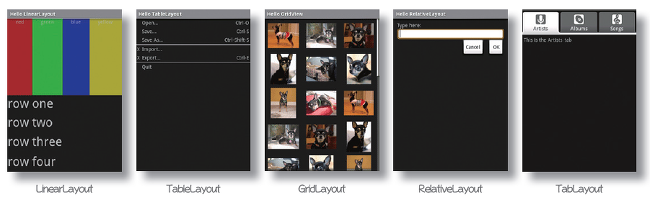
**[안드로이드 레이아웃(layout)!](http://colomy.tistory.com/39)**

POSTED AT 2014.04.20 21:11 | POSTED IN [프로그래밍/ANDROID](http://colomy.tistory.com/category/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D/Android)



위의 사진처럼 레이아웃 종류는 여러가지가 있지요~

레이아웃 각각에 대해서 살짝 알아보는 시간을 가져봤어요ㅎㅎㅎ

**레이아웃(layout)**

절대적인 화면 위치 지정보다 상대적으로 뷰를 배치하는 것이 바람직하여

안드로이드에서는 레이아웃 클래스를 사용해 뷰들을 배치합니다

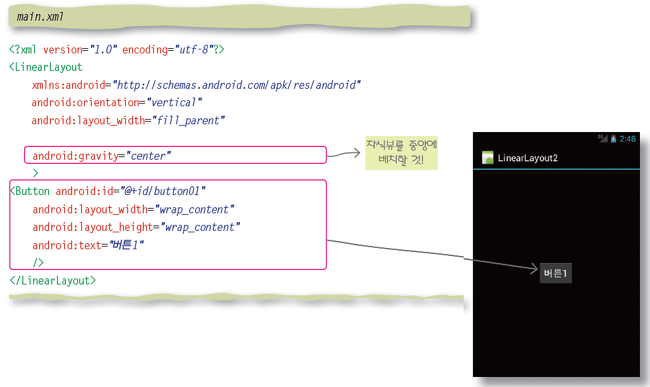
**레이아웃 클래스**

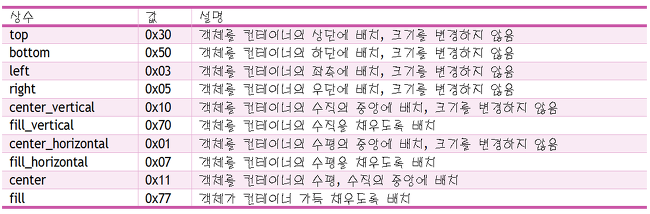
|  |  |
| --- | --- |
| 구 분 | 설 명 |
| LinearLayout | 자식들을 수직이나 수평으로 배치 |
| TableLayou | 자식들을 테이블 형태로 배치 |
| GridLayoutt | 자식들을 바둑판 모양으로 배치 |
| RelativeLayout | 자식들을 부모나 다른 자식에 상대적으로 배치 |
| TabLayout | 탭을 이용해 겹쳐진 자식 중 하나를 선택 |
| AbsoluteLayout | 절대 위치로 배치 |
| FrameLayout | 모든 자식들을 좌측 상단에 겹치게 배치 |

**레이아웃 속성 값들**



**레이아웃 그래비티 속성**

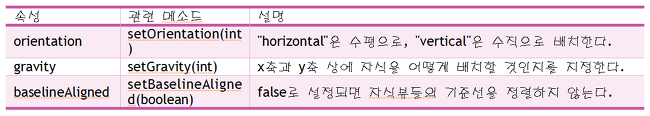




**선형 레이아웃(linear layout)**

기본적인 배치 관리자,

**선형 레이아웃 속성**



setOrientation(int) : 수평이나 수직으로 배치가능

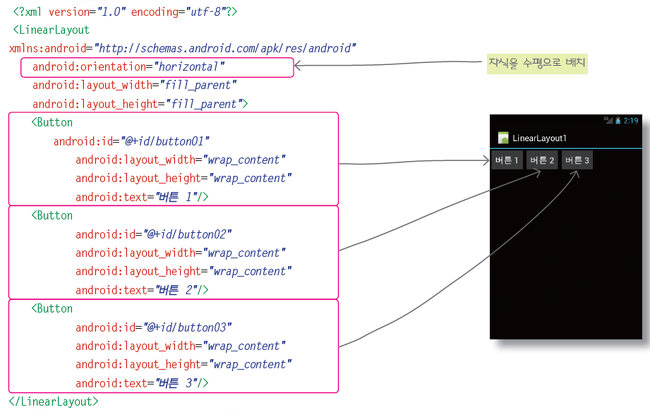
setGravity(int) : x축, y축 상 자식을 어떻게 배치할 것인지 지정

setBaselineAligned(boolean) : false로 설정되면 자식뷰들의 기준선을 정렬하지 않음,

                                                         true시 자식들의 하단을 정렬



선형 레이아웃으로 수평배치한 버튼들과 수직배치한 버튼들ㅎㅎ



선형레이아웃의 orientation 값으로 horizontal으로 수평으로 배치된 것을 볼 수 있어요

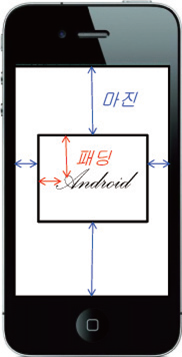


**가중치**

자식 뷰의 중요도를 나타냄

자식 뷰들이 가중치가 1,2,3이면 남은 공간의 1/6 , 2/6, 3/6을 각 각 할당 받음

0으로 설정된 뷰는 더 이상 확대되지 않고 남은 뷰들의 가중치 값을 비교해 크기가 계산 됨



**패딩**

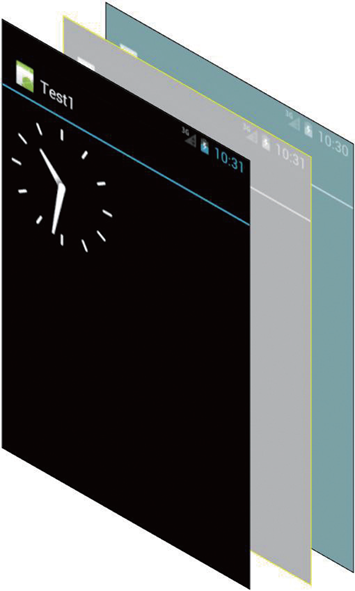
뷰의 경계와 뷰의 내용물 사이 간격

paddingBottom, paddingTop, paddingLeft, paddingRight에 값을 별도로 설정이 가능

**마진**

자식 뷰 주위의 여백

layout\_maginBottom, layout\_maginTop, layout\_maginLeft, layout\_maginRight



**프레임 레이아웃**

여러 자식 뷰들을 겹쳐서 배치

가시성(visibility)를 true로 설정해 순간에 하나의 뷰만 표시하려고 할 때 사용

**테이블 레이아웃**

자식뷰들을 테이블 형태로 배치

<TableRow> </TableRow>로 행을 나눔

\* 각 행마다 길이가 다르면 가장 긴 행의 길이가 테이블의 폭이 됨

\* 모든 셀들의 너비가 자동으로 결정, 높이 지정가능

\* 버튼같은 일반적인 뷰도 하나의 행이 될수 있음

**상대적 레이아웃(RelativeLayout)**

자식 뷰의 위치를 부모 뷰나 다른 자식 뷰들에 상대적으로 지정하는 방법

**절대적 레이아웃(AbsoluteLayout)**

절대적인 좌표값을 주어서 뷰들을 배치하는 방법, 뷰의 속성에 좌표값을 지정해줌

이용하기 쉬운 방법 이지만 크기가 제각각인 안드로이드기반에서 활용 효율이 떨어져 많이 사용되지 않음

**코드로 레이아웃 변경**

LinearLayout manager = (LinearLayout)findViewById(R.id.LayoutManager);

다운캐스팅해 id를 통해 기존객체를 참조

manager.setOrientation(LinearLayout.HORIZONTAL);

배치 방향을 수평으로 변경

Gravity로 정렬가능.

상대적 위치

android:layout\_alignParentTop=”true” 🡨 부모레이아웃의 위쪽에가서 붙어라

android:layout\_below=”@+id/button” 🡨저아이디부터 채우기 시작해라

android:layout\_above=”@+id/button” 🡨저아이디부터 까지 시작해라

android:layout\_alignParentBottom=”true” 🡨 부모레이아웃의 아래쪽에가서 붙어라

android:layout\_centerInParent=”true” 🡨 부모레이아웃의 가운데에가서 붙어라

layout\_align..많음..

scrollview 🡨 감싸면 스크롤 생김

Hori..scrollView

imageview 🡨 이미지넣는것.. src

속성

Visibility 가시성 보였다 안보였다.

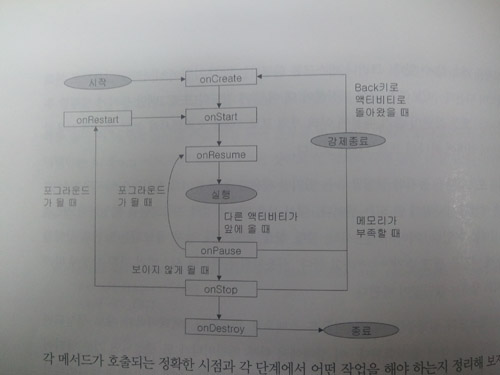
직접 setContentView를 하지못할 때

직접 처리해야되는거 LayoutInflater 클래스 제공하니 이걸로 처리. 🡨 부분레이아 웃을때

부분화면만들기

따로 레이아웃만들고 소스단에서. Layoutinflation 처리하면된다.

**액티브티 수명주기 생명주기**

****

onPause 할때 임시저장해놓을수 있다.

SharedPreferences pref = context.getSharedPreferences("Music", 3);

SharedPreferences.Editor editor = pref.edit();

Editor.putInt(“score”,100);

Editor.commit();

onResume 불러올 때

SharedPreferences pref = context.getSharedPreferences("Music", 3);

SharedPreferences.Editor editor = pref.edit();

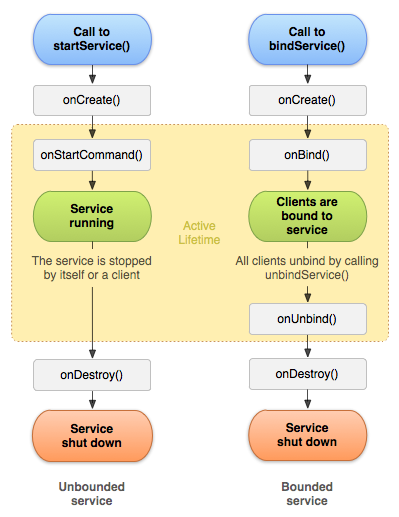
Editor.getInt(“score”,0);

서비스 생명주기

onCreate(처음호출되면) -> … onStartCommand(서비스 쪽에서 명령어를 받아주는곳)

**[Android] 서비스(Service)의 기초 3 - 서비스의 생명주기**

매우 기초적이지만 매우 중요한 ! 서비스의 생명주기(Service Lifecycle)에 대해서 알아보겠습니다. 아래 그림에서 왼쪽이 시작타입의 서비스, 오른쪽이 연결타입의 서비스의 생명주기 입니다.



**시작타입의 경우(왼쪽)**

 만약 startService()를 호출해서 서비스를 시작했다면 서비스 스스로가 stopSelf()를 호출하거나 다른 컴포넌트가 stopService()를 호출할 때까지 서비스가 계속 실행됩니다.

**연결타입의 경우(오른쪽)**

 만약 하나 이상의 컴포넌트가 bindService()를 호출해서 서비스를 시작했다면 서비스는 모든 연결된 컴포넌트가 연결을 해제하면 서비스는 소멸됩니다.

**서비스 (백그라운드에서 실행되는 어플)**

* 매니페스트에서 <service>등록..

Context.startService() 시작

Context.bindService() 바인드

stopService() 종료

unbindService()..

broadCastReceiver 메시지를 전달할 수 있는 방법

만드는방법

Extends Service 상속

* onBind 오버라이딩

매니페스트

* Service등록 Enabled = true, Exported=true

|  |
| --- |
| **package** com.example.servicetest;  **import** android.app.Activity;  **import** android.content.Intent;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.view.View.OnClickListener;  **import** android.widget.Button;  **public** **class** MainActivity **extends** Activity {  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    Button btn = (Button)findViewById(R.id.***serviceBtn***);  btn.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  Intent i = **new** Intent(getApplicationContext(),MyService.**class**);  i.putExtra("command", "start");  startService(i);  }  });  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| **package** com.example.servicetest;  **import** android.app.Service;  **import** android.content.Intent;  **import** android.os.IBinder;  **import** android.util.Log;  **public** **class** MyService **extends** Service {  **final** **public** String TAG = "MYSERVICE";    Thread printThread = **new** Thread(){  **public** **void** run() {  **for** (**int** i = 0; i < 100; i++) {  Log.*d*(TAG, "---"+i);  **try** {  Thread.*sleep*(1000);  } **catch** (InterruptedException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }    };  };    **public** MyService() {  }  @Override  **public** **void** onCreate() {  **super**.onCreate();  Log.*d*(TAG, "onCreate");  }    @Override  **public** **int** onStartCommand(Intent intent, **int** flags, **int** startId) {  Log.*d*(TAG, "onStartCommand");    **if**(**null**!=intent){  String s = intent.getStringExtra("command");  **if**(**null**!=s && "start".equals(s)){  printThread.start();  }  }    **return** **super**.onStartCommand(intent, flags, startId);  }    @Override  **public** **void** onDestroy() {  Log.*d*(TAG, "onDestroy");  **super**.onDestroy();  }    @Override  **public** IBinder onBind(Intent intent) {  **return** **null**;  }  } |
| <service android:enabled=*"true"* android:name=*"MyService"* android:exported=*"true"*></service> |

브로드 캐스트

sms받기

|  |
| --- |
| **package** com.example.mysmsreceiver;  **import** android.content.BroadcastReceiver;  **import** android.content.Context;  **import** android.content.Intent;  **import** android.util.Log;  **public** **class** MySMSReceiver **extends** BroadcastReceiver{  **final** String TAG = "MySMSReceiver";  **public** MySMSReceiver() {  }    @Override  **public** **void** onReceive(Context context, Intent intent) {  Log.*d*(TAG, "onReceive()");  Intent myIntent = **new** Intent(context,MainActivity.**class**);  myIntent.addFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***|Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  context.startActivity(intent);  }  } |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"com.example.mysmsreceiver"*  android:versionCode=*"1"*  android:versionName=*"1.0"* >  <uses-sdk  android:minSdkVersion=*"23"*  android:targetSdkVersion=*"23"* />  **<uses-permission android:name=*"android.permission.RECEIVE\_SMS"*/>**  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  <activity  android:name=*".MainActivity"*  android:label=*"@string/app\_name"* >  <intent-filter>  <action android:name=*"android.intent.action.MAIN"* />  <category android:name=*"android.intent.category.LAUNCHER"* />  </intent-filter>  </activity>  **<receiver android:name=*".MySMSReceiver"* android:enabled=*"true"* android:exported=*"true"*>**  **<intent-filter>**  **<action android:name=*"android.provider.Telephony.SMS\_RECEIVED"*/>**  **</intent-filter>**  **</receiver>**  </application>  </manifest> |

브로드캐스트2

|  |
| --- |
| /\*\*  \* SMS 수신을 위한 브로드캐스트 수신자입니다.  \*  \* **@author** Mike  \*/  **public** **class** SMSBroadcastReceiver **extends** BroadcastReceiver {  /\*\*  \* 로깅을 위한 태그  \*/  **public** **static** **final** String ***TAG*** = "SMSBroadcastReceiver";  @Override  **public** **void** onReceive(Context context, Intent intent) {  Log.i(***TAG***, "onReceive() 메소드 호출됨.");  // SMS 수신 시의 메시지인지 다시 한번 확인합니다.  **if** (intent.getAction().equals("android.provider.Telephony.SMS\_RECEIVED")) {  Log.i(***TAG***, "SMS를 수신하였습니다.");  // 우선순위가 낮은 다른 SMS 수신 앱들이 문자를 전달받지 못하도록 전달을 취소합니다.  abortBroadcast();  // SMS 메시지를 파싱합니다.  Bundle bundle = intent.getExtras();  Object messages[] = (Object[])bundle.get("pdus");  SmsMessage smsMessage[] = **new** SmsMessage[messages.length];  **int** smsCount = messages.length;  **for**(**int** i = 0; i < smsCount; i++) {  // PDU 포맷으로 되어 있는 메시지를 복원합니다.  smsMessage[i] = SmsMessage.createFromPdu((**byte**[])messages[i]);  }  // SMS 수신 시간 확인  Date curDate = **new** Date(smsMessage[0].getTimestampMillis());  Log.i(***TAG***, "SMS Timestamp : " + curDate.toString());  // SMS 발신 번호 확인  String origNumber = smsMessage[0].getOriginatingAddress();    // SMS 메시지 확인  String message = smsMessage[0].getMessageBody().toString();  Log.i(***TAG***, "SMS : " + origNumber + ", " + message);  // 메시지를 보여줄 액티비티를 띄워줍니다.  Intent myIntent = **new** Intent(context, SMSDisplayActivity.**class**);    // 플래그를 이용합니다.  myIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);    myIntent.putExtra("number", origNumber);  myIntent.putExtra("message", message);  myIntent.putExtra("timestamp", curDate.toString());  context.startActivity(myIntent);  }  }  } |
| /\*\*  \* SMS를 수신했을 때 화면에 보여지는 액티비티입니다.  \*  \* **@author** Mike  \*/  **public** **class** SMSDisplayActivity **extends** Activity {  Button titleButton;  Button closeButton;  TextView messageTextView;  @Override  **public** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  // 타이틀 영역을 안보이게 합니다.  requestWindowFeature(Window.***FEATURE\_NO\_TITLE***);  setContentView(R.layout.sms\_display);  // 레이아웃의 객체들을 참조합니다.  titleButton = (Button) findViewById(R.id.titleButton);  closeButton = (Button) findViewById(R.id.closeButton);  messageTextView = (TextView) findViewById(R.id.messageTextView);  // 닫기 버튼을 누르면 화면을 닫습니다.  closeButton.setOnClickListener(**new** OnClickListener() {  **public** **void** onClick(View v) {  finish();  }  });    // 전달된 인텐트를 처리합니다.  Intent passedIntent = getIntent();  **if** (passedIntent != **null**) {  processIntent(passedIntent);  }  }  /\*\*  \* 이 액티비티가 SINGLE\_TOP 플래그로 되어 있으면 이 메소드로 인텐트가 전달됩니다.  \*/  **protected** **void** onNewIntent(Intent intent) {  // 전달된 인텐트를 처리합니다.  processIntent(intent);  **super**.onNewIntent(intent);  }  /\*\*  \* 전달된 인텐트를 처리합니다.  \*  \* **@param** intent  \*/  **private** **void** processIntent(Intent intent) {  // 발신 번호  String number = intent.getStringExtra("number");  // 수신 내용  String message = intent.getStringExtra("message");    // 시간  String timestamp = intent.getStringExtra("timestamp");  **if** (number != **null**) {  titleButton.setText(number + " 에서 문자 수신");  messageTextView.setText("[" + timestamp + "] " + message);  messageTextView.invalidate();  }  }  }  <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"org.androidtown.basic.receiver.sms"* >  **<uses-permission android:name=*"android.permission.RECEIVE\_SMS"* />**  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  <activity  android:name=*".MainActivity"*  android:label=*"@string/app\_name"* >  <intent-filter>  <action android:name=*"android.intent.action.MAIN"* />  <category android:name=*"android.intent.category.LAUNCHER"* />  </intent-filter>  </activity>  <activity android:name=*"SMSDisplayActivity"* >  </activity>  **<receiver android:name=*"SMSBroadcastReceiver"*>**  **<intent-filter>**  **<action android:name=*"android.provider.Telephony.SMS\_RECEIVED"*/>**  **</intent-filter>**  **</receiver>**  </application>  </manifest> |

Alert 창뛰우기

|  |
| --- |
| AlertDialog.Builder gsDialog = **new** AlertDialog.Builder(context);  gsDialog.setTitle("GPS Status OFF !!!");  gsDialog.setMessage("Change Setting !!");  gsDialog.setPositiveButton("OK", **new** DialogInterface.OnClickListener() {  **public** **void** onClick(DialogInterface dialog, **int** which) {  // GPS설정 화면으로 이동  Intent intent = **new** Intent(android.provider.Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  intent.addCategory(Intent.***CATEGORY\_DEFAULT***);  context.startActivity(intent);  }  }).create().show(); |

프레그먼트

허니콤부터 나오기시작함

* 두개의 화면을 액티비티로 만들고 액티비티 간 전환
* 하나의 액티비티 위에 프래그 먼트를 두고 프래그 먼트간 전환

프래그먼트로 만들어 두면 스마트폰에서는 프래그먼트간 화면 전환, 태블릿에는 두개의 프래그먼트를 하나의 액티비티 위에 동시에 보여주는 화면 분할이 가능함

화면 제어.교체할 때 트랜젝션을 사용한다.

