

# 빅데이터시스템

TECHNOLOGY ANALYTICS

파이썬크롤링 API이용



# 목 차

## 01 크롤링

크롤링, 웹API

## 02 네이버API

가입, 요청과응답, 뉴스크롤링(전체작업설계, 프로그램구성설계, 프로그램, 실행, 확인)

## 03 공공데이터API

가입, 검색, 활용신청, 인증키확인, 서비스확인, 요청과응답, 전체작업설계, 프로그램구성설계, 프로그램, 실행

## 04 nodejs와 Flask API 연동

nodejs, 설치, 디렉토리구성, 프로그램코딩, 실행, 접속

# \*. 전수업리뷰

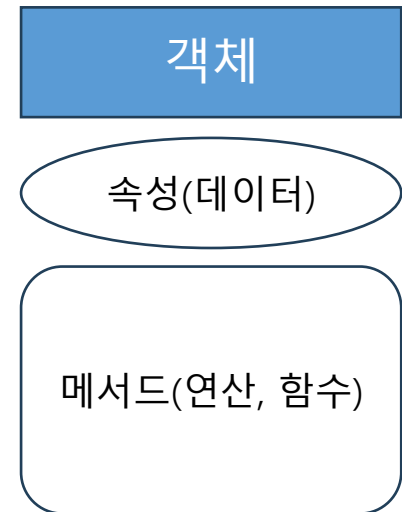
## ⌚ 파이썬 변수와 객체

### ■ 변수

- 값을 저장하는 메모리 공간
- 파이썬에서는 변수를 미리 선언하지 않음
- 변수에 저장해서 사용하는 값의 자료형으로 변수의 자료형이 결정

### ■ 객체

- 변수 형태의 속성과 함수 형태의 메서드를 가진 것
- 각 객체는 자기의 속성(내부 데이터)과 메서드(내부 연산)를 가짐
- 타 프로그래밍 언어와 달리 파이썬에서는 모든 변수와 자료형이 객체로 되어 있음



# \*. 전수업리뷰

## ⌚ 자료형과 연산자

### ■ 숫자형

```
>>> a = 123          정수
>>> b = 12.34        실수
>>> c = 1 + 2j        복소수(a + bi)
>>> c.real            실수
1.0
>>> c.imag            허수
2.0
>>> c.conjugate()     켈레복소수
(1 - 2j)
>>> abs(c)            절대값
2.23606797749979
>>> d = 0o12          8진수
>>> d
10
>>> e = 0x12A         16진수
>>> e
298
```

### ■ 논리형

```
>>> l = True
>>> l
True
```

```
C:\> 명령 프롬프트 - python
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Wdossa>
C:\Users\Wdossa>cd W_seok
C:\W_seok>mkdir test
C:\W_seok>
C:\W_seok>cd test
C:\W_seok\Wtest>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023)
Type "help", "copyright", "credits" or "license()" for more
>>>
>>>
```

### ■ 연산자

```
>>> a = 3
>>> b = 4
>>> a + b            더하기
7
>>> a - b            빼기
-1
>>> a * b            곱하기
12
>>> a / b            나누기
0.75
>>> a ** b           제곱
81
>>> a % b            나머지
3
>>> a // b           몫
0
```

# \*. 전수업리뷰

## ⌚ 자료형과 연산자

### ■ 문자열 자료형

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
N	o	w		i	s		b	e	t	t	e	r		t	h	a	n		n	e	v	e	r

### ■ 문자열

```
>>> s1 = 'Hello Python'
>>> s1
'Hello Python'
>>> s2 = "Hello Python"
>>> s2
'Hello Python'
>>> s3 = '''Hello Python'''
>>> s3
'Hello Python'
>>> s4 = """"Hello Python""""
>>> s4
'Hello Python'
```

### ■ 연산자

```
>>> head = "Python"
>>> tail = " is fun"
>>> head + tail
'Python is fun'
>>> head * 2
'PythonPython'
>>> print("=" * 5)
=====
>>> a = "Now is better than never"
>>> a[0]
'N'
>>> a[-1]
'r'
```

```
>>> b = a[0] + a[1] + a[2]
>>> b
'Now'
>>> a[0:3]
'Now'
>>> a[:3]
'Now'
>>> a[19:]
'never'
>>> a[:]
'Now is better than never'
```

