



Chapter

1

크롤링과 스크레이핑

1. 크롤링, 스크레이핑
2. 고급 스크레이핑
3. 데이터 소스의 서식과 가공

학습 목표

- ✓ Python 웹 스크레이핑을 이해한다.
- ✓ HTML 태그와 CSS를 이해한다.
- ✓ 웹 상에서 사용되는 자료 유형을 Python 에서 다룬다.

주요 내용

- ✓ HTML, CSS
- ✓ 웹 스크레이핑
- ✓ 텍스트 데이터, 바이너리 데이터

1

크롤링, 스크레이핑



빅데이터 수집

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 인터넷의 빅데이터

- 빅데이터는 대규모의 데이터의 집합을 의미
- 데이터의 수집 만으로는 어떤 의미도 없으며, 데이터를 활용했을 때 가치를 부여
- 즉, 빅데이터는 자료의 수집부터 분석을 통해 비즈니스에 활용하는 것까지를 의미
- 빅데이터의 분석은 수 많은 데이터에서 어떤 규칙을 찾는 것
- 최근에 빅데이터의 인기는 인터넷과 스마트 폰등을 이용한 실시간 자료의 수집이 가능해졌기 때문
 - 블로그와 SNS를 이용한 트렌드 분석
 - 인터넷 전자상거래 상품 데이터베이스 분석
 - 금융 정보를 이용한 예측
 - 공공데이터를 이용한 인구, 미세먼지 등 데이터 분석

빅데이터 수집

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 스크레이핑(scraping)

- 웹 사이트의 특정 정보를 추출하는 기술
- 공개된 정보는 대부분 HTML 형식으로 되어 있어 들 중에서 필요한 데이터로 저장하기 위해 데이터 가공이 필요
- 이를 위해서 데이터 구조를 파악하는 것이 필요
- 최근에는 로그인을 통해서만 유용한 정보에 접근 할 수 있는 경우가 많아 로그인 이후 필요한 웹 페이지 접근 기술이 필요

❖ 크롤링(crawling)

- 웹 사이트를 프로그램이 정기적으로 정보를 추출하는 기술
 - 크롤링을 하는 프로그램을 크롤러(crawler) 또는 스파이더(spider)

❖ 머신러닝에 사용할 수 있는 데이터 구조

- 수집된 자료는 데이터의 구조를 분석하고 필요한 부분만 추출하여 과정을 통해 머신러닝에 사용 가능
- 데이터 형식은 파일 또는 데이터 베이스에 저장하여 활용

빅데이터 수집

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 웹 컴포넌트: HTML과 HTTP : HTML

- 태그(tag)는 꺾쇠 괄호 < >로 둘러싸여 있고, 그 안에 정보에 대한 의미를 작성
- 그 의미가 끝나는 부분에 슬래시(/)를 사용하여 해당 태그를 종료

```
<title> Hello, World </title>
```

제목 요소, 값은 Hello, World

- HTTP(Hypertext Transaction Protocol)는 인터넷에서 컴퓨터 간에 정보를 주고받을 때 사용하는 일종의 약속
 - 일반적으로 컴퓨터 과학에서는 이러한 약속을 프로토콜(protocol)

❖ 웹의 동작 순서

- 웹에 있는 정보를 보기 위해 먼저 하는 일은 웹 브라우저를 시작하고, 거기에 주소 정보를 입력하는 것
- 주소 정보의 공식 이름은 URL(Uniform Resource Locator)

http://www.domain.com:1234/path/to/resource?a=b&x=y

↑
protocol

↑
host

↑
port

↑
resource path

↑
query

[URL의 구조]

빅데이터 수집

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- URL에는 해당 서버가 위치한 인터넷 주소 정보인 도메인 네임(domain name)이 존재
- 흔히 도메인 정보 또는 서버 주소라고도 하는 이 주소를 통해 웹의 정보를 제공하는 서버에 접속
- 일반적으로 컴퓨터는 인터넷 프로토콜 주소(Internet Protocol address), 즉 IP 주소(IP address)라고 부르는 주소 값을 가짐
- IP 주소를 컴퓨터의 주소로 생각하면 이 주소에 접속하기 위해 사용하는 도메인 네임과 연결하기 위한 도메인 네임 서버(Domain Name Server, DNS)를 운영



I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 웹상의 정보를 추출하는 방법

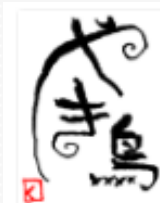
- 웹 사이트에 있는 데이터 추출을 위해 urllib 라이브러리를 사용
 - HTTP 또는 FTP를 이용해 데이터를 다운로드
- urllib는 URL을 다루는 모듈을 모아 놓은 패키지
 - 특히 urllib.request 모듈은 웹 사이트에 있는 데이터에 접근하는 기능을 제공

함수	설명
<code>urlretrieve(url, name)</code>	URL 주소의 파일을 다운로드
<code>urlopen()</code>	곧바로 파일을 저장하지 않고 메모리상에 load

I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [1]: import urllib.request # 패키지 실행
url="http://uta.pw/shodou/img/28/214.png" # 파일을 다운로드 할 주소
savename="test_download.png" # 저장할 파일 이름
urllib.request.urlretrieve(url,savename) # 파일 다운로드
```



test_download

```
Out[1]: ('test_download.png', <http.client.HTTPMessage at 0x2359343f2b0>)
```

```
In [2]: # 메모리 상에 load 후 바이너리 파일로 변환하여 파일을 저장
url="http://uta.pw/shodou/img/28/214.png" # 파일을 다운로드 할 주소
savename="open_download.png" # 저장할 파일 이름
# 파일 다운로드
memory=urllib.request.urlopen(url).read() # URL 리소스를 열로 read 메소드 데이터 읽기
# 파일로 저장
with open(savename, mode="wb") as f: # w 쓰기 모드, b 바이너리 모드
    f.write(memory) # 메소드로 다운로드한 바이너리 데이터를 파일로 저장
```



open_download

```
In [3]: # 공공데이터 포털
# https://www.data.go.kr/dataset/15029863/fileData.do;jsessionid=MVBP95oBrG7TnP4UuntMZ
url="https://www.data.go.kr/dataset/fileDownload.do?atchFileId=FILE_000000001455071&f"
savename="gas_20180620.csv" # 저장할 파일 이름
urllib.request.urlretrieve(url,savename) # 파일 다운로드
```



gas_20180620

```
Out[3]: ('gas_20180620.csv', <http.client.HTTPMessage at 0x2359343f588>)
```


I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 웹에서 데이터 추출하기

- 웹에서 XML 또는 HTML 등의 텍스트 기반 데이터를 다운로드

```
In [4]: # 데이터 읽어 들이기
url="http://api.aoikujira.com/ip/ini" # 데이터를 가져올 할 주소
res=urllib.request.urlopen(url) # URL 리소스를 열기
data=res.read() # 바이너리 데이터로 읽어 들이기

# 바이너리를 문자열로 변환(HTML 소스 불러오기)
text=data.decode("utf-8") # decode 메소드를 이용한 바이너리를 문자열로 변환
print(text) # 문자열로 출력
```

```
[ip]
API_URI=http://api.aoikujira.com/ip/get.php
REMOTE_ADDR=122.128.186.202
REMOTE_HOST=122.128.186.202
REMOTE_PORT=58116
HTTP_HOST=api.aoikujira.com
HTTP_USER_AGENT=Python-urllib/3.7
HTTP_ACCEPT_LANGUAGE=
HTTP_ACCEPT_CHARSET=
SERVER_PORT=80
FORMAT=ini
```

I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [5]: # 데이터 읽어 들이기
url="https://www.fun-coding.org/crawl_basic2.html" # 데이터를 가져올 할 주소
res=urllib.request.urlopen(url) # URL 리소스를 열기
data=res.read() # 바이너리 데이터로 읽어 들이기

# 바이너리를 문자열로 변환(HTML 소스 불러오기)
text=data.decode("utf-8") # decode 메소드를 이용한 바이너리를 문자열로 변환
print(text) # 문자열로 출력
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>웹크롤링 기본: 크롤링(crawling) 이해 및 기본 - 잔재미코딩</title>
  <meta name='title' content='웹크롤링 기본: 크롤링(crawling) 이해 및 기본 -
잔재미코딩'>
  <meta name="description" content="잔재미코딩은 IT 교육 콘텐츠와 강의 전문 연
구소입니다.">
```

