

XML Parser

6조 20080945 이성훈
20081005 이재환
20111519 김기순

목차

1. Parsing의 정의, 필요성
2. DOM Parser
3. SAX Parser
4. XML Pull Parser

1. Parsing의 정의, 필요성

Parsing이란?

주어진 문장을 분석하거나 문법적 관계를 해석하는 것

Parsing의 필요성

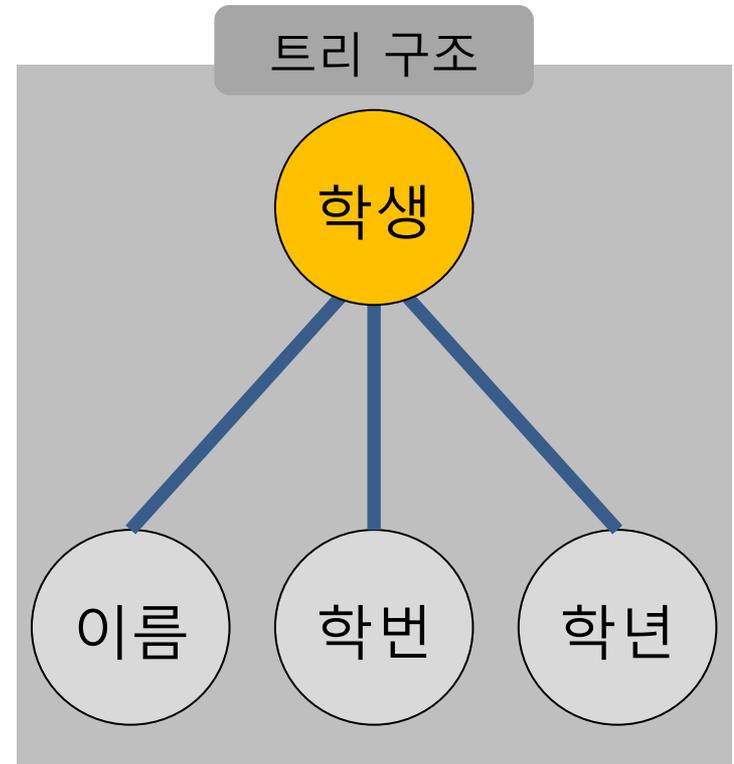
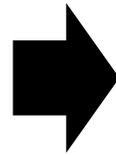
- 프로그래밍이 간편
- 플랫폼 독립적 프로그래밍 언어에 구애 받지 않음
- 필요한 데이터를 빠르게 처리 가능
- 웹상의 XML이 수정되어도 프로그램을 변경하지 않아도 됨

2. DOM Parser

DOM 파서란?

문서의 모든 내용을 해석한 후 메모리에
트리 구조로 펼친 후 읽어 들임

```
<학생>  
  <이름>홍길동</이름>  
  <학번>12345678</학번>  
  <학년>2학년</학년>  
</학생>
```



DOM 파서 장단점

장점

특정 노드에 대한 임의접근이 자유롭다.
원하는 노드를 몇 번이고 읽을 수 있다.
문서의 수정도 가능하다.

단점

메모리 사용이 많다.
처음시작이 다소 느리다.
(문서 전체를 로딩해야 하므로)

설명

XML 문서를 열기 위한 최초 작업

1. DocumentBuilderFactory 객체 생성

생성 방법: 정적(static) 메소드 **newInstance()**를 통해 생성

생성 예: DocumentBuilderFactory **factory** = DocumentBuilderFactory.newInstance();

2. DocumentBuilder 객체 생성

생성 방법: **newDocumentBuilder()** 메소드를 통해 생성

생성 예: DocumentBuilder **builder** = **factory.newDocumentBuilder()**;

두 작업을 마쳤다면 이제 XML 문서를 파싱 할 준비가 끝남

설명

앞서 만든 **builder** 객체로 **parse()** 메소드 호출

Document parse (InputStream **stream**)

Document parse (String **URI**)

Document parse (File **file**)



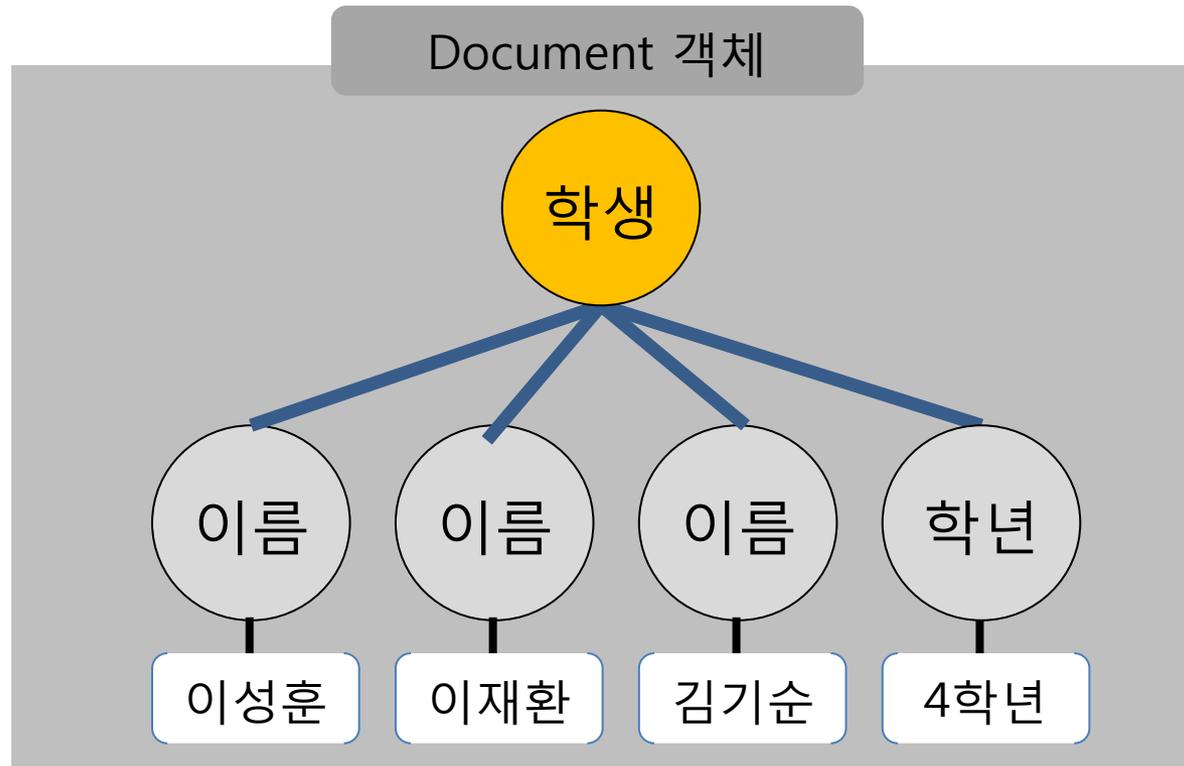
매개변수

작성 예: Document **doc** = **builder.parse(stream)**;

1. Parse() 메소드는 받은 문서를 분석하여 메모리에 트리형태로 전개
2. 트리구조 정보를 담고있는 Document 객체 반환

접근 방법

**Document 객체를 얻었다면
요소에 빠른 속도로 접근할 수 있다.**

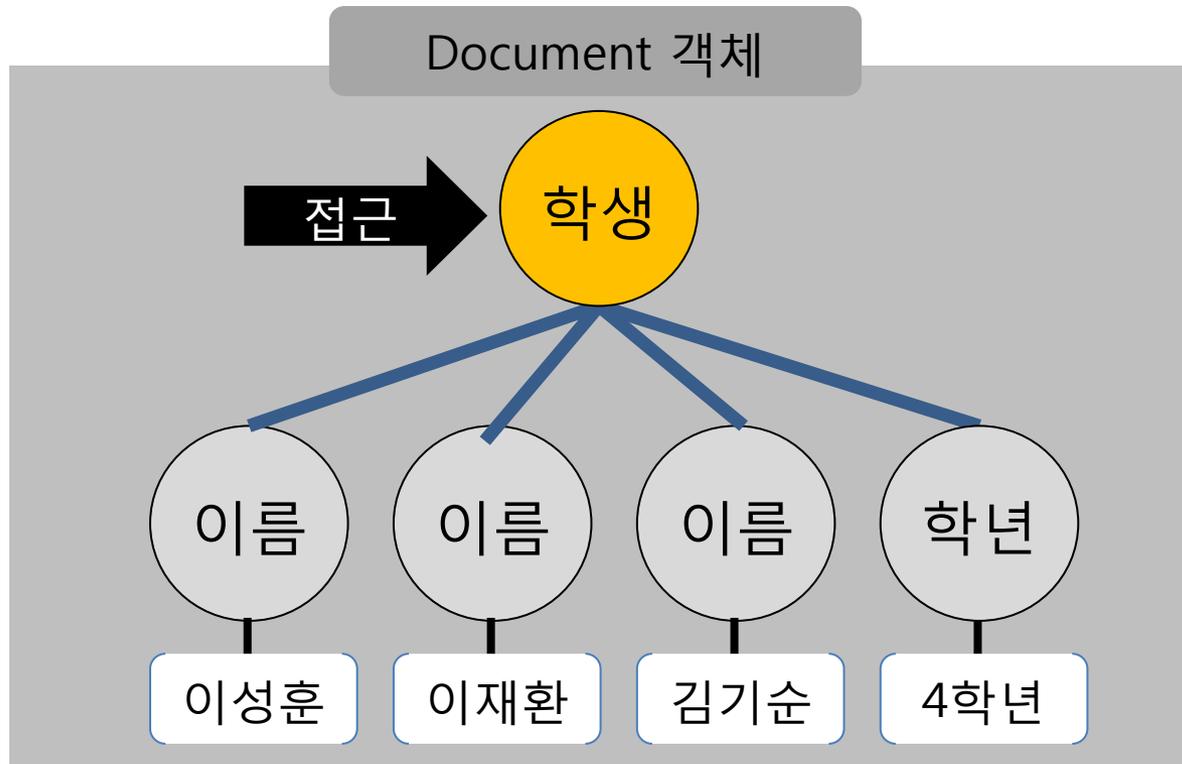


접근 방법

Document 객체 내 **루트 엘리먼트에 접근하기 위한** 메소드

Element `getDocumentElement()`

사용 예: Element **root** = doc.`getDocumentElement()`;

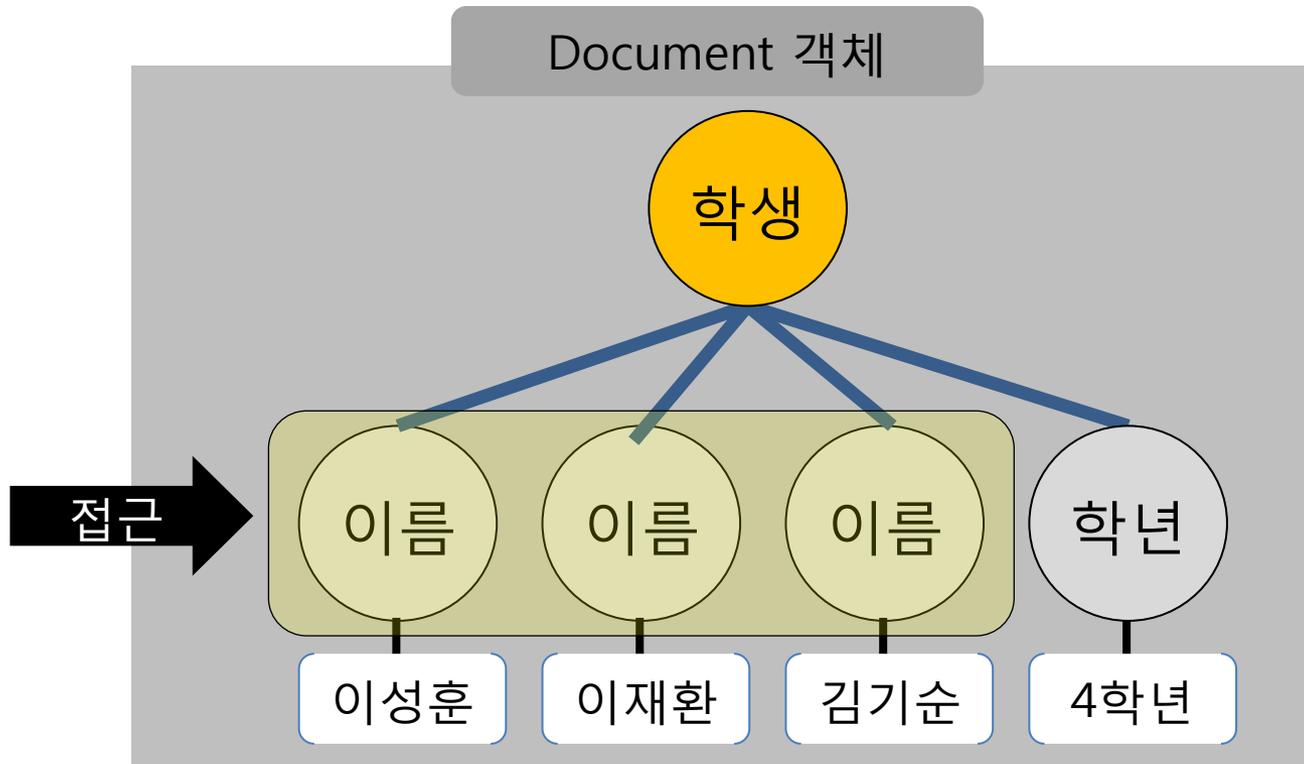


접근 방법

루트에서 태그명과 일치하는 모든 자식요소를 찾는 메소드

NodeList `getElementsByTagName("태그명")`

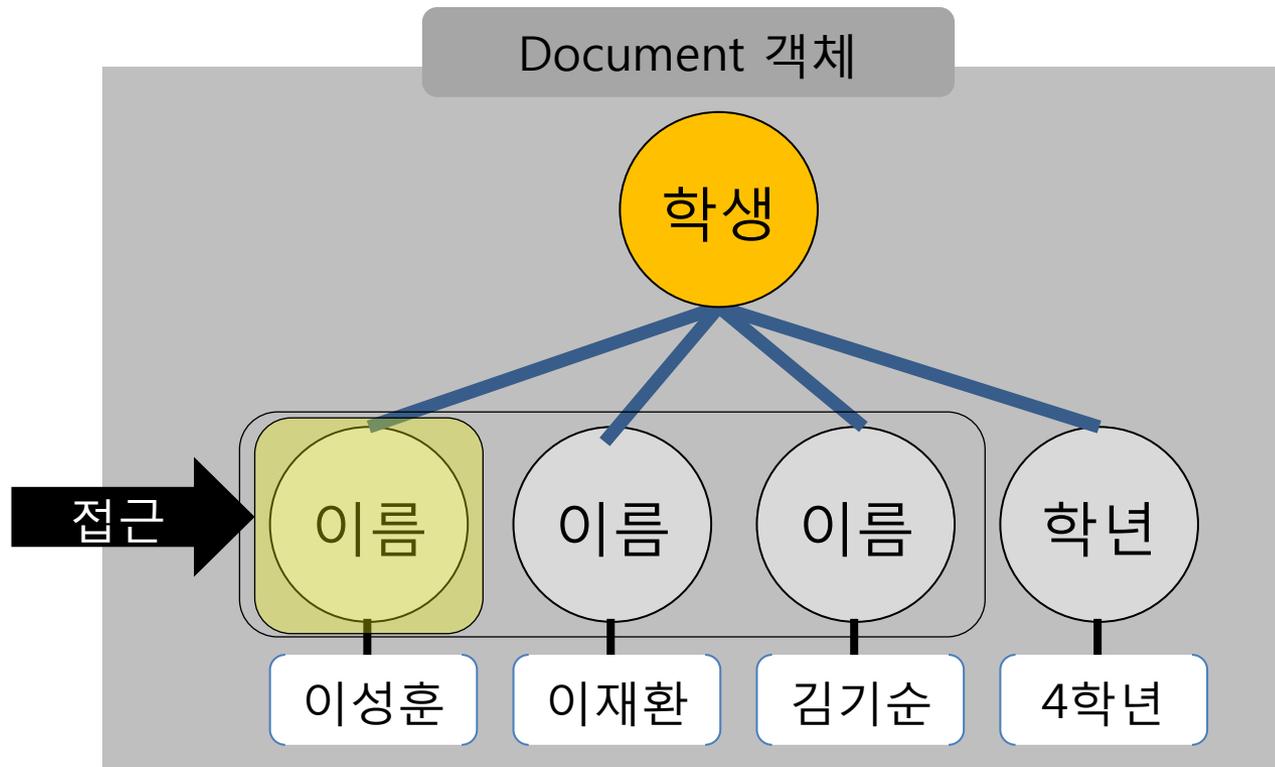
사용 예: `NodeList childs = root.getElementsByTagName("이름");`



