

	urllib을 이용한 웹 크롤링
--	-------------------

```
import urllib.request
d = urllib.request.urlopen("https://wikidocs.net")

data = d.read()
data = data.decode("utf-8")
print( data ) # print( d.read().decode("utf-8" ) )

<!DOCTYPE HTML>
<html lang="ko">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="google-site-verification" content="mzkAy71X1qQFWihQN535LoiToXg34MUg9nuor70g98" />
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <div class="card">
          <div class="card-body">
            <h1>
              <h2>
                <h3>
                  <h4>
                    <h5>
                    <h6>
                    <h7>
                    <h8>
                    <h9>
                    <h10>
                    <h11>
                    <h12>
                    <h13>
                    <h14>
                    <h15>
                    <h16>
                    <h17>
                    <h18>
                    <h19>
                    <h20>
                    <h21>
                    <h22>
                    <h23>
                    <h24>
                    <h25>
                    <h26>
                    <h27>
                    <h28>
                    <h29>
                    <h30>
                    <h31>
                    <h32>
                    <h33>
                    <h34>
                    <h35>
                    <h36>
                    <h37>
                    <h38>
                    <h39>
                    <h40>
                    <h41>
                    <h42>
                    <h43>
                    <h44>
                    <h45>
                    <h46>
                    <h47>
                    <h48>
                    <h49>
                    <h50>
                    <h51>
                    <h52>
                    <h53>
                    <h54>
                    <h55>
                    <h56>
                    <h57>
                    <h58>
                    <h59>
                    <h60>
                    <h61>
                    <h62>
                    <h63>
                    <h64>
                    <h65>
                    <h66>
                    <h67>
                    <h68>
                    <h69>
                    <h70>
                    <h71>
                    <h72>
                    <h73>
                    <h74>
                    <h75>
                    <h76>
                    <h77>
                    <h78>
                    <h79>
                    <h80>
                    <h81>
                    <h82>
                    <h83>
                    <h84>
                    <h85>
                    <h86>
                    <h87>
                    <h88>
                    <h89>
                    <h90>
                    <h91>
                    <h92>
                    <h93>
                    <h94>
                    <h95>
                    <h96>
                    <h97>
                    <h98>
                    <h99>
                    <h100>
                    <h101>
                    <h102>
                    <h103>
                    <h104>
                    <h105>
                    <h106>
                    <h107>
                    <h108>
                    <h109>
                    <h110>
                    <h111>
                    <h112>
                    <h113>
                    <h114>
                    <h115>
                    <h116>
                    <h117>
                    <h118>
                    <h119>
                    <h120>
                    <h121>
                    <h122>
                    <h123>
                    <h124>
                    <h125>
                    <h126>
                    <h127>
                    <h128>
                    <h129>
                    <h130>
                    <h131>
                    <h132>
                    <h133>
                    <h134>
                    <h135>
                    <h136>
                    <h137>
                    <h138>
                    <h139>
                    <h140>
                    <h141>
                    <h142>
                    <h143>
                    <h144>
                    <h145>
                    <h146>
                    <h147>
                    <h148>
                    <h149>
                    <h150>
                    <h151>
                    <h152>
                    <h153>
                    <h154>
                    <h155>
                    <h156>
                    <h157>
                    <h158>
                    <h159>
                    <h160>
                    <h161>
                    <h162>
                    <h163>
                    <h164>
                    <h165>
                    <h166>
                    <h167>
                    <h168>
                    <h169>
                    <h170>
                    <h171>
                    <h172>
                    <h173>
                    <h174>
                    <h175>
                    <h176>
                    <h177>
                    <h178>
                    <h179>
                    <h180>
                    <h181>
                    <h182>
                    <h183>
                    <h184>
                    <h185>
                    <h186>
                    <h187>
                    <h188>
                    <h189>
                    <h190>
                    <h191>
                    <h192>
                    <h193>
                    <h194>
                    <h195>
                    <h196>
                    <h197>
                    <h198>
                    <h199>
                    <h200>
                    <h201>
                    <h202>
                    <h203>
                    <h204>
                    <h205>
                    <h206>
                    <h207>
                    <h208>
                    <h209>
                    <h210>
                    <h211>
                    <h212>
                    <h213>
                    <h214>
                    <h215>
                    <h216>
                    <h217>
                    <h218>
                    <h219>
                    <h220>
                    <h221>
                    <h222>
                    <h223>
                    <h224>
                    <h225>
                    <h226>
                    <h227>
                    <h228>
                    <h229>
                    <h230>
                    <h231>
                    <h232>
                    <h233>
                    <h234>
                    <h235>
                    <h236>
                    <h237>
                    <h238>
                    <h239>
                    <h240>
                    <h241>
                    <h242>
                    <h243>
                    <h244>
                    <h245>
                    <h246>
                    <h247>
                    <h248>
                    <h249>
                    <h250>
                    <h251>
                    <h252>
                    <h253>
                    <h254>
                    <h255>
                    <h256>
                    <h257>
                    <h258>
                    <h259>
                    <h260>
                    <h261>
                    <h262>
                    <h263>
                    <h264>
                    <h265>
                    <h266>
                    <h267>
                    <h268>
                    <h269>
                    <h270>
                    <h271>
                    <h272>
                    <h273>
                    <h274>
                    <h275>
                    <h276>
                    <h277>
                    <h278>
                    <h279>
                    <h280>
                    <h281>
                    <h282>
                    <h283>
                    <h284>
                    <h285>
                    <h286>
                    <h287>
                    <h288>
                    <h289>
                    <h290>
                    <h291>
                    <h292>
                    <h293>
                    <h294>
                    <h295>
                    <h296>
                    <h297>
                    <h298>
                    <h299>
                    <h300>
                    <h301>
                    <h302>
                    <h303>
                    <h304>
                    <h305>
                    <h306>
                    <h307>
                    <h308>
                    <h309>
                    <h310>
                    <h311>
                    <h312>
                    <h313>
                    <h314>
                    <h315>
                    <h316>
                    <h317>
                    <h318>
                    <h319>
                    <h320>
                    <h321>
                    <h322>
                    <h323>
                    <h324>
                    <h325>
                    <h326>
                    <h327>
                    <h328>
                    <h329>
                    <h330>
                    <h331>
                    <h332>
                    <h333>
                    <h334>
                    <h335>
                    <h336>
                    <h337>
                    <h338>
                    <h339>
                    <h340>
                    <h341>
                    <h342>
                    <h343>
                    <h344>
                    <h345>
                    <h346>
                    <h347>
                    <h348>
                    <h349>
                    <h350>
                    <h351>
                    <h352>
                    <h353>
                    <h354>
                    <h355>
                    <h356>
                    <h357>
                    <h358>
                    <h359>
                    &lt
```

```
import re
print(re.search("[가-힣]",data) )

<re.Match object; span=(419, 420), match='온'>
```

```
status = d.getheaders()
for s in status :
    print(s)

('Server', 'nginx/1.12.2')
('Date', 'Sun, 16 Jun 2019 13:23:16 GMT')
('Content-Type', 'text/html; charset=utf-8')
('Transfer-Encoding', 'chunked')
('Connection', 'close')
('Vary', 'Cookie')
('Set-Cookie', 'sessionid=6pseprjzpv960tew0pkw86o225iyu7bo; expires=Sun, 30-Jun-2019 13:23:16 GMT; HttpOnly; Max-Age=1209600; Path=/')
```

```
d.status
```