# \*. 전수업리뷰

## ⌚ 자료형과 연산자

### ■ 문자열 자료형

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
N	o	w		i	s		b	e	t	t	e	r		t	h	a	n		n	e	v	e	r

### ■ 연산자(계속)

```
>>> c = "python"
>>> c.count('y')
1
>>> c.find('y')
1
>>> c.find('x')
-1
>>> c.index('y')
1
```

```
>>> b = ","
>>> d = b.join('abcd')
>>> d
'a, b, c, d'
>>> c.upper()
'PYTHON'
>>> c.lower()
'python'
>>> d = " py "
>>> d.lstrip()
'py '
>>> d.rstrip()
' py'
```

```
>>> d.strip()
'py'
```

```
>>> a = "Python is difficult."
>>> a.replace("difficult", "funny")
'Python is funny.'
>>> a.split()
['Python', 'is', 'difficult.']
>>> b = "a, b, c, d"
>>> b
'a, b, c, d'
>>> b.split(',')
['a', 'b', 'c', 'd']
```

# \*. 전수업리뷰

## ⌚ 자료형과 연산자

### ■ 리스트 자료형 연산자

```
>>> a = [1, 2, 3]
>>> b = ['Life', 'is', 'too', 'short']
>>> c = [1, 2, 'Life', 'is']
>>> d = [1, 2, [3, 4], ['Life', 'is']]
>>> d[0]
1
>>> d[2]
[3, 4]
>>> d[3][-1]
'is'
>>> d[0:3]
[1, 2, [3, 4]]
>>> a + b
[1, 2, 3, 'Life', 'is', 'too', 'short']
>>> b[0] + " hi~ ^^;"
'Life hi~ ^^;'
```

```
>>> a * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
>>> a[2] = 99
>>> a
[1, 2, 99]
>>> a[1:2] = ['a', 'b', 'c']
>>> a
[1, 'a', 'b', 'c', 99]
>>> a[-1] = ['d', 'e', 'f']
>>> a
[1, 'a', 'b', 'c', ['d', 'e', 'f']]
>>> del a[-1]
>>> a
[1, 'a', 'b', 'c']
>>> a.append(5)
>>> a
[1, 'a', 'b', 'c', 5]
```

```
>>> a = [3, 4, 1, 9]
>>> a.reverse()
>>> a
[9, 1, 4, 3]
>>> a.insert(0, 99)
>>> a
[99, 9, 1, 4, 3]
>>> a.remove(99)
>>> a
[9, 1, 4, 3]
>>> a = [2, 1, 0, 2, 3, 2, 4, 2]
>>> a.count(2)
4
```

# \*. 전수업리뷰

## ⌚ 자료형과 연산자

### ■ 튜플 자료형 연산자

```
>>> t1 = (1, )
>>> t2 = (1, 2, 3)
>>> t3 = 1, 2, 3
>>> t4 = (1, 2, (3, 4), ('Life', 'is'))
>>> t4[0]
1
>>> t4[3][-1]
'is'
>>> t4[0:3]
(1, 2, (3, 4))
>>> t1 + t2
(1, 1, 2, 3)
>>> t1 + "hi~"
```

Traceback (most recent call last):  
File "<pyshell#157>", line 1, in <module>

```
>>> t2 * 3
(1, 2, 3, 1, 2, 3, 1, 2, 3)
>>> t2[2] = 99
```

Traceback (most recent call last):  
File "<pyshell#159>", line 1, in <module>  
t2[2] = 99  
TypeError: 'tuple' object does not support item assignment



