Lab. urllib 사용하기

-Tools
  --Microsoft Visual Studio Code

1. Web Scraping이란?
  1)Web 문서로부터 필요한 정보만을 추출하여 제공하는 기술을 말한다.
  2)이 기술을 통해 상품 Catalog를 제작하거나 News 기사, Blog나 Cafe의 게시물, 회사의 profile과 금융 data 등을 수집할 수 있다.
  3)그러기 위해서는 먼저 추출하고자 하는 정보들이 구성되어 있는 영역을 확인해야 한다.

2. Internet에서 data 수집하기
  1)Data 수집은 data 준비 절차의 첫 단계로 어떤 형식의 data를 수집할지, 어떤 방법과 경로로 data를 수집할 것인가를 고민해야 한다.
  2)수집된 data의 품질은 data 분석 결과에 많은 영향을 미치며, data가 얼마나 잘 정제되어 있는지에 따라서도 전체 data 분석에 드는 시간과 노력이 좌우되기 때문에 data 수집 과정은 매우 중요한 단계라 할 수 있다.

3. Web site에서 사용된 web 기술 확인
  1)해당 web site에 사용된 web 기술을 확인하는 유용한 도구는 builtwith module이다.
  2)Install
    $ pip install builtwith
  3)이 module은 전달된 URL을 가진 web site를 download하고 분석할 것이다.
  4)분석이 되면 web site에 사용된 기술을 알려준다.

```
    import builtwith
    builtwith.parse('http://webscraping.com')
    ----------------------------------------------------------
    {'web-servers': ['Nginx'], 'javascript-frameworks': ['Modernizr', 'jQuery'],
     'web-frameworks': ['Twitter Bootstrap']}
```

4. Web site 소유자 찾기
  1)website의 소유자를 찾으려면 해당 website의 domain명에 누가 등록됐는지 확인하는 whois protocol을 사용할 수 있다.
  2)Install
    $ pip install python-whois

```
    import whois
    print(whois.whois('appspot.com')
    ----------------------------------------------------------
    {'domain_name': ['APPSPOT.COM', 'appspot.com'],
     'registrar': 'MarkMonitor, Inc.',
     'whois_server': 'whois.markmonitor.com',
     'referral_url': None,
     'updated_date': [datetime.datetime(2019, 2, 6, 10, 33, 49),
      datetime.datetime(2019, 2, 6, 2, 33, 49)],
     'creation_date': [datetime.datetime(2005, 3, 10, 2, 27, 55),
      datetime.datetime(2005, 3, 9, 18, 27, 55)],
```

```
 'expiration_date': [datetime.datetime(2020, 3, 10, 1, 27, 55),
  datetime.datetime(2020, 3, 9, 0, 0)],
 'name_servers': ['NS1.GOOGLE.COM',
  'NS2.GOOGLE.COM',
  'NS3.GOOGLE.COM',
  'NS4.GOOGLE.COM',
  'ns1.google.com',
  'ns4.google.com',
  'ns3.google.com',
  'ns2.google.com'],
 'status': ['clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited',
  'clientTransferProhibited https://icann.org/epp#clientTransferProhibited',
  'clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited',
  'serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited',
  'serverTransferProhibited https://icann.org/epp#serverTransferProhibited',
  'serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited',
  'clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)',
  'clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)',
  'clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)',
  'serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)',
  'serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)',
  'serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)'],
 'emails': ['abusecomplaints@markmonitor.com', 'whoisrequest@markmonitor.com'],
 'dnssec': 'unsigned',
 'name': None,
 'org': 'Google LLC',
 'address': None,
 'city': None,
 'state': 'CA',
 'zipcode': None,
 'country': 'US'}
```

5. Website crawling with urllib module

1)Web site를 scrap을 하려면 먼저 crawling 이라고 알려진 과정, 즉 관심 있는 data를 가진 web page를 download할 필요가 있다.

2)Web site crawling을 할 수 있는 몇 가지 방법이 있으며, 대상 web site 구조에 맞춰 적절히 선택한다.

3)Web site download

-Web page를 crawling하려면 우선 web page를 download 해야 한다.

-다음은 web 주소가 URL인 web site를 download하고자 Python의 urllib3 module을 사용하는 간단한 Python script이다.

```
from urllib.request import urlopen

def download(url):
    return urlopen(url).read().decode('utf-8')

print(download('https://www.sesoc.global'))
```

-이 함수는 URL이 전달됐을 때 web page를 download하고 HTML을 반환한다.

-이 source code는 web page를 download할 때 처리가 안되는 오류에 직면할 수도 있는 문제가 있다.

-이를 테면 요청한 page가 존재하지 않는 것이 그런 사례이다.