우선 큰레이아웃 하나 넣고 (공백) 그안에 여러 화면을 add comit rolback등을 프래그먼트 매니저로 사용한다.

|  |
| --- |
| Main  **public** **class** MainActivity **extends** Activity {  **public** **static** **class** PlaceholderFragment **extends** Fragment {  **public** PlaceholderFragment() {  }  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View rootView = inflater.inflate(R.layout.***fragment\_main***, container, **false**);  **return** rootView;  }  }    PlaceholderFragment fg1 = **null**;//new PlaceholderFragment();  MenuFragment fg2 = **null**;//new MenuFragment();    @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);      fg1 = **new** PlaceholderFragment();  fg2 = **new** MenuFragment();  //  **if** (savedInstanceState == **null**) {  getFragmentManager().beginTransaction().add(R.id.***container***, fg1).commit();  }  //  // Button b = (Button)findViewById(R.id.button1);  // Log.d("visualkhh", ""+b);  // b.setOnClickListener(new OnClickListener() {  // @Override  // public void onClick(View v) {  //// getFragmentManager().beginTransaction().replace(R.id.container, fg2).commit();  // }  // });  }    **boolean** sw = **true**;  **public** **void** onB1Click(View v){  **if**(sw){  getFragmentManager().beginTransaction().replace(R.id.***container***, fg2).commit();  }**else**{  getFragmentManager().beginTransaction().replace(R.id.***container***, fg1).commit();  }  sw = !sw;  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| Menu  **public** **class** MenuFragment **extends** Fragment {  **public** MenuFragment() {  }  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View rootView = inflater.inflate(R.layout.***fragment\_menu***, container, **false**);  **return** rootView;  }  } |
| Fg1  <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.myfragment.MainActivity$PlaceholderFragment"* >  <TextView  android:id=*"@+id/textView1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentTop=*"true"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginTop=*"107dp"*  android:text=*"main view"* />  <Button  android:id=*"@+id/button1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_below=*"@+id/textView1"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginTop=*"42dp"*  android:onClick=*"onB1Click"*  android:text=*"go menu"* />  </RelativeLayout> |
| Fg2  <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.myfragment.MainActivity$PlaceholderFragment"* >  <TextView  android:id=*"@+id/textView1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentTop=*"true"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginTop=*"85dp"*  android:text=*"menu view"* />  <Button  android:id=*"@+id/button1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_centerHorizontal=*"true"*  android:layout\_centerVertical=*"true"*  android:onClick=*"onB1Click"*  android:text=*"back"* />  </RelativeLayout> |

터치이벤트

|  |
| --- |
| Button b = (Button)findViewById(R.id.***button1***);  b.setOnTouchListener(**new** OnTouchListener() {  @Override  **public** **boolean** onTouch(View v, MotionEvent event) {  **if**(event.getAction() == MotionEvent.***ACTION\_DOWN***){  //터치  }**else** **if**(event.getAction() == MotionEvent.***ACTION\_UP***){  // 뛸때  }  **return** **true**;  }  }); |

커스텀뷰만들기

|  |
| --- |
| **package** com.example.myfragment;  **import** android.content.Context;  **import** android.util.AttributeSet;  **import** android.util.Log;  **import** android.view.MotionEvent;  **import** android.view.View;  **public** **class** MyView **extends** View{  **public** MyView(Context context) {  **super**(context);  }  **public** MyView(Context context, AttributeSet attrs) {  **super**(context, attrs);  }  @Override  **public** **boolean** onTouchEvent(MotionEvent event) {  **if**(event.getAction() == MotionEvent.***ACTION\_DOWN***){  Log.*d*("vi", "tt");  }**else** **if**(event.getAction() == MotionEvent.***ACTION\_UP***){  Log.*d*("vi", "tu");  }  **return** **true**;  }      } |
| <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.myfragment.MainActivity$PlaceholderFragment"* >  <TextView  android:id=*"@+id/textView1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentTop=*"true"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginTop=*"85dp"*  android:text=*"menu view"* />  <Button  android:id=*"@+id/button1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_centerHorizontal=*"true"*  android:layout\_centerVertical=*"true"*  android:onClick=*"onB1Click"*  android:text=*"back"* />    <com.example.myfragment.MyView  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:layout\_below=*"@+id/button1"*  android:background=*"#ffeeccff"*  />  </RelativeLayout> |

제스쳐 딕텍터

|  |
| --- |
| **public** **class** MainActivity **extends** Activity {  …  GestureDetector gd = **null**;  …  **gd = new GestureDetector(this, new GestureDetector.SimpleOnGestureListener(){**  **@Override**  **public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {**  **Log.*d*("vi", "gd~~ scroll");**  **return super.onScroll(e1, e2, distanceX, distanceY);**  **}**  **@Override**  **public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {**  **Log.*d*("vi", "gd~~ onFling");**  **return super.onFling(e1, e2, velocityX, velocityY);**  **}**  **});**    }  **@Override**  **public boolean onTouchEvent(MotionEvent event) {**  **if(null!=gd){**  **return gd.onTouchEvent(event);**  **}else{**  **return super.onTouchEvent(event);**  **}**  **}**  **…**  } |

키보드 이벤트

키이벤트

|  |
| --- |
| @Override  **public** **void** onBackPressed() {  Toast.*makeText*(getApplicationContext(), "back key", Toast.***LENGTH\_LONG***).show();  **return**;  //super.onBackPressed();  }    @Override  **public** **boolean** onKeyDown(**int** keyCode, KeyEvent event) {  // **TODO** Auto-generated method stub  **return** **super**.onKeyDown(keyCode, event);  } |

단말 회전했을 때.. 또는 단말쪽에 이벤트가 생겼을 때

|  |
| --- |
| 매니페스트  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  <activity  android:name=*".MainActivity"*  android:label=*"@string/app\_name"*  **android:configChanges=*"orientation|screenSize|keyboardHidden"***  >  <intent-filter>  <action android:name=*"android.intent.action.MAIN"* />  <category android:name=*"android.intent.category.LAUNCHER"* />  </intent-filter>  </activity>  </application> |
| @Override  **public** **void** onConfigurationChanged(Configuration newConfig) {  **if**(newConfig.orientation == Configuration.***ORIENTATION\_LANDSCAPE***){  Toast.*makeText*(getApplicationContext(), "ORIENTATION\_LANDSCAPE", Toast.***LENGTH\_LONG***).show();  }**else** **if**(newConfig.orientation == Configuration.***ORIENTATION\_PORTRAIT***){  Toast.*makeText*(getApplicationContext(), "ORIENTATION\_PORTRAIT", Toast.***LENGTH\_LONG***).show();  }  **super**.onConfigurationChanged(newConfig);  } |

웹브라우저

|  |
| --- |
| Button b = (Button)findViewById(R.id.***button1***);  b.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  WebView wv = (WebView)findViewById(R.id.***webView***);  wv.getSettings().setJavaScriptEnabled(**true**);  wv.loadUrl("http://m.naver.com");  }  }); |
| <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.mywebview.MainActivity"* >  <Button  android:id=*"@+id/button1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentLeft=*"true"*  android:layout\_alignParentTop=*"true"*  android:layout\_marginLeft=*"28dp"*  android:text=*"open page"* />  <WebView  android:id=*"@+id/webView"*  android:layout\_below=*"@+id/button1"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  >    </WebView>  </RelativeLayout> |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"com.example.mywebview"*  android:versionCode=*"1"*  android:versionName=*"1.0"* >  **<uses-permission android:name=*"android.permission.INTERNET"*/>**  …  </manifest> |

내부 페이지 뛰우기

|  |  |
| --- | --- |
| Button b = (Button)findViewById(R.id.***button1***);  b.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  WebView wv = (WebView)findViewById(R.id.***webView***);  wv.getSettings().setJavaScriptEnabled(**true**);  //wv.loadUrl("http://m.naver.com");  wv.loadUrl("file:///android\_asset/simple.html");  //wv.loadData(data, jj, encoding);  }  }); |  |

애니메이션

|  |  |
| --- | --- |
| Button b = (Button)findViewById(R.id.***button1***);  b.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  Animation translate = AnimationUtils.*loadAnimation*(c, R.anim.***translate***);  TextView t = (TextView)findViewById(R.id.***tt***);  t.startAnimation(translate);    }  }); | |
| res/anim/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=*"1500"*  />    </set> |  |

페이지 슬라이딩

|  |
| --- |
| **public** **class** MainActivity **extends** Activity {  LinearLayout sp= **null**;  Button b = **null**;  Animation aleft = **null**;  Animation aright = **null**;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    aleft = AnimationUtils.*loadAnimation*(**this**, R.anim.***translate\_left***);  aright = AnimationUtils.*loadAnimation*(**this**, R.anim.***translate\_right***);    **aleft.setAnimationListener(new AnimationListener() {**  **@Override**  **public void onAnimationStart(Animation animation) {**  **}**    **@Override**  **public void onAnimationRepeat(Animation animation) {**  **}**  **@Override**  **public void onAnimationEnd(Animation animation) {**  **b.setText("close");**  **}**  **});**      sp = (LinearLayout) findViewById(R.id.***slidingPanel***);  b = (Button)findViewById(R.id.***button***);    b.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  sp.setVisibility(View.***VISIBLE***);  sp.startAnimation(aleft);  }  });  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| Left anim  <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"*    />  </set> |
| Right anim  <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"*    />  </set> |

뷰페이져 플링핑.

|  |
| --- |
| **package** com.example.viewpager;  **import** android.app.Activity;  **import** android.os.Bundle;  **import** android.support.v4.view.PagerAdapter;  **import** android.support.v4.view.ViewPager;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.view.View.OnClickListener;  **import** android.view.ViewGroup;  **import** android.widget.Button;  **import** android.widget.LinearLayout;  **import** android.widget.TextView;  **public** **class** MainActivity **extends** Activity {      **class MyAdapter extends PagerAdapter{**  **String[] names ={"k","h","h"};**  **@Override**  **public int getCount() {**  **return names.length;**  **}**  **@Override**  **public boolean isViewFromObject(View view, Object o) {**  **return view.equals(o);**  **}**  **@Override**  **public void destroyItem(ViewGroup container, int position, Object object) {**  **container.removeView((View)object);**  **}**  **@Override**  **public Object instantiateItem(ViewGroup container, int position) {**  **LinearLayout layout = new LinearLayout(getApplicationContext());**  **layout.setOrientation(LinearLayout.*VERTICAL*);**  **TextView t = new TextView(getApplicationContext()) ;**  **t.setText(names[position]);**  **t.setTextSize(40.0f);**  **t.setTextColor(0xff000000);**  **layout.addView(t);**  **container.addView(layout);**  **return layout;**  **}**    **}**    ViewPager pager = **null**;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    pager = (ViewPager)findViewById(R.id.***pager***);  MyAdapter adapter = **new** MyAdapter();  pager.setAdapter(adapter);    Button b = (Button)findViewById(R.id.***button1***);  b.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  **pager.setCurrentItem(2);**  }  });      }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
|  |

Progress , SeekBar

|  |
| --- |
| **public** **class** MainActivity **extends** Activity {  Context context = **this**;  Button b = **null**;  Button b2 = **null**;  ProgressBar p=**null**;  ProgressBar p2=**null**;  SeekBar sb = **null**;  TextView sText = **null**;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    b = (Button)findViewById(R.id.***button1***);  p = (ProgressBar)findViewById(R.id.***progressBar1***);  b2 = (Button)findViewById(R.id.***button2***);  sb = (SeekBar)findViewById(R.id.***seekBar1***);  sText = (TextView)findViewById(R.id.***stext***);  b.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  p.setProgress(50);  }  });    sb.setOnSeekBarChangeListener(**new** OnSeekBarChangeListener() {  @Override  **public** **void** onStopTrackingTouch(SeekBar seekBar) {  }  @Override  **public** **void** onStartTrackingTouch(SeekBar seekBar) {  }  **public** **void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromUser) {  sText.setText("val "+progress);  }  });    b2.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  ProgressDialog dialog = **new** ProgressDialog(context);  dialog.setProgressStyle(ProgressDialog.***STYLE\_SPINNER***);  dialog.setTitle("..");  dialog.setMessage("plase wait..");  dialog.show();  //dialog.dismiss();  }  });    }    @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.myprogress.MainActivity"* >  <ProgressBar  android:id=*"@+id/progressBar1"*  style=*"?android:attr/progressBarStyleHorizontal"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignParentLeft=*"true"*  android:layout\_alignParentRight=*"true"*  android:layout\_alignParentTop=*"true"*  android:max=*"100"* />  <Button  android:id=*"@+id/button1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_below=*"@+id/progressBar1"*  android:layout\_centerHorizontal=*"true"*  android:layout\_marginTop=*"20dp"*  android:text=*"Button"* />  <Button  android:id=*"@+id/button2"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/button1"*  android:layout\_below=*"@+id/button1"*  android:layout\_marginTop=*"34dp"*  android:text=*"Button"* />  <SeekBar  android:id=*"@+id/seekBar1"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/progressBar1"*  android:layout\_alignRight=*"@+id/progressBar1"*  android:layout\_below=*"@+id/button2"*  android:layout\_marginTop=*"16dp"*  android:max=*"100"* />  <TextView  android:id=*"@+id/stext"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:layout\_alignLeft=*"@+id/button2"*  android:layout\_below=*"@+id/seekBar1"*  android:text=*"TextView"* />  </RelativeLayout> |

액션바 actionbar

|  |  |
| --- | --- |
| **public** **class** MainActivity **extends** Activity {  ..  **protected** **void** onCreate(Bundle savedInstanceState) {  **…**  }    **//메뉴만들어질때**  **@Override**  **public boolean onCreateOptionsMenu(Menu menu) {**  **// Inflate the menu; this adds items to the action bar if it is present.**  **getMenuInflater().inflate(R.menu.*main*, menu);**  **return true;**  **}**  **//메뉴선택시**  **@Override**  **public boolean onOptionsItemSelected(MenuItem item) {**  **// Handle action bar item clicks here. The action bar will**  **// automatically handle clicks on the Home/Up button, so long**  **// as you specify a parent activity in AndroidManifest.xml.**  **int id = item.getItemId();**  **if (id == R.id.*action\_settings*) {**  **return true;**  **}else if(id==R.id.*action\_search*){**  **Toast.*makeText*(getApplicationContext(), "selected search men", Toast.*LENGTH\_LONG*).show();;**  **}**  **return super.onOptionsItemSelected(item);**  **}**  } | |
|  |  |

탭

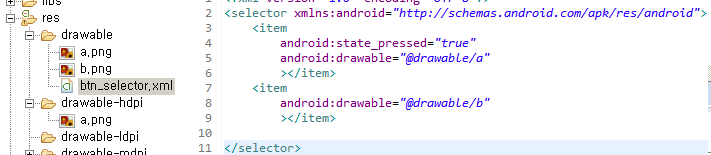
|  |  |
| --- | --- |
|  | 그닥 추천하지 않는다.    Fragment로 직접 제작하라. |

키패드 속성바꾸기

textFild에서 타입만 바꾸꿪주면된다 inputType=”number”등등

셀렉터 selector

버튼 속성에 따라 보여지는 그림



직접 버튼을 만들어서 셀렉터를 만들어보자

|  |
| --- |
| **public** **class** BitMapButton **extends** Button{  **public** BitMapButton(Context context) {  **super**(context);  init();  }  **public** BitMapButton(Context context, AttributeSet attrs) {  **super**(context, attrs);  init();  }  **private** **void** init() {  setBackgroundResource(R.drawable.***a***);  }    @Override  **public** **boolean** onTouchEvent(MotionEvent event) {  **if**(event.getAction() == MotionEvent.***ACTION\_DOWN***){  setBackgroundResource(R.drawable.***b***);  }**else** **if**(event.getAction() == MotionEvent.***ACTION\_UP***){  setBackgroundResource(R.drawable.***a***);  }  **return** **true**;  }  } |

리스트뷰

* 어뎁터를 이용해서 처리한다

|  |
| --- |
| **public** **class** MainActivity **extends** Activity {    **class** MyAdapter **extends** BaseAdapter{  String[] a ={"a","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d","B","c","d"};  **@Override**  **public int getCount() {**  **return a.length;**  **}**  **@Override**  **public Object getItem(int position) {**  **return a[position];**  **}**  **@Override**  **public long getItemId(int position) {**  **return position;**  **}**  **@Override**  **public View getView(int position, View convertView, ViewGroup parent) {**  **TextView t = new TextView(getApplicationContext());**  **t.setText(a[position]);**  **t.setTextColor(0xff000000);**    **return t;**  **}**    }  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    MyAdapter adapter = **new** MyAdapter();    ListView list = (ListView)findViewById(R.id.***list***);  list.setAdapter(adapter);    }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.mylist.MainActivity"* >  <TextView  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:text=*"@string/hello\_world"* />  <ListView  android:id=*"@+id/list"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"* >  </ListView>  </RelativeLayout> |

직접 부분 레이아웃을 가지고 리스트뷰를 만들자

|  |
| --- |
| 메인 액티비티 |
| **public** **class** MainActivity **extends** Activity {    **class** MyAdapter **extends** BaseAdapter{  String[] a ={"a","B","c","d","B","c","d","B","c","d","B","d","B","c","d"};  @Override  **public** **int** getCount() {  **return** a.length;  }  @Override  **public** Object getItem(**int** position) {  **return** a[position];  }  @Override  **public** **long** getItemId(**int** position) {  **return** position;  }  @Override  **public View getView(int position, View convertView, ViewGroup parent) {**  **// TextView t = new TextView(getApplicationContext());**  **// t.setText(a[position]);**  **// t.setTextColor(0xff000000);**  **ItemView v = null;**  **if(convertView==null){**  **v = new ItemView(getApplicationContext());**  **}else{**  **v = (ItemView) convertView; //재사용할거 있으면**  **}**  **v.age.setText(position+"<-");**  **v.name.setText(a[position]);**  **return v;**  **}**  }  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    MyAdapter adapter = **new** MyAdapter();    ListView list = (ListView)findViewById(R.id.***list***);  list.setAdapter(adapter);    }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  } |
| 부분레이아웃잡아줄 객체  **package** com.example.mylist;  **import** android.content.Context;  **import** android.util.AttributeSet;  **import** android.view.LayoutInflater;  **import** android.widget.LinearLayout;  **import** android.widget.TextView;  **public** **class** ItemView **extends** LinearLayout{  **public** TextView name = **null**;  **public** TextView age = **null**;  **public** ItemView(Context context) {  **super**(context);  init(context);  }  **public** ItemView(Context context, AttributeSet attrs) {  **super**(context, attrs);  init(context);  }  **private void init(Context context) {**  **LayoutInflater inflater = (LayoutInflater)context.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);**  **inflater.inflate(R.layout.*item*, this);**  **name = (TextView)findViewById(R.id.*name*);**  **age = (TextView)findViewById(R.id.*age*);**    **}**  } |
| Item.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=*"wrap\_content"*  android:orientation=*"horizontal"* >    <ImageView  android:layout\_width=*"100dp"*  android:layout\_height=*"100dp"*  android:src=*"@drawable/ic\_launcher"*  />  <LinearLayout  android:orientation=*"vertical"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"wrap\_content"*  android:layout\_gravity=*"center\_vertical"*  >  <TextView  android:id=*"@+id/name"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"wrap\_content"*  android:textSize=*"30dp"*  android:textColor=*"#ff0000ff"*  />  <TextView  android:id=*"@+id/age"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"wrap\_content"*  android:textSize=*"20dp"*  android:textColor=*"#ffff0000"*  />    </LinearLayout>  </LinearLayout> |
|  |

피티너

ArrayAdapter를 가지고 처리하는방법

|  |
| --- |
| **public** **class** MainActivity **extends** Activity {  **public** String[] a= {"A","b"};  Spinner spn = **null**;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    spn = (Spinner)findViewById(R.id.***spn***);    ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_item***,a);  spn.setAdapter(adapter);  spn.setOnItemSelectedListener(**new** OnItemSelectedListener() {  @Override  **public** **void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  Toast.*makeText*(getApplicationContext(), a[position], Toast.***LENGTH\_LONG***).show();  }  @Override  **public** **void** onNothingSelected(AdapterView<?> parent) {  }  });  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  android:paddingBottom=*"@dimen/activity\_vertical\_margin"*  android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*  android:paddingRight=*"@dimen/activity\_horizontal\_margin"*  android:paddingTop=*"@dimen/activity\_vertical\_margin"*  tools:context=*"com.example.myspiner.MainActivity"* >  <TextView  android:id=*"@+id/textView1"*  android:layout\_width=*"wrap\_content"*  android:layout\_height=*"wrap\_content"*  android:text=*"@string/hello\_world"* />  <Spinner  android:layout\_width=*"match\_parent"*  android:layout\_height=*"wrap\_content"*  android:layout\_below=*"@+id/textView1"*  android:layout\_marginTop=*"40dp"*  android:id=*"@+id/spn"* />  </RelativeLayout> |

ListActivity로 구성할수도있다 list를

|  |
| --- |
| **package** com.example.mylist2;  **import** android.app.Activity;  **import** android.app.ListActivity;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.widget.ArrayAdapter;  **public** **class** MainActivity **extends** ListActivity {  String[] a ={"A","b"};  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  //setContentView(R.layout.activity\_main);    setListAdapter(**new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***,a));  }    @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.***main***, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

그리드뷰는

컬럼숫자를 view에서 설정해주면 격자로 보여줄수 있다.