200

```
import urllib.request

d = urllib.request.urlopen("https://www.naver.com")
print ( d.status )

if d.status == 200 :
    print( d.read().decode("utf-8 "))
```

```
200
<!doctype html>
```

.. 중략

```
import urllib.request

d = urllib.request.urlopen("https://www.naver.com")
print ( d.status )

if d.status == 200 :
    print( d.read().decode("utf-8 "))
```

...

```
import urllib.parse
a = urllib.parse.quote("전문 대학교")
print(a)
```

```
%EC%84%A0%EB%AC%B8%20%EB%8C%80%ED%95%99%EA%B5%90
```

```
a = urllib.parse.quote_plus("전문 대학교")
print(a)
```

```
%EC%84%A0%EB%AC%B8+%EB%8C%80%ED%95%99%EA%B5%90
```

BeautifulSoup 을 이용한 HTML 파싱

```

Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> html="""
<html>
<head>
<title> test web </title>
</head>
<body>
<p align="center"> text contents </p>

</body>
</html> """
>>> from bs4 import BeautifulSoup
>>> bs=BeautifulSoup(html)
>>> print(bs.prettify())
<html>
<head>
<title>
test web
</title>
</head>
<body>
<p align="center">
text contents
</p>

</img>
</body>
</html>

>>> bs.find("title")
<title> test web </title>
>>>
  
```

BeautifulSoup 객체

BeautifulSoup import

들여쓰기로 깨끗이 정렬하여 리턴

find("태그명")

```

Python 3.4.3 Shell
File Edit Shell Debug Options Window
>>>
>>> bs.find('p')
<p align="center"> text contents </p>
>>>
Ln: 86 Col: 4
  
```

find("태그명")

```

Python 3.4.3 Shell
File Edit Shell Debug Options Window
>>>
>>> bs.find('a')
>>>
>>>
Ln: 89 Col: 4
  
```

못찾은 경우

```
File Edit Shell Debug Options Window Help
>>>
>>> html = """
<html>
  <head>
    <title> test web </title>
  </head>
  <body>
    <p align="center"> text contents 1 </p>
    <p align="right"> text contents 2 </p>
    <p align="left"> text contents 3 </p>
    
  </body>
</html>"""
>>>
>>> bs = BeautifulSoup(html)
>>>
```

Ln: 103 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> bs.find('p',align="center")
<p align="center"> text contents 1 </p>
>>>
>>> bs.find('p',align="right")
<p align="right"> text contents 2 </p>
>>>
>>> bs.find('p',align="left")
<p align="left"> text contents 3 </p>
>>>
```

Ln: 121 Col: 4

find("태그명", 속성명="값 ")

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> bs.find_all('p')
[<p align="center"> text contents 1 </p>, <p align="center"> text contents 2 </p>, <p align="center"> text contents 3 </p>]
>>>
```

