

파이썬 프로그래밍 기초2

목차

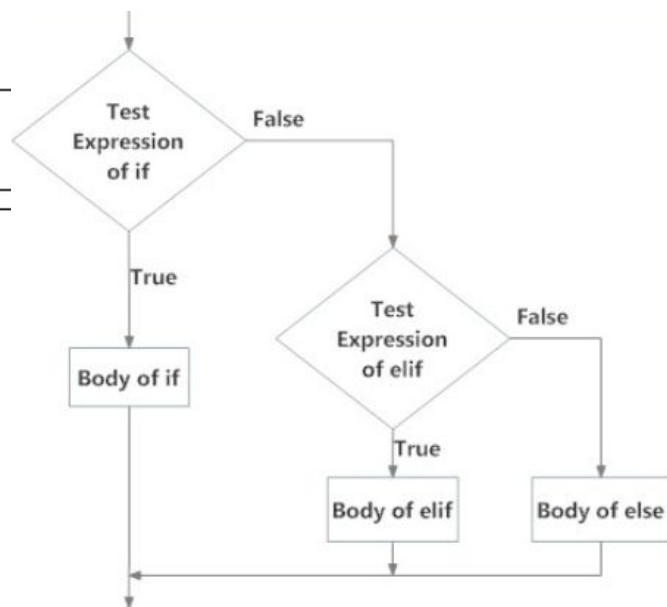
1. 조건문
2. 반복문
3. 문자열 인덱싱
4. 문자열 슬라이싱
5. 함수

조건문 #1

if_example.py

- if의 조건에 맞으면 아래의 코드블럭을 실행한다.
- if의 조건에 맞지 않으면 **elif**와 **else**를 실행할 수 있다.
- indent(들여쓰기) 중요 - Tab키사용

```
>>> if name == 'park':  
>>>     print("my name is park")  
  
>>> name = input('what is your name? ')  
>>> if name == 'park':  
>>>     print('my name is park')  
>>> elif name == 'lee':  
>>>     print('my name is lee')  
>>> else:  
>>>     print("I don't know your name")  
what is your name? park  
my name is park
```



반복문 #1

for_example1.py

- 반복문의 가장 기본적인 형태는 횟수를 기준으로 반복한다.
- 반드시 들여쓰기(indent)가 있어야 한다.

```
>>> for i in [0, 1, 2]:  
>>>     print(i)  
>>>     print("hi~")  
0  
hi  
1  
hi  
2  
hi
```

반복문 #2

for_example1.py

```
>>> for i in range(0,5):  
>>>     print(i, end=' ')  
0 1 2 3 4
```

- 'for i in range(0, 10):' i는 인덱스라는 이름으로 불리며, 0에서 9까지 매 루프마다 값이 변함
- 'for i in range(0, 10):' range(0, 10)은 사실 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]와 같은 의미
- 숫자를 하나씩 일일이 적으며 표현하기엔 너무 귀찮은 점이 많고 비효율적이어서 나온 것이다.
- range(x, y)라고 할 때, x~(y-1)까지 범위의 리스트를 생성해줌
- i는 0부터 9까지 값이 변하며 코드블럭에 있는 print(i)라는 내용을 수행함

반복문 #3

for_example1.py

[문제] **for**문을 이용하여 3부터 10까지의 합을 구하고 더해지는 과정과 그 합을 출력하시오.

```
>>> sum=0
>>> for i in range(3, 11):
>>>     sum+=i
>>>     print(sum)
```

반복문 #3

for_example1.py

[문제] 별의 개수를 입력받아 아래와 같이 출력해 보자.

```
3
*
**
***
```

별의 개수 출력: 6

```
*
**
***
****
*****
*****
*****
```

반복문 #3

for_example1.py

[정답] 별의 개수를 입력받아 아래와 같이 출력해 보자.

```
star = int(input('별의 개수 출력: '))

for i in range(1,star+1):
    print(i*'*')
```


반복문 #3

for_example1.py

[정답] 별의 개수를 입력받아 아래와 같이 출력해 보자.

```
star = int(input('별의 개수 출력: '))

for i in range(1,star+1):
    print(i*'*')
```

반복문 #4

-중첩for문

- for문을 중첩할 수 있다.

for_example2.py

```
>>> for i in range(1, 10):  
>>>     for j in range(1, 10):  
>>>         print(i * j)
```

반복문 #4

[문제]

- 구구단의 n단을 입력 받아 출력해 보자

for_example2.py

```
input number : 9
```

```
9 18 27 36 45 54 63 72 81
```

반복문 #4

[정답]

- 구구단의 n단을 입력 받아 출력해 보자

for_example2.py

```
gugu_dan = int(input('input number : '))  
for i in range(1,10):  
    print(gugu_dan*i, end=' ')
```