멀티 터치 체크

|  |
| --- |
| @Override  **public** **boolean** onTouchEvent(MotionEvent event) {  **int** action = event.getAction();  **int** fcnt = event.getPointerCount();  event.getX(0);  event.getX(1); |

View의

onSizeChanged 함수는

크기가 정해지면 호출되는 함수도

그래픽

* Ondraw
* onTouch

|  |  |
| --- | --- |
| **public** **class** MyCustomView **extends** View{  Paint paint= **null**;  **public** MyCustomView(Context context) {  **super**(context);  init(context);  }  **public** MyCustomView(Context context, AttributeSet attrs) {  **super**(context, attrs);  init(context);  }    **private** **void** init(Context context) {  paint = **new** Paint();  paint.setColor(Color.***RED***);  }      @Override  **protected** **void** onDraw(Canvas canvas) {  **super**.onDraw(canvas);  canvas.drawRect(100, 100, 200, 200, paint);  }  } | |
| @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    LinearLayout l = (LinearLayout) findViewById(R.id.***sublayout***);  MyCustomView v = **new** MyCustomView(**this**);  v.setBackgroundColor(Color.***CYAN***);  l.addView(v);  } |  |
| **protected** **void** onDraw(Canvas canvas) {  **super**.onDraw(canvas);  canvas.drawRect(100, 100, 200, 200, paint);      //편하기 쓰라고 drawable 있음 ..Shape랑..  ShapeDrawable drawable1 = **new** ShapeDrawable();  RectShape shape1 = **new** RectShape();  drawable1.setShape(shape1);  drawable1.setBounds(300, 100, 500, 300);  drawable1.draw(canvas);    LinearGradient gd = **new** LinearGradient(0, 0, 0, 300, Color.***BLACK***, Color.***YELLOW***, Shader.TileMode.***CLAMP***);  paint.setShader(gd);    shape1.resize(300, 300);  drawable1.setBounds(400,300,700,600);  drawable1.draw(canvas);    } |  |

그래픽

더블버퍼링

* bitmap객체에 미리 적어놓고 나중에 draw한다.

매트릭스

매트릭스가지고 스케일 및 이동시킬수있다

매트릭스 계산할수있음

Matrix..

그림판만들기

* 21일3장

|  |
| --- |
| // Path  Path path = **new** Path();  path.moveTo(20, 20);  path.lineTo(120, 20);  path.lineTo(160, 90);  path.lineTo(180, 80);  path.lineTo(200, 120);   * canvas.drawPath(path, pathPaint); |

서피스뷰 SurfaceView

* 하드웨어 가속 쓰기위한 것. 22일

모든 뷰들도 하드웨어 가속쓸수있다

* Android : hardwareAccelerated=”true” 넣어줘야함. 매니페스트 Activity에..

소프트웨어 카메라 camera

* 화면자체를 수정해줄수있는 것..
* 매트릭스같은거.. 매트릭스 뽑아내서 사용자에서 쓰면된다.

|  |
| --- |
| **public** **class** CoverFlow **extends** ~~Gallery~~ {  **private** Camera camera = **new** Camera();  /\*\*  \* 회전 각도  \*/  **public** **static** **int** *maxRotationAngle* = 55;  /\*\*  \* 최대 확대 수준  \*/  **public** **static** **int** *maxZoom* = -60;  **private** **int** centerPoint;  /\*\*  \* 생성자  \*  \* **@param** context  \*/  **public** CoverFlow(Context context) {  **super**(context);  init();  }  /\*\*  \* 생성자  \*  \* **@param** context  \* **@param** attrs  \*/  **public** CoverFlow(Context context, AttributeSet attrs) {  **super**(context, attrs);  init();  }  /\*\*  \* 생성자  \*  \* **@param** context  \* **@param** attrs  \* **@param** defStyle  \*/  **public** CoverFlow(Context context, AttributeSet attrs, **int** defStyle) {  **super**(context, attrs, defStyle);  init();  }  /\*\*  \* 초기화  \*/  **private** **void** init() {  **this**.setStaticTransformationsEnabled(**true**);  }  **public** **int** getMaxRotationAngle() {  **return** *maxRotationAngle*;  }  **public** **void** setMaxRotationAngle(**int** rotationAngle) {  *maxRotationAngle* = rotationAngle;  }  **public** **int** getMaxZoom() {  **return** *maxZoom*;  }  **public** **void** setMaxZoom(**int** zoom) {  *maxZoom* = zoom;  }  **private** **int** getCenterOfCoverflow() {  **return** (getWidth() - getPaddingLeft() - getPaddingRight()) / 2 + getPaddingLeft();  }  **private** **static** **int** getCenterOfView(View view) {  **return** view.getLeft() + view.getWidth() / 2;  }  **protected** **boolean** getChildStaticTransformation(View child, Transformation t) {  **final** **int** childCenter = *getCenterOfView*(child);  **final** **int** childWidth = child.getWidth() ;  **int** rotationAngle = 0;  t.clear();  t.setTransformationType(Transformation.***TYPE\_MATRIX***);  **if** (childCenter == centerPoint) {  transformImageBitmap((ImageView) child, t, 0);  } **else** {  rotationAngle = (**int**) (((**float**) (centerPoint - childCenter)/ childWidth) \* *maxRotationAngle*);  **if** (Math.*abs*(rotationAngle) > *maxRotationAngle*) {  rotationAngle = (rotationAngle < 0) ? -*maxRotationAngle* : *maxRotationAngle*;  }  transformImageBitmap((ImageView) child, t, rotationAngle);  }  **return** **true**;  }  **protected** **void** onSizeChanged(**int** w, **int** h, **int** oldw, **int** oldh) {  centerPoint = getCenterOfCoverflow();  **super**.onSizeChanged(w, h, oldw, oldh);  }  **private** **void** transformImageBitmap(ImageView child, Transformation t, **int** rotationAngle) {  camera.save();  **final** Matrix imageMatrix = t.getMatrix();;  **final** **int** imageHeight = child.getLayoutParams().height;;  **final** **int** imageWidth = child.getLayoutParams().width;  **final** **int** rotation = Math.*abs*(rotationAngle);  camera.translate(0.0f, 0.0f, 100.0f);  **if** ( rotation < *maxRotationAngle* ) {  **float** zoomAmount = (**float**) (*maxZoom* + (rotation \* 1.5));  camera.translate(0.0f, 0.0f, zoomAmount);  }  camera.rotateY(rotationAngle);  camera.getMatrix(imageMatrix);  imageMatrix.preTranslate(-(imageWidth/2), -(imageHeight/2));  imageMatrix.postTranslate((imageWidth/2), (imageHeight/2));  camera.restore();  }  } |

쓰레드.. 동시접근에 따른 Handler 사용해라

* 다른쓰레드에서 뷰를 접근하려고할 때.. 접근을못하니(동시접근문제때문에) handler를 사용한다

|  |
| --- |
| **public** **class** SampleThreadRunnableActivity **extends** Activity {  ProgressBar bar;  TextView textView01;  **boolean** isRunning = **false**;  Handler handler;  ProgressRunnable runnable;    **public** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    bar = (ProgressBar) findViewById(R.id.progress);  textView01 = (TextView) findViewById(R.id.textView01);    handler = **new** Handler();  runnable = **new** ProgressRunnable();  }    **public** **void** onStart() {  **super**.onStart();    bar.setProgress(0);  Thread thread1 = **new** Thread(**new** Runnable() {  **public** **void** run() {  **try** {  **for** (**int** i = 0; i < 20 && isRunning; i++) {  Thread.*sleep*(1000);    **handler.post(runnable);**  **//postDelayed로 몇초후실행되게도할수있다**  }  } **catch** (Exception ex) {  Log.*e*("SampleThreadActivity", "Exception in processing message.", ex);  }  }  });    isRunning = **true**;  thread1.start();  }  **public** **void** onStop() {  **super**.onStop();  isRunning = **false**;  }      **public class ProgressRunnable implements Runnable {**  **public void run() {**  **bar.incrementProgressBy(5);**  **if (bar.getProgress() == bar.getMax()) {**  **textView01.setText("Runnable Done");**  **} else {**  **textView01.setText("Runnable Working ..." + bar.getProgress());**  **}**    **}**    **}**    } |

또다른방식은 핸들을 재정의해서 handleMessage(Message msg)

로 훅킹받아라

메시지보낼땐(다른쓰레드에서)

Message message = handler.obtainMessage();

Bundle bundle = new Bundle();

Bundle.putString(“a”,”data”);

Message.setData(bundle)

Handler.sendMessage(message)로 호출하면 메인쓰레드가서 처리된다. 따라서 접근가능

**별도의 쓰레드쪽으로 데이터 전송할 때 루퍼 Looper**

* 메인쓰레드에서 별도의 쓰레드로 보낼 때 루퍼를 사용한다.

|  |
| --- |
| **public** **class** MainActivity **extends** ActionBarActivity {  TextView textView01, textView02;  EditText editText01, editText02;  /\*\*  \* 메인 스레드의 핸들러  \*/  MainHandler mainHandler;  /\*\*  \* 새로 만든 스레드  \*/  ProcessThread thread1;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  mainHandler = **new** MainHandler();  thread1 = **new** ProcessThread();  textView01 = (TextView) findViewById(R.id.textView01);  textView02 = (TextView) findViewById(R.id.textView02);  editText01 = (EditText) findViewById(R.id.editText01);  editText02 = (EditText) findViewById(R.id.editText02);  // 버튼 이벤트 처리  Button processBtn = (Button) findViewById(R.id.processBtn);  processBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  String inStr = editText01.getText().toString();  Message msgToSend = Message.*obtain*();  msgToSend.obj = inStr;  thread1.handler.sendMessage(msgToSend);  //post로 Runnable을 보낼수도있다.  }  });  thread1.start();  }  /\*\*  \* 새로 정의한 스레드  \*/  **class** ProcessThread **extends** Thread {  // 새로운 스레드를 위한 핸들러  ProcessHandler handler;  **public** ProcessThread() {  handler = **new** ProcessHandler();  }  **public** **void** run() {  // 루퍼 사용  Looper.*prepare*();  Looper.*loop*();  }  }  **class** ProcessHandler **extends** Handler {  **public** **void** handleMessage(Message msg) {  Message resultMsg = Message.*obtain*();  resultMsg.obj = msg.obj + " Mike!!!";  mainHandler.sendMessage(resultMsg);  }  }  **class** MainHandler **extends** Handler {  **public** **void** handleMessage(Message msg) {  String str = (String) msg.obj;  editText02.setText(str);  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

AsyncTask를 이용하여 좀더 쉽게 쓰레드

* 백그라운드 작업을 좀더 쉽고 간단하게 하고싶다면 AsyncTask클래스 사용
* Execute()메서드를 실행하여 백그라운드 작업 수행하고 필요경우 그결과 메인쓰레드에서 실행하므로 UI객체에 접근가능
* 새로만든 스레드에서 백그라운드 작업수행 doInBackground
* 백그라운드 들어가기전 onPreExecute
* 진행중일때 onProgressUpdate
* 백그라운드 작업끝난뒤 onPostExecute

|  |
| --- |
| **public** **class** MainActivity **extends** ActionBarActivity {  TextView textView01;  ProgressBar progress;  BackgroundTask task;  **int** value;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  textView01 = (TextView) findViewById(R.id.textView01);  progress = (ProgressBar) findViewById(R.id.progress);  // 실행 버튼 이벤트 처리  Button executeBtn = (Button) findViewById(R.id.executeBtn);  executeBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  // 새로운 Task 객체를 만들고 실행  task = **new** BackgroundTask();  task.execute(100);  }  });  // 취소 버튼 이벤트 처리  Button cancelBtn = (Button) findViewById(R.id.cancelBtn);  cancelBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  task.cancel(**true**);  }  });  }  /\*\*  \* 새로운 Task 객체를 정의  \*/  **class BackgroundTask extends AsyncTask<Integer , Integer , Integer> {**  **protected void onPreExecute() {**  **value = 0;**  **progress.setProgress(value);**  **}**  **protected Integer doInBackground(Integer ... values) {**  **while (isCancelled() == false) {**  **value++;**  **if (value >= 100) {**  **break;**  **} else {**  **publishProgress(value);**  **}**  **try {**  **Thread.*sleep*(100);**  **} catch (InterruptedException ex) {}**  **}**  **return value;**  **}**  **protected** **void** onProgressUpdate(Integer ... values) {  progress.setProgress(values[0].intValue());  textView01.setText("Current Value : " + values[0].toString());  }  **protected** **void** onPostExecute(Integer result) {  progress.setProgress(0);  textView01.setText("Finished.");  }  **protected** **void** onCancelled() {  progress.setProgress(0);  textView01.setText("Cancelled.");  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

애니메이션 쓰레드로 처리

|  |
| --- |
| **package** org.androidtown.animation.thread;  **public** **class** ThreadAnimationView **extends** SurfaceView **implements** Callback {  **private** ImageThread thread;  **public** ThreadAnimationView(Context context, AttributeSet attrs) {  **super**(context, attrs);  init(context);  }  **private** **void** init(Context context) {  SurfaceHolder holder = getHolder();  holder.addCallback(**this**);  thread = **new** ImageThread(context, holder);  }  **public** **void** surfaceCreated(SurfaceHolder holder) {  thread.start();  }  **public** **void** surfaceChanged(SurfaceHolder holder, **int** format, **int** width, **int** height) { }  **public** **void** surfaceDestroyed(SurfaceHolder holder) {  **try** {  thread.join();  } **catch** (InterruptedException ex) { }  }  **class** ImageThread **extends** Thread {  SurfaceHolder mHolder;  **private** **int** mImageWidth;  **private** **int** mImageHeight;  **private** **int** mCount = 0;  **private** Drawable mFrontImage[] = **new** Drawable[4];  **private** Bitmap mBackImage[] = **new** Bitmap[4];  **public** ImageThread(Context context, SurfaceHolder surfaceHolder) {  mHolder = surfaceHolder;  Resources res = context.getResources();  mFrontImage[0] = res.getDrawable(R.drawable.emo\_im\_laughing);  mFrontImage[1] = res.getDrawable(R.drawable.emo\_im\_happy);  mFrontImage[2] = res.getDrawable(R.drawable.emo\_im\_sad);  mFrontImage[3] = res.getDrawable(R.drawable.emo\_im\_crying);  mBackImage[0] = BitmapFactory.decodeResource(res,  R.drawable.gallery\_selected\_default);  mBackImage[1] = BitmapFactory.decodeResource(res,  R.drawable.gallery\_selected\_focused);  mBackImage[2] = BitmapFactory.decodeResource(res,  R.drawable.gallery\_selected\_pressed);  mBackImage[3] = BitmapFactory.decodeResource(res,  R.drawable.gallery\_unselected\_default);  mImageWidth = mFrontImage[1].getIntrinsicWidth();  mImageHeight = mFrontImage[1].getIntrinsicHeight();  }  **public** **void** run() {  **while** (**true**) {  Canvas c = **null**;  **try** {  c = mHolder.lockCanvas(**null**);  **synchronized** (mHolder) {  doDraw(c);  mCount++;  sleep(100);  }  } **catch** (InterruptedException ex) {  Log.e("ThreadAnimationView", "Exception in thread.", ex);  } **finally** {  **if** (c != **null**) {  mHolder.unlockCanvasAndPost(c);  }  }  }  }  **private** **void** doDraw(Canvas canvas) {  **if** (canvas != **null**) {  canvas.drawBitmap(mBackImage[mCount % 4], 20, 270, **null**);    mFrontImage[mCount % 4].setBounds(70, 320,  70 + mImageWidth, 320 + mImageHeight);  mFrontImage[mCount % 4].draw(canvas);  }  }  }  } |

ImageSwitcher 를 이용한 이미지 애니메이션

|  |
| --- |
| **public** **class** MainActivity **extends** ActionBarActivity {  ImageSwitcher switcher;  Handler mHandler = **new** Handler();  ImageThread thread;  **boolean** running;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // 시작 버튼 이벤트 처리  Button startBtn = (Button) findViewById(R.id.startBtn);  startBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  startAnimation();  }  });  // 중지 버튼 이벤트 처리  Button stopBtn = (Button) findViewById(R.id.stopBtn);  stopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  stopAnimation();  }  });  switcher = (ImageSwitcher) findViewById(R.id.switcher);  switcher.setVisibility(View.***INVISIBLE***);  switcher.setFactory(**new** ViewSwitcher.ViewFactory() {  **public** View makeView() {  ImageView imageView = **new** ImageView(getApplicationContext());  imageView.setBackgroundColor(0xFF000000);  imageView.setScaleType(ImageView.ScaleType.***CENTER\_INSIDE***);  imageView.setLayoutParams(**new** ImageSwitcher.LayoutParams(  ViewGroup.LayoutParams.***MATCH\_PARENT***, ViewGroup.LayoutParams.***MATCH\_PARENT***));  **return** imageView;  }  });  }  /\*\*  \* 애니메이션 시작  \*/  **private** **void** startAnimation() {  switcher.setVisibility(View.***VISIBLE***);  thread = **new** ImageThread();  thread.start();  }  /\*\*  \* 애니메이션 중지  \*/  **private** **void** stopAnimation() {  running = **false**;  **try** {  thread.join();  } **catch**(InterruptedException ex) { }  switcher.setVisibility(View.***INVISIBLE***);  }  /\*\*  \* 이미지 처리 스레드  \* **@author** michael  \*  \*/  **class** ImageThread **extends** Thread {  **int** duration = 250;  **final** **int** imageId[] = { R.drawable.emo\_im\_crying,  R.drawable.emo\_im\_happy,  R.drawable.emo\_im\_laughing,  R.drawable.emo\_im\_surprised };  **int** currentIndex = 0;  **public** **void** run() {  running = **true**;  **while** (running) {  **synchronized** (**this**) {  mHandler.post(**new** Runnable() {  **public** **void** run() {  switcher.setImageResource(imageId[currentIndex]);  }  });  currentIndex++;  **if** (currentIndex == imageId.length) {  currentIndex = 0;  }  **try** {  Thread.*sleep*(duration);  } **catch** (InterruptedException ex) { }  }  }  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

프레임 단위 애니메이션 적용

|  |
| --- |
| **public** **class** MainActivity **extends** ActionBarActivity {  AnimationDrawable animDrawable = **null**;  ImageView imageView01;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  imageView01 = (ImageView)findViewById(R.id.imageView01);  // 시작 버튼 이벤트 처리  Button startBtn = (Button) findViewById(R.id.startBtn);  startBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  startAnimation();  }  });  // 중지 버튼 이벤트 처리  Button stopBtn = (Button) findViewById(R.id.stopBtn);  stopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  stopAnimation();  }  });  // 이미지 가져오기  Resources res = getResources();  BitmapDrawable frame01 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_crying);  BitmapDrawable frame02 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_happy);  BitmapDrawable frame03 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_laughing);  BitmapDrawable frame04 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_surprised);  // 프레임으로 추가하기  **int** duration = 250;  **animDrawable = new AnimationDrawable();**  **animDrawable.setOneShot(false);**  **animDrawable.addFrame(frame01, duration);**  **animDrawable.addFrame(frame02, duration);**  **animDrawable.addFrame(frame03, duration);**  **animDrawable.addFrame(frame04, duration);**  }  /\*\*  \* 애니메이션 시작  \*/  **private** **void** startAnimation() {  **imageView01.setBackground(animDrawable);**  **animDrawable.setVisible(true, true);**  **animDrawable.start();**  }  /\*\*  \* 애니메이션 중지  \*/  **private** **void** stopAnimation() {  animDrawable.stop();  animDrawable.setVisible(**false**, **false**);  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| **public** **class** MainActivity2 **extends** ActionBarActivity {  AnimationDrawable animDrawable = **null**;  ImageView imageView01;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  imageView01 = (ImageView)findViewById(R.id.imageView01);  // 시작 버튼 이벤트 처리  Button startBtn = (Button) findViewById(R.id.startBtn);  startBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  startAnimation();  }  });  // 중지 버튼 이벤트 처리  Button stopBtn = (Button) findViewById(R.id.stopBtn);  stopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  stopAnimation();  }  });  // 이미지 가져오기  Resources res = getResources();  BitmapDrawable frame01 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_crying);  BitmapDrawable frame02 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_happy);  BitmapDrawable frame03 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_laughing);  BitmapDrawable frame04 = (BitmapDrawable)res.~~getDrawable~~(R.drawable.emo\_im\_surprised);  // 프레임으로 추가하기  **int** duration = 250;  animDrawable = **new** AnimationDrawable();  animDrawable.setOneShot(**false**);  animDrawable.addFrame(frame01, duration);  animDrawable.addFrame(frame02, duration);  animDrawable.addFrame(frame03, duration);  animDrawable.addFrame(frame04, duration);  }  /\*\*  \* 애니메이션 시작  \*/  **private** **void** startAnimation() {  imageView01.setBackground(animDrawable);  animDrawable.setVisible(**true**, **true**);  animDrawable.start();  }  /\*\*  \* 애니메이션 중지  \*/  **private** **void** stopAnimation() {  animDrawable.stop();  animDrawable.setVisible(**false**, **false**);  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

