```
3
    1. requests module
       1)Python HTTP for Humans'
 5
       2)Python에서 웹 데이터를 받아올 때 가장 많이 사용하는 module이다.
 6
       3)http://docs.python-requests.org/en/master/
 7
       4)Install
 8
         $ pip install requests
 9
10
11
       5)웹 페이지에 접속하기
12
         -요청하면 서버가 응답한 값을 반환한다.
13
         -바로 응답 코드를 확인할 수 있다.
14
         -또는 status__code 속성을 통해서도 응답 코드를 확인할 수 있다.
15
16
            import requests
17
            url = 'https://www.naver.com'
            naver = requests.get(url)
19
            print(naver)
20
21
            <Response [200]>
22
            import requests
23
24
            url = 'https://www.naver.com'
25
            naver = requests.get(url)
26
            print(naver)
27
            print(naver.status_code)
28
29
            <Response [200]>
30
            200
31
32
33
      6)요청 페이지를 찾을 수 없을 때 404코드를 반환한다.
34
35
         import requests
36
37
         def url_check(url):
38
             res = requests.get(url)
39
40
             print(res)
41
42
             sc = res.status code
43
44
             if sc == 200:
45
                print("%s 요청성공"%(url))
46
             elif sc == 404:
47
                print("%s 찾을 수 없음" %(url))
48
             else:
                print("%s 알수 없는 에러 : %s"%(url, sc))
49
50
51
         url_check("https://www.naver.com")
52
53
         url_check("https://www.naver.com//a")
54
55
         <Response [200]>
56
         https://www.naver.com 요청성공
57
          <Response [404]>
58
         https://www.naver.com//a 찾을 수 없음
59
60
61
       7)header 가져오기
62
63
         import requests
64
65
         url = "https://www.naver.com"
66
67
         res = requests.get(url)
68
69
         print(res)
70
         print(res.headers)
71
72
         <Response [200]>
73
         {'Server': 'NWS', 'Date': 'Tue, 27 Aug 2019 11:28:32 GMT', 'Content-Type': 'text/html; charset=UTF-8',
          'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'Set-Cookie':
         'PM_CK_loc=e28d849e90b7a5d7c3ee6e69e31d9ba3a3ac7923a7b342c194bb9591a177e4a7; Expires=Wed, 28 Aug 2019
         11:28:32 GMT; Path=/; HttpOnly', 'Cache-Control': 'no-cache, no-store, must-revalidate', 'Pragma': 'no-cache', 'P3P':
         'CP="CAO DSP CURa ADMa TAIa PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT DEM STA PRE"',
         'X-Frame-Options': 'DENY', 'X-XSS-Protection': '1; mode=block', 'Content-Encoding': 'gzip', 'Strict-Transport-Security':
         'max-age=63072000; includeSubdomains', 'Referrer-Policy': 'unsafe-url'}
74
75
76
       8)응답 객체에서 header를 dict 형태로 가져온다.
77
```

Lab. Using requests Module

78

import requests

```
79
 80
          url = "https://www.naver.com"
 81
          res = requests.get(url)
 82
          print(res)
 83
 84
          headers = res.headers
85
          print(headers['Set-Cookie'])
 86
 87
          <Response [200]>
          PM_CK_loc=e28d849e90b7a5d7c3ee6e69e31d9ba3a3ac7923a7b342c194bb9591a177e4a7; Expires=Wed, 28 Aug 2019
 88
          11:30:28 GMT; Path=/; HttpOnly
 89
90
 91
          import requests
 92
 93
          url = "https://www.naver.com"
 94
          res = requests.get(url)
 95
          print(res)
 96
 97
          headers = res.headers
 98
99
          for header in headers:
100
             print(headers[header])
101
102
          <Response [200]>
103
          NWS
104
          Tue, 27 Aug 2019 11:31:58 GMT
105
          text/html; charset=UTF-8
106
          chunked
107
          keep-alive
          PM_CK_loc=e28d849e90b7a5d7c3ee6e69e31d9ba3a3ac7923a7b342c194bb9591a177e4a7; Expires=Wed, 28 Aug 2019
108
          11:31:58 GMT; Path=/; HttpOnly
109
          no-cache, no-store, must-revalidate
110
          no-cache
          CP="CAO DSP CURa ADMa TAIa PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT DEM STA PRE"
111
          DENY
112
113
          1; mode=block
114
          gzip
          max-age=63072000; includeSubdomains
115
116
          unsafe-url
117
118
       9)HTML code 보기
119
120
121
          import requests
122
          url = 'https://www.naver.com'
123
          naver = requests.get(url)
124
          print(naver.text)
125
126
127
       10)HTML code 보기2
128
          -content 속성을 사용하면 한글을 binary 형태(인코딩)로 바꿔서 가져온다.
