**SW사관학교 정글 - Chapter 3**

**[수업 목표]**

1. 파이썬 기초 문법을 안다.
2. 파이썬으로 알고리즘 문제를 풀 수 있다.

**01. Python**

파이썬이란

* 📖 네덜란드의 프로그래머 Guido van Rossum이 '읽기 쉬운 코드'에 중점을 두어 개발한 프로그래밍 언어입니다.
  + '읽기 쉽게' 만든 언어이기 때문에, 문법이 쉬워 초보자가 배우기 좋은 언어입니다.
  + 무료/오픈소스로 운영되고 커뮤니티가 크기 때문에 자연어처리, 머신러닝 등 직접 구현하기 어려운 기능들도 다른 개발자의 라이브러리를 이용해 쉽게 코드를 작성할 수 있습니다.

파이썬 설치

* 👉 **파이썬을 설치한다?** 컴퓨터는 001010100101과 같은 신호만 알아들을 수 있으므로 사람이 쓰는 파이썬 문법으로 된 글을 0과 1로 된 신호로 변환해주어야 합니다. 이 '번역 패키지'를 설치하는 과정이에요.
* 윈도우
  + cmd혹은 PowerShell 창에서 python3를 입력해 보세요.
  + Python이 설치되어 있지 않다면 Microsoft Store에서 무료 구매 및 설치할 수 있습니다.
  + 이미 python이 설치되어 있다면 다음과 같은 화면이 나올 것입니다.

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

* 위의 방식 이외에 아래와 같이 직접 python.org에서 download받아 설치할 수도 있습니다.
  + **1.버튼을 눌러 다운로드**해주세요. (3.12.x 이상의 최신 버전을 설치!) [**설치페이지 바로가기>>**](https://www.python.org/downloads/)

텍스트, 스크린샷, 운영 체제, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* + 2.다운로드된 파이썬 파일을 열어 **설치**합니다.
    - 🚨 **주의) Add python.ext to PATH 에 체크!**텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

      자동 생성된 설명텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

      자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

      자동 생성된 설명
* 맥
  + 🔥 Mac에는 기본적으로 파이썬 2.x 버전이 설치되어 있지만, 버전3으로 업데이트되면서 문법과 사용 가능한 라이브러리 등에 많은 차이가 있습니다. 자료를 따라오기 위해서는 3.x 버전의 파이썬이 필요하니 아래 가이드에 따라 설치해주세요.
  + brew를 설치했다면 터미널에서 다음 명령으로 쉽게 python 3.9를 설치할 수 있습니다.
    - brew install [python@3.9](mailto:python@3.9)
  + 또는, 아래와 같이 python.org에서 직접 다운로드하여 설치할 수 있습니다.
    - **버튼을 눌러 다운로드** (3.12.x 이상의 최신 버전을 설치!) [**설치페이지 바로가기>>**](https://www.python.org/downloads/)텍스트, 스크린샷, 운영 체제, 소프트웨어이(가) 표시된 사진

      자동 생성된 설명
    - 다운로드된 파이썬 파일을 열어 **설치**합니다. 아래와 같은 폴더가 나타나면 설치가 완료된 것입니다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

VSCode로 파이썬 파일 실행하기

* 📖 VSCode (Visual Studio Code)는 최근 인기를 얻은 IDE(Integrated development environment)로, 각종 프로그램의 작성, 수정, 실행을 도와주는 MS 워드 같은 프로그램입니다.
* 1.우선 jungle 폴더 안에 파이썬 코드를 넣을 prac\_python 폴더를 만든다.
* 2.VSCode를 실행한다.
  + VSCode icon을 더블클릭해서 실행한 경우에는 File > Open Folder... 메뉴를 클릭해서 위에서 만든 폴더를 지정한다.
  + Terminal에서 실행하고자 할 경우에는 cd prac\_python 한 후에 code .를 실행하면 현재 directory를 기준으로 VSCode가 실행된다.
* 3.왼쪽에서 폴더 우클릭 > New File을 선택하여 hello.py라는 이름으로 파일을 만든다.(여기서 Python extension을 설치할 것이냐고 물어보면 설치하는 것이 좋습니다.)
* 4.다음 내용을 파일에 붙여 넣는다.
  + print('Hello, jungle')
  + 붙여 넣은 후 ctrl+S를 눌러 파일을 저장하는 것을 잊지 않는다.
  + (File > Auto Save 가 check 해두면 매번 저장하지 않아도 자동 저장이 됩니다.)
* 5.파일 창 오른쪽 위의 play button(▶️)을 클릭 > Run code 를 선택해 실행한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 6.결과 창에 Hello, jungle이 출력되는 것을 확인한다.

파이썬 기초 문법

* 콘솔창을 열고 파이썬을 연습해봅시다.
* 변수 선언
  + 자바스크립트와 다르게 새 변수를 만들 때 let을 쓰지 않습니다. 출력할 때는 print()를 씁니다.
    - a = 3 # 3을 a에 넣는다
    - print(a)
    - b = a # b에 3을 넣는다.
    - print(b)
    - a = 5 # a에 5라는 새로운 값을 넣는다.
    - print(a, b) # 5 3
  + 콘솔창에서는 print() 함수를 쓰지 않고 변수 이름만 적고 실행해도 저장된 값을 볼 수 있습니다.
    - a = 3
    - a # 3이 출력된다.