Ln: 158 Col: 4

find_all("태그명")
모두 찾아 리스트로 반환

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>> head_tag = bs.find('head')
>>> head_tag
<head>
<title> test web </title>
</head>
>>> head_tag.find('title')
<title> test web </title>
>>>
>>> head_tag.find('p') # 자신의 태그안에 없는 태그입니다
>>>
```

찾은 태그 안에서 다른 태그(자식요소) 찾을 수 있음.

Ln: 168 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> body_tag = bs.find('body')
>>> list1 = body_tag.find_all(['p', 'img'])
>>>
>>> for tag in list1:
>>>     print(tag)

<p align="center"> text contents 1 </p>
<p align="center"> text contents 2 </p>
<p align="center"> text contents 3 </p>

</img>
>>>
```

p와 img 태그 모두 찾아 반환

Ln: 182 Col: 4

p 태그 모두 찾아 반환

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>> bs.find_all('p')
[<p align="center"> text contents 1 </p>, <p align="center"> text contents 2 </p>, <p align="center"> text contents 3 </p>]
>>>
```

p 태그 모두 찾아 반환

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>> import re
>>> tags = bs.find_all(re.compile("^p"))
>>> tags
[<p align="center"> text contents 1 </p>, <p align="center"> text contents 2 </p>, <p align="center"> text contents 3 </p>]
>>>
```

^p
- p로 시작하는 모든 문자 정규식

속성이 align="center"를 가지고 있는 모든 태그찾음

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> bs.find_all(align="center")
[<p align="center"> text contents 1 </p>, <p align="center"> text contents 2 </p>, <p align="center"> text contents 3 </p>]
>>>
```

속성이 width="500"를 가지고 있는 모든 태그찾음

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> bs.find_all(width="500")
[
</img>]
>>>
```

문자 데이터 중
"text contents 1" 모두 찾음

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> bs.find_all(text="text contents 1 ")
['text contents 1 ']
>>>
```

Ln: 286 Col: 4

정규식 활용

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> import re
>>> bs.find_all(text=re.compile("text +"))
['text contents 1 ', 'text contents 2 ', 'text contents 3 ']
>>>
```

Ln: 294 Col: 4

p태그를 찾되 단, 최대 2개까지만 찾음

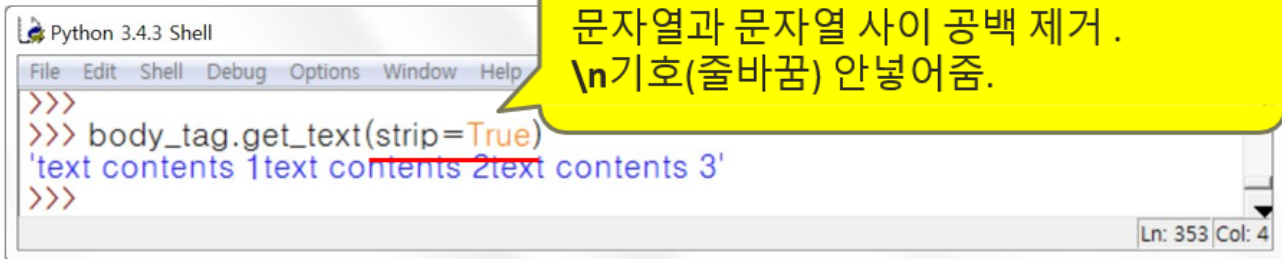
```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> bs.find_all("p", limit=2)
[<p align="center"> text contents 1 </p>, <p align="center"> text contents 2 </p>]
>>>
```

Ln: 305 Col: 4

찾은 태그 요소에서 문자데이터만 추출
(한번에 하나만 가지고 올 수 있음)

찾은 태그 요소에서 모든 문자데이터 추출

찾은 태그 요소에서 모든 문자데이터 추출.
단 문자열을 하나의 문자열로 이어붙여줌.
문자열과 문자열 사이 \n기호(줄바꿈) 삽입

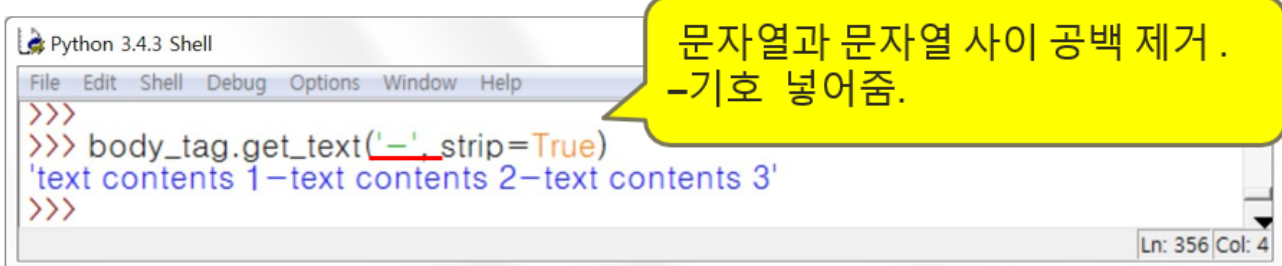


Python 3.4.3 Shell

```
>>>
>>> body_tag.get_text(strip=True)
'text contents 1text contents 2text contents 3'
>>>
```

문자열과 문자열 사이 공백 제거.
\\n기호(줄바꿈) 안넣어줌.