# \*. 전수업리뷰

## ⌚ 자료형과 연산자

### ■ 딕셔너리 자료형 연산자

```
>>> dic = {'name':'Hong', 'phone':'01012345678', 'birth':'0814'}
>>> dic[1] = 'a'
>>> dic
{'name': 'Hong', 'phone': '01012345678', 'birth': '0814', 1: 'a'}
>>> dic['pet'] = 'dog'
>>> dic
{'name': 'Hong', 'phone': '01012345678', 'birth': '0814', 1: 'a', 'pet': 'dog'}
>>> del dic[1]
>>> dic
{'name': 'Hong', 'phone': '01012345678', 'birth': '0814', 'pet': 'dog'}
>>> dic['name']
'Hong'
>>> dic.keys()
dict_keys(['name', 'phone', 'birth', 'pet'])
>>> list(dic.keys())
['name', 'phone', 'birth', 'pet']
```

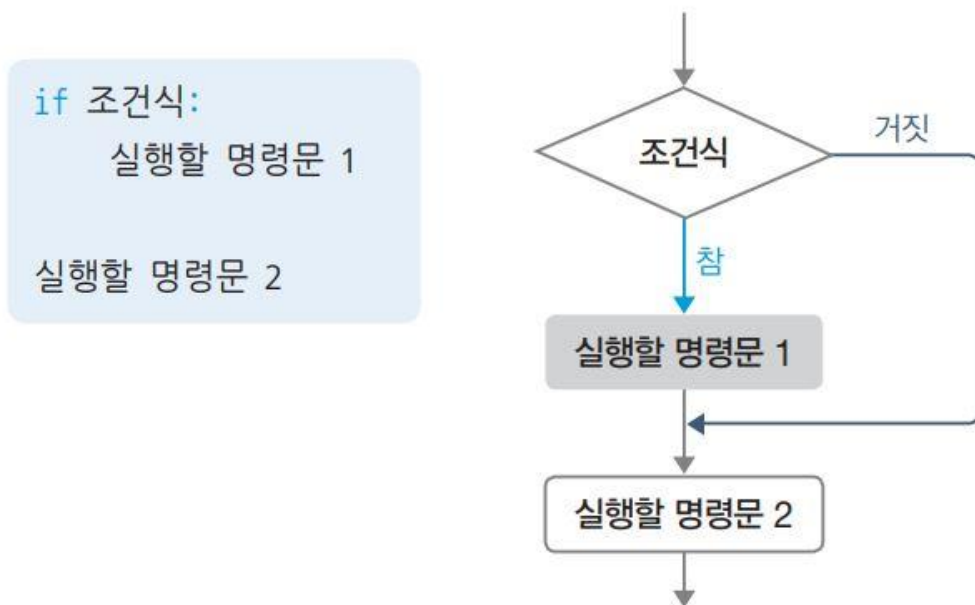
```
>>> dic.values()
dict_values(['Hong', '01012345678', '0814', 'dog'])
>>> list(dic.values())
['Hong', '01012345678', '0814', 'dog']
>>> dic.items()
dict_items([('name', 'Hong'), ('phone', '01012345678'), ('birth', '0814'), ('pet', 'dog')])
>>> dic.clear()
>>> dic
{}