자료형과 기본 연산

* 숫자형
  + 자바스크립트와 마찬가지로 숫자를 값으로 가질 수 있으며, 정수, 실수 모두 가능합니다.
    - a = 4.8
  + 숫자 간에는 사칙연산이 가능합니다. 몫과 나머지도 구할 수 있어요.
    - a = 7
    - b = 2
    - a+b # 9
    - a-b # 5
    - a\*b # 14
    - a/b # 3.5
    - a//b # 3 (몫)
    - a%b # 1 (나머지)
    - a\*\*b # 49 (거듭제곱)
    - a+3\*b # 13 (여러 연산을 한 줄에 할 경우 사칙연산의 순서를 따른다.)
  + 변수에 저장된 값에 연산을 한 후 다시 그 값을 같은 변수에 저장할 수 있습니다. 자바스크립트와 다르게 증감연산자(i++, i-- 등)는 없습니다.
    - a = 5
    - a = a + 3 # 5에 3을 더한 값을 다시 a에 저장
    - print(a) # 8
    - a += 1 # 줄여 쓸 수도 있다. 8에 1을 더해 9를 a에 저장. 사칙연산 다 가능하다.

문자열

* 글씨를 따옴표로 감싸 문자열 자료형으로 정의할 수 있습니다. 작은따옴표('')와 큰따옴표("") 모두 사용할 수 있고 짝이 맞아야 합니다. 그렇기 때문에 문자열에 따옴표가 포함될 경우, 다른 종류의 따옴표를 써서 문자열을 정의해 줄 수 있습니다.
  + word1 = "jungle"
  + word2 = 'coding'
  + mac = "McDonald's"
  + sentence = 'He said, "Hello!"'
* 따옴표로 감싸지 않으면 변수명으로 인식합니다.
  + a = 1
  + b = "a"
  + c = a
  + print(a, b, c) # 1 "a" 1
  + d = x # x라는 변수가 정의되지 않았기 때문에 에러메시지가 나온다.
  + d = "x"
* 문자열 간의 더하기는 두 문자열을 이어붙인 문자열을 반환합니다.
  + first\_name = "Harry"
  + last\_name = "Potter"
  + first\_name + last\_name # HarryPotter
  + first\_name + " " + last\_name # Harry Potter
  + a = "3"
  + b = "5"
  + a + b # 35 문자열 간의 덧셈이기 때문에 앞뒤로 이어붙인 문자열이 반환된다.
  + a + 5 # 파이썬에서는 문자열과 숫자형은 더할 수 없다
  + a \* 5 # 33333 a\*5는 a+a+a+a+a과 같으므로 "3"이 5번 반복되는 문자열이 반환된다.
* 파이썬으로도 미리 정해져있는 메소드를 이용하여 다양한 연산이 가능합니다.
  + 모든 알파벳을 대문자로 바꾸기
    - myname = 'jungle'
    - myname.upper() # JUNGLE
  + 특정 문자를 기준으로 문자열을 나누고 싶은 경우
    - # 이메일 주소에서 도메인 'gmail'만 추출하기
    - myemail = 'test@gmail.com'
    - result = myemail.split('@') # ['test','gmail.com'] (뒤에 배울 '리스트'라는 자료형이다)
    - result[0] # test (리스트의 첫번째 요소)
    - result[1] # gmail.com (리스트의 두 번째 요소
    - result2 = result[1].split('.') # ['gmail','com']
    - result2[0] # gmail -> 우리가 알고 싶었던 것
    - result2[1] # com
    - # 한 줄로 쓸 수도 있다.
    - myemail.split('@')[1].split('.')[0]
  + 특정 문자를 다른 문자로 바꾸기
    - txt = '서울시-마포구-망원동'
    - result = txt.replace('-', '>') # '서울시>마포구>망원동'

불 자료형

* 파이썬에도 참, 거짓을 나타내는 불 자료형이 있는데 자바스크립트와 다르게 첫 글자가 대문자여야 합니다 ㅎㅎ
  + x = True # 참
  + y = False # 거짓
  + # 소문자로 쓰면 자료형으로 인식하지 않고 변수명이라 생각해 에러가 난다.
  + z = true # name 'true' is not defined
  + # 비교연산자
  + 4 > 2 # True 크다
  + 5 < 1 # False 작다
  + 6 >= 5 # True 크거나 같다
  + 4 <= 4 # True 작거나 같다
  + 3 == 5 # False 같다
  + 4 != 7 # True 같지 않다
* 불 자료형에는 논리연산자를 이용할 수 있습니다.
  + a = 4 > 2 # True
  + not a # False NOT 연산자로 참을 거짓으로, 거짓을 참으로 바꿔준다.
  + b = 2!=2 # False
  + a and b # False AND 연산자로 모두 참이어야 참을 반환한다.
  + a or b # True OR 연산자로 둘 중 하나만 참이면 참이다.

