DOM Parser 실습

배 희호 교수
경복대학교
소프트웨어융합과

XML

- XML은 Extensible Markup Language (확장 가능한 마크업 언어)
  - Data Structure를 설명하기 위한 Text 기반 표현
- XML은 인간과 기계 모두 읽을 수 있음
  - SGML(Standardized Generalized Markup Language) 에서 유래
  - 1998년 W3C(World Wide Web Consortium)에서 표준이 됨
- XML은 서로 다른 System 간에 Data를 교환하는 데 적합

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

XML

- XML 문서 구성 요소
  - Elements (요소)
    - <high scale="F">103</high>
  - Tags (태그) : Start tag와 End tag 쌍으로 이루어짐
    - <high scale="F">103</high>
  - Attributes(속성)
    - <high scale="F">103</high>
  - Entities(엔티티)
    - XML에서 특수 문자나 재사용 가능한 문자열을 정의하고 사용할 수 있는 기능
    - <afternoon>Sunny &amp; hot</afternoon>
  - Data
    - <high scale="F">103</high>

XML

- XML 문서는 하나 이상의 처리 명령 또는 지시문으로 시작할 수 있음

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="ss.css"?>
```

- XML 문서의 나머지 부분을 모두 포함하는 Root Element가 정확히 하나만 있어야 함

```
   <weatherReport>        Start tag
        ...
   </weatherReport>       End tag
```

- XML에서 모든 Element에는 Start tag와 End tag가 모두 있어야 함
  - 축약된 형태의 Tag도 가능 <break />

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

XML

- XML의 Tag는 대소문자를 구분
  - <hotel>과 <Hotel>은 서로 다른 Tag로 인식됨
  - 문서 전체에서 일관된 Tag Style을 유지하는 것이 중요함
- Element는 적절하게 중첩 되어야 함
  - 예) <b><i> bold and italic </b></i> (X)
- Attributes의 값은 이중 따옴표로 묶어야 함
  - 예) <time unit="days"></time>
- Attributes와 Elements는 상호 교환 가능
  - Elements는 Programming Language에서 보다 쉽게 처리 가능
  - Attributes에는 정교한 Metadata가 포함될 수 있음,

```
<name>
    <first>David</first>
    <last>Smith</last>
</name>
```

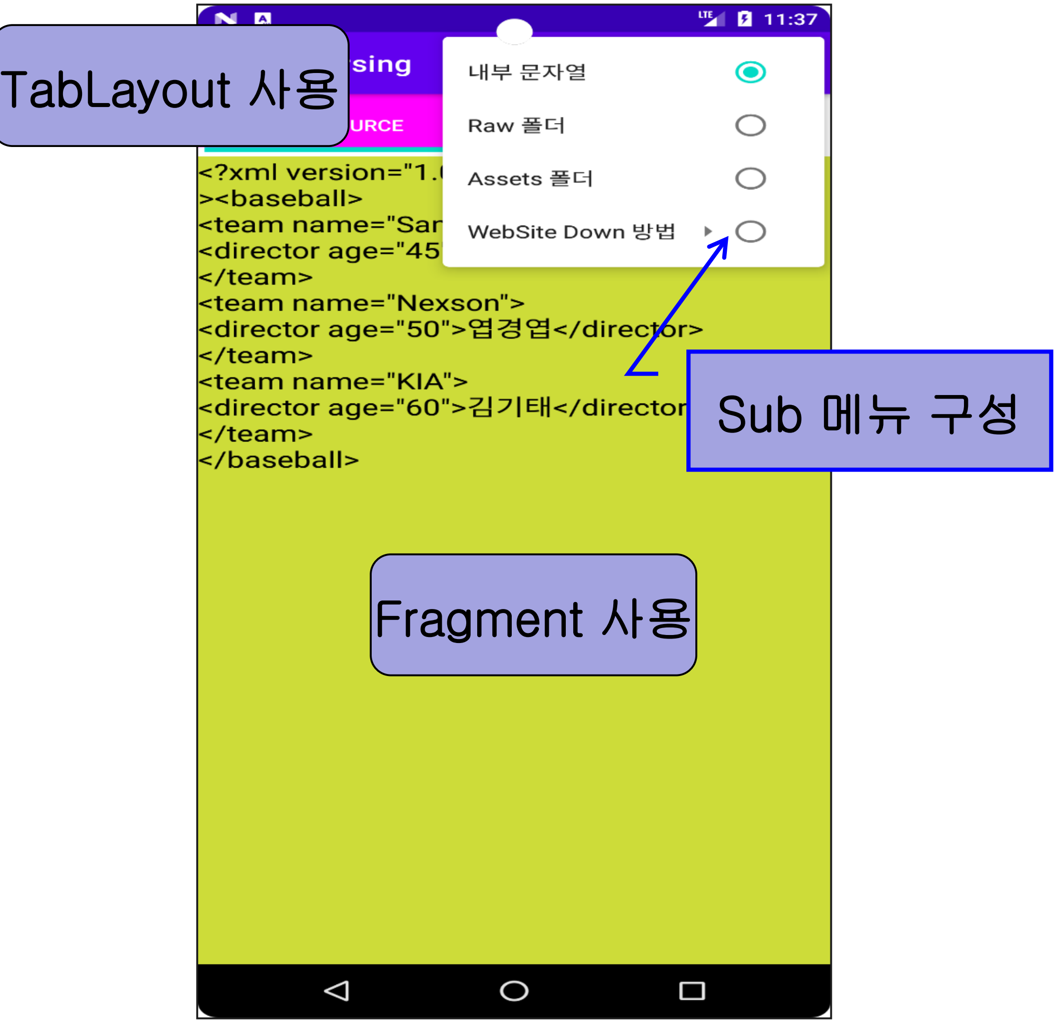
```
<name first="David"
          last="Smith">
</name>
```

XML

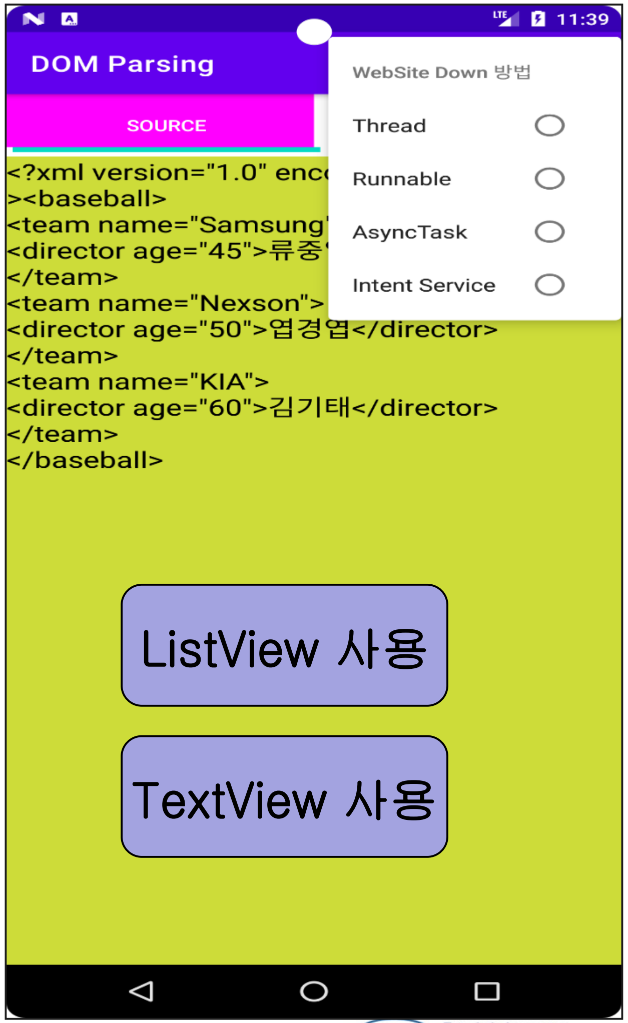
- HTML과 XML
  - HTML과 XML은 둘 다 SGML 언어이기 때문에 비슷해 보임
  - Tags로 묶인 Elements를 사용
    - 예) <body>이것은 요소입니다</body>
  - Tag의 Attributes(속성)을 사용
  - 예) <font face="Verdana" size="+1" color="red">
  - HTML은 SGML에 정의되는 반면 XML은 SGML의 (매우 작은) 부분 집합 임

DOM Parsing

- 다음과 같이 DOM Parsing 방법을 이용하여 Parsing해 보자



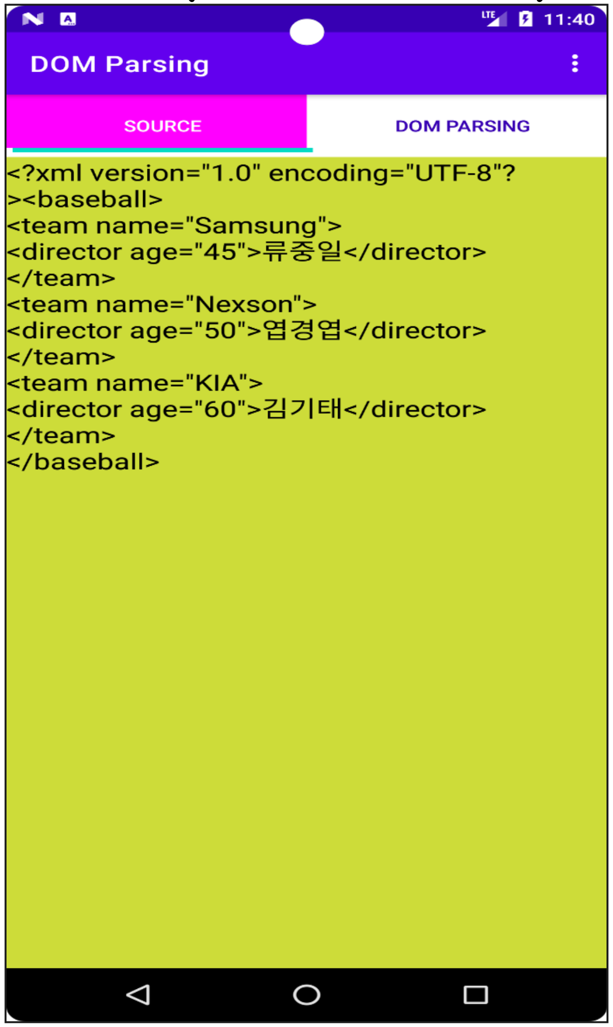
```
<?xml version="1.(
><baseball>
<team name="Sar
<director age="45
</team>
<team name="Nexson">
<director age="50">엽경엽</director>
</team>
<team name="KIA">
<director age="60">김기태</director
</team>
</baseball>
```



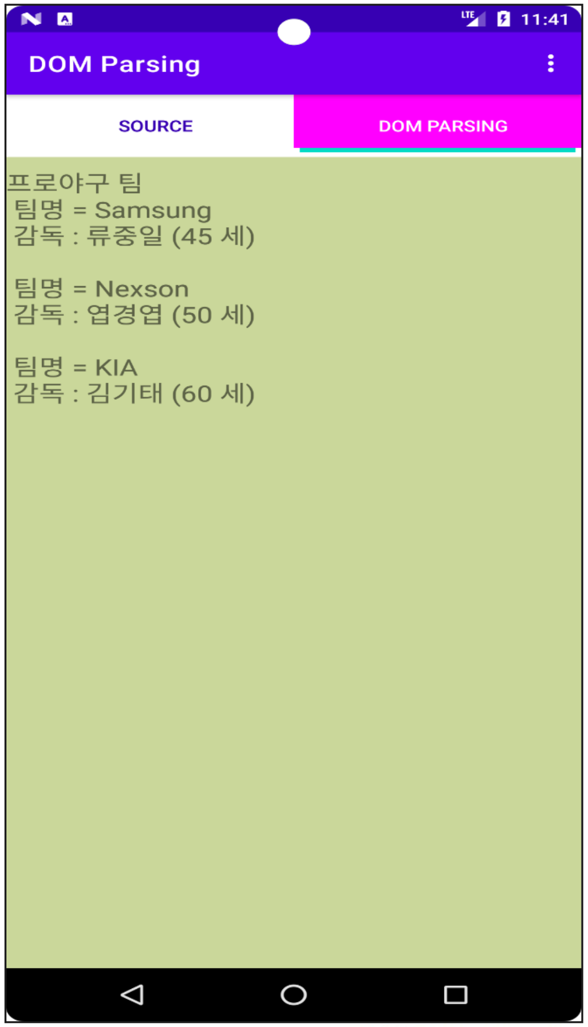
```
<?xml version="1.0" enco
><baseball>
<team name="Samsung"
<director age="45">류중
</team>
<team name="Nexson">
<director age="50">엽경엽</director>
</team>
<team name="KIA">
<director age="60">김기태</director>
</team>
</baseball>
```

DOM Parsing

- 실행 결과 (내부 문자열)



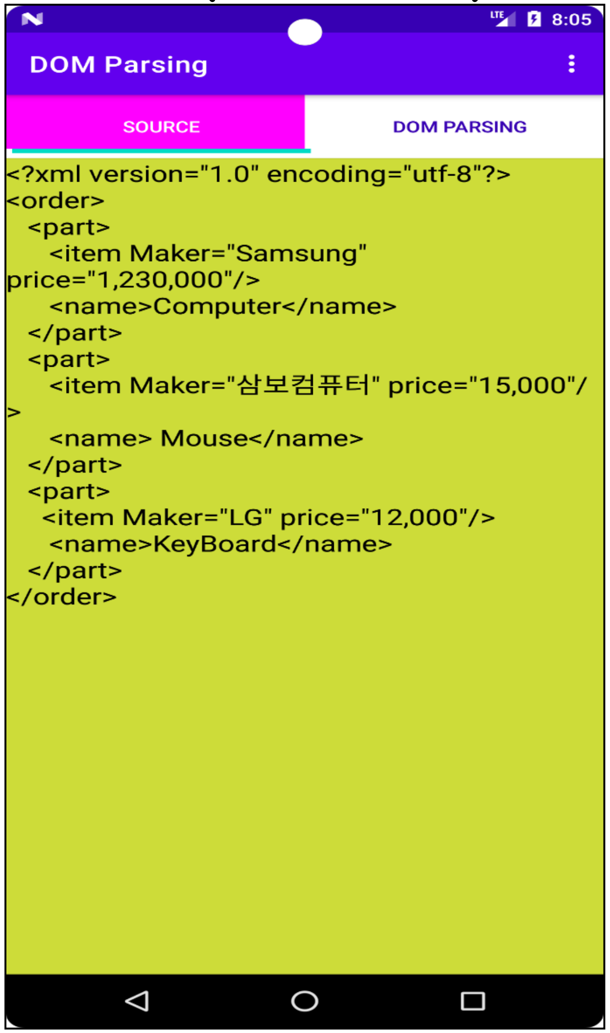
```
<?xml version="1.0" encoding="UTF-8"?
><baseball>
<team name="Samsung">
<director age="45">류중일</director>
</team>
<team name="Nexson">
<director age="50">엽경엽</director>
</team>
<team name="KIA">
<director age="60">김기태</director>
</team>
</baseball>
```



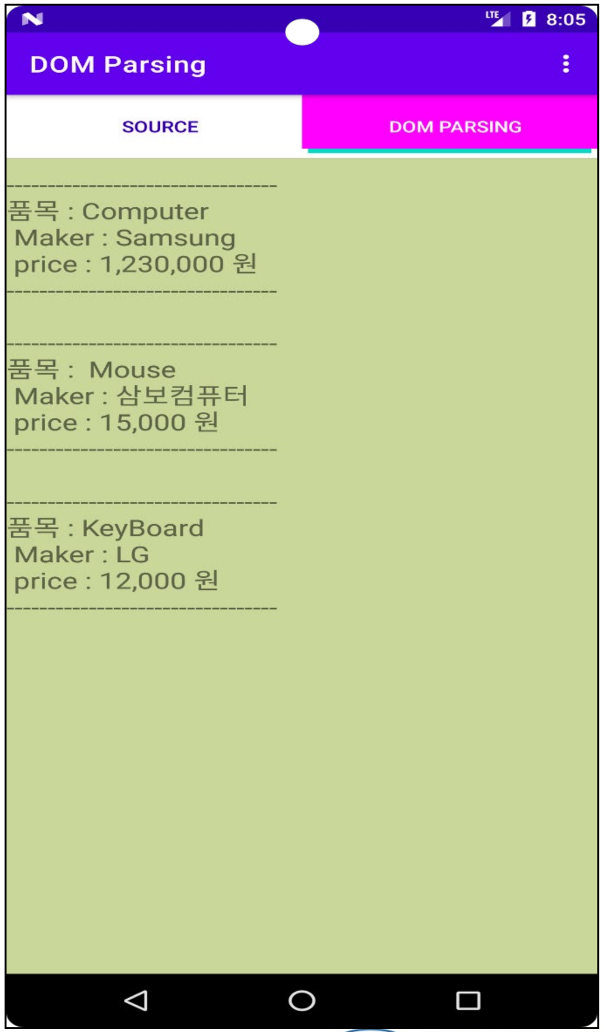
```
프로야구 팀
 팀명 = Samsung
 감독 : 류중일 (45 세)

