학습 내용

3부. 데이터 분석 라이브러리 활용

- 10장. N차원 배열 다루기
- 11장. 데이터프레임과 시리즈
- 12장. 데이터 시각화
- 13장. 총괄 예제
- 14장. 웹 데이터 수집
  - 1. 퓨티풀솝과 파서
  - 2. requests를 이용한 웹 데이터 수집
  - 3. Selenium을 이용한 웹 데이터 수집
- 15장. 데이터베이스 연동

웹 데이터 수집 종류

웹 데이터 수집

- 뉴스 기사 크롤링
  - RSS 서비스(예: https://www.weather.go.kr/w/pop/rss-guide.do)
  - XML 형식
- SNS 데이터 크롤링
  - 트위터, 댓글 등
  - Text 형식

웹 데이터 수집

- 웹 페이지 URL에서 원하는 정보를 수집
- HTML 형식

데이터를 선택적으로 가져옴

- 공공데이터 수집
  - Open API를 이용
  - URL이 존재
  - CSV 형식, 정형화된 데이터
    - Pandas 패키지의 read_csv("url")을 이용
  - JSON 형식
    - Pandas 패키지의 read_json("url")을 이용

전체 데이터를 가져옴

웹 크롤링 한계

웹 데이터 수집

- 모든 데이터를 다 가져올 수 있을까?
  - 크롤러 제한 규칙 적용
  - HTTP 헤더 등 접근 제한
  - 동적 웹 사이트
  - 캡챠 (자동입력방지)
  - 로그인 해야만 보이는 컨텐츠
  - IP 차단
  - 법적으로 크롤링 금지 등

1절. 퓨티풀솝과 파서

1.1. Beautiful Soup

1절. 뷰티풀솝과 파서

- 뷰티풀솝(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 관용구를 제공하며 문서를 분석하고 필요한 것을 추출하는 도구
- 들어오는 문서를 유니코드로 보내고 문서를 UTF-8로 자동 변환
- 뷰티풀솝은 lxml 및 html5lib과 같은 파이썬 파서 라이브러리를 사용할 수 있음
  - 이를 이용하면 속도를 개선시키거나 호환성이 높은 응용프로그램을 만들 수 있음
- 공식 사이트
  - https://www.crummy.com/software/BeautifulSoup/
- Documentation
  - https://www.crummy.com/software/BeautifulSoup/bs4/doc/

1.2. 파서 라이브러리

1절. 뷰티풀솝과 파서

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

1.3. Selector API

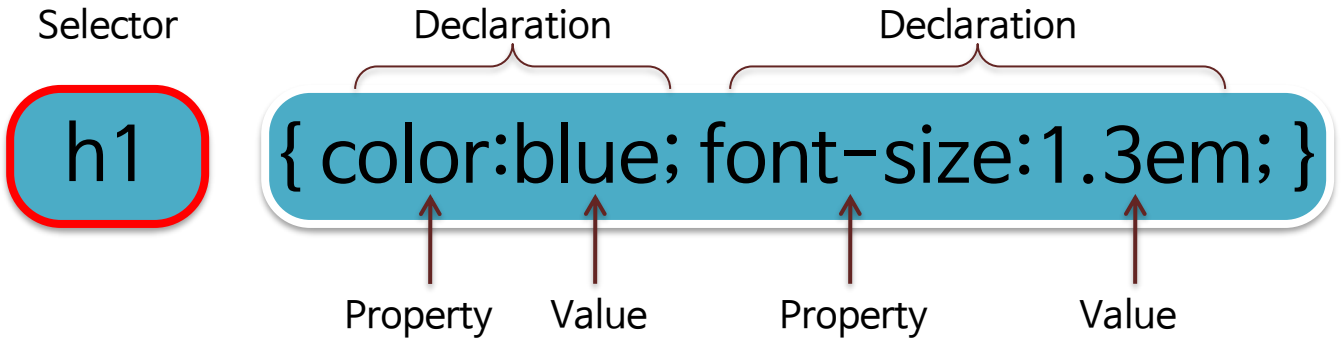
- BeautifulSoup은 가장 일반적으로 사용되는 CSS 선택자를 지원
- BeautifulSoup의 Selector API는 select()와 select_one(), find() 등
- select()와 select_one() 메서드만 알아도 원하는 요소를 찾기에 충분
- soup.select("CSS 선택자") : CSS 선택자에 해당하는 모든 요소를 반환
- soup.select_one("CSS 선택자") : CSS 선택자에 맞는 오직 첫 번째 태그 요소만 반환

- https://www.crummy.com/software/BeautifulSoup/bs4/doc/

1.4. CSS 선택자

1절. 뷰티풀솝과 파서

- CSS(Cascading Style Sheet)
  - CSS는 문서의 콘텐츠와 레이아웃, 글꼴 및 시각적 요소들로 표현되는 문서의 외관(디자인)을 분리하기 위한 목적으로 만들어졌음
- CSS 선택자
  - CSS를 이용해서 HTML 문서에 시각적 요소를 부여할 때 문서 내의 어느 요소에 부여할지를 결정하기 위해 사용하는 것
  - CSS 선택자(Selector)는 HTML 문서의 태그 이름, class 속성, id 속성 등을 이용해서 작성할 수 있음



