

LESSON

selenium 모듈

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n"\
    "" + filePath + objName + "####.obj\n\n"\
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 4:
```

selenium 모듈 정의

- 웹 어플리케이션 테스트를 위한 프레임워크
 - 주로 제작한 홈페이지를 테스트하기 위해 사용
- 다양한 언어에서 지원하며 (C++, JAVA, 파이썬 등) 사용자가 아닌 프로그램이 웹 브라우저를 제어할 수 있도록 지원
- 웹 브라우저마다 클라이언트 프로그램(Web Driver)이 별도로 필요(웹 브라우저 ↔ 프로그램 간 통신 목적)
- 크롤링보다는 웹을 제어하는 목적이 더 큼

selenium 모듈 설치

- ❖ 외부 모듈로 BeautifulSoup과 같이 콘솔창, Jupyter Notebook, Anaconda 등을 자유롭게 설치

- pip install selenium

```
[zoostar@~]$ pip install selenium
Collecting selenium
  Downloading https://files.pythonhosted.org/packages/e1/71/90/28/9f9d0613b0a44e5dd6a7f5ca98459853/selenium-3.141.0-py2.py3-none-any.whl (904kB)
    100% |#####| 911kB
Requirement already satisfied: urllib3 in /anaconda3/lib/python3.6/site-packages (from selenium) (1.22)
Installing collected packages: selenium
Successfully installed selenium-3.141.0
```

```
import selenium

print(selenium.__version__)

3.141.0
```

Web Driver 설치

❖ 컴퓨터 운영체제와 웹 브라우저 종류에 따라 개별 설치

-브라우저의 버전, PC 환경 등의 변수가 많아 크롬(Chrome)에서의 실습 권장

[참고] 주요 웹 드라이버 다운로드 경로

- ① 크롬 웹 드라이버 : <https://chromedriver.chromium.org/downloads>
- ② 엣지 웹 드라이버 : <https://docs.microsoft.com/ko-kr/microsoft-edge/webdriver>
- ③ 파이어폭스 웹 드라이버 : <https://github.com/mozilla/geckodriver/releases>
- ④ 인터넷익스플로러 웹 드라이버
: <https://www.microsoft.com/en-us/download/details.aspx?id=44069>

Web Driver 설치

- 실습 환경에 맞는 종류를 다운로드 받아 적당한 경로에 압축 해제(경로에 한글 X)
- 압축 해제한 파일명은 수정하지 말 것

ChromeDriver - WebDriver for Chrome

- CHROMEDRIVER
- CAPABILITIES & CHROMEOPTIONS
- CHROME EXTENSIONS
- CHROMEDRIVER CANARY
- CONTRIBUTING
- ▼ DOWNLOADS
 - VERSION SELECTION
- ▼ GETTING STARTED
 - ANDROID
 - CHROMEOS
- ▼ LOGGING
 - PERFORMANCE LOG

ChromeDriver






WebDriver is an open source tool for automated testing of i
ChromeDriver is a standalone server that implements the \
ChromeOS).

You can view the current implementation status of the Web

All versions available in Downloads

- Latest stable release: [ChromeDriver 79.0.3945.36](#)
- Latest beta release: [ChromeDriver 80.0.3987.16](#)

Index of /80.0.3987.16/

	Name	Last modified	Size
	Parent Directory		-
	chromedriver_linux64.zip	2019-12-19 17:39:26	4.71MB
	chromedriver_mac64.zip	2019-12-19 17:39:28	6.68MB
	chromedriver_win32.zip	2019-12-19 17:39:29	4.17MB
	notes.txt	2019-12-19 17:39:33	0.00MB

selenium 모듈 사용법

- selenium 모듈 호출 후 설치한 Web driver 경로를 지정
- 원하는 홈페이지를 해당 웹 드라이버로 실행

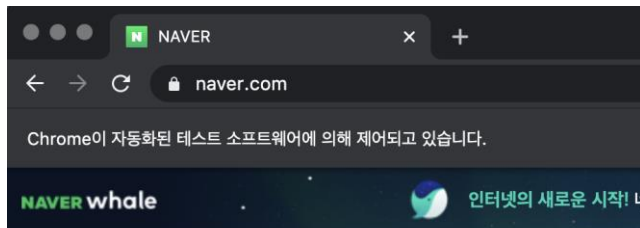
```
import selenium

from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')
```



윈도우의 경우 경로를 적을 때 역슬래시가
이스케이프 문자로 W를 하나 더 적어 주어야 함
C:\\Users\\WW\\Downloads\\WW\\ChromeDriver



NAVER

LESSON

selenium 모듈 활용

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "####.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 3:
```