Ln: 353 Col: 4

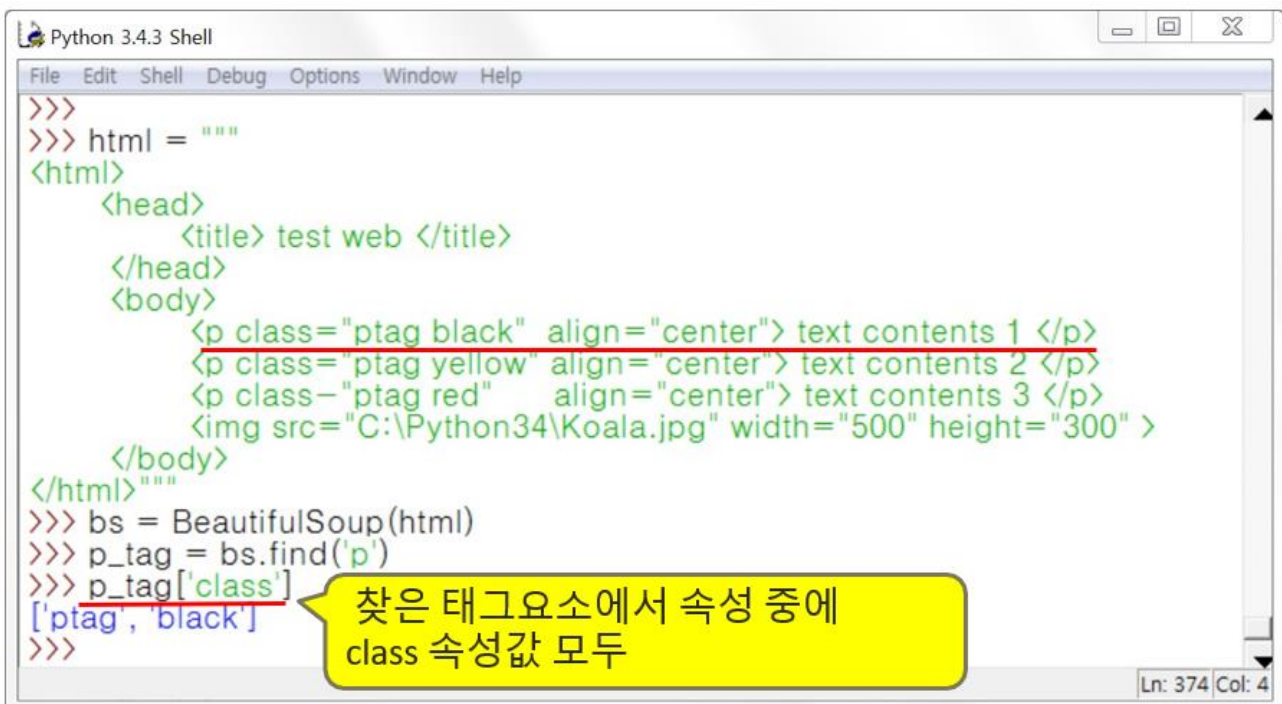


Python 3.4.3 Shell

```
>>>
>>> body_tag.get_text('-', strip=True)
'text contents 1-text contents 2-text contents 3'
>>>
```

문자열과 문자열 사이 공백 제거.
-기호 넣어줌.

Ln: 356 Col: 4



Python 3.4.3 Shell

```
>>>
>>> html = """
<html>
  <head>
    <title> test web </title>
  </head>
  <body>
    <p class="ptag black" align="center"> text contents 1 </p>
    <p class="ptag yellow" align="center"> text contents 2 </p>
    <p class="ptag red" align="center"> text contents 3 </p>
    
  </body>
</html>"""
>>> bs = BeautifulSoup(html)
>>> p_tag = bs.find('p')
>>> p_tag['class']
['ptag', 'black']
>>>
```

찾은 태그요소에서 속성 중에
class 속성값 모두

Ln: 374 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> p_tag['class'][1] = 'white'
>>> p_tag['class']
['ptag', 'white']
>>>
```

찾은 태그요소에서 속성 중에
class 속성값 중 1인덱스의 값
white로 변경

찾은 태그요소에서 속성 중에
class 속성값 모두

Ln: 379 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> p_tag['id'] = 'P-TAG'
>>> p_tag['id']
'P-TAG'
>>>
```

찾은 태그요소에서 속성 중에
id 속성값 변경

Ln: 383 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>> p_tag['align']
'center'
>>> del p_tag['align']
>>> p_tag['align']
Traceback (most recent call last):
  File "<pyshell#185>", line 1, in <module>
    p_tag['align']
  File "C:\Python34\lib\site-packages\beautifulsoup4-4.4.0-py3.4.egg\bs4\element.py", line 954, in __getitem__
    return self.attrs[key]
KeyError: 'align'
>>>
```

찾은 태그요소에서 속성 중에
align 속성 삭제

Ln: 396 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>> p_tag.attrs
{'id': 'P-TAG', 'class': ['ptag', 'white']}
>>>
```

찾은 태그요소에서
모든 속성과 속성 값

Ln: 400 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> html = """
<html>
  <head>
    <title> test web </title>
  </head>
  <body>
    <p class="ptag black" align="center"> text contents 1 </p>
    <p class="ptag yellow" align="center"> text contents 2 </p>
    <p class="ptag red" align="center"> text contents 3 </p>
    
  </body>
</html>"""
>>> bs = BeautifulSoup(html)
>>> body_tag = bs.find('body')
>>> body_tag
<body>
<p align="center" class="ptag black"> text contents 1 </p>
<p align="center" class="ptag yellow"> text contents 2 </p>
<p align="center" class="ptag=" red=" "> text contents 3 </p>