실습을 위한 준비

1절. 뷰티풀솝과 파서 > 1.4. CSS 선택자

```
1  import requests
2  from requests_file import FileAdapter
3
4  s = requests.Session()
5  s.mount('file://', FileAdapter())
```

```
1  res = s.get('file:///sample.html')
```

```
1  res
```

<Response [200]>

```
1  from bs4 import BeautifulSoup
```

```
1  soup = BeautifulSoup(res.content, "html.parser")
```

```
1  el = soup.select_one("h1")
2  print(el)
```

<h1>Hello CSS</h1>

```
1  <html>
2  <head><title>HTML Sample</title>
3  </head>
4  <body>
5      <h1>Hello CSS</h1>
6      <div id="subject">선택자</div>
7      <div class="contents">선택자를 어떻게 작성하느냐에 따라
   <span>다른 <b>요소가 반환</b></span> 됩니다.</div>
8      <div>CSS 선택자는 다양한 곳에서 <b>활용</b>됩니다.</div>
9  </body>
10 </html>
```

선택자 종류(1/3)

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

- 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")

선택자 종류(2/3)

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

- 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")

선택자 종류(3/3)

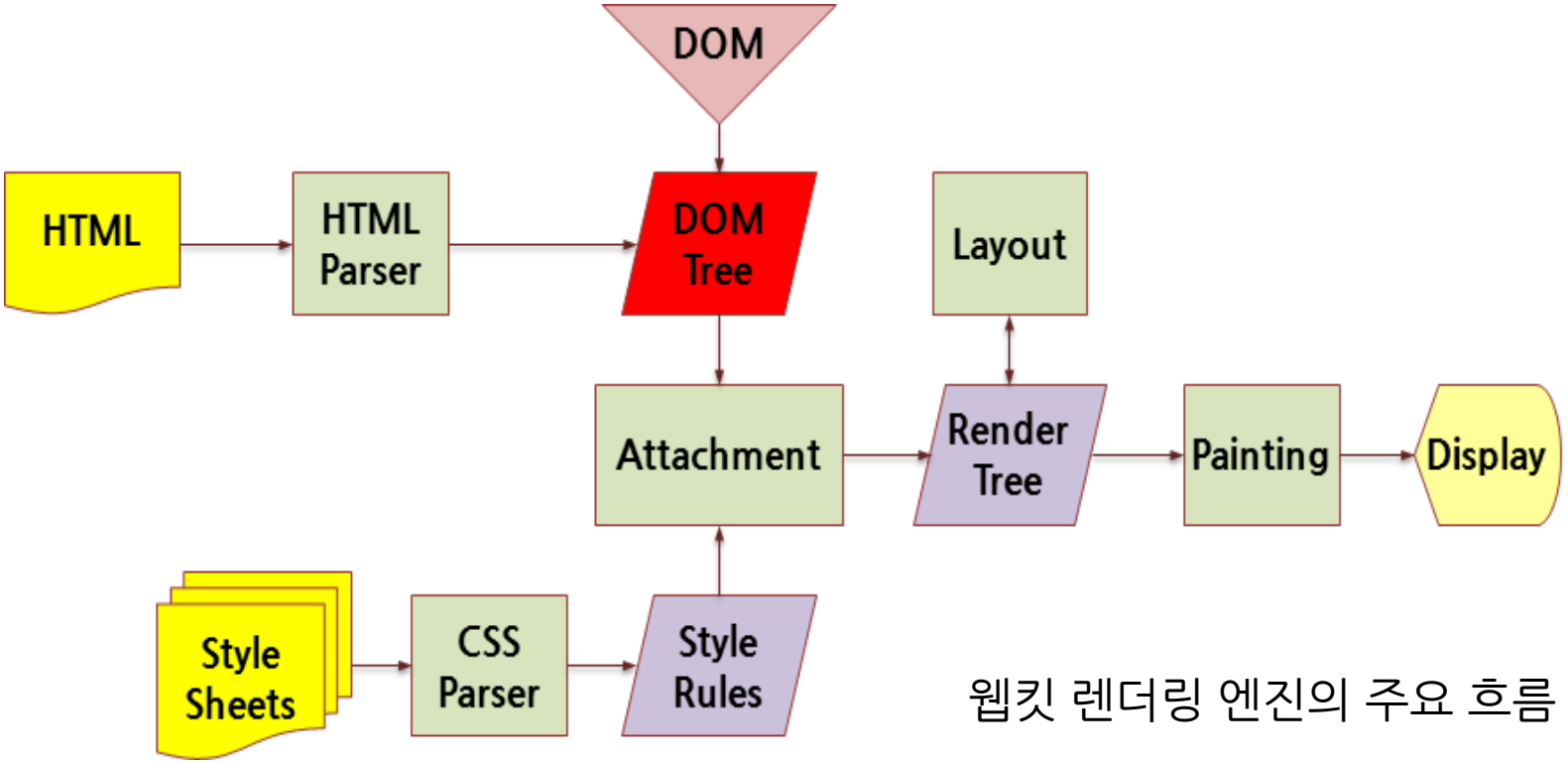
- 클래스(class) 선택자 (".class")
  - HTML 문서에서 class 속성의 값과 일치하는 요소를 선택
  - 선택자 이름 앞에 "."을 이용하여 선언.
  - soup.select("div.contents")

