

# 스파르타온라인 파이썬 혼자놀이 패키지 - 1일차

## [수업 목표]

1. 파이썬 기초 문법을 안다.
2. 원하는 페이지를 크롤링할 수 있다.
3. 이미지 url로부터 파일을 다운로드받을 수 있다.

## [목차]

- 01. 오늘 배울 것
- 02. 파이썬 시작하기
- 03. 파이썬 기초 공부
- 04. 파이썬 패키지 사용하기
- 05. 이미지 웹 스크래핑(크롤링)하기
- 06. 끝 & 속제 설명

## 01. 오늘 배울 것

- 오늘은 첫 시간인 만큼 파이썬과 친해지는 시간을 가져봅시다!
- ▼ 이번 시간이 끝나면 완성하는 것

바탕 화면 > sparta > imgs



## 02. 파이썬 시작하기

### ▼ 1) 파이썬을 설치한다는 것의 의미



#### 파이썬을 설치한다?

→ 일종의 번역팩을 설치한다고 생각하면 됩니다. 컴퓨터는 101010001 과 같은 언어만 알아듣는다고 했지요? 파이썬 문법으로 된 것을 101010001로 변환해줄 수 있도록, 번역 패키지를 설치하는 것입니다.

### ▼ 2) 파이썬에서 새 프로젝트 만들기



시작하기에 앞서 '가상환경'을 설정해줄 거예요!!

- 바탕화면에 sparta 폴더를 만들어줍니다.
- 파이썬 실행!
- + Create New Project 를 선택해주세요.  
(이미 프로젝트를 여신 분은 File > New Project...를 클릭해주세요)
- Location에서 방금 만든 sparta 폴더를 찾아주세요
- Create 클릭해서 새 프로젝트를 만들어줍니다.

### ▼ 3) 첫 파이썬 파일 실행하기!

- sparta 폴더 안에 hello.py라는 이름으로 파일을 하나 만들어줍니다.
- 다음 내용을 붙여넣어주세요.

```
print ('hello sparta')
```

- Run > Run... > hello.py 실행!

### ▼ 4) 파이썬 문법을 시작하기에 앞서..

- 파이썬은 매우 직관적인 언어이고, 할 수 있는 것도 많습니다. 그런데, 개발자들도 모든 문법을 기억하기란 쉽지 않습니다. 오늘 배우는 것 외에 필요한 것들은 구글링해서 찾아보면 됩니다!
- 단, tuple과 set 자료형, class는 기초 단계에선 다루지 않겠습니다. (몰라도 괜찮고, 많은 개발자들도 모릅니다. 나중에 필요한 일이 생겼을 때 배워야 의미를 알 수 있습니다!)

## 03. 파이썬 기초 공부

### ▼ 5) 파이썬 기초 문법

#### ▼ 변수 & 기본연산

```
a = 3      # 3을 a에 넣는다
b = a      # a를 b에 넣는다
a = a + 1  # a+1을 다시 a에 넣는다

num1 = a*b # a*b의 값을 num1이라는 변수에 넣는다
num2 = 99  # 99의 값을 num2이라는 변수에 넣는다

# 변수의 이름은 마음대로 지을 수 있음!
# 진짜 "마음대로" 짓는 게 좋을까? var1, var2 이렇게?
```

#### ▼ 자료형

- 숫자, 문자형

```
name = 'bob' # 변수에는 문자열이 들어갈 수도 있고,
num = 12 # 숫자가 들어갈 수도 있고,

is_number = True # True 또는 False -> "Boolean"형이 들어갈 수도 있습니다.

#####
# 그리고 List, Dictionary 도 들어갈 수도 있죠. 그게 원지는 아래에서!
```

- 리스트 형 (Javascript의 배열형과 동일)

```
a_list = []
a_list.append(1) # 리스트에 값을 넣는다
a_list.append([2,3]) # 리스트에 [2,3]이라는 리스트를 다시 넣는다

# a_list의 값은? [1,[2,3]]
# a_list[0]의 값은? 1
# a_list[1]의 값은? [2,3]
# a_list[1][0]의 값은? 2
```

- Dictionary 형 (Javascript의 dictionary형과 동일)

```
a_dict = {}
a_dict = {'name':'bob','age':21}
a_dict['height'] = 178

# a_dict의 값은? {'name':'bob','age':21, 'height':178}
# a_dict['name']의 값은? 'bob'
# a_dict['age']의 값은? 21
# a_dict['height']의 값은? 178
```

