**버튼 만들기**

2017-08-30

이승진

**학습목표**

여러가지 형태의 버튼 만들기

Google Material Icon을 활용

버튼에 비트맵 아이콘 표시하기

여러 버튼이 클릭 리스너 객체 하나를 공유하기

**목차**

[1. 아이콘 2](#_Toc491880355)

[1) Google Material Design 2](#_Toc491880356)

[2) 아이콘 파일 복사 3](#_Toc491880357)

[2. Buttons 화면 추가 4](#_Toc491880358)

[1) 새 액티비티 생성 4](#_Toc491880359)

[2) Buttons 화면 레이아웃 5](#_Toc491880360)

[3. Buttons 화면 Java 코드 10](#_Toc491880361)

[1) ButtonsActivity.java 10](#_Toc491880362)

[2) 메뉴 추가 12](#_Toc491880363)

[3) 화면 전환 기능 구현 14](#_Toc491880364)

[4) 화면 제목 수정 15](#_Toc491880365)

[5) 결과 소스코드 16](#_Toc491880366)

# 아이콘

## Google Material Design

최근 구글이 새 디자인 기술을 발표했다.

Google Material Design 이 그 이름이다. (https://material.google.com/)

웹 디자인과 앱 디자인 양쪽에 Google Material Design 기술을 활용할 수 있다.

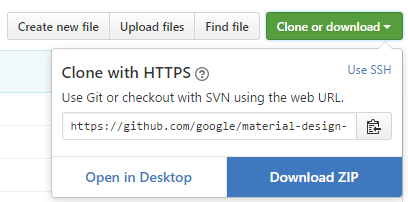
Google Material Design 기술에는 아이콘(icon)도 포함되어 있다.

Google Material Design이 제공하는 아이콘 목록을

① https://design.google.com/icons 페이지에서 볼 수 있고,

② https://github.com/google/material-design-icons 페이지에서 다운로드 할 수 있다.

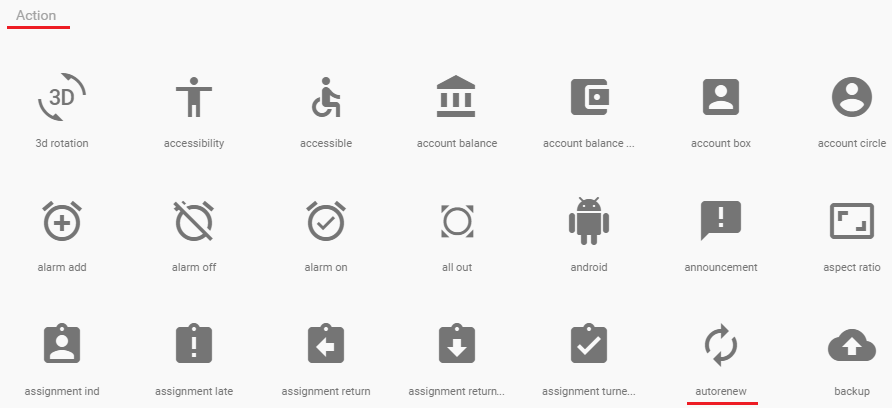
② 페이지에서 Clone or download - Download ZIP 버튼을 클릭하여, 전체를 다운로드 받자.



원하는 모양의 아이콘을 ① 페이지에서 찾자.

① 페이지에서 선택한 아이콘의 그룹 이름과 아이콘 이름을 기억하자.

다운로드 받은 압축 파일 내부에서 폴더명이 아이콘 그룹 이름이고, png 파일명이 아이콘 이름이다.



예를 들어 Action 그룹에 autorenew 아이콘이 있다.

다운로드 받은 압축 파일 내부에서, action/drawable-#### 폴더 아래에

ic\_autorenew\_black\_##dp.png, ic\_autorenew\_white\_##dp.png 파일들을 찾을 수 있을 것이다.

drawable-#### 폴더명에서 #### 부분은 스마트폰 화면의 해상도를 의미한다.

스마트폰 해상도별로 적당한 크기의 아이콘 파일이 별도로 제공된다.

ic\_autorenew\_black\_##dp.png 아이콘 파일명에서 ## 부분은 아이콘의 크기를 의미한다.

아이콘 파일명에서 black, white는 아이콘 색을 의미한다.

어두운 배경에는 white 아이콘을, 밝은 배경에는 black 아이콘을 사용해야 한다.

## 아이콘 파일 복사

다운로드 받은 압축 파일 내부의 action/drawable-**hdpi**/ic\_autorenew\_black\_##dp.png 파일들을

프로젝트 폴더 아래의 app/src/main/res/mipmap-**hdpi** 폴더 아래에 복사한다.

다운로드 받은 압축 파일 내부의 action/drawable-**mdpi**/ic\_autorenew\_black\_##dp.png 파일들을

프로젝트 폴더 아래의 app/src/main/res/mipmap-**mhdpi** 폴더 아래에 복사한다.