트윈 애니메이션을 적용

|  |
| --- |
| /\*\*  \* 트윈 애니메이션을 적용하는 간단한 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // 첫번째 버튼 이벤트 처리  Button scaleBtn = (Button) findViewById(R.id.scaleBtn);  scaleBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  Animation anim = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.scale);  v.startAnimation(anim);  }  });  // 두번째 버튼 이벤트 처리  Button scale2Btn = (Button) findViewById(R.id.scale2Btn);  scale2Btn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  Animation anim = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.scale2);  v.startAnimation(anim);  }  });  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| Scale.xml  <?xml version="1.0" encoding="utf-8"?>  <scale xmlns:android="http://schemas.android.com/apk/res/android"  android:duration="2500"  android:pivotX="50%"  android:pivotY="50%"  android:fromXScale="1.0"  android:fromYScale="1.0"  android:toXScale="2.0"  android:toYScale="2.0"  /> |
| Alpha  <?xml version="1.0" encoding="utf-8"?>  <alpha xmlns:android="http://schemas.android.com/apk/res/android"  android:fromAlpha="0.0"  android:toAlpha="1.0"  android:duration="10000"  /> |
| Rotate  <?xml version="1.0" encoding="utf-8"?>  <rotate xmlns:android="http://schemas.android.com/apk/res/android"  android:fromDegrees="0"  android:toDegrees="360"  android:pivotX="50%"  android:pivotY="50%"  android:duration="10000"  /> |
| Scale2  <?xml version="1.0" encoding="utf-8"?>  <set xmlns:android="http://schemas.android.com/apk/res/android"  >  <scale  android:duration="2500"  android:pivotX="50%"  android:pivotY="50%"  android:fromXScale="1.0"  android:fromYScale="1.0"  android:toXScale="2.0"  android:toYScale="2.0"  />  <scale  android:startOffset="2500"  android:duration="2500"  android:pivotX="50%"  android:pivotY="50%"  android:fromXScale="1.0"  android:fromYScale="1.0"  android:toXScale="0.5"  android:toYScale="0.5"  />  </set> |
| Translate  <?xml version="1.0" encoding="utf-8"?>  <translate xmlns:android="http://schemas.android.com/apk/res/android"  android:fromXDelta="0%p"  android:toXDelta="-100%p"  android:duration="20000"  android:repeatCount="-1"  android:fillAfter="true"  /> |
| **애니메이션에 setAnimationListener 등록으로 애니끝나거나 시작되거나 리스너 걸수 있다.** |

막대그래프 모양처럼 만들어 애니메이션을 적용

|  |
| --- |
| **public** **class** MainActivity **extends** ActionBarActivity {  LinearLayout mainLayout;  Resources res;  Animation growAnim;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  res = getResources();  growAnim = AnimationUtils.*loadAnimation*(**this**, R.anim.grow);  mainLayout = (LinearLayout)findViewById(R.id.mainLayout);  // 아이템 추가  addItem("Apple", 80);  addItem("Orange", 100);  addItem("Kiwi", 40);  }  /\*\*  \* 아이템 추가 메소드  \*  \* **@param** name  \* **@param** value  \*/  **private** **void** addItem(String name, **int** value) {  LinearLayout itemLayout = **new** LinearLayout(**this**);  itemLayout.setOrientation(LinearLayout.***HORIZONTAL***);  LinearLayout.LayoutParams params = **new** LinearLayout.LayoutParams(  LinearLayout.LayoutParams.***WRAP\_CONTENT***,  LinearLayout.LayoutParams.***WRAP\_CONTENT***);  LinearLayout.LayoutParams params2 = **new** LinearLayout.LayoutParams(  LinearLayout.LayoutParams.***WRAP\_CONTENT***,  LinearLayout.LayoutParams.***WRAP\_CONTENT***);  LinearLayout.LayoutParams params3 = **new** LinearLayout.LayoutParams(  LinearLayout.LayoutParams.***WRAP\_CONTENT***,  LinearLayout.LayoutParams.***WRAP\_CONTENT***);  // 텍스트뷰 추가  TextView textView = **new** TextView(**this**);  textView.setText(name);  params.width = 180;  params.setMargins(0, 4, 0, 4);  itemLayout.addView(textView, params);  // 프로그레스바 추가  ProgressBar proBar = **new** ProgressBar(**this**, **null**, android.R.attr.***progressBarStyleHorizontal***);  proBar.setIndeterminate(**false**);  proBar.setMax(100);  proBar.setProgress(100);  proBar.setAnimation(growAnim);  params2.height = 80;  params2.width = value \* 3;  params2.gravity = Gravity.***LEFT***;  itemLayout.addView(proBar, params2);  mainLayout.addView(itemLayout, params3);  }  /\*\*  \* 화면에 보여지기 전에 호출되는 메소드  \*/  **public void onWindowFocusChanged(boolean hasFocus) {**  **super**.onWindowFocusChanged(hasFocus);  Toast.*makeText*(**this**, "onWindowFocusChanged : " + hasFocus, Toast.***LENGTH\_SHORT***).show();  **if** (hasFocus) {  growAnim.start();  } **else** {  growAnim.reset();  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| **Grow.xml**  **<?xml version="1.0" encoding="utf-8"?>**  **<set xmlns:android="http://schemas.android.com/apk/res/android">**  **<scale android:fromXScale="0.0" android:toXScale="1.0"**  **android:fromYScale="1.0" android:toYScale="1.0"**  **android:pivotX="0%p" android:pivotY="50%p"**  **android:duration="3000"**  **/>**  **</set>** |

네트워크

|  |
| --- |
| /\*\*  \* 안드로이드에서 소켓 클라이언트로 연결하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  EditText input01;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  input01 = (EditText) findViewById(R.id.input01);  // 버튼 이벤트 처리  Button button01 = (Button) findViewById(R.id.button01);  button01.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  String addr = input01.getText().toString().trim();  ConnectThread thread = **new** ConnectThread(addr);  thread.start();  }  });  }  /\*\*  \* 소켓 연결할 스레드 정의  \*/  **class** ConnectThread **extends** Thread {  String hostname;  **public** ConnectThread(String addr) {  hostname = addr;  }  **public** **void** run() {  **try** {  **int** port = 11001;  Socket sock = **new** Socket(hostname, port);  ObjectOutputStream outstream = **new** ObjectOutputStream(sock.getOutputStream());  outstream.writeObject("Hello AndroidTown on Android");  outstream.flush();  ObjectInputStream instream = **new** ObjectInputStream(sock.getInputStream());  String obj = (String) instream.readObject();  Log.*d*("MainActivity", "서버에서 받은 메시지 : " + obj);  sock.close();  } **catch**(Exception ex) {  ex.printStackTrace();  }  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

HTTP호출

|  |
| --- |
| /\*\*  \* HTTP 요청 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  EditText input01;  TextView txtMsg;  **public** **static** String *defaultUrl* = "http://m.naver.com";  Handler handler = **new** Handler();  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  input01 = (EditText) findViewById(R.id.input01);  input01.setText(*defaultUrl*);  txtMsg = (TextView) findViewById(R.id.txtMsg);  // 버튼 이벤트 처리  Button requestBtn = (Button) findViewById(R.id.requestBtn);  requestBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  String urlStr = input01.getText().toString();  ConnectThread thread = **new** ConnectThread(urlStr);  thread.start();  }  });  }  /\*\*  \* 소켓 연결할 스레드 정의  \*/  **class** ConnectThread **extends** Thread {  String urlStr;  **public** ConnectThread(String inStr) {  urlStr = inStr;  }  **public** **void** run() {  **try** {  **final** String output = request(urlStr);  handler.post(**new** Runnable() {  **public** **void** run() {  txtMsg.setText(output);  }  });  } **catch**(Exception ex) {  ex.printStackTrace();  }  }  **private** String request(String urlStr) {  StringBuilder output = **new** StringBuilder();  **try** {  URL url = **new** URL(urlStr);  HttpURLConnection conn = (HttpURLConnection)url.openConnection();  **if** (conn != **null**) {  conn.setConnectTimeout(10000);  conn.setRequestMethod("GET");  conn.setDoInput(**true**);  conn.setDoOutput(**true**);  **int** resCode = conn.getResponseCode();  **if** (resCode == HttpURLConnection.***HTTP\_OK***) {  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream())) ;  String line = **null**;  **while**(**true**) {  line = reader.readLine();  **if** (line == **null**) {  **break**;  }  output.append(line + "\n");  }  reader.close();  conn.disconnect();  }  }  } **catch**(Exception ex) {  Log.*e*("SampleHTTP", "Exception in processing response.", ex);  ex.printStackTrace();  }  **return** output.toString();  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

RSS받아오기

|  |
| --- |
| mainActivity  **public** **class** MainActivity **extends** ActionBarActivity {  **private** **static** **final** String ***TAG*** = "MainActivity";  **private** **static** String *rssUrl* = "https://news.google.co.kr/news?pz=1&cf=all&ned=kr&hl=ko&topic=e&output=rss";  ProgressDialog progressDialog;  Handler handler = **new** Handler();  RSSListView list;  RSSListAdapter adapter;  ArrayList<RSSNewsItem> newsItemList;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // create a ListView instance  ViewGroup.LayoutParams params = **new** ViewGroup.LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***, ViewGroup.LayoutParams.***MATCH\_PARENT***);  list = **new** RSSListView(**this**);  adapter = **new** RSSListAdapter(**this**);  list.setAdapter(adapter);  list.setOnDataSelectionListener(**new** OnDataSelectionListener() {  **public** **void** onDataSelected(AdapterView parent, View v, **int** position, **long** id) {  RSSNewsItem curItem = (RSSNewsItem) adapter.getItem(position);  String curTitle = curItem.getTitle();  Toast.*makeText*(getApplicationContext(), "Selected : " + curTitle, Toast.***LENGTH\_LONG***).show();  }  });  newsItemList = **new** ArrayList<RSSNewsItem>();  LinearLayout mainLayout = (LinearLayout) findViewById(R.id.mainLayout);  mainLayout.addView(list, params);  **final** EditText edit01 = (EditText) findViewById(R.id.edit01);  edit01.setText(*rssUrl*);  Button show\_btn = (Button) findViewById(R.id.show\_btn);  show\_btn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  String inputStr = edit01.getText().toString();  showRSS(inputStr);  }  });  }  **private** **void** showRSS(String urlStr) {  **try** {  progressDialog = ProgressDialog.*show*(**this**, "RSS Refresh", "RSS 정보 업데이트 중...", **true**, **true**);  RefreshThread thread = **new** RefreshThread(urlStr);  thread.start();  } **catch** (Exception e) {  Log.*e*(***TAG***, "Error", e);  }  }  **class** RefreshThread **extends** Thread {  String urlStr;  **public** RefreshThread(String str) {  urlStr = str;  }  **public** **void** run() {  **try** {  DocumentBuilderFactory builderFactory = DocumentBuilderFactory.*newInstance*();  DocumentBuilder builder = builderFactory.newDocumentBuilder();  URL urlForHttp = **new** URL(urlStr);  InputStream instream = getInputStreamUsingHTTP(urlForHttp);  Document document = builder.parse(instream);  **int** countItem = processDocument(document);  Log.*d*(***TAG***, countItem + " news item processed.");  // post for the display of fetched RSS info.  handler.post(updateRSSRunnable);  } **catch**(Exception ex) {  ex.printStackTrace();  }  }  }  **public** InputStream getInputStreamUsingHTTP(URL url)  **throws** Exception  {  HttpURLConnection conn = (HttpURLConnection) url.openConnection();  conn.setRequestMethod("POST");  conn.setDoInput(**true**);  conn.setDoOutput(**true**);  **int** resCode = conn.getResponseCode();  Log.*d*(***TAG***, "Response Code : " + resCode);  InputStream instream = conn.getInputStream();  **return** instream;  }  /\*\*  \* process DOM document for RSS  \*  \* **@param** doc  \*/  **private** **int** processDocument(Document doc) {  newsItemList.clear();  Element docEle = doc.getDocumentElement();  NodeList nodelist = docEle.getElementsByTagName("item");  **int** count = 0;  **if** ((nodelist != **null**) && (nodelist.getLength() > 0)) {  **for** (**int** i = 0; i < nodelist.getLength(); i++) {  RSSNewsItem newsItem = dissectNode(nodelist, i);  **if** (newsItem != **null**) {  newsItemList.add(newsItem);  count++;  }  }  }  **return** count;  }  **private** RSSNewsItem dissectNode(NodeList nodelist, **int** index) {  RSSNewsItem newsItem = **null**;  **try** {  Element entry = (Element) nodelist.item(index);  Element title = (Element) entry.getElementsByTagName("title").item(0);  Element link = (Element) entry.getElementsByTagName("link").item(0);  Element description = (Element) entry.getElementsByTagName("description").item(0);  NodeList pubDataNode = entry.getElementsByTagName("pubDate");  **if** (pubDataNode == **null**) {  pubDataNode = entry.getElementsByTagName("dc:date");  }  Element pubDate = (Element) pubDataNode.item(0);  Element author = (Element) entry.getElementsByTagName("author").item(0);  Element category = (Element) entry.getElementsByTagName("category").item(0);  String titleValue = **null**;  **if** (title != **null**) {  Node firstChild = title.getFirstChild();  **if** (firstChild != **null**) {  titleValue = firstChild.getNodeValue();  }  }  String linkValue = **null**;  **if** (link != **null**) {  Node firstChild = link.getFirstChild();  **if** (firstChild != **null**) {  linkValue = firstChild.getNodeValue();  }  }  String descriptionValue = **null**;  **if** (description != **null**) {  Node firstChild = description.getFirstChild();  **if** (firstChild != **null**) {  descriptionValue = firstChild.getNodeValue();  }  }  String pubDateValue = **null**;  **if** (pubDate != **null**) {  Node firstChild = pubDate.getFirstChild();  **if** (firstChild != **null**) {  pubDateValue = firstChild.getNodeValue();  }  }  String authorValue = **null**;  **if** (author != **null**) {  Node firstChild = author.getFirstChild();  **if** (firstChild != **null**) {  authorValue = firstChild.getNodeValue();  }  }  String categoryValue = **null**;  **if** (category != **null**) {  Node firstChild = category.getFirstChild();  **if** (firstChild != **null**) {  categoryValue = firstChild.getNodeValue();  }  }  Log.*d*(***TAG***, "item node : " + titleValue + ", " + linkValue + ", " + descriptionValue +  ", " + pubDateValue + ", " + authorValue + ", " + categoryValue);  newsItem = **new** RSSNewsItem(titleValue, linkValue, descriptionValue,  pubDateValue, authorValue, categoryValue);  } **catch** (DOMException e) {  e.printStackTrace();  }  **return** newsItem;  }  Runnable updateRSSRunnable = **new** Runnable() {  **public** **void** run() {  **try** {  Resources res = getResources();  Drawable rssIcon = res.~~getDrawable~~(R.drawable.rss\_icon);  **for** (**int** i = 0; i < newsItemList.size(); i++) {  RSSNewsItem newsItem = (RSSNewsItem) newsItemList.get(i);  newsItem.setIcon(rssIcon);  adapter.addItem(newsItem);  }  adapter.notifyDataSetChanged();  progressDialog.dismiss();  } **catch**(Exception ex) {  ex.printStackTrace();  }  }  };  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |
| RssListView  **public** **class** RSSListView **extends** ListView {  /\*\*  \* DataAdapter for this instance  \*/  **private** RSSListAdapter adapter;    /\*\*  \* Listener for data selection  \*/  **private** OnDataSelectionListener selectionListener;    **public** RSSListView(Context context) {  **super**(context);  init();  }  **public** RSSListView(Context context, AttributeSet attrs) {  **super**(context, attrs);  init();  }    /\*\*  \* set initial properties  \*/  **private** **void** init() {  // set OnItemClickListener for processing OnDataSelectionListener  setOnItemClickListener(**new** OnItemClickAdapter());  }  /\*\*  \* set DataAdapter  \*  \* **@param** adapter  \*/  **public** **void** setAdapter(BaseAdapter adapter) {  **super**.setAdapter(adapter);  }  /\*\*  \* get DataAdapter  \*  \* **@return**  \*/  **public** BaseAdapter getAdapter() {  **return** (BaseAdapter)**super**.getAdapter();  }    /\*\*  \* set OnDataSelectionListener  \*  \* **@param** listener  \*/  **public** **void** setOnDataSelectionListener(OnDataSelectionListener listener) {  **this**.selectionListener = listener;  }  /\*\*  \* get OnDataSelectionListener  \*  \* **@return**  \*/  **public** OnDataSelectionListener getOnDataSelectionListener() {  **return** selectionListener;  }    **class** OnItemClickAdapter **implements** OnItemClickListener {    **public** OnItemClickAdapter() {    }  **public** **void** onItemClick(AdapterView parent, View v, **int** position, **long** id) {    **if** (selectionListener == **null**) {  **return**;  }    // get row and column  **int** rowIndex = -1;  **int** columnIndex = -1;    // call the OnDataSelectionListener method  selectionListener.onDataSelected(parent, v, position, id);    }    }    } |
| OnDataSelectionListener  **public** **interface** OnDataSelectionListener {  **public** **void** onDataSelected(AdapterView parent, View v, **int** position, **long** id);    } |
| RSSListAdapter  **public** **class** RSSListAdapter **extends** BaseAdapter {  **private** Context mContext;  **private** List<RSSNewsItem> mItems = **new** ArrayList<RSSNewsItem>();  **public** RSSListAdapter(Context context) {  mContext = context;  }  **public** **void** addItem(RSSNewsItem it) {  mItems.add(it);  }  **public** **void** setListItems(List<RSSNewsItem> lit) {  mItems = lit;  }  **public** **int** getCount() {  **return** mItems.size();  }  **public** Object getItem(**int** position) {  **return** mItems.get(position);  }  **public** **boolean** areAllItemsSelectable() {  **return** **false**;  }  **public** **boolean** isSelectable(**int** position) {  **return** **true**;  }  **public** **long** getItemId(**int** position) {  **return** position;  }  **public** View getView(**int** position, View convertView, ViewGroup parent) {  RSSNewsItemView itemView;  **if** (convertView == **null**) {  itemView = **new** RSSNewsItemView(mContext, mItems.get(position));  } **else** {  itemView = (RSSNewsItemView) convertView;    itemView.setIcon(mItems.get(position).getIcon());  itemView.setText(0, mItems.get(position).getTitle());  itemView.setText(1, mItems.get(position).getPubDate());  itemView.setText(2, mItems.get(position).getCategory());  itemView.setText(3, mItems.get(position).getDescription());  }  **return** itemView;  }  } |
| RSSNewsItem  **public** **class** RSSNewsItem {  **private** String title;  **private** String link;  **private** String description;  **private** String pubDate;  **private** String author;  **private** String category;    **private** Drawable mIcon;    /\*\*  \* Initialize with icon and data array  \*/  **public** RSSNewsItem() {  }  /\*\*  \* Initialize with icon and strings  \*/  **public** RSSNewsItem(String title, String link, String description, String pubDate, String author, String category) {  **this**.title = title;  **this**.link = link;  **this**.description = description;  **this**.pubDate = pubDate;  **this**.author = author;  **this**.category = category;  }  /\*\*  \* Set icon  \*  \* **@param** icon  \*/  **public** **void** setIcon(Drawable icon) {  mIcon = icon;  }  /\*\*  \* Get icon  \*  \* **@return**  \*/  **public** Drawable getIcon() {  **return** mIcon;  }  **public** String getTitle() {  **return** title;  }  **public** **void** setTitle(String title) {  **this**.title = title;  }  **public** String getLink() {  **return** link;  }  **public** **void** setLink(String link) {  **this**.link = link;  }  **public** String getDescription() {  **return** description;  }  **public** **void** setDescription(String description) {  **this**.description = description;  }  **public** String getPubDate() {  **return** pubDate;  }  **public** **void** setPubDate(String pubDate) {  **this**.pubDate = pubDate;  }  **public** String getAuthor() {  **return** author;  }  **public** **void** setAuthor(String author) {  **this**.author = author;  }  **public** String getCategory() {  **return** category;  }  **public** **void** setCategory(String category) {  **this**.category = category;  }  /\*\*  \* Compare with the input object  \*  \* **@param** other  \* **@return**  \*/  **public** **int** compareTo(RSSNewsItem other) {  **if** (title.equals(other.getTitle())) {  **return** -1;  } **else** **if** (link.equals(other.getLink())) {  **return** -1;  } **else** **if** (description.equals(other.getDescription())) {  **return** -1;  } **else** **if** (pubDate.equals(other.getPubDate())) {  **return** -1;  } **else** **if** (author.equals(other.getAuthor())) {  **return** -1;  } **else** **if** (category.equals(other.getCategory())) {  **return** -1;  }    **return** 0;  }  } |
| RSSNewsItemView  **public** **class** RSSNewsItemView **extends** LinearLayout {  /\*\*  \* Icon  \*/  **private** ImageView mIcon;  /\*\*  \* TextView 01  \*/  **private** TextView mText01;  /\*\*  \* TextView 02  \*/  **private** TextView mText02;  /\*\*  \* TextView 03  \*/  **private** TextView mText03;  /\*\*  \* WebView 04  \*/  **private** WebView mText04;  **public** RSSNewsItemView(Context context, RSSNewsItem aItem) {  **super**(context);  // Layout Inflation  LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  inflater.inflate(R.layout.listitem, **this**, **true**);  // Set Icon  mIcon = (ImageView) findViewById(R.id.iconItem);  mIcon.setImageDrawable(aItem.getIcon());  // Set Text 01  mText01 = (TextView) findViewById(R.id.dataItem01);  mText01.setText(aItem.getTitle());  // Set Text 02  mText02 = (TextView) findViewById(R.id.dataItem02);  mText02.setText(aItem.getPubDate());  // Set Text 03  mText03 = (TextView) findViewById(R.id.dataItem03);  String category = aItem.getCategory();  **if** (category != **null**) {  mText03.setText(category);  }  // Set Text 04  mText04 = (WebView) findViewById(R.id.dataItem04);  //mText04.setText(aItem.getDescription());    setTextToWebView(aItem.getDescription());    }  /\*\*  \* set Text  \*  \* **@param** index  \* **@param** data  \*/  **public** **void** setText(**int** index, String data) {  **if** (index == 0) {  mText01.setText(data);  } **else** **if** (index == 1) {  mText02.setText(data);  } **else** **if** (index == 2) {  mText03.setText(data);  } **else** **if** (index == 3) {  //mText04.setText(data);    setTextToWebView(data);    } **else** {  **throw** **new** IllegalArgumentException();  }  }    **private** **void** setTextToWebView(String msg) {  Log.*d*("RSSNewsItemView", "setTextToWebView() called.");    //String outData = "<meta http-equiv='Content-Type' content='text/html; charset=utf-8' /><html><body>"  // + msg  // + "</body></html>";  String outData = msg;    // 이미지 src 에서 http:// 가 아닌 // 일 경우의 대체  outData = outData.replace("\"//", "\"http://");    mText04.loadDataWithBaseURL("http://localhost/", outData, "text/html", "utf-8", **null**);  }    /\*\*  \* set Icon  \*  \* **@param** icon  \*/  **public** **void** setIcon(Drawable icon) {  mIcon.setImageDrawable(icon);  }  } |
|  |