</img></body>
>>>
```

Ln: 453 Col: 4

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
>>> for child in body_tag.children:
    print(child)
```

찾은 태그요소에서
모든 자식 태그 요소들

```

<p align="center" class="ptag black"> text contents 1 </p>

<p align="center" class="ptag yellow"> text contents 2 </p>

<p align="center" class="ptag=" red=" "> text contents 3 </p>


</img>
>>>
```

Ln: 497 Col: 4

Selenium 활용한 웹 크롤링

[크롤링(Crawling)]

- 웹 페이지의 내용을 가지고 오는 것을 크롤링 또는 스크래핑(Scraping) 이라고 함
- 구글이나 네이버, 다음과 같은 검색 엔진 사이트들은 검색이 속도를 높이기 위해 로봇(robot)이라는 프로그램을 만들어서 자동으로 웹 페이지를 크롤링
- 무분별한 크롤링을 막고 제어하기 위해 1994년 6월 로봇 배제 규약 - robot.txt(로봇 접근 관련 내용) : 크롤링 허가/불허가 여부를 이 파일에 적어 놓자고 약속한 규약
- 크롤링하는 로봇프로그램은 크롤링하고 자하는 사이트의 <http://www.aaa.com/robot.txt> 파일을 찾아 분석 후 수집해도 되는 콘텐츠만 수집해야 함. 단, 강제는 아닌 권고.

[Robot.txt]

예) 홈페이지 전체가 모든 검색엔진에 허용되지 않음

User-agent: *

Disallow: /

예) 홈페이지 전체가 모든 검색엔진에 허용되지 원함

User-agent: *

Disallow:

예) 홈페이지 디렉토리중 일부만 검색엔진에 허용되지 싶음

User-agent: *

Disallow: /my_photo/

Disallow: /my_diary/

예) 홈페이지 전체를 허용하지만, 특정 검색엔진 (EvilRobot)만 거부

User-agent: EvilRobot

Disallow: /

예) <https://www.google.com/robots.txt>

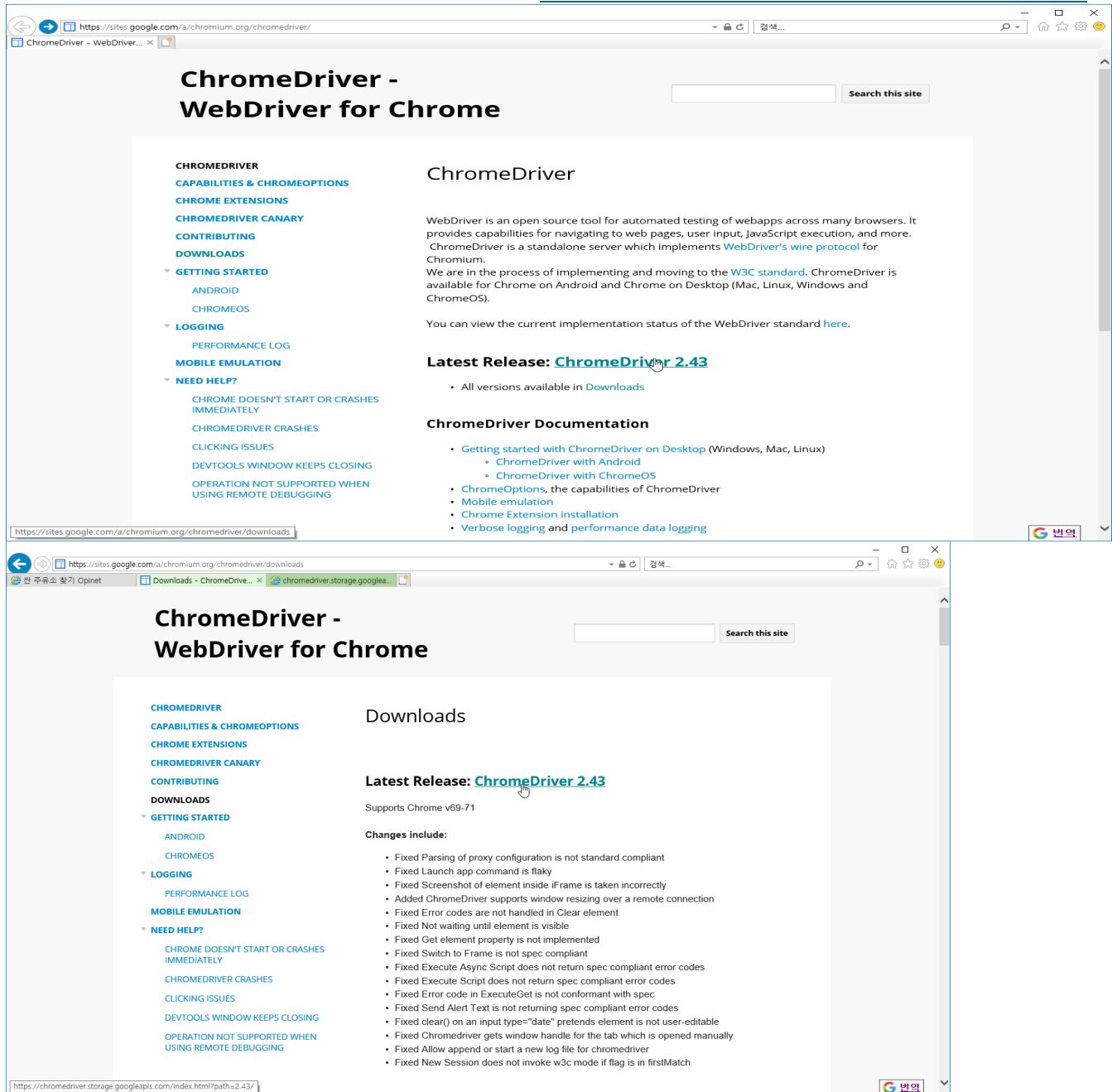
[Selenium 활용한 웹 크롤링(Crawling)]