129
          -binary 형태로 HTML을 가져올 경우 text속성을 이용하였을 때 발생하는 한글 문자 깨지는 현상을 방지할 수 있다.
130
131
            import requests
132
            url = 'https://www.naver.com'
133
            naver = requests.get(url)
134
            print(naver.content)
135
136
137
       11)인코딩 확인
138
139
          import requests
140
          url = 'https://www.naver.com'
141
          naver = requests.get(url)
142
          print(naver.encoding)
143
144
          UTF-8
145
146
       12)Data 보내기
147
148
          -requests로 요청할 때 data를 실어 보낼 수 있다.
149
          -querystring 같은 경우 URL에 직접 표현할 수 있지만, querystring을 만들어야 하는 번거로움이 있다.
150
          -하지만 data를 dict 형태로 만들어 보내는 방식으로 번거로움을 줄일 수 있다.
151
          -data 뿐만 아니라 header, cookie 같은 data도 원하는 값으로 변경하여 요청 가능하다.
152
          -특정 page는 header의 user-agent가 비었거나, cookie가 비어있을 경우 정상적으로 HTML 처리에 문제가 있을 수 있다.
153
          -이럴 때는 header나 cookie를 직접 만들어야 한다.
154
          -querystring data를 만들어 요청하기
155
156
            import requests
157
            url = "https://pjt3591oo.github.io/"
158
            res = requests.get(url, params={"key1": "value1", "key2": "value2"})
159
            print(res.url)
160
                           _____
```

```
161
                            https://pjt3591oo.github.io/?key1=value1&key2=value2
162
163
164
                 13)get()함수는 HTTP method의 GET에 대응된다.
                       -post(), put(), delete(), head(), options()는 각각 POST, PUT, DELETE, HEAD, OPTIONS에 대응된다.
165
166
167
                 14)Session 객체
168
                       -여러 개의 page를 연속으로 crawling할 때는 효과적이다.
169
                       -HTTP header 또는 Basic 인증 등의 설정을 한 번만 하고 여러번 재사용 가능하다.
170
171
172
           2. pprint module 이용하기
173
                 1)대량의 data를 보기 쉽게 표시해주는 표준 module
174
                 2)pprint(prettyprint)
175
176
                       import requests
177
                       import pprint
178
                       url = 'https://www.naver.com'
179
                       naver = requests.get(url)
180
                       pprint.pprint(naver.text)
181
182
183
184
           3. requests를 사용하여 API에 접근하기
185
                 1)기상청 RSS(https://www.weather.go.kr/w/pop/rss-quide.do)를 이용하자.
186
                 2)2021년 동네 예보 RSS
187
                       https://www.weather.go.kr/w/resources/pdf/dongnaeforecast rss.pdf
                 3)주소선택후 rss button click하면 zone을 알 수 있다.
188
189
                       -예:서울특별시 강남구 역삼oldsymbol{1}동
190
                       http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=1168064000
191
192
                       import requests
193
                       import pprint
194
                       api_uri = 'http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=1168064000'
195
                       weather_data = requests.get(api_uri).text
196
                       pprint.pprint(weather_data)
197
198
                       ('<?xml version="1.0" encoding="UTF-8" ?>\n'
                         '<rss version="2.0">\n'
199
200
                        '<channel>\n'
                        '<title>기상청 동네예보 웹서비스 - 서울특별시 강남구 역삼1동 도표예보</title>\n'
201
                        '<link>http://www.kma.go.kr/weather/main.jsp</link>\n'
202
203
                        '<description>동네예보 웹서비스</description>\n'
204
                        '<language>ko</language>\n'
                        '<generator>동네예보</generator>\n'
205
                        '<pubDate>2023년 02월 13일 (월)요일 14:00</pubDate>\n'
206
207
                        ' <item>\n'
                        '<author>기상청</author>\n'
208
209
                        '<category>서울특별시 강남구 역삼1동</category>\n'
210
                        '<title>동네예보(\mathtt{SE}) : 서울특별시 강남구 역삼1동
211
                        '[X=61,Y=125]</title><link>http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST&amp;dongCod
                        e=1168064000</link>\n'
212
                        '<guid>http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST&amp;dongCode=1168064000</guid
                        >\n'
213
                        '<description>\n'
214
                        ...