문자열 인덱싱 #1

- 문자열은 뭉쳐있는 것 같지만 각각 '인덱스'를 가지고 있다.
- 인덱스는 글자별로 부여되며 0부터 시작된다.

indexing_example.py

1	2	3	4	5
st[0]	st[-4]	st[-3]	st[-2]	st[-1]

```
>>> st = [1, 2, 3, 4, 5]
>>> print(st[-1], st[-2], st[-3])
5 4 3
```

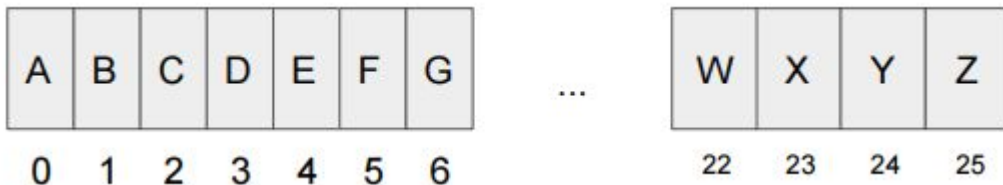
```
>>> abc_string = "abcdefg"
>>> print(abc_string[0])
a
>>> print(abc_string[3])
d
```

문자열 인덱싱 #1

[문제] 아래 `abc_string`을 이용하여 `apple`을 출력하시오.

`abc_string='abcdefghijklmnopqrstuvwxyz'`

`indexing_example.py`



문자열 인덱싱 #1

[정답] 아래 `abc_string`을 이용하여 `apple`을 출력하시오.

`indexing_example.py`

`abc_string='abcdefghijklmnopqrstuvwxyz'`

```
1  abc_string = 'abcdefghijklmnopqrstuvwxyz'
2  print(abc_string[0], end='')
3  print(abc_string[15], end='')
4  print(abc_string[15], end='')
5  print(abc_string[11], end='')
6  print(abc_string[4], end='')
```

문자열 슬라이싱 #1

- 부분부분을 잘라내는 것
- 문자열 슬라이싱을 통해 원하는 부분의 문자열을 가져올 수 있음
- 인덱싱은 하나씩 가져오는 것이고, 슬라이싱은 범위를 지정하는 것이다.

slicing_example.py



문자열 슬라이싱 #1

[문제] 아래 문자열을 인덱싱을 이용하여 **apple**, **tomato**, **grape**만 따로 출력해 보시오.

fruit = "bananaappletomatoavocadogrape"

indexing_example.py

문자열 슬라이싱 #1

[정답] 아래 문자열을 인덱싱을 이용하여 apple, tomato, grape만 따로 출력해 보시오.

fruit = "bananaappletomatoavocadogrape"

indexing_example.py

```
fruit='bananaappletomatoavocadogrape'  
print(fruit[6:11])  
print(fruit[11:17])  
print(fruit[-5:])
```

함수 #1

함수 만든다는 선언



함수의 이름



```
>>> def greet():
```

```
    print("반갑습니다.")
```

```
    print("파이썬의 세계로 오신 것을 환영합니다.")
```

함수에 속하는 문장들

함수 만들기 끝낼 때 이 위치에서 엔터키 한번 더 입력!!

```
>>>
```

```
def 함수이름():
```

```
    문장
```

함수 #1

- 인자가 있는 함수를 호출하면 매개변수가 인자값 저장

함수 호출

```
>>> greet2( "Everyone" )
```

함수 정의

```
>>> def greet2( name ):  
    print("반갑습니다.", name)  
    print(name, "님은 파이썬의 세계로 오셨습니다.")
```

변수 name에 "Everyone" 전달 및 저장

함수 #1

function_example1.py

- 한번 만들어 놓으면 호출하여 코드를 간단히 만들 수 있다.

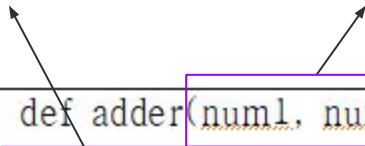
```
>>> def greet():  
    print("반갑습니다.")  
    print("파이썬의 세계로 오신 것을 환영합니다.")
```

```
>>> greet() # 함수 호출  
반갑습니다.  
파이썬의 세계로 오신 것을 환영합니다.  
>>> greet()  
반갑습니다.  
파이썬의 세계로 오신 것을 환영합니다.
```

함수 #2

- 함수의 유형1
- return 있음, 매개변수(argument) 있음

function_example2.py



```
>>> def adder(num1, num2):           #함수 adder의 정의
>>>     return num1 + num2          #num1과 num2의 덧셈 결과를 넘김

>>> print(adder(5, 3))              #adder 함수 호출, 그리고 그 결과 출력
8
```

- return은 함수의 출력, argument는 함수의 입력

함수 #2

- 함수의 유형2
- return 있음, 매개변수(argument)없음

function_example2.py

```
>>> def adder():  
>>>     num1=5  
>>>     num2=7  
>>>     return num1+num2  
  
>>> print(adder())  
12
```