 팀명 = Nexson
 감독 : 엽경엽 (50 세)

 팀명 = KIA
 감독 : 김기태 (60 세)
```

DOM Parsing

- 실행 결과(raw 폴더)



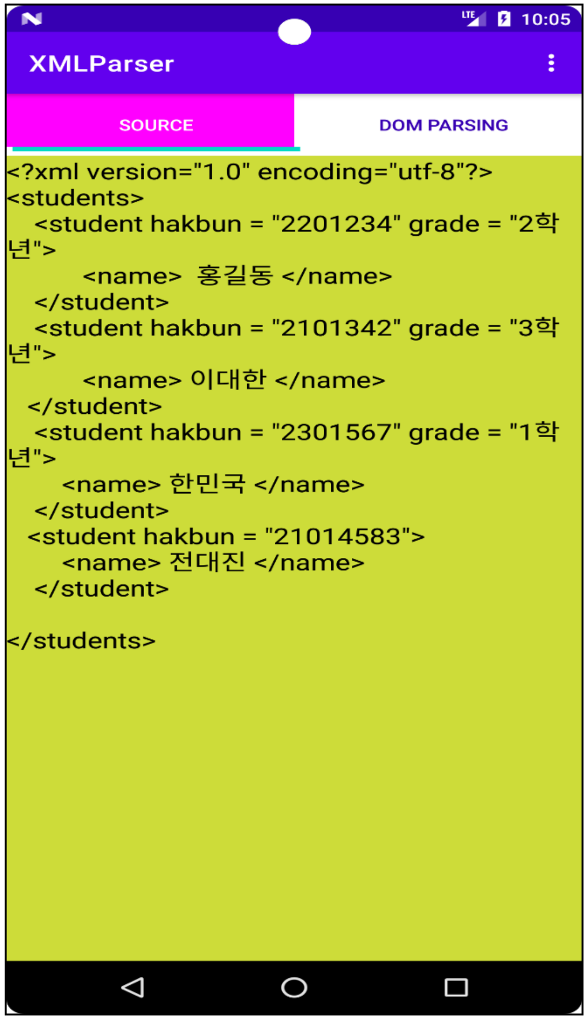
```
<?xml version="1.0" encoding="utf-8"?>
<order>
  <part>
    <item Maker="Samsung"
price="1,230,000"/>
    <name>Computer</name>
  </part>
  <part>
    <item Maker="삼보컴퓨터" price="15,000"/
>
    <name> Mouse</name>
  </part>
  <part>
   <item Maker="LG" price="12,000"/>
    <name>KeyBoard</name>
  </part>
</order>
```



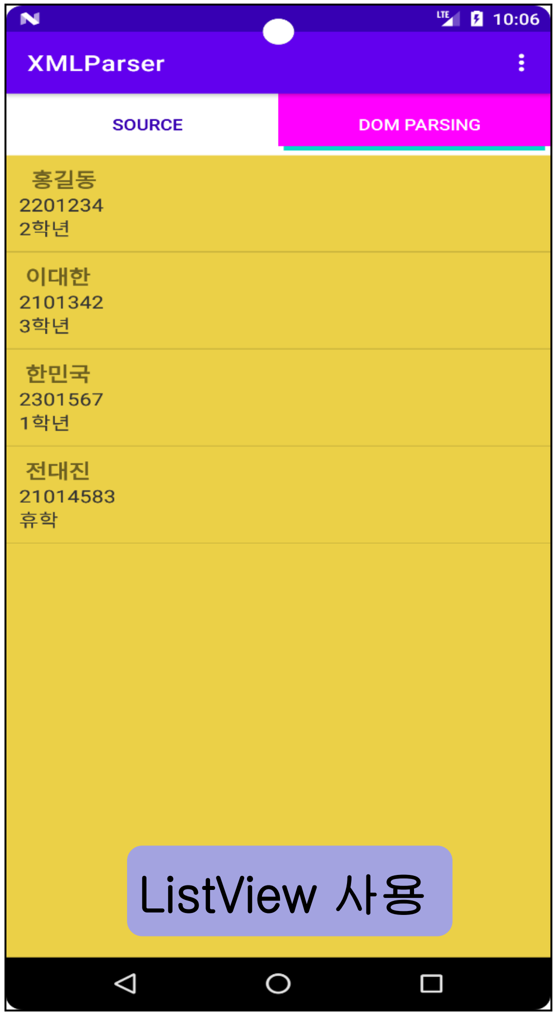
```
------------------------------------
품목 : Computer
 Maker : Samsung
 price : 1,230,000 원
------------------------------------