접근 방법

노드 리스트에서 **하나의 요소**를 선택하는 방법
노드 리스트.item(번호)

접근 예: Node **child** = childs.item(0)



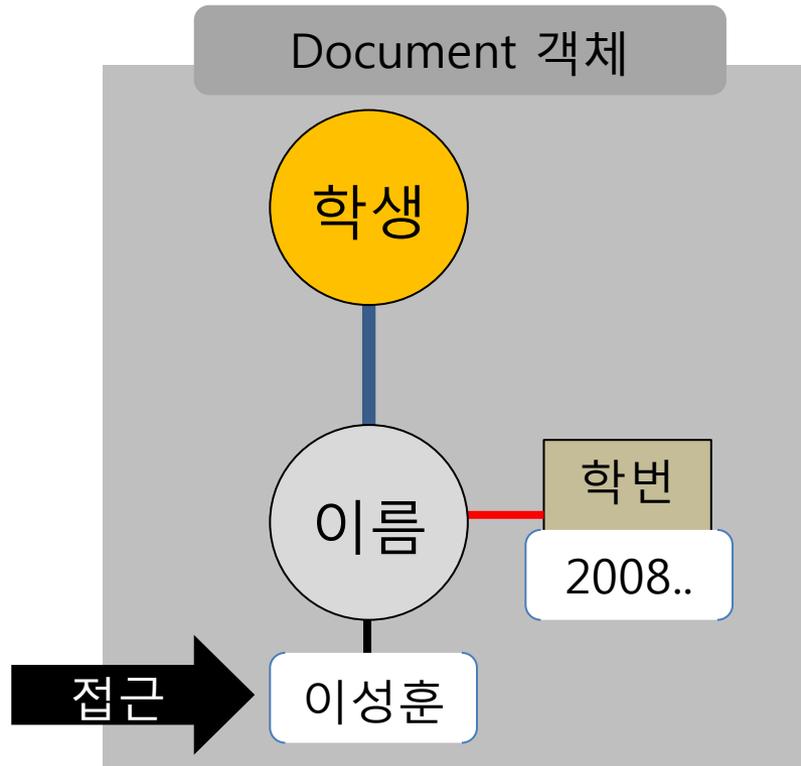
첫 번째 이름요소에 접근

접근 방법

요소의 내용에 접근

노드명.getFirstChild()

접근 예: Node **childvalue** = child.getFirstChild()



DOM에서는 요소의 내용 값을 요소의 자식노드로 취급한다.

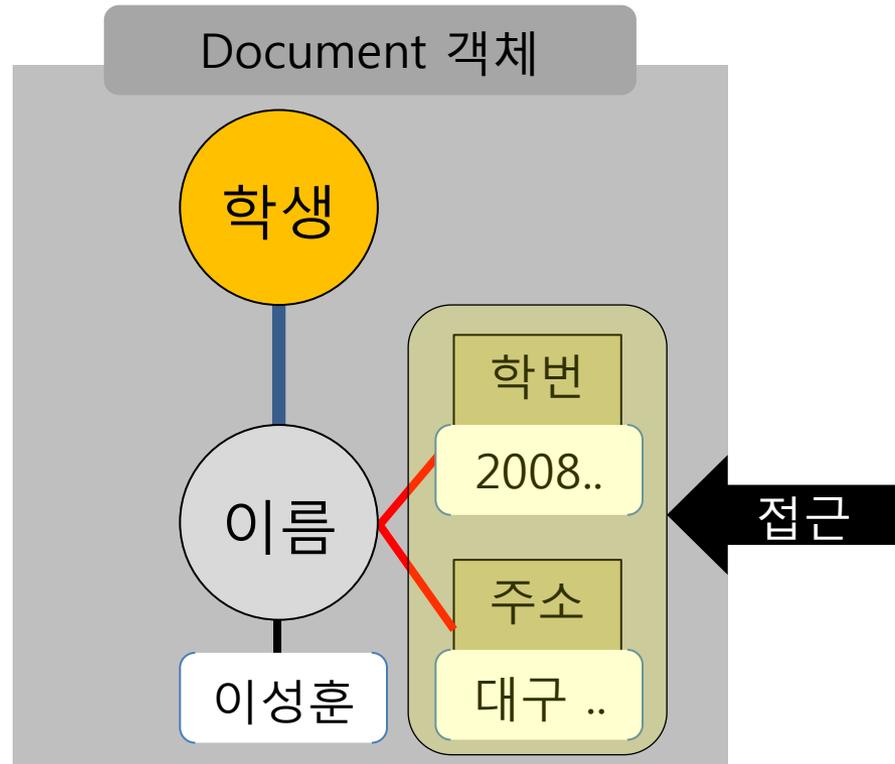
(단, 요소의 속성은 위 규칙을 따르지 않는다. 위 그림에서는 학번에 해당)

접근 방법

요소의 **모든 속성**에 접근하는 메소드

NamedNodeMap **getAttributes()**

사용 예: NamedNodeMap Attrs= child.**getAttributes()**

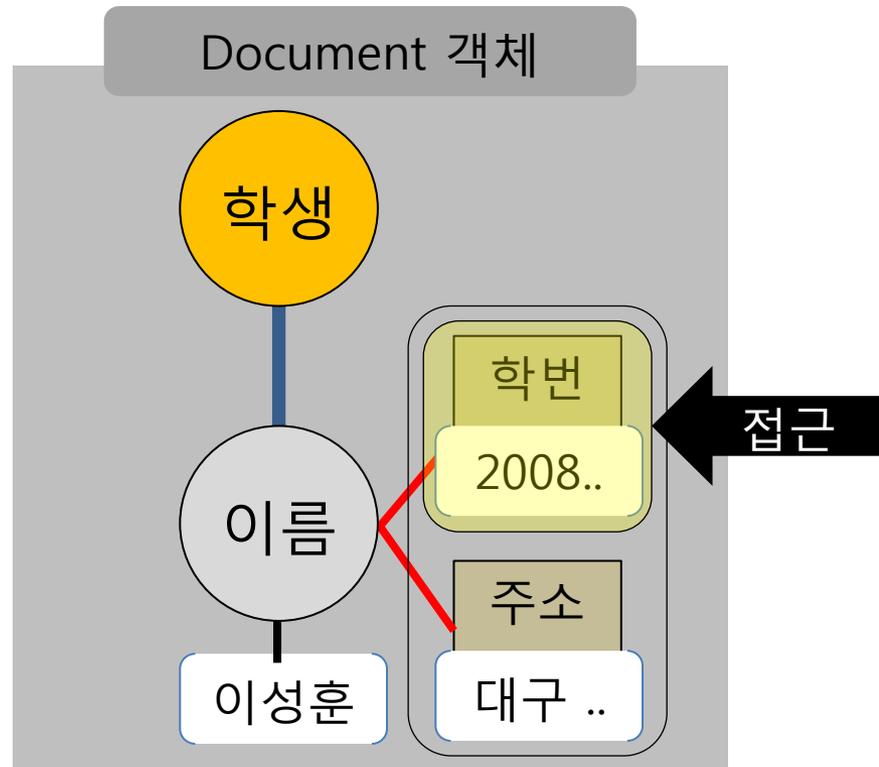


접근 방법

속성목록에서 **하나의 속성**에 접근하는 방법

속성목록.item(번호)

사용 예: Node attr= Attrs.item(0)



접근 방법

그 밖의 노드관련 주요 메소드

String getNodeName() - 노드 이름

Short getNodeType() - 노드 타입

String getNodeValue() - 노드 값

노드의 이름,타입,값 반환

Node getFirstChild() - 첫 번째 자식노드

Node getLastChild() - 마지막 자식노드

Node getNextSibling() - 자신의 다음노드

Node getPreviousSibling() - 자신의 이전노드

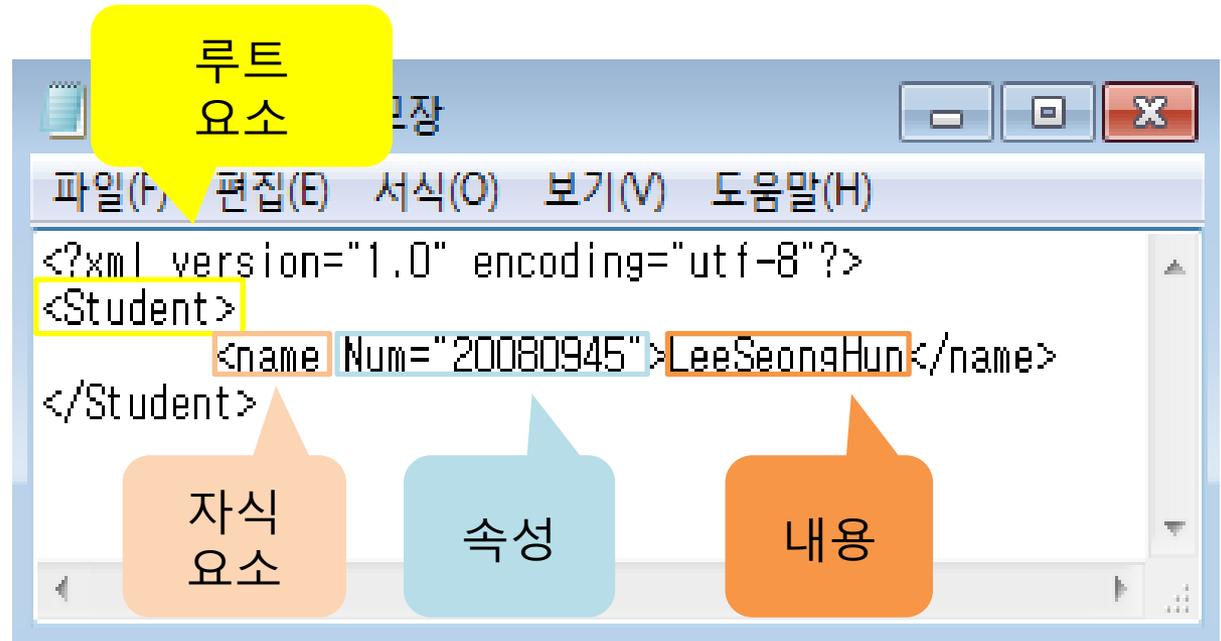
Node getParentNode() - 자신의 부모노드

NodeList getChildNodes() - 모든 자식노드

노드를 중심으로 주변 노드를 찾는 메소드

예제 수행

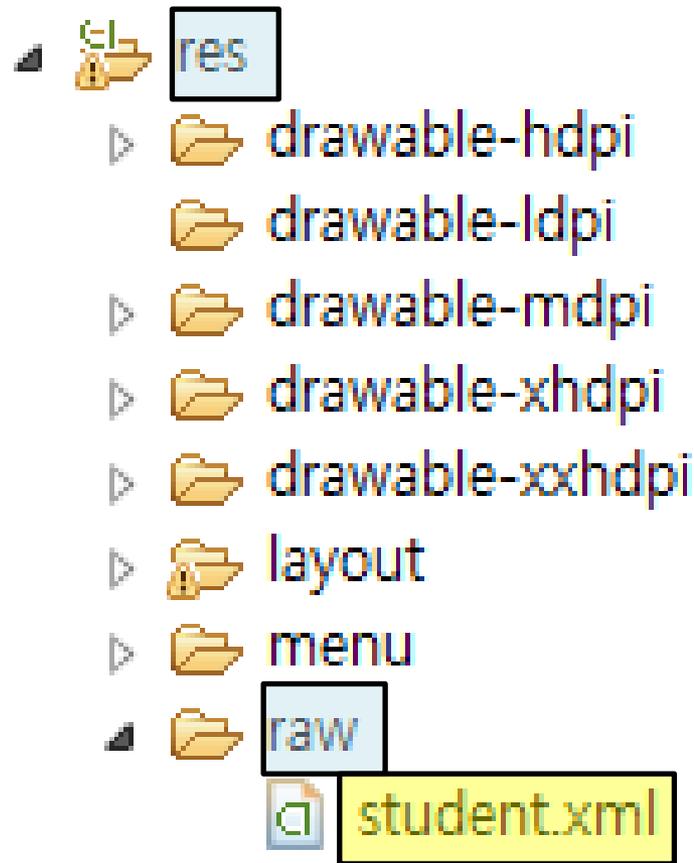
버튼을 누르면 XML문서를 파싱하는 프로그램



메모장으로 student.xml 작성

예제 수행 환경

이클립스에서 안드로이드 프로젝트 생성 후



res – raw 디렉터리에 저장

예제 수행 환경

The image shows a screenshot of an Android development environment. On the left, the XML code for 'activity_main.xml' is displayed. The code defines a vertical LinearLayout containing a Button and a TextView. The Button has the text 'Parse XML' and is linked to an 'onClick' method. The TextView is intended to display the result. On the right, a preview window titled 'DomParser' shows a button labeled 'Parse XML' and a large empty area labeled 'Result' below it. Red and blue boxes highlight the XML tags and the app interface elements respectively.

```
activity_main.xml
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <Button
        android:id="@+id/parse"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="mOnClick"
        android:text="Parse XML"
    />
    <TextView
        android:id="@+id/result"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textSize="12sp"
        android:text="Result"
    />
</LinearLayout>
```

DomParser

Parse XML

Result

버튼 하나와 텍스트 뷰로 이루어진 리니어 레이아웃

코드 작성

```
MainActivity.java ✖  
  
package com.example.domparser;  
  
import java.io.ByteArrayInputStream;  
  
public class MainActivity extends Activity {  
    TextView mResult;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mResult = (TextView)findViewById(R.id.result);  
    }  
  
    public void mOnClick(View v){  
    }  
}
```

