熱血講義

프리렉의 열혈강의 시리즈

Python 파이썬

열혈강의 Python 파이썬

❖제 21-3 강

주제 : XPath

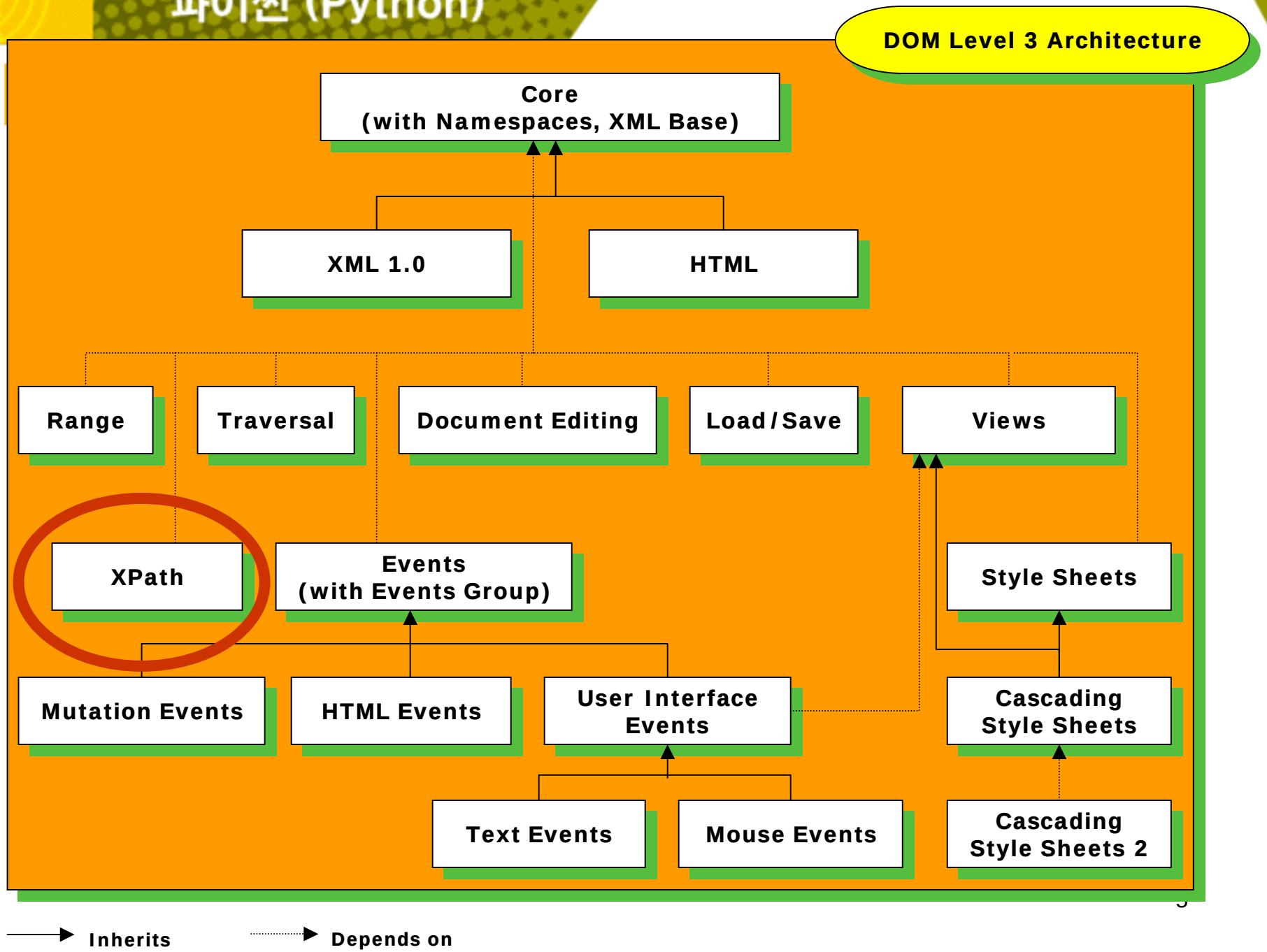
이강성(gslee@mail.kw.ac.kr)

❖강의 목차

1. 개요
2. SAX
3. DOM
4. Traversal
5. XPath
6. XSLT

XPath

- XML Path Language
- DOM 레벨 3에 추가된 인터페이스
- http://www.w3.org/TR/xpath20/
- XML 문서의 일부분을 참조/검색 하기 위한 언어
  - 절차적 언어는 아님 (흐름 제어 구문이 없다)
  - 노드 검색을 위한 기능으로만 사용된다
  - 'user/name/text()'
- XSLT, XQuery, XPointer에 사용됨
  - http://www.w3.org/TR/xslt20/
  - http://www.w3.org/TR/xquery/
  - http://www.w3.org/TR/xptr/
- 여기서는 4XPath 사용 (PyXML도 사용가능)
  - http://4suite.org 에서 다운로드 설치