다운로드 받은 압축 파일 내부의 action/drawable-**xhdpi**/ic\_autorenew\_black\_##dp.png 파일들을

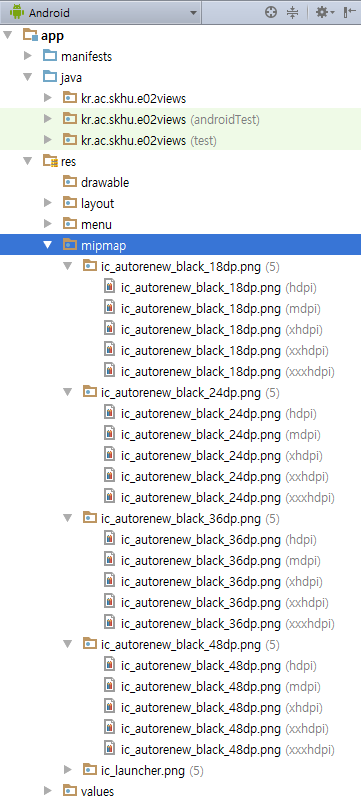
프로젝트 폴더 아래의 app/src/main/res/mipmap-**xhdpi** 폴더 아래에 복사한다.

다운로드 받은 압축 파일 내부의 action/drawable-**xxhdpi**/ic\_autorenew\_black\_##dp.png 파일들을

프로젝트 폴더 아래의 app/src/main/res/mipmap-**xxhdpi** 폴더 아래에 복사한다.

다운로드 받은 압축 파일 내부의 action/drawable-**xxxhdpi**/ic\_autorenew\_black\_##dp.png 파일들을

프로젝트 폴더 아래의 app/src/main/res/mipmap-**xxxhdpi** 폴더 아래에 복사한다.



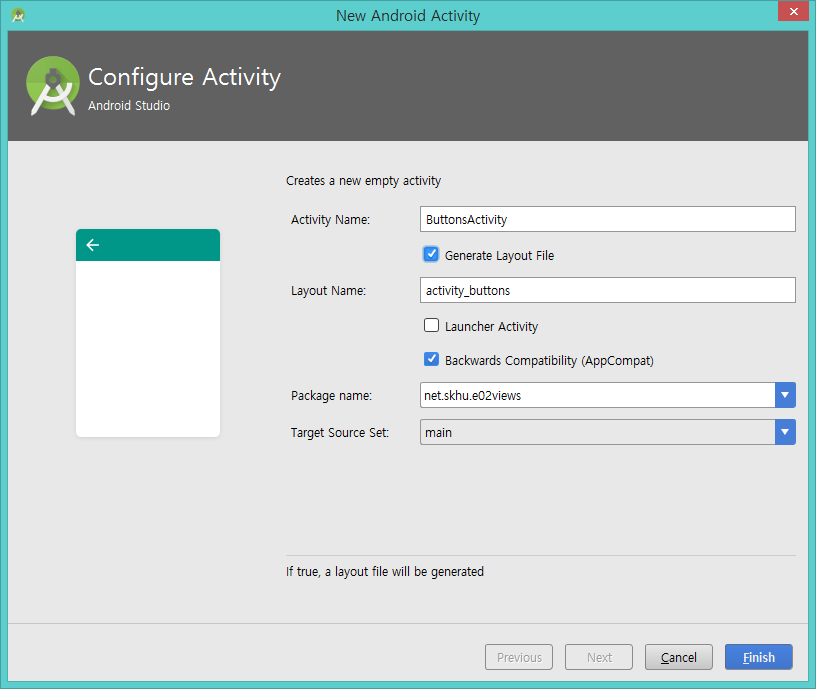
# Buttons 화면 추가

## 새 액티비티 생성

E02View 프로젝트에 새 액티비티를 추가하자.