1) selenium 설치






pip install selenium

2) '크롬 웹 드라이버' Chrome Driver 다운로드

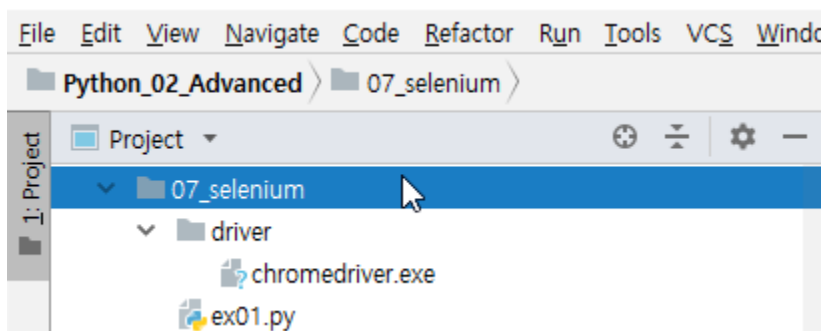
<https://sites.google.com/a/chromium.org/chromedriver>



Index of /2.43/

	Name	Last modified	Size	ETag
	Parent Directory		-	
	chromedriver_linux64.zip	2018-10-17 02:46:13	3.89MB	1a67148288f4320e5125649f66e02962
	chromedriver_mac64.zip	2018-10-17 04:09:49	5.71MB	249108ab937a3bf8ae8fd22366b1c208
	chromedriver_win32.zip	2018-10-17 03:01:50	3.45MB	d238c157263ec7f668e0ea045f29f1b7
	notes.txt	2018-10-17 05:00:45	0.02MB	a84902c9429641916b085a72ad5de724

다음과 같이 driver 폴더 생성하고 크롬 드라이버 복사

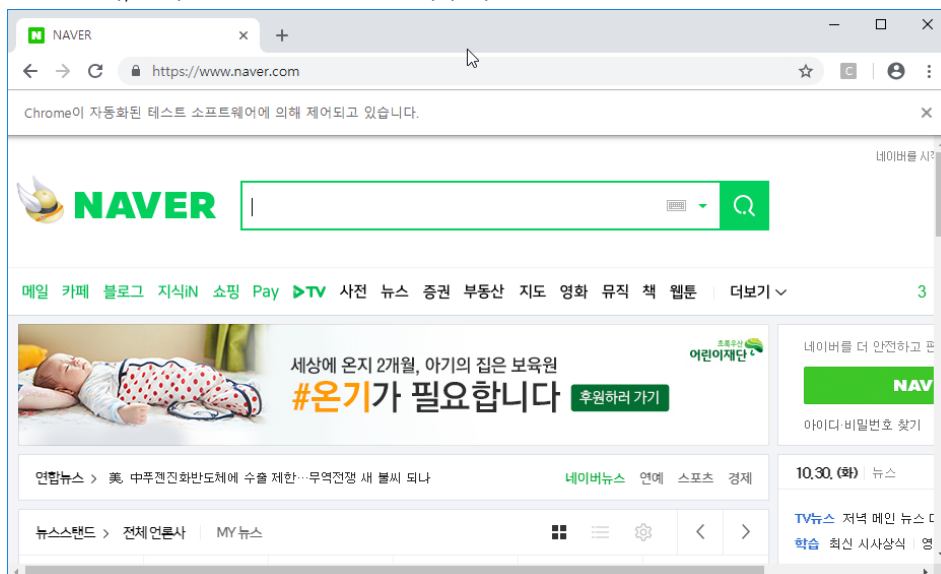


ex01.py

```
from selenium import webdriver

driver = webdriver.Chrome('driver/chromedriver')
driver.get("http://naver.com")
```

실행 결과, 자동으로 크롬 웹 브라우저 뜸.



ex02_login.py 특정 위치에서 타이핑

```

from selenium import webdriver

driver = webdriver.Chrome('driver/chromedriver')
driver.get("https://nid.naver.com/nidlogin.login")

elem_login = driver.find_element_by_id("id")
elem_login.clear()
elem_login.send_keys("자신의 네이버 계정")

elem_login = driver.find_element_by_id("pw")
elem_login.clear()
elem_login.send_keys("자신의 네이버 비번")