데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 매개변수를 추가해 요청을 전송하는 방법

- URL에 매개변수를 추가해 요청을 전송
- 기상청 RSS 서비스

http://www.weather.go.kr/weather/lifenindustry/sevice_rss.jsp

중기예보

중기 예보	전국	108	RSS ▶	전라북도	146	RSS ▶
	서울-경기도	109	RSS ▶	전라남도	156	RSS ▶
	강원도	105	RSS ▶	경상북도	143	RSS ▶
	충청북도	131	RSS ▶	경상남도	159	RSS ▶
	충청남도	133	RSS ▶	제주특별자치도	184	RSS ▶

<http://www.weather.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108>
매개변수

I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [6]: import urllib.request # 패키지 실행
import urllib.parse # 패키지 실행
# 데이터 읽어 들이기
API="http://www.weather.go.kr/weather/forecast/mid-term-rss3.jsp" # 데이터 기본 주소
# 매개변수(딕셔너리 자료형)를 URL 인코딩
value={"stnId":"108"}
params=urllib.parse.urlencode(value) # 매개변수를 URL 인코딩
print(params)

# 요청 URL 생성
url=API+"?" +params # URL 리소스를 열기
print("url={0}".format(url))

# 다운로드
res=urllib.request.urlopen(url) # URL 리소스 열기
data=res.read() # 바이너리 데이터로 읽어 들이기

# 바이너리를 문자열로 변환(HTML 소스 불러오기)
text=data.decode("utf-8") # decode 메소드를 이용한 바이너리를 문자열로 변환
print(text) # 문자열로 출력
```

stnId=108

url=http://www.weather.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108

<?xml version="1.0" encoding="utf-8" ?>

I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- URL 끝부분에 ?를 입력하고 <key>=<value> 형식으로 매개변수 입력
- 여러 개의 매개 변수인 경우 &를 사용하여 구분

http://www.example.com?key1=v1&key2=v2&key3=v3...

❖ 매개변수를 명령줄에서 지정하기

- 앞선 프로그램에서는 매개변수를 코드에서 입력해야 하므로 다른 지역은 매개변수를 지정하려면 프로그램을 수정
- 명령줄에서 바로 지역번호를 입력하여 사용

I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [7]: # 라이브러리 읽어 들이기
import sys
import urllib.request as req
import urllib.parse as parse

# 입력줄 매개변수 추출
text=[] # 문서 저장할 리스트 초기화
while (True) :
    # 입력줄을 이용하여 지역번호 입력
    regionNumber=input("USAGE : download-forecast-argv : ")

    # 반복구분 종료 조건
    if(regionNumber.upper()=="EXIT") :
        break
    elif (int(regionNumber) not in [108,109,105,131,133,146,156,134,159,184]) :
        continue

    # 매개변수를 URL 인코딩(한글을 포함하는 경우 필수로 실행)
    API="http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"
    values={"stnId":regionNumber}
    params=parse.urlencode(values)
    url=API+"?" +params
    print("URL=",url)

    # 페이지 다운로드
    data=req.urlopen(url).read()
    text.append(data.decode("utf-8"))
```

I 데이터 다운로드 하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
USAGE : download-forecast-argv : 108
```

```
URL= http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108
```

```
USAGE : download-forecast-argv : 184
```

```
URL= http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=184
```

```
USAGE : download-forecast-argv : exit
```

- 카카오 톡 도움말(8,9,10)

<https://cs.kakao.com/helps?locale=ko&service=8>

BeautifulSoup로 스크레이핑하기

❖ BeautifulSoup로 스크레이핑하기

- 스크레이핑이란 웹 사이트에서 데이터를 추출하여 원하는 정보를 얻어 내는 것
- 파이썬에서 스크레이핑할 때 빼놓을 수 없는 라이브러리가 BeautifulSoup (beautifulsoup4 패키지 설치)
 - BeautifulSoup을 HTML과 XML에서 정보 추출이 가능
 - HTML과 XML 분석을 해주는 라이브러리
 - 자체로 다운로드 기능은 없음
- HTML 구조로 요소를 추출 **BeautifulSoup() 함수 사용**
 - HTML 구조로 요소를 추출하는 것은 HTML 구조를 하나하나 적어나가는 것은 매우 복잡
 - 간단하게 요소를 찾아내는 방법이 필요

markup parser	설명
html.parser	기본옵션으로 빠르지만 유연하지 못함(단순한 html 문서에서 사용)
lxml	매우 빠르며 유연
xml	XML 파일에만 사용
html5lib	매우 느리지만 유연(구조가 복잡한 HTML 문서에 사용)

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재민 학습 자료로만 사용가능 합니다.

```
In [1]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup

#분석할 HTML
html = '''
<html><body>
<h1>스크레이핑이란?</h1>
<p>웹 페이지를 분석하는 것</p>
<p>원하는 부분을 추출하는 것</p>
</body></html>
'''

# HTML 분석하기
soup=BeautifulSoup(html, "html.parser") # BeautifulSoup 인스턴스 생성

# 원하는 부분 추출
h1=soup.html.body.h1
p1=soup.html.body.p
p2=p1.next_sibling.next_sibling

# 요소의 글자 출력
print("h1 = {}".format(h1.string))
print("p1 = {}".format(p1.string))
print("p2 = {}".format(p2.string))
```

h1 = 스크레이핑이란?

p1 = 웹 페이지를 분석하는 것

p2 = 원하는 부분을 추출하는 것

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재능 학습 자료로만 사용가능 합니다.

❖ id로 요소를 찾는 방법

- id 속성을 지정하여 요소를 찾는 find() 메소드를 제공

find(tag_name, attrs={}) 메소드를 사용

```
In [9]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup

# 파일 열기
fp=open("C:/Users/datam_000/Documents/Python/Module04/Ch01/HTML_Exam.html", "r", encoding='utf-8')

soup=BeautifulSoup(fp, 'html.parser') # BeautifulSoup 인스턴스 생성
divs=soup.find("div") # div 태그
divs_class=soup.find("div", class_="ex_class") # div 태그, class
divs_id=soup.find("div", id="ex_id") # div 태그, id=ex_id
p=soup.find("p") # p 태그
print("*** div 태그\n{0}".format(divs))
print("*** div 태그, class=ex_class\n{0}".format(divs_class))
print("*** div 태그, id=ex_id\n{0}".format(divs_id))
print("*** p 태그\n{0}".format(p))

fp.close() # 파일 닫기
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
*** div 태그
<div>
<p>a</p>
<p>b</p>
<p>c</p>
</div>
*** div 태그, class=ex_class
<div class="ex_class">
<p>d</p>
<p>e</p>
<p>f</p>
</div>
*** div 태그, id=ex_id
<div id="ex_id">
<p>g</p>
<p>h</p>
<p>i</p>
</div>
*** p 태그
<p>a</p>
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 여러 개의 요소 추출하기

find_all(tag_name, attrs={}) 메소드를 사용

```
In [10]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup

# 파일 열기
fp=open("C:/Users/datam_000/Documents/Python/Module04/Ch01/HTML_Exam.html", "r",
        encoding='UTF8')

soup=BeautifulSoup(fp, 'html.parser') # BeautifulSoup 인스턴스 생성
divs=soup.find_all("div") # 모든 div 태그
print("*** div 태그#\n{0}".format(divs))

fp.close() # 파일 닫기
```

```
*** div 태그
[<div>
<p>a</p>
<p>b</p>
<p>c</p>
</div>, <div class="ex_class">
<p>d</p>
<p>e</p>
<p>f</p>
</div>, <div id="ex_id">
<p>g</p>
<p>h</p>
<p>i</p>
</div>]
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
*** div 태그  
[<div>  
<p>a</p>  
<p>b</p>  
<p>c</p>  
</div>, <div class="ex_class">  
<p>d</p>  
<p>e</p>  
<p>f</p>  
</div>, <div id="ex_id">  
<p>g</p>  
<p>h</p>  
<p>i</p>  
</div>]
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

■ 온라인 파일 열기