함수 #2

- 함수의 유형3
- return 없음, 매개 변수(argument)없음

function_example2.py

```
>>> def adder():    #함수 adder의 정의
>>>     num1=5
>>>     num2=7
>>>     print(num1+num2)

>>> adder()    #adder 함수 호출
```


함수 #2

- 함수의 유형4
- return 없음, 매개 변수(argument)있음

function_example2.py

```
>>> def adder(num1, num2):      #함수 adder의 정의
>>>     print(num1+num2)

>>> adder(5,3)                  #adder 함수 호출
```

함수 #3

function_example3.py

[문제] 아래와 같이 함수를 호출할 때 5050이 출력되도록 함수를 정의하시오

```
>>> print("0 to 100:", sum_all(0,100))  
5050
```

```
def sum_all(num1, num2):  
    sum=0  
    for i in range(num2+1):  
        sum+=i  
    return sum
```

```
print("0 to 100:", sum_all(0,100))
```

함수 #4

- 튜플과 함수
- 여러개의 값을 리턴받을 수 있다.

function_example4.py

```
>>> def test():  
    return (10,20)  
  
>>> a,b = test()  
>>> print(a,b)  
10 20
```

함수 #4

[문제]

- 함수와 튜플을 이용하여 여러개의 입력과 출력 받기
- 함수를 정의하시오

function_example4.py

```
a = 2  
b = 8  
  
print(sum_and_mul(a, b))
```

함수 #4

[문제]

- 함수와 튜플을 이용하여 여러개의 입력과 출력 받기
- 함수를 정의하시오

```
1  def sum_and_mul(x,y):  
2      ...return x + y, x * y  
3  
4  a = 2  
5  b = 8  
6  
7  print(sum_and_mul(a,b))
```

function_example4.py

함수 #4

-함수를 사용하는 이유

- 반복적인 작업을 할 때, 계산 절차를 미리 정해놓고 활용 가능
 - **ex)** 수학에서의 공식 활용과 비슷
- 그 함수가 어떻게 구현되는지 몰라도 블랙박스로 활용도 가능
 - 입력, 출력만 알면 활용성이 무궁무진
 - **ex)** 스마트폰에서 카메라 버튼을 누르면 카메라 어플리케이션이 켜져서 활용할 수 있지만, 우리는 그 내부가 어떻게 생겼는지 알 필요는 없음
 - 복잡한 연산공식들이 이미 최고의 프로그래머들에 의해 구현되어있고 우리는 레고블럭 맞추듯이 구성만 하면 구현이 되는 경우가 많음

배운것 정리해보기

- 조건문 **if**
 - indent 주의
 - if~elif
- 반복문 **for**
- 인덱싱
- 슬라이싱
- 함수의 **4**가지 유형
- 함수와 튜플

파이썬 크롤러 #1

크롤링이란

- 자동화된 프로그램을 이용하여 웹페이지에 있는 정보를 가져오는 행위를 말합니다
- 네이버 뉴스를 크롤링 한다고 하면, 네이버 뉴스에 있는 텍스트 정보를 내 컴퓨터에 옮겨 텍스트 파일로 만들거나, 엑셀파일 등으로 만들 수 있습니다
- 해외에서는 **Web Crawling**이라는 말도 쓰지만, **Web Scraping**이라는 말을 많이 씁니다
- 그렇다면 크롤링의 대상은?
- 주로 특정한 정보를 보여주며 일정한 형태를 가진 페이지를 많이 크롤링 함
- 뉴스페이지, **SNS**, 인기검색어, 쇼핑몰의 상품페이지 등
- 우리는 데이터 분석을 위해 네이버 뉴스를 크롤링 할 것임

파이썬 크롤러 #1

- 셀레니움

- 웹페이지에 자동으로 접근하게 해줄 툴이다.
- 브라우저가 어떠한 '행동'을 하도록 코딩할 수 있다.
- 원래는 웹페이지의 테스트를 위해 개발된 도구였지만, 현재는 크롤링 목적으로도 많이 이용된다.

파이썬 크롤러 #1

- 셀레니움 설치

- 셀레니움을 사용하기 전에 먼저 셀레니움을 설치해야 한다.
- 윈도우키 + R키를 눌러 실행창에서 **cmd**를 입력하여 명령프롬프트 창을 띄워 아래와 같이 입력해 본다.

```
python -m pip install selenium
```

python : 환경변수 **path**에 등록된 **python**을 실행하기 위한 명령어

-m : **python**의 모듈을 실행하겠다는 의미

pip : **pip**라는 파이썬 레포지토리 관리 모듈을 사용하겠다는 의미

insatll : **pip** 모듈중 **install**을 쓰고 싶다는 의미

selenium : 셀레니움 라이브러리 설치

레포지토리 : 다른 프로그래머들이 구현한 라이브러리를 저장하고 관리하는 저장소

파이썬 크롤러 #1

- 셀레니움 설치

- `python -m pip install selenium`의 의미?

pip라는 파이썬 레포지토리 관리
모듈을 사용

?