DOM Level 3 Architecture
Core
(with Namespaces, XML Base)
XML 1.0
HTML
Range
Traversal
Document Editing
Load/Save
Views
XPath
Events
(with Events Group)
Style Sheets
Mutation Events
HTML Events
User Interface
Events
Cascading
Style Sheets
Text Events
Mouse Events
Cascading
Style Sheets 2
Inherits
Depends on

결과 자료형

- XPath 처리 결과 자료 형
  - 노드 집합(node set)
  - 문자열 : unicode string
  - 숫자 : floating point number
  - 부울값 : true or false

Context(문맥)

- Context
  - XPath 식이 적용되고 있는 노드

```
<xsl:for-each select="user">
  <xsl:value-of select="name">
</xsl:for-each>
```

  - XSLT, XPointer 에 자주 적용됨
- Context 구성 요소
  - Context node – 현재 작업 노드
  - Context size – 전체 문맥 노드의 수
  - Context position – 현재 문맥 노드의 위치
  - Variable bindings – 어떤 값에 연결된 변수의 집합
  - Defined namespaces –영역 안에 정의된 이름 공간 정의 집합

Location path(위치 경로)

- **Location path(위치 경로)**
  - 가장 일반적으로 사용되는 **XPath** 표현식
  - 마치 디렉토리나 파일 경로지정과 유사
  - 다양한 조건의 테스트에 의한 경로 지정이 가능
  - 여러 노드가 지정될 수 있다

테스트 문서 (sample04.xml)

```
<?xml version="1.0" ?>
<userlist>
    <user sex="male">
        <name>gslee</name>
        <email>gslee@mail.kw.ac.kr</email>
    </user>
    <user sex="female">
        <name>spam</name>
        <email>spam@mail.kw.ac.kr</email>
    </user>
    <user sex="male">
        <name>HongGilDong</name>
        <email>home@mail.kw.ac.kr</email>
    </user>
    <user sex="female">
        <name>Sunny</name>
        <email>sunny@mail.kw.ac.kr</email>
    </user>
</userlist>
```

Path expression(경로 표현식)

- '/' 혹은 '//'로 분리된 단계 표현식(step expression)의 연결
- 첫 '/'는 루트 노드 의미
- '//'는 문맥 노드 혹은 자손 노드를 의미
- 단계(step) 표현식
  - ForwardAxis :: NodeTest
  - ReverseAxis :: NodeTest
  - AbbreviatedForwardStep
  - AbbreviatedReverseStep

Axis

- 문맥에서 진행방향을 결정

ForwardAxis
- child
- descendant
- attribute
- self
- descendant-or-self
- following-sibling
- following
- namespace

ReverseAxis
- parent
- ancestor
- preceding-sibling
- preceding
- ancestor-or-self

Node test

- Axis의 노드를 줄이는데 사용
- 이름 테스트
  - 노드 이름을 테스트 혹은 '*'
- 종류 테스트
  - processing-instruction()
  - comment()
  - text()
  - node()

Axis::NoteTest

- 예제
  - `descendant::email`
  - `ancestor::*`
  - `preceding-sibling::processing-instruction()`
  - `attribute::*`
  - `following-sibling::*/descendant::name/child::text()`
  - `child::ADDRBOOK/child::ENTRY/child::PHONENUM/child::text()`

Abbreviations(축약)

- 유형

표현식	축약형
child::	생략됨
attribute::	@
self::	.
parent::	..
/descendant-or-self::node()	//

```
following-sibling::*/descendant::PHONENUM/text()
ADDRESSBOOK/ENTRY/PHONENUM/text()
```

Predicates(술부)

- [..] 로 표현되는 식
- 부울 값 혹은 숫자를 리턴하는 표현식
- 부울 값인 경우 참인 노드만 취한다
- 숫자인 경우, 지정된 위치의 노드만이 취해진다
- 예

```
child::user[attribute::sex="male"]/child::email[1]
```

혹은

```
user[@sex="male"]/email[1]
```

Samples

- Sample XPath 표현식
  - 'user/name'
  - 'user/name/text()'
  - 'user[2]/name/text()'
  - 'user[@sex="male"]'
  - '//email'
  - '/userlist//email'
  - 'user/email/../name'