HTML로 접근하기

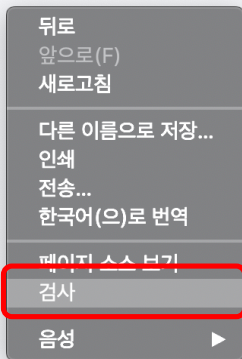
- BeautifulSoup 모듈과 비슷하게 HTML 문서의 구조에 따라 접근할 수 있음
- BeautifulSoup에서 원하는 태그, 속성 등을 가져오기 위해 단일 객체는 `find()`, 복수 객체는 `find_all()` 함수로 접근했다면, selenium에서는 태그 종류에 따라 각각 함수가 있음

단일 객체	복수 객체(리스트 형태)
<code>find_element_by_id()</code> <code>find_element_by_css_selector()</code> <code>find_element_by_class_name()</code> ...	<code>find_elements_by_class_name()</code> <code>find_elements_by_xpath()</code> <code>find_elements_by_tag_name()</code> ...

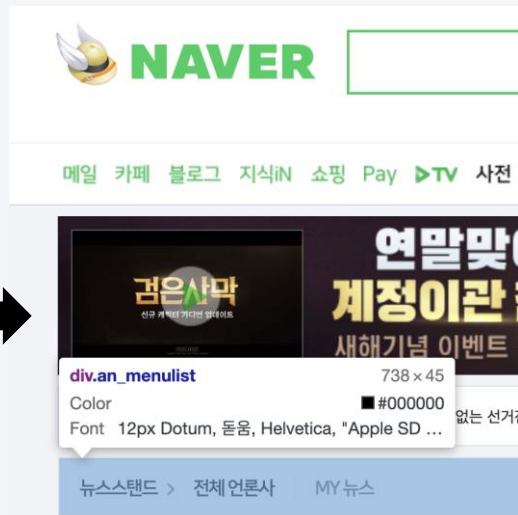
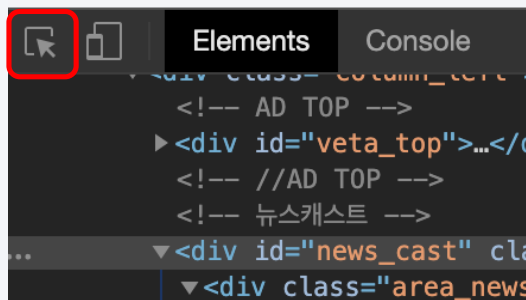
HTML로 접근하기

원하는 데이터의 태그를 쉽게 검색하는 방법(크롬 브라우저)

① 브라우저에서 마우스 우클릭 - [검사] 선택



② 마우스 커서 모양 버튼 클릭 후
원하는 데이터로 마우스 이동



HTML로 접근하기 : find_element_by_id()

🔗 원격으로 브라우저를 열고 해당 id의 정보를 가져옴

```
<div id="news_cast" class="section_newscast">
  <div class="area_newstop">
    <div class="cast_flash">
      <h3 class="cf_tit">
```

```
import selenium
```

```
from selenium import webdriver
```

```
path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')
```

```
element = driver.find_element_by_id("news_cast")
print(element)
```

```
<selenium.webdriver.remote.webelement.WebElement
4756-8ee4-d380c7c774e5">
```

```
print(element.tag_name)
```

div

```
print(element.text)
```

연합뉴스

"공수처는 귀태" vs "검찰이 더
검찰, 백원우 前민정비서관 소환...

네이버뉴스

연예

스포츠

...