```

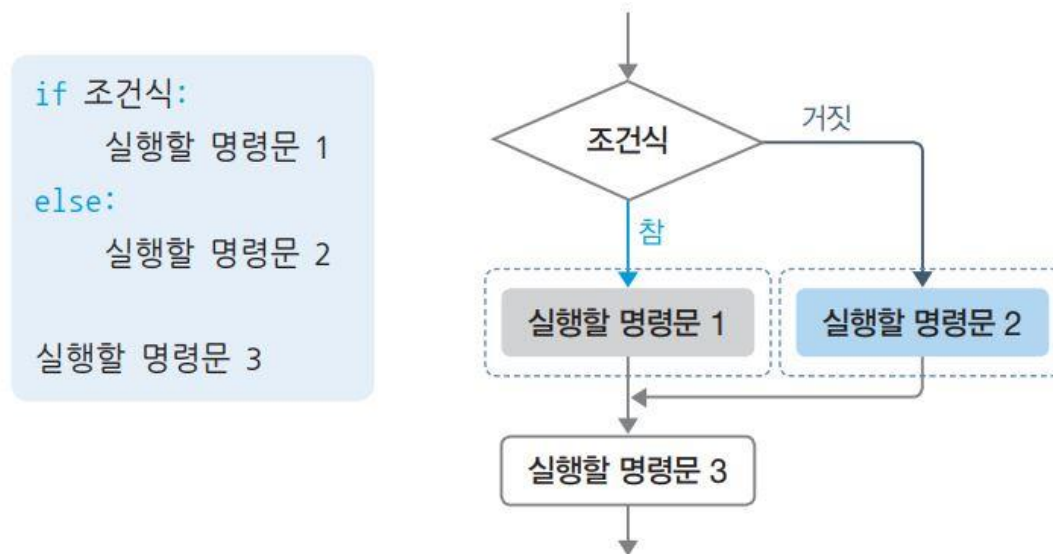
# \*. 전수업리뷰

## ⌚ 제어문

### ■ if문



### ■ if-else문

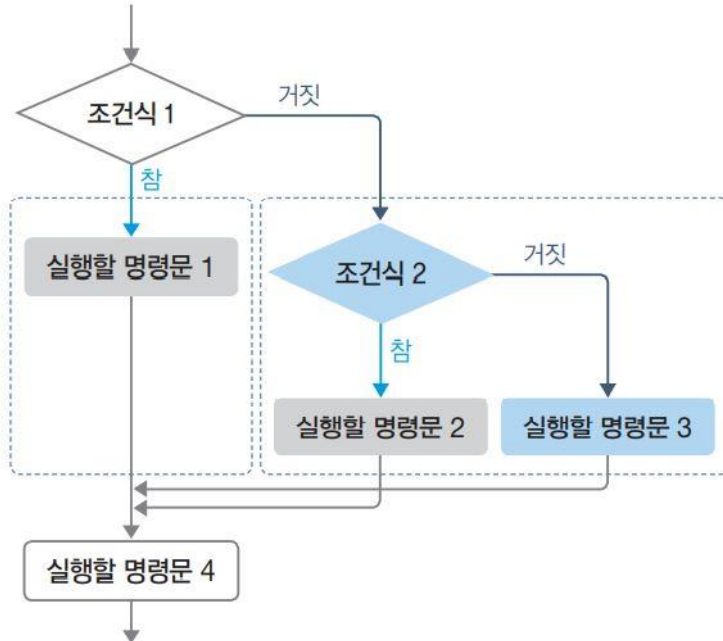


# \*. 전수업리뷰

## ⌚ 제어문

### ■ if-elif-else문

```
if 조건식 1:
    실행할 명령문 1
elif 조건식 2:
    실행할 명령문 2
else:
    실행할 명령문 3
실행할 명령문 4
```



### ■ 조건 연산자

```
>>> x = 3
>>> y = 2
>>> x == y
False
>>> x != y
True
>>> x >= y
True
>>> x in [1, 2, 3]
True
>>> x not in [1, 2, 3]
False
>>> money = 1300
>>> if money >= 1200:
    print("OK")
```

OK

# \*. 전수업리뷰

## ⌚ 제어문

### ■ for문

```

>>> list = ['one', 'two', 'three']
>>> for i in list:
    x = i + '!'
    print(x)
    
```

```

one!
two!
three!
    
```

### ■ while문

```

>>> i = 0
>>> while i < 5:
    i += 1
    print('*' * i)
    
```

```

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

# \*. 전수업리뷰

## ⌚ 사용자함수

### ■ 함수 정의

```
>>> def sum1(a, b):
    x = a + b
    return x
```

```
>>> def sum2(*args):
    x = 0
    for i in args:
        x += i
    return x
```

### ■ 함수 호출

```
>>> a = 5
>>> b = 3
>>> sum1(a, b)
8
>>> sum1(3, 5)
8
>>> sum2(1, 2, 3, 4, 5)
15
>>> sum2(2, 3.5, 10)
15.5
```



# \*. 전수업리뷰

## ⌚ 모듈과 패키지

### ■ 파일 쓰기

```
>>> f = open("test1.txt", 'w')
>>> f
<_io.TextIOWrapper name='test1.txt' mode='w'
encoding='cp949'>
>>> f.close()
>>> f = open("test2.txt", 'w')
>>> for i in range(1, 6):
    data = ("%d번째 줄입니다. \n" % i)
    f.write(data)
<Enter>
>>> f.close()