버튼의 onClick 리스너안에 DOM 파싱 코드 작성

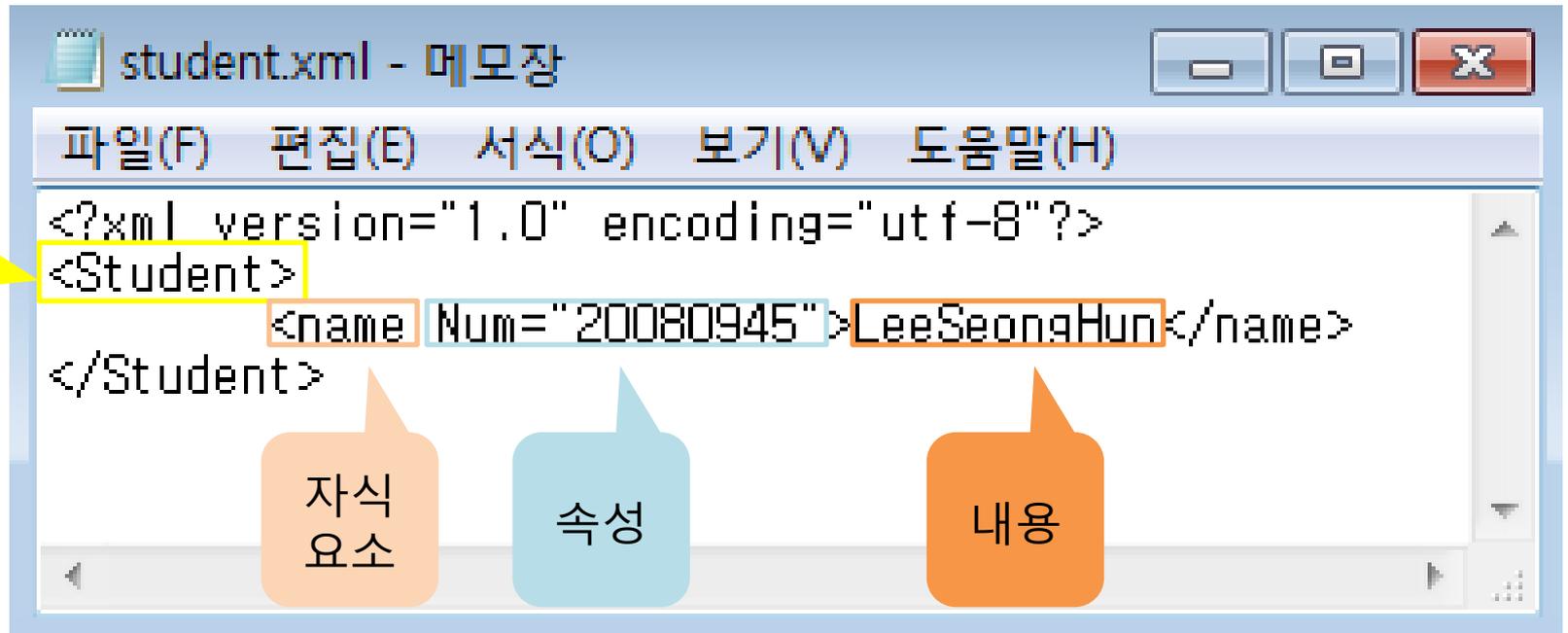
코드 작성

```
try{  
  ① // xml문서를 파싱하기위한 준비  
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
    DocumentBuilder builder = factory.newDocumentBuilder();  
  
    // raw디렉터리 내의 student.xml파일을 읽어와 스트림에 저장  
    Resources res = getResources();  
    InputStream istream = res.openRawResource(R.raw.student);  
  
  ② // parse메서드로 스트림을 파싱하여 Document 객체를 얻음  
    Document doc = builder.parse(istream);
```

문서를 읽어 트리 형태로 통합된 Document 객체를 얻는 과정

예제 수행

이쯤에서 다시 보는 student.xml



The screenshot shows a Notepad window titled "student.xml - 메모장". The menu bar includes "파일(F)", "편집(E)", "서식(O)", "보기(V)", and "도움말(H)". The XML content is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Student>
  <name Num="20080945">LeeSeongHun</name>
</Student>
```

Annotations in the image include:

- A yellow callout bubble on the left points to the root element `<Student>` with the text "루트 요소".
- An orange callout bubble points to the `Num` attribute with the text "자식 요소".
- A light blue callout bubble points to the `>` symbol with the text "속성".
- An orange callout bubble points to the text `LeeSeongHun` with the text "내용".

코드 작성

```
Element root = doc.getDocumentElement(); // 루트 엘리먼트
NodeList childs = root.getElementsByTagName("name");
// 루트 엘리먼트에서 name이란 이름을 가진 요소 전체
Node child = childs.item(0); // 첫번째 name 요소

NamedNodeMap Attrs = child.getAttributes(); // name의 속성값
Node attr = Attrs.item(0); // name의 첫번째 속성

Node childvalue = child.getFirstChild(); // 내용도 하나의 자식으로 취급

String rootName = root.getNodeName(); // 루트 명
String childName = child.getNodeName(); // 자식 요소 명
String childValue = childvalue.getNodeValue(); // 자식 요소 값
String AttrName = attr.getNodeName(); // 속성 명
String AttrValue = attr.getNodeValue(); // 속성 값
mResult.setText("루트요소 이름 : " + rootName +
               "\n자식요소 이름 : " + childName +
               "\n자식요소 내용 : " + childValue +
               "\n자식요소 속성명 : " + AttrName +
               "\n자식요소 속성값 : " + AttrValue);
```

이제 Student.xml의 각 요소 값을 출력해보자.

실행 결과



DomParser

Parse XML

Result



DomParser

Parse XML

루트요소 이름 :Student
자식요소 이름 :name
자식요소 내용 :LeeSeongHun
자식요소 속성명 :Num
자식요소 속성값 :20080945

Parse XML 버튼을 누르면 내용 출력

실제 웹 문서 파싱해보기

한경닷컴

<http://rss.hankyung.com/>

파싱할 임의의
뉴스 선택

 test
헤드라인 조회
마오밍시 화학공장 반대시위...광저우 선전으로 확산
중국 '사정 태풍' 속 고위 공직자 투신자살
日문부상 "자국 영토 올바르게 가르치는 것 당연"
중국이 북한을 쉽게 버려선 안 되는 3대 이유는
교육부, 日초등교과서 '독도 왜곡' 규탄..."비교육적 행위"
[일지] 일본 '역사 왜곡 교과서' 파문史 들여다보니...
아베 정권, 日초등교과서 '독도 점령' 야욕...'과거 미화' 강화

뉴스		
증권	http://rss.hankyung.com/new/news_stock.xml	
경제/금융	http://rss.hankyung.com/new/news_economy.xml	
부동산	http://rss.hankyung.com/new/news_estate.xml	
산업	http://rss.hankyung.com/new/news_industry.xml	
국제	http://rss.hankyung.com/new/news_intl.xml	
정치	http://rss.hankyung.com/new/news_politics.xml	
사회	http://rss.hankyung.com/new/news_society.xml	
스포츠/문화	http://rss.hankyung.com/new/news_sports.xml	
사설/칼럼	http://rss.hankyung.com/new/news_column.xml	
인기뉴스		
포토	http://rss.hankyung.com/new/news_photo.xml	
화제의 뉴스	http://rss.hankyung.com/new/news_hot.xml	
칼럼 및 콘텐츠		
한경 칼럼	http://rss.hankyung.com/new/column_all.xml	
증권 칼럼	http://rss.hankyung.com/new/column_stock.xml	
부동산칼럼	http://rss.hankyung.com/new/column_land.xml	
재테크칼럼	http://rss.hankyung.com/new/column_ft.xml	
자동차칼럼	http://rss.hankyung.com/new/column_auto.xml	
골프 칼럼	http://rss.hankyung.com/new/column_golf.xml	

실제 웹 문서 파싱해보기

HanKyung | 뉴스 | 국제

자주 업데이트되는 콘텐츠를 가진 피드를 보고 있습니다. 피드에 가입하면 일반 피드 목록에 추가됩니다. 피드의 업데이트된 정보는 자동으로 사용자의 컴퓨터로 다운로드되며 Internet Explorer 또는 다른 프로그램에서 볼 수 있습니다. 피드에 대해 자세히 알아봅니다.

★ 이 피드에 가입

서울-베이징 '미세먼지 개선 협력' 첫 공동합의
2014년 4월 4일 내일, 오전 3:08:24 →
통합위원회 환경팀 신설-대기질 포럼 개최서울시와 중국 베이징(北京)시가 미세먼지를 포함한 대기 질 문제에 공동 대응을 약속하며, 서울과 베이징이 대기 질과 관련해 외
국도시와 협력 합의문을 체결한 건 이번이 처음이

伊 엔지니어 419명 이베이에 경매로 올라와
2014년 4월 4일 내일, 오전 2:20:48 →
미국계 회사에서 근무하다 실직 위기에 놓인 이탈리아 엔지니어 419명이 경매사이트 이베이에 매물로 올라왔다. 이 엔지니어들은 컴퓨터과학자 등은 회사가 지난 1월 이탈리아 전역에
근무하는 엔지니어,

2009년 총기참사 美軍기지서 또 무차별총격...4명 사망
2014년 4월 4일 내일, 오전 1:42:01 →
용의자 난사 후 자살 최소 16명 부상...오바마 "철저 조사"지난 2009년 총격 사건으로 13명이 숨진 미국 텍사스주
건이 발생해 4명이 사망하고 최소 16명이 부상했다.미국
다시 무차별 총격 사

칠레지진 쓰나미 일본 도달...최고 60cm 관측
2014년 4월 4일 내일, 오전 1:23:33 →
칠레 북부의 태평양 연안에서 발생한 규모 8.2의 강진으로 인한 쓰나미가 3일 아침 일본에 도달했다.쓰나미의 최대 파고는 60cm로 관측됐다.쓰나미는 北海道)에서부터 지바
(千葉)현까지의 광범위한 태평양 연안부

뒤로(B)
앞으로(O)
복사한 주소로 이동(D) Ctrl+Shift+L
다른 이름으로 배경 저장(S)...
배경으로 설정(G)
배경 화면 복사(C)
모두 선택(A)
붙여넣기(P)
Bing 번역 서비스
Windows Live 전자 메일
모든 바로 연결
바로 가기 만들기(T)
소스 보기(V)
요소 검사(L)
인코딩(E)
인쇄(I)...
인쇄 미리 보기(N)...
새로 고침(R)
속성(P)

마우스 오른쪽버튼 클릭 → 소스보기 선택

실제 웹 문서 파싱해보기

The screenshot shows an Internet Explorer browser window with the address bar displaying `http://rss.hankyung.com/`. The main content area shows the XML code of an RSS feed. The XML code includes a channel section with metadata and several item sections with titles, links, and descriptions. The right sidebar displays a list of news items, each with a title and a date.

```
<?xml version="1.0" encoding="UTF-8"?><rss version="2.0">
  <channel>
    <title><![CDATA[Hankyung | 뉴스 | 국제]]></title>
    <link>http://www.hankyung.com/news/intl/ </link>
    <language>ko</language>
    <copyright>Copyright (c) 2005 hankyung.com All rights reserved</copyright>
    <pubDate>2014-04-03 18:40:11</pubDate>
    <lastBuildDate>2014-04-03 18:40:11</lastBuildDate>
    <description>한경닷컴 RSS 서비스</description>
    <image>
      <title>한경닷컴 RSS 서비스</title>
      <url>http://img.hankyung.com/img/bi_top_news02.gif</url>
      <link>http://www.hankyung.com/news/intl/ </link>
    </image>
    <item>
      <title><![CDATA[서울-베이징 '미세먼지 개선 협력' 첫 공동합의]]></title>
      <link><![CDATA[http://www.hankyung.com/news/app/newsview.php?aid=2014040350538&sid=0105&aid=2014040350538]]></link>
      <description><![CDATA[통합위원회 환경팀 신설·대기질 포럼 개최서울시와 중국 베이징(北京)시가]></description>
      <author><![CDATA[]]></author>
      <pubDate>2014-04-03 18:08:24</pubDate>
    </item>
    <item>
      <title><![CDATA[伊 엔지니어 419명 이베이에 경매로 올라와]]></title>
      <link><![CDATA[http://www.hankyung.com/news/app/newsview.php?aid=2014040363258&sid=0105&aid=2014040363258]]></link>
      <description><![CDATA[미국계 회사에서 근무하다 실직 위기에 놓인 이탈리아 엔지니어 419명이 경]></description>
      <author><![CDATA[]]></author>
      <pubDate>2014-04-03 17:20:48</pubDate>
    </item>
    <item>
      <title><![CDATA[2009년 총기참사 美軍기지서 또 무차별총격...4명 사망]]></title>
      <link><![CDATA[http://www.hankyung.com/news/app/newsview.php?aid=2014040361748&sid=0105&aid=2014040361748]]></link>
      <description><![CDATA[용의자 난사 후 자살·최소 16명 부상...오바마 "철저 조사"지난 2009년 총격]></description>
      <author><![CDATA[]]></author>
      <pubDate>2014-04-03 16:42:01</pubDate>
    </item>
    <item>
      <title><![CDATA[칠레지진 쓰나미 일본 도달...최고 60cm 관측]]></title>
      <link><![CDATA[http://www.hankyung.com/news/app/newsview.php?aid=2014040360398&sid=0105&aid=2014040360398]]></link>
      <description><![CDATA[칠레 북부의 태평양 연안에서 발생한 규모 8.2의 강진으로 인한 쓰나미가 3일]></description>
      <author><![CDATA[]]></author>
    </item>
  </channel>
</rss>
```

The right sidebar shows a list of news items with the following titles and dates:

- 2014년 4월 4일 내일, 오전 3: 통합위원회 환경팀 신설·대기질 포럼 개최서울시와 중국 베이징(北京)시가
- 2014년 4월 4일 내일, 오전 2: 미국계 회사에서 근무하다 실직 위기에 놓인 이탈리아 엔지니어 419명이 경
- 2009년 총기참사 美軍
- 2014년 4월 4일 내일, 오전 1: 용의자 난사 후 자살·최소 16명 부상...오바마 "철저 조사"지난 2009년 총격
- 칠레지진 쓰나미 일본
- 2014년 4월 4일 내일, 오전 1: 칠레 북부의 태평양 연안에서 발생한 규모 8.2의 강진으로 인한 쓰나미가 3일

XML문서 출력

실제 웹 문서 파싱해보기

루트
요소

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title><![CDATA[Hankyung | 뉴스 | 국제]]></title>
    <link>http://www.hankyung.com/news/intl/ </link>
    <language>ko</language>
    <copyright>Copyright (c) 2005 hankyung.com All rights reserved</copyright>
    <pubDate>2014-04-03 18:40:11</pubDate>
    <lastBuildDate>2014-04-03 18:40:11</lastBuildDate>
    <description>한경닷컴 RSS 서비스</description>
    <image>
      <title>한경닷컴 RSS 서비스</title>
      <url>http://img.hankyung.com/img/bi_top_news02.gif</url>
      <link>http://www.hankyung.com/news/intl/ </link>
    </image>
    <item>
      <title><![CDATA[서울-베이징 '미세먼지 개선 협력' 첫 공동합의]]></title>
      <link><![CDATA[http://www.hankyung.com/news/app/newsview.php?aid=20140403180824]]></link>
      <description><![CDATA[통합위원회 환경팀 신설·대기질 포럼 개최서울시와 베이징시 환경당국...]]></description>
      <author><![CDATA[]]></author>
      <pubDate>2014-04-03 18:08:24</pubDate>
    </item>
    <item>
      <title><![CDATA[伊 엔지니어 419명 이베이에 경매로 올라와]]></title>
      <link><![CDATA[http://www.hankyung.com/news/app/newsview.php?aid=20140403180824]]></link>
    </item>
  </channel>
</rss>
```

