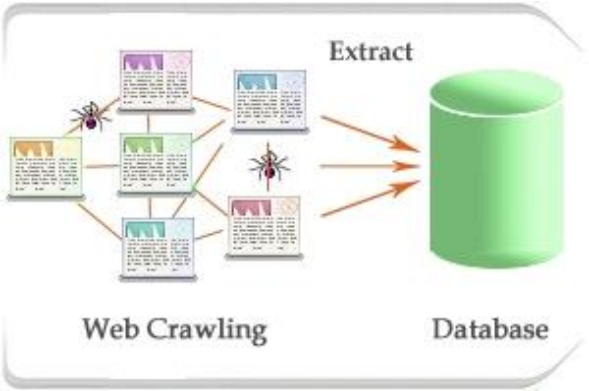
# 비정형 데이터 수집

삼성전자공과대학교 3학년 3학기

# Web Crawling

❑ Web Crawler를 이용하여 웹사이트를 탐색해 원하는 정보를 가져오는 작업

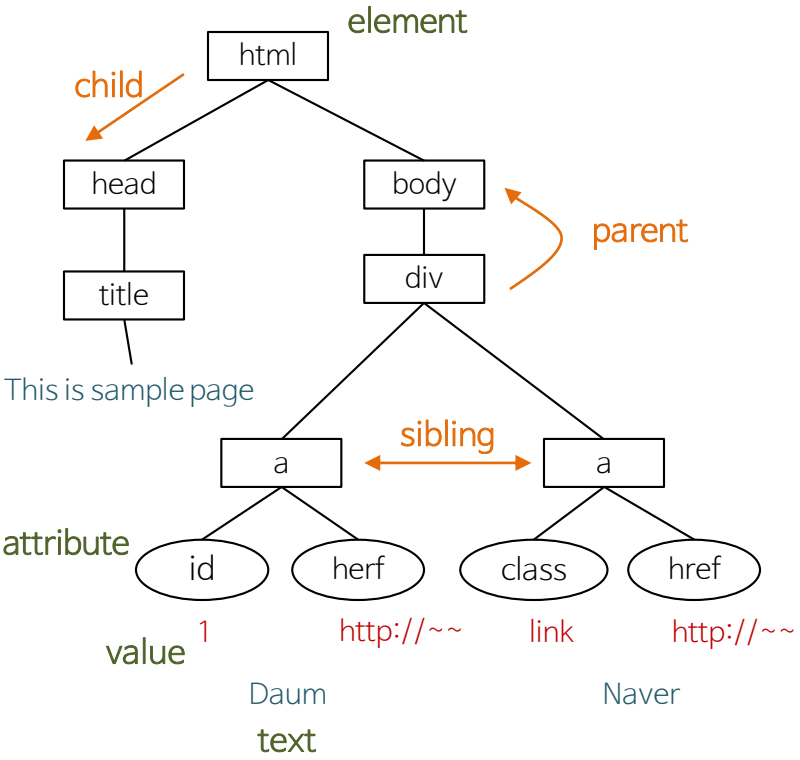



❑ Crawling : 웹사이트에서 링크를 따라가며 원하는 정보를 가져오고, 가공

❑ Scraping : 로컬 머신, 데이터베이스, 인터넷등의 소스로부터 정보 추출

# HTML Document Model

```
〈html〉
  〈head〉
    〈title〉This is sample page〈/title〉
  〈head〉
  〈body〉
  〈div〉
    〈a id="1" href="http://www.daum.net"〉Daum〈/a〉
    〈a class="link" href="http://www.naver.net"〉Naver〈/a〉
  〈div〉
  〈/body〉
〈/html〉
```



# 기본 개념

- WWW, HyperText, URL
- request / response
- response code : 200, 404, 500 (https://ko.wikipedia.org/wiki/HTTP_상태코드)
- HTTP and HTTP Commands (GET, POST)
- HTTP 헤더
- 사용자 에이전트
- 쿠키
- HTML 태그: table, tr, td, div, a href
- XHR (XMLHttpRequest, =Ajax)

- pip install beautifulsoup4
- pip install lxml
- pip install html5lib

# BeautifulSoup4

```
from bs4 import BeautifulSoup as bs
import requests

url = 'http://~~~~~~'
html = requests.get(url)

soup = bs(html.text)
```

| Parser | 문법 | 장점 | 단점 |
|---|---|---|---|
| html.parser | BeautifulSoup(markup, "html.parser") | •각종 기능 완비<br>•적절한 속도<br>•관대함 (파이썬 2.7.3과 3.2에서.) | |
| lxml HTML | BeautifulSoup(markup, "lxml") | •아주 빠름 | •외부 C 라이브러리 의존 |
| lxml XML | BeautifulSoup(markup, ["lxml", "xml"]) BeautifulSoup(markup,"xml") | •아주 빠름<br>•XML 지원 | •외부 C 라이브러리 의존 |
| html5lib | BeautifulSoup(markup, html5lib) | •웹 브라우저의 방식으로 페이지를 해석함<br>•유효한 HTML5를 생성함 | •아주 느림<br>•외부 파이썬 라이브러리 의존<br>•파이썬 2 전용 |