```
In [11]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup
from urllib.request import urlopen

# 온라인 파일 열기
soup=BeautifulSoup(urlopen("http://www.naver.com"), 'html.parser') # BeautifulSoup 인스턴스
a=soup.find_all("a", class_="ah_da") # 모든 a 태그, class="ah_da"
print("*** a 태그, class=\\\"ah_da\\\"\\n{0}\".format(a))

*** a 태그, class="ah_da"
[<a class="ah_da" data-clk="lve.kwdhistory" href="http://datalab.naver.com/keyword/re
altimeDetail.naver?datetime=2019-06-13T00:10:00&query=%EB%B9%84%EC%95%84%EC%9D%B4
&where=main">
<span class="blind">데이터랩 그래프 보기</span>
<span class="ah_ico_datagraph"></span>
</a>, <a class="ah_da" data-clk="lve.kwdhistory" href="http://datalab.naver.com/keywo
:
:
:
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재학 학생 자료로만 사용가능 합니다.

```
In [12]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup
import urllib.request as req

# 온라인 파일 열기
URL="http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"
soup=BeautifulSoup(req.urlopen(URL), 'html.parser') # BeautifulSoup 인스턴스 생성
title=soup.find("title").string #
wf=soup.find("wf").string #
print("Title#\n{0}".format(title))
print("wf#\n{0}".format(wf))
```

Title

기상청 육상 중기예보

wf

기압골의 영향으로 15일은 강원도와 충북, 경상도, 18일은 중부지방, 19일은 제주도를 제외
한 전국에 비가 오겠고, 그 밖의 날은 고기압의 영향으로 맑은 날이 많겠습니다.
기온
은 평년(최저기온: 14~20℃, 최고기온: 22~29℃)과 비슷하겠습니다.
강수량은 평년(3~
16mm)보다 중부지방은 많겠으나, 남부지방은 비슷하겠고, 제주도는 적겠습니다.

:

:

:

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

```
In [13]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup
import urllib.request as req

# 온라인 파일 열기
URL="http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"
soup=BeautifulSoup(req.urlopen(URL), 'html.parser') # BeautifulSoup 인스턴스 생성
title=soup.find("title").string # title tag
wf=soup.find_all("wf") # 모든 wf tag
print("Title#\n{0}".format(title))
for x in wf :
    text=x.string # wf tag 하나의 문자열
    print("wf : {0}".format(text))
```

Title

기상청 육상 중기예보

wf : 기압골의 영향으로 15일은 강원도와 충북, 경상도, 18일은 중부지방, 19일은 제주도를 제외한 전국에 비가 오겠고, 그 밖의 날은 고기압의 영향으로 맑은 날이 많겠습니다.
기온은 평년(최저기온: 14~20℃, 최고기온: 22~29℃)과 비슷하겠습니다.
강수량은 평년(3~16mm)보다 중부지방은 많겠으나, 남부지방은 비슷하겠고, 제주도는 적겠습니다.

:

:

:

wf : 맑음

wf : 맑음

wf : 구름많음

❖ CSS(cascading style sheets) 선택자

- CSS는 웹 문서의 전반적인 스타일을 미리 저장해 둔 스타일시트로 문서 전체의 일관성을 유지할 수 있고, 세세한 스타일 지정의 필요를 줄어줄게 함
- CSS 선택자를 지정해서 원하는 요소를 추출

메소드	설명
<code>soup.select_one(<선택자>)</code>	CSS 선택자로 요소 하나를 추출
<code>soup.select(<선택자>)</code>	CSS 선택자로 요소 여러 개를 리스트로 추출

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재학 학생 자료로만 사용가능 합니다.

```
In [14]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup

# 파일 열기
fp=open("C:/Users/datam_000/Documents/Python/Module04/Ch01/CSS_Exam.html", "r",
        encoding='UTF8')

soup=BeautifulSoup(fp, 'html.parser') # BeautifulSoup 인스턴스 생성

# CSS 쿼리로 추출하기
# 제목 부분 추출하기
h1=soup.select_one("div#meigen > h1").string
print("h1={0}".format(h1))

# 목록 부분 추출
li_list=soup.select("div#meigen > ul.items > li")
print("\nli_list={0}\n".format(li_list))
for li in li_list :
    print("li={0}".format(li.string))

fp.close() # 파일 닫기
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

h1=파이썬 프로그램

```
li_list=[<li><a href="/Python/Basics">Python 기초</a></li>, <li><a href="/Python/Gui">GUI 프로그래밍</a></li>, <li><a href="/Python/Data">Python 데이터</a></li>, <li><a href="/Python/Django">Django 기초</a></li>, <li><a href="/Python/Applications">Python 활용</a></li>, <li><a href="/Python/Tips">Python 팁</a></li>, <li><a href="/Home/Contact">Contact</a></li>, <li><a href="javascript:showSearch()"><i aria-hidden="true" class="fa fa-search"></i>검색</a></li>]
```

```
li=Python 기초  
li=GUI 프로그래밍  
li=Python 데이터  
li=Django 기초  
li=Python 활용  
li=Python 팁  
li=Contact  
li=None
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재교육자료로만 사용가능 합니다.

❖ 네이버 금융에서 환율 정보 추출

- 네이버 금융의 시장 지표 페이지

<https://finance.naver.com/marketindex/>



BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
<div class="market_include">
<div class="market_data">
  <div class="market1">
    <div class="title">
      <h2 class="h_market1"><span>환전 고시 환율</span></h2>

    </div>
```

```
  <div class="market_include">
<div class="market_data">
  <div class="market1">
    <div class="title">
      <h2 class="h_market1"><span>환전 고시 환율</span></h2>

    </div>
    <!-- data -->
    <div class="data">

      <ul class="data_lst" id="exchangeList">

        <li class="on">
          <a href="/marketindex/exchangeDetail.nhn?marketindexCd=FX_USDKRW" class="head_usd" onClick="clickcr(this, 'fr1.usdt', '', '', event);">
            <h3 class="h_lst"><span class="blind">미국 USD</span></h3>

