

# 13장. 웹 스크래이핑



**웹 스크레이핑 = 웹 크롤링**

# 📄 Scraping°/쿠팡?

## 인터넷에 있는 웹 페이지를 방문해서 자료를 수집하는 일.

## 웹 크롤링이라고도 한다.

## ▶ urllib 모듈 사용

```
import urllib.request
```

```
from urllib import request
```

```
from urllib import request
```

```
url = "https://www.naver.com/"
contents = request.urlopen(url)
print(contents.read())
```

[illegible]

# 웹 스크레이핑 = 웹 크롤링

## BeautifulSoup 모듈

HTML과 XML 문서를 파싱하기 위한 파이썬 라이브러리이다.

웹 스크레이핑에 유용한 HTML에서 데이터를 추출하는 데 사용함.

### ▶ BeautifulSoup 설치

**pip install BeautifulSoup4**

## Beautiful Soup Documentation

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.



# 웹 스크레이핑 = 웹 크롤링

- `find()`는 처음 나오는 태그 요소를 찾아옴

```
from bs4 import BeautifulSoup

html_str = '''
<html>
  <body>
    <ul class = 'item'>
      <li>인공지능</li>
      <li>Big Data</li>
      <li>로봇</li>
    </ul>
    <ul class = 'comlang'>
      <li>Python</li>
      <li>C/C#</li>
      <li>Java</li>
    </ul>
  </body>
</html>
'''
```

```
html = BeautifulSoup(html_str, "html.parser")
#print(html)
first_ul = html.find('ul')
print(first_ul)
print(first_ul.text)
```

```
<ul class="item">
<li>인공지능</li>
<li>Big Data</li>
<li>로봇</li>
</ul>
```

인공지능  
Big Data  
로봇

# 웹 스크레이핑 = 웹 크롤링

- `findAll()` 은 모든 태그 요소를 찾아서 리스트로 반환함

```
html = BeautifulSoup(html_str, "html.parser")
first_ul = html.find('ul', {'class': 'item'})
all_li = first_ul.findAll('li')
print(all_li)
print(all_li[1])
print(all_li[1].text)

for li in all_li[1:2]:
    print(li.text)
```

Dictionary 자료구조  
{키 : 값}

```
[<li>인공지능</li>, <li>빅데이터</li>, <li>로봇</li>, <li>Python</li>, <li>C/C#</li>,
<li>빅데이터</li>
빅데이터
빅데이터
```

# 네이버에서 웹 크롤링하기

## ✓ Naver에서 필요한 정보 추출하기



```
▼<div class="group_flex">
  ▶<div class="logo_area"> </div>
  ▼<div class="service_area">
    ▼<a id="NM_set_home_btn" href="https
      "네이버를 시작페이지로"
      ::after
    </a>
```

```
from urllib import request
from bs4 import BeautifulSoup

url = request.urlopen("http://www.naver.com")
html = BeautifulSoup(url, 'html.parser')

div = html.find('div', {'class': 'service_area'})
first_a = div.find('a')
print(first_a)
print(first_a.text)
```

# 네이버에서 웹 크롤링하기

## 실습 문제

---

네이버 시작 페이지의 우측 상단의 링크 중에서 '주니어네이버'를 추출하세요.  
(파일이름 : naver\_begin\_a.py)

---

👉 실행 결과  
주니어네이버

```
from urllib import request
from bs4 import BeautifulSoup

url = request.urlopen("http://www.naver.com")
html = BeautifulSoup(url, 'html.parser')

div = html.find('div', {'class': 'service_area'})
all_a = div.findAll('a')
print(all_a[1])
print(all_a[1].text)
```

# 네이버에서 웹 크롤링하기

## ✓ Naver 메뉴 가져오기



```
▼<div class="group_nav">
  ▼<ul class="list_nav type_fix"> == $0
    ▼<li class="nav_item">
      ▼<a href="https://mail.naver.com/" class="nav"
        <i class="ico_mail"></i>
        "메일"
      </a>
```

```
from urllib import request
from bs4 import BeautifulSoup

url = request.urlopen("https://www.naver.com/")
html = BeautifulSoup(url, 'html.parser')
ul = html.find('ul', {'class': 'list_nav type_fix'})
# print(ul)
lis = ul.findAll('li')
print(lis)
for li in lis:
    a = li.find('a')
    print(a.text)

# 직접 a 태그로 접근
all_a = ul.findAll('a')
for a in all_a:
    print(a.text)

# '카페' 접근
print(all_a[1].text)
```