Project 창에서 app 폴더를 마우스 오른쪽 버튼으로 클릭하고

메뉴에서 New - Activity - Empty Activity 클릭



**파일 생성**

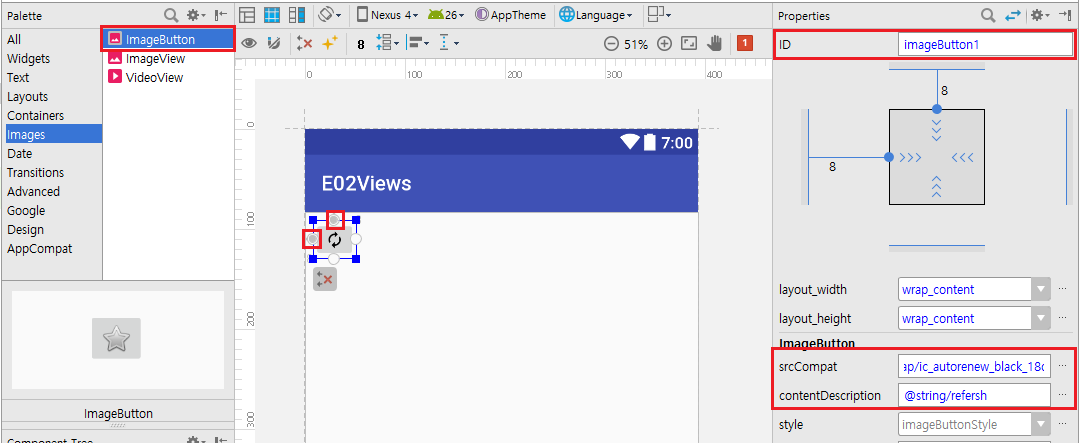
Activity Name: ButtonsActivity.java

Layout Name: activity\_buttons.xml

## Buttons 화면 레이아웃

activity\_button.xml 파일 수정

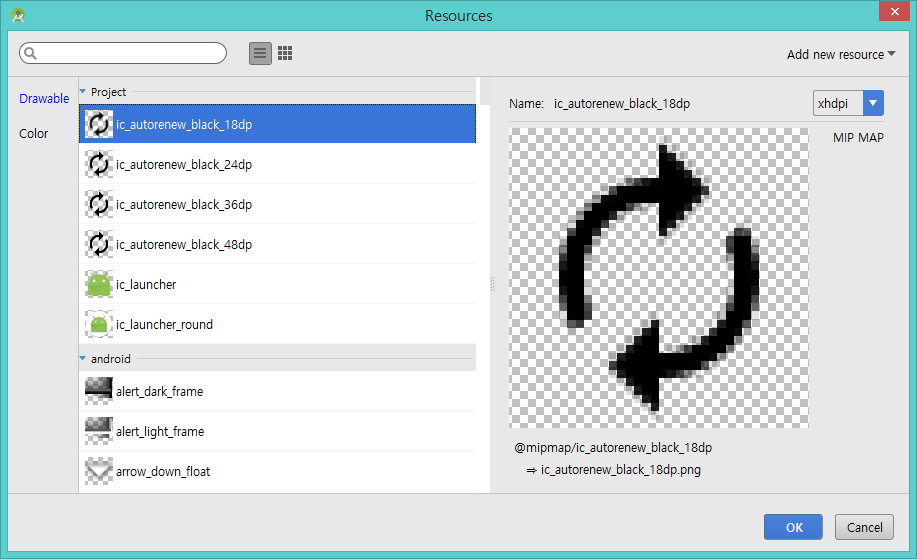
### ImageButton 추가



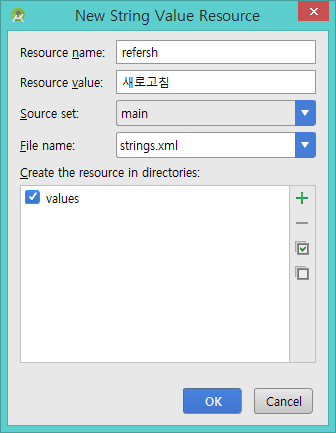
화면 왼쪽 Palette 창에서 ImageButton 항목을 액티비티 화면에 드래그 드롭하여 ImageButton를 추가한다.

ic\_autorenew\_black\_18dp 이미지를 선택한다.

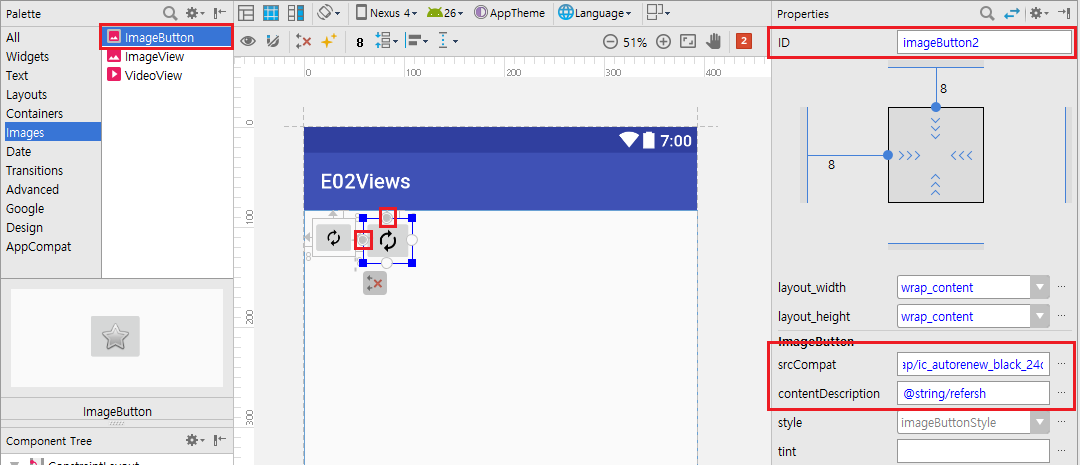
이것은 18dp 크기의 autorenew 아이콘 리소스 파일이다.



contentDescription 속성은 시각 장애인들을 위해서 화면의 내용을 소리로 표현할 때 ImageButton을 설명하는 문구이다. contentDescription이 필수 속성은 아니고 권장 속성이다.