            <div class="head_info point_up">
              <span class="value">1,163.50</span>
              <span class="txt_krw"><span class="blind">원</span></span>
              <span class="change">2.00</span>
              <span class="blind">상승</span>
            </div>
          </a>
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

```
In [15]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup
import urllib.request as req

# URL 주소 가져오기
url="https://finance.naver.com/marketindex/"
res=req.urlopen(url)

# HTML 분석하기
soup=BeautifulSoup(res, 'html.parser') # BeautifulSoup 인스턴스 생성

# CSS 쿼리로 추출하기
# 제목 부분 추출하기
h2=soup.select_one("div.title > h2.h_market1 > span").string
print("*** {0} ***".format(h2))

title=soup.select_one("h3.h_1st > span.blind").string
val=soup.select_one("div.head_info > span.value").string
print("{0}={1}₩n".format(title, val))

title_list=soup.select("h3.h_1st > span.blind")
val_list=soup.select("div.head_info > span.value")
n=len(title_list)
for i in range(0,n) :
    print("{0} : {1}".format(title_list[i].string, val_list[i].string))
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

*** 환전 고시 환율 ***

미국 USD=1,163.50

미국 USD : 1,163.50

일본 JPY(100엔) : 1,081.77

유럽연합 EUR : 1,315.57

중국 CNY : 169.27

일본 엔/달러 : 107.3100

달러/유로 : 1.1395

달러/영국파운드 : 1.2742

달러인덱스 : 95.7100

WTI : 57.43

휘발유 : 1503.88

국제 금 : 1396.2

국내 금 : 52016.35

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ CSS 선택자로 지정할 수 있는 서식

■ 기본서식

서식	설명
*	모든 요소 선택
<요소 이름>	요소 이름 기반으로 선택
.<클래스 이름>	클래스 이름 기반으로 선택
#<id 이름>	id 속성 기반으로 선택

■ 선택자들의 관계를 지정하는 서식

서식	설명
<선택자>, <선택자>	쉼표로 구분된 여러 개의 선택자를 모두 선택
<선택자> <선택자>	앞 선택자의 후손 중 뒤 선택자에 해당하는 것을 모두 선택
<선택자> > <선택자>	앞 선택자의 자손 중 뒤 선택자에 해당하는 것을 모두 선택
<선택자> + <선택자>	같은 계층에서 바로 뒤에 있는 요소를 선택
<선택자1> ~ <선택자2>	선택자 1 부터 선택자 2 까지의 요소를 모두 선택

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재인 학습 자료로만 사용가능 합니다.

- 선택자 속성을 기반으로 지정하는 서식

서식	설명
<요소>[<속성>]	해당 속성을 가진 요소를 선택
<요소>[<속성>=<값>]	해당 속성의 값이 지정한 값과 같은 요소를 선택
<요소>[<속성>~=<값>]	해당 속성의 값이 지정한 값을 단어로 포함하고 있다면 선택
<요소>[<속성> =<값>]	해당 속성의 값으로 시작하면 선택
<요소>[<속성>^=<값>]	해당 속성의 값이 지정한 값으로 시작하면 선택
<요소>[<속성>\$=<값>]	해당 속성의 값이 지정한 값으로 끝나면 선택
<요소>[<속성>*<값>]	해당 속성의 값이 지정한 값을 포함하고 있다면 선택

- 위치 또는 상태를 지정하는 서식

서식	설명
<요소>:root	루트 요소
<요소>:nth-child(n)	n번째 자식요소
<요소>:nth-last-child(n)	뒤에서 n번째 자식요소
<요소>:nth-of-type(n)	n번째 해당 종류의 요소

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재인 학습 자료로만 사용가능 합니다.

■ 위치 또는 상태를 지정하는 서식

서식	설명
<요소>:first-child	첫 번째 자식요소
<요소>:last-child	마지막 자식 요소
<요소>:first-of-type	첫 번째 해당 종류의 요소
<요소>:last-of-type	마지막 해당 종류의 요소
<요소>:only-child	자식으로 유일한 요소
<요소>:only-of-type	자식으로 유일한 종류의 요소
<요소>:empty	내용이 없는 요소
<요소>:lang(code)	특정 언어로 code를 지정한 요소
<요소>:not(s)	s 이외의 요소
<요소>:enabled	활성화된 UI 요소
<요소>:disabled	비활성화된 UI 요소
<요소>:checked	체크돼 있는 UI 요소

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재교육 자료로만 사용가능 합니다.

```
In [16]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup

# 파일 열기
fp=open("C:/Users/datam_000/Documents/Python/Module04/Ch01/CSS_Sel.html","r",
        encoding='UTF8')

soup=BeautifulSoup(fp, 'html.parser') # BeautifulSoup 인스턴스 생성

# CSS 선택자로 추출하기
sel=lambda q : print(soup.select_one(q).string)
sel("#app") # id 속성이 app인 것 추출
sel("li#app") # li 태그에서 id 속성이 app인 것 추출
sel("ul > li#app") # ul 태그의 자식 li 태그에서 id 속성이 app인 것 추출
sel(".items #app") # class="items" 다음에 id="app" 선택
sel(".items > #app") # class="items" 자식의 id="app" 선택
sel("ul.items > li#app") # ul 태그 class="items" 자식의 li 태그 id="app" 선택

sel("li[id='app']") # id="app"인 li 태그(속성 검색 방법)
sel("li:nth-of-type(5)") # 5번째 li 태그 선택

# select와 find_all 메소드 사용
print(soup.select("li")[4].string)
print(soup.find_all("li")[4].string)

fp.close() # 파일 닫기
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [17]: # 라이브러리 읽어 들이기
from bs4 import BeautifulSoup

# 파일 열기
fp=open("C:/Users/datam_000/Documents/Python/Module04/Ch01/fr_ve.html", "r",
        encoding='UTF8')

soup=BeautifulSoup(fp, 'html.parser') # BeautifulSoup 인스턴스 생성

# CSS 선택자로 추출하기
# 두 번째 ul 태그의 4번째 요소
print(soup.select_one("ul:nth-of-type(2) > li:nth-of-type(4)").string, "\n")

# li 태그의 4번째 요소
frve_list=soup.select("li:nth-of-type(4)")
for st in frve_list :
    print(st.string)
print()
```

아보카도

오렌지

아보카도

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

```
In [18]: # id="ve-list"의 자식 li 태그의 data-lo속성이 'us'
print(soup.select("#ve-list > li[data-lo='us']")[2].string, "\n")

# id="ve-list"의 자식 li 태그의 class="red"
print(soup.select_one("#ve-list > li.red").string, "\n")

# find 메소드 사용
cond={"data-lo": "us", "class": "black"}
print(soup.find("li", cond).string, "\n")

# find 메소드 연속 사용
cond={"data-lo": "us", "class": "black"}
print(soup.find(id="ve-list").find("li", cond).string)

fp.close() # 파일 닫기
```

아보카도

파프리카

가지

가지

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재인 학습 자료로만 사용가능 합니다.

❖ 링크에 있는 것을 한꺼번에 내려 받기

- 링크 대상이 상대 경로일 때에 HTML의 내용에 추가적인 처리가 필요
- 상대 경로를 절대 경로로 변환하는 것이 필요
 - urllib.parse.urljoin(base, path)을 사용

```
In [19]: # 라이브러리 읽어 들이기
from urllib.parse import urljoin

# 기본주소
base="http://example.com/tml/a.html"

# 상대주소를 절대주소로 처리
print(base, "\n")
print(urljoin(base, "b.html"))
print(urljoin(base, "sub/c.html"))
print(urljoin(base, "../index.html"))
print(urljoin(base, "../image/img.png"))
print(urljoin(base, "../css/css_doc.css"), "\n")

# 절대주소입력(기존의 주소 무시)
print(urljoin(base, "http://www.naver.com"))
print(urljoin(base, "http://www.daum.net"))
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재미 학습 자료로만 사용가능 합니다.

`http://example.com/tml/a.html`

`http://example.com/tml/b.html`

`http://example.com/tml/sub/c.html`

`http://example.com/index.html`

`http://example.com/image/img.png`

`http://example.com/css/css_doc.css`

`http://www.naver.com`

`http://www.daum.net`

- 재귀적으로 HTML 페이지 처리
 - "a.html"에서 "b.html"로 링크 이동하고, "b.html"에서 "c.html"로 링크하여 이동하는 경우 3개의 페이지를 모두 다운로드하여 분석하는 것이 필요
 - 이러한 구조의 데이터는 함수를 이용한 재귀 처리
 - 어떤 함수 내부에서 해당함수 자신을 호출하는 것이 재귀

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

```
In [20]: # 파이썬 매뉴얼을 재귀적으로 다운받는 프로그램
# 모듈 읽어 들이기 --- (※1)
from bs4 import BeautifulSoup
from urllib.request import *
from urllib.parse import *
from os import makedirs
import os.path, time, re

# 이미 처리한 파일인지 확인하기 위한 변수 --- (※2)
proc_files = {}

# HTML 내부에 있는 링크를 추출하는 함수 --- (※3)
def enum_links(html, base):
    soup = BeautifulSoup(html, "html.parser")
    links = soup.select("link[rel='stylesheet']") # CSS
    links += soup.select("a[href]") # 링크
    result = []
    # href 속성을 추출하고, 링크를 절대 경로로 변환 --- (※4)
    for a in links:
        href = a.attrs['href']
        url = urljoin(base, href)
        result.append(url)
    return result
```


BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

```
In [21]: # 파일을 다운받고 저장하는 함수 --- (※5)
def download_file(url):
    o = urlparse(url)
    dir_path="C:/Users/datam_000/Documents/Python/Module04/Ch01/"
    savepath = dir_path + o.netloc + o.path
    if re.search(r"/$", savepath): # 폴더라면 index.html
        savepath += "index.html"
    savedir = os.path.dirname(savepath)
    # 모두 다운됐는지 확인
    if os.path.exists(savepath): return savepath
    # 다운받을 폴더 생성
    if not os.path.exists(savedir):
        print("mkdir=", savedir)
        makedirs(savedir)
    # 파일 다운받기 --- (※6)
    try:
        print("download=", url)
        urlretrieve(url, savepath)
        time.sleep(1) # 1초 휴식 --- (※7)
        return savepath
    except:
        print("다운 실패: ", url)
        return None
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재학 학생 자료로만 사용가능 합니다.

