장 먼저 레이아웃을 구성하고 자바 코드에서 ViewFlipper의 showPrevious()와 showNext() 메소드를 사용해서 “이전”과 “다음” 버튼을 구현해야 한다.



CODE

XML - /res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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/prev"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="이전" />

        <Button
            android:id="@+id/next"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="다음" />
    </LinearLayout>

    <ViewFlipper
        android:id="@+id/flipper"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#ff0000"
            android:text="RED" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#00ff00"
            android:text="GREEN" />
    </ViewFlipper>
</LinearLayout>
```

```

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#0000ff"
            android:text="BLUE" />
    </ViewFlipper>

</LinearLayout>

```

〈ViewFlipper〉에 화면에 보여주고 싶은 뷰를 추가하면 된다. 여기서는 RED, GREEN, BLUE 색상을 배경으로 가지고 있는 〈TextView〉를 추가했다.



CODE

Main.java

```

package com.androidside.viewflipperdemoa1;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ViewFlipper;

public class Main extends Activity implements View.OnClickListener {
    Button prev;
    Button next;
    ViewFlipper flipper;

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

        flipper = (ViewFlipper) findViewById(R.id.flipper);
        prev = (Button) findViewById(R.id.prev);
        next = (Button) findViewById(R.id.next);

        prev.setOnClickListener(this);
        next.setOnClickListener(this);
    }
}

```

```

@Override
public void onClick(View v) {
    // prev 버튼이 클릭되었을 때
    if (v == prev) {
        flipper.showPrevious();

        // next 버튼이 클릭되었을 때
    } else if (v == next) {
        flipper.showNext();
    }
}
}

```

ViewFlipper 클래스의 showPrevious() 메소드를 호출하면 현재 보여지고 있는 ViewFlipper의 이전 뷰를 보여주며, showNext() 메소드를 호출하면 다음 뷰를 보여준다.

자동으로 화면 이동하기

참고 프로젝트 ViewFilpperDemoB1

뷰플리퍼를 사용하면 여러 화면을 한 화면에서 하나씩 보여줄 수 있다. 또한 뷰플리퍼에 추가된 여러 화면을 자동으로 보여주게 만들 수도 있다.

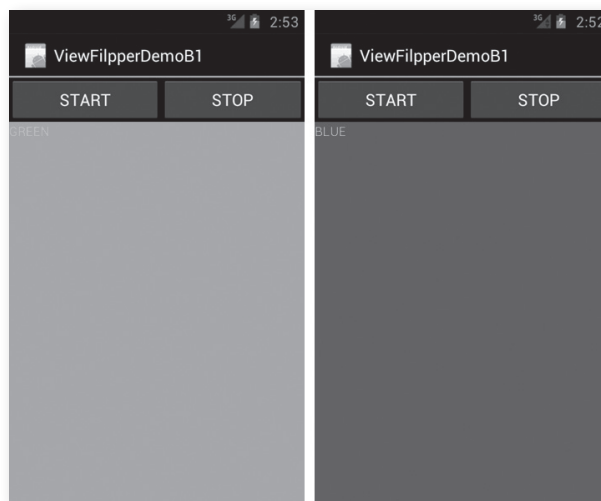


그림 4-70

이 예제는 “START” 버튼을 클릭하면 ViewPager에 추가된 RED, GREEN, BLUE 색상을 가진 TextView를 자동으로 반복하면서 보여주고, “STOP” 버튼을 클릭하면 자동 반복을 멈춘다. 이렇게 만들기 위해서는 ViewPager에서 startFlipping() 메소드를 호출해서 반복을 시작하고 stopFlipping() 메소드를 호출해서 반복을 멈추면 된다.



CODE

레이아웃 XML - /res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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/start"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="START" />

        <Button
            android:id="@+id/stop"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="STOP" />
    </LinearLayout>

    <ViewPager
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#ff0000"
            android:text="RED" />
    </ViewPager>
</LinearLayout>
```

```

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#00ff00"
            android:text="GREEN" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#0000ff"
            android:text="BLUE" />
    </ViewFlipper>

</LinearLayout>

```

〈ViewFlipper〉에 화면에 보여주고 싶은 뷰를 추가하면 된다. 여기서는 RED, GREEN, BLUE 색상을 배경으로 가지고 있는 〈TextView〉를 추가했다.



CODE

Main.java

```

package com.androidside.viewflipperdemob1;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ViewFlipper;

public class Main extends Activity implements View.OnClickListener {
    Button start;
    Button stop;
    ViewFlipper flipper;

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

        flipper = (ViewFlipper) findViewById(R.id.flipper);
        start = (Button) findViewById(R.id.start);
    }
}

```

```

        stop = (Button) findViewById(R.id.stop);

        start.setOnClickListener(this);
        stop.setOnClickListener(this);

        // 플리핑 간격을 1초로 한다.
        flipper.setFlipInterval(1000);
    }

    @Override
    public void onClick(View v) {
        // start 버튼이 클릭되었을 때 플리핑을 시작한다.
        if (v == start) {
            flipper.startFlipping();

            // stop 버튼이 클릭되었을 때 플리핑을 멈춘다.
        } else if (v == stop) {
            flipper.stopFlipping();
        }
    }
}

```

setFlipInterval() 메소드로 보여줄 뷰의 간격을 설정할 수 있다. 또한 setAutoStart() 메소드를 사용하면 뷰가 반복하는 것을 자동으로 시작하게 할 수도 있다.



METHOD SUMMARY

android.widget.ViewFlipper 클래스의 메소드

boolean	isAutoStart() 자동으로 시작하게 설정되었다면 true를 반환한다.
boolean	isFlipping() 현재 플리핑 중이라면 true를 반환한다.
void	setAutoStart(boolean autoStart) 자동 시작 여부를 설정한다.
void	setFlipInterval(int milliseconds) 플리핑 간격을 설정한다.
void	startFlipping() 플리핑을 시작한다.
void	stopFlipping() 플리핑을 중단한다.

만약에 전환하는 뷰에 애니메이션을 추가하여 좀 더 색다르게 보이고 싶다면 다음과 같은 코드를 적용하면 된다.

```
flipper.setInAnimation(AnimationUtils.loadAnimation(this, R.anim.left_in));  
flipper.setOutAnimation(AnimationUtils.loadAnimation(this, R.anim.left_out));
```

이 코드는 뷰가 보여질 때 애니메이션을 적용하는 것이며, 애니메이션 효과는 원하는 대로 작성할 수 있다. 이 코드에서 선언한 애니메이션 파일은 `/res/anim/left_in.xml`과 `/res/anim/left_out.xml`이다. 관련 코드를 살펴보고 싶다면 별도로 제공하는 `ViewFlipperDemoB2` 프로젝트를 참고하기 바란다.