필요한 부분

문서의 구조를 파악하고 가져올 부분을 정함

실제 웹 문서 파싱해보기

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
```

이문서의 구조를 간단히 나타내면

루트
요소

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
```

```
  <copyright>All rights reserved</copyright>
```

...

```
<item>
  <title> 헤드라인 </title>
```

```
  <url>
    <img alt="image" data-bbox="725 473 820 516"/>
```

...

```
</item>
```

```
  <title>첫 공동합의]]</title>
  <url>'newsview.php?aid=20140404:
  기질 포럼 개최서울시와 준
```

```
<item> ... </item>
```

```
<item> ... </item>
```

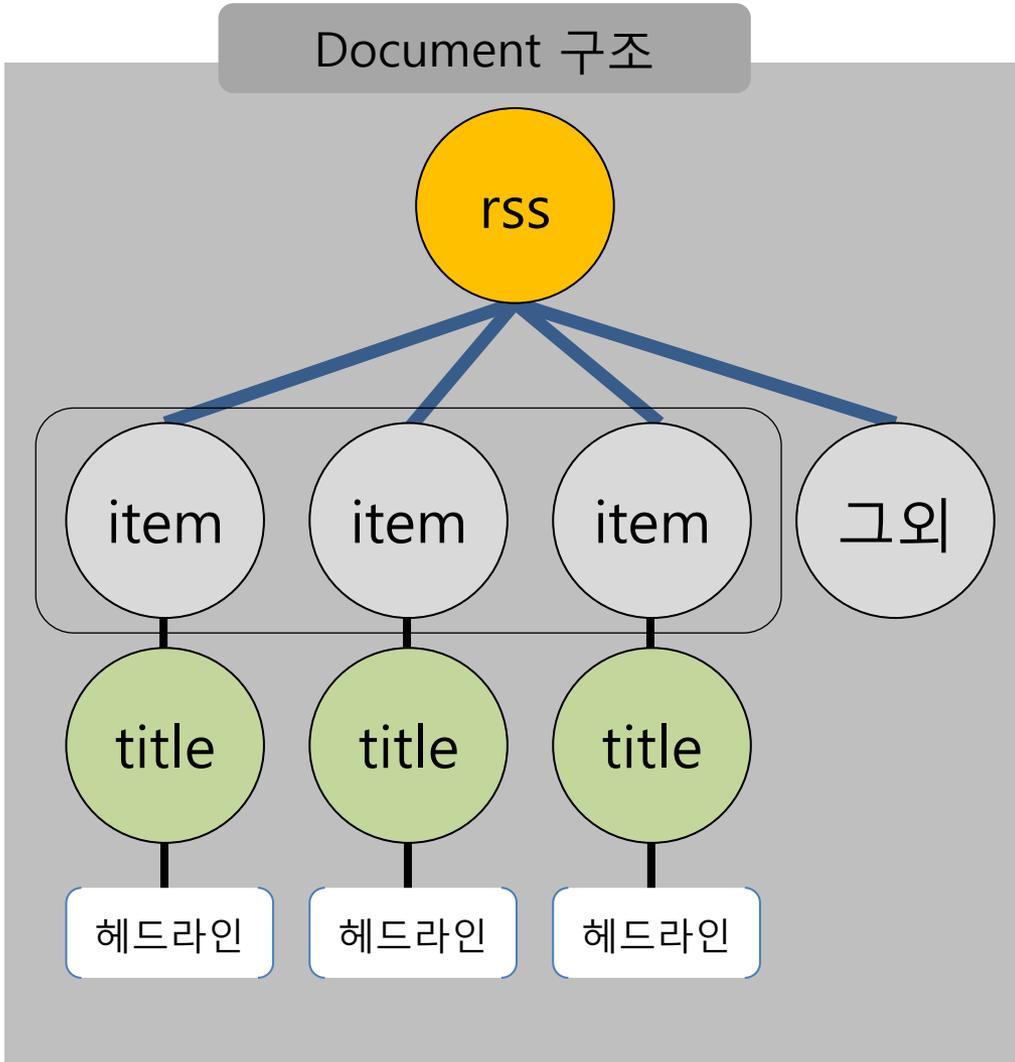
```
  <title>로 올라와]]</title>
```

```
<link><![CDATA[http://www.nankyung.com/news/app/newsview.php?aid=20140404:]]>
```

필요한 부분

접근방법을 생각해보자

실제 웹 문서 파싱해보기



<접근 절차>

1. 루트 요소 rss에 접근
2. item 태그명을 가진 요소 접근
3. item 요소내의 자식요소중 title 요소 접근
4. title 요소의 내용 접근

예제 수행 환경

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linear"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="헤드라인 조회"
    />
    <ListView
        android:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />
</LinearLayout>
```

버튼 하나와 리스트 뷰로 이루어진 리니어 레이아웃

예제 수행 환경

AndroidManifest.xml 파일에 아래코드 추가

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

<주의>

안드로이드 버전이 허니콤(3.0) 이상일 경우

메인쓰레드에선 웹 문서 파싱이 불가능

1. 이클립스에서 작업버전을 낮추거나
2. 별도의 쓰레드를 만들어 파싱 코드 작성

코드 작성

코드 중 엘리먼트 접근부분

루트 요소 접근

```
////////// 엘리먼트 접근
Element root = doc.getDocumentElement(); // 루트 엘리먼트
// item이란 이름을 가진 모든 요소를 찾는다.
NodeList items = root.getElementsByTagName("item");
for(int i=0;i<items.getLength();i++){ // 찾은 갯수만큼 루프
    Node item = items.item(i); // 하나의 item요소 접근
    // item요소의 두번째 자식요소 title 접근(첫번째 자식은 텍스트 노드)
    Node title = item.getFirstChild().getNextSibling();
    Node value = title.getFirstChild(); // title 요소의 내용 접근
    Headline.add(value.getNodeValue()); // 내용을 Arraylist에 삽입
}
```

Item 요소 접근

Title 요소의 내용 접근

실행 결과

 test

헤드라인 조회

 test

헤드라인 조회

마오밍시 화학공장 반대시위...광저우 선전으로 확산

중국 '사정 태풍' 속 고위 공직자 투신자살

日문부상 "자국 영토 올바르게 가르치는 것 당연"

중국이 북한을 쉽게 버려선 안 되는 3대 이유는

교육부, 日초등교과서 '독도 왜곡' 규탄..."비교육적 행위"

[일지] 일본 '역사 왜곡 교과서' 파문史 들여다보니...

아베 정권, 日초등교과서 '독도 점령' 야욕...'과거 미화' 강화

헤드라인 조회 버튼을 누르면 내용 출력

MainActivity 코드(첫 번째 예제)

```
import java.io.InputStream;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import android.os.Bundle;
import android.app.Activity;
import android.content.res.Resources;
import android.view.*;
import android.widget.*;

public class MainActivity extends Activity {
    TextView mResult;

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

        mResult = (TextView)findViewById(R.id.result);
    }

    public void mOnClick(View v){
        try{
            // xml문서를 파싱하기위한 준비
            DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
            DocumentBuilder builder =
            factory.newDocumentBuilder();

            // raw디렉터리 내의 student.xml파일을 읽어와 스트림
            // 에 저장
            Resources res = getResources();
            InputStream istream =
            res.openRawResource(R.raw.student);

            // parse메소드로 스트림을 파싱하여 Document 객체를
            // 얻음
            Document doc = builder.parse(istream);
```

```
Element root = doc.getDocumentElement(); // 루트 엘리먼트
NodeList childs = root.getElementsByTagName("name");
// 루트 엘리먼트에서 name이란 이름을 가진 요소 전체
Node child = childs.item(0); // 첫번째 name 요소

NamedNodeMap Attrs = child.getAttributes(); // name의 속성값
Node attr = Attrs.item(0); // name의 첫번째 속성

Node childvalue = child.getFirstChild(); // 내용도 하나의 자식으로 취급

String rootName = root.getNodeName(); // 루트 명
String childName = child.getNodeName(); // 자식 요소 명
String childValue = childvalue.getNodeValue(); // 자식 요소 값
String AttrName = attr.getNodeName(); // 속성 명
String AttrValue = attr.getNodeValue(); // 속성 값
mResult.setText("루트요소 이름 : " + rootName +
"\n자식요소 이름 : " + childName +
"\n자식요소 내용 : " + childValue +
"\n자식요소 속성명 : " + AttrName +
"\n자식요소 속성값 : " + AttrValue);
}
catch (Exception e){
    mResult.setText("오류");
}
}
```

MainActivity 코드(두 번째 예제)

```
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.app.Activity;
import android.view.*;
import android.widget.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

public class MainActivity extends Activity {
    ArrayList<String> Headline = new ArrayList<String>();
    ArrayAdapter<String> Adapter;
    ListView list;

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btn = (Button)findViewById(R.id.btn);
        list = (ListView)findViewById(R.id.list);

        btn.setOnClickListener(mOnClick);
    }

    View.OnClickListener mOnClick = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            DataLoading thread = new DataLoading();
            thread.start();
        }
    };

    // 데이터 로딩 쓰레드
    class DataLoading extends Thread {
        public void run(){
            try{
                getText();
                mHandler.sendMessage(0);// 데이터를 얻어오면 핸들러에게
                신호전달
            }
            catch(Exception e){
                e.printStackTrace();
            }
        }
    }
}
```

```
Handler mHandler = new Handler(){
    public void handleMessage(Message msg){
        if(msg.what == 0){
            Adapter = new ArrayAdapter<String>(MainActivity.this,
            android.R.layout.simple_list_item_1, Headline);
            list.setAdapter(Adapter);
        }
    }
};

// url을 통해 웹문서를 읽어옴
public void getText(){
    try{
        ////////// xml 문서를 파싱하기위한 준비
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();

        ////////// xml 웹 문서를 Document객체로 만드는 과정
        String url= "http://rss.hankyung.com/new/news_intl.xml"; // 주소
        URL nURL = new URL(url);
        InputStream xml = nURL.openStream();
        Document doc = builder.parse(xml);

        ////////// 엘리먼트 접근
        Element root = doc.getDocumentElement(); // 루트 엘리먼트
        // item이란 이름을 가진 모든 요소를 찾는다.
        NodeList items = root.getElementsByTagName("item");
        for(int i=0;i<items.getLength();i++){ // 찾은 갯수만큼 루프
            Node item = items.item(i); // 하나의 item요소 접근
            // item요소의 두번째 자식요소 title 접근(첫번째 자식은 텍스트 노드)
            Node title = item.getFirstChild().getNextSibling();
            Node value = title.getFirstChild(); // title 요소의 내용 접근
            Headline.add(value.getNodeValue()); // 내용을 ArrayList에 삽입
        }
    }
    catch (MalformedURLException e)
    {
        e.printStackTrace();
    }
    catch(Exception e){
        e.printStackTrace();
    }
}
```

3. SAX Parser

- SAX Parsing의 **장점**

- 메모리를 차지하는 공간이 **적음**

- XML데이터를 모두 메모리에 적재하지 않고 **라인 (시작태그 - 종료태그)단위로 마킹**

- 대용량 XML을 Parsing해도 속도가 **빠름**

```
<?xml version="1.0" encoding="utf-8" ?>
- <car>
1라인 <sedan type="대형" cc="3000cc" price="4000">granduer</sedan>
2라인 <sedan type="중형" cc="2000CC" price="3000">sonata</sedan>
</car>
```

- SAX Parsing의 단점

- 지나간 엘리먼트를 읽어오기 위해서는 처음부터 다시 읽어야 함
- 특정 엘리먼트에 대해서 어떻게 동작할지 사용자가 직접 정의

```
public void startElement(String uri, String localName, String qName, Attributes attributes){
    Log.i("sax", "startElement()");

    if(localName.equals("TITLE")){
        MainActivity.result.append("#n TITLE : ");
    }
    else if(localName.equals("ARTIST")){
        MainActivity.result.append("#n ARTIST : ");
    }
    else if(localName.equals("COUNTRY")){
        MainActivity.result.append("#n COUNTRY : ");
    }
    else if(localName.equals("PRICE")){
        MainActivity.result.append("#n PRICE : ");
    }
    else if(localName.equals("YEAR")){
        MainActivity.result.append("#n YEAR : ");
    }
}
```

```
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
```

SAX API 프로그래밍을 하려면?

- ① 세 개의 패키지를 import
 - [org.xml.sax], [org.xml.sax.helpers], [javax.xml.parsers]
- ② 이벤트 핸들러 클래스를 설계하여 이벤트 핸들러를 등록
- ③ 파서 객체를 생성
- ④ 이벤트 핸들러로 이벤트 객체 생성
- ⑤ 파서 객체에 이벤트 객체를 적용시켜서 XML 문서 파싱

① 이벤트 핸들러 클래스 설계

- DefaultHandler 클래스로부터 상속받아 이벤트 핸들러 클래스를 설계

```
class MyHandler extends DefaultHandler {  
    public void startDocument(){ //문서의 시작  
    }  
    public void startElement(){ //요소의 시작  
    }  
    public void endElement( ... ){ //요소의 끝  
    }  
    public void endDocument( ... ){ //문서의 종료  
    }  
}
```

② 파서 객체 생성

- 추상클래스 SAXParserFactory의 newInstance메소드를 사용하여 생성

```
SAXParserFactory spf = SAXParserFactory.newInstance();
```

```
SAXParser sp = spf.newSAXParser();
```

③ 이벤트 객체 생성

- 앞서 ①에서 설계한 MyHandler 클래스로 객체 생성

```
MyHandler myh = new MyHandler();
```

④ 요소 추출

- 메소드에 대한 이해

```
public void startElement(String nsURI, String name,  
                        String qName, Attributes atts){}
```

//startElement()는 XML태그를 만나면 호출

//name : XML네임스페이스의 접두사를 제외한 태그 이름

//qName : XML네임스페이스의 접두사를 포함한 태그 이름

//atts : 태그에 선언한 속성