레포지토리란? 다른
프로그래머들이 구현한
라이브러리를 저장하고 관리하는
저장소임.
여기에 여러가지 라이브러리를
설치하고 'import' 구문으로 이용
가능

환경변수 path에 등록된 python을
실행하기 위한 명령어

python

-m

pip

install

selenium

셀레니움
라이브러리 설치!

python의 모듈을 실행하겠다는
의미

pip 모듈에는 여러가지 명령어가
있는데 그 중 **install**을 쓰겠다는
의미 만일, 삭제를 하고 싶다면
remove라는 명령어로 가능

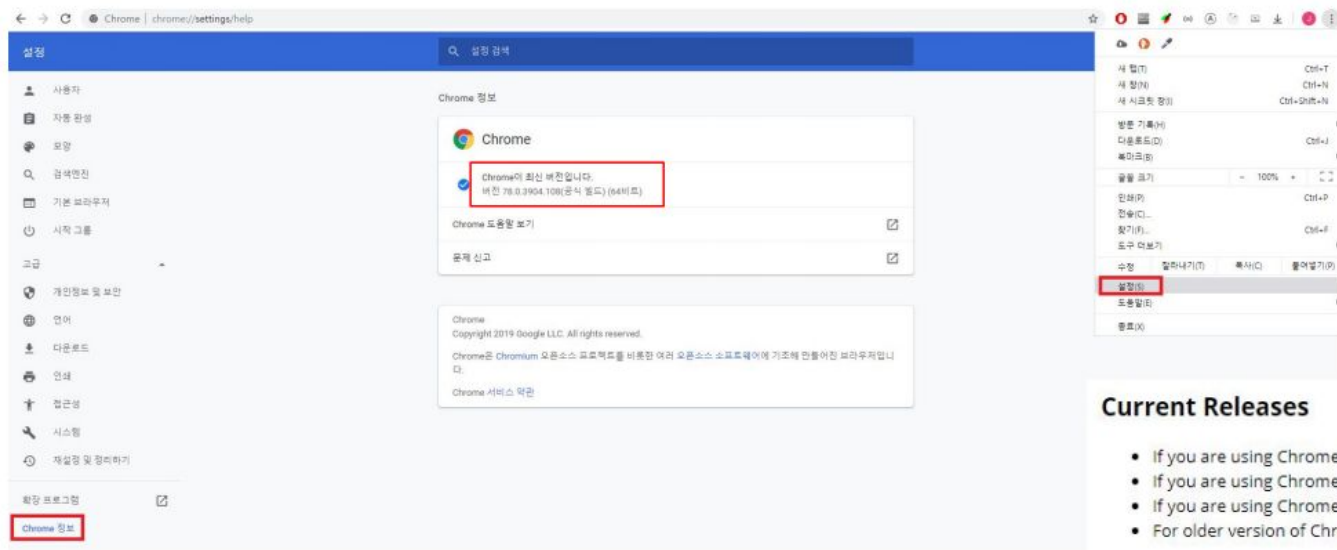
파이썬 크롤러 #1

- 크롬드라이브 설치

- <https://sites.google.com/a/chromium.org/chromedriver/downloads> 의 위치에서

자신의 버전에 알맞은 버전의 크롬 드라이브를 다운받아서 설치

- 압축을 풀고 c:\chromedriver의 폴더에 chromedriver.exe 파일 넣기



The image shows a Chrome browser window with the settings page open. The left sidebar has 'Chrome 정보' (Chrome Info) selected. The main content area shows the 'Chrome 정보' (Chrome Info) section, which includes the Chrome version (78.0.3904.105) and the ChromeDriver version (78.0.3904.105). The version number is highlighted with a red box. Below this, there is a section titled 'Current Releases' which lists the recommended ChromeDriver versions for different Chrome versions. The version for Chrome 78 is highlighted with a red box.

Chrome 정보

Chrome의 최신 버전입니다.
버전 78.0.3904.105(공식 빌드) (64비트)

Chrome 다운로드 보기

문제 신고

Chrome
Copyright 2019 Google LLC. All rights reserved.
Chrome은 Chromium 오픈소스 프로젝트를 비롯한 여러 오픈소스 소프트웨어에 기초해 만들어진 브라우저입니다.
Chrome 서비스 약관

Current Releases

- If you are using Chrome version 79, please download [ChromeDriver 79.0.3945.36](#)
- If you are using Chrome version 78, please download [ChromeDriver 78.0.3904.105](#)
- If you are using Chrome version 77, please download [ChromeDriver 77.0.3865.40](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