^Z
dir test*
```

### ■ 파일 읽기

```
>>> f = open("test2.txt", 'r')
>>> line = f.readline()
>>> print(line)
1번째 줄입니다.
>>> while True:
    line = f.readline()
    if not line: break
    print(line)
<Enter>
2번째 줄입니다.
3번째 줄입니다.
4번째 줄입니다.
5번째 줄입니다.
.....
>>> f.close()
```

# \*. 전수업리뷰

## ⌚ 데이터분석용 라이브러리 numpy

```
>>> import numpy as np
>>> np.__version__
'1.23.5'
>>> ar1 = np.array([1, 2, 3, 4, 5])
>>> ar1
array([1, 2, 3, 4, 5])
>>> type(ar1)
<class 'numpy.ndarray'>
>>> ar2 = np.array([[10, 20, 30], [40, 50, 60]])
>>> ar2
array([[10, 20, 30],
       [40, 50, 60]])
>>> ar3 = np.arange(1, 11, 2)
>>> ar3
array([1, 3, 5, 7, 9])
>>> ar4 = np.array([1, 2, 3, 4, 5, 6]).reshape((3, 2))
```

```
>>> ar4
array([[1, 2],
       [3, 4],
       [5, 6]])
>>> ar5 = np.zeros((2, 3))
>>> ar5
array([[0., 0., 0.],
       [0., 0., 0.]])
>>> ar6 = ar2[0:2, 0:2]
>>> ar6
array([[10, 20],
       [40, 50]])
>>> ar7 = ar2[0, :]
>>> ar7
array([10, 20, 30])
>>> ar8 = ar1 + 10
>>> ar8
array([11, 12, 13, 14, 15])
```

```
>>> ar1 + ar8
array([12, 14, 16, 18, 20])
>>> ar8 - ar1
array([10, 10, 10, 10, 10])
>>> ar1 * 2
array([ 2, 4, 6, 8, 10])
>>> ar1 / 2
array([0.5, 1. , 1.5, 2. , 2.5])
>>> ar9 = np.dot(ar2, ar4)
>>> ar9
array([[220, 280],
       [490, 640)])
```

```
[[10, 20, 30],
 [40, 50, 60]]
```

```
[[1, 2],
 [3, 4],
 [5, 6]]
```

# \*. 전수업리뷰

## ⌚ pandas

### ■ Series 자료형

```
>>> import pandas as pd
>>> pd.__version__
'1.0.3'
>>> data1 = [10, 20, 30, 40, 50]
>>> data1
[10, 20, 30, 40, 50]
>>> data2 = ['1반', '2반', '3반', '4반', '5반']
>>> data2
['1반', '2반', '3반', '4반', '5반']
```

```
>>> sr1 = pd.Series(data1)
>>> sr1
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

```
>>> sr2 = pd.Series(data2)
>>> sr2
0    1반
1    2반
2    3반
3    4반
4    5반
dtype: object
>>> sr3 = pd.Series([101, 102, 103, 104, 105])
>>> sr3
0    101
1    102
2    103
3    104
4    105
dtype: int64
>>> sr4 = pd.Series(['월', '화', '수', '목', '금'])
>>> sr4
0    월
1    화
2    수
3    목
4    금
dtype: object
```

# \*. 전수업리뷰

## ⌚ pandas

### ■ Series 자료형

```
>>> sr5 = pd.Series(data1, index = [1000,
1001, 1002, 1003, 1004])
```

```
>>> sr5
1000  10
1001  20
1002  30
1003  40
1004  50
dtype: int64
```

```
>>> sr6 = pd.Series(data1, index = data2)
```

```
>>> sr6
1반   10
2반   20
3반   30
4반   40
5반   50
dtype: int64
```

```
>>> sr7 = pd.Series(data2, index = data1)
```

```
>>> sr7
10  1반
20  2반
30  3반
40  4반
50  5반
dtype: object
```

```
>>> sr8 = pd.Series(data2, index = sr4)
```

```
>>> sr8
월   1반
화   2반
수   3반
목   4반
금   5반
dtype: object
```

```
>>> sr8[2]
```

```
'3반'
```

```
>>> sr8['수']
```

```
'3반'
```

```
>>> sr8[-1]
```

```
'5반'
```