- Dictionary 형과 List형의 조합

```
people = [{'name':'bob','age':20},{'name':'carry','age':38}]

# people[0]['name']의 값은? 'bob'
# people[1]['name']의 값은? 'carry'

person = {'name':'john','age':7}
people.append(person)

# people의 값은? [{'name':'bob','age':20},{'name':'carry','age':38},{'name':'john','age':7}]
# people[2]['name']의 값은? 'john'
```

## ▼ 조건문

- if / else 로 구성!

```
if age > 20:
    print('성인입니다') # 조건이 참이면 성인입니다를 출력
else:
    print('청소년이에요') # 조건이 거짓이면 청소년이에요를 출력

is_adult(30)
# 무엇이 출력될까요?
```

## ▼ 반복문



파이썬에서의 반복문은, 리스트의 요소들을 하나씩 꺼내쓰는 형태입니다.

- 즉, 무조건 리스트와 함께 쓰입니다!

```
fruits = ['사과','배','감','귤']

for fruit in fruits:
    print(fruit)
```

```
# 사과, 배, 감, 귤 하나씩 꺼내어 찍힙니다.
```

- 살짝 응용해볼까요? - 과일 갯수 세기 함수

#### ▼ [코드스니펫] - 리스트 예제

```
fruits = ['사과', '배', '배', '감', '수박', '귤', '딸기', '사과', '배', '수박']
```

```
fruits = ['사과', '배', '배', '감', '수박', '귤', '딸기', '사과', '배', '수박']

count = 0
for fruit in fruits:
    if fruit == '사과':
        count += 1

print(count)

# 사과의 갯수를 세어 보여줍니다.
```

- 다른 예제를 살펴봅시다.

#### ▼ [코드스니펫] - 딕셔너리 예제

```
people = [{'name': 'bob', 'age': 20},
          {'name': 'carry', 'age': 38},
          {'name': 'john', 'age': 7},
          {'name': 'smith', 'age': 17},
          {'name': 'ben', 'age': 27}]
```

```
people = [{'name': 'bob', 'age': 20},
          {'name': 'carry', 'age': 38},
          {'name': 'john', 'age': 7},
          {'name': 'smith', 'age': 17},
          {'name': 'ben', 'age': 27}]

# 모든 사람의 이름과 나이를 출력해봅시다.
for person in people:
    print(person['name'], person['age'])

# 이번엔, 반복문과 조건문을 응용해봅시다.
# 20세 보다 많은 사람만 출력하기
for person in people:
    if person['age'] > 20:
        print(person['name'], person['age'])
```

#### ▼ 그 밖의 내장 함수들

##### 👉 문자열 자르기

```
txt = 'sparta@gmail.com'

result = txt.split('@')[1].split('.')[0]

print(result)
```

##### 👉 문자열 바꾸기

```
txt = 'sparta@gmail.com'

result = txt.replace('gmail', 'naver')
```

```
print(result)
```

## 04. 파이썬 패키지 사용하기

### ▼ 6) 파이썬 패키지(package) 설치하기



패키지? 라이브러리? →

Python 에서 패키지는 모듈(일종의 기능들 묶음)을 모아 놓은 단위이고, 이런 패키지의 묶음을 라이브러리라고 합니다. 하지만 보통 패키지와 라이브러리는 혼용해서 많이 써요 😊 지금 여기서는 외부 라이브러리를 사용하기 위해서 패키지를 설치합니다.

즉, 여기서는 **패키지 설치 = 외부 라이브러리 설치!**

### ▼ 1. 가상 환경(virtual environment) 이란? - 프로젝트별로 패키지들을 담을 공구함



#### 문제상황:

회사에서는 패키지 A, B, C를 설치해서 쓰고,  
개인 프로젝트에서는 패키지 B, C, D, E를 설치해서 쓰고 있었어요.

그런데 회사팀장님이 B를 이전 버전인 B' 로 쓰자고 하시네요.  
그렇게 되면, 같은 컴퓨터에 깔려 있는 개인 프로젝트에서는 B' 로 쓰면 코드를 다 바꿔야 해요 😞

어떻게 하면 좋을까요?