리스트와 딕셔너리

* 파이썬의 리스트와 딕셔너리는 자바스크립트와 비슷합니다.
  + a\_list = []
  + a\_list.append(1) # 리스트에 값을 넣는다
  + a\_list.append([2,3]) # 리스트에 [2,3]이라는 리스트를 다시 넣는다
  + a\_list # [1,[2,3]]
  + len(a\_list) # 2 리스트의 길이
  + a\_list[0] # 1
  + a\_list[1] # [2,3]
  + a\_list[1][0] # 2
  + 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
  + 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'
* 리스트 간의 덧셈과 자연수 곱셈이 정의되어있습니다.
  + a = [3, 3, 1]
  + b = [5, 2]
  + a + b # [3, 3, 1, 5, 2]
  + a \* 2 # [3, 3, 1, 3, 3, 1]

함수

* 함수의 정의
  + # 수학문제에서
  + f(x) = 2\*x+3
  + y = f(2)
  + y의 값은? 7
  + # 참고: 자바스크립트에서는
  + function f(x) {

return 2\*x+3

}

* + # 파이썬에서
  + def f(x):

return 2\*x+3 # 중괄호 대신에 들여쓰기로 각 블록의 범위를 표시한다.

* + f(2) # 7
* 함수의 응용
  + def sum\_all(a,b,c):

return a+b+c

* + def mul(a,b):

return a\*b

* + result = sum\_all(1,2,3) + mul(10,10)
  + # result라는 변수의 값은?

조건문

* if / else 로 구성!
  + def oddeven(num): # oddeven이라는 이름의 함수를 정의한다. num을 변수로 받는다.

if num % 2 == 0: # num을 2로 나눈 나머지가 0이면

return True # True (참)을 반환한다.

else: # 아니면,

return False # False (거짓)을 반환한다.

* + result = oddeven(20) # True
  + def is\_adult(age):

if age > 20:

print('성인입니다') # 조건이 참이면 성인입니다를 출력

else:

print('청소년이에요') # 조건이 거짓이면 청소년이에요를 출력

* + is\_adult(30) # 성인입니다
* else if는 elif로 줄여씁니다.
  + def is\_adult(age):

if age > 20:

print('성인입니다')

elif age >= 13:

print('청소년이에요')

else:

print('어린이네요!')

* + is\_adult(30) # 성인입니다

반복문

* 👉 파이썬에서의 반복문은, 리스트의 요소들을 하나씩 꺼내쓰는 형태입니다.
* 예를 들어, 아래와 같이 리스트의 각 요소에 대해서 다음과 같이 반복문을 쓸 수 있습니다.
  + fruits = ['사과','배','감','귤']
  + for fruit in fruits:

print(fruit)

* + # 사과
  + # 배
  + # 감
  + # 귤
* 이것을 자바스크립트 식으로 쓰면 아래처럼 나타낼 수 있어요.
  + fruits = ['사과','배','감','귤']
  + for i in range(len(fruits)): # i 가 0, 1, 2, 3일 때

fruit = fruits[i]

print(fruit)

* + # 사과
  + # 배
  + # 감
  + # 귤
* 💡 위에서 볼 수 있다시피 우리가 이미 배운 다른 프로그래밍 언어인 자바스크립트와 생김새는 다소 다르지만 변수를 선언하고 함수를 쓰고 조건문/반복문을 이용하는 등 각각의 동작 자체는 비슷합니다.

**02. 파이썬 응용문제**

파이썬도 문제를 풀면서 문법을 익혀봅시다.

✍️ Q1. 과일 갯수 세기 함수 만들기

* 아래와 같이 과일 이름이 들어있는 리스트가 있을 때, 특정 과일이 몇 개 들어있는지 세는 함수를 만들어볼게요.
  + fruits = ['사과','배','배','감','수박','귤','딸기','사과','배','수박']
* 1.사과의 갯수를 센다면 어떻게 하면 좋을지 슈도코드로 적는다.
  + 과일 리스트가 있을 때,
  + 결과를 저장할 변수를 만들고 초깃값은 0으로 한다.
  + 리스트 안에 있는 각 과일에 대해서
  + 이름이 사과이면
  + 결과를 1 증가시킨다.
  + 결과 출력
* 2.코드로 나타낸다.
  + 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.
* 3.사과가 아닌 다른 과일에 대해서도 쓸 수 있도록 함수로 만든다.
  + def count\_fruits(target):
  + count = 0
  + for fruit in fruits:

if fruit == target:

count += 1

* + return count
  + subak\_count = count\_fruits('수박')
  + print(subak\_count) #수박의 개수
  + gam\_count = count\_fruits('감')
  + print(gam\_count) #감의 갯수

✍️ Q2. 사람의 나이 출력하기

* 아래와 같이 각 사람의 이름과 나이가 든 딕셔너리가 각각의 요소인 리스트가 있을 때, 이름을 넣으면 나이를 돌려주는 함수를 만들어봅시다.

people = [{'name': 'bob', 'age': 20},

{'name': 'carry', 'age': 38},

{'name': 'john', 'age': 7},

{'name': 'smith', 'age': 17},

{'name': 'ben', 'age': 27}]

* 1.우선 데이터를 살펴본다. 각 사람의 이름과 나이를 출력해본다.
  + people = [{'name': 'bob', 'age': 20},

{'name': 'carry', 'age': 38},

{'name': 'john', 'age': 7},

{'name': 'smith', 'age': 17},

{'name': 'ben', 'age': 27}]

* + for person in people:

print(person['name'], person['age'])

* 2.'bob'의 나이를 출력하는 알고리즘을 슈도코드로 나타낸다.
  + 사람 정보가 담긴 리스트가 있을 때,
  + 리스트 안의 딕셔너리인 각 사람에 대해서
  + 만약 사람의 이름이 'bob'이면
  + 그 사람의 나이를 출력해라.
* 3.실제 코드로 작성한다.
  + people = [{'name': 'bob', 'age': 20},

