



1. 리스트

- **리스트(list):** 하나의 변수에 여러 값을 할당하는 자료형이다.
- 파이썬에서는 리스트처럼 여러 데이터를 하나의 변수에 할당하는 기법을 시퀀스 자료형이라고 한다. 시퀀스 자료형은 여러 자료를 순서대로 넣는다는 뜻이다.
- 리스트는 하나의 자료형으로만 저장하지 않고, 정수형이나 실수형 같은 다양한 자료형을 포함할 수 있다.

```
colors = ['red', 'blue', 'green']
```

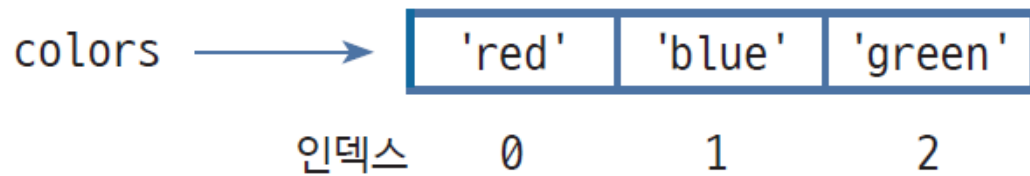
colors →

'red'	'blue'	'green'
-------	--------	---------

1. 리스트

- **인덱싱:** 리스트에 있는 값에 접근하기 위해, 이 값의 상대적인 주소를 사용하는 것이다.

```
colors = ['red', 'blue', 'green']  
print(colors[0])  
print(colors[1])  
print(colors[2])  
print(len(colors))
```



1. 리스트

- **슬라이싱**: 리스트의 인덱스를 사용하여 전체 리스트에서 일부를 잘라내어 반환한다.

```
cities = ['서울', '도쿄', '뉴욕', '북경', '런던', '파리', '시드니', '방콕']  
print(cities[0:6])  
print(cities[0:5])  
print(cities[5:])  
print(cities[-8:])
```

```
['서울', '도쿄', '뉴욕', '북경', '런던', '파리']  
['서울', '도쿄', '뉴욕', '북경', '런던']  
['파리', '시드니', '방콕']  
['서울', '도쿄', '뉴욕', '북경', '런던', '파리', '시드니', '방콕']
```

1. 리스트

- **덧셈 연산** : 덧셈 연산을 하더라도 따로 어딘가 변수에 할당해 주지 않으면 기존 변수는 변화가 없다

```
>>> color1 = ['red', 'blue', 'green']
>>> color2 = ['orange', 'black', 'white']
>>> print(color1 + color2)           # 두 리스트 합치기
['red', 'blue', 'green', 'orange', 'black', 'white']
>>> len(color1)                     # 리스트 길이
3
>>> total_color = color1 + color2
>>>
```

```
>>> total_color = color1 + color2
>>> total_color
['red', 'blue', 'green', 'orange', 'black', 'white']
```

1. 리스트

- **곱셈 연산** : 리스트의 곱셈 연산은 기준 리스트에 n을 곱했을 때, 같은 리스트를 n배만큼 늘려 준다.

```
>>> color1 * 2                                # color1 리스트 2회 반복  
['red', 'blue', 'green', 'red', 'blue', 'green']
```

- **in 연산** : 포함 여부를 확인하는 연산으로, 하나의 값이 해당 리스트에 들어 있는지 확인할 수 있다.

```
>>> 'blue' in color2                          # color2 변수에서 문자열 'blue'의 존재 여부 반환  
False
```

1. 리스트

- 리스트 추가 및 삭제

함수	기능	용례
append()	새로운 값을 기존 리스트의 맨 끝에 추가	color.append('white')
extend()	새로운 리스트를 기존 리스트에 추가(덧셈 연산과 같은 효과)	color.extend(['black','purple'])
insert()	기존 리스트의 i번째 인덱스에 새로운 값을 추가. i번째 인덱스를 기준으로 뒤쪽의 인덱스는 하나씩 밀림	color.insert(0, 'orange')
remove()	리스트 내의 특정 값을 삭제	color.remove('white')
del	특정 인덱스값을 삭제	del color[0]

1. 리스트

- 패킹(packing): 한 변수에 여러 개의 데이터를 할당하는 것.
- 언패킹(unpacking): 한 변수의 데이터를 각각의 변수로 반환하는 것.

```
t = [1, 2, 3]
a, b, c = t
print(t)
print(a, b, c)
```

```
>>> t = [1, 2, 3]
>>> a, b, c = t
>>> print(t)
[1, 2, 3]
>>> print(a, b, c)
1 2 3
□
```

Q) 리스트 값이 3인데, 4개로 언패킹 하게 되면?

2. 자료구조

- 파이썬에서의 자료구조.

자료구조명	특징
스택(stack)	나중에 들어온 값을 먼저 나갈 수 있도록 해 주는 자료구조(last in first out)
큐(queue)	먼저 들어온 값을 먼저 나갈 수 있도록 해 주는 자료구조(first in first out)
튜플(tuple)	리스트와 같지만, 데이터의 변경을 허용하지 않는 자료구조
세트(set)	데이터의 중복을 허용하지 않고, 수학의 집합 연산을 지원하는 자료구조
딕셔너리 (dictionary)	전화번호부와 같이 키(key)와 값(value) 형태의 데이터를 저장하는 자료구조, 여기서 키값은 다른 데이터와 중복을 허용하지 않음
collections 모듈	위에 열거된 여러 자료구조를 효율적으로 사용할 수 있도록 지원하는 파이썬 내장(built-in) 모듈

2. 자료구조

- **스택(Stack)**

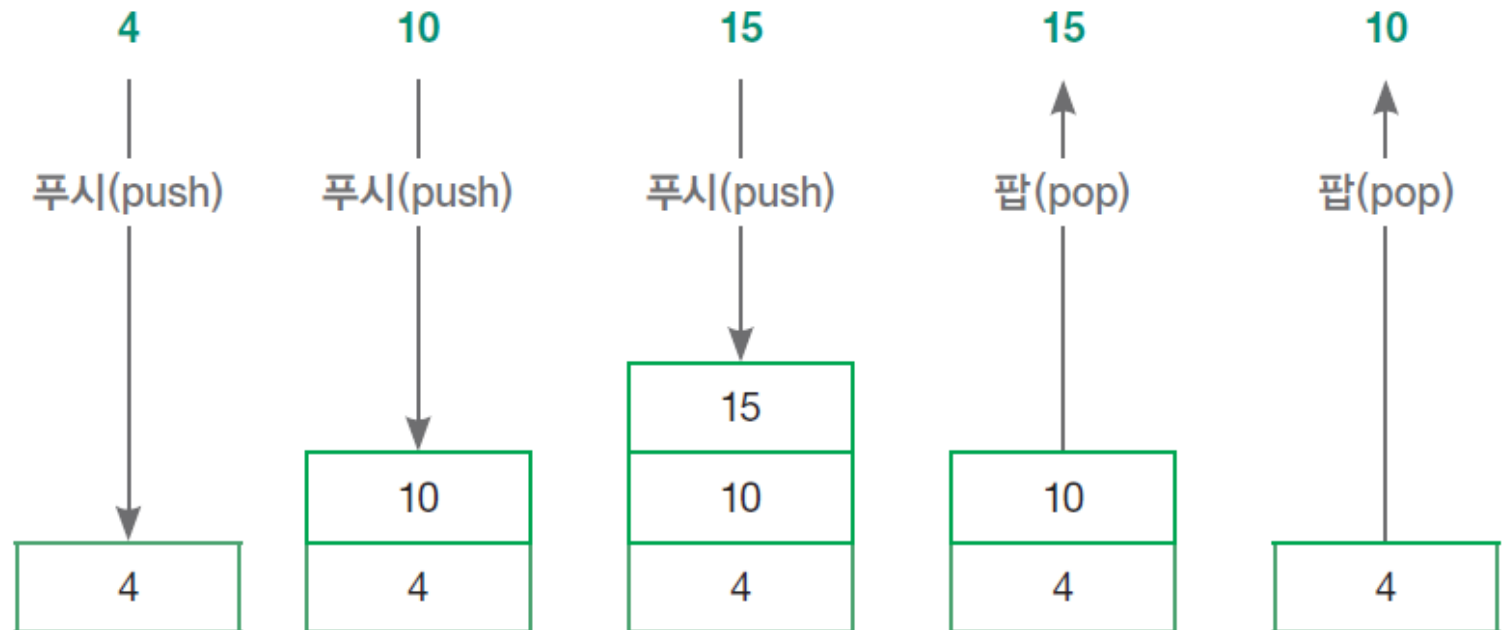
- 자료구조의 핵심 개념 중 하나로, 간단히 표현하면 'Last In First Out LIFO'으로 정의
- 마지막에 들어간 데이터가 가장 먼저 나오는 형태로, 데이터의 저장 공간을 구현하는 것이다.