```
public void characters(char ch[], int start, int length){}
```

//characters()는 XML의 태그가 아닌 데이터를 알려주는 메소드

//데이터 내용은 ch의 start 인덱스에서 length만큼의 문자에 해당

⑤ 속성의 이름과 값 알아내기

- `startElement()` 메소드는 속성에 대한 정보를 `Attributes`를 통해 알아냄

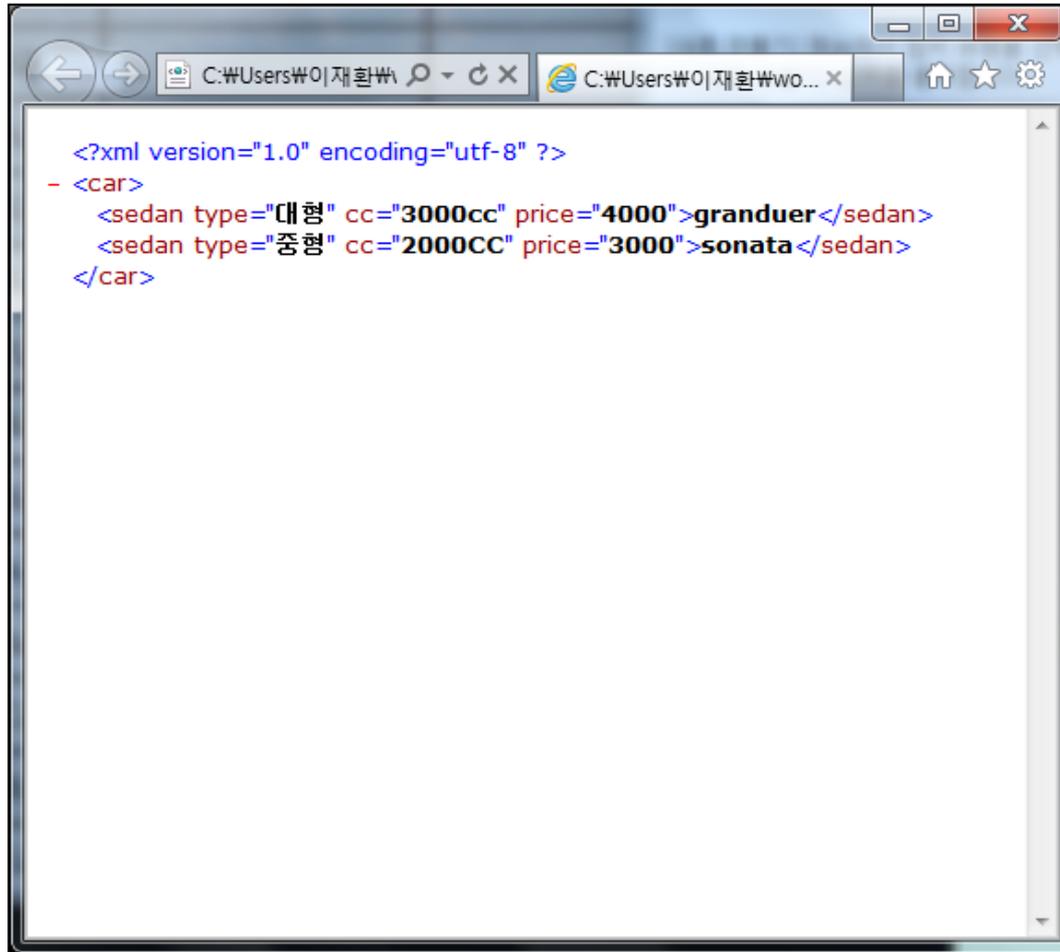
`Attributes`의 `getLength()` 메소드는 **속성의 개수**

`Attributes`의 `getQName()` 메소드는 **속성의 이름**

`Attributes`의 `getValue()` 메소드는 **속성의 값**

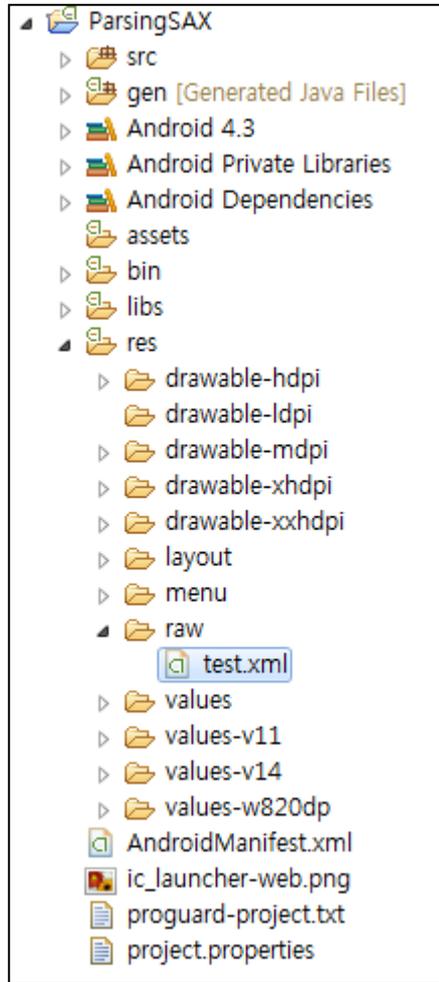
간단한 예제1(내부 XML 파싱)

- 버튼을 누르면 다음의 XML을 파싱하는 프로그램

A screenshot of a web browser window. The address bar shows two tabs with the path 'C:\Users\이재환\wo...'. The main content area displays XML code with syntax highlighting. The code is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
- <car>
  <sedan type="대형" cc="3000cc" price="4000">granduer</sedan>
  <sedan type="중형" cc="2000CC" price="3000">sonata</sedan>
</car>
```

XML생성



- ① 프로젝트 생성 후 [res] 디렉터리에 [raw] 폴더 생성
- ② [raw] 폴더에 [test.xml] 생성

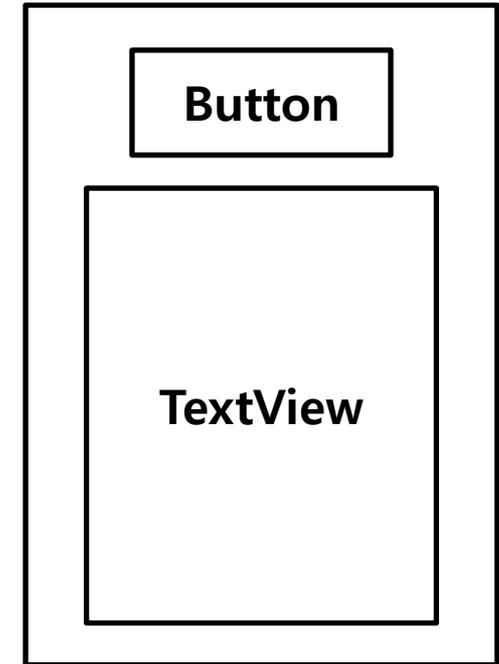
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <car>
3     <sedan type="대형" cc="3000cc" price="4000">
4         granduer
5     </sedan>
6     <sedan type="중형" cc="2000CC" price="3000">
7         sonata
8     </sedan>
9 </car>
10
```

- ③ 다음과 같이 XML데이터 작성

Activity_main.xml

- ① [layout]-[activity_main.xml]에서 Button 생성
- ② 동일한 방법으로 TextView 생성

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical" >
6
7   <Button
8     android:id="@+id/parse"
9     android:layout_width="fill_parent"
10    android:layout_height="wrap_content"
11    android:text="parse" />
12
13   <TextView
14     android:id="@+id/result"
15     android:layout_width="fill_parent"
16     android:layout_height="fill_parent"
17     android:gravity="left/top"
18     android:textSize="10pt" />
19
20 </LinearLayout>
```



SAXHandler.java

- ① [src]-[packge]-[SAXHandler.java] 를 생성하고 DefaultHandler를 상속
- ② 콘텐츠를 저장하는 str 객체, 속성을 저장하는 attr객체, element의 시작과 끝을 알리는 isAppend 변수 선언

```
class SAXHandler extends DefaultHandler {  
    StringBuffer str = new StringBuffer();  
    StringBuffer attr = new StringBuffer();  
    boolean isAppend = false;
```

- ③ startElement()는 시작태그를 만나면 실행되는 메소드고 "sedan"이라는 태그를 만나면 isAppend에 true값을 주어 라인시작을 알리고 속성명과 속성값을 attr에 저장
(Log.i()는 이클립스의 LogCat에 로그정보를 보기 위해사용, 생략가능)

```
public void startElement(String uri, String localName, String qName, Attributes attributes){  
    if(qName.equals("sedan")){  
        isAppend = true;  
    }  
    for(int index = 0; index < attributes.getLength(); index++){  
        attr.append(attributes.getQName(index) + " : " + attributes.getValue(index) + "#n");  
    }  
    Log.i("sax", "startElement()");  
}
```

SAXHandler.java

- ④ 콘텐츠를 저장하고 콘텐츠와 속성을 출력하는 characters()메소드

```
public void characters(char[] ch, int start, int length)
    throws SAXException{
    String temp = new String(ch, start, length);
    if(isAppend==true&&temp.trim().length()!=0){
        str.append(temp);
        MainActivity.result.append("\n"+str+"\n-----\n");
        MainActivity.result.append(attr);
        str.delete(0, str.length());
        attr.delete(0, attr.length());
    }
}
```

- ⑤ endElement()는 종료태그를 만나면 실행되는 메소드고 엘리먼트 isAppend를 false값을 줌으로써 라인의 끝을 알림

```
public void endElement(String uri, String localName, String qName)
    throws SAXException {
    isAppend = false;
    Log.ℳ("sax", "endElement()");
}
```

SAXHandler.java

- ⑤ startDocument(), endDocument() XML문서의 시작과 끝을 알리는 메소드이고 특별한 오버라이딩은 하지 않고 호출되는지 확인하기 위해 Log.i()만 사용

```
public void startDocument() throws SAXException{
    Log.i("sax", "startDocument()");
}
public void endDocument() throws SAXException{
    Log.i("sax", "endDocument()");
}
```

MainActivity.java

- ① [src]-[packge]-[MainActivity.java] 에서 Activity를 상속받은 MainActivity에 사용할 TextView와 Button객체 생성

```
public class MainActivity extends Activity {  
    static TextView result;  
    Button parse;
```

- ② onCreate()메소드에 사용될 TextView와 Button ID를 불러옴

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    result = (TextView) findViewById(R.id.result);  
    findViewById(R.id.parse).setOnClickListener(new OnClickListener(){
```

MainActivity.java

- ③ Button parse는 버튼이 눌리면 파스를 해야하므로 이벤트 추가
- ④ 앞서 만들어놓은 [raw]-[test.xml]을 가져오기 위한 Resource클래스, XML을 입력하기위한 InputStream클래스 선언

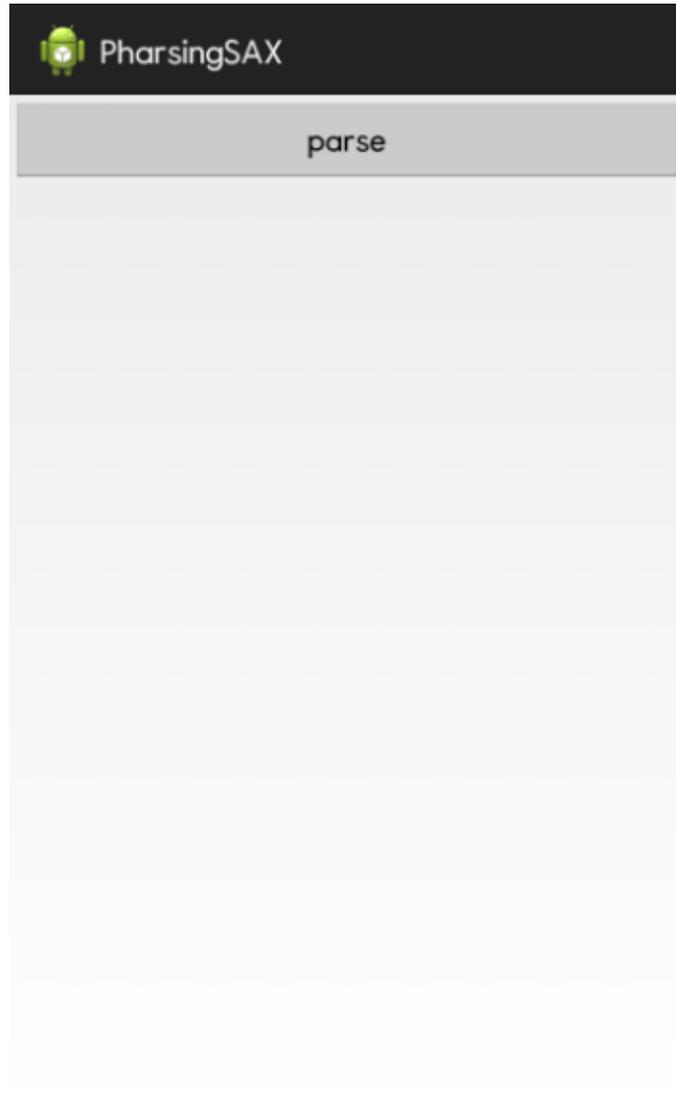
```
@Override
public void onClick(View v) {
    result.setText("");
    Resources res = getResources();
    InputStream is = res.openRawResource(R.raw.test);
```

- ⑤ 파서 객체를 선언하고 앞서 생성한 이벤트 핸들러로 파싱

```
try{
    SAXParserFactory factory = SAXParserFactory.newInstance();
    SAXParser parser = factory.newSAXParser();
    SAXHandler handler = new SAXHandler();
    parser.parse(is, handler);
}catch (Exception e){
    e.printStackTrace();
}
```

SAX Parser

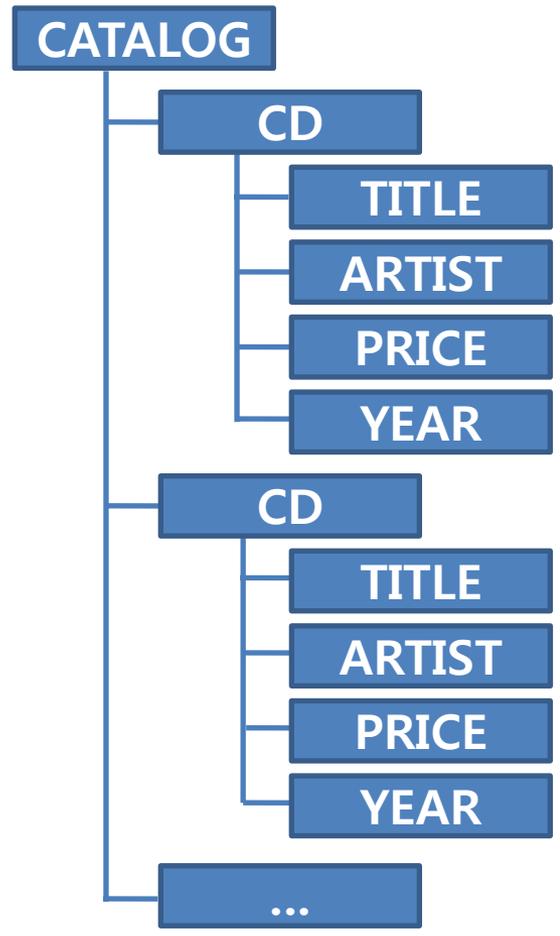
실행결과



간단한 예제2(외부 XML 파싱)

- 버튼을 누르면 다음의 XML을 파싱하는 프로그램 (http://www.xmlfiles.com/examples/cd_catalog.xml)