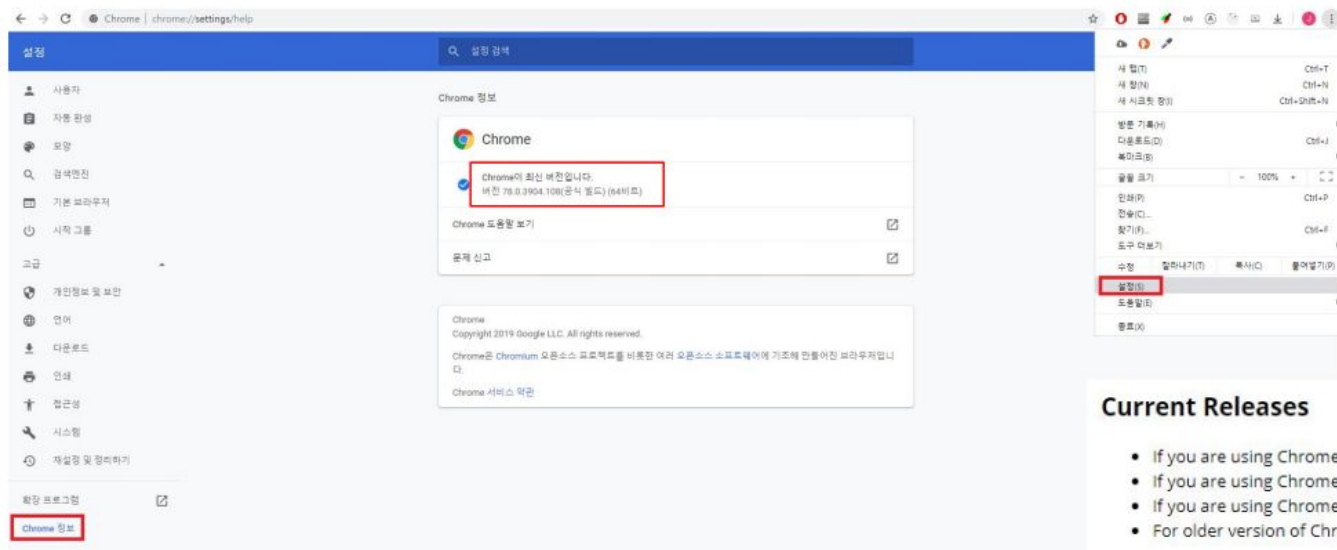
파이썬 크롤러 #1

- 크롬드라이브 설치

- <https://sites.google.com/a/chromium.org/chromedriver/downloads> 의 위치에서

자신의 버전에 알맞은 버전의 크롬 드라이브를 다운받아서 설치

- **참고** c:\chromedriver의 폴더에 chromedriver.exe 파일 넣기



파이썬 크롤러 #1

- 크롬드라이브 설치

Index of /78.0.3904.105/

Name	Last modified	Size	ETag
Parent Directory	-	-	-
chromedriver linux64.zip	2019-11-18 18:20:40	5.27MB	6d3fcd82a27e1d787b6af2515f933806
chromedriver mac64.zip	2019-11-18 18:20:42	7.14MB	12b3858a57bfe0a3d450cd194e0992aa
chromedriver win32.zip	2019-11-18 18:20:43	4.62MB	8f3f3ca4134c14b41848686f0146b200
notes.txt	2019-11-18 18:20:47	0.00MB	009b4b01151a0417c58075380e08254b

윈도우 환경에서 수행할
것이기 때문에, **win32**버전
다운로드!

> 내 PC > 로컬 디스크 (C:) > chromedriver

이름	수정된 날짜	유형	크기
chromedriver.exe	2019-11-21 오전...	응용 프로그램	8,629KB

파이썬 크롤러 #1

- 크롬드라이브 설치

Index of /78.0.3904.105/

Name	Last modified	Size	ETag
Parent Directory	-	-	-
chromedriver linux64.zip	2019-11-18 18:20:40	5.27MB	6d3fcd82a27e1d787b6af2515f933806
chromedriver mac64.zip	2019-11-18 18:20:42	7.14MB	12b3858a57bfe0a3d450cd194e0992aa
chromedriver win32.zip	2019-11-18 18:20:43	4.62MB	8f3f3ca4134c14b41848686f0146b200
notes.txt	2019-11-18 18:20:47	0.00MB	009b4b01151a0417c58075380e08254b

윈도우 환경에서 수행할
것이기 때문에, **win32**버전
다운로드!