-그럴 경우 urllib3는 예외를 발생한 후 script의 실행을 멈춘다.

-그럴 경우를 대비해서 아래의 code로 변경하자.

```
from urllib.request import urlopen
from urllib.error import HTTPError

def download(url):
  print('downloading:', url)
  try :
    html = urlopen(url).read().decode('utf-8')
  except HTTPError as e:
    print('Download error:', e.reason)
    html = None
  return html
```

-이제 download 오류 상황이 되면 예외 처리가 되고 None을 반환한다.
-이 기능을 test하려면 500 오류를 반환하는 http://httpstat.us/500 page를 download하면 된다.

```
download('http://httpstat.us/500')
-------------------------------------------
Downloading: http://httpstat.us/500
Download error: Internal Server Error

download('http://www.samsung.com')
-----------------------------------------------
Downloading: http://www.samsung.com
Download error: Forbidden
```

6. Crawling & Scraping

1)Crawling

-Web Site의 data를 그대로 취득하는 것

2)Scraping

-Crawling하여 모든 data에서 필요한 것만 추출하거나 변환하는 처리를 포함.

3)정규식

-정규식은 변경 사항에 대해 유연하게 대처할 수는 있지만 만들기 어렵고 가독성도 떨어진다.
-따라서 적용하기에 너무 취약하고 web page가 바뀌면 사용하는 정규식이 쉽게 무용지물이 된다는 점의 문제점이 있다.

4)pandas.read_html() 함수 사용하기

-다음의 third-party package들의 설치여부를 확인한다.
  --html5lib, lxml, BeautifulSoup4
-HTML file의 table 요소를 DataFrame으로 불러온다.
-DataFrame이 들어 있던 list가 return된다.
-Table 요소가 여러 개 있는 경우, 여러 개의 DataFrame이 저장된다.

```
import pandas as pd

url = 'http://www.fdic.gov/bank/individual/failed/banklist.html'

dfs = pd.read_html(url)

dfs[0].info()
-------------------------------------------------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 555 entries, 0 to 554
Data columns (total 7 columns):
Bank Name              555 non-null object
City                   555 non-null object
ST                     555 non-null object
CERT                   555 non-null int64
Acquiring Institution    555 non-null object
Closing Date           555 non-null object
Updated Date           555 non-null object
dtypes: int64(1), object(6)
memory usage: 30.4+ KB
```

7. urllib 이용하기

1)내장 module이기 때문에 설치할 필요가 없다.

2)urlopen()

-urlopen()에 url을 지정하면 web page를 추출할 수 있다.

```
from urllib.request import urlopen, quote
google = urlopen('http://google.com')
google = google.read()
#print(google[:200])

url = 'http://google.com?q='

#quote() : 질의 문자열의 빈칸을 +기호로 대체한다.
#google 검색기에서는 문자열 사이의 빈칸이 +기호로 표현되어야 하기 때문.
url_with_query = url + quote('python web scraping')

web_search = urlopen(url_with_query)
web_search = web_search.read()
print(web_search[:200])

----------------------------------------------------------------------
from urllib.request import urlopen, Request
url =
'http://www.kyobobook.co.kr/bestSellerNew/bestseller.laf?mallGb=KOR&linkClass=e&range=1&kind=0&orderClick=DAb'

req = Request(url)
page = urlopen(req)
print(page)
print(page.code)
print(page.headers)
print(page.url)
print(page.info().get_content_charset())
print(page.read().decode('euc-kr'))
```

3)read() 사용하기

-HTML을 binary 형태로 가져온다.