```
In [22]: # HTML을 분석하고 다운받는 함수 --- (※8)
def analyze_html(url, root_url):
    savepath = download_file(url)
    if savepath is None: return
    if savepath in proc_files: return # 이미 처리했다면 실행하지 않음 --- (※9)
    proc_files[savepath] = True
    print("analyze_html=", url)
    # 링크 추출 --- (※10)
    html = open(savepath, "r", encoding="utf-8").read()
    links = enum_links(html, url)
    for link_url in links:
        # 링크가 루트 이외의 경로를 나타낸다면 무시 --- (※11)
        if link_url.find(root_url) != 0:
            if not re.search(r".css$", link_url): continue
        # HTML이라면
        if re.search(r".(html|htm)$", link_url):
            # 재귀적으로 HTML 파일 분석하기
            analyze_html(link_url, root_url)
            continue
        # 기타 파일
        download_file(link_url)
```

BeautifulSoup로 스크레이핑하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
재일 학습 자료로만 사용가능 합니다.

```
In [23]: if __name__ == "__main__":  
# URL에 있는 모든 것 다운받기 --- (*12)  
url = "https://docs.python.org/3.5/library/"  
analyze_html(url, url)
```

```
mkdir= C:/Users/datam_000/Documents/Python/Module04/Ch01/docs.python.org/3.5/library  
download= https://docs.python.org/3.5/library/  
analyze_html= https://docs.python.org/3.5/library/  
mkdir= C:/Users/datam_000/Documents/Python/Module04/Ch01/docs.python.org/3.5/_static  
download= https://docs.python.org/3.5/_static/pydoctHEME.css  
download= https://docs.python.org/3.5/_static/pygments.css  
download= https://docs.python.org/3.5/library/intro.html  
analyze_html= https://docs.python.org/3.5/library/intro.html
```

2

고급 스크레이핑



❖ HTTP 통신

- 웹 브라우저와 웹 서버는 HTTP 통신규약(프로토콜)을 사용해서 통신
- 브라우저에서 서버로 요청(request)하면, 서버에서 브라우저로 응답(response)할 때 어떻게 할 것인지를 나타내는 규약
 - 웹 브라우저로 `http://www.naver.com`이라는 웹 서버 탐색
 - 웹 서버가 발견하면 `index.html` 파일을 보고 싶다고 요청
 - `naver.com` 서버가 이러한 요청을 받으면 `index.html` 파일의 내용을 응답
- 같은 URL에 여러 번 접근해도 같은 데이터를 돌려주는 무상태 (stateless)통신

❖ 쿠키

- 웹 브라우저를 통해 사이트에 방문하는 사람의 컴퓨터에 일시적으로 데이터를 저장하는 기능
- 1개의 쿠키에 저장할 수 있는 데이터의 크기는 4096byte로 제한
- HTTP 통신 헤더를 통해 읽고 쓰기가 가능
- 방문자 또는 확인자 측에서 원하는 대로 변경 가능
- 변경하면 문제가 될 비밀번호 등의 정보를 저장하기는 알맞지 않음

로그인이 필요한 사이트에서 **다음받기**

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 세션

- 쿠키를 사용해 데이터를 저장
- 쿠키에는 방문자 고유 ID만 저장하고, 모든 데이터는 웹 서버에 저장
- 저장할 수 있는 데이터에 제한이 없음
- 회원제 웹 사이트 등의 구현이 가능

❖ requests 사용


- urllib.request를 이용해 쿠키를 이용한 접근이 가능
 - 방법이 조금 복잡
- requests 패키지를 사용하면 쉽게 쿠키를 이용한 접근이 가능
- 프로그램(봇 등)이 쉽게 로그인 할 수 없게 보안 처리된 네이버 또는 다음 등 포털 사이트 등은 지금 사용하는 방법으로 로그인 불가능

로그인이 필요한 사이트에서 다음받기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 한빛출판네트웍스


[HOME](#) [한빛미디어](#) [한빛아카데미](#) [한빛비즈](#) [한빛라이프](#) [한빛에듀](#) [리얼타임](#) [한빛정보교과서](#) [한빛대관서비스](#) [로그인](#) [회원가입](#) [마이한빛](#) [장바구니](#)

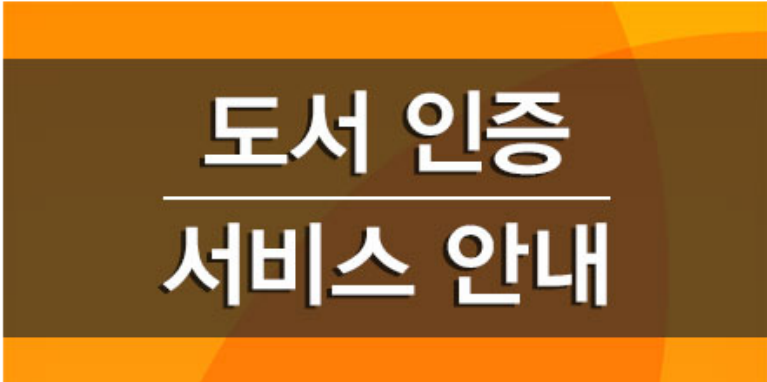
 **한빛출판네트웍스** [BRAND](#) [Channel.H](#) [STORE](#) [SUPPORT](#) [EVENT](#)

[로그인](#) | [아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#) | [이용약관](#) | [개인정보취급방침](#)

로그인

☐ 아이디 저장
[아이디 찾기](#) [비밀번호 찾기](#) [회원가입](#)


로그인



로그인이 필요한 사이트에서 다음받기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
<!-- 로그인 영역 -->
<form name="frm" id="frm" action="#" method="post">
    <input name="retun_url" id="retun_url" type="hidden" value="" class="i_text" size="100" >
    <div class="login_left">
        <fieldset>
            <legend>한빛출판네트워크 로그인</legend>
            <label class="i_label" for="login_id"><strong></strong>
                <input name="m_id" id="m_id" type="text" value="" class="i_text"
placeholder="아이디" onkeydown="javascript:if(event.keyCode==13){login_proc(); return false;}">
            </label>
            <label class="i_label" for="login_pw"><strong></strong>
                <input name="m_passwd" id="m_passwd" type="password" value=""
class="i_text" placeholder="비밀번호" onkeydown="javascript:if(event.keyCode==13){login_proc(); return
false;}">
            </label>
            <label>
                <input type="button" name="login_btn" id="login_btn" value="로그인"
" class="btn_login" >
            </label>
            <label class="i_label2">
                <input type="checkbox" name="keepid" id="keepid" value="1"
class="i_check"><strong>아이디 저장</strong>
            </label>
        </fieldset>
        <ul class="login_btn">
            <li><a href="/member/find_id.html" class="btn_idc">아이디 찾기</a></li>
            <li><a href="/member/find_pw.html" class="btn_pwc">비밀번호 찾기</a></li>
            <li><a href="/member/member_agree.html" class="btn_joinc">회원가입</a></li>
        </ul>
    </div>
</form>
<!-- 로그인 영역 -->
```