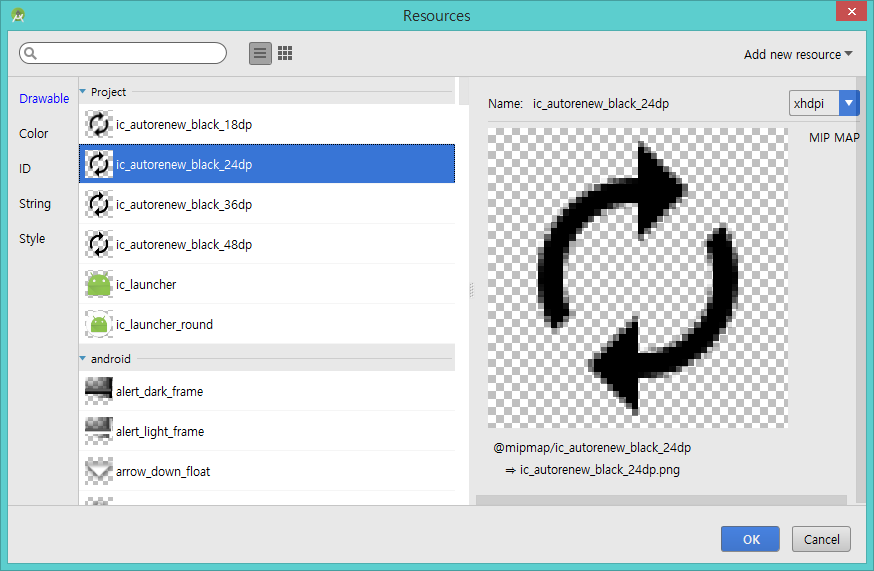


### ImageButton 하나 더 추가

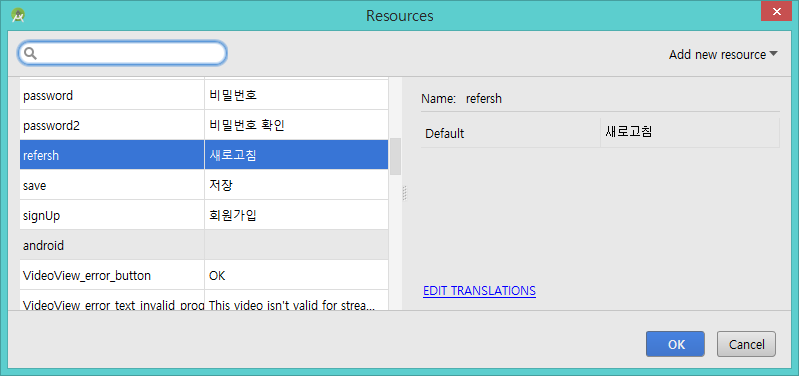


화면 왼쪽 Palette 창에서 ImageButton 항목을 액티비티 화면에 드래그 드롭하여 ImageButton를 추가한다.

24dp 크기의 autorenew 아이콘 리소스 파일을 선택한다.



contentDescription 속성값으로 @string/refersh 문자열 리소스를 선택한다.