# 네이버 금융 크롤링하기

## ✓ 환율정보 수집하기 – `select()` 사용하기

`select(개체이름.선택자이름)` – 전체 검색(리스트로 반환)

`select_one(개체이름.선택자이름)` – 1개 검색

```
from urllib import request
from bs4 import BeautifulSoup

# 네이버 증권 > 시장지표 > 환전 고시 환율
url = request.urlopen("https://finance.naver.com/marketindex")
html = BeautifulSoup(url, "html.parser")
lis = html.select("div.market1 ul li")
for li in lis:
    exchange = li.select_one("span.blind")
    value = li.select_one("span.value")
    #print(exchange.string, value.string)
    print(exchange.string.split(' ')[-1], ': ', value.string)
```

USD : 1,168.00  
JPY(100엔) : 1,069.06  
EUR : 1,381.57  
CNY : 181.72

# 네이버 금융 크롤링하기

## ✓ 환율정보 수집하기 – find() 사용하기

```
from urllib import request
from bs4 import BeautifulSoup

# 네이버 증권 > 시장지표 > 환전 고시 환율
url = request.urlopen("https://finance.naver.com/marketindex")
html = BeautifulSoup(url, "html.parser")
ul = html.find('div', {'class': 'market1'})
lis = ul.findAll('li')

for li in lis:
    exchange = li.find('span', {'class': 'blind'}) # 환율 종류
    value = li.find('span', {'class': 'value'}) # 환율 지수
    #print(exchange.string, ': ', value.string)
    print(exchange.string.split(' ')[-1], ': ', value.string)
```

```
USD : 1,168.00
JPY(100엔) : 1,069.06
EUR : 1,381.57
CNY : 181.72
```

# 네이버 금융 크롤링하기

## ◆ 삼성전자 주식 가격 가져오기

인기 검색 종목			더보기
1. 카카오게임즈	71,600	▲ 14,100	
2. 삼성전자	80,000	▼ 100	
3. 진원생명과학	49,850	▲ 3,200	
4. 카카오	159,500	▼ 500	
5. 이연제약	58,300	▲ 9,650	

네이버 > 금융 홈 > 우측

finance.naver.com/item/main.nhn?code=005930

**NAVER 금융** 종목명·지  
금융 홈 **국내증시** 해외  
**삼성전자** 005930 코스피  
**80,000**  
전일대비 ▼100 | -0.12%  
선차트 1일 | 1주일 | 3개월

# 네이버 금융 크롤링하기

## ◆ 주식정보 – 단일 주식 종목 찾아 오기

```
def getcontent():  
    url = "https://finance.naver.com/item/main.nhn?code=005930"  
    html = request.urlopen(url)  
    contents = BeautifulSoup(html, 'html.parser')  
    return contents  
  
contents = getcontent()  
no_today = contents.find('p', {'class': 'no_today'})  
print(no_today)  
price = no_today.find("span", {'class': 'blind'})  
print("삼성전자 주가 : {}원".format(price.text))
```

```
<p class="no_today">  
  <em class="no_down">  
    <span class="blind">80,100</span>  
    <span class="no8">8</span><span class="no0">0</span>  
  </em>  
</p>  
삼성전자 주가 : 80,100원
```

웹에서는 보이지 않음

# 네이버 금융 크롤링하기

## ◆ 주식정보 찾기 – 여러 종목 가격 가져오기

```
def getcontent(item_code):  
    url = "https://finance.naver.com/item/main.nhn?code=" + item_code  
    html = request.urlopen(url)  
    content = BeautifulSoup(html, 'html.parser')  
    return content  
  
def get_price(item_code):  
    content = getcontent(item_code)  
    # no_today = content.find('p', {'class': 'no_today'})  
    no_today = content.select_one("p.no_today")  
    # price = no_today.find('span', {'class': 'blind'})  
    price = no_today.select_one("span.blind")  
    return price
```

# 네이버 금융 크롤링하기

## ◆ 주식정보 찾기 – 여러 종목 가격 가져오기

```
삼성전자 = get_price("005930")  
네이버 = get_price("035420")  
카카오 = get_price("035720")  
print("삼성전자 주가 : {}원".format(삼성전자.text))  
print("네이버 주가 : {}원".format(네이버.text))  
print("카카오 주가 : {}원".format(카카오.text))
```

```
삼성전자 주가 : 80,000원  
네이버 주가 : 414,000원  
카카오 주가 : 159,500원
```