로그인이 필요한 사이트에서 다음받기

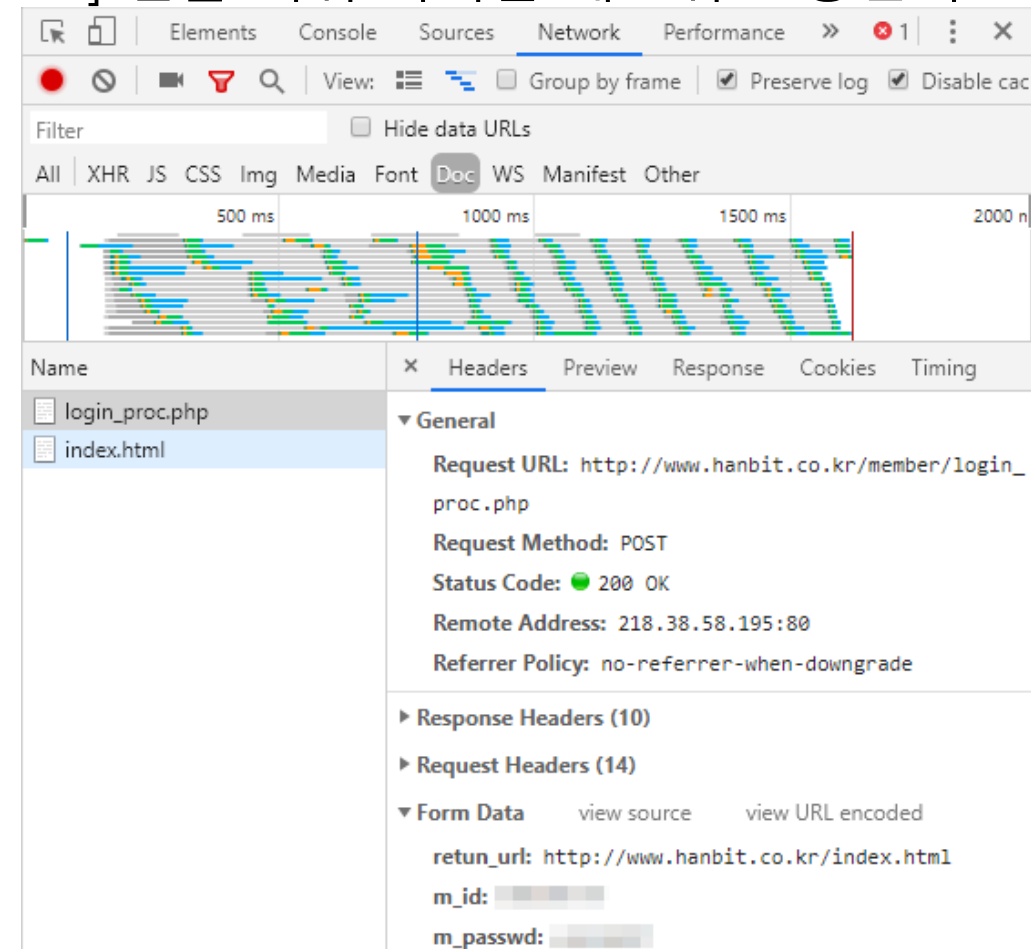
이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 입력양식(input)으로 m_id와 m_passwd라는 값(name 속성의 값)을 입력하여 입력 양식을 제출하면 로그인 되는 구조

❖ 로그인 과정 분석

- 크롬 등에서 [검사] 화면을 띄우고 [Network] 탭을 띄워 어떠한 네트워크 통신이 오가는지 확인
- login_proc.php 선택
 - 로그인 관련 기능 처리
 - m_id와 m_passwd 정보 확인



http://www.hanbit.co.kr/member/login_proc.php
에 입력양식 데이터를 POST로 전달하면 로그인




로그인이 필요한 사이트에서 다음받기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

[HOME](#) [한빛미디어](#) [한빛아카데미](#) [한빛비즈](#) [한빛라이프](#) [한빛에듀](#) [리얼타임](#) [한빛정보교과서](#) [한빛대관서비스](#) [로그아웃](#) [개인정보수정](#) [마이한빛](#) [장바구니](#)

 **한빛출판네트워크** [BRAND](#) [Channel.H](#) [STORE](#) [SUPPORT](#) [EVENT](#) 

[한빛멤버십](#) | [마일리지 / 한빛이코인](#) [위시리스트](#) | [장바구니](#) | [구매이력 \(주문조회\)](#) | [My 쿠폰](#) [My Book](#) | [My eBook](#) | [My 강의](#)



(김용태)님의
회원 등급은 일반 (교수준회원) 입니다.

마일리지
0 점

한빛이코인
0 원

최근 구매이력

주문일자	상품명	주문금액
------	-----	------

My Book

리스트가 없습니다.

My eBook

리스트가 없습니다.

My 강의

리스트가 없습니다.

로그인이 필요한 사이트에서 다음받기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 파이썬으로 로그인

■ 마일리지와 이코인 출력

```
In [24]: # 로그인을 위한 모듈 추출하기
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin

# 아이디와 비밀번호 지정하기[자신의 것을 사용해주세요]
USER = "<ID>"
PASS = "<PassWD>"

# 세션 시작하기
session = requests.session()

# 로그인하기
login_info = {
    "m_id": USER, # 아이디 지정
    "m_passwd": PASS # 비밀번호 지정
}
url_login = "http://www.hanbit.co.kr/member/login_proc.php"
res = session.post(url_login, data=login_info)
res.raise_for_status() # 오류가 발생하면 예외가 발생합니다.
```

로그인이 필요한 사이트에서 데이터 받기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [25]: # 마이페이지에 접근하기
url_mypage = "http://www.hanbit.co.kr/myhanbit/myhanbit.html"
res = session.get(url_mypage)
res.raise_for_status()

# 마일리지와 이코인 가져오기
soup = BeautifulSoup(res.text, "html.parser")
mileage = soup.select_one(".mileage_section1 span").get_text()
ecoin = soup.select_one(".mileage_section2 span").get_text()
print("마일리지 : {}".format(mileage))
print("이코인 : {}".format(ecoin))
```

마일리지: 0
이코인: 0

```
In [26]: # 마이페이지에 접근하기
url_mypage = "http://www.hanbit.co.kr/myhanbit/membership.html"
res = session.get(url_mypage)
res.raise_for_status()

# 날짜별 순수구매금액과 적립마일리지 가져오기
soup = BeautifulSoup(res.text, "html.parser")
date = soup.select_one("table.tbl_type_list2 tr td").get_text()
buy = soup.select_one("table.tbl_type_list2 tr td.right").get_text()
month_mileage = soup.select_one("table.tbl_type_list2 tr td:nth-of-type(4)").get_text()

print("날짜 : {0}".format(date))
print("순수구매금액 : {0}".format(buy))
print("적립마일리지 : {0}".format(month_mileage))
```

날짜 : 2019 / 06
순수구매금액 : 0 원
적립마일리지 : 0 점

❖ requests의 메소드

- HTTP에서 사용하는 GET과 POST 등의 메소드는 requests 모듈에 같은 이름의 메소드가 존재

```
In [27]: # 로그인을 위한 모듈 추출하기
import requests

# GET 요청
r=requests.get("http://google.com")
print(r.text, "₩₩₩")

# POST 요청
formdata={"key1":"value1", "key2":"value2"}
r=requests.post("http://example.com", data=formdata)
print(r.content)

# 그 이외에 PUT, DELETE, HEAD 등의 요청 메소드
r=requests.put("http://httpbin.org/put")
r=requests.delete("http://httpbin.org/delete")
r=requests.head("http://httpbin.org/get")
```

```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ko"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/
```

로그인이 필요한 사이트에서 데이터 받기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 현재 시간에 대한 데이터를 추출하고 텍스트 형식과 바이너리 형식으로 출력

```
In [28]: # 현재시간 데이터 가져오기
import requests
r = requests.get("http://api.aoikujira.com/time/get.php")