```
<?xml version="1.0" encoding="ISO8859-1" ?>
- <CATALOG>
- <CD>
  <TITLE>Empire Burlesque</TITLE>
  <ARTIST>Bob Dylan</ARTIST>
  <COUNTRY>USA</COUNTRY>
  <COMPANY>Columbia</COMPANY>
  <PRICE>10.90</PRICE>
  <YEAR>1985</YEAR>
</CD>
- <CD>
  <TITLE>Hide your heart</TITLE>
  <ARTIST>Bonnie Tylor</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>CBS Records</COMPANY>
  <PRICE>9.90</PRICE>
  <YEAR>1988</YEAR>
</CD>
- <CD>
  <TITLE>Greatest Hits</TITLE>
  <ARTIST>Dolly Parton</ARTIST>
  <COUNTRY>USA</COUNTRY>
  <COMPANY>RCA</COMPANY>
  <PRICE>9.90</PRICE>
  <YEAR>1982</YEAR>
</CD>
- <CD>
  <TITLE>Still got the blues</TITLE>
  <ARTIST>Gary More</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>Virgin redords</COMPANY>
  <PRICE>10.20</PRICE>
  <YEAR>1990</YEAR>
</CD>
- <CD>
  <TITLE>Eros</TITLE>
  <ARTIST>Eros Ramazzotti</ARTIST>
  <COUNTRY>EU</COUNTRY>
```



AndroidManifest.xml

- ① 컴퓨터내에 존재하는 XML과 달리 웹페이지에 존재하는 XML을 파싱하기 위해서 퍼미션 설정
- ② AndroidManifest.xml 코드상에 다음 구문을 추가

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

또는 [permissions]탭에서 설정 가능

SAX Parser

Project Explorer showing the file structure of the SAXParser project. The 'AndroidManifest.xml' file is highlighted with a red box and a circled '1'.

Android Manifest Permissions dialog. The 'Permissions' list contains 'Uses Permission'. The 'Add...' button is circled with a '3', and the 'Name' dropdown menu is circled with a '4' and set to 'android.permission.INTERNET'. The 'Permissions' tab is highlighted with a red box and a circled '2'.

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

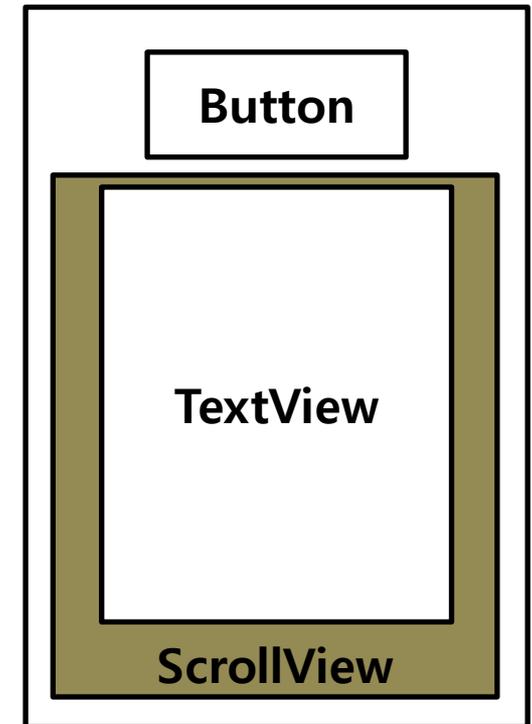
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    ...
/>
```

XML editor showing the generated code for the AndroidManifest.xml file. The `<uses-permission android:name="android.permission.INTERNET"/>` tag is highlighted with a red box. The 'AndroidManifest.xml' tab is highlighted with a red box and a circled '5'.

Activity_main.xml

- ① XML데이터가 많으므로 TextView에서 내용이 잘리지 않도록 ScrollView사용

```
<?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"
    android:orientation="vertical" >
    <Button
        android:id="@+id/parse"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="parse" />
    <ScrollView
        android:id="@+id/mScrollView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >
        <TextView
            android:id="@+id/result"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="left/top"
            android:textSize="10pt" />
    </ScrollView>
</LinearLayout>
```



SAXHandler.java

- ① [src]-[packge]-[SAXHandler.java] 를 생성하고 DefaultHandler를 상속 받아 SAXHandler의 메소드를 오버라이딩
- ② 콘텐츠를 저장하는 str 객체, 총 엘리먼트의 개수를 알리는 변수 count 선언
- ③ 콘텐츠를 저장하고 콘텐츠와 속성을 출력하는 characters()메소드

```
class SAXHandler extends DefaultHandler {  
    StringBuffer str = new StringBuffer();  
    int count = 0;  
    public void characters(char[] ch, int start, int length)  
        throws SAXException{  
        String temp = new String(ch, start, length);  
        if(temp.trim().length() != 0){  
            str.append(temp);  
            MainActivity.result.append(str);  
            str.delete(0, str.length());  
        }  
    }  
}
```

SAXHandler.java

- ④ startElement()는 시작태그를 만나면 실행되는 메소드고 각 태그의 콘텐츠앞에 태그명을 붙여주기 위해 조건문을 사용

```
public void startElement(String uri, String localName, String qName, Attributes attributes){
    Log.i("sax", "startElement()");

    if(localName.equals("TITLE")){
        MainActivity.result.append("#n TITLE : ");
    }
    else if(localName.equals("ARTIST")){
        MainActivity.result.append("#n ARTIST : ");
    }
    else if(localName.equals("COUNTRY")){
        MainActivity.result.append("#n COUNTRY : ");
    }
    else if(localName.equals("PRICE")){
        MainActivity.result.append("#n PRICE : ");
    }
    else if(localName.equals("YEAR")){
        MainActivity.result.append("#n YEAR : ");
    }
}
```

SAXHandler.java

- ⑤ endElement()는 종료태그를 만나면 실행되는 메소드이고 CD태그를 이용하여 구분선을 넣어주고 CD개수를 카운트하는 변수 1증가

```
public void endElement(String uri, String localName, String qName)
    throws SAXException {
    if(localName.equals("CD")){
        count++;
        MainActivity.result.append("#n-----#n");
    }
}
```

- ⑥ startDocument(), endDocument() XML문서의 시작과 끝을 알리는 메소드이고 XML문서 끝을 만나면 총 CD개수를 출력

```
public void startDocument() throws SAXException{
}
public void endDocument() throws SAXException{
    MainActivity.result.append("TOTAL : " + count + "#n-----#n");
}
```

MainActivity.java

- ① [src]-[packge]-[MainActivity.java] 에서 Activity를 상속받은 MainActivity에 사용할 ScrollView, TextView와 Button객체 생성

```
public class MainActivity extends Activity {  
    static TextView result;  
    Button parse;  
    ScrollView mScrollView;
```

- ② onCreate()메소드에 사용될 ScrollView, TextView, Button ID를 불러옴

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mScrollView = (ScrollView) findViewById(R.id.mScrollView);  
    result = (TextView) findViewById(R.id.result);  
    findViewById(R.id.parse).setOnClickListener(new OnClickListener(){
```

MainActivity.java

- ③ Button parse는 버튼이 눌리면 파스를 해야하므로 이벤트 추가
- ④ XML을 읽어오는 XMLReader 객체를 생성하고 URL객체를 통해 XML이 있는 URL를 받아 파싱

```
@Override
public void onClick(View v) {
    result.setText("");
    try{
        SAXParserFactory factory = SAXParserFactory.newInstance();
        SAXParser parser = factory.newSAXParser();
        SAXHandler handler = new SAXHandler();
        XMLReader xmlR = parser.getXMLReader();
        URL url = new URL("http://www.xmlfiles.com/examples/cd_catalog.xml");
        xmlR.setContentHandler(handler);
        xmlR.parse(new InputSource(url.openStream()));
    }catch (Exception e){
        e.printStackTrace();
    }
}
```

android.os.NetworkonMainThreadException

원인

- 안드로이드 3.0 버전 이후부터 발생하는 오류
 - ✓ MainThread에서는 UI작업만 담당하는 역할
 - ✓ ANR(Application Not Responding)을 방지

해결책

- 강제로 사용이 가능하게 함
 - ✓ MainActivity.java에 onCreate()내부 setContentView()아래에 다음의 코드를 추가

```
StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().  
    detectDiskReads().detectDiskWrites().detectNetwork().penaltyLog().build());
```

- ✓ AndroidManifest.xml의 버전을 9로 변경

```
<uses-sdk  
    android:minSdkVersion="9"
```

SAX Parser

실행결과

```
PharsingSAX
parse
```

```
PharsingSAX
parse

Title : Empire Burlesque
ARTIST : Bob Dylan
COUNTRY : USAColumbia
PRICE : 10.90
YEAR : 1985
-----

Title : Hide your heart
ARTIST : Bonnie Tylor
COUNTRY : UKCBS Records
PRICE : 9.90
YEAR : 1988
-----

Title : Greatest Hits
ARTIST : Dolly Parton
COUNTRY : USARCA
PRICE : 9.90
YEAR : 1982
-----
```

```
PharsingSAX
parse

COUNTRY : ÉÚÉlektra
PRICE : 7.20
YEAR : 1985
-----

TITLE : Red
ARTIST : The Communards
COUNTRY : UKLondon
PRICE : 7.80
YEAR : 1987
-----

TITLE : Unchain my heart
ARTIST : Joe Cocker
COUNTRY : USAEMI
PRICE : 8.20
YEAR : 1987
-----

TOTAL :26
-----
```

예제1 소스코드

MainActivity.java

```
package com.example.pharsingsax;

import java.io.InputStream;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import android.app.Activity;
import android.content.res.Resources;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity {
    static TextView result;
    Button parse;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        result = (TextView) findViewById(R.id.result);
        findViewById(R.id.parse).setOnClickListener(new OnClickListener(){
            @Override
            public void onClick(View v) {
                result.setText("");
                Resources res = getResources();
                InputStream is = res.openRawResource(R.raw.test);
                try{
                    SAXParserFactory factory = SAXParserFactory.newInstance();
                    SAXParser parser = factory.newSAXParser();
                    SAXHandler handler = new SAXHandler();
                    parser.parse(is, handler);
                }catch (Exception e){
                    e.printStackTrace();
                }
            }
        });
    }
}
```

SAXHandler.java

```
package com.example.pharsingsax;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

import android.util.Log;

class SAXHandler extends DefaultHandler {
    StringBuffer str = new StringBuffer();
    StringBuffer attr = new StringBuffer();
    boolean isAppend = false;
    public void characters(char[] ch, int start, int length) throws SAXException{
        String temp = new String(ch, start, length);
        if(isAppend==true&&temp.trim().length()!=0){
            str.append(temp);
            MainActivity.result.append("\n"+str+"\n-----\n");
            MainActivity.result.append(attr);
            str.delete(0, str.length());
            attr.delete(0, attr.length());
        }
    }
    public void startElement(String uri, String localName, String qName,
        Attributes attributes){
        if(qName.equals("sedan")){
            isAppend = true;
        }
        for(int index = 0; index < attributes.getLength(); index++){
            attr.append(attributes.getQName(index)+ " : " + attributes.getValue(index)+"\n");
        }
        Log.i("sax", "startElement()");
    }
    public void endElement(String uri, String localName, String qName)
        throws SAXException {
        isAppend = false;
        Log.i("sax", "endElement()");
    }
    public void startDocument() throws SAXException{
        Log.i("sax", "startDocument()");
    }
    public void endDocument() throws SAXException{
        Log.i("sax", "endDocument()");
    }
}
```

예제1 소스코드

Activity_main.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"
    android:orientation="vertical" >

    <Button
        android:id="@+id/parse"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="parse" />

    <TextView
        android:id="@+id/result"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="left/top"
        android:textSize="10pt" />

</LinearLayout>
```

test.xml

```
<?xml version="1.0" encoding="utf-8"?>
<car>
    <sedan type="대형" cc="3000CC" price="4000">
        Granduer
    </sedan>
    <sedan type="중형" cc="2000CC" price="3000">
        Sonata
    </sedan>
</car>
```

예제2 소스코드

MainActivity.java

```
package com.example.pharsingsax;

import java.net.URL;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.InputSource;
import org.xml.sax.XMLReader;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ScrollView;
import android.widget.TextView;

public class MainActivity extends Activity {
    static TextView result;
    Button parse;
    ScrollView mScrollView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().
            detectDiskReads().detectDiskWrites().detectNetwork().penaltyLog().build());
        mScrollView =(ScrollView) findViewById(R.id.mScrollView);
        result = (TextView) findViewById(R.id.result);
        findViewById(R.id.parse).setOnClickListener(new OnClickListener(){
```

```
    @Override
    public void onClick(View v) {
        result.setText("");
        try{
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser parser = factory.newSAXParser();
            SAXHandler handler = new SAXHandler();
            XMLReader xmlR = parser.getXMLReader();
            URL url = new URL("http://www.xmlfiles.com/examples/cd_catalog.xml");
            xmlR.setContentHandler(handler);
            xmlR.parse(new InputSource(url.openStream()));
        }catch (Exception e){
            e.printStackTrace();
        }
    }
    });
}
```

예제2 소스코드

SAXHandler.java

```
package com.example.pharsingsax;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

import android.util.Log;

class SAXHandler extends DefaultHandler {
    StringBuffer str = new StringBuffer();
    int count = 0;
    public void characters(char[] ch, int start, int length) throws SAXException{
        String temp = new String(ch, start, length);
        if(temp.trim().length() != 0){
            str.append(temp);
            MainActivity.result.append(str);
            str.delete(0, str.length());
        }
    }
    public void startElement(String uri, String localName, String qName,
        Attributes attributes){
        Log.i("sax", "startElement()");
        if(localName.equals("TITLE")){
            MainActivity.result.append("\n TITLE : ");
        }
        else if(localName.equals("ARTIST")){
            MainActivity.result.append("\n ARTIST : ");
        }
        else if(localName.equals("COUNTRY")){
            MainActivity.result.append("\n COUNTRY : ");
        }
        else if(localName.equals("PRICE")){
            MainActivity.result.append("\n PRICE : ");
        }
        else if(localName.equals("YEAR")){
            MainActivity.result.append("\n YEAR : ");
        }
    }
}
```

```
public void endElement(String uri, String localName, String qName)
    throws SAXException {
    if(localName.equals("CD")){
        count++;
        MainActivity.result.append("\n-----\n");
    }
}
public void startDocument() throws SAXException{
}
public void endDocument() throws SAXException{
    MainActivity.result.append("TOTAL : " + count + "\n-----
    --\n");
}
}
```

예제2 소스코드

Activity_main.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"
    android:orientation="vertical" >
    <Button
        android:id="@+id/parse"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="parse" />
    <ScrollView
        android:id="@+id/mScrollView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >
        <TextView
            android:id="@+id/result"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="left|top"
            android:textSize="10pt" />
    </ScrollView>
</LinearLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.pharsingsax"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="18" />
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.pharsingsax.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>
```

4. XML Pull Parser

XML Pull Parser

● XML Pull Parser 특징

1. 전체 동작 방식은 SAX와 비슷
2. SAX와 달리 이벤트 핸들러를 작성하지 않아도 되므로 간편
3. 문서의 일부분만 작업해야 할 경우엔 SAX보다 빠르고 단순하게 처리 가능

```
while(EventType != XmlPullParser.END_DOCUMENT){
    switch(EventType){
        case XmlPullParser.START_TAG:
            tag = parser.getName();
            if(tag.equals("student")){
                bool=true;
            }
            break;
        case XmlPullParser.TEXT:
            if(bool==true){
                result.append(parser.getText());
                bool=false;
            }
            break;
    }
    EventType = parser.next();
}
```

Pull Parser 코드작성방법

1. 빌더 생성후 파서 객체 생성

`newInstance()`를 통해 빌더팩토리 생성 후 `newPullParser()`메소드를 통해 파서 객체를 얻는다.