```

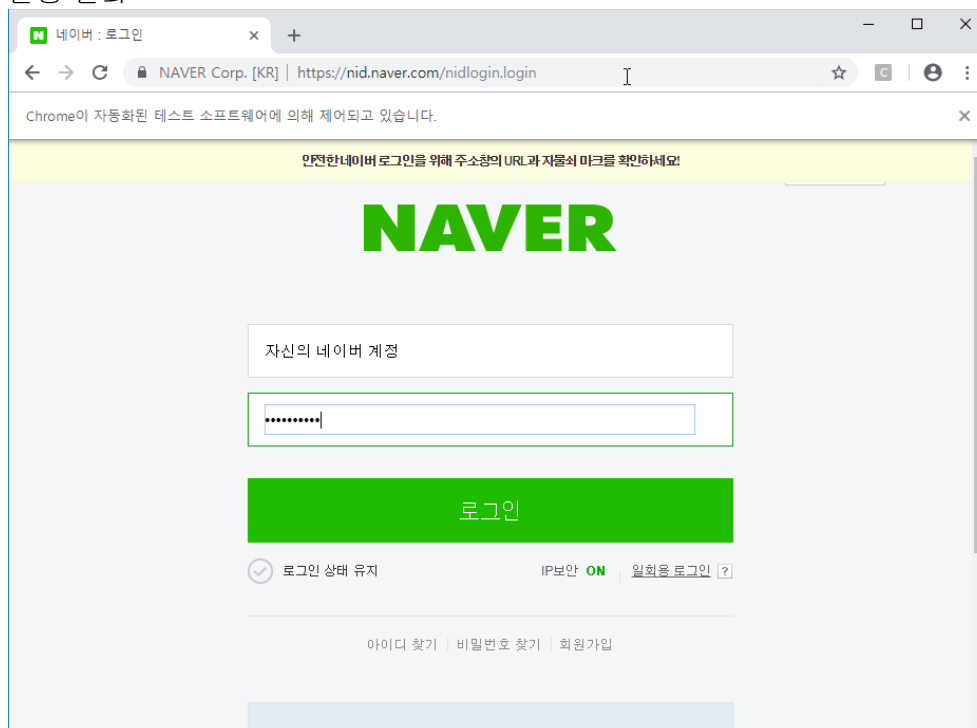
```
elem_login = driver.find_element_by_id("id")
```