{'name': 'carry', 'age': 38},

{'name': 'john', 'age': 7},

{'name': 'smith', 'age': 17},

{'name': 'ben', 'age': 27}]

* + for person in people:

if person["name"] == "bob":

print(person['age'])

* 4.다른 사람의 이름도 넣을 수 있도록 함수로 만든다.
  + people = [{'name': 'bob', 'age': 20},

{'name': 'carry', 'age': 38},

{'name': 'john', 'age': 7},

{'name': 'smith', 'age': 17},

{'name': 'ben', 'age': 27}]

* + def get\_age(myname):

for person in people:

if person['name'] == myname:

return person['age']

return '해당하는 이름이 없습니다'

* + print(get\_age('bob'))
  + print(get\_age('kay'))

**03. 파이썬 패키지**

파이썬 패키지(package) 설치하기

* 💡 패키지는 모듈(일종의 기능들 묶음)을 모아 놓은 단위이고, 이런 패키지의 묶음을 라이브러리라고 합니다. 하지만 일반적으로 패키지와 라이브러리는 두 용어를 혼용해서 많이 써요.
* 1.가상 환경(virtual environment) 이란?
  + 📖 같은 시스템에서 실행되는 다른 파이썬 응용 프로그램들의 동작에 영향을 주지 않기 위해, 파이썬 배포 패키지들을 설치하거나 업그레이드하는 것을 가능하게 하는 **격리된 실행 환경**
  + 출처 : [파이썬 공식 용어집- 가상환경](https://docs.python.org/ko/3/glossary.html#term-virtual-environment)
  + 한 프로젝트에서 쓰는 패키지의 수는 다양하고, 서로 다른 프로젝트에서 요구하는 패키지의 목록이나 버전이 다를 수 있습니다.
  + 이렇게 요구사항이 다른 두 프로젝트를 한 컴퓨터에서 진행해야하는 상황에서 각 프로젝트마다 가상환경을 만들고 그 안에 필요한 패키지를 설치해주면 필요한 버전이 다르거나 두 패키지가 충돌을 일으키는 등의 문제 없이 관리를 할 수 있습니다.
* 2.가상 환경 설치 및 확인
  + 👉 새 프로젝트에서 설치하는 패키지가 많다면 가상환경을 새로 잡아주는 게 좋습니다. 자세한 내용은 아래 참고문헌을 참조하세요
  + 참고 문헌
    - [파이썬 표준 라이브러리 - venv](https://docs.python.org/ko/3/library/venv.html#creating-virtual-environments)
    - [VSCode - Python Tutorial - Install and use packages](https://code.visualstudio.com/docs/python/python-tutorial#_install-and-use-packages)
  + 프로젝트 루트 디렉토리에서 다음 명령을 실행하여 새 가상환경을 만듭니다.
    - python3 -m venv .venv
    - # python3 명령이 통하지 않는 경우 (MS Windows에서 이런 경우 있음)
    - py -3 -m venv .venv
  + Terminal 에서 다음과 같이 가상환경을 활성화 시켜 봅니다.
    - # Shell 환경 (MacOS/Linux/Windows bash)
    - source .venv/bin/activate
    - # PowerShell (Windows 1x)
    - .venv\\Scripts\\activate
  + 가상환경이 활성화되면 아래와 같이 프롬프트의 앞부분이 가상환경 directory의 이름이 됩니다.
    - (.venv) PS C:\\Users\\myname\\Desktop\\Jungle\\prac\_python>
  + Windows의 경우 보안 문제로 활성화가 실패할 수 있습니다. 이 경우 PowerShell 환경에서 다음 명령을 실행하여 서명된 외부 코드의 실행을 허용하여 문제를 해결할 수 있습니다.
    - Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
* 3.pip(python install package) 사용 - requests 패키지 설치해보기
  + 👉 앱을 설치할 때 앱스토어/플레이스토어를 가듯이, 새로운 프로젝트의 라이브러리를 가상환경(공구함)에 설치하려면 pip 를 이용하게 됩니다.
  + pip list명령으로 현재 설치된 package들의 목록을 볼 수 있습니다.
  + pip install <package\_name>명령으로 원하는 패키지를 설치할 수 있습니다.
  + 다음 명령을 실행하여 requests 패키지를 설치합시다.
  + pip install requests

패키지 사용하기

* 우리가 자바스크립트에서 ajax를 써서 GET요청을 보냈던 것처럼, 파이썬에서는 아래 코드처럼 requests 패키지를 이용하여 서울시 대기 OpenAPI에서 미세먼지 정보를 받아올 수 있습니다.
  + ⚠️ 들여쓰기 주의!
  + import requests # requests 라이브러리 설치 필요
  + r = requests.get('<http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99>')
  + rjson = r.json()
  + print(rjson)
  + print(rjson['RealtimeCityAir']['row'][0]['NO2']) # 중구의 NO2 값
* 모든 구의 IDEX\_MVL 값 프린트하기
  + gus = rjson['RealtimeCityAir']['row']
  + for gu in gus:

print(gu['MSRSTE\_NM'], gu['IDEX\_MVL'])

* IDEX\_MVL 값이 60 미만인 구의 이름과 값 프린트하기
  + import requests # requests 라이브러리 설치 필요
  + r = requests.get('<http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99>')
  + rjson = r.json()
  + gus = rjson['RealtimeCityAir']['row']
  + for gu in gus:

if gu['IDEX\_MVL'] < 60:

print (gu['MSRSTE\_NM'], gu['IDEX\_MVL'])

**04. 웹스크래핑**

웹스크래핑이란?