------------------------------------
품목 :  Mouse
 Maker : 삼보컴퓨터
 price : 15,000 원
------------------------------------

------------------------------------
품목 : KeyBoard
 Maker : LG
 price : 12,000 원
------------------------------------
```

DOM Parsing

- 실행 결과(assets 폴더)



```
<?xml version="1.0" encoding="utf-8"?>
<students>
   <student hakbun = "2201234" grade = "2학
년">
        <name>  홍길동 </name>
   </student>
   <student hakbun = "2101342" grade = "3학
년">
        <name> 이대한 </name>
  </student>
   <student hakbun = "2301567" grade = "1학
년">
      <name> 한민국 </name>
   </student>
  <student hakbun = "21014583">
      <name> 전대진 </name>
   </student>

</students>
```

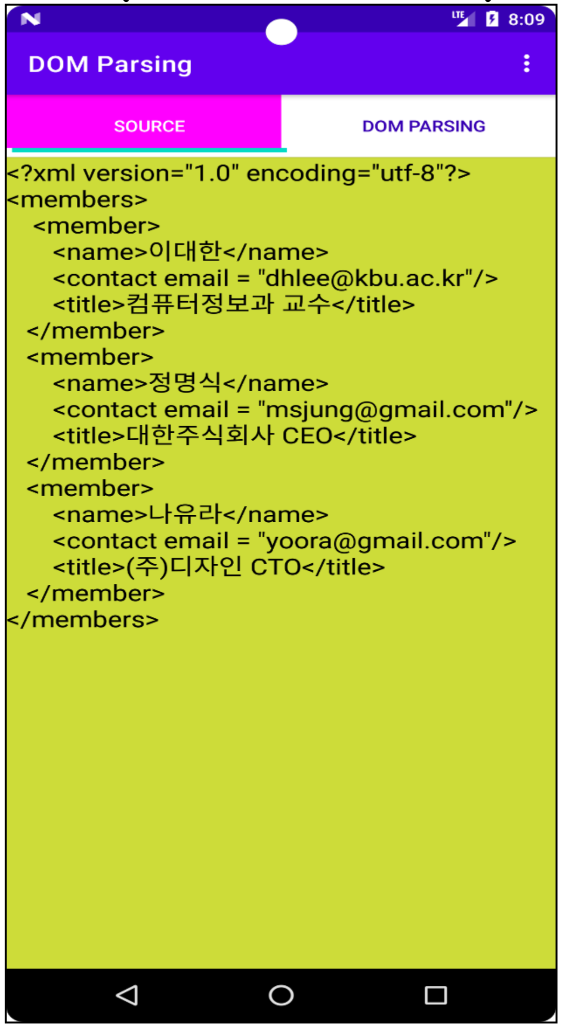


```
홍길동
2201234
2학년
이대한
2101342
3학년
한민국
2301567
1학년
전대진
21014583
휴학
```

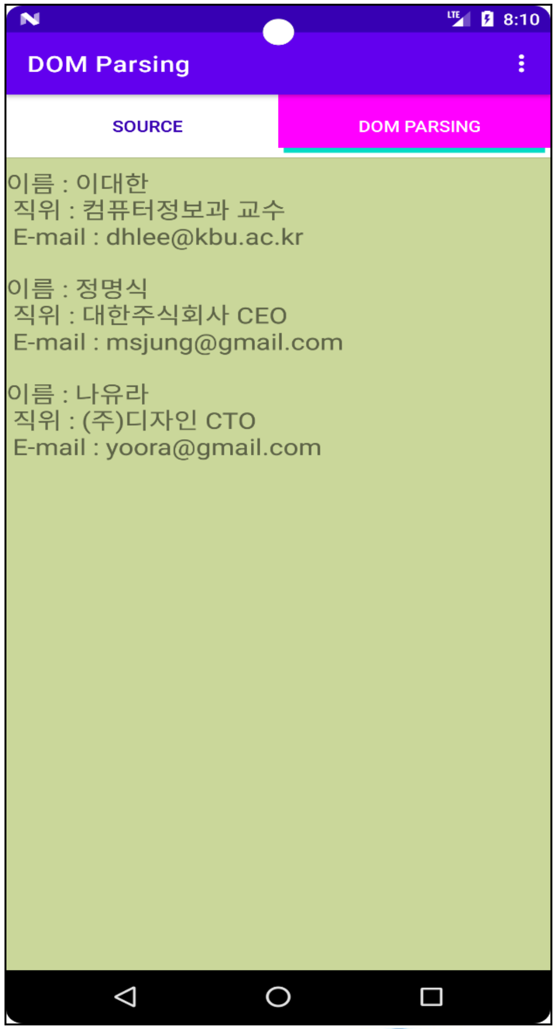
ListView 사용

DOM Parsing

■ 실행 결과(Thread 방법)



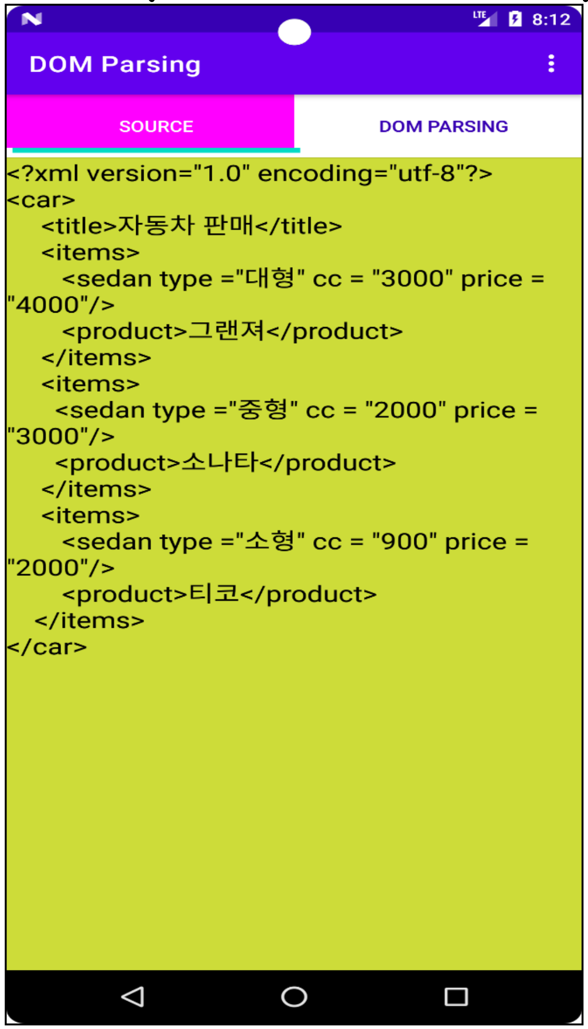
```
<?xml version="1.0" encoding="utf-8"?>
<members>
    <member>
      <name>이대한</name>
      <contact email = "dhlee@kbu.ac.kr"/>
      <title>컴퓨터정보과 교수</title>
   </member>
   <member>
      <name>정명식</name>
      <contact email = "msjung@gmail.com"/>
      <title>대한주식회사 CEO</title>
   </member>
   <member>
      <name>나유라</name>
      <contact email = "yoora@gmail.com"/>
      <title>(주)디자인 CTO</title>
   </member>
</members>
```



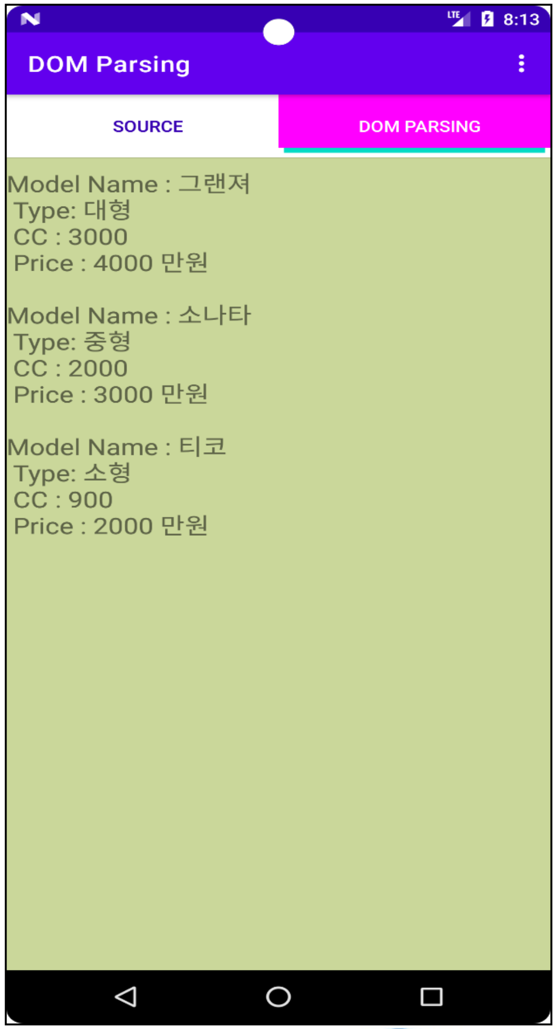
```
이름 : 이대한
 직위 : 컴퓨터정보과 교수
 E-mail : dhlee@kbu.ac.kr