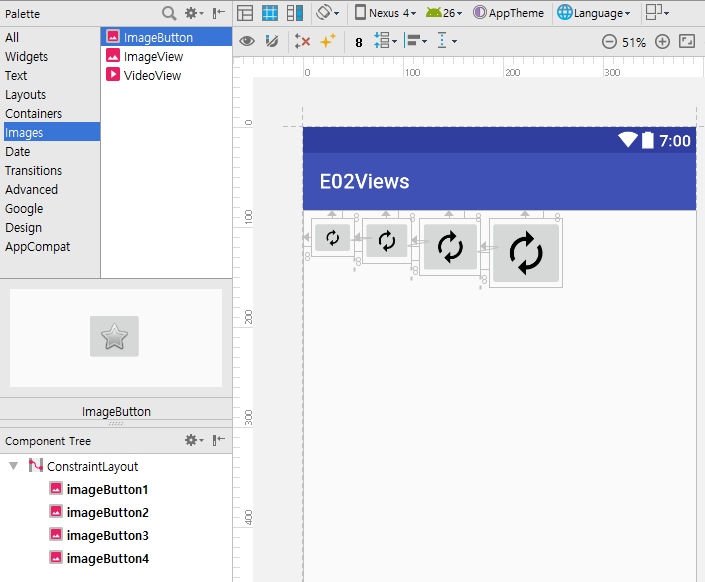


### ImageButton 2개 더 추가

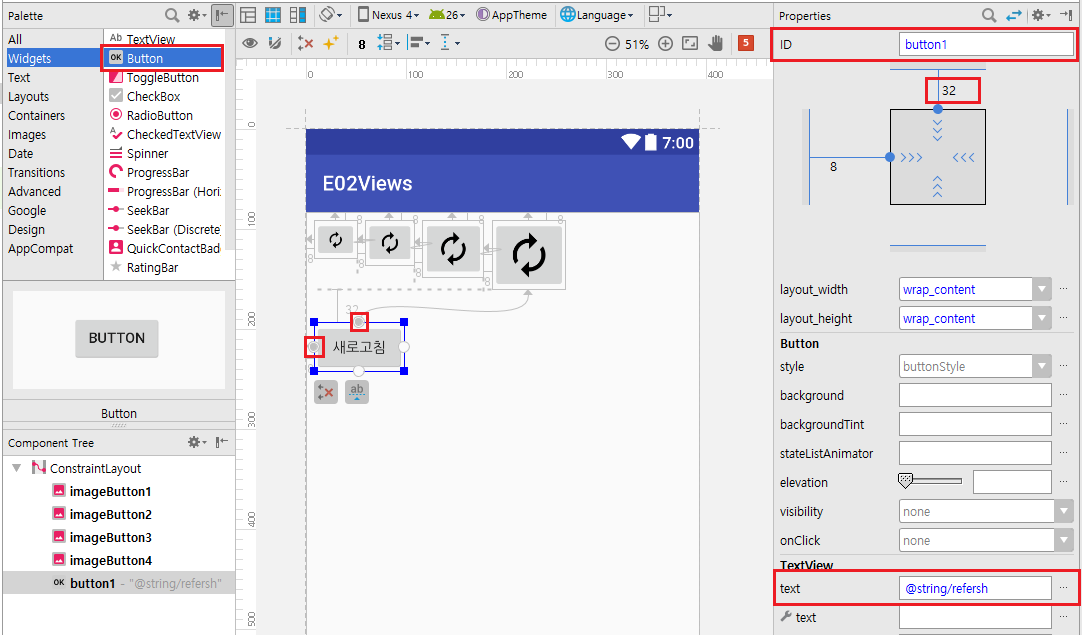
위와 동일한 방법으로 ImageButton을 두 개 더 추가하자.

|  |  |
| --- | --- |
| 속성 | 값 |
| id | imageButton**3** |
| src | @mipmap/ic\_autorenew\_black\_**36**dp |
| contentDescription | @string/refresh |

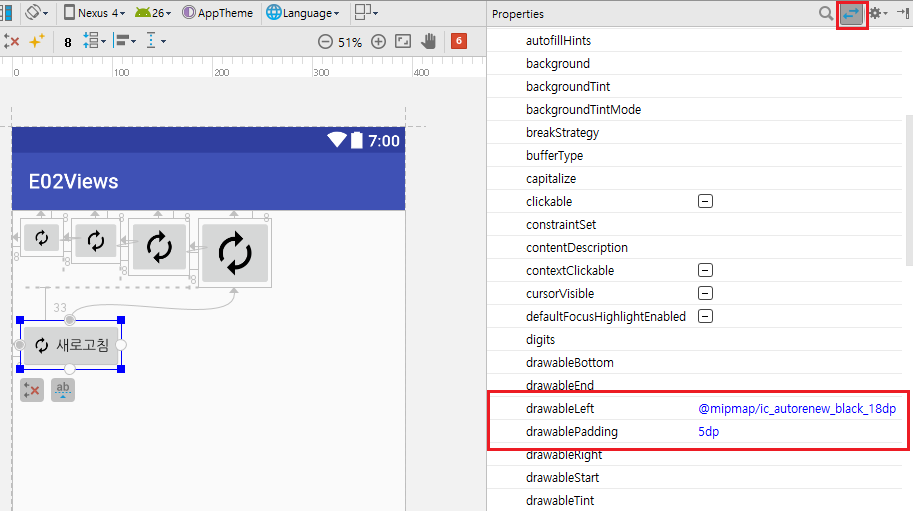
|  |  |
| --- | --- |
| 속성 | 값 |
| id | imageButton**4** |
| src | @mipmap/ic\_autorenew\_black\_**48**dp |
| contentDescription | @string/refresh |



### Button 추가



화면 왼쪽 Palette 창에서 Button 항목을 액티비티 화면에 드래그 드롭하여 Button를 추가한다.



drawableLeft 속성은 버튼의 왼쪽에 표시할 비트맵을 지정하는 속성이다.

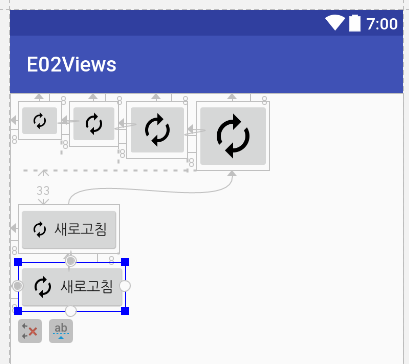
18dp 크기의 autorenew 아이콘 리소스 파일을 선택한다.

@mipmap/ic\_autorenew\_black\_18dp

drawablePadding 속성은 비트맵과 문자열 사이의 여백을 지정하는 속성이다.