* 📖 웹 스크래핑(web scraping)은 **웹 페이지에서 우리가 원하는 부분의 데이터를 수집해오는 것을 뜻합니다**.
* 한국에서는 같은 작업을 *크롤링 crawling* 이라는 용어로 혼용해서 쓰는 경우가 많습니다. 원래는 크롤링은 자동화하여 주기적으로 웹 상에서 페이지들을 돌아다니며 분류/색인하고 업데이트된 부분을 찾는 등의 일을 하는 것을 뜻해요.
* 구글 검색을 할 때는 **web scraping** 으로 검색해야 우리가 배우는 페이지 추출에 대한 결과가 나올 거예요!
* 참고
  + [Web Scraping](https://en.wikipedia.org/wiki/Web_scraping)(wikipedia) / [Web Crawler](https://en.wikipedia.org/wiki/Web_crawler)(wikipedia) [Web Scraping vs Web Crawling: What’s the Difference?](https://dzone.com/articles/web-scraping-vs-web-crawling-whats-the-difference)

✍️ 웹스크래핑 해보기

* YES 24 베스트셀러 페이지에서 책 제목들을 스크래핑해보겠습니다.
  + **링크:** [**YES 24 베스트셀러**](https://www.yes24.com/Product/Category/BestSeller?categoryNumber=001&pageNumber=1&pageSize=24)
  + [예스24](https://www.yes24.com/Product/Category/BestSeller?categoryNumber=001&pageNumber=1&pageSize=24)
* 1.크롬 브라우저에 페이지를 띄우고 개발자도구를 열어 HTML 구조를 파악한다.
  + 각 책들이 yesBestList클래스 내 li 태그로 들어있네요.

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

* + 각 책 정보에서 제목은 a 태그 안에 들어있네요!

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

* + 💡 개발자도구 Elements 탭에서 각 구역을 접어가며 점점 상위 요소를 파악해봅니다.
* 2.HTML 구조를 파악하는 데 도움을 주는 bs4 패키지를 설치한다.
  + (beautifulsoup은 옛날 버전의 패키지로 호환이 안되므로 꼭 bs4 를 받아야 합니다.)
* 3.새 파이썬 파일을 만들어 아래 코드를 붙여넣는다.
  + import requests
  + from bs4 import BeautifulSoup
  + # 타겟 URL을 읽어서 HTML를 받아오고,
  + headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
  + url = '<https://www.yes24.com/Product/Category/BestSeller?categoryNumber=001&pageNumber=1&pageSize=50>'
  + data = requests.get(url,headers=headers)
  + # HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦
  + # soup이라는 변수에 "파싱 용이해진 html"이 담긴 상태가 됨
  + # 이제 코딩을 통해 필요한 부분을 추출하면 된다.
  + soup = BeautifulSoup(data.text, 'html.parser')
  + print(soup) # HTML을 받아온 것을 확인할 수 있다.
  + #############################
  + # (입맛에 맞게 코딩)
  + #############################
* 4.위에서 파악한 구조를 이용하여 우선 select()를 이용해 영화들을 찾는다.⇒ 개발자도구의 copy selector 기능을 사용합니다.
  + import requests
  + from bs4 import BeautifulSoup
  + # 타겟 URL을 읽어서 HTML를 받아오고,
  + headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
  + url = '<https://www.yes24.com/Product/Category/BestSeller?categoryNumber=001&pageNumber=1&pageSize=50>'
  + data = requests.get(url,headers=headers)
  + # HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦
  + # soup이라는 변수에 "파싱 용이해진 html"이 담긴 상태가 됨
  + # 이제 코딩을 통해 필요한 부분을 추출하면 된다.
  + soup = BeautifulSoup(data.text, 'html.parser')
  + books = soup.select('#yesBestList > li:nth-child(1) > div > div.item\_info > div.info\_row.info\_name > a.gd\_name')
  + for m in books:

print(m.text)

* 5.각 영화에서 영화 제목, 설명이 적혀 있는 a 태그를 select\_one()으로 찾아본다.⇒ .text 를 통해 text 부분만 출력한다.⇒ .strip() 을 통해 좌우 공백을 없앤다.
  + import requests
  + from bs4 import BeautifulSoup
  + # 타겟 URL을 읽어서 HTML를 받아오고,
  + headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
  + url = '<https://www.yes24.com/Product/Category/BestSeller?categoryNumber=001&pageNumber=1&pageSize=50>'
  + data = requests.get(url,headers=headers)
  + # HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦
  + # soup이라는 변수에 "파싱 용이해진 html"이 담긴 상태가 됨
  + # 이제 코딩을 통해 필요한 부분을 추출하면 된다.
  + soup = BeautifulSoup(data.text, 'html.parser')
  + books = soup.select('#yesBestList > li')
  + for m in books:
  + title = m.select\_one('a.gd\_name').text.strip()
  + print(title)
* 6.결과

텍스트, 폰트, 스크린샷, 흑백이(가) 표시된 사진

자동 생성된 설명

beautifoulsoup 사용법 요약

* 위에서 본 것처럼 select()는 조건을 만족하는 모든 요소를 리스트에 담아 반환하고, select\_one()은 그 중 가장 위에 나오는 요소를 반환합니다.
* 선택자는 CSS를 작성할 때 썼던 것과 똑같이 쓰면 됩니다.
  + # 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')