# 텍스트 형식으로 데이터 추출하기
text = r.text
print(text)

# 바이너리 형식으로 데이터 추출하기
bin = r.content
print(bin)
```

2019/06/23 19:26:53

b'2019/06/23 19:26:53'

웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로 개인 학습 자료로만 사용가능 합니다.

❖ 웹 API(Application Programming Interface)

- 어떤 사이트가 가지고 있는 기능을 외부에서도 쉽게 사용할 수 있게 공개한 것
- 원래 어떤 프로그램 기능을 외부 프로그램에서 호출해서 사용할 수 있게 만들 것
 - 간단하게 서로 다른 프로그램이 기능을 공유할 수 있게 절차와 규약을 정의한 것
- 웹 API는 HTTP 통신을 사용하여 클라이언트 프로그램이 API를 제공하는 서버에 HTTP 요청을 보내면 서버가 이러한 요청을 기반으로 XML 또는 JSON 형식 등으로 응답

클라이언트 → 서버 → 클라이언트
(HTML요청) (HTML요청)

웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ OpenWeatherMap의 날씨 정보

<http://openweathermap.org>

- 개발자 등록을 하고 API 키를 발급
- 유료 API
 - 현재 날씨, 5일까지의 날씨는 무료사용
 - 1분에 60번까지 호출 가능

웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

5 day / 3 hour forecast


[API doc](#) [Subscribe](#)

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

	Free	Startup	Developer	Professional	Enterprise
Price per month Price is fixed, no other hidden costs (VAT is not included)	Free	40 USD / month	180 USD / month	470 USD / month	2,000 USD / month
Subscribe	Get API key and Start	Subscribe	Subscribe	Subscribe	Subscribe


웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.
클릭

 Support Center

Weather in your city

Hello YongTae kim



Weather

Maps ▾

API

Price

Partners

Stations

Widgets

News

About ▾

클릭

New Products

Setup

API keys

Services

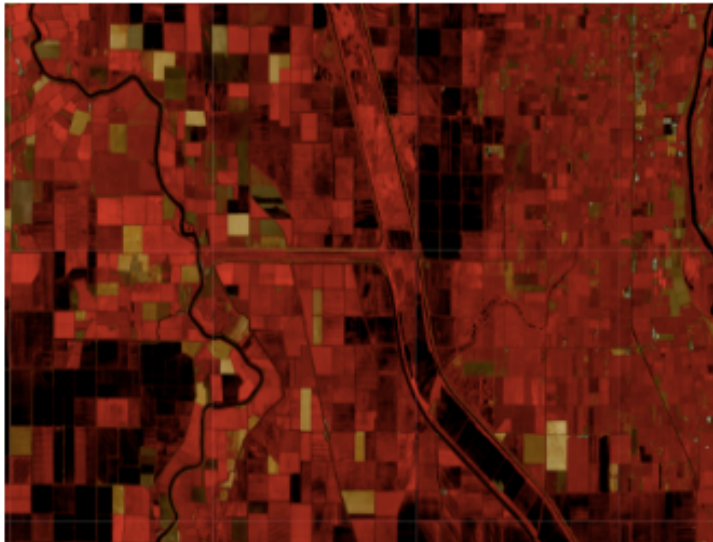
Payments

Billing plans

Block logs

History bulk

Logout



API for Agricultural on agromonitoring.com

Try our simple and fast APIs to satellite imagery, weather data and other products such as:

- Satellite imagery archive (True & False color, NDVI & EVI indices)
- Weather (current data, forecast and history)
- Accumulated temperature and precipitation
- Soil temperature and moisture

All information and API documentation is on agromonitoring.com. Read more in our [Blog](#).

How to start

웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

Support Center

Weather in your cityHello YongTae kim

OpenWeatherMap

WeatherMapsAPIPricePartnersStationsWidgetsNewsAbout

New ProductsSetupAPI keysServicesPaymentsBilling plansBlock logsHistory bulk

Logout

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key

0a02c9c74f92093dfa1b47666806574f

Name

Default

Create key

* Name

Generate

도시 목록 데이터

<http://bulk.openweathermap.org/sample/city.list.json.gz>

I 웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [29]: import requests
import json
# API 키를 지정합니다. 자신의 키로 변경해서 사용
apikey = "0a02c9c74f92093df a1b47666806574f"

# 날씨를 확인할 도시 지정하기
cities = ["Seoul,KR", "Tokyo,JP", "New York,US"]

# API 지정
api = "http://api.openweathermap.org/data/2.5/weather?q={city}&APPID={key}"

# 켈빈 온도를 섭씨 온도로 변환하는 함수
k2c = lambda k: k - 273.15

# 각 도시의 정보 추출하기
for name in cities:
    # API의 URL 구성하기
    url = api.format(city=name, key=apikey)
    # API에 요청을 보내 데이터 추출하기
    r = requests.get(url)
    # 결과를 JSON 형식으로 변환하기
    data = json.loads(r.text)
    # 결과 출력하기 --- (※8)
    print("+ 도시 =", data["name"])
    print("| 날씨 =", data["weather"][0]["description"])
    print("| 최저 기온 =", k2c(data["main"]["temp_min"]))
    print("| 최고 기온 =", k2c(data["main"]["temp_max"]))
    print("| 습도 =", data["main"]["humidity"])

    print("| 기압 =", data["main"]["pressure"])
    print("| 풍향 =", data["wind"]["deg"])
    print("| 풍속 =", data["wind"]["speed"])
    print("")
```

웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로 개인 학습 자료로만 사용가능 합니다.

JSON(JavaScript Object Notation)

속성-값 쌍 또는 "키-값 쌍"으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷

```
{
  "coord": {
    "lon": 126.98,
    "lat": 37.57
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 296.33,
    "pressure": 1010,
    "humidity": 64,
    "temp_min": 295.15,
    "temp_max": 298.15
  },
  "visibility": 10000,
  "wind": {
    "speed": 1,
    "deg": 180
  },
  "clouds": {
    "all": 1
  },
  "dt": 1561293117,
  "sys": {
    "type": 1,
    "id": 5501,
    "message": 0.0081,
    "country": "KR",
    "sunrise": 1561234279,
    "sunset": 1561287412
  },
  "timezone": 32400,
  "id": 1835848,
  "name": "Seoul",
  "cod": 200
}
```

웹 API로 데이터 추출하기

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 국내 웹 API

API Store

<https://www.apistore.co.kr/main.do>

- 포탈 사이트(네이버 개발자 센터와 다음 개발자 센터)

<https://developers.naver.com/main/>

<https://developers.daum.net/>

- 쇼핑 정보(옥션)

<http://developer.auction.co.kr/>

- 주소전환(행정자치부, 우체국)

<http://www.juso.go.kr/openIndexPage.do>

<https://biz.epost.go.kr/ui/index.jsp>

3



데이터 소스의 서식과 가공

❖ 텍스트 데이터와 바이너리 데이터

- 텍스트 데이터는 일반적으로 텍스트 에디터로 편집할 수 있는 데이터 포맷
 - 자연어(한국어, 영어, 일본어 등) 숫자 등으로 구성
 - 특수하게 줄 바꿈과 탭 등 제어 문자도 포함
 - XML, JSON, CSV 등
- 텍스트 데이터 이외의 데이터를 바이너리 데이터
 - 바이너리는 문자와 상관 없이 데이터를 사용할 수 있는 데이터 영역을 활용하는 데이터 형식
 - 텍스트 에디터로 열수 없으며, 시각적으로 확인해도 의미를 알 수 없는 문자열로 표현
 - 텍스트 데이터 보다 크기가 작음
 - 텍스트 3byte 파일을 바이너리 1byte로 저장
 - 동영상, 이미지 등은 대부분 바이너리 데이터

❖ XML(extensible markup language) 분석

- XML은 텍스트 데이터를 기반을 하는 형식
- 범용적인 형식으로 널리 사용
- 웹 API가 XML 형식을 활용
- 특정 목적에 따라서 태그로 감싸 마크업 하는 범용적인 형식
- XML은 데이터를 계층 구조로 표현
 - 어떤 데이터 아래 서브 데이터를 추가 가능
- XML 기본 구조

<요소 속성="값">내용</요소>

- <요소> 태그로 감싸 마크업
- 원하는 요소 이름을 사용
- 하나의 요소에 속성을 사용해 여러 값을 추가로 지정

<product id="S001" price="45000">SD 카드</product>

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 다른 요소의 그룹으로 묶어 요소들이 계층 구조를 갖도록 작성

```
<products type="전자제품">  
  <product id="S001" price="45000">SD 카드</product>  
  <product id="S002" price="32000">마우스</product>  
</products>
```

❖ 파이썬으로 XML 분석하기

- BeautifulSoup을 이용하여 XML을 분석

<http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108>

```
▼ <rss version="2.0">  
  ▼ <channel>  
    <title>기상청 육상 중기예보</title>  
    ▼ <link>  
      http://www.kma.go.kr/weather/forecast/mid-term_01.jsp  
    </link>  
    <description>기상청 날씨 웹서비스</description>  
    <language>ko</language>  
    <generator>기상청</generator>  
    <pubDate>2019년 06월 23일 (일)요일 18:00</pubDate>  
    ▼ <item>  
      <author>기상청</author>  
      <category>육상중기예보</category>  
      <title>전국 육상 중기예보 - 2019년 06월 23일 (일)요일 18:00 발표</title>  
      ▼ <link>
```