XML-RPC를 사용하는 방법

Remote Procedure Call

서버에있는 함수를 클라이언트에 있는것처럼 쓰자-\_-;

JSON-RPC도있다.

30일차..

데이터 베이스

|  |
| --- |
| /\*\*  \* 데이터베이스를 사용하는 가장 기본적인 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  String databaseName;  String tableName;  TextView status;  **boolean** databaseCreated = **false**;  **boolean** tableCreated = **false**;  **SQLiteDatabase db;**  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **final** EditText databaseNameInput = (EditText) findViewById(R.id.databaseNameInput);  **final** EditText tableNameInput = (EditText) findViewById(R.id.tableNameInput);  Button createDatabaseBtn = (Button) findViewById(R.id.createDatabaseBtn);  createDatabaseBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  databaseName = databaseNameInput.getText().toString();  createDatabase(databaseName);  }  });  Button createTableBtn = (Button) findViewById(R.id.createTableBtn);  createTableBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  tableName = tableNameInput.getText().toString();  createTable(tableName);  **int** count = insertRecord(tableName);  println(count + " records inserted.");  }  });  status = (TextView) findViewById(R.id.status);  }  **private** **void** createDatabase(String name) {  println("creating database [" + name + "].");  **try** {  db = **openOrCreateDatabase**(  name,  Activity.***MODE\_PRIVATE***,  **null**);  databaseCreated = **true**;  println("database is created.");  } **catch**(Exception ex) {  ex.printStackTrace();  println("database is not created.");  }  }  **private** **void** createTable(String name) {  println("creating table [" + name + "].");  db.execSQL("create table if not exists " + name + "("  + " \_id integer PRIMARY KEY autoincrement, "  + " name text, "  + " age integer, "  + " phone text);" );  tableCreated = **true**;  }  **private** **int** insertRecord(String name) {  println("inserting records into table " + name + ".");  **int** count = 3;  db.execSQL( "insert into " + name + "(name, age, phone) values ('John', 20, '010-7788-1234');" );  db.execSQL( "insert into " + name + "(name, age, phone) values ('Mike', 35, '010-8888-1111');" );  db.execSQL( "insert into " + name + "(name, age, phone) values ('Sean', 26, '010-6677-4321');" );  **return** count;  }  /\*\*  \* insert records using parameters  \*/  **private** **int** insertRecordParam(String name) {  println("inserting records using parameters.");  **int** count = 1;  ContentValues recordValues = **new** ContentValues();  recordValues.put("name", "Rice");  recordValues.put("age", 43);  recordValues.put("phone", "010-3322-9811");  **int** rowPosition = (**int**) db.insert(name, **null**, recordValues);  **return** count;  }  /\*\*  \* update records using parameters  \*/  **private** **int** updateRecordParam(String name) {  println("updating records using parameters.");  ContentValues recordValues = **new** ContentValues();  recordValues.put("age", 43);  String[] whereArgs = {"Rice"};  **int** rowAffected = db.update(name,  recordValues,  "name = ?",  whereArgs);  **return** rowAffected;  }  /\*\*  \* delete records using parameters  \*/  **private** **int** deleteRecordParam(String name) {  println("deleting records using parameters.");  String[] whereArgs = {"Rice"};  **int** rowAffected = db.delete(name,  "name = ?",  whereArgs);  **return** rowAffected;  }  **private** **void** println(String msg) {  Log.*d*("SampleDatabase", msg);  status.append("\n" + msg);  }  …  } |

데이터베이스를 조회

|  |
| --- |
| /\*\*  \* 데이터베이스를 조회하는 방법을 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **private** TextView status;  **public** **static** **final** String ***TAG*** = "MainActivity";  **private** **static** String *DATABASE\_NAME* = **null**;  **private** **static** String *TABLE\_NAME* = "employee";  **private** **static** **int** *DATABASE\_VERSION* = 1;  **private** DatabaseHelper dbHelper;  **private** SQLiteDatabase db;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  status = (TextView) findViewById(R.id.status);  **final** EditText input01 = (EditText) findViewById(R.id.input01);  Button queryBtn = (Button) findViewById(R.id.queryBtn);  queryBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  *DATABASE\_NAME* = input01.getText().toString();  **boolean** isOpen = openDatabase();  **if** (isOpen) {  executeRawQuery();  executeRawQueryParam();  }  }  });  }  **private** **boolean** openDatabase() {  println("opening database [" + *DATABASE\_NAME* + "].");  dbHelper = **new** DatabaseHelper(**this**);  db = dbHelper.getWritableDatabase();  **return** **true**;  }  **private** **void** executeRawQuery() {  println("\nexecuteRawQuery called.\n");  Cursor c1 = db.rawQuery("select count(\*) as Total from " + *TABLE\_NAME*, **null**);  println("cursor count : " + c1.getCount());  c1.moveToNext();  println("record count : " + c1.getInt(0));  c1.close();  }  **private** **void** executeRawQueryParam() {  println("\nexecuteRawQueryParam called.\n");  String SQL = "select name, age, phone "  + " from " + *TABLE\_NAME*  + " where age > ?";  String[] args= {"30"};  **Cursor c1 = db.rawQuery(SQL, args);**  **int recordCount = c1.getCount();**  **println("cursor count : " + recordCount + "\n");**  **for (int i = 0; i < recordCount; i++) {**  **c1.moveToNext();**  **String name = c1.getString(0);**  **int age = c1.getInt(1);**  **String phone = c1.getString(2);**  **println("Record #" + i + " : " + name + ", " + age + ", " + phone);**  **}**  **c1.close();**  }  **private** **void** executeRawQueryParam2() {  println("\nexecuteRawQueryParam2 called.\n");  String[] columns = {"name", "age", "phone"};  String whereStr = "where age > ?";  String[] whereParams = {"30"};  Cursor c1 = db.query(*TABLE\_NAME*, columns,  whereStr, whereParams,  **null**, **null**, **null**);  **int** recordCount = c1.getCount();  println("cursor count : " + recordCount + "\n");  **for** (**int** i = 0; i < recordCount; i++) {  c1.moveToNext();  String name = c1.getString(0);  **int** age = c1.getInt(1);  String phone = c1.getString(2);  println("Record #" + i + " : " + name + ", " + age + ", " + phone);  }  c1.close();  }  **private** **void** println(String msg) {  Log.*d*(***TAG***, msg);  status.append("\n" + msg);  }  **private** **class** DatabaseHelper **extends** SQLiteOpenHelper {  **public** DatabaseHelper(Context context) {  **super**(context, *DATABASE\_NAME*, **null**, *DATABASE\_VERSION*);  }  **public** **void** onCreate(SQLiteDatabase db) {  println("creating table [" + *TABLE\_NAME* + "].");  **try** {  String DROP\_SQL = "drop table if exists " + *TABLE\_NAME*;  db.execSQL(DROP\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in DROP\_SQL", ex);  }  String CREATE\_SQL = "create table " + *TABLE\_NAME* + "("  + " \_id integer PRIMARY KEY autoincrement, "  + " name text, "  + " age integer, "  + " phone text)";  **try** {  db.execSQL(CREATE\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in CREATE\_SQL", ex);  }  println("inserting records.");  **try** {  db.execSQL( "insert into " + *TABLE\_NAME* + "(name, age, phone) values ('John', 20, '010-7788-1234');" );  db.execSQL( "insert into " + *TABLE\_NAME* + "(name, age, phone) values ('Mike', 35, '010-8888-1111');" );  db.execSQL( "insert into " + *TABLE\_NAME* + "(name, age, phone) values ('Sean', 26, '010-6677-4321');" );  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in insert SQL", ex);  }  }  **public** **void** onOpen(SQLiteDatabase db) {  println("opened database [" + *DATABASE\_NAME* + "].");  }  **public** **void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  Log.*w*(***TAG***, "Upgrading database from version " + oldVersion + " to " + newVersion + ".");  }  }  …  } |

테이블 갑자기 바뀌었을 때.. 데이터 베이스 헬퍼로 처리하자

DatabaseHelper

* 처음사용자 만들어질때 onCreate() 호출됨
* 버전이 바뀌어 업그레이드 될떄는 onUpgrade호출됨

|  |
| --- |
| /\*\*  \* 데이터베이스 헬퍼를 사용하는 가장 기본적인 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **public** **static** **final** String ***TAG*** = "MainActivity";  TextView status;  CustomerDatabase database;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // open database  **if** (database != **null**) {  database.close();  database = **null**;  }  database = CustomerDatabase.getInstance(**this**);  **boolean** isOpen = database.open();  **if** (isOpen) {  Log.*d*(***TAG***, "Customer database is open.");  } **else** {  Log.*d*(***TAG***, "Customer database is not open.");  }  Button insertBtn = (Button) findViewById(R.id.insertBtn);  insertBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **int** count = insertRecord();  println(count + " records inserted.");  }  });  Button deleteBtn = (Button) findViewById(R.id.deleteBtn);  deleteBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  deleteRecord();  println("records deleted.");  }  });  status = (TextView) findViewById(R.id.status);  }  **private** **int** insertRecord() {  println("inserting records.");  **int** count = 3;  database.execSQL( "insert into CUSTOMER\_INFO(name, age, mobile) values ('John', 20, '010-7788-1234');" );  database.execSQL( "insert into CUSTOMER\_INFO(name, age, mobile) values ('Mike', 35, '010-8888-1111');" );  database.execSQL( "insert into CUSTOMER\_INFO(name, age, mobile) values ('Sean', 26, '010-6677-4321');" );  **return** count;  }  **private** **void** deleteRecord() {  println("deleting records.");  **int** count = 3;  database.execSQL( "delete from CUSTOMER\_INFO where age > 10" );  }  **private** **void** println(String msg) {  Log.*d*("SampleDatabaseHelper", msg);  status.append("\n" + msg);  }  …  } |
| CustomerDatabase  /\*\*  \* 고객 정보 데이터를 관리하기 위한 데이터베이스 클래스 정의  \*  \* **@author** Mike  \*  \*/  **public** **class** CustomerDatabase {  /\*\*  \* TAG for debugging  \*/  **public** **static** **final** String ***TAG*** = "CustomerDatabase";  /\*\*  \* Singleton instance  \*/  **private** **static** CustomerDatabase *database*;  /\*\*  \* database name  \*/  **public** **static** String *DATABASE\_NAME* = "customer.db";  /\*\*  \* table name for CUSTOMER\_INFO  \*/  **public** **static** String *TABLE\_CUSTOMER\_INFO* = "CUSTOMER\_INFO";  /\*\*  \* table name for MEMO  \*/  **public** **static** String *TABLE\_MEMO* = "MEMO";  /\*\*  \* version  \*/  **public** **static** **int** *DATABASE\_VERSION* = 1;  /\*\*  \* Helper class defined  \*/  **private** DatabaseHelper dbHelper;  /\*\*  \* Database object  \*/  **private** SQLiteDatabase db;  **private** Context context;  /\*\*  \* Constructor  \*/  **private** CustomerDatabase(Context context) {  **this**.context = context;  }  **public** **static** CustomerDatabase getInstance(Context context) {  **if** (*database* == **null**) {  *database* = **new** CustomerDatabase(context);  }  **return** *database*;  }  /\*\*  \* open database  \*  \* **@return**  \*/  **public** **boolean** open() {  println("opening database [" + *DATABASE\_NAME* + "].");  dbHelper = **new** DatabaseHelper(context);  db = dbHelper.getWritableDatabase();  **return** **true**;  }  /\*\*  \* close database  \*/  **public** **void** close() {  println("closing database [" + *DATABASE\_NAME* + "].");  db.close();  *database* = **null**;  }  /\*\*  \* execute raw query using the input SQL  \* close the cursor after fetching any result  \*  \* **@param** SQL  \* **@return**  \*/  **public** Cursor rawQuery(String SQL) {  println("\nexecuteQuery called.\n");  Cursor c1 = **null**;  **try** {  c1 = db.rawQuery(SQL, **null**);  println("cursor count : " + c1.getCount());  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in executeQuery", ex);  }  **return** c1;  }  **public** **boolean** execSQL(String SQL) {  println("\nexecute called.\n");  **try** {  Log.*d*(***TAG***, "SQL : " + SQL);  db.execSQL(SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in executeQuery", ex);  **return** **false**;  }  **return** **true**;  }  **private** **class** DatabaseHelper **extends** SQLiteOpenHelper  {  **public** DatabaseHelper(Context context)  {  **super**(context, *DATABASE\_NAME*, **null**, *DATABASE\_VERSION*);  }  **public** **void** onCreate(SQLiteDatabase db)  {  // TABLE\_CUSTOMER\_INFO  println("creating table [" + *TABLE\_CUSTOMER\_INFO* + "].");  // drop existing table  String DROP\_SQL = "drop table if exists " + *TABLE\_CUSTOMER\_INFO*;  **try** {  db.execSQL(DROP\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in DROP\_SQL", ex);  }  // create table  String CREATE\_SQL = "create table " + *TABLE\_CUSTOMER\_INFO* + "("  + " \_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, "  + " NAME TEXT, "  + " AGE INTEGER, "  + " MOBILE TEXT, "  + " CREATE\_DATE TIMESTAMP DEFAULT CURRENT\_TIMESTAMP "  + ")";  **try** {  db.execSQL(CREATE\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in CREATE\_SQL", ex);  }  // TABLE\_MEMO  println("creating table [" + *TABLE\_MEMO* + "].");  // drop existing table  DROP\_SQL = "drop table if exists " + *TABLE\_MEMO*;  **try** {  db.execSQL(DROP\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in DROP\_SQL", ex);  }  // create table  CREATE\_SQL = "create table " + *TABLE\_MEMO* + "("  + " \_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, "  + " CUSTOMER\_ID INTEGER, "  + " CONTENTS TEXT, "  + " TITLE TEXT, "  + " CREATE\_DATE TIMESTAMP DEFAULT CURRENT\_TIMESTAMP "  + ")";  **try** {  db.execSQL(CREATE\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in CREATE\_SQL", ex);  }  }  **public** **void** onOpen(SQLiteDatabase db)  {  println("opened database [" + *DATABASE\_NAME* + "].");  }  **public** **void** onUpgrade(SQLiteDatabase db, **int** oldVersion,  **int** newVersion)  {  println("Upgrading database from version " + oldVersion + " to " + newVersion + ".");  **if** (oldVersion < 2) { // version 1  }  }  }  **private** **void** println(String msg) {  Log.*d*(***TAG***, msg);  }  } |

CursorAdapter

* 검색해서 가져온거 메모리에 다올려놓지말고 Adapter를통해 그때그때 쓰자
* 커서로더로 바뀌긴했다.

|  |
| --- |
| /\*\*  \* 데이터베이스 조회 결과를 커서어댑터를 이용해 선택위젯에 보여주는 방법을 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **public** **static** **final** String ***TAG*** = "SampleCursorAdapter";  **private** **static** String *DATABASE\_NAME* = "employeeDB";  **private** **static** String *TABLE\_NAME* = "employee";  **private** **static** **int** *DATABASE\_VERSION* = 1;  **private** DatabaseHelper dbHelper;  **private** SQLiteDatabase db;  **private** TextView status;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  status = (TextView) findViewById(R.id.status);  ListView list = (ListView) findViewById(R.id.list);  **boolean** isOpen = openDatabase();  **if** (isOpen) {  **Cursor cursor = executeRawQueryParam();**  **startManagingCursor(cursor);**  **String[] columns = new String[] {"name", "age", "phone"};**  **int[] to = new int[] { R.id.name\_entry, R.id.age\_entry, R.id.phone\_entry };**  **SimpleCursorAdapter mAdapter = new ~~SimpleCursorAdapter~~(this, R.layout.listitem, cursor, columns, to);**  **list.setAdapter(mAdapter);**  }  }  **private** **boolean** openDatabase() {  println("opening database [" + *DATABASE\_NAME* + "].");  dbHelper = **new** DatabaseHelper(**this**);  db = dbHelper.getWritableDatabase();  **return** **true**;  }  **private** Cursor executeRawQueryParam() {  println("\nexecuteRawQueryParam called.\n");  String SQL = "select \_id, name, age, phone "  + " from " + *TABLE\_NAME*  + " where age > ?";  String[] args= {"10"};  Cursor c1 = db.rawQuery(SQL, args);  **return** c1;  }  **private** **class** DatabaseHelper **extends** SQLiteOpenHelper {  **public** DatabaseHelper(Context context) {  **super**(context, *DATABASE\_NAME*, **null**, *DATABASE\_VERSION*);  }  **public** **void** onCreate(SQLiteDatabase db) {  println("creating table [" + *TABLE\_NAME* + "].");  **try** {  String DROP\_SQL = "drop table if exists " + *TABLE\_NAME*;  db.execSQL(DROP\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in DROP\_SQL", ex);  }  String CREATE\_SQL = "create table " + *TABLE\_NAME* + "("  + " \_id integer PRIMARY KEY autoincrement, "  + " name text, "  + " age integer, "  + " phone text)";  **try** {  db.execSQL(CREATE\_SQL);  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in CREATE\_SQL", ex);  }  println("inserting records.");  **try** {  db.execSQL( "insert into " + *TABLE\_NAME* + "(name, age, phone) values ('John', 20, '010-7788-1234');" );  db.execSQL( "insert into " + *TABLE\_NAME* + "(name, age, phone) values ('Mike', 35, '010-8888-1111');" );  db.execSQL( "insert into " + *TABLE\_NAME* + "(name, age, phone) values ('Sean', 26, '010-6677-4321');" );  } **catch**(Exception ex) {  Log.*e*(***TAG***, "Exception in insert SQL", ex);  }  }  **public** **void** onOpen(SQLiteDatabase db) {  println("opened database [" + *DATABASE\_NAME* + "].");  }  **public** **void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  Log.*w*(***TAG***, "Upgrading database from version " + oldVersion +  " to " + newVersion + ".");  db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  onCreate(db);  }  }  **private** **void** println(String msg) {  Log.*d*(***TAG***, msg);  status.append("\n" + msg);  }  …  } |