- **푸시(push)**

- 스택에서 데이터를 저장

- **팝(pop)**

- 데이터를 추출



2. 자료구조

- 파이썬에서는 리스트를 사용하여 스택을 구현

- 1) 리스트라는 저장 공간 생성
- 2) `append()` 함수로 데이터를 저장(push)
- 3) 추출(pop)

```
>>> a = [1, 2, 3, 4, 5]
>>> a.append(10)
>>> print(a)
[1, 2, 3, 4, 5, 10]
>>> a.append(20)
>>> print(a)
[1, 2, 3, 4, 5, 10, 20]
>>> print(a.pop())
20
>>> print(a.pop())
10
```

Q) 입력한 텍스트를 역순으로 추출하기 (02_stack.py)

```
Input a word: Hello World
['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd']
['d', 'l', 'r', 'o', 'W', ' ', 'o', 'l', 'l', 'e', 'H']
dlroW olleH
```

2. 자료구조

- 큐(Queue)

- 스택과 다르게 먼저 들어간 데이터가 먼저 나오는 'Fist in First Out(FIFO)'의 메모리 구조

```
a = [1, 2, 3, 4, 5]
a.append(10)
a.append(20)
print(a.pop(0))
print(a.pop(0))
```

3. 튜플과 집합

- 튜플(Tuple)

- 리스트와 같은 개념이지만, 데이터를 변경할 수 없는 자료구조

ex) 학번, 이름, 주민등록 번호 등 변경할 수 없는 데이터를 위해 사용

```
t = (1, 2, 3)
print (t + t, t + 2)
len(t)
```

```
t[1] = 5
```

```
t[1] = 5
TypeError: 'tuple' object does not support item assignment
```

3. 튜플과 집합

- **집합(set)**

- 값을 순서 없이 저장하면서 중복을 허용하지 않는 자료형
- 튜플과 다르게 삭제나 변경이 가능
- 다양한 집합 연산을 제공

```
s = set([1, 2, 3, 4, 5, 1, 2, 3])  
print(s)
```

```
{1, 2, 3, 4, 5}
```

- `add()` : 원소 하나를 추가
- `remove()` 또는 `discard()` : 원소 하나를 제거
- `update()` : 새로운 리스트를 그대로 추가
- `clear()` : 모든 변수를 지우기

4. Dictionary

- 딕셔너리(dictionary)
 - 전화번호부와 같이 키(key)와 값(value) 형태로 데이터를 저장하는 자료구조

학번	이름	생년월일	주소
20150230	홍길동	1995-04-03	서울시 동대문구
20150233	김영철	1995-04-20	성남시 분당구
20150234	오영심	1996-01-03	성남시 중원구
20150236	최성철	1995-12-27	인천시 계양구

4. Dictionary

- 파이썬에서 딕셔너리의 선언
 - 중괄호 {}를 사용하여 키와 값의 쌍으로 구성

딕셔너리 변수 = {키 1:값 1, 키 2:값 2, 키 3:값 3, ...}

ex) 학번, 이름 정보 저장

학번(키)	이름(값)
20140012	Janhyeok
20140059	Jiyong
20150234	JaeHong
20140058	Wonchul

```
>>> student_info = {20140012:'Sungchul', 20140059:'Jiyong', 20140058:'JaeHong'}
```

```
>>> student_info[20140012]  
'Sungchul'
```


4. Dictionary

ex) 국가와 전화번호 연결

```
country_code = {}  
country_code = {"America": "1", "Australia": "61", "Japan": "81", "Korea": "82",  
"China": "86"}  
print(country_code)  
  
print(country_code.keys())  
  
country_code["German"] = 49  
print(country_code)  
print(country_code.values())
```

```
{'America': '1', 'Australia': '61', 'Japan': '81', 'Korea': '82', 'China': '86'}  
dict_keys(['America', 'Australia', 'Japan', 'Korea', 'China'])  
{'America': '1', 'Australia': '61', 'Japan': '81', 'Korea': '82', 'China': '86', 'German': 49}  
dict_values(['1', '61', '81', '82', '86', 49])
```

4. Dictionary

ex) 국가와 전화번호 연결

```
print(country_code.items())

for k, v in country_code.items():
    print("Key:", k)
    print("Value:", v)
```

```
dict_items([('America', '1'), ('Australia', '61'), ('Japan', '81'), ('Korea', '82'), ('China', '86'), ('German', '49')])
Key: America
Value: 1
Key: Australia
Value: 61
Key: Japan
Value: 81
Key: Korea
Value: 82
Key: China
Value: 86
Key: German
Value: 49
```

4. Dictionary

- 딕셔너리 관련 함수
 - if문을 사용하여 특정 키나 값이 해당 변수에 포함되어 있는지 확인하는 것이다.

```
>>> "Korea" in country_code.keys()           # 키에 "Korea"가 있는지 확인
True
>>> 82 in country_code.values()              # 값에 82가 있는지 확인
True
```

5. 람다 함수

- 람다(lambda) 함수 → 함수형 프로그래밍
 - 함수의 이름 없이, 함수처럼 사용할 수 있는 익명의 함수
- 일반적으로 람다 함수는 이름을 지정하지 않아도 사용 가능

```
1 def func1(x, y):  
2     return x + y  
3  
4 print(func1(1, 4))
```

```
6 f = lambda x, y: x + y  
7 print(f(1, 4))
```

ex) # block중에 record가 포함되지 않고 -11문자열을 포함하는 block만 RDD로 뽑는다
l = t.filter(lambda data: "/record/" not in data[1] and "-11" in data[1])

6. 맵리듀스

- 맵리듀스(MapReduce)

- 파이썬뿐 아니라, 빅데이터를 처리하기 위한 기본 알고리즘으로도 많이 사용
- 맵리듀스에는 *map()* 함수와 *reduce()* 함수가 있다.

- *map()* 함수

- 연속 데이터를 저장하는 시퀀스형에서 요소마다 같은 **기능을 적용**할 때 사용
- 일반적으로 리스트나 튜플처럼 요소가 있는 시퀀스 자료형에 사용

```
9  ex = [1, 2, 3, 4, 5]
10  f = lambda x : x ** 2
11  result = map(f, ex)
12  print ("Result1=", result)
```

```
14  print ("Result2=", list(result))
```

6. 맵리듀스

- 리스트 컴프리헨션 사용

[<the_expression> for <the_element> in <the_iterable>]

for <the_element> in <the_iterable>:

<the_expression>

```
17 # list comprehension
18 multiples = [n**2 for n in ex]
19 print("Result3=", multiples)
```

6. 맵리듀스

- **map() 함수: filtering**

- 리스트 컴프리헨션과 달리 else문을 반드시 작성해 해당 경우가 존재하지 않는 경우를 지정

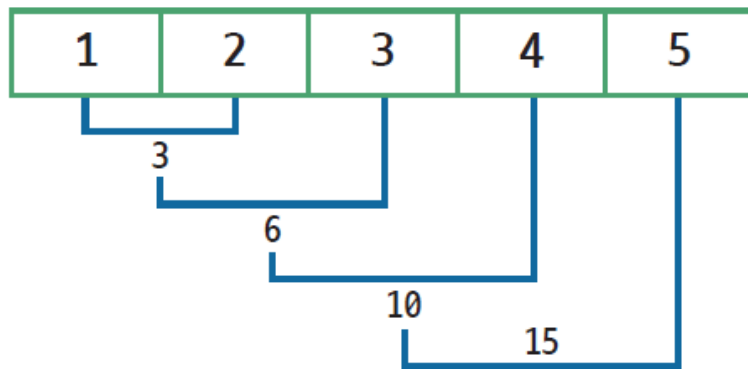
```
21 # filtering
22 print("Result4=", list(map(lambda x: x** 2 if x % 2 == 0 else x, ex)))
23 result2 = [x ** 2 if x % 2 == 0 else x for x in ex]
24 print("Result5=", result2)
```

6. 맵리듀스

- **reduce() 함수**

- 리스트와 같은 시퀀스 자료형에 차례대로 함수를 적용하여 모든 값을 **통합**하는 함수
- 리스트 변수의 모든 값을 람다 함수로 모두 적용

```
27 from functools import reduce
28 result3 = reduce(lambda x, y : x + y, ex)
29 print("Reduce result1=", result3)
```



```
31 x = 0
32 for y in ex:
33     x += y
34
35 print("Reduce result2=", x)
```


7. 웹 스크래핑

- 웹 스크래핑

- HTML로 구성되어 있는 웹 문서의 정보를 얻어오는 것
- HTML의 **규칙을 파악** 하여 필요한 HTML에서 필요한 정보 취득

```
1  import urllib.request
2
3  url = "http://storage.googleapis.com/patents/grant_full_text/2014/ipg140107.zip"
4
5  print("Start download")
6  fname, header = urllib.request.urlretrieve(url, 'ipg140107.zip')
7  print("End download")
```

→ 특정 그림이나 자료를 자동화하여 다운로드 가능 ex) 구글 검색을 통한 특정 인물 사진 다운로드

7. 웹 스크래핑

- 정규 표현식

- 정규 표현식(Regular Expression)은 문자 표현식
- 문자열의 규칙을 파악하여 데이터 집합을 추출

일반 문자	정규 표현식
010-0000-0000	<code>^\d{3}-\d{4}-\d{4}\$</code>
203.252.101.40	<code>^\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}\$</code>

. ^ \$ * + ? { } [] \ | ()

← 메타문자

7. 웹 스크래핑

- 정규 표현식

. ^ \$ + ? { } [] \ | ()

메타 문자	설명
[]	문자 클래스
.	\w를 제외한 모든 문자와 매치 (점 하나는 글자 하나를 의미)
*	0회 이상 반복 (업어도 상관 없음)
+	1회 이상 반복 (무조건 한번 이상 등장해야 함)
{m, n}	m회 이상 n회 이하
	or 조건식을 의미
^	문자열의 시작 의미
\$	문자열의 끝을 의미
?	0회 이상 1회 이하
\w	이스케이프, 또는 메타 문자를 일반 문자로 인식하게 한다
()	그룹핑, 추출할 패턴을 지정한다.