-read()로 추출할 수 있는 응답 본문의 값은 bytes 자료형이므로 문자열(str)로 다루려면 문자 code를 지정해서 decoding 해야 한다.

```
208
209         from urllib.request import urlopen, Request
210         url =
            'http://www.kyobobook.co.kr/bestSellerNew/bestseller.laf?mallGb=KOR&linkClass=e&range
            =1&kind=0&orderClick=DAb'
211
212         req = Request(url)
213         page = urlopen(req)
214         print(page)
215         print(page.code)
216         print(page.headers)
217         print(page.url)
218         print(page.info().get_content_charset())
219         print(page.read())
220
221       -최근에는 HTML5의 기본 encoding 방ㅅ기인 utf-8로 작성된 web page가 많으므로 UTF-8을 전체로 decoding
          하는 것이 좋다.
222       -하지만 한국어를 포함하는 site를 crawling할 때에는 여러 개의 encoding이 섞여 있을 수도 있으므로 HTTP
          header를 참조해서 적절한 encoding방식으로 decoding 해야 한다.
223       -HTTP header에서 encoding 방식 추출하기
224         --HTTP response의 Content-Type header를 참조하면 해당 page에서 사용되고 있는 encoding 방식을 알아
            낼 수 있다.
225
226         text/html
227         text/html;charset=UTF-8
228         text/html;charset=EUC-KR
229
230         --HTTPMessage 객체의 get_content_charset()을 사용하면 더 쉽게 encoding을 추출할 수 있다.
231
232           import sys
233           from urllib.request import urlopen
234           f = urlopen('http://www.hanbit.co.kr/store/books/full_book_list.html')
235
236           # HTTP 헤더를 기반으로 인코딩 방식을 추출(명시돼 있지 않을 경우 utf-8을 사용한다).
237           encoding = f.info().get_content_charset(failobj="utf-8")
238
239           # encoding 방식을 표준 오류에 출력.
240           print('encoding:', encoding, file=sys.stderr)
241
242           # 추출한 encoding 방식으로 decoding한다.
243           text = f.read().decode(encoding)
244
245           print(text)
246
247
248     4)데이터 요청하기
249       -urllib는 Request() 함수를 이용하여 요청 객체를 만들 때 두 번째 인자에는 데이터, 세 번째 인자에는 header가 들
          어간다.
250       -만약 두 번째 인자 값이 존재한다면 POST 요청, 존재하지 않는다면 GET 요청을 보낸다.
251       -즉 두 번째 인자 값의 존재 여부에 따라 GET 요청인지 POST 요청인지 결정된다.
252       -data를 만들 때 encode() 함수를 이용하여 binary 형태로 encode 해서 보내야 한다.
253
254       from urllib.request import urlopen, Request
255       import urllib
```

```
256
257     url = "http://blog.naver.com/pjt3591oo"
258
259     # post 요청 시 보낼 데이터 만들기
260     data = {'key1': 'value1', 'key2': 'value2'}
261     data = urllib.parse.urlencode(data)
262     data = data.encode('utf-8')
263
264     print(data)
265
266     # post 요청
267     req_post = Request(url, data=data, headers={}) # 2번째 인자 데이터, 세 번째 인자 헤더
268     page = urlopen(req_post)
269
270     print(page)
271     print(page.url)
272
273     # get 요청
274     req_get = Request(url+"?key1=value1&key2&value2", None, headers={}) # 2번째 인자 데이터,
        세 번째 인자 헤더
275     page = urlopen(req_get)
276
277     print(page)
278     print(page.url)
279
280
281   4)없는 page 요청하기
282
283       from urllib.request import urlopen, Request
284
285       url = "https://pjt3591oo.github.io/1"
286
287       req_post = Request(url)
288       page = urlopen(req_post)
289
290       print(page)
291       print(page.url)
292       ---------------------------------
293       urllib.error.HTTPError: HTTP Error 404 : Not Found
294
295
296   5)urllib와 외부 module인 requests와의 차이점
297     a. requests와 urllib는 요청 시 요청 객체를 만드는 방법에 차이가 있다.
298     b. data를 보낼 때 requests는 dict 형태로 보내고, urllib는 encode하여 binary 형태로 전송한다.
299     c. requests는 요청 method(GET, POST)를 명시하지만, urllib는 data의 여부에 따라 GET 요청과 POST 요청을
        구분한다.
300     d. 없는 page 요청 시 requests는 error를 띄우지 않지만, urllib는 error를 띄운다.
301     -이런 이유로 urlib보다는 requests 외부 module을 자주 이용하게 된다.
302
303
304
305 8. Lab. urllib
306   -기상청의 RSS 서비스 사용하기
307
```

```
308   import urllib.request
309   import urllib.parse
310
311   API = "http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"
312
313   # 매개변수를 URL 인코딩.
314   values = {
315      'stnId': '108'              #기상 정보를 알고 싶은 지역 지정
316      #전국(108), 서울/경기(109), 강원(105), 충북(131), 충남(133), 전북(146), 전남(156), 경북(143), 경남
         (159), 제주(184)
317   }
318   params = urllib.parse.urlencode(values)
319
320   # 요청 전용 URL을 생성.
321   url = API + "?" + params
322   print("url=", url)
323
324   # 다운로드.
325   data = urllib.request.urlopen(url).read()
326   text = data.decode("utf-8")
327   print(text)
328
329
330
331 9. Lab. urllib
332   - command 창에서 입력한 parameter를 이용한 접근
333   import sys
334   import urllib.request as req
335   import urllib.parse as parse
336
337   # 명령줄 매개변수 추출
338   if len(sys.argv) <= 1:
339      print("USAGE: filename <Region Number>")
340      sys.exit()
341
342   regionNumber = sys.argv[1]
343
344   # 매개변수를 URL 인코딩.
345   API = "http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"
346   values = {
347      'stnId': regionNumber
348   }
349   params = parse.urlencode(values)
350   url = API + "?" + params
351   print("url=", url)
352
353   # 다운로드.
354   data = req.urlopen(url).read()
355   text = data.decode("utf-8")
356   print(text)
357
358
359
360 10. 정규표현식으로 Scraping
```

```
from urllib.request import urlopen
import re
from html import unescape

hanbit = urlopen('http://www.hanbit.co.kr/store/books/full_book_list.html')
html = hanbit.read().decode()

# re.findall()을 사용해 도서 하나에 해당하는 HTML을 추출한다.
for partial_html in re.findall(r'<td class="left"><a.*?</td>', html, re.DOTALL):
    # 도서의 URL을 추출합니다.
    url = re.search(r'<a href="(.*?)">', partial_html).group(1)
    url = 'http://www.hanbit.co.kr' + url

    # Tag를 제거해서 도서의 제목을 추출한다.
    title = re.sub(r'<.*?>', '', partial_html)
    title = unescape(title)
    print('url:', url)
    print('title:', title)
    print('---')
```