**05. MongoDB Atlas**

* DB의 종류

텍스트, 스크린샷, 운영 체제, 디자인이(가) 표시된 사진

자동 생성된 설명

* 👉 Database는 크게 두 가지 종류로 나뉩니다.
  + RDBMS(SQL)은 행/열의 생김새가 정해진 엑셀에 데이터를 저장하는 것과 유사합니다. 데이터 50만 개가 적재된 상태에서, 갑자기 중간에 열을 하나 더하기는 어렵지만, 정형화되어 있는 만큼 데이터가 일관적이고 분석에 용이합니다. MS-SQL, My-SQL 등이 여기에 속하죠.
  + NoSQL은 딕셔너리 형태로 데이터를 저장해두는 DB로, 데이터 하나하나 마다 같은 필드 값들을 가질 필요가 없어 자유로운 형태의 데이터 적재에 유리한 대신, 일관성이 부족할 수 있습니다. 우리가 쓸 MongoDB가 이에 속합니다.

MongoDB

* 📖 MongoDB는 다양한 플랫폼에서 사용할 수 있는 NoSQL 타입의 데이터베이스 프로그램으로, JSON과 비슷한 형태로 자료를 정리합니다.
* MongoDB의 자료는 각각의 딕셔너리인 도큐먼트가 모여 컬렉션, 컬렉션이 모여 DB가 되는 형태입니다. 위에서 크롤링한 영화 정보를 예로 들면, 제목/순위/별점이 있는 각 영화 정보가 도큐먼트이고, 이것들을 모인 컬렉션을 크롤링 용 DB에 저장할 수 있겠죠. 만약 네이버 기사 제목을 크롤링했다면 그 결과는 같은 DB에 다른 컬렉션으로 저장할 수 있을 것입니다.

MongoDB Atlas 가입하기

* [Desktop 2023.02.22 - 11.13.07.03.mp4](https://prod-files-secure.s3.us-west-2.amazonaws.com/9d823e1e-56a0-477f-b622-1de1e2d46354/a6b463f3-263e-4521-b266-c41d2c5060cc/Desktop_2023.02.22_-_11.13.07.03.mp4)
* 👉 아래 순서에 따라 차근 차근 따라해주세요
* 1.가입하기
  + 구글로 로그인하고 → Accept Privacy ... Service 에 체크 → Submit 합니다.

텍스트, 스크린샷, 폰트, 그래픽 디자인이(가) 표시된 사진

자동 생성된 설명

* + **[코드스니펫] 회원가입하기**
    - <https://account.mongodb.com/account/register>
* 2.질문에 맞게 체크하고 넘어가기(크게 중요하지는 않습니다!)텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

  자동 생성된 설명
* 3.M0를 클릭하고, Seoul 선택하고 Create 클릭하기 (화면 축소하시면 M0 확인 가능합니다.)

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

* 4.Username에 sparta, Password에 jungle입력 후 Create User 클릭텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

  자동 생성된 설명
* 5.IP Address에 0.0.0.0 입력 후 Add Entry 클릭텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

  자동 생성된 설명
* 6.Hide Quickstart guide ... 에 체크 → Go to Databases합니다.

텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명

* 7.아래와 같은 화면이 잠시 동안 나온 뒤에

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

* 8.새로고침 후 최종 아래와 같은 화면을 만나면 끝!텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

  자동 생성된 설명
  + [2022-08-29 15-04-27.mp4](https://prod-files-secure.s3.us-west-2.amazonaws.com/9d823e1e-56a0-477f-b622-1de1e2d46354/f6ca0614-90fe-43ea-9652-478bd1960300/2022-08-29_15-04-27.mp4)

MongoDB Atlas 연결하기

* 😎 **pymongo 라이브러리의 역할**
* 예를 들어, MS Excel를 파이썬으로 조작하려면, 특별한 라이브러리가 필요하지 않겠어요?
* 마찬가지로, mongoDB 라는 프로그램을 조작하려면, 특별한 라이브러리, pymongo가 필요하답니다!
* 1.패키지 설치하기
  + pymongo, dnspython
  + 터미널을 열고, 마지막 경로가 pythonprac 인지 확인!
  + (venv) 로 가상환경이 활성화 되었는지 확인!
  + 가상환경 활성화가 기억이 안난다면! - **venv 활성화하기**
    - 😎 가상환경을 만든 것이 끝이 아닙니다! 활성화 시켜줘야 라이브러리를 컴퓨터가 읽어줘요!
    - Ctrl + 백틱()` 을 눌러 터미널을 열어줍니다(열려 있다면 그대로 진행해줍니다)
    - 터미널환경에서 projects 폴더가 마지막인지 확인 해주시고 코드를 입력해줍니다!
      * 윈도우는 ./venv/Scripts/activate
      * 맥은 source venv/bin/activate
    - 터미널에서 (venv) 라고 뜨게 된다면 가상환경이 활성화 된거에요!

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

* + pip install pymongo dnspython 을 입력하고 엔터!
* 2.pymongo로 조작하기
  + **[코드스니펫] pymongo 기본 코드**
    - from pymongo import MongoClient
    - client = MongoClient('여기에 URL 입력')
    - db = client.dbsparta
* 3.mongoDB와 Python 연결하기
  + 화면 보기
    - mongo atlas 화면에서 Connect를 클릭해주세요!

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

* + - 연결 방법 화면에서 Drivers을 클릭해주세요!