7. 웹 스크래핑

- 정규 표현식 - 기본 메타문자 []

- [와] 사이의 문자와 매칭 ex) [abc] → **a** or **b** or **c**
- Q) 비틀스의 노래 가사 중 'yesterday'와 'Yesterday'를 검색하려면? → **[Yy]esterday**
- Q) [abc] 로 검색하면, "a, before, deep, dud, sunset" 중 검색되는 것은? → **a, before**
- '-' 를 사용하여 구간 지정 가능
 - [A-Za-z] or [가-힣] or [0-9]

- 정규 표현식 - 반복 메타문자 -, +, *, ?, { }

```
[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9]
```

```
[0-9]+-[0-9]+-[0-9]+
```

7. 웹 스크래핑

- 정규 표현식 - 반복 메타문자 -, +, *, ?, { }
- {} → 횟수 조절
- + → 1번 이상 반복
- * → 0번 이상 반복

```
[0-9]{3}-[0-9]{3,4}-[0-9]{4}
```

- ex) *tomot*ow*

```
tomorrow  
tomoow  
tomorrrow
```

7. 웹 스크래핑

- 정규 표현식 - 그외 메타문자 (), ., |, ^, \$, \w
- () → 묶음 표시
- . → 문자 하나
- | → OR
- ^ → NOT, 시작 메타문자
- \$ → 마지막 메타문자

```
import re
p = re.compile('a|b') # a 또는 b를 찾는다.
m = p.findall("abcdefg")
print(m)
```

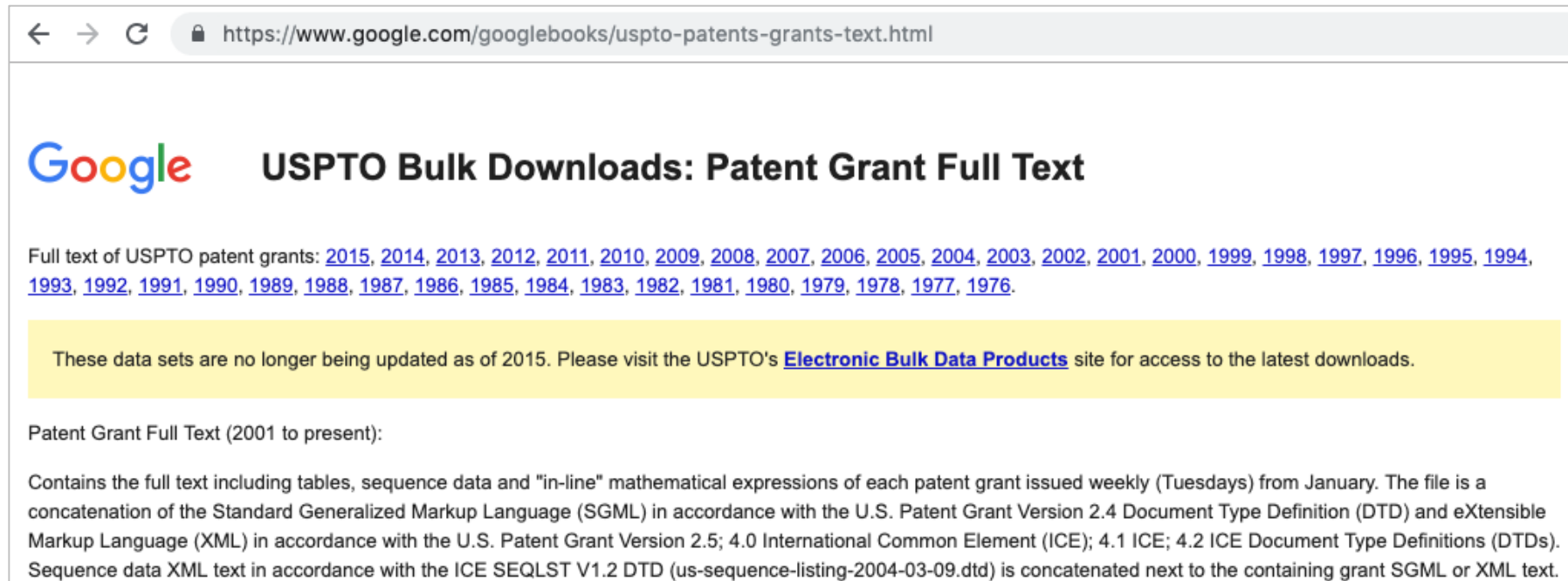
```
import re
m1 = re.findall("^Life", "Life is too short")
m2 = re.findall("^is", "Life is too short")
print("m1 결과 : ", m1)
print("m2 결과 : ", m2)
```

```
import re
m1 = re.findall("short$", "Life is too short")
m2 = re.findall("short$", "Life is too short. So what?")

print("m1 결과 : ", m1)
print("m2 결과 : ", m2)
```

7. 웹 스크래핑

- 정규 표현식 - 예제
 - 미국 특허 정보 데이터 세트 홈페이지
 - <https://www.google.com/googlebooks/uspto-patents-grants-text.html>



7. 웹 스크래핑

- 정규 표현식 - 예제

```
67 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150106.zip">
68 ipg150106.zip</a>&nbsp;
69 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150113.zip">
70 ipg150113.zip</a>&nbsp;
71 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150120.zip">
72 ipg150120.zip</a>&nbsp;
73 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150127.zip">
74 ipg150127.zip</a>&nbsp;
75 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150203.zip">
76 ipg150203.zip</a>&nbsp;
77 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150210.zip">
78 ipg150210.zip</a>&nbsp;
79 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150217.zip">
80 ipg150217.zip</a>&nbsp;
81 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150224.zip">
82 ipg150224.zip</a>&nbsp;
83 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150303.zip">
84 ipg150303.zip</a>&nbsp;
85 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150310.zip">
86 ipg150310.zip</a>&nbsp;
87 <a href="http://storage.googleapis.com/patents/grant_full_text/2015/ipg150317.zip">
88 ipg150317.zip</a>&nbsp;
89
90
91 <h3 id="2014">2014</h3>
```

- 정규 표현식 연습 사이트 (<http://www.regexr.com>)

7. 웹 스크래핑

- HTML 파싱

- 웹 페이지의 HTML을 분석하여 필요한 URL을 추출하는 작업
- <https://finance.naver.com/item/main.nhn?code=017670>

<https://finance.naver.com/item/main.nhn?code=017670>

페이지
endpoint

변수
querystring
(name=value;)

7. 웹 스크래핑

- HTML 파싱

- <https://finance.naver.com/item/main.nhn?code=000880>

한화 000880 코스피 2019.05.23 11:24 기준 (장중) 실시간 기업개요			
25,950 전일대비 ▼650 -2.44%	전일 26,600	고가 26,650 (상한가 34,550)	거래량 121,193
	시가 26,500	저가 25,750 (하한가 18,650)	거래대금 3,155 백만
선차트 1일 1주일 3개월 1년 3년 5년 10년		봉차트 일봉 주봉 월봉	

- 웹 페이지의 '페이지 소스 보기' 기능으로 HTML 코드 확인 → 추출 프로그램 작성

7. 웹 스크래핑

- HTML 파싱

25,950 전일대비 ▼650 -2.44%	전일 26,600	고가 26,650 (상한가 34,550)	거래량 121,901
	시가 26,500	저가 25,750 (하한가 18,650)	거래대금 3,173 백만
선차트 1일 1주일 3개월 1년 3년 5년 10년			봉차트 일봉 주봉 월봉

```
<dl class="blind">
  <dt>종목 시세 정보</dt>
  <dd>2019년 05월 23일 11시 22분 기준 장중</dd>
  <dd>종목명 한화</dd>
  <dd>종목코드 000880 코스피</dd>
  <dd>현재가 25,850 전일대비 하락 750 마이너스 2.82 퍼센트</dd>
  <dd>전일가 26,600</dd>
  <dd>시가 26,500</dd>
  <dd>고가 26,650</dd>
  <dd>상한가 34,550</dd>
  <dd>저가 25,750</dd>
  <dd>하한가 18,650</dd>
  <dd>거래량 118,143</dd>
  <dd>거래대금 3,076백만</dd>
</dl>
```

8. 웹 스크래핑 - 실습

- HTML 파싱
 - 주식 데이터에서 해당 부분을 파싱하는 코드를 작성

한화 000880 코스피 2019.05.23 11:24 기준(장중) 실시간 기업개요

25,950 전일대비 ▼650 -2.44%	전일 26,600 고가 26,650 (상한가 34,550) 거래량 121,193
시가 26,500	

선차트 1일 | 1주일 | 3개월 | 1년 | 3년 | 5년 | 10년

```
<dl class="blind">
  <dt>종목 시세 정보</dt>
  <dd>2019년 05월 23일 11시 22분 기준 장중</dd>
  <dd>종목명 한화</dd>
  <dd>종목코드 000880 코스피</dd>
  <dd>현재가 25,850 전일대비 하락 750 마이너스 2.82 퍼센트</dd>
  <dd>전일가 26,600</dd>
  <dd>시가 26,500</dd>
  <dd>고가 26,650</dd>
  <dd>상한가 34,550</dd>
  <dd>저가 25,750</dd>
  <dd>하한가 18,650</dd>
  <dd>거래량 118,143</dd>
  <dd>거래대금 3,076백만</dd>
</dl>
```