#### 해결책:

다 담아둘 필요 없이 공구함을 2개 만들어서,

공구함1에 A, B', C를 담아두고,  
공구함2에 B, C, D, E를 담아두고 쓰면 관리하기 편하겠죠?

그래서, 가상환경이라는 개념이 등장했습니다.  
즉, **프로젝트별 공구함** 이예요.



정리하자면,

#### 가상환경(virtual environment)은

같은 시스템에서 실행되는 다른 파이썬 응용 프로그램들의 동작에 영향을 주지 않기 위해, 파이썬 배포 패키지들을 설치하거나 업그레이드하는 것을 가능하게 하는 **격리된 실행 환경** 입니다.

출처 : [파이썬 공식 용어집- 가상환경](#)

### ▼ 2. 가상 환경 설치 및 확인



우린 이미 완료했으니, 확인만!

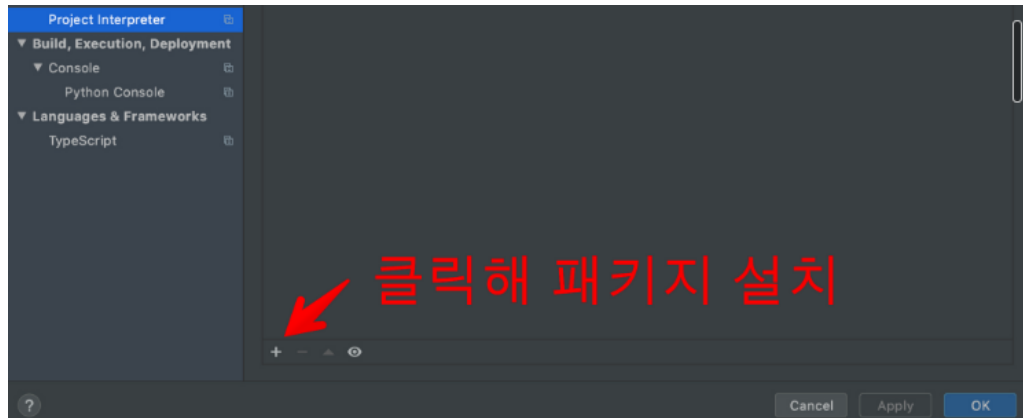
- Windows : File → setting → project interpreter
- Mac : Preference → Project Interpreter

### ▼ 3. pip(python install package) 사용 - dload 패키지 설치해보기

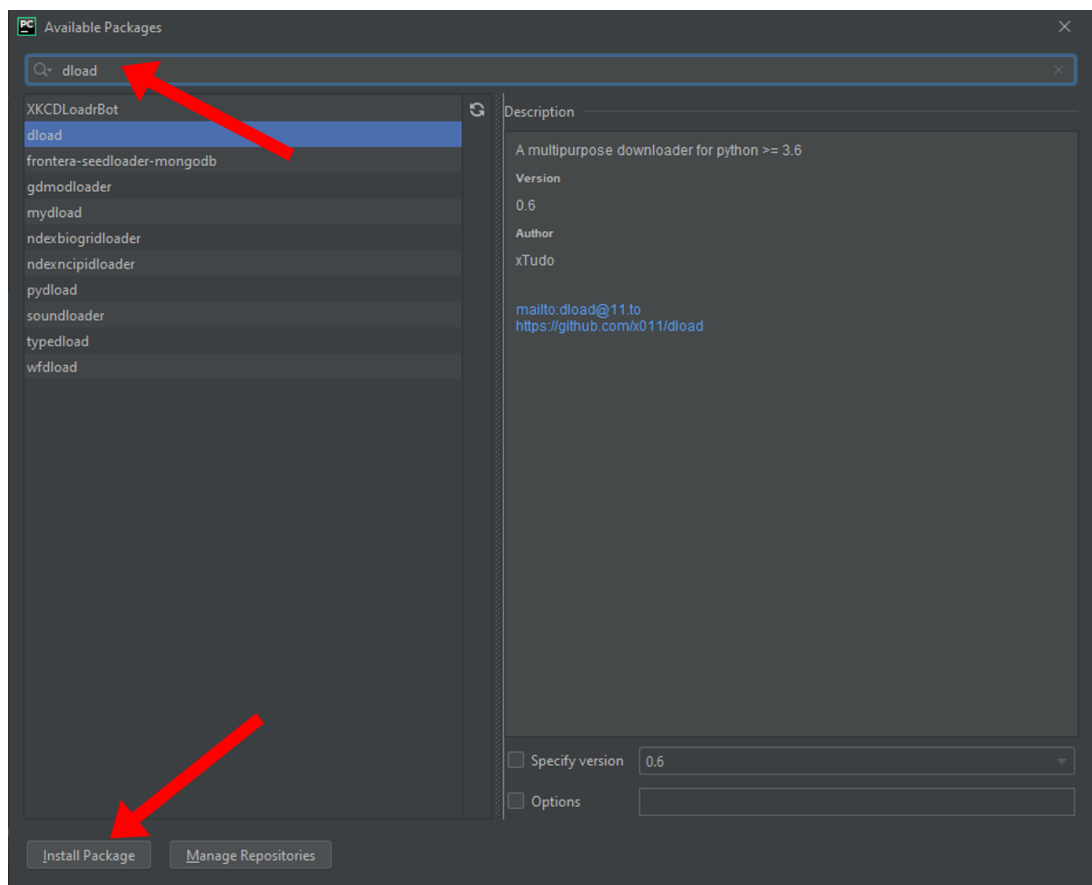
👉 앱을 설치할 때 앱스토어/플레이스토어를 가듯이, 새로운 프로젝트의 라이브러리를 가상환경(공구함)에 설치하려면 pip 를 이용하게 됩니다.