# XPath

■ http://www.w3schools.com/xml/xpath_intro.asp

| 표현식 | 설명 |
|---|---|
| bookstore | 현재 위치에서 모든 bookstore 요소 선택 |
| /bookstore | 루트로 부터 bookstore 요소 선택 |
| //bookstore | 현재 문서의 모든 bookstore 요소에서 검색 |
| . | 현재 요소선택 |
| .. | 부모 요소선택 |
| //@lang | 현재 문서에서 모든 lang 속성 선택 |
| /bookstore/book[1] | bookstore노드의 자식 요소중 첫번째 book 요소 선택 |
| /bookstore/book[last()] | bookstore노드의 자식 요소중 마지막 book 요소 선택 |
| /bookstore/book[last()-1] | bookstore노드의 자식 요소중 마지막 하나 전 book 요소 선택 |

# XPath

| 표현식 | 설명 |
| --- | --- |
| /bookstore/book[position()<3] | bookstore요소의 자식 요소중 처음 두개의 book 요소 선택 |
| //title[@lang] | lang 속성을 가지고 있는 모든 title 요소 선택 |
| //title[@lang='en'] | lang 속성값이 "en"인 모든 title 요소 선택 |
| /bookstore/book[price>35.00] | bookstore요소의 자식 요소중 price 요소의 값이 35보다 큰 자식 요소를 가진 book 요소 선택 |
| /bookstore/book[price>35.00]/title | bookstore요소의 자식 요소중 price 요소의 값이 35보다 큰 자식 요소를 가진 book 요소의 자식 title 요소 선택 |
| /bookstore/* | bookstore의 모든 자식 요소 선택 |
| //title[@*] | 적어도 하나의 속성을 가지고 있는 모든 title 요소 선택 |

# CSS Selector

■ http://www.w3schools.com/cssref/css_selectors.asp

| Pattern | Matches | 예제 |
|---|---|---|
| * | 모든 요소 | * |
| tag | 태그 이름과 일치하는 요소 | div |
| #id | ID 속성이 "id"인 요소 | div#wrap, #logo |
| .class | CSS 클래스가 "class"인 요소 | div.left, .result |
| [attr] | "attr" 이란 이름의 속성을 가진 요소 | a[href] |
| [^attrPrepix] | 속성이름이 "attrPrefix"로 시작하는 속성을 가진 요소 | [^data-], div[^data-] |
| [attr=val] | "attr"이란 이름의 속성값이 "val"인 요소 | Img[width=500] |
| [attr^=valPrefix] | "attr"이란 이름의 속성값이 "valPrefix"로 시작하는 요소 | a[href^=http:] |
| [attr$=valSuffix] | "attr"이란 이름의 속성값이 "valSuffix"로 끝나는 요소 | img[src$=.png] |

# CSS Selector

| Pattern | Matches | 예제 |
|---|---|---|
| [attr*=some] | “attr”이란 이름의 속성값에 “some”을 포함하는 요소 | a[href*=/search/] |
| [attr~=regex] | “attr”이란 이름의 속성값이 regexp에 부합하는 요소 | img[src~=(?i)₩₩.(png\|jpe?g)] |
| E F | E의 자손 요소 F | div a, .log h1 |
| E 〉F | E의 자식(바로 하위)요소 F | ol > li |
| E + F | E의 바로 다음 형제 요소 F | li + li , div.head + div |
| E ~ F | E의 바로 전 형제 요소 F | h1 ~ p |
| E, F, G | E, F, G 요소 | a[href], div, h3 |
| :lt(n) | N번째 보다 앞의 형제 요소 | td:lt(3) => 0, 1, 2 |
| :gt(n) | N번째 이후의 형제 요소 | td:lt(1) => 2, 3, 4 |

# CSS Selector

| Pattern | Matches | 예제 |
|---|---|---|
| :eq(n) | n번째 형제 요소 | td:eq(0) |
| :has(selector) | selector에 맞는 요소를 가진 요소 | div:has(p) |
| :not(selector) | selector에 맞는 요소를 가지지 않은 요소 | div:not(.logo) |
| :contain(text) | text가 "text"를 포함하고 있는 요소를 가진 요소 (대소문자 구분 없음) | p:contains(jsoup) |
| :matches(regex) | text가 regexp에 맞는 요소를 가진 요소 | td:matches(₩₩d+) |
| :containsOwn(text) | 자신의 text가 "text"를 포함하고 있는 요소 | p:containsOwn(jsoup) |
| :matchesOwn(regex) | 자신의 text가 regexp에 맞는 요소 | td:matchesOwn(₩₩d+) |

# XPath vs. CSS Selector

| 대상 | CSS Selector | XPath |
|---|---|---|
| All Elements | * | //* |
| All P Elements | p | //p |
| All Child Elements | p > * | //p/* |
| Element By ID | #foo | //*[@id='foo'] |
| Element By Class | .foo | //*[contains(@class,'foo')] |
| Element With Attribute | *[title] | //*[@title] |
| First Child of All P | p > *:first-child | //p/*[0] |

https://opentutorials.org/course/580

# Selenium

- pip install selenium
- https://sites.google.com/a/chromium.org/chromedriver/downloads
- http://phantomjs.org/download.html