```
XmlPullParserFactory factory = XmlPullParserFactory.newInstance();  
XmlPullParser parser = factory.newPullParser();
```

2. 분석하고 싶은 xml파일을 설정

`XmlPullParser` 클래스의 `setInput(InputStream, String)`을 통해 분석할 xml과 인코딩방식을 설정

```
parser.setInput(is, null);
```

3. 문서를 순차적으로 읽으면서 이벤트를 진행

현재 이벤트를 분석하고 `next()` 메소드로 다음 이벤트로 이동시킨다.

```
while(EventType != XmlPullParser.END_DOCUMENT){
    switch(EventType){
        case XmlPullParser.START_DOCUMENT:
            break;
        case XmlPullParser.START_TAG:
            break;
        case XmlPullParser.TEXT:
            break;
        case XmlPullParser.END_TAG:
            break;
    }
    EventType = parser.next();
}
```

XMLPullParser 분석 진행방법(속성)

속성	의미
START_DOCUMENT	XML파일의 시작에 도달했을 때 반환
START_TAG	요소의 시작 태그를 만났을 때 반환
TEXT	요소의 텍스트를 만났을 때 반환 (<code><tag>TEXT</tag></code> 에서 TEXT)
END_TAG	요소의 종료 태그를 만났을 때 반환(<code></tag></code>)
END_DOCUMENT	XML파일의 끝에 도달했을 때 반환

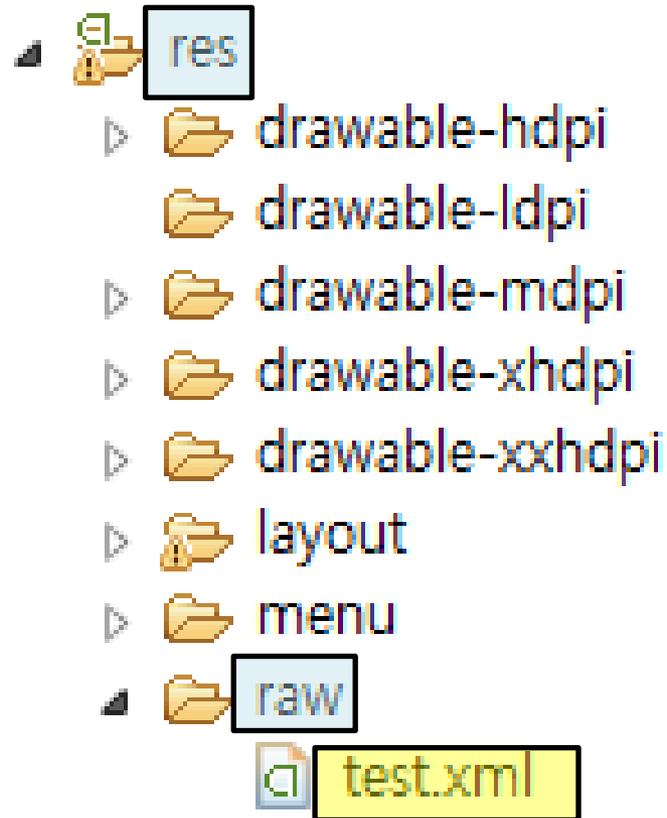
`getEventType()`으로 현재 속성을 가져오고 `next()`로 다음 속성으로 이동

XmlPullParser 형식을

이용한 간단한 예제

예제 1

이클립스에서 안드로이드 프로젝트 생성 후



res – raw 디렉터리에 test.xml 생성

예제 1

test.xml 파일에 다음과 같이 입력

```
test.xml ☒
1 <?xml version="1.0" encoding="UTF-8"?>
2 <mobile>
3     <student 학번="20080945" 학년="4학년" >
4         이성훈
5     </student>
6     <student 학번="20111519" 학년="4학년" >
7         김기순
8     </student>
9 </mobile>
10
11
```

예제 1

LinearLayout 내부에 Button과 TextView를 추가

```
1 <LinearLayout xmlns:android="http://schemas.a
2   android:orientation="vertical"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent">
5   <Button
6     android:id="@+id/parse"
7     android:layout_width="fill_parent"
8     android:layout_height="wrap_content"
9     android:text="버튼클릭"/>
10  <TextView
11    android:id="@+id/result"
12    android:layout_width="fill_parent"
13    android:layout_height="fill_parent"
14    android:gravity="left/top"
15    android:textSize="10pt" />
16 </LinearLayout>
17
```



예제 1

MainActivity 클래스의 상속을 Activity로 고치고 TextView와 Button 객체를 생성 후 onCreate() 메소드에 사용될 TextView와 Button을 불러옴

```
public class MainActivity extends Activity {
    TextView result;
    Button parse;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_main);
        result= (TextView)findViewById(R.id.result);
        findViewById(R.id.parse).setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
```

예제 1

버튼이 클릭되었을때 이벤트를 발생시키는 **onClick()** 선언
Resources클래스와 **InputStream**클래스를 통해 xml파일을 읽어옴

```
public void onClick(View v) {  
    result.setText("");  
    Resources res = getResources();  
    InputStream is = res.openRawResource(R.raw.test);
```

PullPaser인스턴스를 생성하여 파서 객체를 얻고 **setInput()**메소드를 이용하여 xml을 저장

```
try {  
    XmlPullParserFactory factory = XmlPullParserFactory.newInstance()  
    XmlPullParser parser = factory.newPullParser();  
    parser.setInput(is, null);
```

예제 1

- ① 태그가 맞게 들어왔는지 체크하기 위한 boolean데이터타입을 선언하고 tag를 저장하기 위한 tag변수와 문자열을 저장하고 출력해 주기 위해 StringBuffer클래스를 사용합니다.
- ② 이벤트타입을 얻기위해 getEventType메소드로 저장을 하고 SAX의 이벤트핸들러 대신 루프를 써서 end태그가 아니면 루프를 돌면서 파싱을 하게 한다.

```
boolean bool = false;
String tag;
StringBuffer attr = new StringBuffer();
int EventType = parser.getEventType();
while(EventType != XmlPullParser.END_DOCUMENT){
    switch(EventType){
```

예제 1

START_TAG를 만나면 getName()메소드를 이용하여 현재 태그의 이름을 저장하고 그 태그가 student을 경우 태그가 들어왔으므로 true로 설정하고StringBuffer의 변수 attr.append()를 이용해 속성의이름과 속성의 내용을 저장한다.

```
while(EventType != XmlPullParser.END_DOCUMENT){
    switch(EventType){
        case XmlPullParser.START_TAG:
            tag = parser.getName();
            if(tag.equals("student")){
                bool=true;
                int index;
                for(index=0;index<parser.getAttributeCount();index++){
                    attr.append(parser.getAttributeName(index)
                                + " : "
                                +parser.getAttributeValue(index)
                                +"\n");
                }
            }
            break;
    }
}
```

예제 1

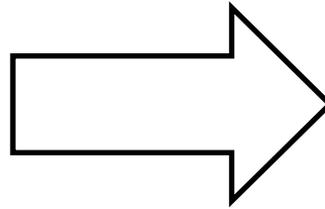
TEXT태그를 만났을경우 태그의 내용과 StringBuffer로 추가시킨 문자열을 TextView에 출력시켜 줍니다. 그리고 next()메소드를 통해 다음 태그로 이벤트를 이동시킨다.

```
        case XmlPullParser.TEXT:
            if(bool==true){
                result.append(parser.getText());
                result.append("\n-----\n");
                result.append(attr);
                attr.delete(0, attr.length());
                bool=false;
            }
            break;
        } //end of switch
        EventType = parser.next();
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

예제 1 실행결과



버튼클릭



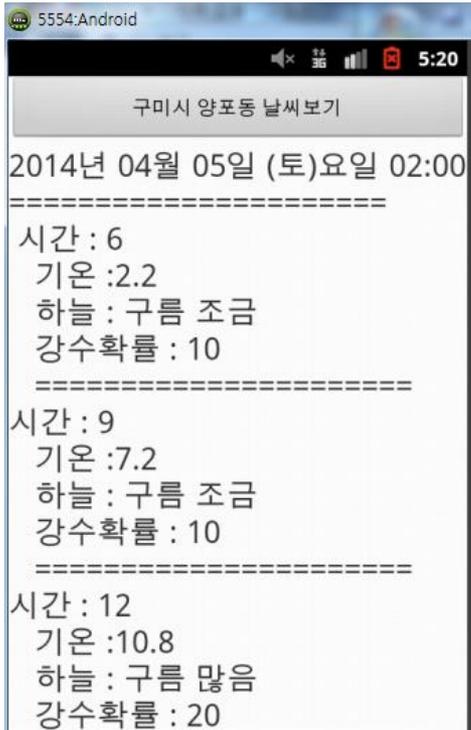
XmlPullParser를 이용한

간단한 웹문서 파싱

예제 2

기상청 RSS

http://www.kma.go.kr/weather/lifenindustry/sevice_rss.jsp



웹 RSS 날씨위젯

RSS란?
RSS(Really Simple Syndication, Rich Site Summary)란 블로그처럼 콘텐츠 업데이트가 자주 일어나는 웹사이트에서, 업데이트된 정보를 손쉽게 가져올 수 있도록 도와주는 형태로 만들어진 데이터 형식입니다. RSS서비스를 이용하면 업데이트된 정보를 찾아볼 수 있고, 업데이트 될 때마다 빠르고 편리하게 확인할 수 있습니다.

RSS 서비스

복사된 RSS주소를 RSS리더기에 추가
RSS리더기를 통해 실시간으로 정보를 확인

한RSS 다운받기 구글RSS 다운받기

동네예보 > 시간별예보

동네예보 경상북도 검색 구미시 검색 양포동 검색 RSS

중기예보

중기 예보	전국	RSS	전라북도	RSS
	서울-경기도	RSS	전라남도	RSS
	강원도	RSS	경상북도	RSS
	충청북도	RSS	경상남도	RSS
	충청남도	RSS	제주특별자치도	RSS

동네예보 RSS정의 중기예보 RSS정의

파싱할 동네주소
선택 후 RSS
버튼클릭

예제 2

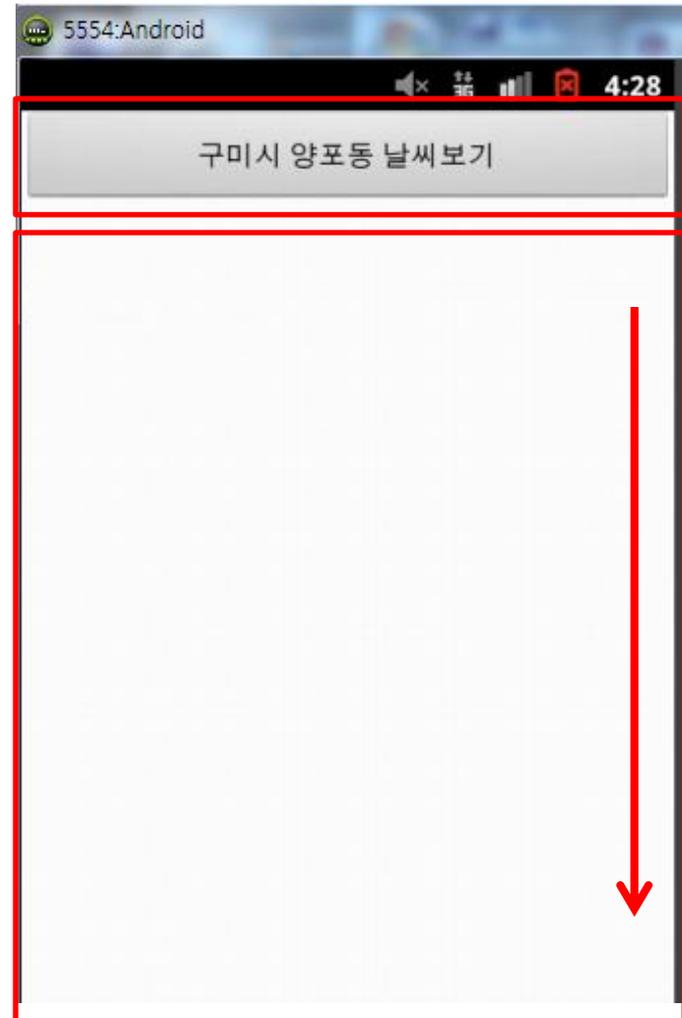
```
www.kma.go.kr/wid/queryDFSRSS.jsp?zone=4719069000
<rss version="2.0">
  <channel>
    <title>기상청 동네예보 웹서비스 - 경상북도 구미시 양포동 도표예보</title>
    <link>http://www.kma.go.kr/weather/main.jsp</link>
    <description>동네예보 웹서비스</description>
    <language>ko</language>
    <generator>동네예보</generator>
    <pubDate>2014년 04월 04일 (금)요일 23:00</pubDate>
  </channel>
  <item>
    <author>기상청</author>
    <category>경상북도 구미시 양포동</category>
    <title>동네예보(도표) : 경상북도 구미시 양포동 [X=86,Y=96]</title>
    <link>
      http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST&dongCode=4719069000
    </link>
    <guid>
      http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST&dongCode=4719069000
    </guid>
    <description>
      <header>
        <tm>201404042300</tm>
        <ts>7</ts>
        <x>86</x>
        <y>96</y>
      </header>
      <body>
        <data seq="0">
          <hour>3</hour>
          <day>1</day>
          <temp>4.9</temp>
          <tmx>14.5</tmx>
          <tmn>2.9</tmn>
          <sky>3</sky>
          <pty>0</pty>
          <wfKor>구름 많음</wfKor>
          <wfEn>Mostly Cloudy</wfEn>
          <pop>20</pop>
          <r12>0.0</r12>
          <s12>0.0</s12>
          <ws>2.9000000000000004</ws>
        </data>
      </body>
    </description>
  </item>
</rss>
```