8. 웹 스크래핑 - 실습

- HTML 파싱
 - 가져올 정보를 파악 → 패턴 작성 → 정보 추출

```
(\)([\s\S]+?)(\</dl>)
```

```
(\)([\s\S]+?)(\</dd>)
```

```
<dd>하한가 18,650</dd>  
<dd>거래량 118,143</dd>  
<dd>거래대금 3,076백만</dd>  
</dl>
```

8. 웹 스크래핑 - 실습

```
1  import urllib.request
2  import re
3
4  url = "http://finance.naver.com/item/main.nhn?code=000880"
5  html = urllib.request.urlopen(url)
6  html_contents = str(html.read().decode("ms949"))
7
8  # 첫 번째 HTML 패턴
9  stock_results = re.findall("(\\<dl class=\\\"blind\\\"\\>)([\\s\\S]+?)(\\<\\/dl\\>)", html_contents)
10 hanwha_stock = stock_results[0]      # 두 개의 튜플 값 중 첫 번째 패턴
11 hanwha_index = hanwha_stock[1]      # 세 개의 튜플 값 중 두 번째 패턴
12
13 # 주식 정보만 추출함
14 index_list= re.findall("(\\<dd\\>)([\\s\\S]+?)(\\<\\/dd\\>)", hanwha_index)
15
16 for index in index_list:
17     print(index[1])                  # 세 개의 튜플 값 중 두 번째 값
```

9. Beautiful Soup

- HTML과 XML 파일로부터 데이터를 추출하기 위한 라이브러리
- \$ pip install beautifulsoup4
- \$ pip install requests

```
from bs4 import BeautifulSoup
html_doc = """
<html lang="ko" class="svgless">
<head>
<title>NAVER</title>
</head>
<body>
<p class='main'>Python Crawling Study</p>
<h3 class="ah_ltit">실시간 급상승</h3>
<a href="http://datalab.naver.com/keyword/realtimeList.naver?where=main"
  class="ah_ha" data-clk="lve.rankhistory">
  <span class="ah_ico_datalab">DataLab.</span>급상승 트래킹<span class="ah_ico_hlink"></span>
</a>
<div class="ah_tab">
<a href="#" role="tab" class="ah_tab_btn ah_tab_on" data-tab="1to10" data-clk="lve.tab1">1~10위</a>
<a href="#" role="tab" class="ah_tab_btn" data-tab="11to20" data-clk="lve.tab2">11~20위</a>
</div>
Loun ipsum, Loun ipsum, Loun ipsum, Loun ipsum, Loun ipsum
</body>
</html>
"""

soup = BeautifulSoup(html_doc, 'html.parser')

print(soup.prettify())
```

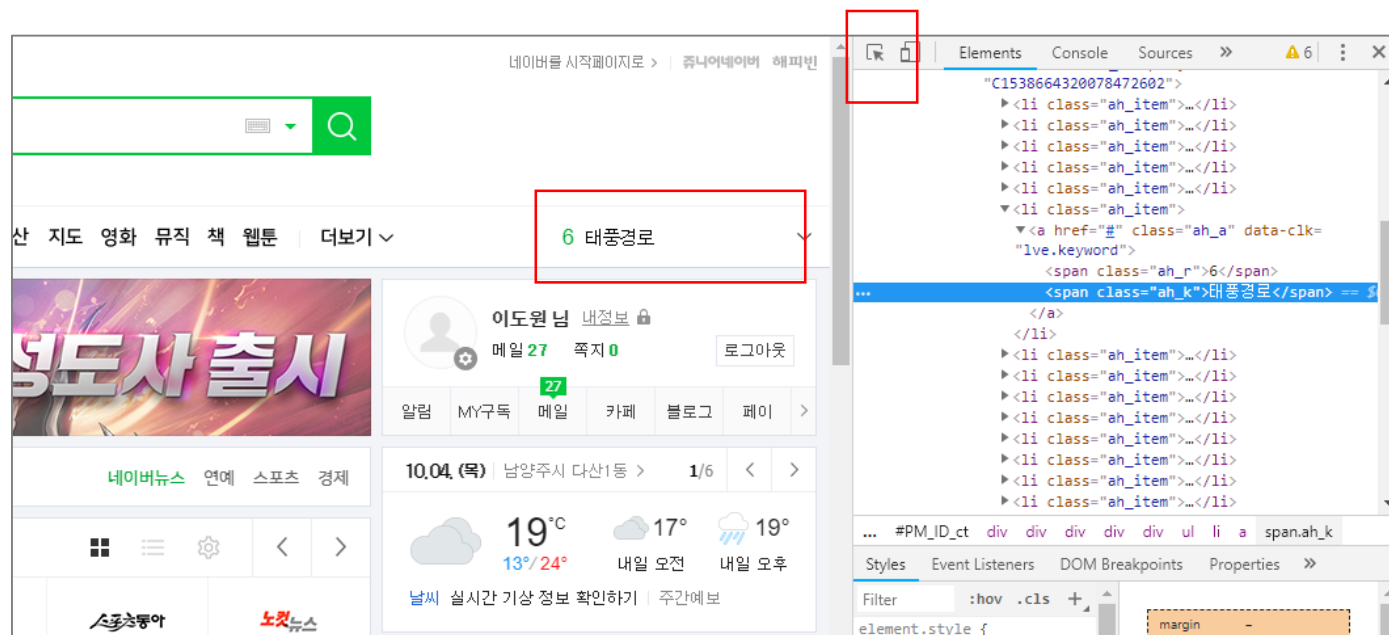
10. 네이버 실시간 검색어 분석

- Crawling (or Scraping)

- 웹 페이지를 그대로 가져온 다음 데이터를 추출하는 것 → Crawler
- 웹 상의 정보를 자동으로 검색하고 색인 할 때 사용되는 것 → Spider, Bot, Intelligence Agent

실시간 검색어 가져 올 수 없음

- 크롬 웹브라우저 > 개발자모드 > Select 도구 → 크롤링 하길 원하는 부분을 선택



10. 네이버 실시간 검색어 분석

```
▼<a href="#" class="ah_a" data-clk="lve.keyword">  
  <span class="ah_r">2</span>  
  <span class="ah_k">단 하나의 사랑</span> == $0  
</a>  
</li>  
▶<li class="ah_item">...</li>  
▶<li class="ah_item">...</li>
```

실시간 검색어 가져 올 수 없음

- 마우스 우클릭 > Copy > Copy Selector ➔ 복사

```
#PM_ID_ct > div.header > div.section_navbar > div.area_hotkeyword.PM_CL_realtimeKeyword_base >  
div.ah_roll.PM_CL_realtimeKeyword_rolling_base > div > ul > li:nth-child(12) > a > span.ah_k
```

```
#PM_ID_ct > div.header > div.section_navbar > div.area_hotkeyword.PM_CL_realtimeKeyword_base >  
div.ah_roll.PM_CL_realtimeKeyword_rolling_base > div > ul > li:nth-child(2) > a > span.ah_k
```

10. 네이버 실시간 검색어 분석

```
1 from bs4 import BeautifulSoup
2 import requests
3
4 url = "https://www.naver.com"
5 response = requests.get(url)
6 source = response.text
7
8 soup = BeautifulSoup(source, 'html.parser')
9
10 topList = soup.select("#PM_ID_ct > div.header > div.section_navbar > div.area_hotkeyword.PM_CL_realtimeKeyword_base >
    div.ah_roll.PM_CL_realtimeKeyword_rolling_base > div > ul > li > a > span.ah_k")
11
12 for top in topList:
13     print(top.text)
```

실시간 검색어 가져 올 수 없음

펭귄 문제
단 하나의 사랑
명지대
봄밤
드라마 봄밤
스냅챗 아기
애기얼굴 어플
펭귄 프사
강효상
펭귄
부대찌개 3인분을 먹으면
신혜선
정해인

11. 네이버 뉴스

https://finance.naver.com/news/market_notice.nhn?&page=1

NAVER 금융

종목명·펀드명·환율명·원자재명 입력



통합검색

금융 홈 국내증시 해외증시 시장지표 펀드 투자전략 **뉴스** MY 추천종목

· (중요) MY 매매내역 및 MY펀드서비스가 종료되었습니다. · 해외 증시휴장일 정보 제공 종료 · [한국거래소] 공매도 종합 포털 · [IRGO] 상장기업 IR

뉴스

실시간 속보

주요뉴스

뉴스포커스

시황·전망
기업·종목분석
해외증시
채권·선물

금융 홈 > 뉴스 > 투자정보 > 공시정보

속보 인크로스 57래일 연속 상승... 전.