- project interpreter 화면에서 + 버튼을 누르면 아래 창이 뜹니다!

💡 맥은 아래 쪽에, 윈도우는 오른쪽에 있어요. 사진은 맥 버전!



- dload를 검색해서, 인스톨하면 끝!



### ▼ 7) dload 라이브러리 사용해보기

- `dload` 는 URL로부터 파일 다운로드를 해주는 패키지입니다. 이 패키지를 이용해서 이미지를 내려받아볼게요!

#### ▼ [코드스니펫] - dload 써보기

```
import dload

dload.save("https://spartacodingclub.kr/static/css/images/ogimage.png")
```


- 다른 이름으로 저장도 해볼까요?

```
import dload

dload.save("https://spartacodingclub.kr/static/css/images/ogimage.png", 'sparta.png')
```

## 05. 이미지 웹 스크래핑(크롤링)하기

### ▼ 8) 웹 스크래핑이란?

- 신문 스크래핑  해보셨나요? 신문에서 원하는 기사만 오려서 정보를 추리는 것을 스크래핑이라고 하죠? 웹 스크래핑도 똑같습니다.



웹 스크래핑(web scraping)은 웹 페이지에서 우리가 원하는 부분의 데이터를 수집해오는 것입니다. (한국에서는 같은 작업을 크롤링 *crawling* 이라는 용어로 혼용해서 씁니다.)

- 구글 검색할 때는 **web scraping** 으로 검색해야 우리가 배우는 페이지 추출에 대한 결과가 나오니 참고하세요.
- 참고  
[Web Scraping\(wikipedia\)](#) / [Web Crawler\(wikipedia\)](#)  
[Web Scraping vs Web Crawling: What's the Difference?](#)

### ▼ 9) 파이썬으로 브라우저 제어하기



웹 스크래핑을 하기 위해서는 특정 웹페이지를 구성하는 HTML 정보를 받아와야하는데요! 여러 방법이 있지만 우리는 파이썬으로 크롬 브라우저를 직접 제어하여 웹페이지를 띄우고 HTML을 읽어오도록 하겠습니다

- `selenium` 패키지를 이용하면 브라우저를 제어할 수 있습니다.

#### ▼ 우선 패키지를 설치해볼까요? `dload` 설치할 때와 똑같은 방법으로 하면 됩니다 🤖

- project interpreter 로 접근해서 + 버튼 누르기
  - ↳ Windows : file → setting → project interpreter
  - ↳ Mac : preference → project interpreter로 접근
- selenium 검색해서 install 버튼 눌러 설치
- 셀레니움을 사용하기 위해서는 웹드라이버라는 파일도 필요하답니다.



크롬 브라우저를 열어 Settings > About Chrome에서 버전이 85.X이 있는지 확인해주세요!