필요한 태그만 지정하여 파싱

예제 2

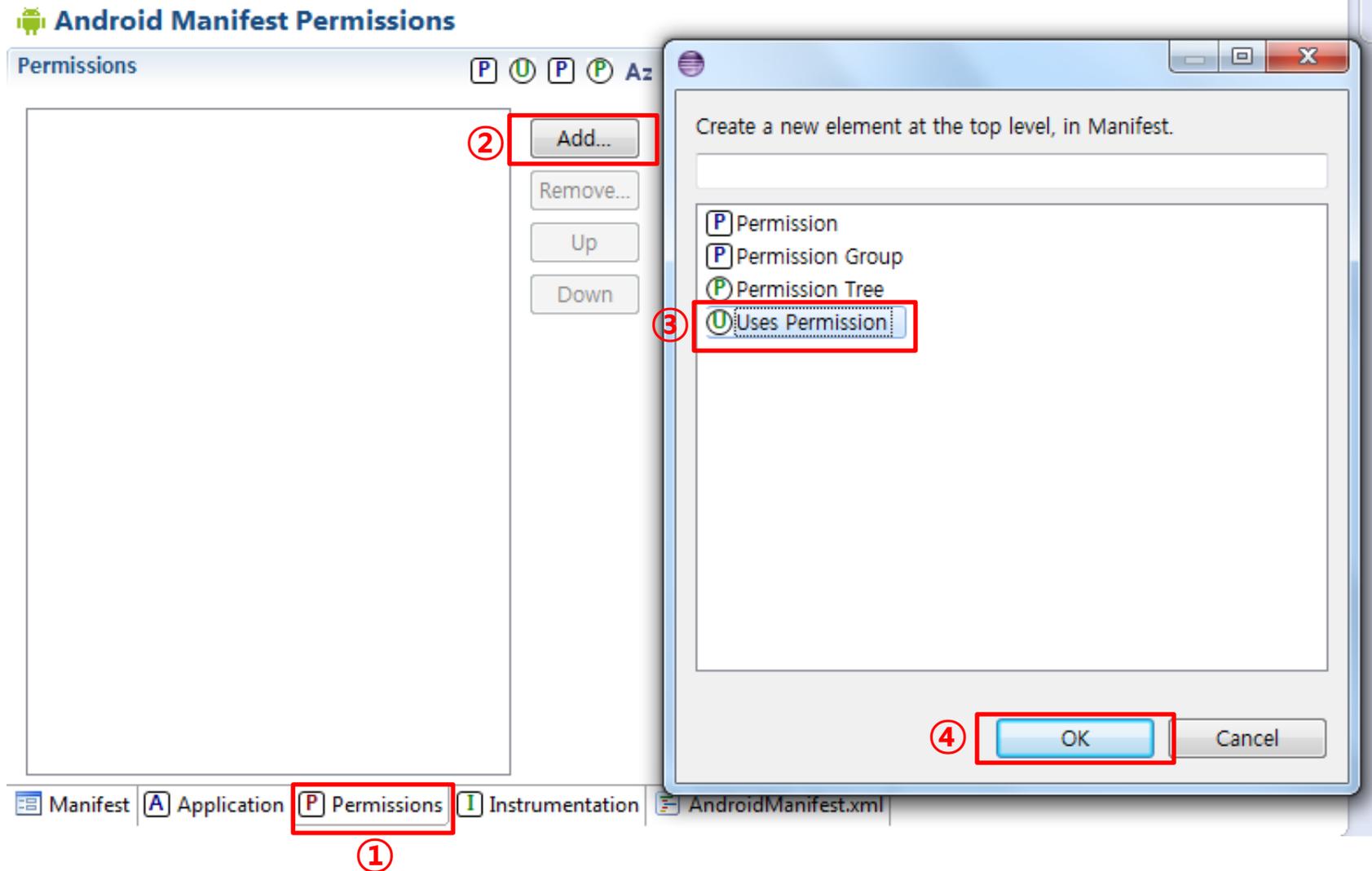
Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:id="@+id/parse"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="구미시 양포동 날씨보기" />
    <ScrollView
        android:id="@+id/mScrollView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TextView
            android:id="@+id/result"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:gravity="left|top"
            android:textSize="10pt" />
    </ScrollView>
</LinearLayout>
```



예제 2

AndroidManifest.xml



AndroidManifest.xml

Android Manifest Permissions

Permissions P U P P Az

U Uses Permission

Add...

Remove...

Up

Down

Attributes for Uses Permission

The tag requests a {@link #AndroidManifestPermission <permission>} that the containing package must be granted in order for it to operate correctly.

Name

- android.permission.GET_ACCOUNTS
- android.permission.GET_PACKAGE_SIZE
- android.permission.GET_TASKS
- android.permission.GLOBAL_SEARCH
- android.permission.HARDWARE_TEST
- android.permission.INJECT_EVENTS
- android.permission.INSTALL_LOCATION_PROVIDER
- android.permission.INSTALL_PACKAGES
- android.permission.INTERNAL_SYSTEM_WINDOW
- 5 android.permission.INTERNET**
- android.permission.KILL_BACKGROUND_PROCESSES
- android.permission.MANAGE_ACCOUNTS
- android.permission.MANAGE_APP_TOKENS
- android.permission.MASTER_CLEAR
- android.permission.MODIFY_AUDIO_SETTINGS
- android.permission.MODIFY_PHONE_STATE

예제 2

MainActivity클래스에서 TextView, Button의 객체와 내용이 잘리지 않기 위한 ScrollView객체를 선언하고 String형으로 외부 xml사이트를 저장하고 URL클래스를 선언 후 onCreate()메소드에서 사용될 각각의 객체들을 불러온다.

```
public class MainActivity extends Activity {
    TextView result;
    Button parse;
    ScrollView mScrollView;
    String uri = "http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=4719069000.xml";
    URL newsURL;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_main);
        mScrollView = (ScrollView)findViewById(R.id.mScrollView);
        result = (TextView) findViewById(R.id.result);
        findViewById(R.id.parse).setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
```

예제 2

- ① onClick()은 버튼이 클릭되었을 때 이벤트를 발생시키는 메소드으 로 그 내부에는 URL클래스를 객체 선언하여 uri에 저장된 사이트를 접속하여 xml데이터를 읽어 inputStream에 저장시킨다.
- ② PullParser인스턴스를 생성하여 파서 객체를 얻고 setInput()메소드를 이용하여 xml을 저장한다.

```
public void onClick(View v) {
    result.setText("");
    try {
        newsURL = new URL(uri);
        InputStream in = newsURL.openStream();
        XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
        XmlPullParser parser = factory.newPullParser();
        parser.setInput(in, "utf-8");
        String tag = "";
        int EventType = parser.getEventType();
        while (EventType != XmlPullParser.END_DOCUMENT) {
            switch(EventType){
```

예제 2

루프문을 이용하여 EventType이 END_DOCUMENT를 만날때 까지 계속 실행시킨다.

START_TAG를 만나면 getName()을 이용하여 현재 태그명을 저장하고 해당 태그명이 일치할경우 TextView에 해당 출력물을 출력 시킨다.

```
while (EventType != XmlPullParser.END_DOCUMENT) {  
    switch(EventType){  
        case XmlPullParser.START_TAG:  
            tag = parser.getName();  
            if(tag.equals("hour")){  
                result.append("시간 : ");  
            }else if(tag.equals("temp")){  
                result.append("기온 :");  
            }else if(tag.equals("wfKor")){  
                result.append("하늘 : ");  
            }else if(tag.equals("pop")){  
                result.append("강수확률 : ");  
            }  
            break;  
    }  
}
```

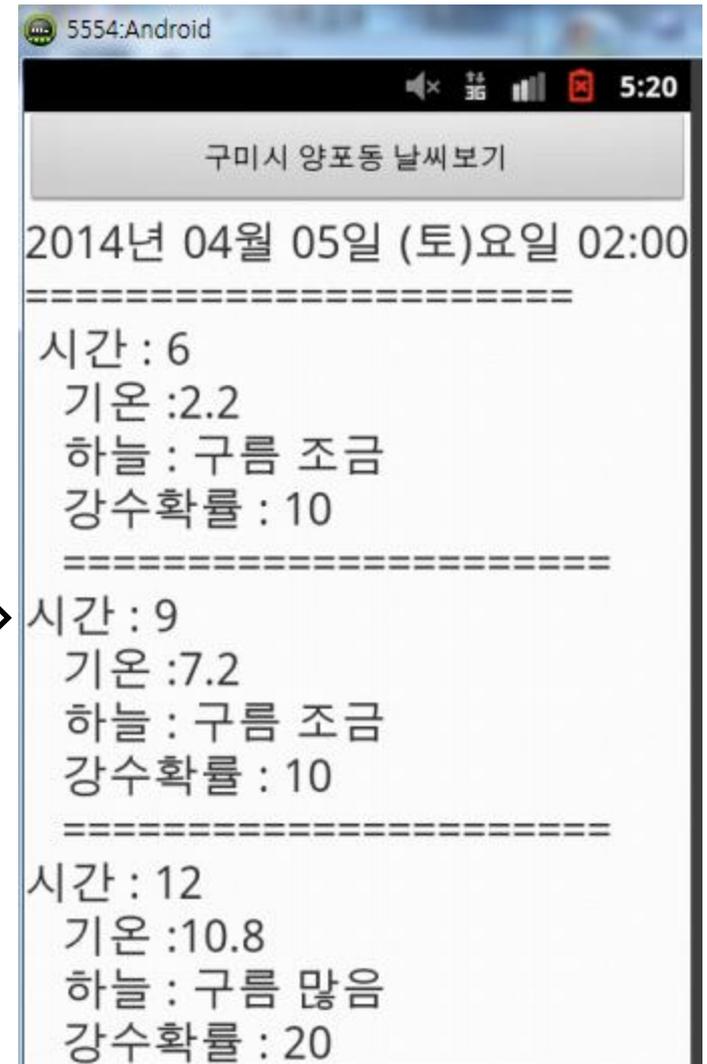
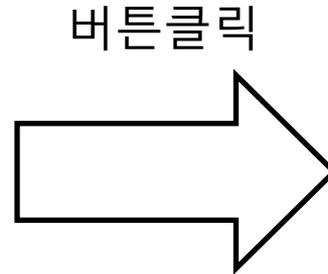
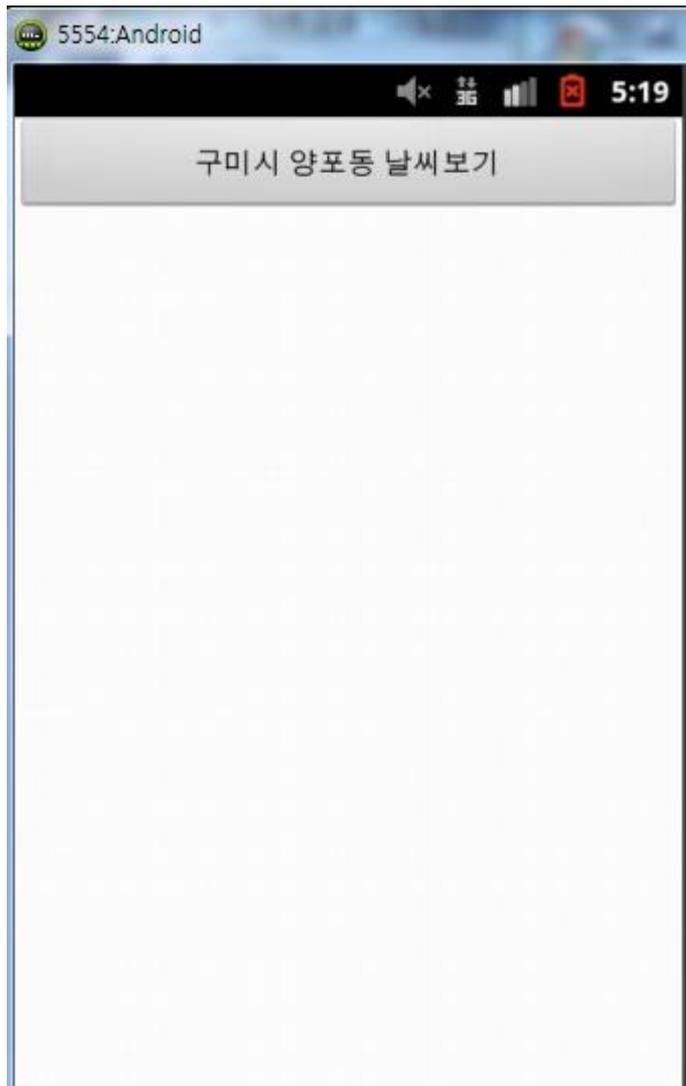
예제 2

TEXT와 END_TAG도 마찬가지로 현재 해당 태그명의 조건에 있을경우 그에 따른 결과를 출력해 주면 된다.
그리고 next()메소드를 통해 다음 태그로 이벤트를 이동시킨다.

```
case XmlPullParser.TEXT:
    if(tag.equals("hour")){
        result.append(parser.getText());
    }else if(tag.equals("temp")){
        result.append(parser.getText());
    }else if(tag.equals("wfKor")){
        result.append(parser.getText());
    }else if(tag.equals("pop")){
        result.append(parser.getText());
    }else if(tag.equals("pubDate")){
        result.append(parser.getText());
    }
    break;
```

```
case XmlPullParser.END_TAG:
    tag = parser.getName();
    if(tag.equals("data")){
        result.append("=====\n");
    }
    if(tag.equals("pubDate")){
        result.append("\n=====\n");
    }
    break;
}
EventType = parser.next();
}
```

예제 2 실행결과



예제1 소스코드

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:id="@+id/parse"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="버튼클릭"/>
    <TextView
        android:id="@+id/result"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="left/top"
        android:textSize="10pt" />
</LinearLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.xmlpullparser"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.xmlpullparser.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>
```

예제2 소스코드

MainActivity.java

```
package com.example.weather;

import java.io.InputStream;
import java.net.URL;

import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserFactory;

import android.support.v7.app.ActionBarActivity;
import android.support.v7.app.ActionBar;
import android.support.v4.app.Fragment;
import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ScrollView;
import android.widget.TextView;
import android.os.Build;

public class MainActivity extends Activity {
    TextView result;
    Button parse;
    ScrollView mScrollView;
    String uri =
"http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=4719069000.xml";
    URL newsURL;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_main);
        mScrollView =
(ScrollView)findViewById(R.id.mScrollView);
        result = (TextView) findViewById(R.id.result);

        findViewById(R.id.parse).setOnClickListener(new OnClickListener()
{
```

```
@Override
public void onClick(View v) {
    result.setText("");
    try {
        newsURL = new URL(uri);
        InputStream in = newsURL.openStream();
        XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
        XmlPullParser parser = factory.newPullParser();
        parser.setInput(in, "utf-8");
        String tag = "";
        int EventType = parser.getEventType();
        while (EventType != XmlPullParser.END_DOCUMENT) {
            switch(EventType){
                case XmlPullParser.START_TAG:
                    tag = parser.getName();
                    if(tag.equals("hour")){
                        result.append("시간 : ");
                    }else if(tag.equals("temp")){
                        result.append("기온 :");
                    }else if(tag.equals("wfKor")){
                        result.append("하늘 : ");
                    }else if(tag.equals("pop")){
                        result.append("강수확률 : ");
                    }
                    break;
                case XmlPullParser.TEXT:
                    if(tag.equals("hour")){
                        result.append(parser.getText());
                    }else if(tag.equals("temp")){
                        result.append(parser.getText());
                    }else if(tag.equals("wfKor")){
                        result.append(parser.getText());
                    }else if(tag.equals("pop")){
                        result.append(parser.getText());
                    }else if(tag.equals("pubDate")){
                        result.append(parser.getText());
                    }
                    break;
                case XmlPullParser.END_TAG:
                    tag = parser.getName();
                    if(tag.equals("data")){
                        result.append("=====\n");
                    }
                    if(tag.equals("pubDate")){
```

예제2 소스코드

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:id="@+id/parse"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="구미시 양포동 날씨보기"/>
    <ScrollView
        android:id="@+id/mScrollView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TextView
            android:id="@+id/result"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:gravity="left/top"
            android:textSize="10pt" />
    </ScrollView>
</LinearLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.weather"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.weather.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>
```

Thank You !