이름 : 정명식
 직위 : 대한주식회사 CEO
 E-mail : msjung@gmail.com

이름 : 나유라
 직위 : (주)디자인 CTO
 E-mail : yoora@gmail.com
```

DOM Parsing

실행 결과(Runnable 방법)



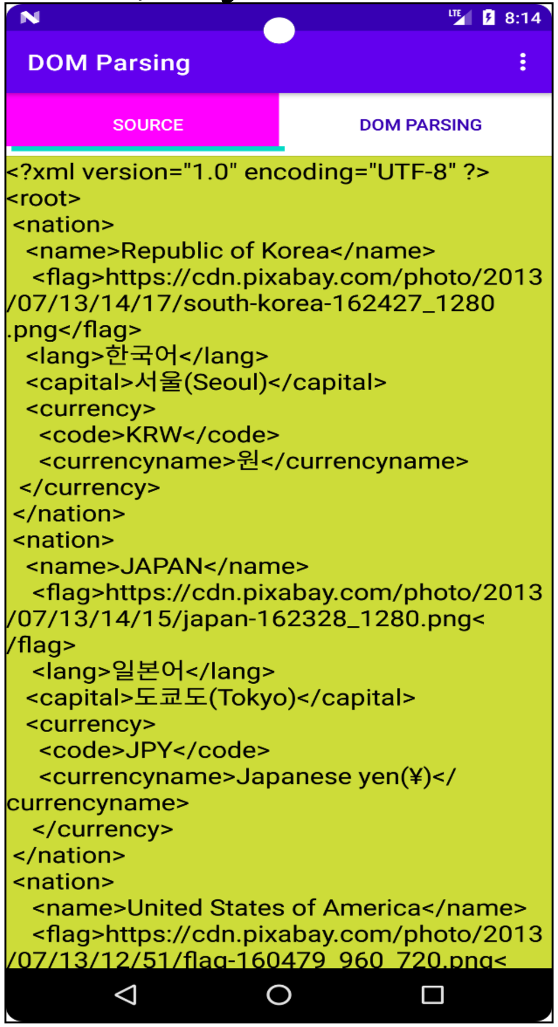
```
<?xml version="1.0" encoding="utf-8"?>
<car>
    <title>자동차 판매</title>
    <items>
      <sedan type ="대형" cc = "3000" price =
"4000"/>
      <product>그랜저</product>
    </items>
    <items>
     <sedan type ="중형" cc = "2000" price =
"3000"/>
     <product>소나타</product>
    </items>
    <items>
      <sedan type ="소형" cc = "900" price =
"2000"/>
      <product>티코</product>
   </items>
</car>
```



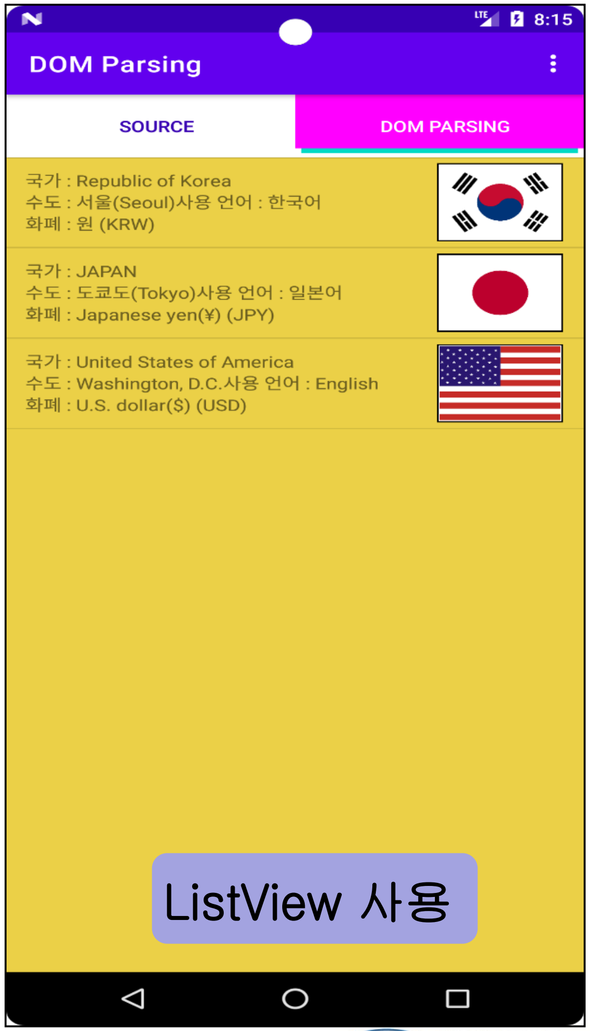
```
Model Name : 그랜져
 Type: 대형
 CC : 3000
 Price : 4000 만원

Model Name : 소나타
 Type: 중형
 CC : 2000
 Price : 3000 만원

Model Name : 티코
 Type: 소형
 CC : 900
 Price : 2000 만원
```

DOM Parsing

■ 실행 결과(AsyncTask 방법)



```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
 <nation>
   <name>Republic of Korea</name>
    <flag>https://cdn.pixabay.com/photo/2013
/07/13/14/17/south-korea-162427_1280
.png</flag>
   <lang>한국어</lang>
   <capital>서울(Seoul)</capital>
   <currency>
     <code>KRW</code>
     <currencyname>원</currencyname>
  </currency>
 </nation>
 <nation>
   <name>JAPAN</name>
    <flag>https://cdn.pixabay.com/photo/2013
/07/13/14/15/japan-162328_1280.png<
/flag>
    <lang>일본어</lang>
   <capital>도쿄도(Tokyo)</capital>
   <currency>
     <code>JPY</code>
     <currencyname>Japanese yen(¥)</
currencyname>
    </currency>
 </nation>
 <nation>
    <name>United States of America</name>
    <flag>https://cdn.pixabay.com/photo/2013
/07/13/12/51/flag-160479_960_720.png<
```



국가 : Republic of Korea
수도 : 서울(Seoul)사용 언어 : 한국어
화폐 : 원 (KRW)

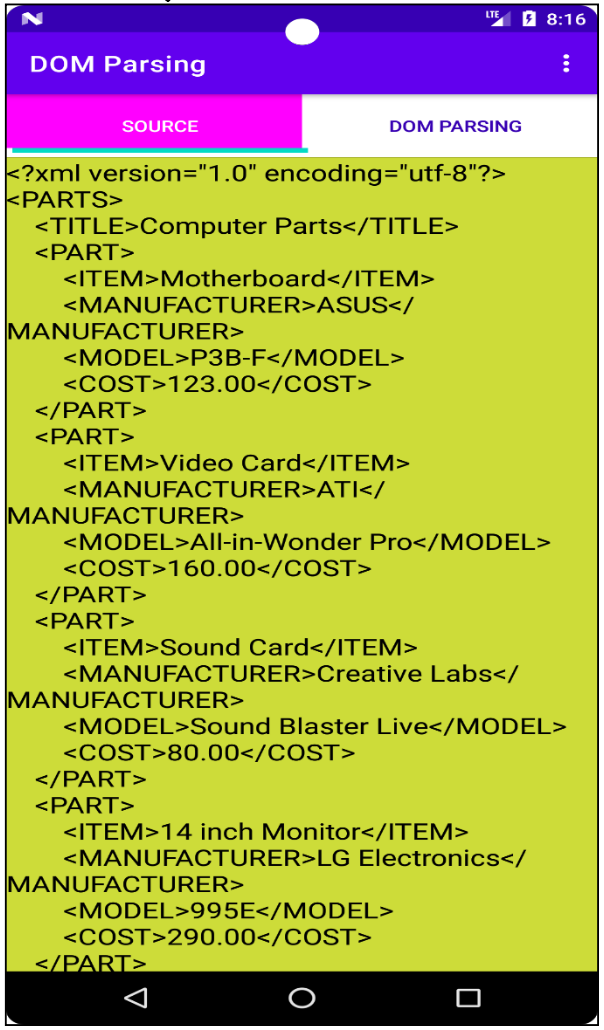
국가 : JAPAN
수도 : 도쿄도(Tokyo)사용 언어 : 일본어
화폐 : Japanese yen(¥) (JPY)

국가 : United States of America
수도 : Washington, D.C.사용 언어 : English
화폐 : U.S. dollar($) (USD)

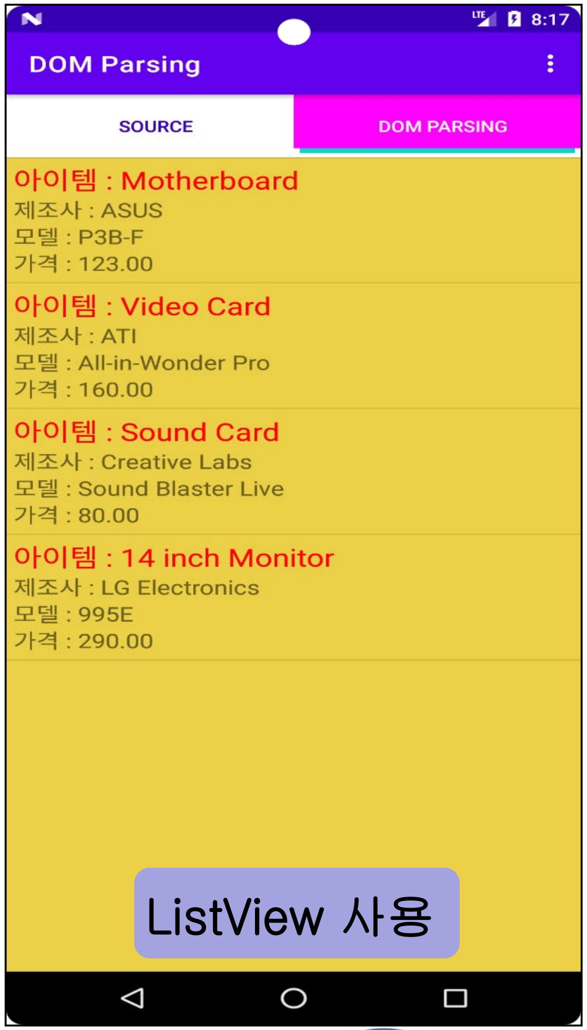
ListView 사용

DOM Parsing

- 실행 결과(intentService 방법)