215
216
217
218
                 4)get method의 params option을 활용하기
219
220
                       api_url = 'http://www.kma.go.kr/wid/queryDFSRSS.jsp'
221
                            payload = {'zone': '1168064000'}
222
223
224
                            weather_data = requests.get(api_url, payload).text
225
226
                            weather_data
227
                            '<?xml version="1.0" encoding="UTF-8" ?>\n<rss version="2.0">\n<channel>\n<title>기상청 동네예보 웹서비스 - 서울특별시
228
                            강남구 역삼1동 도표예보</title>\nlink>http://www.kma.go.kr/weather/main.jsp</link>\n<description>동네예보
                            웹서비스</description>\n<language>ko</language>\n<generator>동네예보</generator>\n<pubDate>2023년 02월 13일
                            (월)요일 14:00</pubDate>\n <item>\n<author>기상청</author>\n<category>서울특별시 강남구
                            역사1동</category>\n<title>동네예보(도표): 서울특별시 강남구 역삼1동
                            [X=61,Y=125]</title><link> \underline{http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST\&amp;dongCollineseries.jsp?searchType=INTEREST&amp;dongCollineseries.jsp?searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCollineseries.jsp.searchType=INTEREST&amp;dongCo
                            de=1168064000</link>\n<guid>http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST&amp;do
                            \underline{ngCode} = 1168064000 < /guid > (n < tes > 4 < /te > (n < te > 4 < /te > (n <
                             < x > 61 < / x > n < y > 125 < / y > n < / header > n < body > n < data seq = "0" > n < hour > 18 < / hour > n < day > 0 < / day > n
                             <wfKor>흐림</wfKor>\n <wfEn>Cloudy</wfEn>\n <pop>30</pop>\n <r12>0.0</r12>\n <s12>0.0</s12>\n
```

 $<r06>0.0</r06>\n <s06>0.0</s06>\n </data>\n <data seq="1">\n <hour>21</hour>\n <day>0</day>\n <hour>21</hour>$

```
< reh > 70 < / reh > \n < r
229
230
             . . .
231
232
233
        5)xml.etree.ElementTree module 사용하기
234
           -Python에서 xml data를 다루기 위한 module
235
236
             import xml.etree.ElementTree as ET
237
             import pandas as pd
238
239
             xml_data = ET.fromstring(weather_data)
240
241
             for tag in xml_data.iter('data'):
               print(tag.find('hour').text + "/" + tag.find('temp').text)
242
243
244
             18/-2.0
245
             21/-4.0
             24/-5.0
246
247
             . . .
248
249
250
             list = []
251
             for tag in xml_data.iter('data'):
252
               dic = {'hour': tag.find('hour').text,
               'day' : tag.find('day').text,
253
254
               'temp': tag.find('temp').text,
               'tmx': tag.find('tmx').text,
255
256
               'tmn': tag.find('tmn').text,
257
               'sky' : tag.find('sky').text,
'pty' : tag.find('pty').text,
258
259
               'wfKor': tag.find('wfKor').text,
260
               'wfEn': tag.find('wfEn').text}
261
               list.append(dic)
262
263
             df = pd.DataFrame(list, columns=['hour', 'day', 'temp','tmx','tmn','sky', 'pty', 'wfKor', 'wfEn'])
264
             df
265
266
                                tmx tmn
267
               hour day temp
                                                sky pty wfKor
                                                                       wfEn
                                  -999.0 -999.0 2
-999.0 -999.0 2
268
             0 180
                       -2.0
                                                       0
                                                             구름 조금 Partly Cloudy