> 내 PC > 로컬 디스크 (C:) > chromedriver

이름	수정한 날짜	유형	크기
chromedriver.exe	2019-11-21 오전...	응용 프로그램	8,629KB

파이썬 크롤러 #1

- 예제1번 일단 실행해 보기

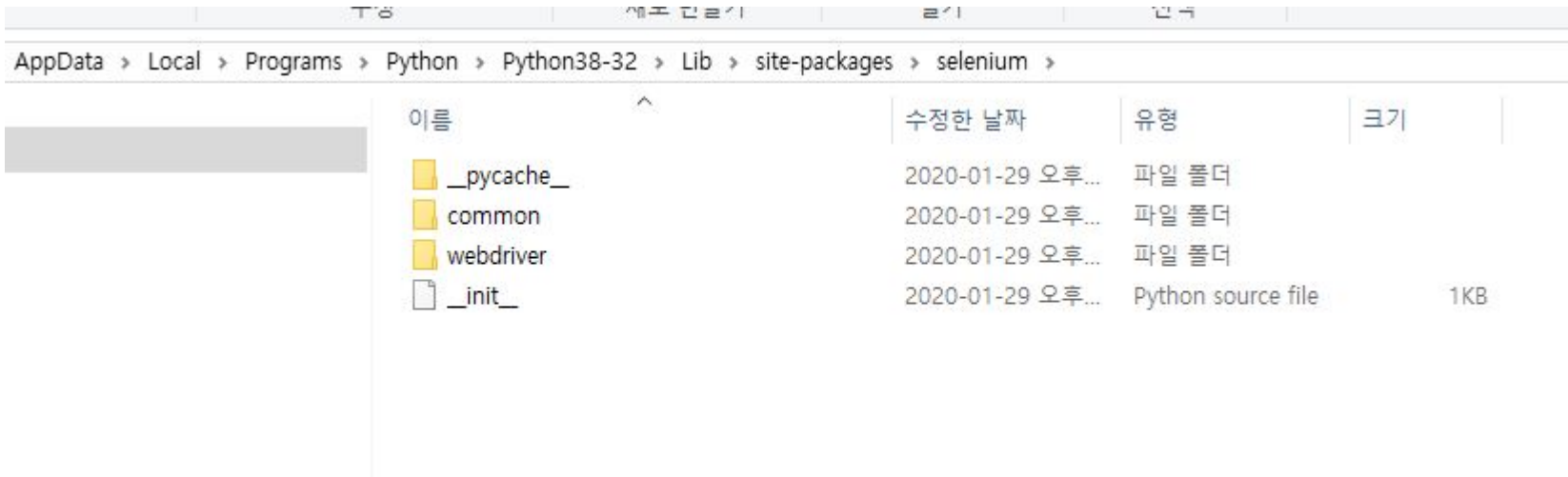
selenium_example1.py

예제1

```
from selenium import webdriver
import time
path = "c:/chromedriver/chromedriver.exe"
driver = webdriver.Chrome(path)
time.sleep(10)
```


파이썬 크롤러 #1

- 예제1번 일단 실행해 보기



파이썬 크롤러 #1

- 예제1번 일단 실행해 보기

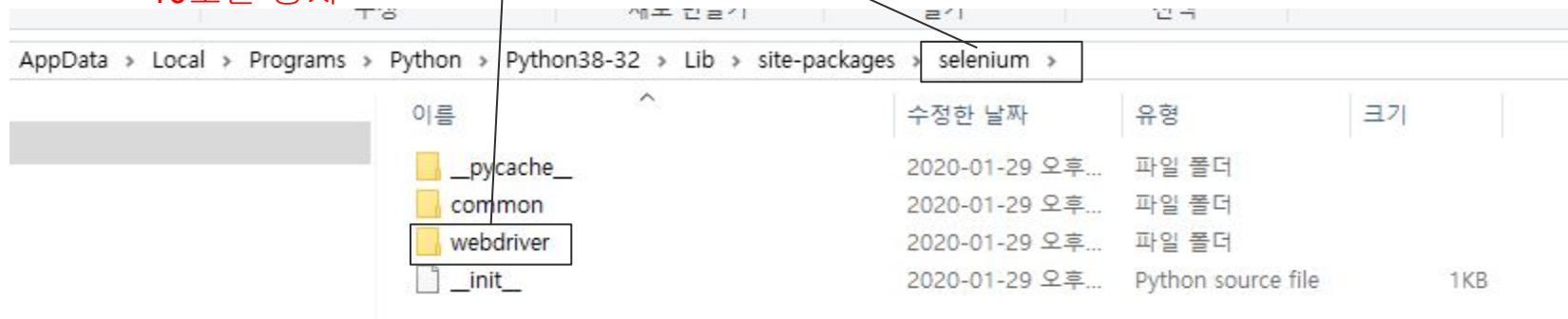
예제1

```
from selenium import webdriver
import time
path = "c:/chromedriver/chromedriver.exe"
driver = webdriver.Chrome(path)
time.sleep(10)
```

셀레니움 폴더 속의 **webdriver**를 가져오겠다

webdriver속의 **Chrome()**함수를 실행시키겠다

10초간 정지



파이썬 크롤러 #2

selenium_example1.py

- naver제어하기

예제2

```
from selenium import webdriver
import time
path = "c:/chromedriver/chromedriver.exe"
driver = webdriver.Chrome(path)
```