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

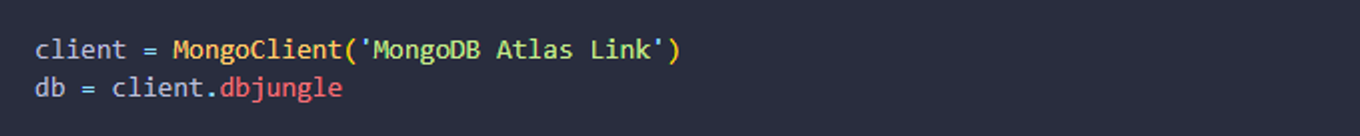
자동 생성된 설명

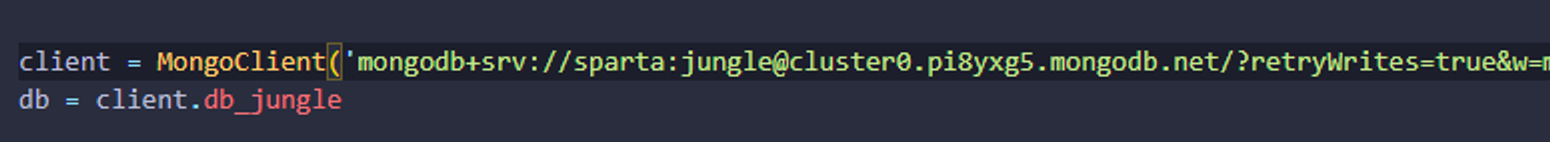
* + - 드라이버를 Python으로 버전을 3.6 or later로 클릭하신뒤, 아래에 생성된 링크 버튼을 클릭해 주소를 복사해주세요!

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* + - 붙여넣은 링크를 아래 코드의 MongoClient()안에 넣어주세요!





* 4.잘 연결 됐는지 테스트해보기
  + 아래와 같이 입력! (데이터 넣기. 곧 배워요!)
    - doc = {

'name':'영수',

'age':24

}

* + - db.users.insert\_one(doc)
* 5.다시 Cluster0의 Collections를 확인하면! 데이터가 잘 들어온 것을 확인 할 수 있답니다!
  + 👉 이제 한번 연결을 했으니, 복잡한 과정 없이 세 줄만 복사해서 쓰면 되겠네요!
  + 👉 Collections는 '즐겨찾기' 해두면 더 편하겠죠텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

    자동 생성된 설명
  + 💻 ⁉️ Mac에서 설치가 잘 되지 않는다면 아래를 확인해주세요!

- Q. DB Atlas 사용시, pymongo.errors.ServerSelectionTimeoutError: ~ [SSL: CERTIFICATE\_VERIFY\_FAILED]~ 를 포함한 에러 메시지가 뜨면서 실행이 되지 않습니다.

- A. 에러 메시지

텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

* + 사용하고 있는 인터넷 환경에 따라 보안 관련 추가 설정을 해주어야할 때가 있어요.
  + certifi 패키지를 가상환경에 먼저 설치해주세요~!
  + 패키지 설치하는 방법은 3주차 수업자료 = 05. 파이썬 패키지 설치하기 를 참고하세요.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* + 아래 형광펜 칠한 코드 부분을 추가해서 사용해보세요~! 특히 client = MongoClient 부분 코드 맨 끝에 있는 tlsCAFile=ca 를 추가하는 것 잊지 마세요!
  + 아래 코드에서 내주소 를 실제 내 DB 주소로 바꾸는 것 잊지 마세요~! 😉
    - from pymongo import MongoClient
    - import certify
    - ca = certifi.where()
    - client = MongoClient('mongodb+srv://test:sparta@cluster0.내주소.mongodb.net/내DB명?retryWrites=true&w=majority', tlsCAFile=ca)
    - db = client.dbsparta
    - doc = {

'name':'bob',

'age':27

}

* + - db.users.insert\_one(doc)

Q. DB Atlas 사용시, DB 접속권한 업데이트하기 - [app.py](http://app.py) 실행시 DNS 관련한 에러 메시지가 뜨면서 접속이 안됩니다. (**예전에는 접속이 되었는데 지금은 DB 접속이 안되어요.** AWS 에서는 접속이 안되어요 포함)

A. 이 경우에는 DB 접속 권한이 사라져서 그렇습니다. DB 접속 권한을 업데이트해주면 됩니다.

- 이런 문제가 발생하는 이유는 아래를 참고하세요~

- 처음 DB Atlas 세팅시에는 내 컴퓨터에서만 DB에 접속될 수 있게 현재 세팅시 접속된 특정 IP, 즉, 해당 시점의 내 컴퓨터의 네트워크 주소만 접속될 수 있게 자동 세팅됩니다. 그런데 일반적인 컴퓨터 접속환경에서는 이 IP 주소가 유동적으로 변경됩니다. (변경되는 주기는 공유기 세팅마다 다릅니다.) 또 어떤 인터넷을 사용하느냐에 따라서 인터넷이 달라지면-접속환경이 달라지면- IP 주소도 바뀝니다. 즉, 내 컴퓨터 네트워크 주소가 변경되어버리는 것이지요.