### Button 하나 더 추가

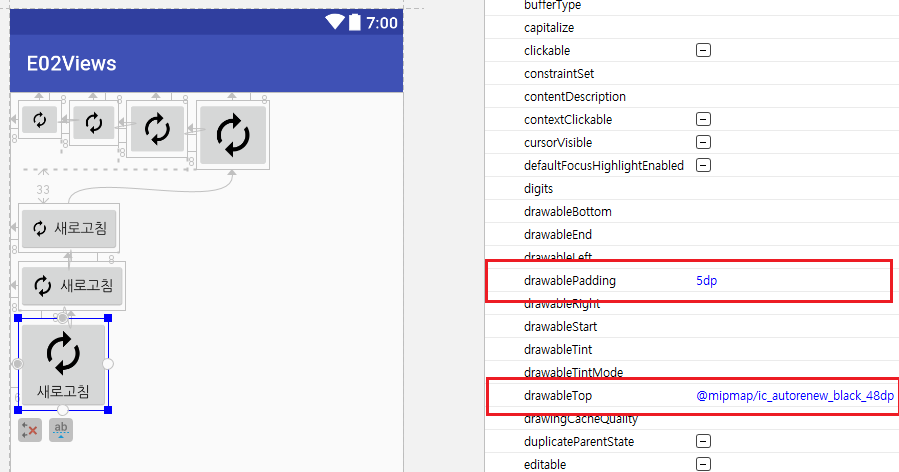
위와 동일한 방법으로 버튼을 하나 더 추가하자.



|  |  |
| --- | --- |
| 속성 | 값 |
| id | Button**2** |
| text | @string/refresh |
| drawableLeft | @mipmap/ic\_autorenew\_black\_**24**dp |
| drawablePadding | 5dp |

### Button 하나 더 추가

위와 동일한 방법으로 버튼을 하나 더 추가하자.



|  |  |
| --- | --- |
| 속성 | 값 |
| id | Button**3** |
| text | @string/refresh |
| drawable**Top** | @mipmap/ic\_autorenew\_black\_**48**dp |
| drawablePadding | 5dp |

drawableTop 속성은 버튼의 위쪽에 표시할 비트앱을 지정하는 속성이다.

# Buttons 화면 Java 코드

## ButtonsActivity.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | package net.skhu.e02views;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.ImageButton;  import android.widget.Toast;  public class ButtonsActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_buttons*);  View.OnClickListener listener = new View.OnClickListener() {  @Override  public void onClick(View view) {  String msg;  int buttonId = view.getId();  switch (buttonId) {  case R.id.*button1*: msg = "첫 번째 버튼이 클릭 되었습니다."; break;  case R.id.*button2*: msg = "두 번째 버튼이 클릭 되었습니다."; break;  case R.id.*button3*: msg = "세 번째 버튼이 클릭 되었습니다."; break;  case R.id.*imageButton1*: msg = "첫 번째 이미지 버튼이 클릭 되었습니다."; break;  case R.id.*imageButton2*: msg = "두 번째 이미지 버튼이 클릭 되었습니다."; break;  case R.id.*imageButton3*: msg = "세 번째 이미지 버튼이 클릭 되었습니다."; break;  case R.id.*imageButton4*: msg = "네 번째 이미지 버튼이 클릭 되었습니다."; break;  default: msg = "알 수 없는 버튼이 클릭 되었습니다."; break;  }  Toast.*makeText*(ButtonsActivity.this, msg, Toast.*LENGTH\_SHORT*).show();  }  };  Button button1 = (Button)findViewById(R.id.*button1*);  Button button2 = (Button)findViewById(R.id.*button2*);  Button button3 = (Button)findViewById(R.id.*button3*);  button1.setOnClickListener(listener);  button2.setOnClickListener(listener);  button3.setOnClickListener(listener);  ImageButton imageButton1 = (ImageButton)findViewById(R.id.*imageButton1*);  ImageButton imageButton2 = (ImageButton)findViewById(R.id.*imageButton2*);  ImageButton imageButton3 = (ImageButton)findViewById(R.id.*imageButton3*);  ImageButton imageButton4 = (ImageButton)findViewById(R.id.*imageButton4*);  imageButton1.setOnClickListener(listener);  imageButton2.setOnClickListener(listener);  imageButton3.setOnClickListener(listener);  imageButton4.setOnClickListener(listener);  }  } |

버튼이 클릭되면 자동으로 호출되는 리스너 객체를 버튼마다 하나씩 따로 만들어 주어야만 하는 것은 아니다.

위 코드는 리스너 객체를 하나만 만들어서, 모든 버튼에 그 리스너 객체를 등록하였다.

(줄17~34) anonymous inner class 문법으로 View.OnClickListener의 자식 클래스 객체를 하나 생성하였다.

생성된 객체에 대한 참조를 listener 참조 변수에 대입하였다.

listener 참조 변수의 타입은 View.OnClickListener 클래스 타입이다.

(줄38~40) listener 참조 변수가 가르키는 리스너 객체를, 3개의 Button 객체에 등록하였다.

(줄46~49) listener 참조 변수가 가르키는 리스너 객체를, 4개의 ImageButton 객체에 등록하였다.

(줄19)

3개의 Button 객체와 4개의 ImageButton 객체 중 아무거나 클릭하면 줄19의 onClick 메소드가 호출된다.

onClick 메소드의 파라미터로 전달되는 것은 클릭된 버튼 객체이다.

onClick 메소드의 파라미터 변수의 타입은 View 클래스이다.

View 클래스는 Button 클래스와 ImageButton 클래스의 부모 클래스이다.