# \*. 전수업리뷰

## ⌚ pandas

### ■ Series 자료형

```
>>> sr8[0:4]
월 1반
화 2반
수 3반
목 4반
dtype: object>>> sr8.index
Index(['월', '화', '수', '목', '금'], dtype =
'object')
>>> sr8.values
array(['1반', '2반', '3반', '4반', '5반'], dtype =
object)
```

```
>>> sr1 + sr3
0 111
1 122
2 133
3 144
4 155
dtype: int64
>>> sr4 + sr2
0 월1반
1 화2반
2 수3반
3 목4반
4 금5반
dtype: object
```



# \*. 전수업리뷰

## ⌚ pandas

### ■ DataFrame 자료형

```
>>> data_dic = {
    'year': [2021, 2022, 2023],
    'sales': [350, 480, 1099]
}
>>> data_dic
{'year': [2021, 2022, 2023], 'sales': [350, 380, 1099]}
>>> df1 = pd.DataFrame(data_dic)
>>> df1
   year  sales
0  2021    350
1  2022    380
2  2023   1099
>>> df2 = pd.DataFrame([[89.2, 92.5, 90.8], [92.8, 89.9, 95.2]],
index = ['중간고사', '기말고사'], columns = data2[0:3])
```

```
>>> df2
      1반    2반    3반
중간고사 89.2   92.5   90.8
기말고사 92.8   89.9   95.2
>>> data_list = [['20231101', 'Hong', '90', '95'],
['20231102', 'Kim', '93', '94'], ['20231103', 'Lee', '87', '97']]
>>> df3 = pd.DataFrame(data_list)
>>> df3
      0      1      2      3
0  20231101  Hong    90    95
1  20231102  Kim    93    94
2  20231103  Lee    87    97
>>> df3.columns = ['학번', '이름', '중간고사', '기말고사']
>>> df3
      학번      이름  중간고사  기말고사
0  20231101  Hong    90    95
1  20231102  Kim    93    94
2  20231103  Lee    87    97
```

# \*. 전수업리뷰

## ⌚ pandas

### ■ DataFrame 자료형

```
>>> df3.head(2)
```

	학번	이름	중간고사	기말고사
0	20231101	Hong	90	95
1	20231102	Kim	93	94

```
>>> df3.tail(2)
```

	학번	이름	중간고사	기말고사
1	20231102	Kim	93	94
2	20231103	Lee	87	97

```
>>> df3['이름']
```

```
0 Hong
1 Kim
2 Lee
```

```
Name: 이름, dtype: object
```

```
C:\_seok\test>type score.csv
,0,1,2,3
0,20231101,Hong,90,95
1,20231102,Kim,93,94
2,20231103,Lee,87,97
C:\_seok\test>
```

```
>>> df3.to_csv('./score.csv', header = 'False')
```

```
>>> df4 = pd.read_csv('./score.csv', encoding='utf-8', index_col=0)
```

```
>>> df4
```

	학번	이름	중간고사	기말고사
0	20231101	Hong	90	95
1	20231102	Kim	93	94
2	20231103	Lee	87	97

# \*. 전수업리뷰

## ⌚ matplotlib

### ■ 라인플롯 차트 그리기

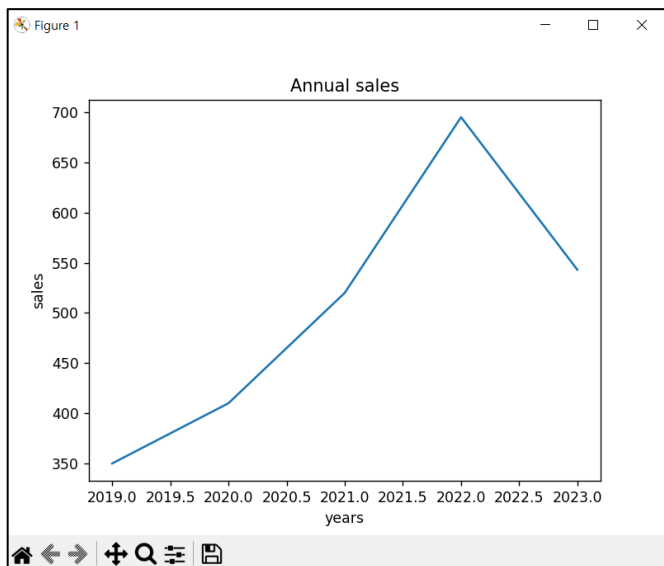
`python -m pip install matplotlib`

```
>>> import matplotlib
```

```
>>> matplotlib.__version__
```