공시정보

한류타임즈(주) (정정)타법인주식및출자증권취득결정	19.05.23 14:20	KOSCOM
주식회사 에스제이캠 기업설명회(IR) 개최 결과	19.05.23 14:20	KOSCOM
(주)테크트렌스 기업설명회(IR) 개최 결과	19.05.23 14:15	KOSCOM
이비테크(주) 기업설명회(IR) 개최 결과	19.05.23 14:12	KOSCOM
(주)장원테크 전환가액의조정(제1회차, 제2회차, 제3회차)	19.05.23 14:11	KOSCOM

뉴스로 보는 증시일정

오늘의 증시일정

미래에셋제5호기업인수목적(주) 최대주주변경	19.05.23 13:43	KOSCOM
미래에셋제5호기업인수목적(주) 본점소재지변경	19.05.23 13:42	KOSCOM
(주)씨앤에스링크 기업설명회(IR) 개최 결과	19.05.23 13:40	KOSCOM
NH-Amundi HANARO 200증권상장지수투자신탁[주식] ..	19.05.23 13:26	KOSCOM
한화 ARIRANG 200증권상장지수투자신탁(주식) 변경상장	19.05.23 13:26	KOSCOM

1 | 2 | 3 | 4 | 5 | 6 | 7 | 다음 > 맨뒤 >>

종목명 검색

tr 257 x 17

1 | 2 | 3 | 4 | 5 | 6 | 7 | 다음 > 맨뒤 >>

<!-- paging -->

<table summary="페이지 네비게이션 리스트" class="Nnavi" align="center">

<caption>페이지 네비게이션</caption>

<tbody>

<tr>

<td>

<table cellspacing="0" cellpadding="0" border="0">

<tbody>

<tr align="center"> == \$0

<td class="on">

1

</td>

<td>...</td>

11. 네이버 뉴스

```
1  from bs4 import BeautifulSoup
2  import requests
3
4  maximum = 0
5  page = 1
6
7  url = "https://finance.naver.com/news/market_notice.nhn"
8  response = requests.get(url)
9  source = response.text
10 soup = BeautifulSoup(source, "html.parser")
11
12 last_page_a = soup.find("td", {"class": "pgRR"})
13 value = last_page_a.find("a").get("href")
14 maximum = value.split("=")[1]
15
16 print("총 " + str(maximum) + " 개의 페이지가 확인 되었습니다.")
```

11. 네이버 뉴스

```
18 whole_source = ""
19 for page_number in range(1, 2):
20     url = "https://finance.naver.com/news/market_notice.nhn?&page=" + str(page_number)
21     response = requests.get(url)
22     whole_source = whole_source + response.text
23
24 soup = BeautifulSoup(whole_source, "html.parser")
25
26 find_title = soup.select("#contentarea_left > div.boardList2 > table > tr > td.publicSubject > a")
27 for title in find_title:
28     print(title.text)
```

총 4234 개의 페이지가 확인 되었습니다 .
한류타임즈 (주) (정정)타법인주식및출자증권취득결정
주식회사 에스제이캠 기업설명회 (IR) 개최 결과
(주)테크트랜스 기업설명회 (IR) 개최 결과
이비테크 (주) 기업설명회 (IR) 개최 결과

실습) 네이버 영화 평점 가져오기

- Beautifulsoup 사용

주의 요함 | movie.naver.com/movie/point/af/list.nhn

NAVER 영화

영화홈

상영작 · 예정작

영화랭킹

예매

평점 · 리뷰

네티즌 평점

네티즌 리뷰

다운로드

인디극장 up

로그인

영화검색

검색

네티즌·관람객 평점

현재 상영작 평정보기

현재 상영작

전체 리스트

총 13,169,227개의 평점이 있습니다.

번호	감상평	글쓴이·날짜
17111804	만추 ★★★★★ 9 신고	alcl**** 20.08.31
17111803	테넷 ★★☆☆☆ 5 소문난 잔치에 먹을거 없다. 신고	kimk**** 20.08.31
17111802	나의 첫 번째 슈퍼스타 ★★★★★ 1 주인공이 도대체 누군지 모르겠고 매니저역은 너무 개성이 없고 연기력이 부족 해요 신고	sury**** 20.08.31
17111801	테넷 ★★★★★ 9 오늘의 친구와 대화나:야 테넷 레전드다 첨에 나온게 다 이어져 소름~친구:오 그래서 먼 내용이야?나:어,,,? ㄹㅇ이거 무한루트임 신고	jaso**** 20.08.31

영화 인기검색어

더보기

1 테넷 - 0

2 다만 악에서 구하소서 - 0

3 반도 ↑ 1

4 오케이 마담 ↓ 1

5 강철비2: 정상회담 - 0

2020.08.30


네티즌 최고 평점

더보기

현재 상영영화

모든 영화

1 가버나움 9.59



2 동주 9.37

3 소년시절의너 9.36

4 소원 9.32

5 패왕별희 디오리지널 9.31

12. Selenium

- 웹 애플리케이션 *테스트 프레임워크*
- 웹 사이트에서 버튼 클릭과 같이 이벤트 처리 가능
- JavaScript 실행 가능
- 웹 브라우저 실행을 대신하기 위한 Web Driver 설치 → Selenium이 사용하기 위한 웹 브라우저
 - <http://chromedriver.chromium.org/downloads>
- \$ pip install selenium

```
from selenium import webdriver

path = "C:\\Users\\down\\Desktop\\work\\python_수업\\chromedriver_win32\\chromedriver.exe"
driver = webdriver.Chrome(path)
```

```
driver.get("http://google.com/")
```

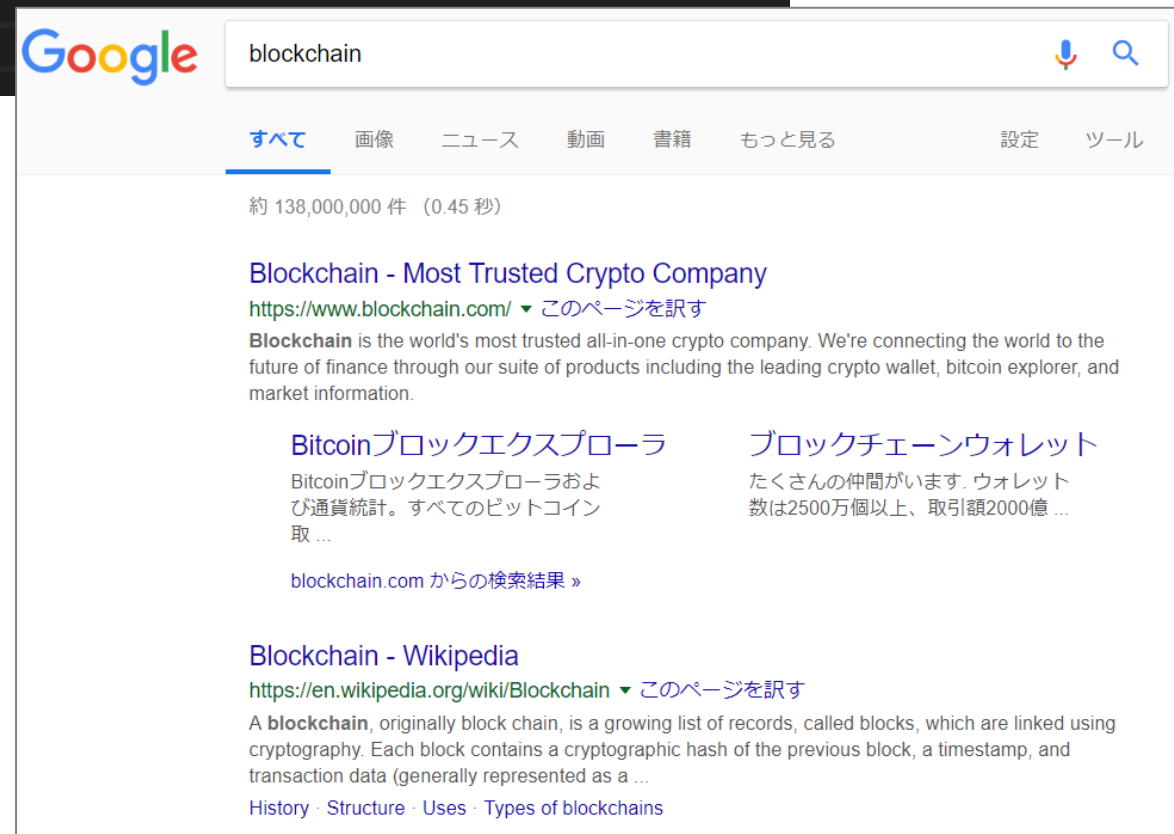
12. Selenium – 구글 검색

```
from selenium import webdriver

path = "C:\\\\Users\\down\\Desktop\\work\\python_수업\\chromedriver_win32\\chromedriver.exe"
driver = webdriver.Chrome(path)

driver.get("http://google.com/")
search_box = driver.find_element_by_name("q")
search_box.send_keys("blockchain")
search_box.submit()
```

실습) 개발자 도구에서 검색어 부분 보기



12. Selenium – 페이스북 로그인

- input 태그에 name이나 id같은 선택자가 있으면 Selenium에서 테스트 가능



The image shows the Facebook login and sign-up interface. At the top, the Facebook logo is on the left. To its right are two input fields: '이메일 또는 휴대폰' (Email or Phone) and '비밀번호' (Password), both highlighted with red boxes. A '로그인' (Login) button is to the right of the password field. Below the login fields is a link that says '계정을 잊으셨나요?' (Forgot your account?).

Below the login section, there is a promotional message: 'Facebook에서 전세계에 있는 친구, 가족, 지인들과 함께 이야기를 나눠보세요.' (Share stories with friends, family, and acquaintances all over the world with Facebook). This is accompanied by a world map with orange person icons and dashed lines connecting them.