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

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

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



웹킷 렌더링 엔진의 주요 흐름

1) 상태 코드

- HTTP 응답 메시지의 상태 라인에는 상태 코드와 상태 메시지를 포함하고 있음

상태 코드	의미	내 용
100번 영역	정보전송	임시적인 응답을 나타내는 것은 Status-Line과 선택적인 헤더들로 구성되어 있고, 빈 줄로 끝을 맺는다.
200번 영역	성공	클라이언트의 요구가 성공적으로 수신되어 처리되었음을 의미 200(성공), 201(POST요청 처리), 204(전송할 데이터 없음)
300번 영역	리다이렉션	해당 요구사항을 처리하기 위해서는 사용자 에이전트에 의해 수행되어야 할 추가적인 동작이 있음을 의미
400번 영역	클라이언트 측 오류	클라이언트가 서버에게 보내는 요구 메시지를 완전히 처리하지 못한 경우와 같이 클라이언트에서 오류가 발생한 경우에 사용 401(사용자인증), 403(접근권한 없음), 404(요청한 URL 없음), 406(acceptable)
500번 영역	서버측 오류	서버 자체에서 발생된 오류상황이나 요구사항을 제대로 처리할 수 없을 때 사용.

2절. requests를 이용한 웹 데이터 수집

●requests 모듈 활용

- 그 외 다양한 함수들을 통해 홈페이지의 각종 정보들을 가져오거나 요청 할 수 있음
- Json 함수, 파라미터 여러 개, 헤더, 쿠키 값 전달, API 호출 등

- https://requests.readthedocs.io/en/latest/

requests

2절. requests를 이용한 웹 데이터 수집

- 파이썬에서 HTTP 요청을 만들기 위한 사실상의 표준

- 2절의 내용
  - 가장 일반적인 HTTP 메소드를 사용하여 요청하기
  - 쿼리 문자열 및 메시지 본문을 사용하여 사용자의 요청과 데이터를 변경하기
  - 요청 및 응답에서 데이터 검사하기
  - 인증 된 요청 만들기
  - 응용 프로그램이 백업 또는 속도 저하를 막을 수 있도록 요청을 구성하기