#### ▼ [코드스니펫] - 셀레니움 웹드라이버 설치 링크

```
https://chromedriver.storage.googleapis.com/index.html?path=85.0.4183.87/
```

- 이 링크에서 운영체제에 맞는 파일을 찾아 다운로드 받고, 압축을 풀어서 sparta 폴더로 옮겨주세요.
- 이제 파이썬으로 브라우저를 제어해보겠습니다!

### ▼ [코드스니펫] - 셀레니움 시작코드

```
from selenium import webdriver
driver = webdriver.Chrome('chromedriver')

driver.get("http://www.naver.com")
```

### ▼ 10) 웹 스크래핑 해보기

🔥 바로 직접 해보면서 배워보도록 하죠! 다음에서 '아이유'를 검색해 이미지들을 스크래핑해올게요 😊

- 이미지를 스크래핑해올 주소

👉 네이버, 다음 둘 다 해봤는데 다음이 이미지 사이즈가 조금 더 크더라고요 😊

### ▼ [코드스니펫] - 다음검색결과

```
https://search.daum.net/search?w=img&nil_search=btn&DA=NTB&enc=utf8&q=%EC%95%84%EC%9D%B4%EC%9C%A0
```

- 패키지 추가 설치하기 (beautifulsoup4)

👉 project interpreter에서 bs4 를 검색해 다운로드받아주세요!

- 크롤링 기본 세팅

### ▼ [코드스니펫] - 크롤링 기본 세팅

```
from bs4 import BeautifulSoup
from selenium import webdriver
import time

driver = webdriver.Chrome('chromedriver') # 웹드라이버 파일의 경로
driver.get("https://search.daum.net/search?w=img&nil_search=btn&DA=NTB&enc=utf8&q=%EC%95%84%EC%9D%B4%EC%9C%A0")
time.sleep(5) # 5초 동안 페이지 로딩 기다리기

req = driver.page_source
# HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만들
# soup이라는 변수에 "파싱 용이해진 html"이 담긴 상태가 됨
# 이제 코딩을 통해 필요한 부분을 추출하면 된다.
soup = BeautifulSoup(req, 'html.parser')

#####
# 이제 여기에 코딩을 하면 됩니다!
#####

driver.quit() # 끝나면 닫아주기
```

- select / select\_one의 사용법을 익혀봅시다.

👉 이미지 링크 가져와보기!

👉 태그 안의 텍스트를 찍고 싶을 땐 → 태그.text  
태그 안의 속성을 찍고 싶을 땐 → 태그['속성']

```
from bs4 import BeautifulSoup
from selenium import webdriver
```



```
import time

driver = webdriver.Chrome('chromedriver') # 웹드라이버 파일의 경로
driver.get("https://search.daum.net/search?w=img&nil_search=btn&DA=NTB&enc=utf8&q=%EC%95%84%EC%9D%B4%EC%9C%A0")
time.sleep(5) # 5초 동안 페이지 로딩 기다리기

req = driver.page_source

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

thumbnails = soup.select("#imgList > div > a > img")

for thumbnail in thumbnails:
    src = thumbnail["src"]
    print(src)

driver.quit() # 끝나면 닫아주기
```