부모 클래스 타입의 참조 변수가, 자식 클래스 객체를 참조할 수 있다. (up-casting)

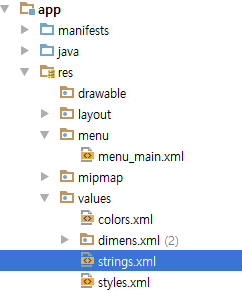
(줄21) int buttonId = view.getId();

클릭된 버튼 객체의 id 값을 얻는다.

이 id 값은, R.id.button1, R.id.button2, R.id.button3, R.id.imageButton1, R.id.imageButton2, R.id.imageButton3, R.id.imageButton4 값 중에서 하나일 것이다.

## 메뉴 추가

### 문자열 리소스 추가

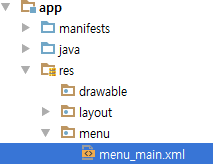


메뉴에 표시할 문자열을, 문자열 리소스 파일인 strings.xml 파일에 추가하자.

**res/values/strings.xml 수정**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <resources>  <string name="app\_name">E02Views</string>  <string name="action\_signUp">회원가입</string>  <string name="action\_memo">메모장</string>  <string name="action\_buttons">버튼</string>  <string name="loginId">로그인 아이디</string>  <string name="input\_loginId">로그인 아이디를 입력하세요.</string>  <string name="password">비밀번호</string>  <string name="input\_password">비밀번호를 입력하세요.</string>  <string name="password2">비밀번호 확인</string>  <string name="input\_password2">비밀번호를 한 번 더 입력하세요.</string>  <string name="email">이메일</string>  <string name="input\_email">이메일 주소를 입력하세요.</string>  <string name="signUp">회원가입</string>  <string name="input\_title">제목을 입력하세요.</string>  <string name="input\_content">내용을 입력하세요.</string>  <string name="save">저장</string>  <string name="refersh">새로고침</string>  </resources> |

### 메뉴 항목 추가



새 메뉴 항목을, MainActivity의 메뉴 리소스 파일인 res/menu/menu\_main.xml 파일을 수정하자.

**res/menu/menu\_main.xml 수정**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | *<?*xml version="1.0" encoding="utf-8"*?>*  <menu xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto">  <item android:id="@+id/action\_signUp"  android:title="@string/action\_signUp" app:showAsAction="never" />  <item android:id="@+id/action\_memo"  android:title="@string/action\_memo" app:showAsAction="never" />  <item android:id="@+id/action\_buttons"  android:title="@string/action\_buttons" app:showAsAction="never" />  </menu> |

(줄9~10) 새 메뉴 항목을 추가하였다.

|  |  |
| --- | --- |
| 애트리뷰트 | 설명 |
| id | 메뉴 항목의 id |
| title | 메뉴 항목에 표시할 텍스트의 문자열 리소스 id |
| showAsAction | 메뉴 항목을 어떻게 표시할지 지정하는 옵션 |

XML 파일에서 문자열 리소스 id : @string/action\_buttons

XML 파일에서 메뉴의 id : @+id/action\_buttons

java 파일에서 문자열 리소스 id : R.string.action\_buttons

java 파일에서 메뉴의 id : R.id.action\_buttons

showAsAction 옵션 값이 "never" 이면, 처음에 메뉴는  아이콘으로만 표시되고,

 아이콘을 클릭해야 메뉴 항목 전체가 나타난다.

안드로이드 화면 상단에 메뉴가 표시되는 제목 줄을 액션바(actionbar)라고 부른다.

액션바에 표시되는  아이콘을, 액션바 오버플로우 아이콘이라고 부른다.

그런데 구형 스마트폰에서는  아이콘이 화면에 보이지 않을 수도 있다.

스마트폰에 백(back) 버튼과 메뉴 버튼이 달려 있는 구형 스마트폰의 경우에는

 액션바 오버플로우 아이콘이 화면에 표시되지 않는다.

메뉴가 나타나게 하려면 스마트폰에 달려있는 메뉴 버튼을 눌러야 한다.

## 화면 전환 기능 구현

MainActivty 화면에서 메모장 메뉴를 클릭하면 ButtonActivity 화면으로 넘어가는 기능을 구현하자.

**MainActivity.java 수정**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | package net.skhu.e02views;  import android.content.Intent;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.Menu;  import android.view.MenuItem;  import android.widget.Toast;  public class MainActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  }  @Override  public boolean onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.*menu\_main*, menu);  return true;  }  @Override public boolean onOptionsItemSelected(MenuItem item) {  int id = item.getItemId();  if (id == R.id.*action\_signUp*) {  Intent intent = new Intent(this, SignupActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*action\_memo*) {  Intent intent = new Intent(this, MemoActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*action\_buttons*) {  Intent intent = new Intent(this, ButtonsActivity.class);  startActivity(intent);  return true;  }  return super.onOptionsItemSelected(item);  }  } |