2.1. requests 모듈

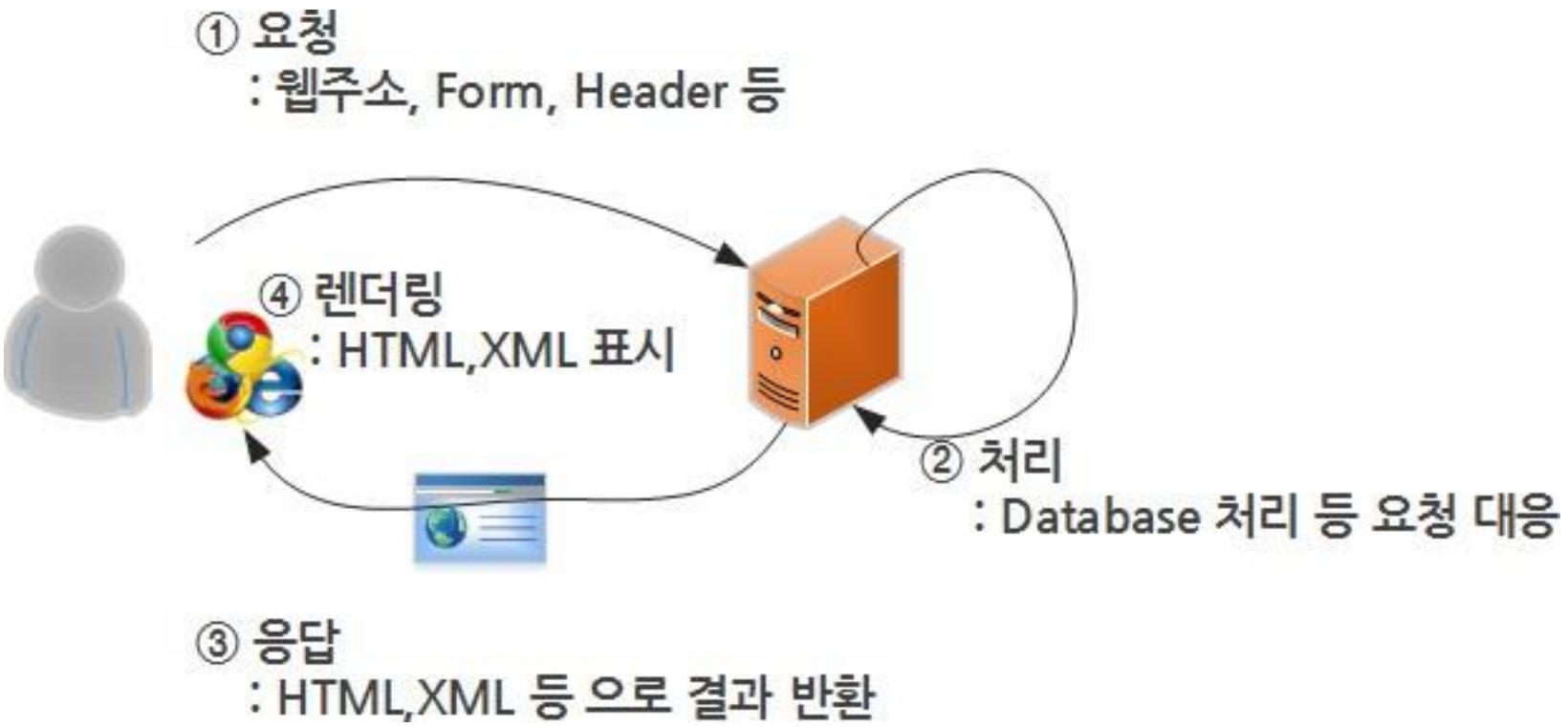
2절. requests를 이용한 웹 데이터 수집

- pip 명령으로 쉽게 설치
  - pip install requests

- requests가 설치되면 응용 프로그램에서 사용
  - import requests

2.1. requests 모듈

2절. requests를 이용한 웹 데이터 수집



2.2. GET 요청

- GET은 HTTP 요청(Request) 방식 중 하나
- requests.get() : 지정된 리소스에서 데이터를 가져옴

```
1  import requests
```

```
1  requests.get('https://api.github.com')
```

```
<Response [200]>
```

2.3. 응답 객체

2절. requests를 이용한 웹 데이터 수집

- 응답(Response)은 요청의 결과를 저장한 개체

```
1  response = requests.get('https://api.github.com')
```

```
1  response.
```

```
apparent_encoding
close
connection
content
cookies
elapsed
encoding
headers
history
is_permanent_redirect
```

1) 상태 코드

- HTTP 응답 메시지의 상태 라인에는 상태 코드와 상태 메시지를 포함하고 있음

상태 코드	의미	내 용
100번 영역	정보전송	임시적인 응답을 나타내는 것은 Status-Line과 선택적인 헤더들로 구성되어 있고, 빈 줄로 끝을 맺는다.
200번 영역	성공	클라이언트의 요구가 성공적으로 수신되어 처리되었음을 의미 200(성공), 201(POST요청 처리), 204(전송할 데이터 없음)
300번 영역	리다이렉션	해당 요구사항을 처리하기 위해서는 사용자 에이전트에 의해 수행되어야 할 추가적인 동작이 있음을 의미
400번 영역	클라이언트 측 오류	클라이언트가 서버에게 보내는 요구 메시지를 완전히 처리하지 못한 경우와 같이 클라이언트에서 오류가 발생한 경우에 사용 401(사용자인증), 403(접근권한 없음), 404(요청한 URL 없음)
500번 영역	서버측 오류	서버 자체에서 발생된 오류상황이나 요구사항을 제대로 처리할 수 없을 때 사용.

1) 상태 코드

2절. requests를 이용한 웹 데이터 수집 > 2.3. 응답 객체

```
1 response.status_code
```

200

```
1 if response.status_code == 200:
2     print('Success!')
3 elif response.status_code == 404:
4     print('Not Found.')
```

상태코드 정보를 사용하여 코드에서 의사 결정을 내릴 수 있음

Success!

Response 인스턴스를 사용하면 상태 코드가 200에서 400 사이 이면 True로 평가되고 그렇지 않으면 False로 평가

```
1 if response:
2     print('Success!')
3 else:
4     print('An error has occurred.')
```

Success!

2) Content

- GET 요청의 응답의 메시지 본문에는 중요한 정보가 포함되어있는 경우가 많음
- 응답 내용을 바이트 단위로 보려면 .content를 사용

```
1  response = requests.get('https://api.github.com')
2  response.content
```

```
b'{"current_user_url":"https://api.github.com/user","current_user
_authorizations_html_url":"https://github.com/settings/connection
s/applications{/client_id}","authorizations_url":"https://api.git
hub.com/authorizations","code_search_url":"https://api.github.co
m/search/code?q={query}{&page,per_page,sort,order}","commit_searc
h_url":"https://api.github.com/search/commits?q={query}{&page,per
```

2) Content

2절. requests를 이용한 웹 데이터 수집 > 2.3. 응답 객체

- .text 속성에 액세스 할 때 문자열 인코딩을 지정할 수 있음
- 내부적으로 추론해서 자동 지정됨

```
1  response = requests.get('https://api.github.com')
2  response.encoding = 'utf-8'
3  response.text
```

```
'{"current_user_url":"https://api.github.com/user","current_user_
authorizations_html_url":"https://github.com/settings/connection
```

```
1  response = requests.get('http://javaspecialist.co.kr')
2  response.content
```

```
b'<!DOCTYPE html>₩n<html manifest="index.manifest">₩n<head>₩r₩n
<meta charset="utf-8">₩r₩n    <meta name="viewport" content="widt
h=device-width, initial-scale=1.0">₩r₩n    <meta name="robots" co
ntent="index, follow">₩r₩n    <meta http-equiv="X-UA-Compatible"
content="IE=edge" >₩r₩n    <meta name="google-site-verification"
content="RPK2fDgygxeiosW4a4sE8zcla1193rlzAf90zQjTtvc" />₩r₩n
₩r₩n    <title>₩xec₩x9e₩x90₩xeb₩xb0₩x94₩xec₩xa0₩x84₩xeb₩xac₩xb8₩x
ea₩xb0₩x80₩xea₩xb7₩xb8₩xeb₩xa3₩xb9</title>₩r₩n    <!-- Favicon --
```

```
1  response = requests.get('http://javaspecialist.co.kr/emp')
2  response.text
```

```
'<!DOCTYPE html> ₩n<html>₩n<head>₩r₩n    <meta charset="utf-8">₩r
₩n    <meta name="viewport" content="width=device-width, initial-
scale=1.0">₩r₩n    <meta name="robots" content="index, follow">₩r
₩n    <meta http-equiv="X-UA-Compatible" content="IE=edge" >₩r₩n
<meta name="google-site-verification" content="RPK2fDgygxeiosW4a4
sE8zcla1193rlzAf90zQjTtvc" />₩r₩n    ₩r₩n    <title>자바전문가그
룹</title>₩r₩n    <!-- Favicon -->₩r₩n    <link href="/favicon.pn
```