```
<?xml version="1.0" encoding="utf-8"?>
<PARTS>
   <TITLE>Computer Parts</TITLE>
   <PART>
      <ITEM>Motherboard</ITEM>
      <MANUFACTURER>ASUS</
MANUFACTURER>
      <MODEL>P3B-F</MODEL>
      <COST>123.00</COST>
   </PART>
   <PART>
      <ITEM>Video Card</ITEM>
      <MANUFACTURER>ATI</
MANUFACTURER>
      <MODEL>All-in-Wonder Pro</MODEL>
      <COST>160.00</COST>
   </PART>
   <PART>
      <ITEM>Sound Card</ITEM>
      <MANUFACTURER>Creative Labs</
MANUFACTURER>
      <MODEL>Sound Blaster Live</MODEL>
      <COST>80.00</COST>
   </PART>
   <PART>
      <ITEM>14 inch Monitor</ITEM>
      <MANUFACTURER>LG Electronics</
MANUFACTURER>
      <MODEL>995E</MODEL>
      <COST>290.00</COST>
   </PART>
```



아이템 : Motherboard
제조사 : ASUS
모델 : P3B-F
가격 : 123.00

아이템 : Video Card
제조사 : ATI
모델 : All-in-Wonder Pro
가격 : 160.00

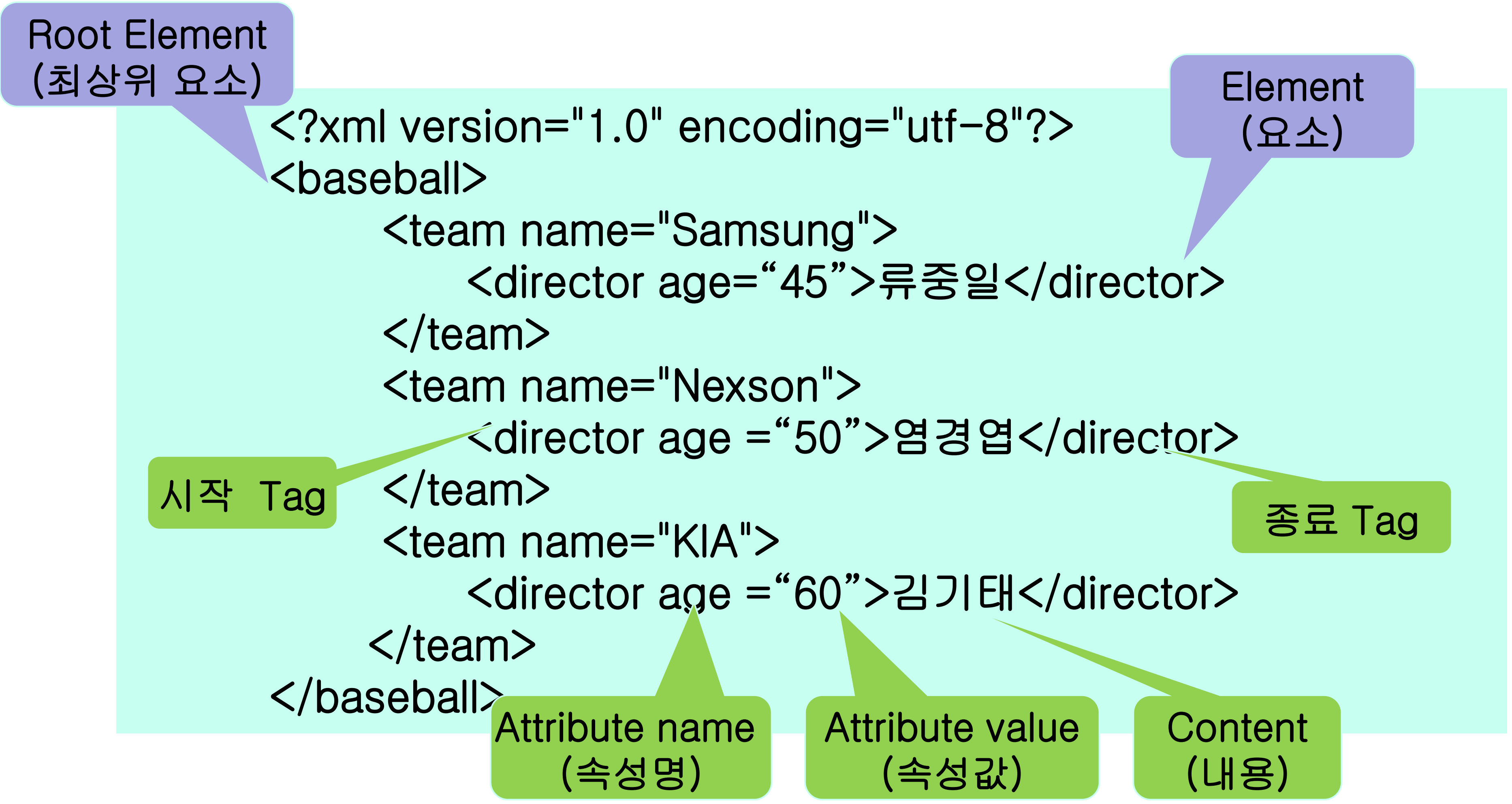
아이템 : Sound Card
제조사 : Creative Labs
모델 : Sound Blaster Live
가격 : 80.00

아이템 : 14 inch Monitor
제조사 : LG Electronics
모델 : 995E
가격 : 290.00

ListView 사용

DOM Parsing

- 내부 문자열로 저장



```
<?xml version="1.0" encoding="utf-8"?>
<baseball>
        <team name="Samsung">
              <director age="45">류중일</director>
        </team>
        <team name="Nexson">
              <director age ="50">염경엽</director>
        </team>
        <team name="KIA">
              <director age ="60">김기태</director>
       </team>
</baseball>
```

DOM Parsing

- [res]-[raw] folder에 저장

order.xml

```
<?xml version="1.0" encoding="utf-8"?>
<order>
    <part>
        <item Maker="Samsung" price="1,230,000"/>
        <name>Computer</name>
    </part>
    <part>
        <item Maker="삼보컴퓨터" price="15,000"/>
        <name>Mouse</name>
    </part>
    <part>
        <item Maker="LG" price="12,000"/>
        <name>KeyBoard</name>
    </part>
</order>
```

종료 Tag
(생략형)

반복되는 Element의 Tag : part

DOM Parsing

Assets 폴더에 저장

student3.xml

```
<?xml version="1.0" encoding="utf-8"?>
<students>
   <student hakbun = "2201234" grade = "2학년">
       <name>  홍길동 </name>
   </student>
   <student hakbun = "2101342" grade = "3학년">
       <name> 이대한 </name>
  </student>
  <student hakbun = "2301567" grade = "1학년">
     <name> 한민국 </name>
  </student>
  <student hakbun = "21014583">
      <name> 전대진 </name>
  </student>
</students>
```

일부 Attribute 생략
빈 Element도 존재

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

DOM Parsing

- Local Web Server에 저장

member.xml

```
<?xml version="1.0" encoding="utf-8"?>
<members>
  <member>
     <name>이대한</name>
     <contact email = "dhlee@kbu.ac.kr"/>
     <title>컴퓨터정보과 교수</title>
 </member>
 <member>
     <name>정명식</name>
     <contact email = "msjung@gmail.com"/>
     <title>대한주식회사 CEO</title>
 </member>
 <member>
     <name>나유라</name>
     <contact email = "yoora@gmail.com"/>
     <title>(주)디자인 CTO</title>
 </member>
</members>
```

Attribute Value
(속성 값)

종료 Tag
(생략형)

DOM Parsing

- Local Web Server에 저장

car.xml

```
<?xml version="1.0" encoding="utf-8"?>
<title>자동차 판매</title>
<car>
  <items>
    <sedan type ="대형" cc = "3000" price = "4000"/>
     <product>그랜져</product>
  </items>
  <items>
    <sedan type ="중형" cc = "2000" price = "3000"/>
    <product>소나타</product>
   </items>
   <items>
     <sedan type ="소형" cc = "900" price = "2000"/>
     <product>티코</product>
  </items>
</car>
```

DOM Parsing

Local Web Server에 저장

nation.xml

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<nations>
  <nation>
    <name>Republic of Korea</name>
    <flag>https://cdn.pixabay.com/photo/2013/07/13/14/17/south-korea-162427_1280.png</flag>
    <lang>한국어</lang>
    <capital>서울(Seoul)</capital>
    <currency>
       <code>KRW</code>
       <currencyname>원</currencyname>
    </currency>
  </nation>
  <nation>
    <name>JAPAN</name>
    <flag>https://cdn.pixabay.com/photo/2013/07/13/14/15/japan-162328_1280.png</flag>
    <lang>일본어</lang>
    <capital>도쿄도(Tokyo)</capital>
    <currency>
      <code>JPY</code>
      <currencyname>Japanese yen(¥)</currencyname>
    </currency>
  </nation>
  <nation>
    <name>United States of America</name>
    <flag>https://cdn.pixabay.com/photo/2013/07/13/12/51/flag-160479_960_720.png</flag>
    <lang>English</lang>
    <capital>Washington, D.C.</capital>
    <currency>
      <code>USD</code>
      <currencyname>U.S. dollar($)</currencyname>
    </currency>
  </nation>
</nations>
```

DOM Parsing

- Local Web Server에 저장 (intent Service)

주문서.xml

```
<PARTS>
   <TITLE>Computer Parts</TITLE>
     <PART>
            <ITEM>Motherboard</ITEM>
            <MANUFACTURER>ASUS</MANUFACTURER>
            <MODEL>P3B-F</MODEL>
            <COST> 123.00</COST>
     </PART>
     <PART>
             <ITEM>Video Card</ITEM>
             <MANUFACTURER>ATI</MANUFACTURER>
             <MODEL>All-in-Wonder Pro</MODEL>
             <COST> 160.00</COST>
    </PART>
    <PART>
             <ITEM>Sound Card</ITEM>
             <MANUFACTURER>Creative Labs</MANUFACTURER>
             <MODEL>Sound Blaster Live</MODEL>
             <COST> 80.00</COST>
    </PART>
    <PART>
              <ITEM>14 inch Monitor</ITEM>
              <MANUFACTURER>LG Electronics</MANUFACTURER>
              <MODEL> 995E</MODEL>
              <COST> 290.00</COST>
     </PART>
</PARTS>
```