HTML로 접근하기 : find_element_by_class_name()

🔗 원격으로 브라우저를 열고 해당 class의 정보를 가져옴

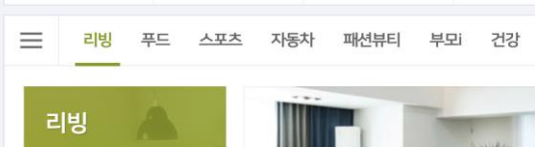
```
import selenium

from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("themecast_category")
print(element)

<selenium.webdriver.remote.webelement.WebElement (session="c47f068;
4c8d-bbac-2f486cfefa99")>
```



element.text

'관심 주제 선택\n리빙\n푸드\n스포츠\n자동차\n패'

HTML로 접근하기 : 주의점

- 웹 브라우저가 실행되기 전이나, 실행 후 브라우저를 종료했다면 세션이 끊기기 때문에 더 이상 코드가 실행되지 않음

```
: element.tag_name
```

```
-----  
NoSuchWindowException
```

```
Traceback
```

```
<ipython-input-401-8e9ba6839ef2> in <module>()  
----> 1 element.tag_name
```

```
NoSuchWindowException: Message: no such window: window was already closed  
(Session info: chrome=79.0.3945.88)
```

이벤트로 제어하기

♣ selenium은 브라우저를 직접 제어하기 때문에 마치 사람이 직접 컨트롤하듯이 마우스 클릭, 키보드 입력, 자바 스크립트 등 이벤트 처리를 할 수 있음

- ① 마우스 클릭 : `click()`
- ② 키보드 입력 : `send_keys()`
- ③ 자바스크립트 삽입 : `execute_script()`
- ④ 입력 양식 전송 : `submit()`
- ⑤ 스크린샷 : `screenshot(파일이름)`
- ⑥ 글자 지움 : `clear()`
- ⑦ 뒤로 가기 : `back()`
- ⑧ 앞으로 가기 : `forward()`



이벤트로 제어하기 : click()

🔗 find_element_by_XX로 검색한 특정 요소를 클릭



예 : Naver 메인 페이지 사전 바로가기 클릭

a.an_a.mn_dic 26 x 16

메일 카페 블로그 지식IN 쇼핑 Pay ▶TV 사전 뉴스 증권 부동산

```

Elements Console Sources Network Performance Memory Application Security Audits
<html lang="ko">
  <head>...</head>
  <body class>
    <!-- 스킵 내비게이션 -->
    <div class="u_skip">...</div>
    <!-- //스킵 내비게이션 -->
    <div id="PM_ID_ct" class="wrap">
      <!-- 헤더 -->
      <div class="header" role="banner">
        <div class="special_bg">...</div>
        <div class="section_navbar">
          <div class="area_navigation" role="navigation">
            <ul class="an_l">...</ul>
            <ul id="PM_ID_serviceNavi" class="an_l" style="display: block;">
              <li class="an_item">
                <a href="https://dict.naver.com/" class="an_a mn_dic" data-clk="svc.dic"> == $0
  ...
  
```

이벤트로 제어하기 : click()

🔗 find_element_by_XX로 검색한 특정 요소를 클릭



예 : Naver 메인 페이지 사전 바로가기 클릭

```
import selenium

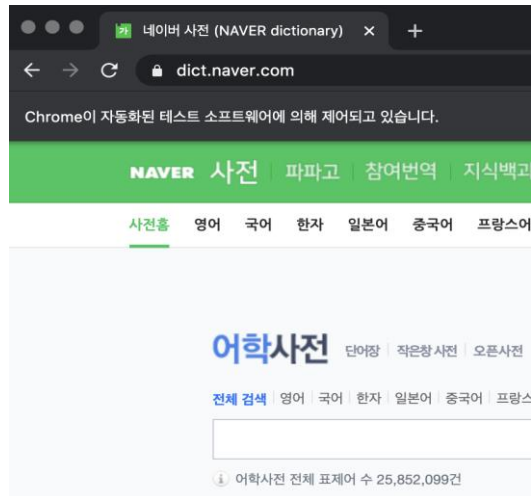
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_class_name("mn_dic")
print(element)

element.click()

<selenium.webdriver.remote.webelement.WebElement (session=48a4-ac96-f521c085258f)>
```