'3.8.0'

```
>>> import matplotlib.pyplot as plt
```



#### 1. 데이터 준비

```
>>> x = [2019, 2020, 2021, 2022, 2023]
```

```
>>> y = [350, 410, 520, 695, 543]
```

#### 2. x축과 y축 데이터를 지정하여 라인플롯 생성

```
>>> plt.plot(x, y, color='blue')
```

[<matplotlib.lines.Line2D object at 0x0000015DB82D58C8>]

#### 3. 차트 제목 설정

```
>>> plt.title('Annual sales')
```

Text(0.5, 1.0, 'Annual sales')

#### 4. x축 레이블 설정

```
>>> plt.xlabel('years')
```

Text(0.5, 0, 'years')

#### 5. y축 레이블 설정

```
>>> plt.ylabel('sales')
```

Text(0, 0.5, 'sales')

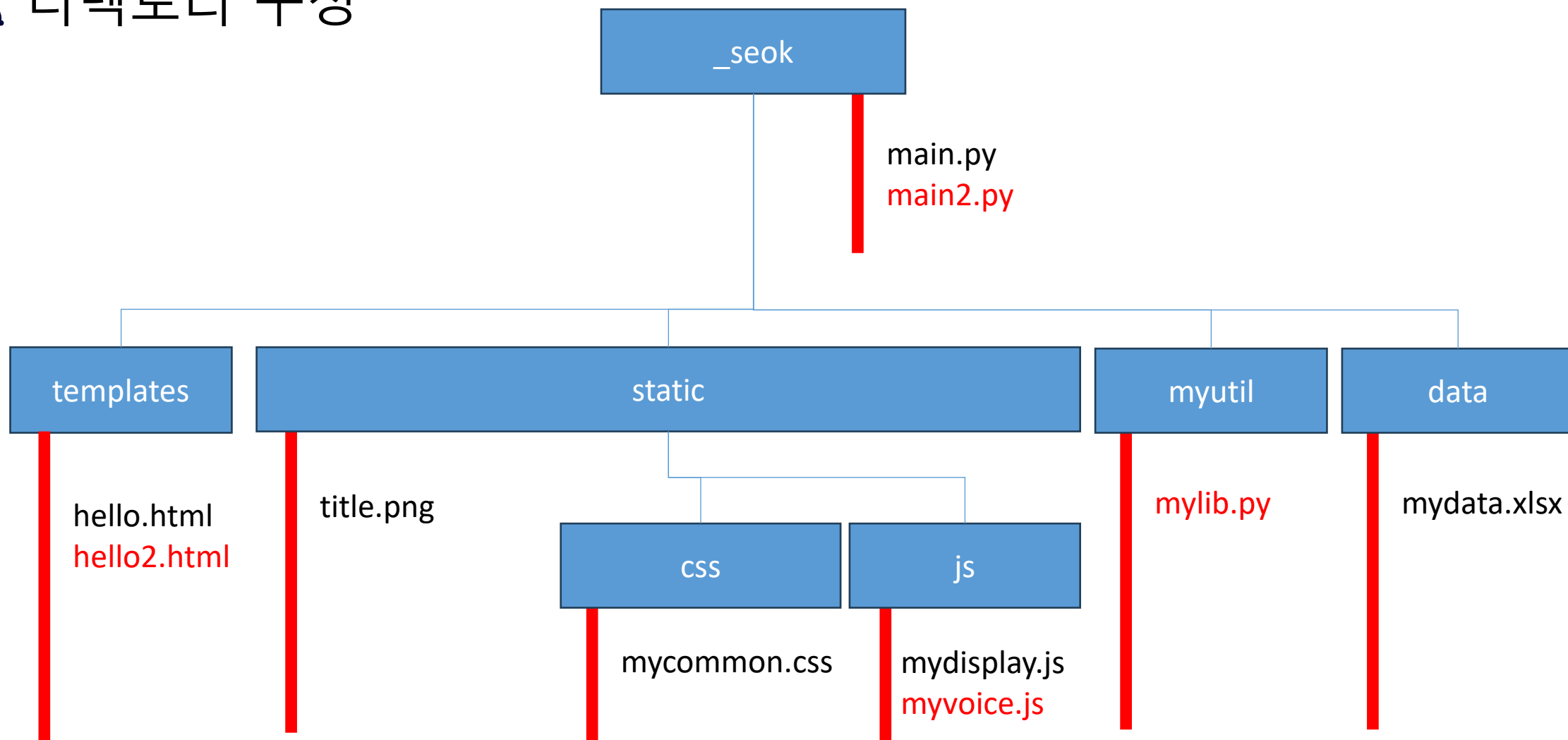