DOM Parsing

- HTTP 통신용 URL Class
  - Thread
    - URL 클래스
      - InputStreamReader으로 읽기
  - Runnable
    - URLConnection 클래스
      - BufferedReader로 읽기
  - AsyncTask
    - HttpURLConnection 클래스
      - Scanner로 읽기

DOM Parsing

사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```

DOM Parsing

사용자 인터페이스

```
    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        style="@style/CustomTabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:elevation="1dp"
        android:padding="4dp"
        app:tabGravity="fill"
        app:tabMode="fixed"/>

    <FrameLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

DOM Parsing

- styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="CustomTabLayout" parent="Widget.Design.TabLayout">
        <item name="tabBackground">@drawable/tab_select</item>
        <item name="tabIndicatorHeight">4dp</item> <!-- 인디케이터 숨김 -->
        <item name="tabSelectedTextColor">@color/white</item>
        <item name="tabTextColor">@color/purple_700</item>
    </style>
</resources>
```

DOM Parsing

- menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <group android:checkableBehavior="single">
        <item
            android:id="@+id/item1"
            android:checked="true"
            android:title="내부 문자열" />
        <item
            android:id="@+id/item2"
            android:title="Raw 폴더" />
        <item
            android:id="@+id/item3"
            android:title="Assets 폴더" />
```

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

DOM Parsing

```
        <item android:title="WebSite Down 방법">
            <menu>
                <group android:checkableBehavior="single">
                    <item
                        android:id="@+id/item4"
                        android:title="Thread" />
                    <item
                        android:id="@+id/item5"
                        android:title="Runnable" />
                    <item
                        android:id="@+id/item6"
                        android:title="AsyncTask" />
                    <item
                        android:id="@+id/item7"
                        android:title="Intent Service"/>
                </group>
            </menu>
        </item>
    </group>
</menu>
```

DOM Parsing

- colors.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="selected_tab_color">#FF00FF</color>
    <color name="unselected_tab_color">#FFFFFF</color>
</resources>
```

DOM Parsing

- tab_select.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@color/selected_tab_color"
                                                      android:state_selected="true" />
    <item android:drawable="@color/unselected_tab_color" />
</selector>
```

DOM Parsing

MyApplication.JAVA

```
public class MyApplication extends Application {
    //private String url = "http://10.50.80.32:8080/XML/";
    private String url = "http://192.168.219.103:8080/XML/";
    private String[] page = {url + "member.xml", url + "car.xml",
            url + "nation.xml", url + "주문서.xml"};
    private String xml;

    public String getPage(int index) {
        return page[index];
    }

    public String getXml() {
        return xml;
    }
    public void setXml(String xml) {
        this.xml = xml;
    }
}
```

DOM Parsing

- MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    private int type = R.id.item1;
    private Fragment sourceFragment = null;
    private Fragment parseFragment = null;
    private TabLayout tabs;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setTitle("DOM Parsing");
```

DOM Parsing

- MainActivity.JAVA

```
FragmentManager manager = getSupportFragmentManager();
tabs = findViewById(R.id.tabs);
tabs.setOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        Fragment fragment;
        if (tab.getPosition() == 0)
            fragment = sourceFragment;
        else
            fragment = parseFragment;
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.replace(R.id.container, fragment);
        transaction.commit();
    }

    @Override
    public void onTabUnselected(TabLayout.Tab tab) {
    }
```

DOM Parsing

- MainActivity.JAVA

```
            @Override
            public void onTabReselected(TabLayout.Tab tab) {
            }
        });
        update();
        TabMenu tabMenu = new TabMenu(manager, tabs);
        tabMenu.menu(sourceFragment);
    }

    protected void update() {
        sourceFragment = new SourceFragment(this, type);
        parseFragment = new ParseFragment(this, type);
        tabs.selectTab(tabs.getTabAt(2));
        tabs.selectTab(tabs.getTabAt(0));
    }
```

DOM Parsing

- MainActivity.JAVA

```
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        type = item.getItemId();
        item.setChecked(true);
        update();
        return true;
    }
}
```

Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY

DOM Parsing

- Tabmenu.JAVA

```
public class TabMenu {
    private FragmentManager manager;
    private TabLayout tabs;

    public TabMenu(FragmentManager manager, TabLayout tabs) {
        this.manager = manager;
        this.tabs = tabs;
    }

    public void menu(Fragment fragment) {
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragment);
        transaction.commit();
        tabs.addTab(tabs.newTab().setText("Source"));
        tabs.addTab(tabs.newTab().setText("DOM Parsing"));
    }
}
```

DOM Parsing

```
<?xml version="1.0" encoding="utf-8" ?>
<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:background="#CDDC39"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="20dp" />
    </ScrollView>
</LinearLayout>
```

DOM Parsing

- SourceFragment.JAVA

```
public class SourceFragment extends Fragment {
    private MyApplication app;
    private Activity activity;
    private int type;
    private TextView textView;

    public SourceFragment(Activity activity, int type) {
        app = (MyApplication) activity.getApplication();
        this.activity = activity;
        this.type = type;
    }
```

DOM Parsing

- SourceFragment.JAVA

```
private Handler handler = new Handler(Looper.getMainLooper()) {
    public void handleMessage(Message message) {
        if (message.what == 1) {
            app.setXml((String) message.obj);
            textView.setText(app.getXml());
        } else {
            Toast.makeText(activity, "다운로드 실패함", Toast.LENGTH_LONG).show();
        }
    }
};

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                                     Bundle savedInstanceState) {
    ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.source_fragment,
                                                          container, false);
    textView = rootView.findViewById(R.id.textView);
    return rootView;
}
```

DOM Parsing

- SourceFragment.JAVA

```
@Override
public void onViewCreated(@NonNull View view,
                                    @Nullable Bundle savedInstanceState) {
    if (type == R.id.item1) {
        DOMMaker domMaker = new DOMMaker(activity);
        app.setXml(domMaker.makeDOM());
    } else if (type == R.id.item2) {
        ReadRaw raw = new ReadRaw(activity);
        app.setXml(raw.get(R.raw.order));
    } else if (type == R.id.item3) {
        ReadAssets assets = new ReadAssets(activity);
        app.setXml(assets.get("students3.xml"));
    } else if (type == R.id.item4) {
```

DOM Parsing

- SourceFragment.JAVA

```
        DownloadThread thread = new DownloadThread(activity, app.getPage(0));
        thread.start();
        try {
            thread.join();
            app.setXml(thread.getResult());
        } catch (InterruptedException e)
            Toast.makeText(activity, e.getMessage(), Toast.LENGTH_SHORT).show();
    } else if (type == R.id.item5) {
        DownloadRunnable runnable =
                                new DownloadRunnable(activity, app.getPage(1));
        Thread thread = new Thread(runnable);
        thread.start();
        try {
            thread.join();
            app.setXml(runnable.getResult());
        } catch (InterruptedException e)
            Toast.makeText(activity, e.getMessage(), Toast.LENGTH_SHORT).show();
    } else if (type == R.id.item6) {
```

DOM Parsing

- SourceFragment.JAVA

```
            DownloadAsyncTask task = new DownloadAsyncTask(activity);
            try {
                app.setXml(task.execute(app.getPage(2)).get());
            } catch (ExecutionException | InterruptedException e) {
                Toast.makeText(activity, e.getMessage(),
                        Toast.LENGTH_SHORT).show();
            }
        } else if (type == R.id.item7) {
            Intent intent = new Intent(activity, DownloadService.class);
            Messenger messenger = new Messenger(handler);
            intent.putExtra("Messenger", messenger);
            intent.putExtra("page", app.getPage(3));
            activity.startService(intent);
        }
        textView.setText(app.getXml());
        textView.setTextColor(Color.BLACK);
    }
}
```

DOM Parsing

- SourceFragment.JAVA

```
private Handler handler = new Handler(Looper.getMainLooper()) {
    public void handleMessage(Message message) {
        xml = (String) message.obj;
        if (message.arg1 == RESULT_OK) {
            textView.setText(xml);
        } else
            Toast.makeText(context, "다운로드 실패함",
                                           Toast.LENGTH_LONG).show();
    }
};

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                                  Bundle savedInstanceState) {
    ViewGroup rootView = (ViewGroup) inflater
                           .inflate(R.layout.source_fragment, container, false);
    textView = rootView.findViewById(R.id.textView);
    return rootView;
}
```

DOM Parsing

- SourceFragment.JAVA

```
@Override
public void onViewCreated(@NonNull View view,
                                                  @Nullable Bundle savedInstanceState) {
    if (type == R.id.item1) {
        DOMMaker domMaker = new DOMMaker(context);
        xml = domMaker.makeDOM();
    } else if (type == R.id.item2) {
        ReadRaw raw = new ReadRaw(context);
        xml = raw.get(R.raw.order);
    } else if (type == R.id.item3) {
        ReadAssets assets = new ReadAssets(context);
        xml = assets.get("student.xml");
    } else if (type == R.id.item4) {
```

DOM Parsing

- SourceFragment.JAVA

```
        DownloadThread thread = new DownloadThread(context, page[0]);
        thread.start();
        try {
            thread.join();
            xml = thread.getResult();
        } catch (InterruptedException e) {
            Toast.makeText(context, e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    } else if (type == R.id.item5) {
        DownloadRunnable runnable = new DownloadRunnable(context, page[1]);
        Thread thread = new Thread(runnable);
        thread.start();
        try {
            thread.join();
            xml = runnable.getResult();
        } catch (InterruptedException e) {
            Toast.makeText(context, e.getMessage(), Toast.LENGTH_SHORT).show();
        }
```

DOM Parsing

- DOMMaker.JAVA

```
public class DOMMaker {
    private Context context;
    public DOMMaker(Context context) {
        this.context = context;
    }

    public String source() {
        StringWriter writer = null;
        String[][] team = {{"Samsung", "류중일"}, {"Nexson", "염경엽"},
                                                    {"KIA", "김기태"}};
        int[] age = {45, 50, 60};
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document document = builder.newDocument();

            Element root = document.createElement("baseball");
            document.appendChild(root);
```

