

Python Web Crawling

by BeautifulSoup4

```
types.Operator):  
    on X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

```
context):  
    context.active_object is not
```

BeautifulSoup 소개

▶ BeautifulSoup?

- HTML 및 XML 문서를 파싱(parsing)하기 위한 파이썬 패키지.
- 문서에서 특정 태그 정보를 추출해 처리할 수 있도록 만들어줌.

▶ BeautifulSoup 설치

- (커맨드 창에서) `python -m pip install beautifulsoup4`
- 'beautifulsoup'를 설치할 경우 버전 3이 설치됨.

▶ BeautifulSoup 불러오기

- `from bs4 import BeautifulSoup`

BeautifulSoup 객체 생성

▶ HTML 파일로 생성

- HTML 문서 파일로 직접 생성하는 방법.

```
from bs4 import BeautifulSoup as bs

with open("HTML파일.html") as f:
    soup = bs(f, "html.parser")
```

▶ urllib 모듈로 생성

- URL 처리를 위한 표준 라이브러리인 urllib 모듈을 사용.

```
from bs4 import BeautifulSoup as bs
import urllib.request as req

with req.urlopen(URL 문자열) as res:
    html = res.read()
    soup = bs(html, "html.parser")
```

BeautifulSoup 객체 생성

▶ requests 패키지로 생성

- 파이썬 HTTP 라이브러리인 requests를 사용.
- 먼저 pip 등을 이용해 requests 패키지를 설치해야 함.
- HTTP 요청을 보내고 response 객체로 응답받음.

```
import requests as req

res = req.get(URL 문자열)
html = res.text
soup = bs(html, "html.parser")
```

▶ find() 함수

- 특정 태그를 찾아서 반환하는 함수.
- 태그에 더해 특정 속성을 지정 가능.
- 결과가 중복될 경우 가장 처음 요소 하나만 가져옴.

```
soup.find("태그명", attrs={"속성명": "속성값", ...})
```

```
from bs4 import BeautifulSoup as bs
import requests as req
```

```
res = req.get("https://search.naver.com/search.naver?query=날씨")
html = res.text
soup = bs(html, "html.parser")
```

```
data1 = soup.find('div')
print(data1)
```

```
data1 = soup.find('div', attrs = {'class': 'main_info'})
print(data1)
```

- ▶ `find_all()`, `findAll()` 함수
 - 특정 태그들을 찾아서 반환하는 함수.
 - 태그에 더해 특정 속성을 지정 가능.
 - 여러 개의 결과를 리스트로 묶어서 반환.

```
soup.find_all("태그명", attrs={"속성명": "속성값", ...})
```

```
...  
data1 = soup.find('div', attrs = {'class': 'main_info'})  
print(data1)
```

```
data2 = data1.find_all("li")  
print(data2)
```

- ▶ `find()`, `find_all()` 함수의 특징
 - `bs4.element.Tag` 객체로 반환.
 - 검색 결과에 대해 다시 검색 가능
 - `text` 요소를 호출하여 본문 읽기 가능

```
data2 = data1.findAll("li")  
print(data2)
```

```
for data in data2:  
    print(data.text)
```

CSS 선택자 문법으로 찾기

▶ select() 함수

- 특정 태그를 찾아서 반환하는 함수.
- CSS 선택자 문법을 따름.
- 여러 개의 결과를 리스트로 묶어서 반환.

```
soup.select("CSS 선택자 문법")
```


속성값 가져오기

▶ 속성값 가져오기

```
Tag객체.get("속성명")
```

```
Tag객체["속성명"]
```

크롤링 중 접속 차단 발생 시

▶ 접속 차단 이유 및 해결 방법

- 접속자가 사용자가 아닌 컴퓨터의 경우 의심되는 행동을 막기 위해 차단하는 사이트가 존재함.
- headers 인수로 User-Agent를 명시하여 해결함.

```
res = req.get("https://www.melon.com/chart/index.htm", headers={"User-Agent": 코드})
```

▶ User-Agent?

- 접속자가 실제 사용자임을 증명하는 요소.
- <http://www.useragentstring.com/index.php>