데이터를 DB정보로 바꾸는방법

* Drug
* 33일

멀티미디어

오디오 재생 MediaPlayer

* MediaPlayer 클래스사용
* .setDataSource(url)
* 준비과정 선실행되야함 .prepare()
* 시작 .start()
* 중지 .stop()
* 현재 재생위치 .getCurrentPosition()
* 중지 .pause()
* 특정위치 이동 .seekTo(position)
* 끝난후 .release()
* 시작중인지 확인 .isPlaying()

|  |
| --- |
| /\*\*  \* 음악파일을 재생하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **static** **final** String ***AUDIO\_URL*** = "http://sites.google.com/site/ubiaccessmobile/sample\_audio.amr";  **private** MediaPlayer mediaPlayer;  **private** **int** playbackPosition = 0;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  Button startBtn = (Button) findViewById(R.id.playBtn);  Button pauseBtn = (Button) findViewById(R.id.pauseBtn);  Button restartBtn = (Button) findViewById(R.id.restartBtn);  startBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View view) {  **try** {  playAudio(***AUDIO\_URL***);  Toast.*makeText*(getApplicationContext(), "음악 파일 재생 시작됨.", 2000).show();  } **catch** (Exception e) {  e.printStackTrace();  }  }  });  pauseBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View view) {  **if** (mediaPlayer != **null**) {  playbackPosition = mediaPlayer.getCurrentPosition();  mediaPlayer.pause();  Toast.*makeText*(getApplicationContext(), "음악 파일 재생 중지됨.", 2000).show();  }  }  });  restartBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View view) {  **if** (mediaPlayer != **null** && !mediaPlayer.isPlaying()) {  mediaPlayer.start();  mediaPlayer.seekTo(playbackPosition);  Toast.*makeText*(getApplicationContext(), "음악 파일 재생 재시작됨.", 2000).show();  }  }  });  }  **private** **void** playAudio(String url) **throws** Exception {  killMediaPlayer();  mediaPlayer = **new** MediaPlayer();  mediaPlayer.setDataSource(url);  mediaPlayer.prepare();  mediaPlayer.start();  }  **protected** **void** onDestroy() {  **super**.onDestroy();  killMediaPlayer();  }  **private** **void** killMediaPlayer() {  **if** (mediaPlayer != **null**) {  **try** {  mediaPlayer.release();  } **catch** (Exception e) {  e.printStackTrace();  }  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

영상 재생하기 VideoView

* VideoView 사용
* 동영상 지정 .setVideoURI(Uri.parse())
* 우선적으로 동작하도록 지정 .requestFocus();
* 미디어 컨트롤러 사용 MediaCOntroller 클래스 사용
* 미디어 컨트롤적용하기 .setMediaCOntroller(controller)
* 리스너 : setOnPreParedListener() 🡨 동영상 재생이 준비 됐을 때
* 리스너 : setOnCompleteListener() 🡨 동영상 재생이 완료됐을 때

|  |
| --- |
| /\*\*  \* 동영상 재생 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **static** **final** String ***VIDEO\_URL*** = "http://sites.google.com/site/ubiaccessmobile/sample\_video.mp4";  **private** VideoView videoView;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  Button startBtn = (Button) findViewById(R.id.startBtn);  Button volumeBtn = (Button) findViewById(R.id.volumeBtn);  startBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View view) {  videoView.seekTo(0);  videoView.start();  }  });  volumeBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View view) {  AudioManager mAudioManager = (AudioManager) getSystemService(AUDIO\_SERVICE);  **int** maxVolume = mAudioManager.getStreamMaxVolume(AudioManager.***STREAM\_MUSIC***);  mAudioManager.setStreamVolume(AudioManager.***STREAM\_MUSIC***, maxVolume, AudioManager.***FLAG\_SHOW\_UI***);  }  });  videoView = (VideoView) findViewById(R.id.videoView);  MediaController mc = **new** MediaController(**this**);  videoView.setMediaController(mc);  videoView.setVideoURI(Uri.*parse*(***VIDEO\_URL***));  videoView.requestFocus();  videoView.setOnPreparedListener(**new** MediaPlayer.OnPreparedListener() {  **public** **void** onPrepared(MediaPlayer player) {  Toast.*makeText*(getApplicationContext(), "동영상이 준비되었습니다.\n'재생' 버튼을 누르세요.", Toast.***LENGTH\_LONG***).show();  }  });  videoView.setOnCompletionListener(**new** MediaPlayer.OnCompletionListener() {  **public** **void** onCompletion(MediaPlayer player) {  Toast.*makeText*(getApplicationContext(), "동영상 재생이 완료되었습니다.", Toast.***LENGTH\_LONG***).show();  }  });  }  **protected** **void** onResume() {  Toast.*makeText*(getApplicationContext(), "동영상 준비중입니다.\n잠시 기다려주세요.", Toast.***LENGTH\_LONG***).show();  **super**.onResume();  }  …  } |

녹음하기

* 미디어 리코더 녹음을 위한 미디어 코더 MediaRecorder
* MediaRecorder
  + 🡪 setAudioSource
  + setAudioEncoder
  + setOutputFormat
  + setOutputFile
* 중지 .stop()
* 반환 release()
* 권한 RECORD\_AUDIO, READ\_EXTERNAL\_STORAGE, WRITE\_EXTERNAL\_STORAGE

|  |
| --- |
| /\*\*  \* 음성 녹음을 하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **final** **private** **static** String ***RECORDED\_FILE*** = "/sdcard/recorded.mp4";  MediaPlayer player;  MediaRecorder recorder;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  Button recordBtn = (Button) findViewById(R.id.recordBtn);  Button recordStopBtn = (Button) findViewById(R.id.recordStopBtn);  Button playBtn = (Button) findViewById(R.id.playBtn);  Button playStopBtn = (Button) findViewById(R.id.playStopBtn);  recordBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (recorder != **null**) {  recorder.stop();  recorder.release();  recorder = **null**;  }  recorder = **new** MediaRecorder();  recorder.setAudioSource(MediaRecorder.AudioSource.***MIC***);  recorder.setOutputFormat(MediaRecorder.OutputFormat.***MPEG\_4***);  recorder.setAudioEncoder(MediaRecorder.AudioEncoder.***DEFAULT***);  recorder.setOutputFile(***RECORDED\_FILE***);  **try** {  Toast.*makeText*(getApplicationContext(), "녹음을 시작합니다.", Toast.***LENGTH\_LONG***).show();  recorder.prepare();  recorder.start();  } **catch** (Exception ex) {  Log.*e*("SampleAudioRecorder", "Exception : ", ex);  }  }  });  recordStopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (recorder == **null**)  **return**;  recorder.stop();  recorder.release();  recorder = **null**;  Toast.*makeText*(getApplicationContext(), "녹음이 중지되었습니다.", Toast.***LENGTH\_LONG***).show();  }  });  playBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (player != **null**) {  player.stop();  player.release();  player = **null**;  }  Toast.*makeText*(getApplicationContext(), "녹음된 파일을 재생합니다.", Toast.***LENGTH\_LONG***).show();  **try** {  player = **new** MediaPlayer ();  player.setDataSource(***RECORDED\_FILE***);  player.prepare();  player.start();  } **catch** (Exception e) {  Log.*e*("SampleAudioRecorder", "Audio play failed.", e);  }  }  });  playStopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (player == **null**)  **return**;  Toast.*makeText*(getApplicationContext(), "재생이 중지되었습니다.", Toast.***LENGTH\_LONG***).show();  player.stop();  player.release();  player = **null**;  }  });  }  **protected** **void** onPause() {  **if** (recorder != **null**) {  recorder.release();  recorder = **null**;  }  **if** (player != **null**) {  player.release();  player = **null**;  }  **super**.onPause();  }  …  }  <uses-permission android:name="android.permission.RECORD\_AUDIO" />  <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" /> |

동영상 녹화 MediaRecorder (surfaceVIew 사용)

* setAudioSource
* setMediaSource
* setOutputFormat
* setAudioEncoder
* setVideoEncoder
* setOutputFile

|  |
| --- |
| /\*\*  \* 동영상을 녹화하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **public** **static** **final** String ***TAG*** = "SampleVideoRecorderActivity";  **private** **static** String *EXTERNAL\_STORAGE\_PATH* = "";  **private** **static** String *RECORDED\_FILE* = "video\_recorded";  **private** **static** **int** *fileIndex* = 0;  **private** **static** String *filename* = "";  MediaPlayer player;  MediaRecorder recorder;  **private** ~~Camera~~ camera = **null**;  SurfaceHolder holder;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // check external storage  String state = Environment.*getExternalStorageState*();  **if** (!state.equals(Environment.***MEDIA\_MOUNTED***)) {  Log.*d*(***TAG***, "External Storage Media is not mounted.");  } **else** {  *EXTERNAL\_STORAGE\_PATH* = Environment.*getExternalStorageDirectory*().getAbsolutePath();  }  // create a SurfaceView instance and add it to the layout  SurfaceView surface = **new** SurfaceView(**this**);  holder = surface.getHolder();  holder.~~setType~~(SurfaceHolder.~~SURFACE\_TYPE\_PUSH\_BUFFERS~~);  FrameLayout frame = (FrameLayout) findViewById(R.id.videoLayout);  frame.addView(surface);  Button recordBtn = (Button) findViewById(R.id.recordBtn);  Button recordStopBtn = (Button) findViewById(R.id.recordStopBtn);  Button playBtn = (Button) findViewById(R.id.playBtn);  Button playStopBtn = (Button) findViewById(R.id.playStopBtn);  recordBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **try** {  **if** (recorder == **null**) {  recorder = **new** MediaRecorder();  }  recorder.setAudioSource(MediaRecorder.AudioSource.***MIC***);  recorder.setVideoSource(MediaRecorder.VideoSource.***CAMERA***);  recorder.setOutputFormat(MediaRecorder.OutputFormat.***MPEG\_4***);  recorder.setAudioEncoder(MediaRecorder.AudioEncoder.***DEFAULT***);  recorder.setVideoEncoder(MediaRecorder.VideoEncoder.***DEFAULT***);  *filename* = createFilename();  Log.*d*(***TAG***, "current filename : " + *filename*);  recorder.setOutputFile(*filename*);  recorder.setPreviewDisplay(holder.getSurface());  recorder.prepare();  recorder.start();  } **catch** (Exception ex) {  Log.*e*(***TAG***, "Exception : ", ex);  recorder.release();  recorder = **null**;  }  }  });  recordStopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (recorder == **null**)  **return**;  recorder.stop();  recorder.reset();  recorder.release();  recorder = **null**;  ContentValues values = **new** ContentValues(10);  values.put(MediaStore.MediaColumns.***TITLE***, "RecordedVideo");  values.put(MediaStore.Audio.Media.***ALBUM***, "Video Album");  values.put(MediaStore.Audio.Media.***ARTIST***, "Mike");  values.put(MediaStore.Audio.Media.***DISPLAY\_NAME***, "Recorded Video");  values.put(MediaStore.MediaColumns.***DATE\_ADDED***, System.*currentTimeMillis*() / 1000);  values.put(MediaStore.MediaColumns.***MIME\_TYPE***, "video/mp4");  values.put(MediaStore.Audio.Media.***DATA***, *filename*);  Uri videoUri = getContentResolver().insert(MediaStore.Video.Media.***EXTERNAL\_CONTENT\_URI***, values);  **if** (videoUri == **null**) {  Log.*d*("SampleVideoRecorder", "Video insert failed.");  **return**;  }  sendBroadcast(**new** Intent(Intent.***ACTION\_MEDIA\_SCANNER\_SCAN\_FILE***, videoUri));  }  });  playBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (player == **null**) {  player = **new** MediaPlayer();  }  **try** {  player.setDataSource(*filename*);  player.setDisplay(holder);  player.prepare();  player.start();  } **catch** (Exception e) {  Log.*e*(***TAG***, "Video play failed.", e);  }  }  });  playStopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  **if** (player == **null**)  **return**;  player.stop();  player.release();  player = **null**;  }  });  }  **private** String createFilename() {  *fileIndex*++;  String newFilename = "";  **if** (*EXTERNAL\_STORAGE\_PATH* == **null** || *EXTERNAL\_STORAGE\_PATH*.equals("")) {  // use internal memory  newFilename = *RECORDED\_FILE* + *fileIndex* + ".mp4";  } **else** {  newFilename = *EXTERNAL\_STORAGE\_PATH* + "/" + *RECORDED\_FILE* + *fileIndex* + ".mp4";  }  **return** newFilename;  }  **protected** **void** onPause() {  **if** (camera != **null**) {  camera.~~release~~();  camera = **null**;  }  **if** (recorder != **null**) {  recorder.release();  recorder = **null**;  }  **if** (player != **null**) {  player.release();  player = **null**;  }  **super**.onPause();  }  …  }  <uses-permission android:name="android.permission.RECORD\_AUDIO" />  <uses-permission android:name="android.permission.CAMERA" />  <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" /> |

사진찍어 저장 (단말 카메라 어플사용)

* Intent사용하여 콜 MediaStore.ACTION\_IMAGE\_CAPTURE
* EXTRA 🡪 putExtra(MediaStore.EXTRA\_OUTPUT,URI.parse()); Uri.fromFile(file)

|  |
| --- |
| **public** **class** MainActivity **extends** ActionBarActivity {  **public** **static** **final** **int** ***REQUEST\_IMAGE\_CAPTURE*** = 1001;  File file = **null**;  ImageView imageView1;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  imageView1 = (ImageView) findViewById(R.id.imageView1);  **try** {  file = createFile();  } **catch** (IOException ex) {  ex.printStackTrace();  }  }  **public** **void** onButton1Clicked(View v) {  Intent intent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  intent.putExtra(MediaStore.***EXTRA\_OUTPUT***, Uri.*fromFile*(file));  **if** (intent.resolveActivity(getPackageManager()) != **null**) {  startActivityForResult(intent, ***REQUEST\_IMAGE\_CAPTURE***);  }  }  **private** File createFile() **throws** IOException {  String imageFileName = "test.jpg";  File storageDir = Environment.*getExternalStorageDirectory*();  File curFile = **new** File(storageDir, imageFileName);  **return** curFile;  }  @Override  **protected** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **if** (requestCode == ***REQUEST\_IMAGE\_CAPTURE*** && resultCode == RESULT\_OK) {  BitmapFactory.Options options = **new** BitmapFactory.Options();  options.inSampleSize = 8;  **if** (file != **null**) {  Bitmap bitmap = BitmapFactory.*decodeFile*(file.getAbsolutePath(), options);  imageView1.setImageBitmap(bitmap);  } **else** {  Toast.*makeText*(getApplicationContext(), "File is null.", Toast.***LENGTH\_LONG***).show();  }  }  }  ...  }  <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  <uses-feature android:name="android.hardware.camera"  android:required="true" /> |

직접 카메라를 억세스해서 처리하기 sufaceView 사용

* camera = Camera.open();
* camera.setPreviewDisplay(mHolder);
* camera.startPreview();
* camera.stopPreview();
* camera.release();
* camera.takePicture(null, null, handler);

|  |
| --- |
| /\*\*  \* 서피스뷰를 이용해 미리보기 화면을 만든 후 사진찍기를 하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **public** **static** String IMAGE\_FILE = "capture.jpg";  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  **final** CameraSurfaceView cameraView = **new** CameraSurfaceView(getApplicationContext());  FrameLayout previewFrame = (FrameLayout) findViewById(R.id.previewFrame);  previewFrame.addView(cameraView);  // 버튼 이벤트 처리  Button saveBtn = (Button) findViewById(R.id.saveBtn);  saveBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  cameraView.capture(**new** Camera.PictureCallback() {  **public** **void** onPictureTaken(**byte**[] data, Camera camera) {  **try** {  Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);  String outUriStr = MediaStore.Images.Media.insertImage(getContentResolver(),  bitmap, "Captured Image", "Captured Image using Camera.");  **if** (outUriStr == **null**) {  Log.d("SampleCapture", "Image insert failed.");  **return**;  } **else** {  Uri outUri = Uri.parse(outUriStr);  sendBroadcast(**new** Intent(Intent.ACTION\_MEDIA\_SCANNER\_SCAN\_FILE, outUri));  }  Toast.makeText(getApplicationContext(), "카메라로 찍은 사진을 앨범에 저장했습니다.", Toast.LENGTH\_LONG).show();  // restart  camera.startPreview();  } **catch** (Exception e) {  Log.e("SampleCapture", "Failed to insert image.", e);  }  }  });  }  });  }  /\*\*  \* 카메라 미리보기를 위한 서피스뷰 정의  \*/  **private** **class** CameraSurfaceView **extends** SurfaceView **implements** SurfaceHolder.Callback {  **private** SurfaceHolder mHolder;  **private** Camera camera = **null**;  **public** CameraSurfaceView(Context context) {  **super**(context);  mHolder = getHolder();  mHolder.addCallback(**this**);  }  **public** **void** surfaceCreated(SurfaceHolder holder) {  camera = Camera.open();  **try** {  camera.setPreviewDisplay(mHolder);  } **catch** (Exception e) {  Log.e("CameraSurfaceView", "Failed to set camera preview.", e);  }  }  **public** **void** surfaceChanged(SurfaceHolder holder, **int** format, **int** width, **int** height) {  camera.startPreview();  }  **public** **void** surfaceDestroyed(SurfaceHolder holder) {  camera.stopPreview();  camera.release();  camera = **null**;  }  **public** **boolean** capture(Camera.PictureCallback handler) {  **if** (camera != **null**) {  camera.takePicture(**null**, **null**, handler);  **return** **true**;  } **else** {  **return** **false**;  }  }  }  ...  }  <uses-permission android:name="android.permission.CAMERA"/>  <uses-permission android:name="android.permission.FLASHLIGHT"/>  <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> |
| <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical" >  <Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:layout\_gravity="center"  android:text="사진찍어 저장하기"  android:textSize="20dp"  android:id="@+id/saveBtn"  />    <FrameLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/previewFrame"  />  </LinearLayout> |

QR코드 읽기 타 어플 실행

|  |
| --- |
| /\*\*  \* 바코드 스캔 화면을 띄우는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **public** **static** **final** **int** REQUEST\_CODE\_SCAN = 1001;  **public** **static** **final** **int** DIALOG\_SCANNER\_NEEDED = 1002;  **public** **static** **final** **int** DIALOG\_SHOW\_URL = 1003;  **public** **static** **final** String PRODUCT\_CODE\_TYPES = "UPC\_A,UPC\_E,EAN\_8,EAN\_13";  **public** **static** **final** String ONE\_D\_CODE\_TYPES = PRODUCT\_CODE\_TYPES + ",CODE\_39,CODE\_93,CODE\_128";  **public** **static** **final** String QR\_CODE\_TYPES = "QR\_CODE";  **public** **static** **final** String ALL\_CODE\_TYPES = **null**;  **private** TextView contentsText;  **private** String scannedUrl;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  contentsText = (TextView) findViewById(R.id.contentsText);  Button scanBtn = (Button) findViewById(R.id.scanBtn);  scanBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  scan();  }  });  }  **private** **void** scan() {  scan(ALL\_CODE\_TYPES);  }  **private** **void** scan(String formats) {  Intent intentScan = **new** Intent("com.google.zxing.client.android.SCAN");  intentScan.addCategory(Intent.CATEGORY\_DEFAULT);  **if** (formats != **null**) {  intentScan.putExtra("SCAN\_FORMATS", formats);  }  **try** {  startActivityForResult(intentScan, REQUEST\_CODE\_SCAN);  } **catch** (ActivityNotFoundException e) {  showDialog(DIALOG\_SCANNER\_NEEDED);  }  }  **protected** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent intent) {  **super**.onActivityResult(requestCode, resultCode, intent);  **if** (requestCode == REQUEST\_CODE\_SCAN) {  Toast toast = Toast.makeText(getBaseContext(), "onActivityResult called with code : " + resultCode, Toast.LENGTH\_LONG);  toast.show();  **if** (resultCode == Activity.RESULT\_OK) {  String contents = intent.getStringExtra("SCAN\_RESULT");  String formatName = intent.getStringExtra("SCAN\_RESULT\_FORMAT");  contentsText.append("\nSCAN RESULT FORMAT : " + formatName);  contentsText.append("\nSCAN RESULT : " + contents);  **if** (contents != **null** && contents.indexOf("http://") >= 0) {  **int** startIndex = contents.indexOf("http://");  scannedUrl = contents.substring(startIndex);  showDialog(DIALOG\_SHOW\_URL);  }  } **else** {  contentsText.append("\nSCAN FAILED.");  }  }  }  **protected** Dialog onCreateDialog(**int** id) {  AlertDialog.Builder builder = **null**;  **switch**(id) {  **case** DIALOG\_SCANNER\_NEEDED:  builder = **new** AlertDialog.Builder(**this**);  builder.setTitle("바코드 스캐너 앱 설치");  builder.setMessage("바코드 스캐너 앱이 필요합니다. 자동 설치할까요?");  builder.setPositiveButton("예", **new** DialogInterface.OnClickListener() {  **public** **void** onClick(DialogInterface dialog, **int** whichButton) {  Uri uri = Uri.parse("market://details?id=com.google.zxing.client.android");  Intent intent = **new** Intent(Intent.ACTION\_VIEW, uri);  startActivity(intent);  }  });  builder.setNegativeButton("아니오", **new** DialogInterface.OnClickListener() {  **public** **void** onClick(DialogInterface dialog, **int** whichButton) {  }  });  **break**;  **case** DIALOG\_SHOW\_URL:  builder = **new** AlertDialog.Builder(**this**);  builder.setTitle("웹으로 보기");  builder.setMessage("스캔한 결과를 웹으로 보시겠습니까?");  builder.setPositiveButton("예", **new** DialogInterface.OnClickListener() {  **public** **void** onClick(DialogInterface dialog, **int** whichButton) {  Intent intent = **new** Intent(Intent.ACTION\_VIEW, Uri.parse(scannedUrl));  startActivity(intent);  }  });  builder.setNegativeButton("아니오", **new** DialogInterface.OnClickListener() {  **public** **void** onClick(DialogInterface dialog, **int** whichButton) {  }  });  **break**;  **default**:  **break**;  }  **return** builder.create();  }  ...  } |