--> 현재 크롬에 떠 있는 웹페이지에서 id 속성 값이 id 인 element 찾기

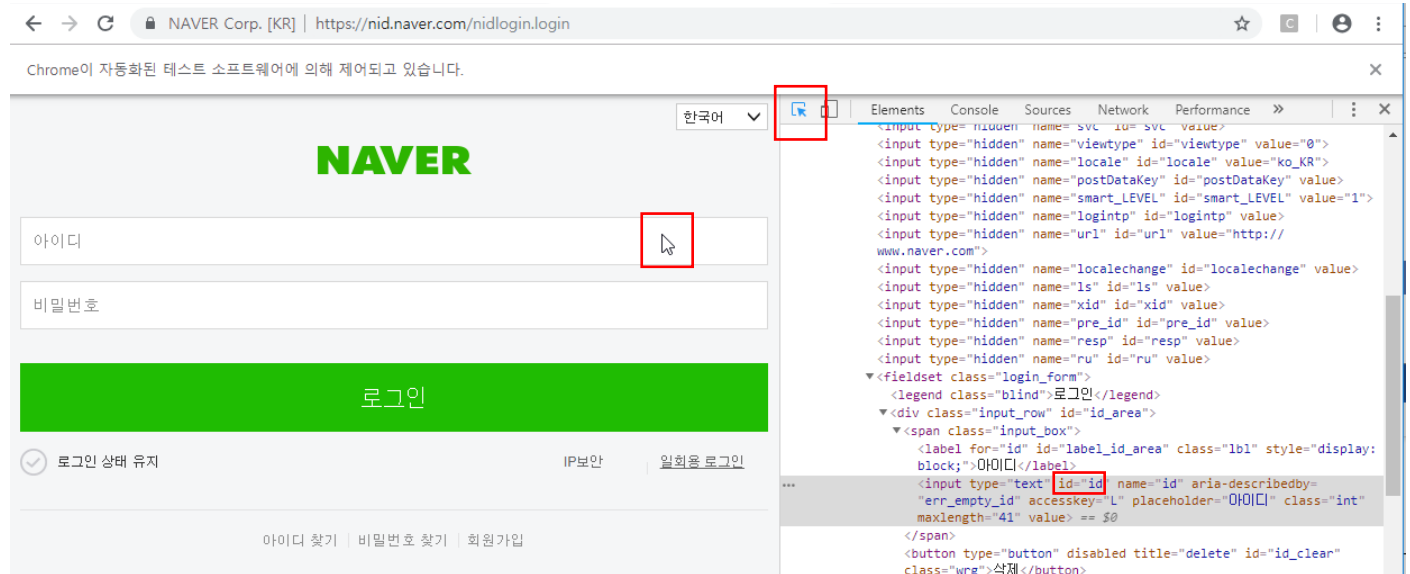
```
elem_login.clear()
```

elem_login.send_keys("자신의 네이버 계정") --> 그곳에 타이핑

실행 결과

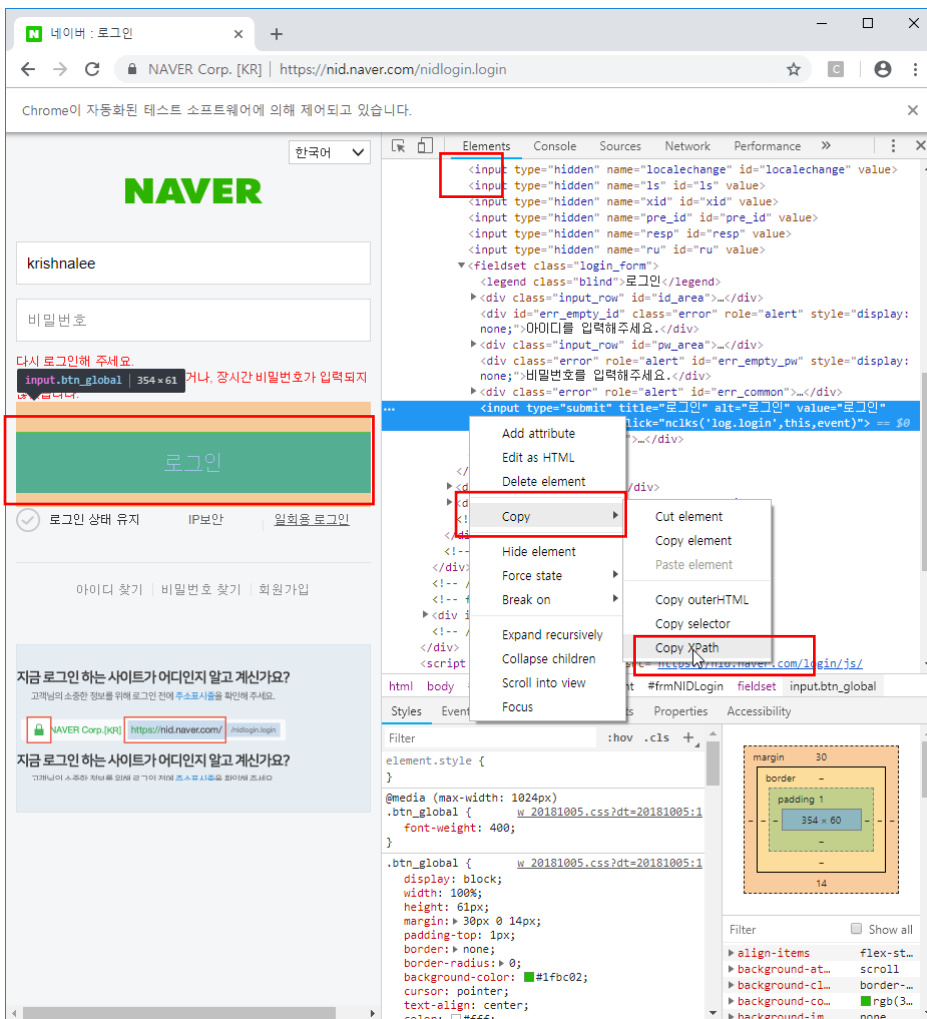


<참고> 찾고자하는 element 의 id 확인 방법 . 크롬에서 F12



ex03_login.py 특정 위치에서 클릭 예

1) 클릭할 곳의 XPath 찾기



복사한 xpath를 다음 코드에 붙여 넣음

```
from selenium import webdriver

driver = webdriver.Chrome('driver/chromedriver')
driver.get("https://nid.naver.com/nidlogin.login")

elem_login = driver.find_element_by_id("id")
elem_login.clear()
elem_login.send_keys("자신의 네이버 계정")

elem_login = driver.find_element_by_id("pw")
elem_login.clear()
elem_login.send_keys("자신의 네이버 비번")

💡
xpath = "//*[@id='frmNIDLogin']/fieldset/input"
driver.find_element_by_xpath(xpath).click()
```

실행 결과:

로그인 버튼이 클릭됨. 네이버는 보안문자를 입력하게 되어 있지만,

보안문자 입력이 없는 사이트는 자동로그인가능.

ex04.py Beautiful soup 과 함께 사용 가능

```
from selenium import webdriver

driver = webdriver.Chrome('driver/chromedriver')
driver.get("https://movie.naver.com/movie/bi/mi/basic.nhn?code=160487")

from bs4 import BeautifulSoup

html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

raw_list = soup.find_all('div', class_="story_area")
print(_raw_list_)
```

실행 결과 :

```
[<div class="story_area">
<div class="title_area">
<h4 class="h_story"><strong class="blind">즐거리</strong> </h4>
</div>
<h5 class="h_tx_story">야귀때가 온 세상을 집어삼켰다!</h5>
<p class="con_tx">밤에만 활동하는 산 자도 죽은 자도 아닌 '야귀(夜鬼)'가 창궐한 세상,
<br/> 위기의 조선으로 돌아온 왕자 '이창'(현빈)은
<br/> 도처에 창궐한 야귀때에 맞서 싸우는 최고의 무관 '박종사관'(조우진)
<br/> 일행을 만나게 되고,
<br/> 야귀때를 소탕하는 그들과 의도치 않게 함께하게 된다.
<br/> 한편, 조선을 집어삼키려는 절대악 '김자준'(장동건)은 이 세상을 뒤엎기 위한
<br/> 마지막 계획을 감행하는데...
<br/>
<br/> 조선필생 VS 조선필망
<br/> 세상을 구하려는 자와 멸망시키려는 자!
<br/> 오늘 밤, 세상에 없던 혈투가 시작된다!</p>
<button class="story_more" id="toggleMakingnoteButton" onclick="storyAndNote.toggleMakingnote();" type="button"><em class="blind">제작노트 보기
</em></button><!-- N=a:mai.story -->
</div>]
```