                                                             구름 조금 Partly Cloudy
269
             1 210
                       -4.0
                                                       0
270
             2 240
                       -5.0
                                  -999.0 -999.0 1
                                                       0
                                                            맑음
                                                                       Clear
271
             3 3
                         -6.0
                                    -1.0
                                            -8.0
                                                    1
                                                          0
                                                               맑음
                                                                          Clear
                     1
                          -7.0
272
             4 6
                     1
                                     -1.0
                                             -8.0
                                                     1
                                                          0
                                                               맑음
                                                                          Clear
273
274
275
276
     4. Lab: RSS Scraping
277
        1)전자신문 RSS
278
          import pandas as pd
279
          import xml.etree.ElementTree as ET
280
          import requests
281
282
          api_url = 'http://rss.etnews.com/Section901.xml'
283
284
          etnews_data = requests.get(api_url).text
285
          etnews_data
286
287
          288
289
          xml data = ET.fromstring(etnews_data)
290
291
          for tag in xml_data.iter('item'):
292
            print(tag.find('title').text + "," + tag.find('pubDate').text)
293
294
          1월 벤처투자 '80% 급감'...회수시장 냉각에 관망세 장기화 우려, Mon, 13 Feb 2023 14:28:00 +0900
295
          "동남아 잡아라" 코웨이, 말레이시아서 에어컨 렌털,Mon, 13 Feb 2023 14:22:00 +0900
296
           식품업계, 작년 최대 실적 경신 잇따라...제당, 30조 돌파 전망,Mon, 13 Feb 2023 14:20:00 +0900
          불경기에 침대시장 지각 변동...에이스침대 '주춤' 지누스 '껑충',Mon, 13 Feb 2023 14:17:00 +0900
297
          무보 노조, 수은법 개정 반발..."중장기 수출보험 위축, 中企 지원 축소 불가피",Mon, 13 Feb 2023 14:10:00 +0900
298
299
          한전, 지난해 전력판매로만 23조원 손실...전체 적자 30조원 육박 전망, Mon, 13 Feb 2023 13:42:00 +0900
300
301
302
303
          list = []
304
305
          for tag in xml_data.iter('item'):
306
              dic = {'Title':tag.find('title').text,
307
                   'Link':tag.find('link').text,
```

308

'Author':tag.find('author').text,

<temp>5.0</temp>\n <tmx>-999.0</tmx>\n <tmn>-999.0</tmn>\n <sky>4</sky>\n <pty>0</pty>\n <wfKor>흐림</wfKor>\n <wfEn>Cloudy</wfEn>\n <pop>30</pop>\n <r12>0.0</r12>\n <s12>0.0</s12>\n

```
309
                   'PubDate':tag.find('pubDate').text,
310
                   'Guid':tag.find('guid').text}
311
              list.append(dic)
312
313
          list
314
315
           [\{'Title': "1월 벤처투자 '80% 급감'...회수시장 냉각에 관망세 장기화 우려",
            'Link': 'https://www.etnews.com/20230213000186',
316
            'Author': '유근일',
'PubDate': 'Mon, 13 Feb 2023 14:28:00 +0900',
317
318
            'Guid': '20230213000186'},
319
           {'Title': "'동남아 잡아라" 코웨이, 말레이시아서 에어컨 렌털'
320
321
            'Link': 'https://www.etnews.com/20230213000179',
            'Author': '정다은',
322
            'PubDate': 'Mon, 13 Feb 2023 14:22:00 +0900',
323
324
              ...
325
326
          df = pd.DataFrame(list, columns=['Title', 'Link', 'Author', 'PubDate', 'Guid'])
327
328
          df.head()
329
330
331
          ...