- BeautifulSoup 내 select에 미리 정의된 다른 방법을 알아봅니다

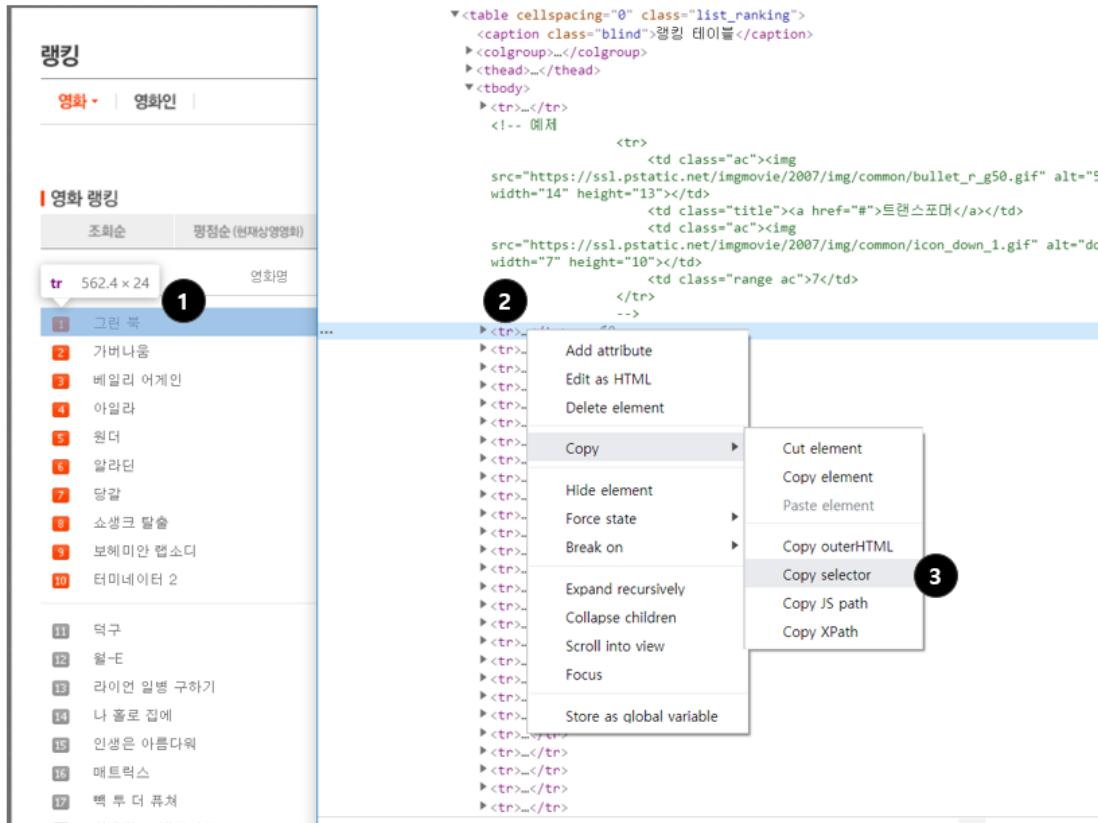
```
# 선택자를 사용하는 방법 (copy selector)
soup.select('태그명')
soup.select('.클래스명')
soup.select('#아이디명')

soup.select('상위태그명 > 하위태그명 > 하위태그명')
soup.select('상위태그명.클래스명 > 하위태그명.클래스명')

# 태그와 속성값으로 찾는 방법
soup.select('태그명[속성="값"]')

# 한 개만 가져오고 싶은 경우
soup.select_one('위와 동일')
```

- 항상 정확하지는 않으나, 크롬 개발자도구를 참고할 수도 있습니다.
  1. 원하는 부분에서 마우스 오른쪽 클릭 → 검사
  2. 원하는 태그에서 마우스 오른쪽 클릭
  3. Copy → Copy selector로 선택자를 복사할 수 있음



#### ▼ 11) 이미지 저장하기

- 위에서 설치해놓은 `dload` 파이썬 패키지를 이용해서 이미지를 다운로드 받아봅시다!
- 우선 이미지를 저장할 `imgs` 폴더를 만들어주세요.
- 각 파일의 이름은 1.jpg, 2.jpg, ... 이런 식으로 저장해줄게요.

```
import dload
from bs4 import BeautifulSoup
from selenium import webdriver
import time

driver = webdriver.Chrome('chromedriver') # 웹드라이버 파일의 경로
driver.get("https://search.daum.net/search?w=img&nil_search=btn&DA=NTB&enc=utf8&q=%EC%95%84%EC%9D%B4%EC%9C%A0")
time.sleep(5) # 5초 동안 페이지 로딩 기다리기

req = driver.page_source

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

thumbnails = soup.select("#imgList > div > a > img")

i=1
for thumbnail in thumbnails:
    src = thumbnail["src"]
    dload.save(src, f'imgs/{i}.jpg')
    i+=1

driver.quit() # 끝나면 닫아주기
```

- 완성!



## 06. 끝 & 숙제 설명

다음에서 좋아하는 연예인 이미지 검색해서 저장하기

### ▼ [코드스니펫] - 크롤링 시작 코드

```
from bs4 import BeautifulSoup
from selenium import webdriver
import time

driver = webdriver.Chrome('chromedriver')
driver.get("#") # 여기에 URL을 넣어주세요
time.sleep(5)

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

#####
# 이제 여기에 코딩을 하면 됩니다!
#####

driver.quit() # 끝나면 닫아주기
```

▼ imgs\_homework 폴더에 아래와 같이 파일이 저장되면 성공! (아래 사진은 수지님)