11. Python으로 Scraping하여 Sqlite3에 저장하기

```
import re
import sqlite3
from urllib.request import urlopen
from html import unescape

def main():
    html = fetch('http://www.hanbit.co.kr/store/books/full_book_list.html')
    books = scrape(html)
    save('books.db', books)

def fetch(url):
    f = urlopen(url)

    # HTTP 헤더를 기반으로 encoding 형식 추출.
    encoding = f.info().get_content_charset(failobj="utf-8")

    # 추출한 encoding 형식을 기반으로 문자열 decoding.
    html = f.read().decode(encoding)
    return html

def scrape(html):
    books = []

    # re.findall()을 사용해 도서 하나에 해당하는 HTML을 추출.
    for partial_html in re.findall(r'<td class="left"><a.*?</td>', html, re.DOTALL):
        # 도서의 URL 추출.
        url = re.search(r'<a href="(.*?)">', partial_html).group(1)
        url = 'http://www.hanbit.co.kr' + url

        # tag를 제거해서 도서의 제목 추출.
```

```
            title = re.sub(r'<.*?>', '', partial_html)
            title = unescape(title)
            books.append({'url': url, 'title': title})

        return books

    def save(db_path, books):
        """
        매개변수 books로 전달된 도서 목록을 SQLite 데이터베이스에 저장.
        데이터베이스의 경로는 매개변수 dp_path로 지정한다.
        반환값: None(없음)
        """

        # Database 열고 연결다.
        conn = sqlite3.connect(db_path)

        # cursor 추출.
        c = conn.cursor()

        # execute()로 SQL 실행.
        # Script를 여러 번 실행할 수 있으므로 기존의 books table 제거.
        c.execute('DROP TABLE IF EXISTS books')

        # books table을 생성.
        c.execute('''
            CREATE TABLE books (
                title text,
                url text
            )
        ''')

        # executemany() 메서드를 사용하면 매개변수로 list를 지정할 수 있다.
        c.executemany('INSERT INTO books VALUES (:title, :url)', books)

        # 변경사항 commit.
        conn.commit()

        # 연결 종료.
        conn.close()

    # python 명령어로 실행한 경우 main() 함수를 호출.
    # 이는 모듈로써 다른 파일에서 읽어 들였을 때 main() 함수가 호출되지 않게 하는 것이다.
    # python 프로그램의 일반적인 작성 방식.
    if __name__ == '__main__':
        main()
```

12. Web site의 data file 읽기
  1)Web site에 있는 data set를 Python으로 loading 하는 방법을 알아보자.
  2)다음은 Titanic data file을 CSV file로 저장하는 방법을 소개한다.
  3)아래의 site에 접속해 보자.
    -https://github.com/vincentarelbundock/Rdatasets/tree/master/csv/datasets
    -https://vincentarelbundock.github.io/Rdatasets/datasets.html

4)Item(datasets에서 찾는다) 중 'Titanic'을 찾아보자.
-Link의 속성을 파악하기 위해서는 Google Chrome보다는 Internet Explorer를 이용하는 것이 좋다.
-Internet Explorer로 해당 site를 방문한 다음, datasets의 Item 중 CSV의 link를 Mouse 오른 Click를 한 후, [속성]을 선택한다.
-속성 창에서 Data set file의 URL을 확인하자.

```
import pandas as pd
url = 'https://vincentarelbundock.github.io/Rdatasets/csv/datasets/Titanic.csv'
df = pd.read_csv(url)
df.head()
```

13. 서울시 구별 CCTV 현황 분석
1)https://github.com/PinkWink/DataScience/blob/master/source_code/01.%20Basic%20of%20Python%2C%20Pandas%20and%20Matplotlib%20%20via%20analysis%20of%20CCTV%20in%20Seoul.ipynb