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

chromedriver이용하여 get()함수로 url접근

```
search_input_box = driver.find_element_by_id("query")
```

```
search_input_box.send_keys("손흥민 골영상")
```

```
search_input_box.submit()
```

```
time.sleep(10)
```

driver.find_element_by_id()함수로 query이름을
search_input_box에 저장
.send_keys()함수로 키워드 입력
.submit()함수로 Enter 눌러라

파이썬 크롤러 #2

- naver제어하기

- 엘리먼트(elements) F12누르면 나옴

-엘리먼트란 웹에서 우리 눈에 보이는 모든 것

! 테스트 소프트웨어에 의해 제어되고 있습니다.



통합검색 동영상 블로그 포스트 카페 웹사이트 이미지 지식IN 더보기 검색옵션

손흥민 골 주요장면



'26 연속 득점' 답답했던 경기를 뚫아주는 손흥민의 선...
해외축구 | 3일 전
네이버 스포츠
▷ 1,662,273



드디어 터졌다! 모두가 기다린 우리들의 새해 첫 골!
해외축구 | 6일 전
네이버 스포츠
▷ 1,411,460



'마라도나 인 줄!' 리그 5호골 기록하는 슈퍼 소닉 손흥민
해외축구 | 2019.12.08.
네이버 스포츠
▷ 7,352,426

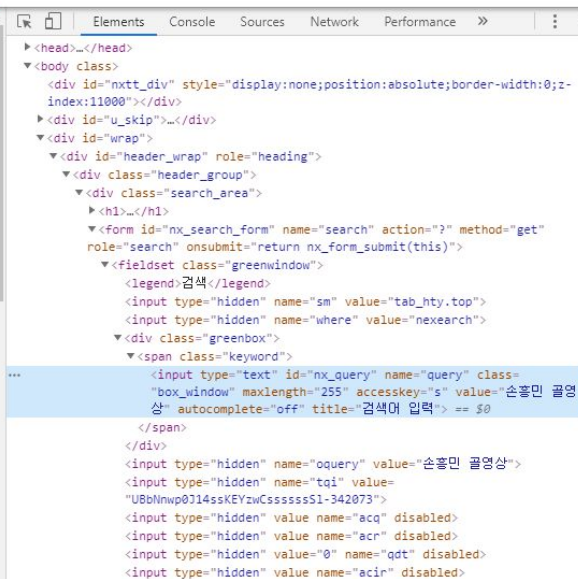
< 1 / 5 >

더보기 >

뉴스토픽

뉴스	연예 스포츠
1 조국과 공범	NEW
2 차기 우리은행장 후보	NEW
3 우한 교민 수용	
4 아산 진천에 수용	
5 선제적 대응	NEW
6 엄마부대 주옥순	NEW
7 대구경북 통합신공항	NEW
8 예방에 총력	NEW
9 바른미래당 탈당	
10 홍남기 신종코로나 확산	NEW

2020.01.29. 17:30 ~ 20:30 기준



파이썬 크롤러 #3

- naver뉴스 수집하기

selenium_example2.py

- 제목은 `driver.find_element_by_id("articleTitle")`로 수집한다.
- 본문은 `driver.find_element_by_id("articleBodyContents")`로 수집한다.

예제3

```
from selenium import webdriver
import time

path = "c:/chromedriver/chromedriver.exe"
driver = webdriver.Chrome(path)

driver.get(
    "https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=102&oid=018&aid=0004533705"
)

article_title_element = driver.find_element_by_id("articleTitle")
article_body_element = driver.find_element_by_id("articleBodyContents")

print(article_title_element.text)
print(article_body_element.text)
```

파이썬 크롤러 #3

- naver뉴스 수집하기
- 날짜 수집하기
- driver.find_elements_by_class_name()

예제4

```
from selenium import webdriver
import time
```

selenium_example2.py

```
path = "c:/chromedriver/chromedriver.exe"
```

```
driver = webdriver.Chrome(path)
```

```
driver.get(
```

```
"https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=105&oid=029
&aid=0002573025"
```

```
)
```

```
article_title_element = driver.find_element_by_id("articleTitle")
```

```
article_body_element = driver.find_element_by_id("articleBodyContents")
```

```
article_date_elements = driver.find_elements_by_class_name("t11")
```

```
print(article_title_element.text)
```

```
print(article_body_element.text)
```

```
(article_input_date_element, article_last_modified_date_element) =
article_date_elements
```

```
print(article_input_date_element.text)
```

```
print(article_last_modified_date_element.text)
```

```
driver.close()
```

파이썬 크롤러 #4

selenium_example2.py

- naver뉴스 수집하기
- csv파일 저장하기
- 예제4의 driver.close()위에 쓸것

```
py -m pip install pandas
```