위치기반 GPS

* Manager를통해 사용

|  |
| --- |
| /\*\*  \* 위치 관리자를 이용해 내 위치를 확인하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // 버튼 이벤트 처리  Button button01 = (Button) findViewById(R.id.button01);  button01.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  // 위치 정보 확인을 위해 정의한 메소드 호출  startLocationService();  }  });  }  /\*\*  \* 위치 정보 확인을 위해 정의한 메소드  \*/  **private** **void** startLocationService() {  // 위치 관리자 객체 참조  LocationManager manager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  // 위치 정보를 받을 리스너 생성  GPSListener gpsListener = **new** GPSListener();  **long** minTime = 10000;  **float** minDistance = 0;  // GPS를 이용한 위치 요청  manager.requestLocationUpdates(  LocationManager.***GPS\_PROVIDER***,  minTime,  minDistance,  gpsListener);  // 네트워크를 이용한 위치 요청  manager.requestLocationUpdates(  LocationManager.***NETWORK\_PROVIDER***,  minTime,  minDistance,  gpsListener);  // 위치 확인이 안되는 경우에도 최근에 확인된 위치 정보 먼저 확인  **try** {  Location lastLocation = manager.getLastKnownLocation(LocationManager.***GPS\_PROVIDER***);  **if** (lastLocation != **null**) {  Double latitude = lastLocation.getLatitude();  Double longitude = lastLocation.getLongitude();  Toast.*makeText*(getApplicationContext(), "Last Known Location : " + "Latitude : " + latitude + "\nLongitude:" + longitude, Toast.***LENGTH\_LONG***).show();  }  } **catch**(Exception ex) {  ex.printStackTrace();  }  Toast.*makeText*(getApplicationContext(), "위치 확인이 시작되었습니다. 로그를 확인하세요.", Toast.***LENGTH\_SHORT***).show();  }  /\*\*  \* 리스너 클래스 정의  \*/  **private** **class** GPSListener **implements** LocationListener {  /\*\*  \* 위치 정보가 확인될 때 자동 호출되는 메소드  \*/  **public** **void** onLocationChanged(Location location) {  Double latitude = location.getLatitude();  Double longitude = location.getLongitude();  String msg = "Latitude : "+ latitude + "\nLongitude:"+ longitude;  Log.*i*("GPSListener", msg);  Toast.*makeText*(getApplicationContext(), msg, Toast.***LENGTH\_SHORT***).show();  }  **public** **void** onProviderDisabled(String provider) {  }  **public** **void** onProviderEnabled(String provider) {  }  **public** **void** onStatusChanged(String provider, **int** status, Bundle extras) {  }  }  ...  }  <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" /> |

지도사용

* Fragment를 사용함
* Play-services 사용하게함(SDK에서 받음)
* Play-services 참조시키고
* 안드로이드 맵키 만들기 37일 4장

|  |
| --- |
| /\*\*  \* 현재 위치의 지도를 보여주는 방법에 대해 알 수 있습니다.  \*  \* Google Play Services 라이브러리를 링크하여 사용합니다.  \* 구글맵 v2를 사용하기 위한 여러 가지 권한이 있어야 합니다.  \* 매니페스트 파일 안에 있는 키 값을 PC에 맞는 것으로 새로 발급받아서 넣어야 합니다.  \*  \* **@author** Mike  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **private** GoogleMap map;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // 지도 객체 참조  **map = ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map)).getMap();**  // 위치 확인하여 위치 표시 시작  startLocationService();  }  /\*\*  \* 현재 위치 확인을 위해 정의한 메소드  \*/  **private** **void** startLocationService() {  // 위치 관리자 객체 참조  LocationManager manager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  // 리스너 객체 생성  GPSListener gpsListener = **new** GPSListener();  **long** minTime = 10000;  **float** minDistance = 0;  // GPS 기반 위치 요청  manager.requestLocationUpdates(  LocationManager.***GPS\_PROVIDER***,  minTime,  minDistance,  gpsListener);  // 네트워크 기반 위치 요청  manager.requestLocationUpdates(  LocationManager.***NETWORK\_PROVIDER***,  minTime,  minDistance,  gpsListener);  Toast.*makeText*(getApplicationContext(), "위치 확인 시작함. 로그를 확인하세요.", Toast.***LENGTH\_SHORT***).show();  }  /\*\*  \* 리스너 정의  \*/  **private** **class** GPSListener **implements** LocationListener {  /\*\*  \* 위치 정보가 확인되었을 때 호출되는 메소드  \*/  **public** **void** onLocationChanged(Location location) {  Double latitude = location.getLatitude();  Double longitude = location.getLongitude();  String msg = "Latitude : "+ latitude + "\nLongitude:"+ longitude;  Log.*i*("GPSLocationService", msg);  // 현재 위치의 지도를 보여주기 위해 정의한 메소드 호출  showCurrentLocation(latitude, longitude);  }  **public** **void** onProviderDisabled(String provider) {  }  **public** **void** onProviderEnabled(String provider) {  }  **public** **void** onStatusChanged(String provider, **int** status, Bundle extras) {  }  }  /\*\*  \* 현재 위치의 지도를 보여주기 위해 정의한 메소드  \*  \* **@param** latitude  \* **@param** longitude  \*/  **private** **void** showCurrentLocation(Double latitude, Double longitude) {  // 현재 위치를 이용해 LatLon 객체 생성  LatLng curPoint = **new** LatLng(latitude, longitude);  map.animateCamera(CameraUpdateFactory.newLatLngZoom(curPoint, 15));  // 지도 유형 설정. 지형도인 경우에는 GoogleMap.MAP\_TYPE\_TERRAIN, 위성 지도인 경우에는 GoogleMap.MAP\_TYPE\_SATELLITE  map.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);  }  ...  } |

|  |
| --- |
| <RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  xmlns:tools=*"http://schemas.android.com/tools"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  >  <fragment  android:id=*"@+id/map"*  android:layout\_width=*"match\_parent"*  android:layout\_height=*"match\_parent"*  class=*"****com.google.android.gms.maps.SupportMapFragment****"* />    </RelativeLayout> |

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"org.androidtown.lbs.map"* >  <!-- 내 프로젝트에 대한 보안 수준 설정. 내 프로젝트의 패키지가 org.androidtown.lbs.map 인 경우 -->  <permission  android:name=*"org.androidtown.lbs.map.permission.MAPS\_RECEIVE"*  android:protectionLevel=*"signature"* />  <!-- 구글맵 API v2의 권한 설정 -->  <uses-permission android:name=*"org.androidtown.lbs.map.permission.MAPS\_RECEIVE"* />  <!-- 구글맵 API v2의 일반 권한 설정. -->  <uses-permission android:name=*"android.permission.INTERNET"* />  <uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />  <uses-permission android:name=*"com.google.android.providers.gsf.permission.READ\_GSERVICES"* />  <!-- 일반 권한 설정 -->  <uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"* />  <uses-permission android:name=*"android.permission.READ\_EXTERNAL\_STORAGE"* />  <uses-permission android:name=*"android.permission.ACCESS\_COARSE\_LOCATION"* />  <uses-permission android:name=*"android.permission.ACCESS\_FINE\_LOCATION"* />  <!-- OpenGL 사용 설정 -->  <uses-feature  android:glEsVersion=*"0x00020000"*  android:required=*"true"* />  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  <!-- 라이브러리 사용 설정 -->  <uses-library android:name=*"com.google.android.maps"* />  <!-- 구글맵 API v2 키 설정 -->  <meta-data  android:name=*"com.google.android.maps.v2.API\_KEY"*  android:value=*"AIzaSyBsEwz6GlQi9oURfabazpTCIUfAbYWXxDI"* />  <!-- GMS 라이브러리 버전 설정 -->  <meta-data  android:name=*"com.google.android.gms.version"*  android:value=*"@integer/google\_play\_services\_version"* />  <activity  android:name=*".MainActivity"*  android:label=*"@string/app\_name"* >  <intent-filter>  <action android:name=*"android.intent.action.MAIN"* />  <category android:name=*"android.intent.category.LAUNCHER"* />  </intent-filter>  </activity>  </application>  </manifest> |

지도위에 마크 처리하기

|  |
| --- |
| /\*\*  \* 현재 위치의 지도를 보여주고 그 위에 오버레이를 추가하는 방법에 대해 알 수 있습니다.  \* 내 위치 표시를 해 줍니다.  \* 방향 센서를 이용해 나침반을 화면에 표시합니다.  \*  \* 구글맵 v2를 사용하기 위한 여러 가지 권한이 있어야 합니다.  \* 매니페스트 파일 안에 있는 키 값을 PC에 맞는 것으로 새로 발급받아서 넣어야 합니다.  \*  \* **@author** Mike  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **private** RelativeLayout mainLayout;  **private** GoogleMap map;  **private** CompassView mCompassView;  **private** SensorManager mSensorManager;  **private** **boolean** mCompassEnabled;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // 메인 레이아웃 객체 참조  mainLayout = (RelativeLayout) findViewById(R.id.mainLayout);  // 지도 객체 참조  map = ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map)).getMap();  // 센서 관리자 객체 참조  mSensorManager = (SensorManager)getSystemService(Context.***SENSOR\_SERVICE***);  // 나침반을 표시할 뷰 생성  **boolean** sideBottom = **true**;  mCompassView = **new** CompassView(**this**);  mCompassView.setVisibility(View.***VISIBLE***);  RelativeLayout.LayoutParams params = **new** RelativeLayout.LayoutParams(  RelativeLayout.LayoutParams.***WRAP\_CONTENT***,  RelativeLayout.LayoutParams.***WRAP\_CONTENT***);  params.addRule(RelativeLayout.***ALIGN\_PARENT\_LEFT***);  params.addRule(sideBottom ? RelativeLayout.***ALIGN\_PARENT\_BOTTOM*** : RelativeLayout.***ALIGN\_PARENT\_TOP***);  mainLayout.addView(mCompassView, params);  mCompassEnabled = **true**;  // 위치 확인하여 위치 표시 시작  startLocationService();  }  @Override  **public** **void** onResume() {  **super**.onResume();  // 내 위치 자동 표시 enable  map.setMyLocationEnabled(**true**);  **if**(mCompassEnabled) {  mSensorManager.registerListener(mListener, mSensorManager.getDefaultSensor(Sensor.~~TYPE\_ORIENTATION~~), SensorManager.***SENSOR\_DELAY\_UI***);  }  }  @Override  **public** **void** onPause() {  **super**.onPause();  // 내 위치 자동 표시 disable  map.setMyLocationEnabled(**false**);  **if**(mCompassEnabled) {  mSensorManager.unregisterListener(mListener);  }  }  /\*\*  \* 현재 위치 확인을 위해 정의한 메소드  \*/  **private** **void** startLocationService() {  // 위치 관리자 객체 참조  LocationManager manager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  // 리스너 객체 생성  GPSListener gpsListener = **new** GPSListener();  **long** minTime = 10000;  **float** minDistance = 0;  // GPS 기반 위치 요청  manager.requestLocationUpdates(  LocationManager.***GPS\_PROVIDER***,  minTime,  minDistance,  gpsListener);  // 네트워크 기반 위치 요청  manager.requestLocationUpdates(  LocationManager.***NETWORK\_PROVIDER***,  minTime,  minDistance,  gpsListener);  Toast.*makeText*(getApplicationContext(), "위치 확인 시작함. 로그를 확인하세요.", Toast.***LENGTH\_SHORT***).show();  }  /\*\*  \* 리스너 정의  \*/  **private** **class** GPSListener **implements** LocationListener {  /\*\*  \* 위치 정보가 확인되었을 때 호출되는 메소드  \*/  **public** **void** onLocationChanged(Location location) {  Double latitude = location.getLatitude();  Double longitude = location.getLongitude();  String msg = "Latitude : "+ latitude + "\nLongitude:"+ longitude;  Log.*i*("GPSLocationService", msg);  // 현재 위치의 지도를 보여주기 위해 정의한 메소드 호출  showCurrentLocation(latitude, longitude);  }  **public** **void** onProviderDisabled(String provider) {  }  **public** **void** onProviderEnabled(String provider) {  }  **public** **void** onStatusChanged(String provider, **int** status, Bundle extras) {  }  }  /\*\*  \* 현재 위치의 지도를 보여주기 위해 정의한 메소드  \*  \* **@param** latitude  \* **@param** longitude  \*/  **private** **void** showCurrentLocation(Double latitude, Double longitude) {  // 현재 위치를 이용해 LatLon 객체 생성  LatLng curPoint = **new** LatLng(latitude, longitude);  map.animateCamera(CameraUpdateFactory.newLatLngZoom(curPoint, 15));  // 지도 유형 설정. 지형도인 경우에는 GoogleMap.MAP\_TYPE\_TERRAIN, 위성 지도인 경우에는 GoogleMap.MAP\_TYPE\_SATELLITE  map.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);  // 현재 위치 주위에 아이콘을 표시하기 위해 정의한 메소드  showAllBankItems(latitude, longitude);  }  **/\*\***  **\* 아이콘을 표시하기 위해 정의한 메소드**  **\*/**  **private void showAllBankItems(Double latitude, Double longitude) {**  **MarkerOptions marker = new MarkerOptions();**  **marker.position(new LatLng(latitude+0.001, longitude+0.001));**  **marker.title("● 지점명 : \n 국민은행(낙성대지점)\n");**  **marker.snippet("● 주소 : \n 서울시 관악구 낙성대동");**  **marker.draggable(true);**  **marker.icon(BitmapDescriptorFactory.fromResource(R.drawable.bank));**  **map.addMarker(marker);**  **}**  /\*\*  \* 센서의 정보를 받기 위한 리스너 객체 생성  \*/  **private final SensorEventListener mListener = new SensorEventListener() {**  **private int iOrientation = -1;**  **public void onAccuracyChanged(Sensor sensor, int accuracy) {**  **}**  **// 센서의 값을 받을 수 있도록 호출되는 메소드**  **public void onSensorChanged(SensorEvent event) {**  **if (iOrientation < 0) {**  **iOrientation = ((WindowManager) getSystemService(Context.*WINDOW\_SERVICE*)).getDefaultDisplay().getRotation();**  **}**  **mCompassView.setAzimuth(event.values[0] + 90 \* iOrientation);**  **mCompassView.invalidate();**  **}**  };  ..  } |
| 나침반  /\*\*  \* 나침반을 표시하기 위한 클래스를 정의합니다.  \*  \* **@author** Mike  \*  \*/  **public** **class** CompassView **extends** View {  **private** Drawable mCompass;  **private** **float** mAzimuth = 0;  **private** **int** PADDING = 2;  **public** CompassView(Context ctx) {  **super**(ctx);  **this**.mCompass = ctx.getResources().~~getDrawable~~(R.drawable.arrow\_n);  }  **protected** **void** onDraw(Canvas canvas) {  canvas.save();    canvas.rotate(360 - mAzimuth, PADDING + mCompass.getMinimumWidth()  / 2, PADDING + mCompass.getMinimumHeight() / 2);  mCompass.setBounds(PADDING, PADDING, PADDING  + mCompass.getMinimumWidth(), PADDING  + mCompass.getMinimumHeight());    mCompass.draw(canvas);  canvas.restore();  **super**.onDraw(canvas);  }  **public** **void** setAzimuth(**float** aAzimuth) {  mAzimuth = aAzimuth;  }  } |

근접경보 기능 근접했을 때 확인

|  |
| --- |
| /\*\*  \* 일정한 지점을 좌표로 지정하고 그 지점에 가까이 갔을 때 알 수 있는 근접경보 기능 사용 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **private** **static** **final** String ***TAG*** = "MainActivity";  **private** LocationManager mLocationManager;  **private** CoffeeIntentReceiver mIntentReceiver;  ArrayList mPendingIntentList;  String intentKey = "coffeeProximity";  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  // 위치 관리자 객체 참조  mLocationManager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  mPendingIntentList = **new** ArrayList();  // 버튼 이벤트 처리  Button startBtn = (Button) findViewById(R.id.startBtn);  startBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  // register two targets  **int** countTargets = 2;  register(1001, 36.222222, 126.222222, 200, -1);  register(1002, 38.222222, 128.222222, 200, -1);  TextView textView01 = (TextView) findViewById(R.id.textView01);  textView01.setText("1001 : " + "36.222222, 126.222222");  TextView textView02 = (TextView) findViewById(R.id.textView02);  textView02.setText("1002 : " + "38.222222, 128.222222");  // 수신자 객체 생성하여 등록  mIntentReceiver = **new** CoffeeIntentReceiver(intentKey);  registerReceiver(mIntentReceiver, mIntentReceiver.getFilter());  Toast.*makeText*(getApplicationContext(), countTargets + "개 지점에 대한 근접 리스너 등록", Toast.***LENGTH\_LONG***).show();  }  });  Button stopBtn = (Button) findViewById(R.id.stopBtn);  stopBtn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  unregister();  Toast.*makeText*(getApplicationContext(), "근접 리스너 해제", Toast.***LENGTH\_LONG***).show();  }  });  }  /\*\*  \* register the proximity intent receiver  \*/  **private** **void** register(**int** id, **double** latitude, **double** longitude, **float** radius, **long** expiration) {  Intent proximityIntent = **new** Intent(intentKey);  proximityIntent.putExtra("id", id);  proximityIntent.putExtra("latitude", latitude);  proximityIntent.putExtra("longitude", longitude);  PendingIntent intent = PendingIntent.*getBroadcast*(**this**, id, proximityIntent, PendingIntent.***FLAG\_CANCEL\_CURRENT***);  mLocationManager.addProximityAlert(latitude, longitude, radius, expiration, intent);  mPendingIntentList.add(intent);  }  **public** **void** onStop() {  **super**.onStop();  unregister();  }  /\*\*  \* 등록한 정보 해제  \*/  **private** **void** unregister() {  **if** (mPendingIntentList != **null**) {  **for** (**int** i = 0; i < mPendingIntentList.size(); i++) {  PendingIntent curIntent = (PendingIntent) mPendingIntentList.get(i);  mLocationManager.removeProximityAlert(curIntent);  mPendingIntentList.remove(i);  }  }  **if** (mIntentReceiver != **null**) {  unregisterReceiver(mIntentReceiver);  mIntentReceiver = **null**;  }  }  /\*\*  \* 브로드캐스팅 메시지를 받았을 때 처리할 수신자 정의  \*/  **private** **class** CoffeeIntentReceiver **extends** BroadcastReceiver {  **private** String mExpectedAction;  **private** Intent mLastReceivedIntent;  **public** CoffeeIntentReceiver(String expectedAction) {  mExpectedAction = expectedAction;  mLastReceivedIntent = **null**;  }  **public** IntentFilter getFilter() {  IntentFilter filter = **new** IntentFilter(mExpectedAction);  **return** filter;  }  /\*\*  \* 받았을 때 호출되는 메소드  \*  \* **@param** context  \* **@param** intent  \*/  **public** **void** onReceive(Context context, Intent intent) {  **if** (intent != **null**) {  mLastReceivedIntent = intent;  **int** id = intent.getIntExtra("id", 0);  **double** latitude = intent.getDoubleExtra("latitude", 0.0D);  **double** longitude = intent.getDoubleExtra("longitude", 0.0D);  Toast.*makeText*(context, "근접한 커피숍 : " + id + ", " + latitude + ", " + longitude, Toast.***LENGTH\_LONG***).show();  }  }  **public** Intent getLastReceivedIntent() {  **return** mLastReceivedIntent;  }  **public** **void** clearReceivedIntents() {  mLastReceivedIntent = **null**;  }  }  ...  } |
| <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" /> |