기본 연산자

- 산술 연산자
  - +, -, *, div, idiv, mod
  - <xsl:value-of select="@price * 0.8">
- 비교 연산자
  - <, >, <=, >=, =, !=
  - lt, gt, le, ge, eq, ne
- 노드 비교
  - is, isnot
  - //book[isbn="1234"] is //book[call="q.abc 890"]
- 논리 연산자
  - and, or
- 순서 비교 연산자
  - <<, >>

함수

함수	설명
position()	현재 문맥 위치 리턴
last()	문맥 크기 / 마지막 문맥 노드 리턴
count(<node set>)	<node set>의 노드의 수 리턴
local-name(<node set>)	첫 노드의 지역 이름 리턴
concat(<string>, .., <string>)	문자열 연결하여 리턴
contains(<string>,<string>)	두 번째 문자열이 첫 번째 문자열 안에 포함되면 '참'을 리턴
not(<boolean>)	'참'이면 '거짓'을, '거짓'이면 '참'을 리턴
sum(<node-set>)	각 노드를 숫자로 변환하여 합을 계산

```
user[last()]/email[contains(., 'pymail')]
```

파이썬으로 실행하기

- 명령행에서 실행하기
  - 4xpath xml파일 XPath표현식

```
/usr/local/lib/python2.2/site-packages/Ft/Share/Bin/4xpath
$ 4xpath sample04.xml "userlist/user/name"
Node Set:
<cElement at 0x84001ec: name u'name', 0 attributes, 1 children>
<cElement at 0x840b104: name u'name', 0 attributes, 1 children>
<cElement at 0x8425094: name u'name', 0 attributes, 1 children>
<cElement at 0x842563c: name u'name', 0 attributes, 1 children>
$ 4xpath sample04.xml "userlist/user/name/text()"
Node Set:
<cText at 0x841643c>
<cText at 0x840b3ec>
<cText at 0x8425564>
<cText at 0x8425ad4>
$ 4xpath --string sample04.xml "userlist/user/name/text()"
String Results:
gslee
```

파이썬으로 실행하기

- PyXML 혹은 4Suite < 0.12

```
# xpath01.py
# 4XPath from PyXML
# or 4XPath from 4Suite < version 0.12
from xml.xpath import Evaluate
from xml.dom.ext.reader.Sax2 import Reader
from xml.dom.ext import PrettyPrint

reader = Reader()  # Reader 객체를 생성한다.
dom = reader.fromUri('sample04.xml')  # URI를 지정한다.

xpath0 = 'user/name'
nodeList = Evaluate(xpath0, dom.documentElement)
for node in nodeList:
        PrettyPrint(node)
```

파이썬으로 실행하기

- 실행 결과

```
<email>gslee@mail.kw.ac.kr</email>

<email>spam@mail.kw.ac.kr</email>

<email>home@mail.kw.ac.kr</email>

<email>sunny@mail.kw.ac.kr</email>
```

파이썬으로 실행하기

- 4Suite >= 0.12

```
# xpath02.py
# 4XPath from 4Suite >= version 0.12
from Ft.Xml.XPath import Evaluate
from xml.dom.ext.reader.Sax2 import Reader
from xml.dom.ext import PrettyPrint

reader = Reader()  # Reader 객체를 생성한다.
dom = reader.fromUri('sample04.xml')  # URI를 지정한다.

xpath0 = 'user/email'
nodeList = Evaluate(xpath0, dom.documentElement)
for node in nodeList:
        PrettyPrint(node)
```

컴파일해서 이용하기

- PyXML 혹은 4Suite < 0.12

```
# xpath03.py
# 4XPath from 4Suite >= version 0.12
from xml.xpath import Compile
from xml.xpath.Context import Context
from xml.dom.ext.reader.Sax2 import Reader
from xml.dom.ext import PrettyPrint

reader = Reader()  # Reader 객체를 생성한다.
dom = reader.fromUri('sample04.xml')  # URI를 지정한다.

expression = Compile('user/email')
context = Context(dom.documentElement)
nodeList = expression.evaluate(context)
for node in nodeList:
        PrettyPrint(node)
```

컴파일해서 이용하기

- **4Suite >= 0.12**

```
# xpath04.py
# 4XPath from 4Suite >= version 0.12
from Ft.Xml.XPath import Compile
from Ft.Xml.XPath.Context import Context
from xml.dom.ext.reader.Sax2 import Reader
from xml.dom.ext import PrettyPrint

reader = Reader()  # Reader 객체를 생성한다.
dom = reader.fromUri('sample04.xml')  # URI를
지정한다.

expression = Compile('user/email')
context = Context(dom.documentElement)
nodeList = expression.evaluate(context)
for node in nodeList:
        PrettyPrint(node)
```