2) Content

- .text 속성에 액세스 할 때 문자열 인코딩을 지정할 수 있음

```
1  response = requests.get('https://api.github.com')
2  response.json()
```

```
{'current_user_url': 'https://api.github.com/user',
 'current_user_authorizations_html_url': 'https://github.com/sett
ings/connections/applications{/client_id}',
 'authorizations_url': 'https://api.github.com/authorizations',
 'code_search_url': 'https://api.github.com/search/code?q={query}
{&page,per_page,sort,order}',
 'commit_search_url': 'https://api.github.com/search/commits?q={q
uery}{&page,per_page,sort,order}',
```

Bs4(BeautifulSoup)

- https://www.crummy.com/software/BeautifulSoup/bs4/doc/

- find & find_all
  - 태그와 class 및 속성으로 가져옴
- Select_one, select
  - CSS로 가져옴

-find find_all
 태그와 class혹은 다른 속성으로 가져옴
-select_one, select
 CSS로 가져옴

1) 환율 정보 가져오기


```
1 import requests
2 from bs4 import BeautifulSoup
```

```
1 url = 'https://finance.naver.com/marketindex/' # 환율 정보
2 market_index = requests.get(url)
```

```
1 market_index
```

```
<Response [200]>
```

```
1 soup = BeautifulSoup(market_index.content, "html.parser")
```

```
1 price = soup.select_one("div.head_info > span.value")
2 print(price)
```

```
<span class="value">1,189.70</span>
```

```
1 print("usd/krw=", price.text)
```

```
usd/krw= 1,189.70
```

2) 다음 검색


```python
import requests
from bs4 import BeautifulSoup
import time
import pandas as pd

# 딕셔너리 리스트로 반환하는 함수
def collect_list(keyword, page):
  url = f"https://search.daum.net/search?w=news&nil_search=btn&DA=PGD&enc=utf8&cluster=y&c
  response = requests.get(url)
  soup = BeautifulSoup(response.text, 'html.parser')
  items_findall_list = []
  items_findall = soup.find_all("div", class_="item-title")

  for idx, item in enumerate(items_findall):
    strong = item.find("strong")
    if strong:
      a = strong.find("a")
      items_findall_list.append({"no":page*10+idx+1, "title":a.text, "href":a['href']})
      #print([page*10+idx+1, a.text, a['href']])
  return items_findall_list
```

3절. Selenium을 이용한 웹 데이터 수집

https://selenium-python.readthedocs.io

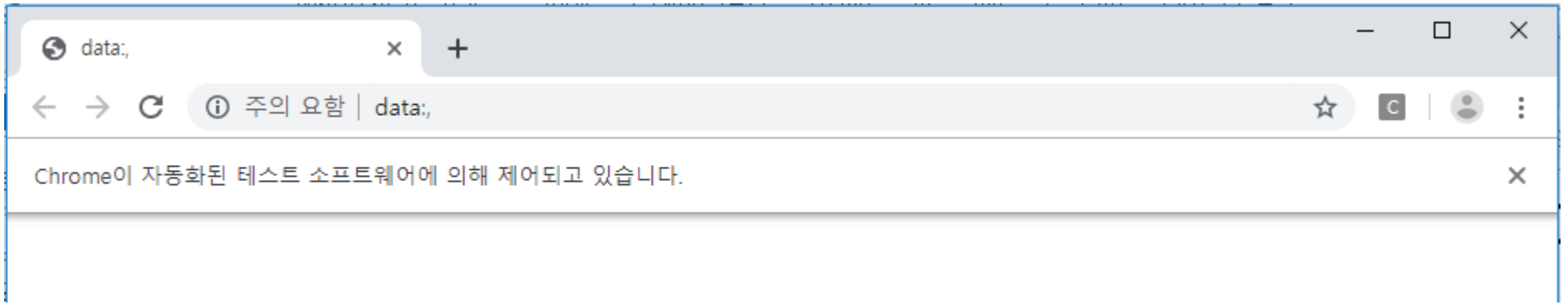
셀레니움 파이썬 바인딩

- 셀레니움(Selenium, https://selenium-python.readthedocs.io/index.html)은 브라우저의 동작을 자동화 해주는 프로그램
- 셀레니움 파이썬 바인딩(Selenium Python bindings)은 셀레니움 WebDriver(웹드라이버)를 사용하여 파이어폭스(Firefox), 인터넷 익스플로러(Ie), 크롬(Chrome) 등 브라우저에 접근하고 브라우저의 동작을 제어 할 수 있는 편리한 API를 제공
- 설치
  - pip install selenium