주소이용

|  |
| --- |
| /\*\*  \* 주소로 위치를 찾거나 위치 좌표를 이용해 주소를 찾는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*/  **public** **class** MainActivity **extends** ActionBarActivity {  **private** **static** String *TAG* = "MainActivity";  TextView contentsText;  Geocoder gc;  EditText edit01;  EditText edit02;  EditText edit03;  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  edit01 = (EditText) findViewById(R.id.edit01);  edit02 = (EditText) findViewById(R.id.edit02);  edit03 = (EditText) findViewById(R.id.edit03);  contentsText = (TextView) findViewById(R.id.contentsText);  // 버튼 이벤트 처리  Button show\_btn = (Button) findViewById(R.id.show\_btn);  show\_btn.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  // 사용자가 입력한 주소 정보 확인  String searchStr = edit01.getText().toString();  // 주소 정보를 이용해 위치 좌표 찾기 메소드 호출  searchLocation(searchStr);  }  });  // 버튼 이벤트 처리  Button show\_btn2 = (Button) findViewById(R.id.show\_btn2);  show\_btn2.setOnClickListener(**new** View.OnClickListener() {  **public** **void** onClick(View v) {  // 사용자가 입력한 위치 좌표 확인  String LatStr = edit02.getText().toString();  String LonStr = edit03.getText().toString();  **double** latitude = 0.0D;  **double** longitude = 0.0D;  **try** {  latitude = Double.*parseDouble*(LatStr);  longitude = Double.*parseDouble*(LonStr);  } **catch**(NumberFormatException ex) {  Log.*d*(*TAG*, "예외 : " + ex.toString());  }  // 위치 좌표를 이용해 주소를 검색하는 메소드 호출  searchLocation(latitude, longitude);  }  });  // 지오코더 객체 생성  gc = **new** Geocoder(**this**, Locale.***KOREAN***);  }  /\*\*  \* 주소를 이용해 위치 좌표를 찾는 메소드 정의  \*/  **private** **void** searchLocation(String searchStr) {  // 결과값이 들어갈 리스트 선언  List<Address> addressList = **null**;  **try** {  addressList = gc.getFromLocationName(searchStr, 3);  **if** (addressList != **null**) {  contentsText.append("\nCount of Addresses for [" + searchStr + "] : " + addressList.size());  **for** (**int** i = 0; i < addressList.size(); i++) {  Address outAddr = addressList.get(i);  **int** addrCount = outAddr.getMaxAddressLineIndex() + 1;  StringBuffer outAddrStr = **new** StringBuffer();  **for** (**int** k = 0; k < addrCount; k++) {  outAddrStr.append(outAddr.getAddressLine(k));  }  outAddrStr.append("\n\tLatitude : " + outAddr.getLatitude());  outAddrStr.append("\n\tLongitude : " + outAddr.getLongitude());  contentsText.append("\n\tAddress #" + i + " : " + outAddrStr.toString());  }  }  } **catch**(IOException ex) {  Log.*d*(*TAG*, "예외 : " + ex.toString());  }  }  /\*\*  \* 위치 좌표를 이용해 주소를 검색하는 메소드 정의  \*/  **private** **void** searchLocation(**double** latitude, **double** longitude) {  List<Address> addressList = **null**;  **try** {  addressList = gc.getFromLocation(latitude, longitude, 3);  **if** (addressList != **null**) {  contentsText.append("\nCount of Addresses for [" + latitude + ", " + longitude + "] : " + addressList.size());  **for** (**int** i = 0; i < addressList.size(); i++) {  Address outAddr = addressList.get(i);  **int** addrCount = outAddr.getMaxAddressLineIndex() + 1;  StringBuffer outAddrStr = **new** StringBuffer();  **for** (**int** k = 0; k < addrCount; k++) {  outAddrStr.append(outAddr.getAddressLine(k));  }  outAddrStr.append("\n\tLatitude : " + outAddr.getLatitude());  outAddrStr.append("\n\tLongitude : " + outAddr.getLongitude());  contentsText.append("\n\tAddress #" + i + " : " + outAddrStr.toString());  }  }  } **catch**(IOException ex) {  Log.*d*(*TAG*, "예외 : " + ex.toString());  }  }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.menu\_main, menu);  **return** **true**;  }  @Override  **public** **boolean** onOptionsItemSelected(MenuItem item) {  // Handle action bar item clicks here. The action bar will  // automatically handle clicks on the Home/Up button, so long  // as you specify a parent activity in AndroidManifest.xml.  **int** id = item.getItemId();  //noinspection SimplifiableIfStatement  **if** (id == R.id.***action\_settings***) {  **return** **true**;  }  **return** **super**.onOptionsItemSelected(item);  }  } |

트위트 연동 twitter4J

* 41일 1장

페이스북 연동

* 41일 2장

푸싱서비스 push

* GCM사용 google Cloud Manager
* 구글api키받아야함
* 42일 1~2장

NFC

* 내디바이스를 태그로 사용할수있음
* 읽기쓰기가능
* 디바이스끼리 p2p로 데이터 주고받기
* 43일

블루투스

* 페어링 필요
* 44일 1장

프래그먼트 다시한번 Fragment

* onCreate(Bundle) 프로그먼트가 처음 만드어질 때 호출
* **onCreateView(LayouyInflator, ViewGroup, Bundle) 프래그머트와 관련되는 뷰 계층을 만들어서리턴**
* **onAttach(Activity) 프래그먼트가 액티비티와 연결될 떄 호출됨**
* onActivityCreated(Bundle)프래그먼트와 연결된 액티비티가 onCreate()메소드의 작업을 완료했을 때 호출
* onStart() 프래그먼트와 연결된 액티비티가 onStart()되어 사용자에게 플래그먼트가 보일 때 호출
* onResume() 프로그먼트와 연결된 액티비티가 onResume()되어 사용자와 상호 작용할 수 있을 때 호출

|  |
| --- |
| /\*\*  \* 프래그먼트를 사용하는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*/  **public** **class** MainActivity **extends** ActionBarActivity **implements** SampleListFragment.ListItemSelectedListener {  @Override  **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  }  **public** **void** onListItemSelected(**int** index) {  **SampleViewerFragment imageViewer = (SampleViewerFragment) getSupportFragmentManager().findFragmentById(R.id.image\_viewer\_fragment);**  **imageViewer.update(index);**  }  ...  }  **public** **class** SampleListFragment **extends** ListFragment {  **private** **int** index = 0;  **private** ListItemSelectedListener selectedListener;  **public** **void** onListItemClick(ListView l, View v, **int** position, **long** id) {  index = position;  selectedListener.onListItemSelected(position);  }  **public** **void** onActivityCreated(Bundle savedInstanceState) {  **super**.onActivityCreated(savedInstanceState);  setListAdapter(ArrayAdapter.*createFromResource*(getActivity(),  R.array.image\_titles, android.R.layout.***simple\_list\_item\_1***));  **if** (savedInstanceState != **null**) {  index = savedInstanceState.getInt("index", 0);  selectedListener.onListItemSelected(index);  }  }  **public** **void** onSaveInstanceState(Bundle outState) {  **super**.onSaveInstanceState(outState);  outState.putInt("index", index);  }  **public** **void** onAttach(Activity activity) {  **super**.~~onAttach~~(activity);  **try** {  selectedListener = (ListItemSelectedListener) activity;  } **catch** (ClassCastException e) {  **throw** **new** ClassCastException(activity.toString() + " must implement ListItemSelectedListener in Activity");  }  }  **public** **interface** ListItemSelectedListener {  **public** **void** onListItemSelected(**int** index);  }  }  **public** **class** SampleViewerFragment **extends** Fragment {  String[] imageTitles = {"Dream 01", "Dream 02", "Dream 03"};  **int**[] imageLocations = {R.drawable.dream01, R.drawable.dream02, R.drawable.dream03};  **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  Bundle savedInstanceState) {  **return** inflater.inflate(R.layout.image\_viewer\_fragment, container, **false**);  }  **public** **void** update(**int** index) {  TextView title = (TextView) getView().findViewById(R.id.title);  ImageView image = (ImageView) getView().findViewById(R.id.image);  title.setText(imageTitles[index]);  image.setImageResource(imageLocations[index]);  }  } |
| <fragment  android:name="org.androidtown.ui.fragment.SampleViewerFragment"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:id="@+id/image\_viewer\_fragment"  android:layout\_weight="75">  </fragment>  <RelativeLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">    <TextView  android:id="@+id/title"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:layout\_alignParentTop="true"  android:gravity="center"  >  </TextView>  <ImageView  android:id="@+id/image"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:src="@drawable/ic\_launcher"  android:layout\_below="@+id/title"  android:layout\_alignLeft="@+id/title"  android:layout\_alignRight="@+id/title"  android:scaleType="fitCenter">  </ImageView>    </RelativeLayout> |

앱위젯만들기

* App Widget Host (그릇)
* App Widget Provider (제공자.)
* 매니페스트에 액티비티가 없다!

|  |
| --- |
|  |
| **public** **class** MyLocationProvider **extends** **AppWidgetProvider** {  **public** **static** **double** *ycoord* = 0.0D;  **public** **static** **double** *xcoord* = 0.0D;  @Override  **public** **void** onDeleted(Context context, **int**[] appWidgetIds) {  **super**.onDeleted(context, appWidgetIds);  }  @Override  **public** **void** onDisabled(Context context) {  **super**.onDisabled(context);  }  @Override  **public** **void** onEnabled(Context context) {  **super**.onEnabled(context);  }  @Override  **public** **void** onReceive(Context context, Intent intent) {  **super**.onReceive(context, intent);  }  @Override  **public** **void** onUpdate(Context context, AppWidgetManager appWidgetManager, **int**[] appWidgetIds) {  **super**.onUpdate(context, appWidgetManager, appWidgetIds);  Log.*d*("MyLocationProvider", "onUpdate() called : " + *ycoord* + ", " + *xcoord*);  **final** **int** size = appWidgetIds.length;  **for** (**int** i = 0; i < size; i++) {  **int** appWidgetId = appWidgetIds[i];  //String uri = "geo:"+ ycoord + "," + xcoord + "?z=10";  //Intent intent = new Intent(android.content.Intent.ACTION\_VIEW, Uri.parse(uri));      String uriBegin = "geo:" + *ycoord* + "," + *xcoord*;  String query = *ycoord* + "," + *xcoord* + "(" + "내위치" + ")";  String encodedQuery = Uri.*encode*(query);  String uriString = uriBegin + "?q=" + encodedQuery + "&z=15";  Uri uri = Uri.*parse*(uriString);    Intent intent = **new** Intent(Intent.***ACTION\_VIEW***, uri);      PendingIntent pendingIntent = PendingIntent.*getActivity*(context, 0, intent, 0);  **RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.mylocation);**  **views.setOnClickPendingIntent(R.id.txtInfo, pendingIntent);**  appWidgetManager.updateAppWidget(appWidgetId, views);  }  context.startService(**new** Intent(context,GPSLocationService.**class**));  }  **public** **static** **class** GPSLocationService **extends** Service {  **public** **static** **final** String ***TAG*** = "GPSLocationService";  **private** LocationManager manager = **null**;  **private** LocationListener listener = **new** LocationListener() {  **public** **void** onStatusChanged(String provider, **int** status, Bundle extras) {  }  **public** **void** onProviderEnabled(String provider) {  }  **public** **void** onProviderDisabled(String provider) {  }  **public** **void** onLocationChanged(Location location) {  Log.*d*(***TAG***, "onLocationChanged() called.");  updateCoordinates(location.getLatitude(), location.getLongitude());  stopSelf();  }  };  **public** IBinder onBind(Intent intent) {  **return** **null**;  }  **public** **void** onCreate() {  **super**.onCreate();  Log.*d*(***TAG***, "onCreate() called.");  manager = (LocationManager) getSystemService(***LOCATION\_SERVICE***);  }  **public** **void** onStart(Intent intent, **int** startId) {  **super**.~~onStart~~(intent, startId);  startListening();  }  **public** **int** onStartCommand(Intent intent, **int** flags, **int** startId) {  startListening();  **return** **super**.onStartCommand(intent, flags, startId);  }  **public** **void** onDestroy() {  stopListening();  Log.*d*(***TAG***, "onDestroy() called.");  **super**.onDestroy();  }  **private** **void** startListening() {  Log.*d*(***TAG***, "startListening() called.");  **final** Criteria criteria = **new** Criteria();  criteria.setAccuracy(Criteria.***ACCURACY\_COARSE***);  criteria.setAltitudeRequired(**false**);  criteria.setBearingRequired(**false**);  criteria.setCostAllowed(**true**);  criteria.setPowerRequirement(Criteria.***POWER\_LOW***);  **final** String bestProvider = manager.getBestProvider(criteria, **true**);  **if** (bestProvider != **null** && bestProvider.length() > 0) {  manager.requestLocationUpdates(bestProvider, 500, 10, listener);  } **else** {  **final** List<String> providers = manager.getProviders(**true**);  **for** (**final** String provider : providers) {  manager.requestLocationUpdates(provider, 500, 10, listener);  }  }  }  **private** **void** stopListening() {  **try** {  **if** (manager != **null** && listener != **null**) {  manager.removeUpdates(listener);  }  manager = **null**;  } **catch** (**final** Exception ex) {  }  }  **private** **void** updateCoordinates(**double** latitude, **double** longitude) {  Geocoder coder = **new** Geocoder(**this**);  List<Address> addresses = **null**;  String info = "";  Log.*d*(***TAG***, "updateCoordinates() called.");  **try** {  addresses = coder.getFromLocation(latitude, longitude, 2);  **if** (**null** != addresses && addresses.size() > 0) {  **int** addressCount = addresses.get(0).getMaxAddressLineIndex();  **if** (-1 != addressCount) {  **for** (**int** index = 0; index <= addressCount; ++index) {  info += addresses.get(0).getAddressLine(index);  **if** (index < addressCount)  info += ", ";  }  } **else** {  info += addresses.get(0).getFeatureName() + ", "  + addresses.get(0).getSubAdminArea() + ", "  + addresses.get(0).getAdminArea();  }  }  Log.*d*(***TAG***, "Address : " + addresses.get(0).toString());  } **catch** (Exception e) {  e.printStackTrace();  }  coder = **null**;  addresses = **null**;  **if** (info.length() <= 0) {  info = "[내 위치] " + latitude + ", " + longitude  + "\n터치하면 지도로 볼 수 있습니다.";  } **else** {  info += ("\n" + "[내 위치] " + latitude + ", " + longitude + ")");  info += "\n터치하면 지도로 볼 수 있습니다.";  }  RemoteViews views = **new** RemoteViews(getPackageName(), R.layout.mylocation);  views.setTextViewText(R.id.txtInfo, info);  ComponentName thisWidget = **new** ComponentName(**this**, MyLocationProvider.**class**);  AppWidgetManager manager = AppWidgetManager.*getInstance*(**this**);  manager.updateAppWidget(thisWidget, views);  *xcoord* = longitude;  *ycoord* = latitude;  Log.*d*(***TAG***, "coordinates : " + latitude + ", " + longitude);  }  }  } |
| mylocation.xml  <?xml version=*"1.0"* encoding=*"utf-8"*?>  <LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*  android:layout\_width=*"fill\_parent"*  android:layout\_height=*"fill\_parent"*>  <LinearLayout  android:layout\_width=*"fill\_parent"*  android:layout\_height=*"fill\_parent"*  android:background=*"@drawable/background"*  android:padding=*"10dp"*>  <TextView  android:layout\_width=*"fill\_parent"*  android:layout\_height=*"fill\_parent"*  android:id=*"@+id/txtInfo"*  android:text=*"내 위치 정보 수신중 ..."*  android:textColor=*"#FFFFFFFF"*  android:gravity=*"center\_horizontal|center\_vertical"*  android:lineSpacingExtra=*"4dp"*/>  </LinearLayout>  </LinearLayout> |
| mylocationinfo.xml  <appwidget-provider xmlns:android=*"http://schemas.android.com/apk/res/android"*  android:minWidth=*"294dp"*  android:minHeight=*"72dp"*  android:updatePeriodMillis=*"1800000" 🡨 업데이트 시간*  android:initialLayout=*"@layout/mylocation"*>  </appwidget-provider> |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"org.androidtown.appwidget.location"* >  <uses-permission android:name=*"android.permission.INTERNET"*/>  <uses-permission android:name=*"android.permission.ACCESS\_FINE\_LOCATION"*/>  <uses-permission android:name=*"android.permission.ACCESS\_COARSE\_LOCATION"*/>  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  **<receiver android:name=*".MyLocationProvider"*>**  **<intent-filter>**  **<action android:name=*"android.appwidget.action.APPWIDGET\_UPDATE"* />**  **</intent-filter>**  **<meta-data android:name=*"android.appwidget.provider"***  **android:resource=*"@xml/mylocationinfo"* />**  **</receiver>**  <service android:name=*".MyLocationProvider$GPSLocationService"*></service>  </application>  </manifest> |

라이브 배경화면

|  |
| --- |
| /\*\*  \* 라이브 배경화면을 만드는 방법에 대해 알 수 있습니다.  \*  \* **@author** Mike  \*/  **public** **class** SampleWallpaperService **extends WallpaperService** {  **private** Handler handler = **null**;  **public** SampleWallpaperService() {  handler = **new** Handler();  }  **public** Engine onCreateEngine() {  **return** **new** SampleWallpaperEngine();  }  **private** **class** SampleWallpaperEngine **extends** Engine {  **public** **static** **final** String ***TAG*** = "SampleWallpaperService";  **private** **boolean** visible = **false**;  **private** **int**[] colors = {0, 0, 0} ;  **private** **final** Runnable runnable = **new** Runnable() {  **public** **void** run() {  drawBackground();  }  };  **private** **void** drawBackground() {  SurfaceHolder holder = getSurfaceHolder();  Canvas canvas = **null**;  **try** {  canvas = holder.lockCanvas();  **if** (canvas != **null**) {  canvas.drawARGB(200, colors[0], colors[1], colors[2]);  }  updateBackground(colors);  } **finally** {  **if** (canvas != **null**) {  holder.unlockCanvasAndPost(canvas);  }  }  handler.removeCallbacks(runnable);  **if** (visible) {  handler.postDelayed(runnable, 25);  }  }  **private** **void** updateBackground(**int**[] colors) {  **if** (colors[2] < 128) {  colors[2]++;  } **else** **if** (colors[1] < 128) {  colors[1]++;  } **else** **if** (colors[0] < 128) {  colors[0]++;  } **else** {  resetColors();  }  }  **private** **void** resetColors() {  colors[0] = 0;  colors[1] = 0;  colors[2] = 0;  }  **public** **void** onDestroy() {  **super**.onDestroy();  handler.removeCallbacks(runnable);  }  **public** **void** onVisibilityChanged(**boolean** visible) {  **super**.onVisibilityChanged(visible);  Log.*d*(***TAG***, "onVisibilityChanged() called : " + visible);  **this**.visible = visible;  **if** (visible) {  drawBackground();  } **else** {  handler.removeCallbacks(runnable);  }  }  **public** **void** onTouchEvent(MotionEvent event) {  Log.*d*(***TAG***, "onTouchEvent() called : " + event.getX() + ", " + event.getY());  resetColors();  **super**.onTouchEvent(event);  }  }  } |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*  package=*"org.androidtown.ui.wallpaper"* >  <uses-feature android:name=*"android.software.live\_wallpaper"* />  <application  android:allowBackup=*"true"*  android:icon=*"@drawable/ic\_launcher"*  android:label=*"@string/app\_name"*  android:theme=*"@style/AppTheme"* >  <service android:label=*"@string/app\_name"*  android:name=*".SampleWallpaperService"*  android:**permission=*"android.permission.BIND\_WALLPAPER"*>**  <intent-filter>  <action android:name=*"android.service.wallpaper.WallpaperService"* />  </intent-filter>  <**meta-data android:name=*"android.service.wallpaper"***  **android:resource=*"@xml/wallpaper"* />**  </service>  </application>  </manifest> |
| wallpaper.xml  <?xml version=*"1.0"* encoding=*"utf-8"*?>  <wallpaper  xmlns:android=*"http://schemas.android.com/apk/res/android"*  android:thumbnail=*"@drawable/ic\_launcher"*  /> |