이벤트로 제어하기 : 주의점

- ❖ selenium은 웹 드라이버로 브라우저를 직접 띄우고 제어하기 때문에 상대적으로 로딩이 오래 걸리는 단점이 있음
- ❖ 많은 작업을 한번에 처리하는 경우 처리 속도를 따라가지 못해 오류 발생 가능
 - time 모듈의 sleep 함수와 함께 활용하여 조금씩 시간차를 두고 실행하는 것을 권장

```
import selenium
import time
from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')
time.sleep(2)
element = driver.find_element_by_class_name("mn_dic")
time.sleep(2)
element.click()
element = driver.find_element_by_id("_id_hanja_href")
time.sleep(2)
element.click()
```

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

NAVER 사전 | 파파고 | 참여번역 | 지식백과

사전홈 영어 국어 한자 일본어 중국어 프랑스어

한자사전

이벤트로 제어하기 : send_keys()

🔗 find_element_by_XX로 검색한 특정 요소에 키보드 입력



예 : Naver 메인 페이지의 검색창에
웹 크롤링 검색어 입력



NAVER

input#query.input_text 405 x 23

메일 카페 블로그 지식IN 쇼핑 Pay TV 사전 뉴스 증권

```

Events Console Sources Network Performance Memory Application Security Audits
<input type="hidden" id="qdt" name="qdt" value disabled="disabled">
<input type="hidden" id="ie" name="ie" value="utf8">
<input type="hidden" id="acir" name="acir" value disabled="disabled">
<input type="hidden" id="os" name="os" value disabled="disabled">
<input type="hidden" id="bid" name="bid" value disabled="disabled">
<input type="hidden" id="pkid" name="pkid" value disabled="disabled">
<input type="hidden" id="eid" name="eid" value disabled="disabled">
<input type="hidden" id="mra" name="mra" value disabled="disabled">
<span class="green_window">
  <input id="query" name="query" type="text" title="검색어 입력" maxlength="255" class="input_text" tabindex="1" accesskey=
    .value=1;" value> == $0
  
```

이벤트로 제어하기 : send_keys()

🔗 find_element_by_XX로 검색한 특정 요소에 키보드 입력



예 : Naver 메인 페이지의 검색창에
웹 크롤링 검색어 입력

```
import selenium

from selenium import webdriver

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_id("query")
print(element)

element.send_keys("웹 크롤링")

<selenium.webdriver.remote.webelement.WebElement:
45f1-ba79-6f2a4a03a0c7">
```

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

NAVER whale



인터넷의 새로운 시작! 네이버 웨일로 차원이 다른 웹서핑을 경험해보세요!



NAVER

웹 크롤링

웹 크롤링

웹 크롤링 테크닉

웹 크롤링과 스크레이핑

웹 클리핑 응용 프로그램

메일 카페 블로그 지식IN 쇼핑

이벤트로 제어하기 : 특수 키 입력

- 키보드를 제어할 때, 특수 키를 입력하기 위해서는 추가적인 모듈 호출이 필요함

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

- Keys.특수키로 사용

```
ARROW_DOWN, ARROW_LEFT, ARROW_RIGHT, ARROW_UP BACKSPACE,  
DELETE, HOME, END, INSERT, ALT, COMMAND, CONTROL, SHIFT ENTER,  
ESCAPE , SPACE, TAB F1, F2 , F3 .....F12
```

이벤트로 제어하기 : 특수 키 입력

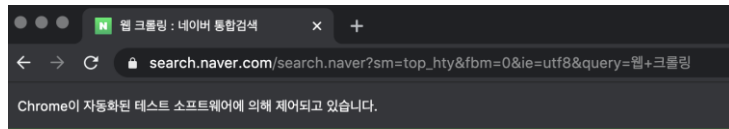


예 : Naver 메인 페이지에서 검색어 입력 후 엔터키 입력

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

path = "/Users/zoostar/Downloads/chromedriver"
driver = webdriver.Chrome(path)
driver.get('https://www.naver.com')

element = driver.find_element_by_id("query")
element.send_keys('웹 크롤링')
element.send_keys(Keys.ENTER)
```



NAVER

웹 크롤링



통합검색

블로그

책

이미지

지식백과

카페

포스트

웹사이트

더보기

검색음