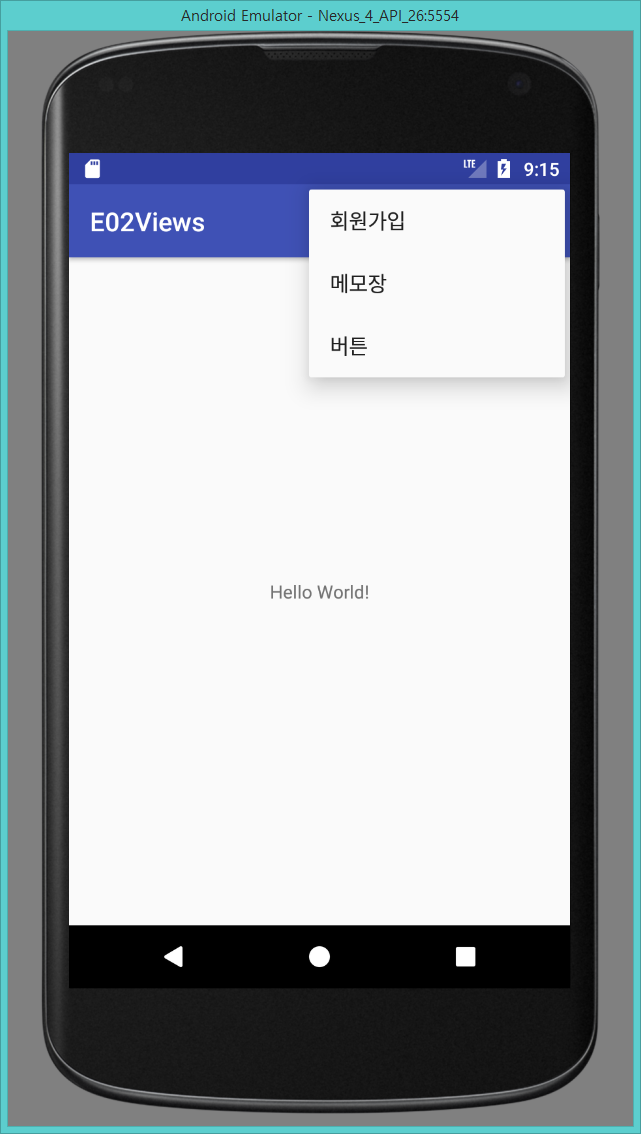
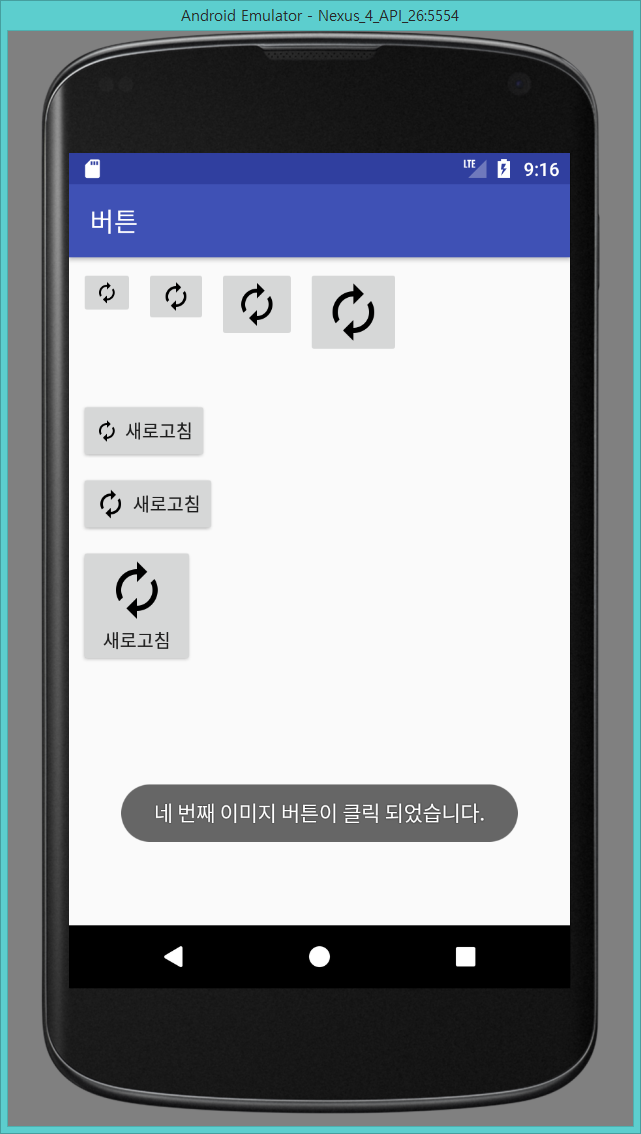
## 화면 제목 수정

**manifests/AndroidManifest.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | *<?*xml version="1.0" encoding="utf-8"*?>*  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="net.skhu.e02views">  <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round"  android:supportsRtl="true"  android:theme="@style/AppTheme">  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  <activity  android:name=".SignupActivity"  android:label="@string/action\_signUp" />  <activity  android:name=".MemoActivity"  android:label="@string/action\_memo" />  <activity  android:name=".ButtonsActivity"  android:label="@string/action\_buttons" />  </application>  </manifest> |

## 결과 소스코드

E02Views\_src3.zip

MainActivity 화면에서  아이콘을 누르면 메뉴가 보인다.