1) 웹드라이버 실행


```
1 from selenium import webdriver
```

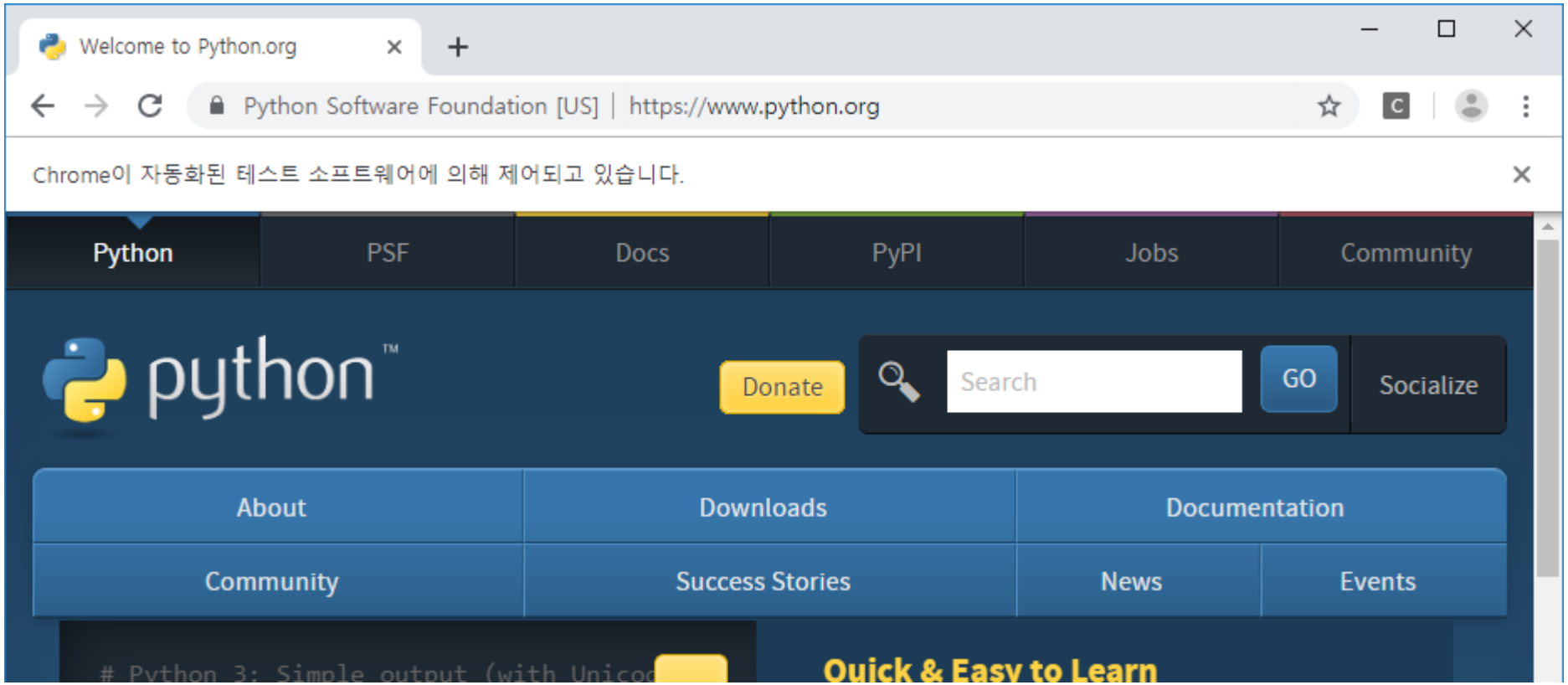
```
1 driver = webdriver.Chrome()
```



- 크롬 버전 확인
- 크롬 드라이버 OS에 맞게 다운로드
- 파이썬이 실행되는 경로에 압출 풀어서 드라이버 파일 넣어두기

2) 사이트 접속


```
1  driver.get("http://www.python.org")
```

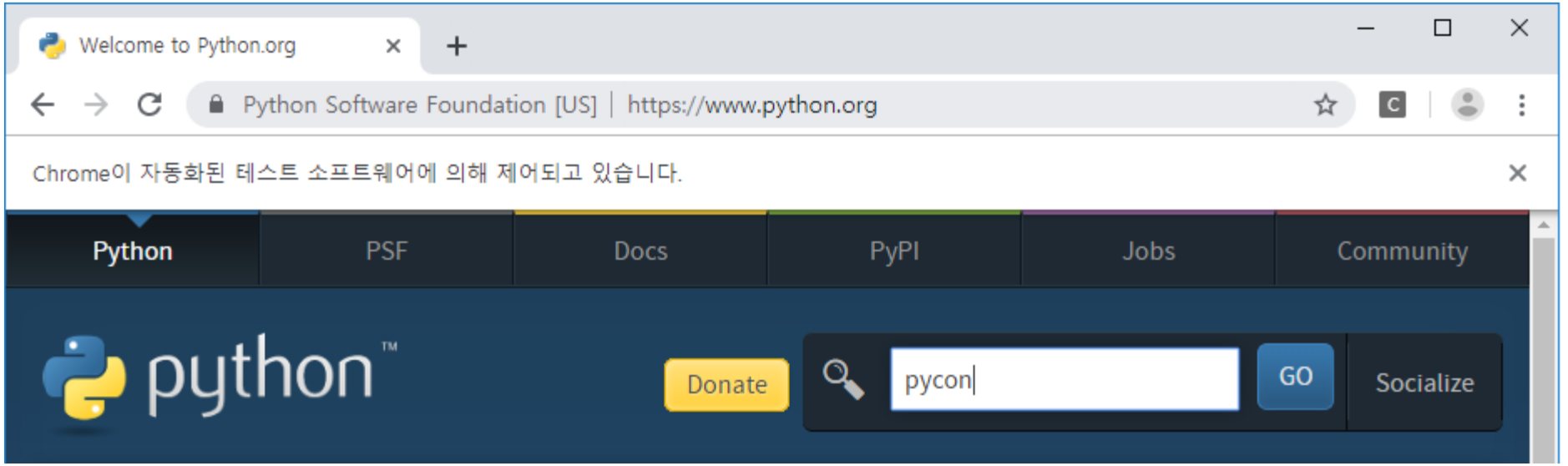


3) 입력양식 채우기

3절. Selenium을 이용한 웹 데이터 수집 > 3.2. 셀레니움 시작하기

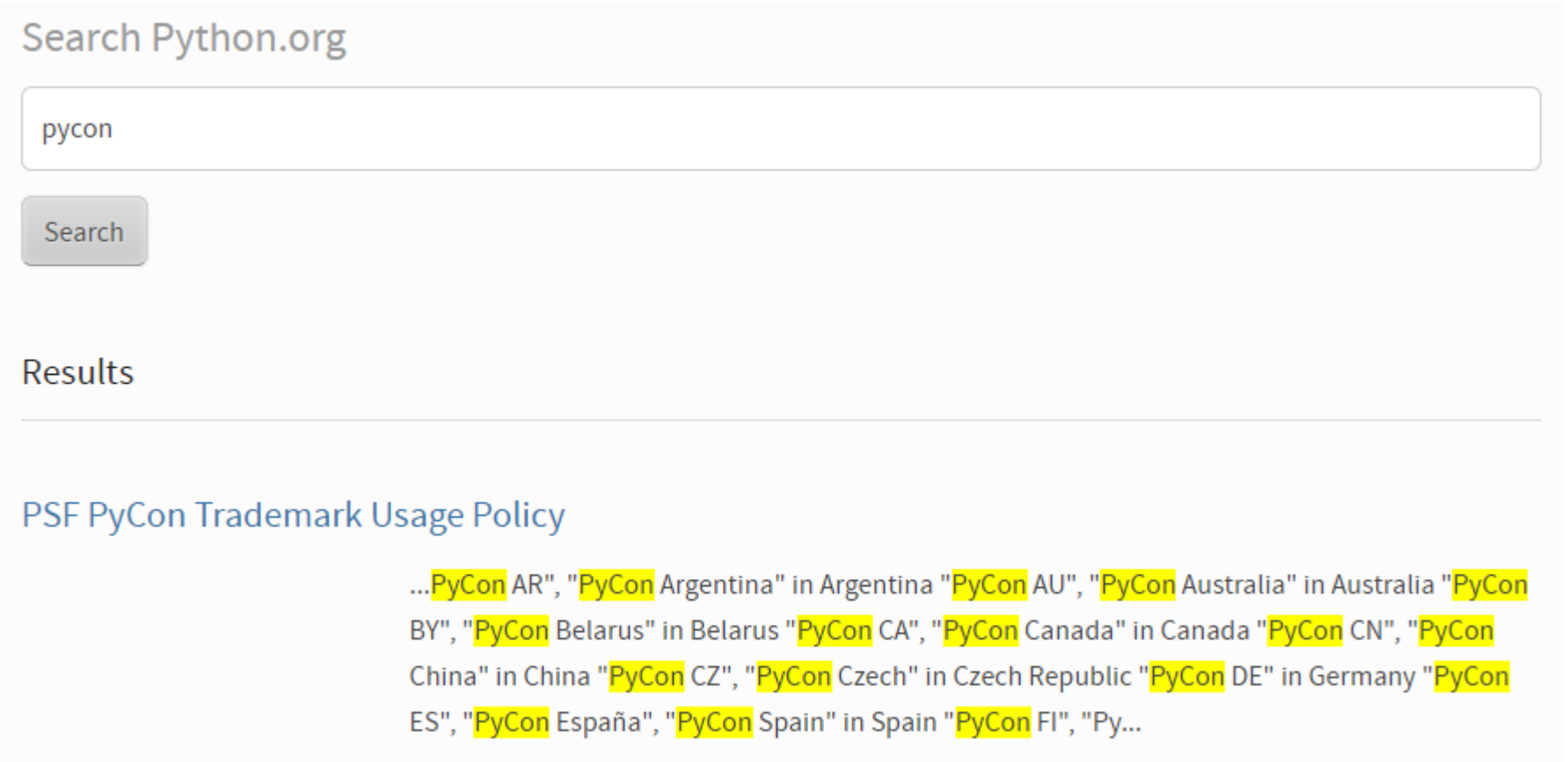
```
1 elem = dv.find_element(By.NAME, 'q')
2 elem.clear()
3 elem.send_keys('pycon')
```

```
<span aria-hidden="true" class="icon-search"></span>

<label class="screen-reader-text" for="id-search-field">Search This Site</label>
<input id="id-search-field" name="q" type="search" role="textbox" class="search-field" placeholder="Search" value="" tabindex="1">

<button type="submit" name="submit" id="submit" class="search-button" title="Submit this Search" tabindex="3">
    GO
</button>
```