To the right of the map is the '가입하기' (Sign Up) section. It starts with the text '항상 지금처럼 무료로 즐기실 수 있습니다.' (You can always enjoy it for free like now). Below this are several input fields: '성(姓)' (Last Name) with a red box and an exclamation mark icon, '이름(성은 제외)' (First Name), '휴대폰 번호 또는 이메일' (Phone number or email), and '새 비밀번호' (New password). At the bottom is a '생일' (Birthday) section with dropdown menus for year (1993), month (10월), and day (4), followed by a link '왜 생년월일을 입력해야 하나요?' (Why do I need to enter my birth date?).

실습) 개발자 도구에서 이메일/비밀번호 찾아보기

12. Selenium – 페이스북 로그인

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup

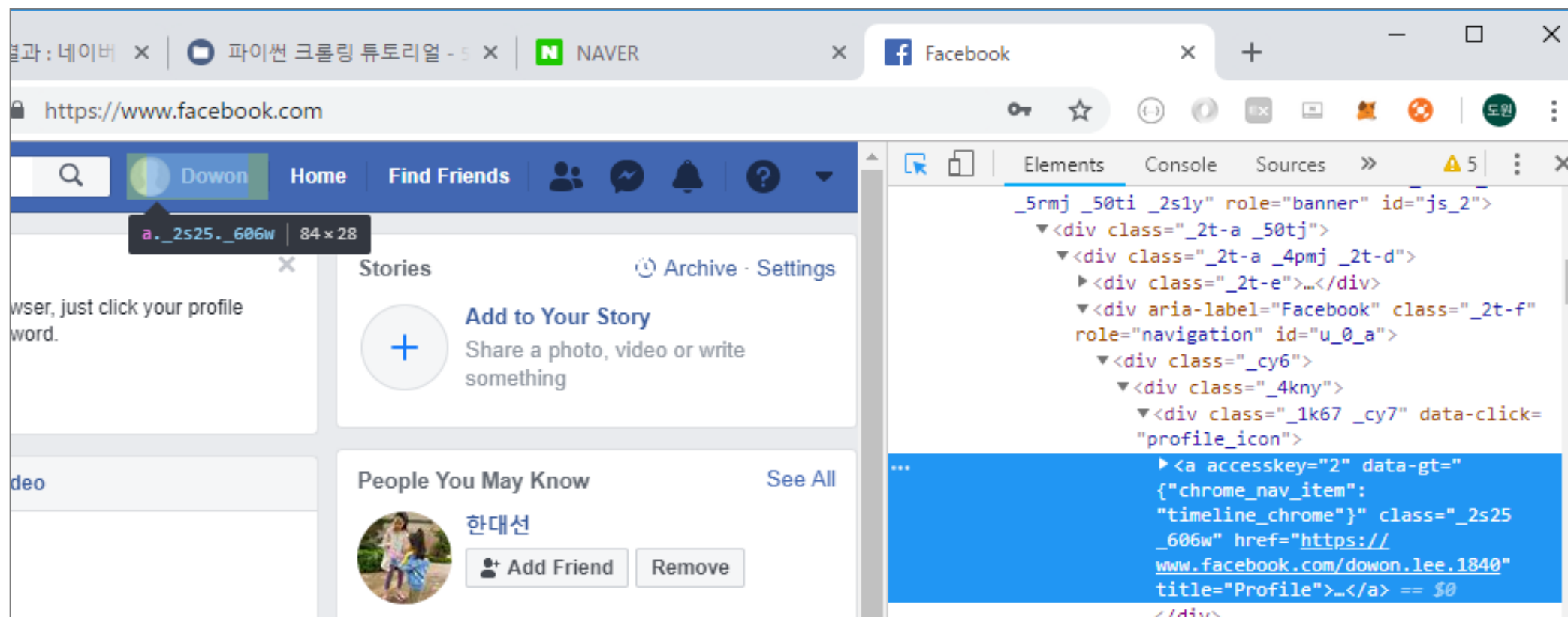
usr = "페이스북 아이디"
pwd = "페이스북 비밀번호"

path = "C:\\Users\\down\\Desktop\\work\\python_수업\\chromedriver_win32\\chromedriver.exe"

driver = webdriver.Chrome(path)
driver.get("https://www.facebook.com/")
assert "Facebook" in driver.title
elem = driver.find_element_by_id("email")
elem.send_keys(usr)
elem = driver.find_element_by_id("pass")
elem.send_keys(pwd)
elem.send_keys(Keys.RETURN)
```

실습) 로그아웃 처리

12. Selenium – 페이스북 프로파일



- 프로필 링크 > Copy > Copy XPath

12. Selenium – 페이지스북 프로파일

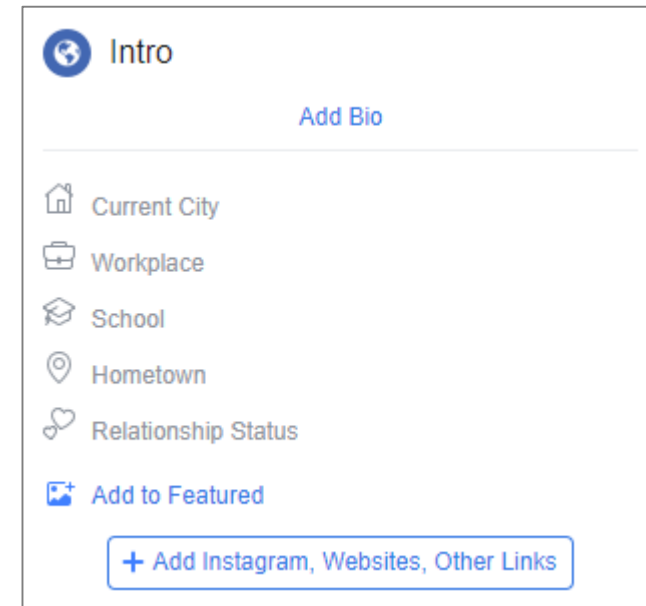
```
a = driver.find_elements_by_xpath('//*[@id="u_0_a"]/div[1]/div[1]/div/a')
driver.get(a[0].get_attribute('href'))
```

```
a = driver.find_element_by_xpath("//*[@id='u_0_a']/div[1]/div[1]/div/a")
driver.get(a.get_attribute('href'))
```

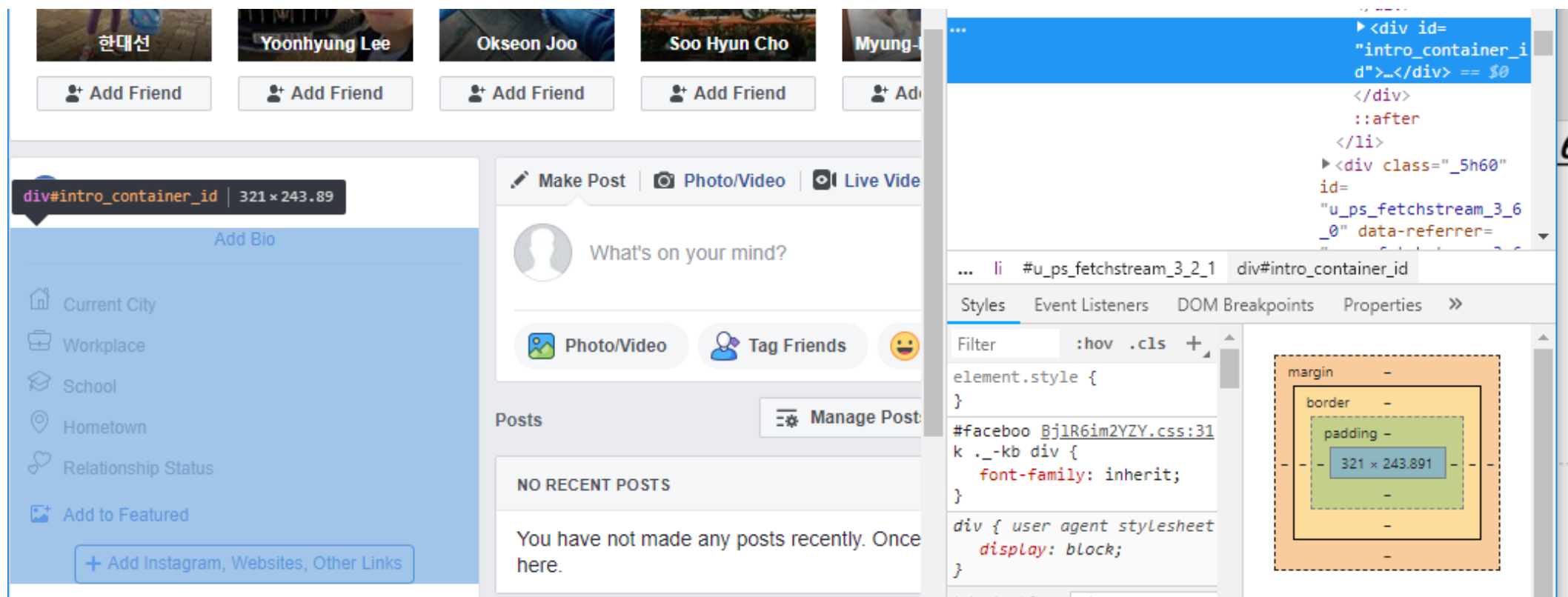
- ~~• find_elements_by_xpath의 결과는 list → index로 추출하려는 속성에 접근~~

