7. Python

7.1 웹 크롤링(Web crawling) – Beautiful Soup, parser

- 뷰리플습(Beautiful Soup)은 스크린 스크래핑(screen-scraping) 프로젝트를 위해 설계된 파이썬 라이브러리
- 구문 분석, 트리 탐색, 검색 및 수정을 위한 몇 가지 간단한 방법과 파이썬 관 용구를 제공하며 문서를 분석하고 필요한 것을 추출하는 도구
- 들어오는 문서를 유니코드로 보내고 문서를 UTF-8로 자동 변환
- 뷰리플습은 lxml 및 html5lib과 같은 파이썬 파서 라이브러리를 사용할 수 있다.

7.2 파서(parser) 라이브러리

파서	사용법	장점	단점
파이썬의 html.parser	BeautifulSoup(markup, "html.parser")	• 보통 속도 • 파이썬 2.7.3, 3.2.2 이상 버전에서 호환됨	• 파이썬 2.7.3 또는 3.2.2 이전버전에서 호환되지 않음
Lxml's HTML parser	BeautifulSoup(markup, "lxml")	• 매우 빠름 • 호환성	• 외부 C에 의존함
Lxml's XML parser	BeautifulSoup(markup, "lxml-xml") BeautifulSoup(markup, "xml")	• 매우 빠름	• XML 파서만 지원 • 외부 C에 의존함

7.3 Selector API

- BeautifulSoup은 가장 일반적으로 사용되는 CSS 선택자를 지원
- BeautifulSoup의 Selector API는 select()와 select_one(), find() 등
- select()와 select_one() 메서드만 알아도 원하는 요소를 찾기에 충분
- soup.select("CSS 선택자") : CSS 선택자에 해당하는 모든 요소를 반환
- soup.select_one("CSS 선택자") : CSS 선택자에 맞는 오직 첫 번째 태 그 요소만 반환

7.4 CSS 선택자 종류

1) 태그 선택자 ("element")

- 태그 선택자는 일반적으로 스타일 정의하고 싶은 html 태그 이름을 사용
- 요소 안의 텍스트는 text, 태그이름은 name 그리고 태그의 속성들은 attrs를 이 용해 조회
- soup.select("h1")

2) 다중(그룹) 선택자 ("selector1, selector2, selectorN")

- 선택자를 ","(comma)로 분리하여 선언하면 여러 개 선택자 적용.
- 해당하는 모든 선택자의 요소를 찾기 때문에 select_one()아닌 select() 메서드를 이용
- soup.select("h1, p")

3) 내포 선택자 ("ancestor descendant")

- 요소가 내포 관계가 있을 때 적용시키기 위한 선택자.
- 선택자와 선택자 사이를 공백으로 띄우고 나열
- soup.select("div b")

4) 자식 선택자 ("parent > child")

- 선택자와 선택자 사이에 >를 입력하며 반드시 부모자식간의 관계에만 스타일이 적용되도록 함.
- 두 단계 이상 건너뛴 관계에서는 자식 선택자가 작동하지 않음
- soup.select("div > b")

5) 클래스(class) 선택자 (".class")

- HTML 문서에서 class 속성의 값과 일치하는 요소를 선택
- 선택자 이름 앞에 "."을 이용하여 선언.
- soup.select("div.contents")

6) 아이디(id) 선택자 (" #id ")

- HTML 문서에서 id 속성의 값과 일치하는 요소를 선택
- id 선택자는 #으로 정의합니다.
- soup.select_one("#subject")

7) 속성 선택자 [name="value"]

- 특정한 속성을 갖는 요소만 선택.
- 속성 선택자는 [와]사이에 속성의 이름과 값을 지정
- soup.select_one("[id=subject]")

7.5 DOM(Document Object Model, 문서 객체 모델)

- HTML 문서를 파싱해서 만들어진 객체
- HTML 문서에서 원하는 요소를 찾기 위해서는 요소를 찾는 메소드를 실행시키기 전에 먼저 DOM이 만들어져 있어야 함