DOM Parsing

- DOMMaker.JAVA

```
for (int i = 0; i < team.length; i++) {
    Element element = document.createElement("team");
    element.setAttribute("name", team[i][0]);
    Element node = document.createElement("director");
    node.setTextContent(team[i][1]);
    node.setAttribute("age", String.valueOf(age[i]));
    element.appendChild(node);
    root.appendChild(element);
}

TransformerFactory factory1 = TransformerFactory.newInstance();
Transformer former = factory1.newTransformer();
former.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "no");
former.setOutputProperty(OutputKeys.INDENT, "yes");
```

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

DOM Parsing

- DOMMaker.JAVA

```
            writer = new StringWriter();
            StreamResult result = new StreamResult(writer);
            DOMSource source = new DOMSource(document);
            former.transform(source, result);
        } catch (ParserConfigurationException | TransformerException e) {
            Toast.makeText(context, "", Toast.LENGTH_SHORT).show();
        }

        return String.valueOf(writer);
    }
}
```

Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY

DOM Parsing

- DownloadThread.JAVA

```
public class DownloadThread extends Thread{
    private Context context;
    private String page;
    private StringBuilder builder;
    private Handler handler;

    public DownloadThread(Context context, String page) {
        this.context = context;
        this.page = page;
        builder = new StringBuilder();
        handler = new Handler();
    }
```

DOM Parsing

- DownloadThread.JAVA

```
@Override
public void run() {
    try {
        URL url = new URL(page);
        InputStream inputStream = url.openStream();
        InputStreamReader streamReader =
                                        new InputStreamReader(inputStream, "UTF-8");
        int ch;
        while ((ch = streamReader.read()) != -1) {
            builder.append((char) ch);
        }
        streamReader.close();
        inputStream.close();
    } catch (final IOException e) {
```

DOM Parsing

- DownloadThread.JAVA

```
            handler.post(new Runnable() {
                @Override
                public void run() {
                    Toast.makeText(context, e.getMessage(),
                                                Toast.LENGTH_SHORT).show();
                }
            });
        }
    }

    public String getResult() {
        return builder.toString();
    }
}
```

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

DOM Parsing

- DownloadRunnable.JAVA

```
public class DownloadRunnable implements Runnable{
    private Context context;
    private String page;
    private StringBuilder builder;
    private Handler handler;

    public DownloadRunnable(Context context, String page) {
        this.context = context;
        this.page = page;
        builder = new StringBuilder();
        handler = new Handler();
    }
```

DOM Parsing

- DownloadRunnable.JAVA

```
@Override
public void run() {
    try {
        URL url = new URL(page);
        URLConnection connection = url.openConnection();
        InputStream inputStream = connection.getInputStream();
        InputStreamReader streamReader =
                                new InputStreamReader(inputStream, "UTF-8");
        BufferedReader reader = new BufferedReader(streamReader);
        String line;
        while ((line = reader.readLine()) != null) {
            builder.append(line +'₩n');
        }
        reader.close();
        streamReader.close();
        inputStream.close();
    } catch (IOException e) {
```

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

DOM Parsing

- DownloadRunnable.JAVA

```
            handler.post(new Runnable() {
                @Override
                public void run() {
                    Toast.makeText(context, e.getMessage(),
                                                            Toast.LENGTH_SHORT).show();
                }
            });
        }
    }

    public String getResult() {
        return builder.toString();
    }
}
```

DOM Parsing

- DownloadAsyncTask.JAVA

```
public class DownloadAsyncTask extends AsyncTask<String, String, String> {
    private Context context;

    public DownloadAsyncTask(Context context) {
        this.context = context;
    }

    @Override
    protected String doInBackground(String... strings) {
        StringBuilder builder = new StringBuilder();
        try {
            URL url = new URL(strings[0]);
            HttpURLConnection connection =
                                        (HttpURLConnection) url.openConnection();
            InputStream inputStream = connection.getInputStream();
            InputStreamReader streamReader =
                                new InputStreamReader(inputStream, "UTF-8");
            Scanner scanner = new Scanner(streamReader);
```

DOM Parsing

- DownloadAsyncTask.JAVA

```
            while (scanner.hasNext()) {
                builder.append(scanner.nextLine() +'₩n');
            }
            scanner.close();
            streamReader.close();
            inputStream.close();
            connection.disconnect();
        } catch (IOException e) {
            publishProgress(e.getMessage());
        }
        return builder.toString();
    }

    @Override
    protected void onProgressUpdate(String... values) {
        Toast.makeText(context, values[0], Toast.LENGTH_SHORT).show();
    }
}
```

KYUNGBOK UNIVERSITY

DOM Parsing

- DownloadService.JAVA

```
public class DownloadService extends IntentService {
    private Handler handler;
    private String result;

    public DownloadService() {
        super("DownloadService");
        handler = new Handler();
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        if (intent == null)
            return;
        Messenger messenger = intent.getParcelableExtra("Messenger");
        String page = intent.getStringExtra("page");
        DownloadThread downThread = new DownloadThread(this, page);
        downThread.start();
```

DOM Parsing

- DownloadService.JAVA

```
try {
    downThread.join();
    result = downThread.getResult();
} catch (InterruptedException e) {
    Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
}
Toast.makeText(this, result, Toast.LENGTH_SHORT).show();
Thread thread = new Thread(new Runnable() {
    @Override
    public void run() {
        Message message = Message.obtain();
        message.what = 1;  // 메시지 타입
        message.obj = result;
        try {
            messenger.send(message);
        } catch (RemoteException e) {
```

DOM Parsing

- DownloadService.JAVA

```
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(getBaseContext(), e.getMessage(),
                                                Toast.LENGTH_SHORT).show();
                    }
                });
            }
        }
    });
    thread.start();
  }
  @Override
  public void onDestroy() {
    //   Toast.makeText(this, "Music Service가 중지되었습니다",
                                            Toast.LENGTH_LONG).show();
  }
}
```

KYUNGBOK UNIVERSITY

DOM Parsing

- DOMParser.JAVA

```
public class DOMParser {
    private Context context;

    public DOMParser(Context context) {
        this.context = context;
    }
```

Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY

DOM Parsing

- DOMParser.JAVA

```
public String parsing1(String xml) {
    StringBuilder builder = new StringBuilder();
    builder.append("프로야구 팀₩n");
    Document document = makeDOM(xml);
    NodeList nodeList = document.getElementsByTagName("team");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            builder.append(" 팀명 = " + element.getAttribute("name") + "₩n");
            Element director =
                    (Element) element.getElementsByTagName("director").item(0);
            builder.append(String.format(" 감독 : %s (%d 세)₩n₩n",
                            director.getTextContent(),
                    Integer.parseInt(director.getAttribute("age"))));
        }
    }
    return builder.toString();
}
```

DOM Parsing

```
public String parsing2(String xml) {
    StringBuffer buffer = new StringBuffer();
    Document document = makeDOM(xml);
    NodeList nodeList = document.getElementsByTagName("part");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            Element item = (Element) element.getElementsByTagName("item").item(0);
            String maker = item.getAttribute("Maker");
            String price = item.getAttribute("price");
            String name = element.getElementsByTagName("name")
                                                   .item(0).getTextContent();
            buffer.append("--------------------------------------₩n");
            buffer.append("품목 : " + name + '₩n');
            buffer.append(" Maker : " + maker + "₩n");
            buffer.append(" price : " + price + " 원₩n");
            buffer.append("--------------------------------------₩n₩n");
        }
    }
    return buffer.toString();
}
```

DOM Parsing

- DOMParser.JAVA

```
public  ArrayList<Student> parsing3(String xml) {
    ArrayList<Student> students = new ArrayList<>();
    Document document = makeDOM(xml);
    NodeList nodeList = document.getElementsByTagName("student");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        Student student = new Student();
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            student.setHakbun(element.getAttribute("hakbun"));
            student.setGrade(element.getAttribute("grade"));
            student.setName(element.getElementsByTagName("name")
                                                        .item(0).getTextContent());
        }
        students.add(student);
    }
    return students;
}
```

KYUNGBOK UNIVERSITY

DOM Parsing

```
public String parsing4(String xml) {
    StringBuffer buffer = new StringBuffer();
    Document document = makeDOM(xml);
    NodeList nodeList = document.getElementsByTagName("member");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            String name = element.getElementsByTagName("name")
                                                  .item(0).getTextContent();
            Element item = ((Element) element.getElementsByTagName("contact")
                                                                  .item(0));
            String email = item.getAttribute("email");
            String title = element.getElementsByTagName("title").item(0).getTextContent(
            buffer.append("이름 : " + name + "₩n 직위 : " + title +
                  "₩n E-mail : " + email + "₩n₩n");
        }
    }
    return buffer.toString();
}
```

DOM Parsing

```
public String parsing5(String xml) {
    StringBuffer buffer = new StringBuffer();
    Document document = makeDOM(xml);
    NodeList nodeList = document.getElementsByTagName("items");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = itemList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            Element item = (Element) element.getElementsByTagName("sedan").item(0);
            String type = item.getAttribute("type");
            String cc = item.getAttribute("cc");
            String price = item.getAttribute("price");
            String modelName = element.getElementsByTagName("product")
                                                    .item(0).getTextContent();
            buffer.append("Model Name : " + modelName + '₩n');
            buffer.append(" Type: " + type + '₩n');
            buffer.append(" CC : " + cc + '₩n');
            buffer.append(" Price : " + price + " 만원₩n₩n");
        }
    }
    return buffer.toString();
}
```

DOM Parsing

- DOMParser.JAVA

```
public ArrayList<Country> parsing6(String xml) {
    ArrayList<Country> countries = new ArrayList<>();
    Document document = pre(xml);
    NodeList nodeList = document.getElementsByTagName("nation");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Country country = new Country();
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            country.setCountry(element.getElementsByTagName("name")
                                                .item(0).getTextContent());
            country.setFlag(element.getElementsByTagName("flag")
                                                .item(0).getTextContent());
            country.setLang(element.getElementsByTagName("lang")
                                                .item(0).getTextContent());
```