332
333
          df.info()
334
335
           <class 'pandas.core.frame.DataFrame'>
336
          RangeIndex: 30 entries, 0 to 29
337
          Data columns (total 5 columns):
338
                  30 non-null object
          Title
                   30 non-null object
339
          Link
340
          Author
                    30 non-null object
341
          PubDate 30 non-null object
                   30 non-null object
342
          Guid
343
          dtypes: object(5)
344
          memory usage: 1.2+ KB
345
346
347
348
     5. Login이 필요한 site에서 download하기
349
        1)한빛출판네트워크 login page
350
          http://www.hanbit.co.kr/member/login.html
351
352
353
          http://www.hanbit.co.kr/myhanbit/myhanbit.html
354
355
        3)로그인 폭
356
           <form name="frm" id="frm" action="#" method="post">
357
           <input name="retun url" id="retun url" type="hidden" value="" class="i text" size="100">
358
           <div class="login_left">
359
             <fieldset>
360
                <legend>한빛출판네트워크 로그인</legend>
361
362
                <label class="i label" for="login id"><strong style="position: absolute; visibility: visible;"></strong>
                  <input name="m_id" id="m_id" type="text" value="" class="i_text" placeholder="아이디"
363
                  onkeydown="javascript:if(event.keyCode==13){login_proc(); return false;}">
364
                </label>
365
366
                <label class="i label" for="login pw"><strong style="position: absolute;"></strong>
                  <input name="m_passwd" id="m_passwd" type="password" value="" class="i_text" placeholder="비밀번호"
367
                  onkeydown="javascript:if(event.keyCode==13){login_proc(); return false;}">
368
                </label>
369
370
                <label>
                  <input type="button" name="login_btn" id="login_btn" value="로그인" class="btn_login">
371
372
                </label>
373
374
                <label class="i_label2">
375
                  <input type="checkbox" name="keepid" id="keepid" value="1" class="i_check"><strong>아이디 저장</strong>
376
                </label>
377
             </fieldset>
378
379
             ul class="login btn">
                <a href="https://www.hanbit.co.kr/member/find_id.html" class="btn_idc">아이디 찾기</a>
380
381
                <a href="https://www.hanbit.co.kr/member/find_pw.html" class="btn_pwc">비밀번호 찾기</a>
                <a href="https://www.hanbit.co.kr/member/member agree.html" class="btn_joinc">회원가입</a>
382
383
             384
           </div>
385
           </form>
386
387
388
        4)m_id, m_passwd라는 값(name 속성의 값)을 입력하고, 입력 양식을 제출하면 즉 submit하면 login되는 구조이다.
```

389 390

```
391
        5)Login 과정 분석
          -Chrome⊴ Network tab
392
393
          -상단의 filter중에서 'Doc'를 클릭한다.
394
          -그 위의 [Preserve log] check
395
             --원래 [Network] tab은 page가 이동할 때 기존 page와 관련된 내용을 지우고, 새로운 page의 내용만 띄운다.
396
             --하지만 이것을 check하면 내용을 지우지 않고 유지해준다.
397
             --login 과정을 분석하려면 web page를 어떻게 이동하는지 알아야하므로 반드시 체크한다.
398
          -로그인을 수행한다.
399
          -그러면, login.html -> login proc.php -> www.hanbit.co.kr -> m.hanbit.co.kr의 과정으로 보인다.
400
          -하나하나의 과정을 클릭하면 자세한 내용을 볼 수 있다.
401
          -login_proc.php를 클릭해보자.
402
          -Request Method가 POST임을 알 수 있다.
403
          -Payload 탭의 Form Data 섹션의 m_id와 m_passwd의 값을 확인할 수 있다.
404
          -다시 말해, login_proc.php페이지에 입력 양식 data를 POST로 전달하면 로그인한다는 것을 알 수 있다.
405
406
407
        6)Python으로 login하기
408
409
          import requests
410
          from bs4 import BeautifulSoup
          from urllib parse import urljoin
411
412
          # 아이디와 비밀번호 지정하기
USER = "*******
413
414
          PASS = "*********
415
416
417
          # 세션 시작하기
          session = requests.session()
418
419
420
          # 로그인하기
421
          login_info = {
              "m_id": USER, # 아이디 지정
422
              "m_passwd": PASS # 비밀번호 지정
423
424
425
426
          url_login = "http://www.hanbit.co.kr/member/login_proc.php"
427
          res = session.post(url_login, data=login_info)
428
          res.raise_for_status() # 오류가 발생하면 예외가 발생.
429
430
          # 마이페이지에 접근하기
          url_mypage = "http://www.hanbit.co.kr/myhanbit/myhanbit.html"
431
432
          res = session.get(url mypage)
          res.raise_for_status()
433
434
435
          # 마일리지와 이코인 가져오기
          soup = BeautifulSoup(res.text, "html.parser")
436
437
          mileage = soup.select_one(".mileage_section1 span").get_text()
          ecoin = soup.select_one(".mileage_section2 span").get_text()
438
439
          print("마일리지: " + mileage)
          print("이코인: " + ecoin)
440
441
442
          마일리지: 0
443
          이코인: 0
444
445
446
     6. Web page image 추출하기
447
448
        import requests
449
       r = requests.get("http://wikibook.co.kr/logo.png")
450
       # Binary 형식으로 데이터 저장하기 with open("test.png", "wb") as f:
451
452
453
           f.write(r.content)
454
        print("saved")
```