```
req = driver.page_source
```

```
soup = BeautifulSoup(req, 'html.parser')
```




12. Selenium – 페이스북 프로파일





실습) profile의 div id 가져오기

```
information_list = soup.select("#intro_container_id")
for information in information_list:
    print(information.text)
```


실습) Github 이슈 가져오기


 Search or jump to...


Pull requestsIssuesMarketplaceExplore

+

done409 / itman

 Watch 0

 Star 0

 Fork 0

<> Code

🔔 Issues 1

🔗 Pull requests 0

📁 Projects 0

📖 Wiki


📊 Insights

⚙️ Settings

First issue #1

EditNew issue


🔔 Open done409 opened this issue 3 hours ago · 1 comment



done409 commented 3 hours ago

Owner+😊...

첫번째 이슈입니다.



done409 commented 3 hours ago

Owner+😊...

두번째 이슈고요...

WritePreview

AA B i “ <> 🔗 ⋮ ⋮ ✓ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

📖 Styling with Markdown is supported

Close issueComment

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone


No milestone

Notifications

🔔 Unsubscribe

You're receiving notifications because you authored the thread.

1 participant



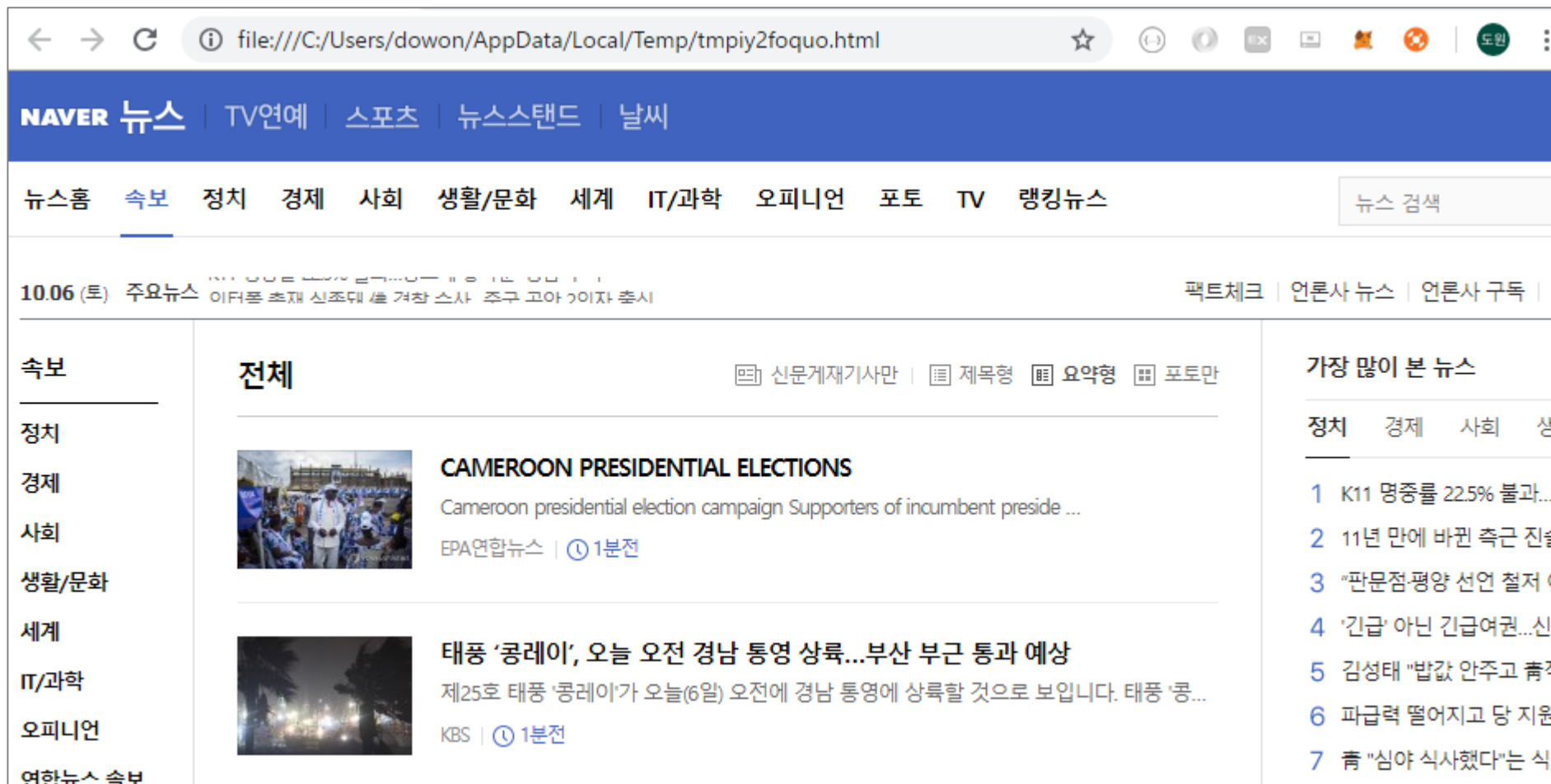
13. Scrapy

- 수많은 웹 페이지로부터 정보를 수집 → 빅데이터로 활용
- Scraping을 위한 라이브러리
 - \$ pip install scrapy
- Scrapy Shell
 - \$ scrapy shell

```
In [1]: fetch('https://news.naver.com/main/list.nhn?mode=LSD&mid=sec&sid1=001')
2018-10-06 00:15:03 [scrapy.core.engine] INFO: Spider opened
2018-10-06 00:15:03 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://news.naver.com/main/list.nhn?mode=LSD&mid=sec&sid1=001> (referer: None)

In [2]: view(response) █
```

13. Scrapy



- \$ print(response.text)

13. Scrapy

- 크롤링 타켓
 - 제목
 - 올린 뉴스 사이트
 - 미리보기 내용

보안/해킹



인섹시큐리티, '2018 오프스왓 세미나' 18일 개최

[디지털데일리 홍하나기자] 디지털포렌식 및 네트워크 보안 전문업체인 인섹시큐리티(인섹)가 18일 오프스왓 세미나를 개최한다. 이번 세미나는 디지털포렌식 및 네트워크 보안 전문가들이 모여 최신 보안 기술과 사례를 공유하는 자리로, 오프스왓 세미나는 인섹의 대표 행사로, 매년 1회 개최된다. 이번 세미나는 오프스왓 세미나는 인섹의 대표 행사로, 매년 1회 개최된다. 이번 세미나는 오프스왓 세미나는 인섹의 대표 행사로, 매년 1회 개최된다.

디지털데일리 | ① 7시간전



사이버공격방어대회·제주사이버보안컨퍼런스 동시 개최

2018년도 사이버공격방어대회와 제주사이버보안컨퍼런스 개최가 한 달 앞으로 다가왔...

전자신문 | ① 8시간전



17일 제1회 블록체인 국제 표준 워크숍 열려

분산원장기술표준포럼(의장 류재철)은 17일 섬유센터에서 제1회 블록체인 국제 표준 워...

전자신문 | ① 10시간전

//*[@id="main_content"]/div[2]/ul[1]/li[1]/dl/dt[2]/a

//*[@id="main_content"]/div[2]/ul[1]/li[2]/dl/dt[2]/a

//*[@id="main_content"]/div[2]/ul[1]/li[3]/dl/dt[2]/a

13. Scrapy

- CSS를 가지고 크롤링 → 올린 뉴스 사이트 출력

```
response.css('.writing::text').extract()
```

```
·헤럴드POP',  
·파이낸셜뉴스',  
·한국경제',  
·EPA연합뉴스',  
·AP연합뉴스',  
·EPA연합뉴스',  
·EPA연합뉴스',  
·EPA연합뉴스',  
·아시아경제',  
·파이낸셜뉴스',  
·뉴스1',  
·EPA연합뉴스']
```

13. Scrapy

- XPath를 가지고 크롤링 → 미리보기 출력

```
response.css('.lede::text').extract()
```

```
"Egyptian-Saudi Super Cup - press conference Egypt's Zamalek head coach ...",  
'Sonia Guajajara In this Sept. 15, 2018 photo, Sonia Guajajara, an indig ...',  
"Egyptian-Saudi Super Cup - press conference Egypt's Zamalek head coach ...",  
"Egyptian-Saudi Super Cup - press conference Egypt's Zamalek player Haze ...",  
"Egyptian-Saudi Super Cup - press conference Egypt's Zamalek head coach ...",  
"'나 혼자 산다'에 출연하는 배우 이시언이 관악산 별장을 찾는다. 5일 방송된 MBC '나 혼자 산다'에서는 관악산으  
로 물놀이를 ...",  
'[인사] 군포시 <4급 승진 >복지문화국장 현승식 <건설교통국장 홍재섭 <군포1동장 안선수 <4급 전보 <기획  
재정국장 배재철 <...',  
'(평양·서울=뉴스1) 공동취재단,김다혜 기자 = 10·4선언 11주년 기념 민족통일대회 참석을 위해 평양을 방문한 우  
리 방북단이 ...',  
"Egyptian-Saudi Super Cup - press conference Egypt's Zamalek head coach ..."]
```

14. *Scrapy - Sample*

- Scrapy 프로젝트 구조