DOM Parsing

- DOMParser.JAVA

```
                country.setCapital(element.getElementsByTagName("capital")
                                                        .item(0).getTextContent());
                country.setCurrency(element.getElementsByTagName("currencyname")
                                                        .item(0).getTextContent());
                country.setCode(element.getElementsByTagName("code")
                                                        .item(0).getTextContent());
            }
            countries.add(country);
        }
        return countries;
    }
```

DOM Parsing

```
public ArrayList<Map<String, String>> parsing7(String xml) {
    ArrayList<Map<String, String>> items = new ArrayList<>();
    Document document = makeDOM(xml);
    NodeList nodeList = document.getElementsByTagName("PART");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Map<String, String> item = new HashMap<>();
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            item.put("ITEM", "아이템 : " + element
                    .getElementsByTagName("ITEM").item(0).getTextContent());
            item.put("MANUFACTURER", "제조사 : " + element
                 .getElementsByTagName("MANUFACTURER").item(0).getTextContent());
            item.put("MODEL", "모델 : " + element.getElementsByTagName("MODEL")
                                                        .item(0).getTextContent());
            item.put("COST", "가격 : " + element.getElementsByTagName("COST")
                                                 .item(0).getTextContent());
            items.add(item);
        }
    }
    return items;
}
```

DOM Parsing

- DOMParser.JAVA

```
    private Document makeDOM(String xml) {
        InputStream stream = new ByteArrayInputStream(xml.getBytes(
                                                    StandardCharsets.UTF_8));
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        Document document = null;
        try {
            DocumentBuilder builder = factory.newDocumentBuilder();
            document = builder.parse(stream);
        } catch (ParserConfigurationException | IOException | SAXException e) {
            Toast.makeText(context, e.getMessage(), Toast.LENGTH_SHORT).show();
        }
        return document;
    }
}
```

Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY

DOM Parsing

- Country.JAVA

```java
public class Country {
    private String country;
    private String lang;
    private String capital;
    private String currency;
    private String code;

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getLang() {
        return lang;
    }
```

DOM Parsing

- Country.JAVA

```
public void setLang(String lang) {
    this.lang = lang;
}

public String getCapital() {
    return capital;
}

public void setCapital(String capital) {
    this.capital = capital;
}

public String getCurrency() {
    return currency;
}
```

DOM Parsing

- Country.JAVA

```
    public void setCurrency(String currency) {
        this.currency = currency;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }
}
```

DOM Parsing

- CountryAdapter.JAVA

```
public class CountryAdapter extends ArrayAdapter<Country> {
    private Context context;

    public CountryAdapter(@NonNull Context context, int resource,
                                                               List<Country> countries) {
        super(context, resource, countries);
        this.context = context;
    }
```

DOM Parsing

- CountryAdapter.JAVA

```
@NonNull
@Override
public View getView(int position, @Nullable View convertView,
                                                    @NonNull ViewGroup parent) {
    ViewHolder holder;
    LayoutInflater inflater = (LayoutInflater) context.getSystemService(
                                              Activity.LAYOUT_INFLATER_SERVICE);
    if (convertView == null) {
        convertView = inflater.inflate(R.layout.country, parent, false);
        holder = new ViewHolder(convertView);
        convertView.setTag(holder);
    } else
        holder = (ViewHolder) convertView.getTag();
    Country country = getItem(position);
```

DOM Parsing

- CountryAdapter.JAVA

```
        holder.textView1.setText(" 국가 : " + country.getCountry());
        holder.textView2.setText(" 수도 : " + country.getCapital() + "사용 언어 : "
                                                                + country.getLang());
        holder.textView3.setText(" 화폐 : " + country.getCurrency()
                                                  + " (" + country.getCode() +")");
        Glide.with(context).load(country.getFlag())
                    .placeholder(R.drawable.ic_launcher).into(holder.imageView);
        return convertView;
    }
```

DOM Parsing

- CountryAdapter.JAVA

```
    private class ViewHolder {
        private TextView textView1;
        private TextView textView2;
        private TextView textView3;
        private ImageView imageView;

        public ViewHolder(View v) {
            textView1 = v.findViewById(R.id.textView1);
            textView2 = v.findViewById(R.id.textView2);
            textView3 = v.findViewById(R.id.textView3);
            imageView = v.findViewById(R.id.imageView);
        }
    }
}
```

DOM Parsing

- Student.JAVA

```
public class Student {
    private String name;
    private String hakbun;
    private int age;
    private String depart;

    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }

    public String getDepart() {
        return depart;
    }
    public void setDepart(String depart) {
        this.depart = depart;
    }
```

DOM Parsing

- Student.JAVA

```
    public String getHakbun() {
        return hakbun;
    }

    public void setHakbun(String hakbun) {
        this.hakbun = hakbun;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

DOM Parsing

- StudentAdapter.JAVA

```
class StudentAdapter extends ArrayAdapter<Student> {
    private Context context;

    public StudentAdapter(@NonNull Context context, int resource,
                                                        List<Student> employees) {
        super(context, resource, employees);
        this.context = context;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView,
                                                @NonNull ViewGroup parent) {
        ViewHolder holder;
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(
                                            Activity.LAYOUT_INFLATER_SERVICE);
```

DOM Parsing

- StudentAdapter.JAVA

```
        if (convertView == null) {
            convertView = inflater.inflate(R.layout.item2_list, parent, false);
            holder = new ViewHolder(convertView);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }
        Student student = getItem(position);
        holder.textView1.setText(student.getName());
        holder.textView2.setText(student.getHakbun());
        holder.textView3.setText(student.getGrade() == "" ?
                                                         "휴학" : student.getGrade() );

        return convertView;
    }
```

Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY

DOM Parsing

- StudentAdapter.JAVA

```
    private class ViewHolder {
        private TextView textView1;
        private TextView textView2;
        private TextView textView3;

        public ViewHolder(View v) {
            textView1 = v.findViewById(R.id.textView1);
            textView2 = v.findViewById(R.id.textView2);
            textView3 = v.findViewById(R.id.textView3);
        }
    }
}
```

DOM Parsing

- Item2_list.JAVA

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:textStyle="bold" />
```

DOM Parsing

- Item2_list.JAVA

```
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#343434"
        android:textSize="16dp" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#343434"
        android:textSize="16dp" />
</LinearLayout>
```

DOM Parsing

- country.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="5dp">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="70dp"
        android:layout_marginLeft="5dp"
        android:layout_weight="1"
        android:gravity="center_vertical"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Country : " />
```

DOM Parsing

- country.xml

```
        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="언어 : " />

        <TextView
            android:id="@+id/textView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Population : " />
    </LinearLayout>
```

DOM Parsing

- country.xml

```
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="90dp"
        android:layout_height="70dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:background="@drawable/square_border_black"
        android:padding="2dp"
        android:scaleType="centerCrop" />
</LinearLayout>
```

DOM Parsing

- square_border_black.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <solid android:color="@color/white" />
    <corners
        android:radius="0dp"/>
    <stroke
        android:width="1dp"
        android:color="@color/black" />
</shape>
```

DOM Parsing

- list_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:padding="5dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#FF0000" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium" />
```

DOM Parsing

- list_item.xml

```
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

DOM Parsing

- fragment_parse.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#CAD79A"
            android:paddingTop="10dp"
            android:text="파싱 결과가 여기에 표시됩니다."
            android:textSize="20dp" />
```

DOM Parsing

- fragment_parse.xml

```
        <ListView
            android:id="@+id/listView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#EBD047"
            android:visibility="invisible" />
    </FrameLayout>
</LinearLayout>
```

DOM Parsing

- ParseFragment.JAVA

```
public class ParseFragment extends Fragment {
    private MyApplication app;
    private Activity activity;
    private int type;
    private TextView textView;
    private ListView listView;

    public ParseFragment(Activity activity, int type) {
        this.activity = activity;
        this.type = type;
        app = (MyApplication) activity.getApplication();
    }
```

DOM Parsing

- ParseFragment.JAVA

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                                Bundle savedInstanceState) {
    ViewGroup rootView;
    if (type == R.id.item3 || type == R.id.item6 || type == R.id.item7) {
        rootView = (ViewGroup) inflater.inflate(R.layout.fragment_parse2,
                                                          container, false);
        listView = rootView.findViewById(R.id.listView);
    } else {
        rootView = (ViewGroup) inflater.inflate(R.layout.fragment_parse1,
                                                            container, false);
        textView = rootView.findViewById(R.id.textView);
    }

    return rootView;
}
```

Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY

DOM Parsing

- ParseFragment.JAVA

```
@Override
public void onViewCreated(@NonNull View view,
                          @Nullable Bundle savedInstanceState) {
    DOMParser parser = new DOMParser(activity);
    if (type == R.id.item3) {
        ArrayList<Student> students = parser.parsing3(app.getXml());
        StudentAdapter adapter = new StudentAdapter(activity,
                R.layout.item2_list, students);
        listView.setAdapter(adapter);
    } else if (type == R.id.item6) {
        ArrayList<Country> userList = parser.parsing6(app.getXml());
        ArrayAdapter<Country> adapter = new CountryAdapter(activity,
                R.layout.country, userList);
        listView.setAdapter(adapter);
    } else if (type == R.id.item7) {
```

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

DOM Parsing

- ParseFragment.JAVA

```
            ArrayList<Map<String, String>> items = parser.parsing7(app.getXml());
            SimpleAdapter adapter = new SimpleAdapter(activity, items,
                    R.layout.list_item,
                    new String[]{"ITEM", "제조사", "MODEL", "COST"},
                    new int[]{R.id.textView1, R.id.textView2, R.id.textView3,
                                                            R.id.textView4});
            listView.setAdapter(adapter);
        } else if (type == R.id.item1)
            textView.setText(parser.parsing1(app.getXml()));
        else if (type == R.id.item2)
            textView.setText(parser.parsing2(app.getXml()));
        else if (type == R.id.item4)
            textView.setText(parser.parsing4(app.getXml()));
        else if (type == R.id.item5)
            textView.setText(parser.parsing5(app.getXml()));
    }
}
```

Futuristic Innovator 京福大學校 KYUNGBOK UNIVERSITY

Messenger

- Messenger는 Android의 IPC(Inter-Process Communication) Mechanism에서 사용되는 Class
- 이 Class는 Message 기반 통신을 위해 설계되었으며, Process 간에 간단한 Data를 주고받는 데 사용
- 주로 Android의 Handler 및 Message와 함께 사용