- 즉, DB에 접근가능하게 등록되어있던 주소와 내 컴퓨터 실제 네트워크 주소가 달라지게 됩니다. 그러면 이전에 DB 등록 주소를 바꾸어주어야겠지요? 비유하자면, **ID 가 0000인 사람만 접속할 수 있게 등록이 되어있는데, 내 ID 가 000에서 다른 것으로 변경되어버리는 것과 똑같습니다.**

- 1.DB Atals 페이지에서 접속 후 아래 페이지로 들어가주세요.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 2.0.0.0.0/0 을 적은 후, confirm 해서 등록
  + 0.0.0.0/0 은 모든 IP 주소라는 뜻입니다. 즉, 접속 IP에 상관없이 어디에서나 접근가능하다는 뜻이에요. 비밀번호와 user명만으로 접속할 수 있게 되는 것입니다.
    - 참고! 실제 현업에서 DB 를 사용할 때에는 보안상의 이유로 특정 IP 주소로만 접근할 수 있게 해둡니다. 우리는 배우는 과정이니 모든 IP 에서 접근할 수 있게 설정을 바꿔줄게요~
    - ADD CURRENT IP ADDRESS 를 누르면 현재 접속한 IP 주소로 등록시킬 수 있지만, 이건 매번 내 컴퓨터 IP 주소가 바뀔 때마다 설정해주어야해요~

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 3.아래처럼 0.0.0.0/0 이 등록되고 status 가 Active 가 뜰 때까지 기다려주세요.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 4.그 후에 다시 [app.py](http://app.py) 를 실행 후에 내가 만든 사이트에 접속해서 제대로 뜨는지 확인해보세요.
  + 만약 정상 실행이 안된다면, 이전 설정이 남아있어서 발생하는 문제일 수 있습니다~ 컴퓨터 재부팅하고 프로젝트를 새로 만들어 실행시켜보세요!
* 5.(위 과정으로 해결안되는 사람만!) 똑같이 DNS 관련 오류가 뜬다면?
  + 내 인터넷 접속환경과 방화벽 설정에서 SRV URI (예. mongodb+srv: 처럼 srv 방식으로 적힌 URI) 자체를 접속이 되지 않게 막는 경우가 있습니다. 이럴 때는 다른 식으로 적힌 url 주소를 사용하면 됩니다~
  + 1.아래 접속 화면에서 3.4 or later 를 선택해주세요~ 그러면 SRV 가 포함되지 않은 url 주소가 나옵니다!텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

    자동 생성된 설명
  + 2.password 는 sparta 로, myFirstDatabase 는 똑같이 DB 주소로 바꾸어 사용하시면 됩니다!

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

* + 3.아래 코드 예시처럼 MongoClient 부분 코드에 적어주세요!
    - 아래는 코드 예시이니 꼭 내 진짜 mongodb 주소로 적어주셔야해요~!
    - client = MongoClient('mongodb://test:sparta@cluster0-shard-00-00.ttp4x.mongodb.net:27017,cluster0-shard-00-01.ttp4x.mongodb.net:27017,cluster0-shard-00-02.ttp4x.mongodb.net:27017/cluster0?ssl=true&replicaSet=atlas-cyee0j-shard-0&authSource=admin&retryWrites=true&w=majority')

pymongo로 mongoDB 조작하기

* DB연결하기 & 데이터 넣기
  + # 'users'라는 collection에 {'name':'bobby','age':21}를 넣습니다.
  + db.users.insert\_one({'name':'영희','age':30})
  + db.users.insert\_one({'name':'철수','age':20})
  + db.users.insert\_one({'name':'john','age':30})
* 모든 결과 값을 보기
  + **[코드스니펫] pymongo(find)**
    - all\_users = list(db.users.find({},{'\_id':False}))
  + # 모든 데이터 뽑아보기
  + all\_users = list(db.users.find({},{'\_id':False}))
  + print(all\_users[0]) # 0번째 결과값을 보기
  + print(all\_users[0]['name']) # 0번째 결과값의 'name'을 보기
  + for a in all\_users: # 반복문을 돌며 모든 결과값을 보기

print(a)

* 특정 결과 값을 뽑아 보기
  + **[코드스니펫] pymongo(find\_one)**
    - user = db.users.find\_one({})
  + user = db.users.find\_one({})
  + print(user)
* 수정하기
  + **[코드스니펫] pymongo(update\_one)**
    - db.users.update\_one({'name':'영수'},{'$set':{'age':19}})
  + # 오타가 많으니 이 줄을 복사해서 씁시다!
  + db.users.update\_one({'name':'영수'},{'$set':{'age':19}})
  + user = db.users.find\_one({'name':'영수'})
  + print(user)
* 삭제하기 (거의 안 씀)
  + **[코드스니펫] pymongo(delete\_one)**
    - db.users.delete\_one({'name':'영수'})
  + db.users.delete\_one({'name':'영수'})
  + user = db.users.find\_one({'name':'영수'})
  + print(user)

pymongo 사용법 요약

* 위에서 쓴 기능들을 요약해보았습니다.
  + # 저장 – 예시
  + doc = {'name':'bobby','age':21}
  + db.users.insert\_one(doc)
  + # 한 개 찾기 – 예시
  + user = db.users.find\_one({'name':'bobby'})
  + # 여러개 찾기 - 예시 ( \_id 값은 제외하고 출력)
  + same\_ages = list(db.users.find({'age':21},{'\_id':False}))
  + # 바꾸기 – 예시
  + db.users.update\_one({'name':'bobby'},{'$set':{'age':19}})
  + # 지우기 – 예시
  + db.users.delete\_one({'name':'bobby'})