4) 이벤트 처리하기


```
1 from selenium.webdriver.common.keys import Keys
2 elem.send_keys(Keys.RETURN)
```



5) 결과 찾아 출력하기


```
result_list = dv.find_elements(By.CSS_SELECTOR, 'li > h3 > a')
for result in result_list:
    print("%s - %s" % (result.text, result.get_attribute('href')))
```

```
PSF PyCon Trademark Usage Policy - https://www.python.org/psf/trademarks/pycon
PyCon Italia 2016 (PyCon Sette) - https://www.python.org/events/python-events/378/
PyCon PL 2014 - https://www.python.org/events/python-events/191/
PyCon Home at python.org - https://www.python.org/community/pycon
PyCon Australia 2013 - https://www.python.org/events/python-events/57/
PyCon AU 2019 - https://www.python.org/events/python-events/776/
PyCon Australia 2014 - https://www.python.org/events/python-events/10/
PyCon Ireland 2012 - https://www.python.org/events/python-events/76/
PyCon Ireland 2016 - https://www.python.org/events/python-events/429/
PyCon Ireland 2022 - https://www.python.org/events/python-events/1320/
PyCon Australia 2014 - https://www.python.org/events/python-events/1447/
PyCon AU 2018 - https://www.python.org/events/python-events/696/
PyCon APAC 2022 - https://www.python.org/events/python-events/1216/
PyCon PH 2024 - https://www.python.org/events/python-events/1661/
2012-07-16 PSF Board Meeting Minutes - https://www.python.org/psf/records/board/minutes/2012-07-16
PyCon Ireland 2023 - https://www.python.org/events/python-events/1568/
PyCon MY 2015 - https://www.python.org/events/python-events/313/
PyCon Ireland 2015 - https://www.python.org/events/python-events/333/
PyCon AU 2015 - https://www.python.org/events/python-events/273/
PyCon Australia 2016 - https://www.python.org/events/python-events/357/
```

6) 브라우저 종료


```
1 driver.close()
```

포트폴리오 1번 문제

1. yes24의 베스트셀러 정보를 제공하는 사이트에서 베스트셀러 정보를 수집해서 파일에 저장하세요.
   - 베스트셀러 정보 수집 주소 : http://www.yes24.com/24/category/bestseller

```
순위,책이름,저자및출판사,가격
1,흔한남매가 선사하는 유쾌한 우애,흔한남매 원저/백난도 글/유난희 그림/흔한컴퍼니 감수 | 미래엔아이세움,"10,800원"
2,"돌아온 피터슨, 다시 인생 법칙을 말하다 ",조던 B. 피터슨 저/김한영 역 | 웅진지식하우스,"16,020원"
3,"잠들면 열리는 비밀상점, 그곳에서 펼쳐지는 힐링 판타지",이미예 저 | 팩토리나인,"12,420원"
4,전천당에 행운의 손님이 등장했다!,히로시마 레이코 글/쟈쟈 그림/김정화 역 | 길벗스쿨,"10,800원"
5,주린이들을 위한 안전판! 염블리표 투자 바이블,염승환 저 | 메이트북스,"16,200원"
6,흔한남매의 초등 어휘력 학습만화 ! ,흔한남매 원저/한은호 글/유희석 그림/흔한컴퍼니 감수 | 다산어린이,"11,700원"
7,우리 땅 독도를 지켜낸 빛나는 영웅들,"설민석,스토리박스 글/정현희 그림/태건 역사 연구소 감수 | 아이휴먼","10,800원"
8,"취득부터 임대, 양도, 증여까지. 주택 세금의 모든 것!",국세청 저 | 국세청,"7,000원"
9,국민 육아멘토 오은영 박사의 현실밀착 육아회화,오은영 저/차상미 그림 | 김영사,"15,750원"
10,"10년 후, 지금보다 더 거대한 변화가 우리 앞에 온다",마우로 기옌 저/우진하 역 | 리더스북,"16,200원"
```

제출파일 : 1_본인이름소스.ipynb, 1_소스및실행결과.html
1_yes24.csv나 1_yes24.txt

포트폴리오 2번 문제

2. 영어번역 자동화 프로그램을 구현하시오. 네이버 맞춤법 검사를 마친 텍스트를 https://translate.kakao.com/를 통해 영어로 번역된 텍스트를 파일로 출력하는 동적 웹크롤링 프로그램을 구현하시오.

제출파일 : 2_본인이름소스.ipynb, 2_실행결과및소스.html

2_맞춤법후.txt
2_자동화영어번역본.txt