실행>cmd
판다스설치

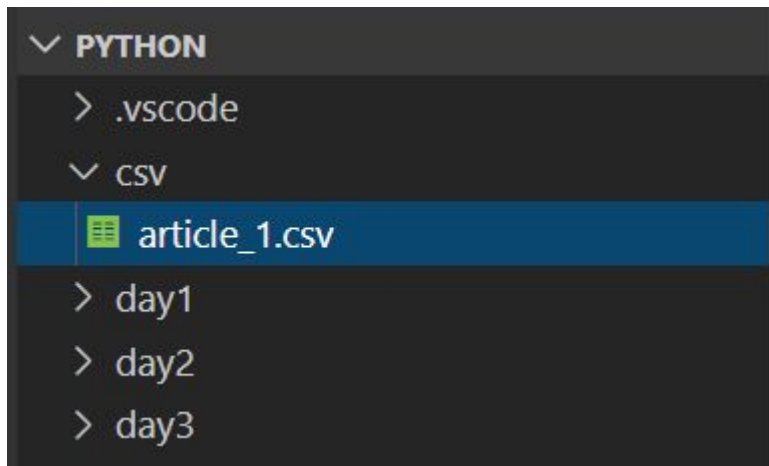
예제5

```
import pandas as pd  
dataframe = pd.DataFrame(  
    {  
        "title": [article_title_element.text],  
        "body": [article_body_element.text],  
        "input_date": [article_input_date_element.text],  
        "last_modified_date": [article_last_modified_date_element.text]  
    }  
)  
csv_save_path = "./csv/article_1.csv"  
dataframe.to_csv(csv_save_path, mode="w", index=False)
```

파이썬 크롤러 #4

- naver뉴스 수집하기
- csv파일 저장하기

selenium_example2.py



이름	수정한 날짜	유형	크기
.vscode	2020-01-29 오후...	파일 폴더	
csv	2020-01-30 오전...	파일 폴더	
day1	2020-01-29 오후...	파일 폴더	
day2	2020-01-30 오전...	파일 폴더	
day3	2020-01-30 오전...	파일 폴더	
debug	2020-01-30 오전...	텍스트 문서	

파이썬 크롤러 #4

- naver뉴스 수집하기

Pandas란?

- 행과 열로 이루어진 데이터(엑셀처럼 표 형식으로 된 데이터)를 다루기에 특화된 라이브러리
- Columns, Rows, Data라는 3가지 핵심 개념이 있음
- 엑셀을 생각하면 쉬움

Python | Pandas DataFrame

Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the **data**, **rows**, and **columns**.

The diagram illustrates a Pandas DataFrame as a table with 6 rows and 5 columns. The columns are labeled 'Name', 'Team', 'Number', 'Position', and 'Age'. The rows are indexed from 0 to 6. A blue arrow labeled 'Columns' points to the column headers. A brown arrow labeled 'Rows' points to the row indices. A purple arrow labeled 'Data' points to the data cells. The data is as follows:

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

파이썬 크롤러 #4

- naver뉴스 수집하기

to_csv 자세히 살펴보기

- csv란? Comma Separated Value의 약자
 - 실제 데이터를 확인해보면 ‘,(comma)’로 나누어진 것을 확인할 수 있음
 - to_csv는 파이썬에서 우리가 갖고 있는 데이터를 csv 형태의 데이터로 컴퓨터에 저장하겠다는 의미
- mode=“w”의 의미는? write(쓰기)모드로 진행하겠다는 뜻
- index=False의 의미는? 데이터 순서 표기를 하지 않겠다는 뜻
 - index=True로 하면 데이터의 순서대로 0 1 2 3 4의 숫자가 붙음

파이썬 크롤러 #4

- naver뉴스 수집하기

to_csv 자세히 살펴보기

- csv란? Comma Separated Value의 약자
 - 실제 데이터를 확인해보면 ‘,(comma)’로 나누어진 것을 확인할 수 있음
 - to_csv는 파이썬에서 우리가 갖고 있는 데이터를 csv 형태의 데이터로 컴퓨터에 저장하겠다는 의미
- mode=“w”의 의미는? write(쓰기)모드로 진행하겠다는 뜻
- index=False의 의미는? 데이터 순서 표기를 하지 않겠다는 뜻
 - index=True로 하면 데이터의 순서대로 0 1 2 3 4의 숫자가 붙음

파이썬 크롤러 #5

- naver뉴스 수집하기

저장결과 확인하기

예제6

```
import pandas as pd
csv_read_path = "../csv/article_1.csv"
dataframe = pd.read_csv(csv_read_path)

print("title")
print(dataframe["title"])
print("body")
print(dataframe["body"])
print("input_date")
print(dataframe["input_date"])
print("last_modified_date")
print(dataframe["last_modified_date"])
```