I 웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [30]: from bs4 import BeautifulSoup
import urllib.request as req
import os.path

url="http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108"
savename="forecast.xml"
if not os.path.exists(savename):
    req.urlretrieve(url, savename) ◀ 로컬 파일로 저장

# BeautifulSoup로 분석하기
xml=open(savename, "r", encoding="utf-8").read()
soup=BeautifulSoup(xml, 'html.parser') ◀ html.parse 사용시 데이터의 태그를 소문자로 처리
                                     따라서 소문자로 태그를 입력

# 각 지역 확인하기
info={}
for location in soup.find_all("location"):
    name=location.find('city').string
    weather=location.find('wf').string
    if not (weather in info):
        info[weather]=[]
    info[weather].append(name)

# 각 지역의 날씨를 구분해서 출력하기
for weather in info.keys():
    print("+", weather)
    for name in info[weather]:
        print("| - ", name)
```

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- + 맑음
 - 서울
 - 인천
 - 수원
 - 파주
 - 이천
 - 평택
 - 춘천
 - 원주
 - 강릉
 - 대전
 - 세종
 - 홍성
 - 청주
 - 충주
 - 영동
- + 구름맑음
 - 광주
 - 목포
 - 여수
 - 순천

❖ JSON 분석

- JSON도 텍스트 데이터를 기반으로 하는 가벼운 데이터 형식
- JSON은 자바스크립트에서 사용하는 객체 표기 방법을 기반
- JSON은 자바스크립트 전용 데이터 형식은 아니며, 다양한 소프트웨어와 프로그래밍 언어끼리 데이터를 교환할 때 사용
- 확장자는 ".json"
- 파이썬 표준 모듈에도 json이 포함
- JSON의 구조
 - 숫자, 문자열, 논리, 배열, 객체, null 6가지 종류의 데이터를 사용

JSON 소개
<http://json.org>

자료형	표현 방법	사용 예
숫자	숫자	30
문자열	큰 따옴표로 감싸 표현	"str"
논리	true 또는 false	true
배열	[n1,n2,n3,...]	[1,2,10,500]
객체	{"key1":value, "key1":value,...}	{"org":50, "com":10}
null	null	null

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 규칙은 단순하지만 배열 안에 객체를 넣거나 객체안에 배열을 넣는 방법으로 복잡한 데이터를 표현
- JSON의 배열은 파이썬의 리스트, 객체는 파이썬의 딕셔너리와 동일

JSON 데이터

<https://api.github.com/repositories>

- 깃허브(github)는 Git을 사용하는 프로젝트를 지원하는 웹 호스팅 서비스로 오픈 소스 코드 저장소로 유명
- 무작위로 리포지토리의 이름과 소유자를 추출해서 출력

I 웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [31]: import urllib.request as req
import os.path, random
import json

# JSON 데이터 내려받기
url = "https://api.github.com/repositories"
savename = "repo.json"
if not os.path.exists(savename):
    req.urlretrieve(url, savename)

# JSON 파일 분석하기
items = json.load(open(savename, "r", encoding="utf-8")) ◀ JSON 문자열 읽기
# 또는
# s = open(savename, "r", encoding="utf-8").read()
# items = json.loads(s)

# 출력하기
for item in items:
    print(item["name"] + " - " + item["owner"]["login"])
```


웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

■ JSON 형식으로 출력

```
In [32]: import json

# JSON 데이터 내려받기
price={
    "date":"2019-07-02",
    "price":{
        "Apple":80,
        "Orange":55,
        "Banana":40
    }
}
s=json.dumps(price) ◀ json.dump()를 사용하여 JSON 형식으로 출력
print(s)
```

```
{"date": "2019-07-02", "price": {"Apple": 80, "Orange": 55, "Banana": 40}}
```

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ 엑셀 파일 분석

- 파이썬에서 엑셀 파일을 읽고 쓸 때는 파이썬-엑셀 라이브러리를 사용
- openpyxl 패키지 설치

엑셀 데이터

http://www.index.go.kr/potal/main/EachDtlPageDetail.do?idx_cd=1041

<http://www.index.go.kr/strata/jsp/downloadStblGams2.jsp>

I 웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [33]: import openpyxl

# 엑셀 파일 열기
filename = "C:/Users/datam_000/Documents/Python/Module04/Ch01/stats_104102.xlsx"
book = openpyxl.load_workbook(filename)

# 맨 앞의 시트 추출하기
sheet = book.worksheets[0]

# 시트의 각 행을 순서대로 추출하기
data = []
for row in sheet.rows:
    data.append([
        row[0].value,
        row[9].value
    ])

# 필요없는 줄(헤더, 연도, 계) 제거하기
del data[0]
del data[1]
del data[2]

# 데이터를 인구 순서로 정렬합니다.
data = sorted(data, key=lambda x:x[1])

# 하위 5위를 출력합니다.
for i, a in enumerate(data):
    if (i >= 5): break
    print(i+1, a[0], int(a[1]))
```

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

- 1 세종 280
- 2 제주 657
- 3 울산 1165
- 4 광주 1463
- 5 대전 1502

I 웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

```
In [34]: import openpyxl

# 엑셀 파일 열기
filename = "C:/Users/datam_000/Documents/Python/Module04/Ch01/stats_104102.xlsx"
book = openpyxl.load_workbook(filename)

# 활성화된 시트 추출하기
sheet = book.active

# 서울을 제외한 인구를 구해서 쓰기
for i in range(0, 9):
    total = int(sheet[str(chr(i + 66)) + "3"].value)
    seoul = int(sheet[str(chr(i + 66)) + "4"].value)
    output = total - seoul
    print("서울 제외 인구 =", output)
    # 쓰기
    sheet[str(chr(i + 66)) + "21"] = output
    cell = sheet[str(chr(i + 66)) + "21"]

    # 폰트와 색상 변경해보기
    cell.font = openpyxl.styles.Font(size=14, color="FF0000")
    cell.number_format = cell.number_format

# 엑셀 파일 저장하기
filename = "C:/Users/datam_000/Documents/Python/Module04/Ch01/population.xlsx"
book.save(filename)
print("ok")
```

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

서울 제외 인구 = 39565
서울 제외 인구 = 40203
서울 제외 인구 = 40484
서울 제외 인구 = 40753
서울 제외 인구 = 40997
서울 제외 인구 = 41225
서울 제외 인구 = 41507
서울 제외 인구 = 41766
서울 제외 인구 = 41921
ok

웹의 다양한 데이터 형식

이 자료는 혁신성장 청년인재 집중양성 사업 강의 자료로
개인 학습 자료로만 사용가능 합니다.

❖ Pandas를 이용해 엑셀 파일 읽고 쓰기

- Pandas로 엑셀을 수정하려면 xlrd 모듈이 필요

```
In [35]: import pandas as pd
```

```
# 엑셀 파일 열기
```

```
filename = "C:/Users/datam_000/Documents/Python/Module04/Ch01/stats_104102.xlsx" # 파일
```

```
sheet_name = "stats_104102" # 시트 이름
```

```
book = pd.read_excel(filename, sheetname=sheet_name, header=1) # 첫 번째 줄부터 헤더
```

```
# 2015년 인구로 정렬
```

```
book = book.sort_values(by=2015, ascending=False)
```

```
book
```

	Unnamed: 0	2009	2010	2011	2012	2013	2014	2015	2016	2017
0	계	49773	50515	50734	50948	51141	51328	51529	51696	51778
9	경기	11460	11787	11937	12093	12235	12358	12522	12716	12873
1	서울	10208	10312	10250	10195	10144	10103	10022	9930	9857
		:						:		
6	대전	1484	1504	1516	1525	1533	1532	1518	1514	1502
5	광주	1433	1455	1463	1469	1473	1476	1472	1469	1463

4

요약



요약

- ❖ Python은 웹 스크레이핑을 위한 다양한 라이브러리를 제공하고 있음
- ❖ 웹 상의 다양한 데이터를 Python으로 Load 할 수 있음
 - Python에서 스크레이핑 기초를 습득함으로써 웹 데이터 수집의 기본지식 확보
 - 빅데이터 처리를 위한 자료 수집기술 확보
 - 빅데이터 수집과 관련된 다양한 기법 습득
 - 현장 문제에 적용 가능한 수준까지 프로그래밍 기술 습득