```
tutorial/  
  scrapy.cfg          # deploy configuration file  
  
tutorial/  
  __init__.py         # project's Python module, you'll import your code from here  
  
  items.py            # project items definition file  
  
  middlewares.py      # project middlewares file  
  
  pipelines.py        # project pipelines file  
  
  settings.py         # project settings file  
  
  spiders/  
    __init__.py       # a directory where you'll later put your spiders
```

14. Scrapy – Project 작성

- \$ *scrapy startproject myscrapper*

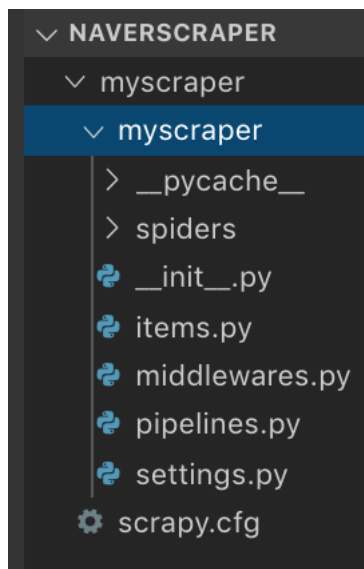
```
dowon@DOWON-MacBook ~/Desktop/강 의 /00.강 의 자 료 _초 안 /Python/data_analysis/naverscraper$ scrapy startproject myscrapper
New Scrapy project 'myscraper', using template directory '/usr/local/lib/python3.7/site-packages/scrapy/templates/project', created in:
/Users/dowon/Desktop/강 의 /00.강 의 자 료 _초 안 /Python/data_analysis/naverscraper/myscraper
```

You can start your first spider with:

```
cd myscrapper
```

```
scrapy genspider example example.com
```

```
dowon@DOWON-MacBook ~/Desktop/강 의 /00.강 의 자 료 _초 안 /Python/data_analysis/naverscraper$
```



14. Scrapy – Spider 작성

- \$ *scrapy genspider mybots*

"news.naver.com/main/list.nhn?mode=LS2D&mid=shm&sid1=105&sid2=732"

```
PS C:\Users\down\Desktop\work\python_수업\ch17\myscaper\myscaper> scrapy genspider mybots "news.naver.com/main/list.nhn?mode=LS2D&mid=shm&sid1=105&sid2=732"
Created spider 'mybots' using template 'basic' in module:
myscaper.spiders.mybots
PS C:\Users\down\Desktop\work\python_수업\ch17\myscaper\myscaper> 
```

14. Scrapy – Spider 작성

- ./spiders/mybots.py 수정

```
# -*- coding: utf-8 -*-
import scrapy

class MybotsSpider(scrapy.Spider):
    name = 'mybots'
    allowed_domains = ['news.naver.com/main/list.nhn?mode=LS2D&mid=shm&sid1=105&sid2=732']
    start_urls = ['http://news.naver.com/main/list.nhn?mode=LS2D&mid=shm&sid1=105&sid2=732/']

    def parse(self, response):
        pass
```


14. Scrapy – Spider 작성

- *items.py* 수정

```
import scrapy

class MyscaperItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    title = scrapy.Field()
    author = scrapy.Field()
    preview = scrapy.Field()
```

14. Scrapy – Spider 작성

- ./spiders/mybots.py 수정

```
import scrapy
from myscaper.items import MyscaperItem

class MybotsSpider(scrapy.Spider):
    name = 'mybots'
    allowed_domains = ['naver.com']
    start_urls = ['https://news.naver.com/main/list.nhn?mode=LS2D&mid=shm&sid1=105&sid2=732']

    def parse(self, response):
        titles = response.xpath('//*[@id="main_content"]/div[2]/ul/li/dl/dt[2]/a/text()').extract()
        authors = response.css('.writing::text').extract()
        previews = response.css('.lede::text').extract()

        # for item in zip(titles, authors, previews):
        #     scraped_info = {
        #         'title' : item[0].strip(),
        #         'author' : item[1].strip(),
        #         'preview' : item[2].strip(),
        #     }
        #     yield scraped_info
        items = []
        for idx in range(len(titles)):
            item = MyscaperItem()
            item['title'] = titles[idx]
            item['author'] = authors[idx]
            item['preview'] = previews[idx]
            items.append(item)
        return items
```

14. Scrapy – Spider 작성

- *settings.py* 수정

```
BOT_NAME = 'myscaper'

SPIDER_MODULES = ['myscaper.spiders']
NEWSPIDER_MODULE = 'myscaper.spiders'

# Crawl responsibly by identifying yourself (and your website) on the user-agent
#USER_AGENT = 'myscaper (+http://www.yourdomain.com)'

# Obey robots.txt rules
ROBOTSTXT_OBEY = False
```

```
FEED_FORMAT = "csv"
FEED_URI = "my_news.csv"
```

- *\$ scrapy crawl mybots*

```
2018-10-06 00:51:05 [scrapy.core.scraper] DEBUG: Scraped from <200 https://news.naver.com/main/list.nhn?mode=
{'author': '한국경제',
'preview': '호주 스타트업 트래블바이비트에 250만 달러 투자 [ 오세성 기자 ] 글로벌 가상화폐(암호화폐) 거래소
'세계 주요 공 ...',
'title': '\r\n'
'\t\t\t\t\t\t\t\t\t\t SK인포섹, 사물인터넷(IoT) 보안 가이드북 발간\r\n'
'\t\t\t\t\t\t\t\t\t\t'}
2018-10-06 00:51:05 [scrapy.core.engine] INFO: Closing spider (finished)
2018-10-06 00:51:05 [scrapy.extensions.feedexport] INFO: Stored csv feed (19 items) in: naver_news.csv
2018-10-06 00:51:05 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/request_bytes': 262,
'downloader/request_count': 1,
'downloader/request_method_count/GET': 1,
'downloader/response_bytes': 19835,
'downloader/response_count': 1,
'downloader/response_status_count/200': 1,
'finish_reason': 'finished',
'finish_time': datetime.datetime(2018, 10, 5, 15, 51, 5, 626339),
'item_scraped_count': 19,
'log_count/DEBUG': 21,
'log_count/INFO': 8,
```

14. Scrapy – Spider 실행

- \$ CSV 파일 확인

```
1 author,preview,title
2
3 디지털데일리, [디지털데일리 홍하나기자] 디지털포렌식 및 네트워크 보안 전문업체인 인섹시큐리티 (대표 김종광)는 악성코드 탐지 전문 업체 읍스왓 ..., "
4
5 인섹시큐리티, '2018 읍스왓 세미나' 18일 개최
6
7 "
8
9 전자신문, 2018년도 사이버공격방어대회와 제주사이버보안컨퍼런스 개최가 한 달 앞으로 다가왔다. 사이버공격방어대회 본선은 29일부터 30일 ..., "
10
11 사이버공격방어대회·제주사이버보안컨퍼런스 동시 개최
12
13 "
14
15 전자신문, "분산원장기술표준포럼 (의장 류재철)은 17일 섬유센터에서 제1회 블록체인 국제 표준 워크숍을 개최한다. 이번 행사는 'ISO, I ...',"
16
17 17일 제1회 블록체인 국제 표준 워크숍 열려
18
19 "
20
21 아이뉴스24, <아이뉴스24> [아이뉴스24 김국배 기자] SK텔레콤이 계열사인 국내 1위 정보보안 업체 SK인포섹 인수 추진을 공식화하고 있 ..., "
22
23 SK텔레콤, SK인포섹 인수 검토..."융합보안 확장"
24
25 "
26
27 한국경제, 10월 14일 전후 테스트넷에 비잔티움 하드포크 적용 전망 메인넷 적용도 연내 이뤄질 듯 [ 오세성 기자 ] 이더리움이 테스트넷 ..., "
28
29 코앞으로 다가온 이더리움 하드포크, PoS 전환 본격화
30
31 "
```

실습) 네이버 영화 평점 가져오기

- Scrapy 사용

주의 요함 | movie.naver.com/movie/point/af/list.nhn

NAVER 영화

영화홈

상영작 · 예정작

영화랭킹

예매

평점 · 리뷰

네티즌 평점

네티즌 리뷰

다운로드

인디극장 up

로그인

영화검색

검색

네티즌·관람객 평점

현재 상영작 평정보기

현재 상영작

전체 리스트

총 13,169,227개의 평점이 있습니다.

번호	감상평	글쓴이·날짜
17111804	만추 ★★★★★ 9 신고	alcl**** 20.08.31
17111803	테넷 ★★☆☆☆ 5 소문난 잔치에 먹을거 없다. 신고	kimk**** 20.08.31
17111802	나의 첫 번째 슈퍼스타 ★★★★★ 1 주인공이 도대체 누군지 모르겠고 매니저역은 너무 개성이 없고 연기력이 부족해요 신고	sury**** 20.08.31
17111801	테넷 ★★★★★ 9 오늘의 친구와 대화나:야 테넷 레전드다 첨에 나온게 다 이어져 소름~친구:오 그래서 먼 내용이야?나:어,,,? ㄹㅇ이거 무한루트임 신고	jaso**** 20.08.31

영화 인기검색어

더보기

1 테넷 - 0

2 다만 악에서 구하소서 - 0

3 반도 ↑ 1

4 오케이 마담 ↓ 1

5 강철비2: 정상회담 - 0

2020.08.30


네티즌 최고 평점

더보기

현재 상영영화

모든 영화

1 가버나움 9.59



2 동주 9.37

3 소년시절의너 9.36

4 소원 9.32

5 패왕별희 디오리지널 9.31