#### 6. 라인플롯 표시

```
>>> plt.show()
```

# \*. 전수업리뷰

## ⌚ 디렉토리 구성

※스마트폰으로 접속할 수 있도록 소켓 수정  
음성인식 모듈 추가  
라이브러리 추가  
음성인식용 자바스크립트 추가



# \*. 전수업리뷰

## ⌚ 웹서버 실행

- cmd 창을 열고 작업

명령 프롬프트 - python main2.py

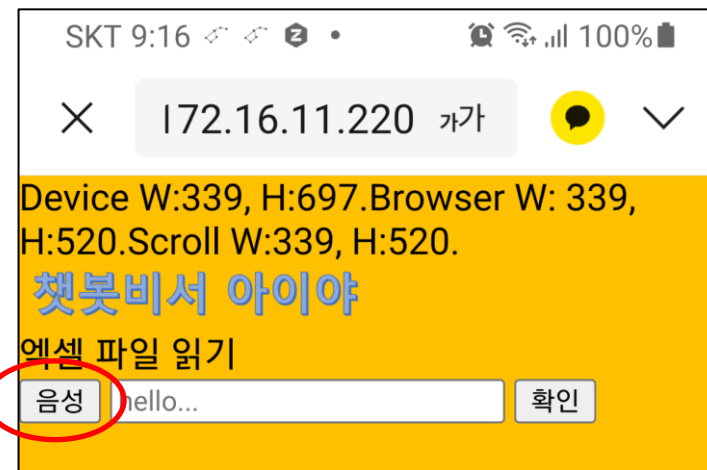
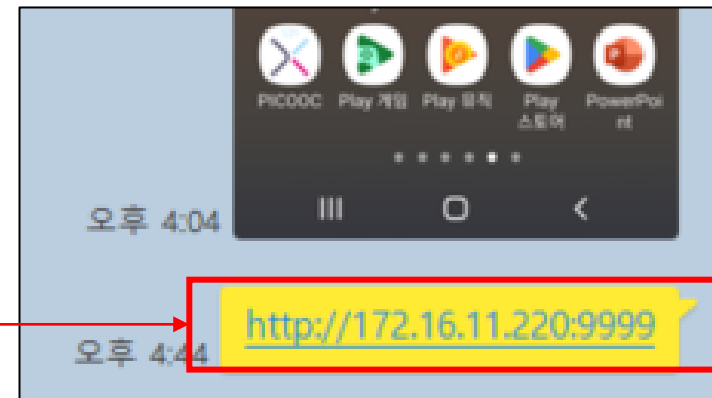
```
D:\seok>python main2.py
* Serving Flask app 'main2'
* Debug mode: off
WARNING: This is a development server. Do not use it in production.
* Running on http://172.16.11.220:9999
Press CTRL+C to quit
```

d:

cd \_seok

python main2.py

<동일한 Wifi 존>  
카카오톡에 주소 입력 후 클릭

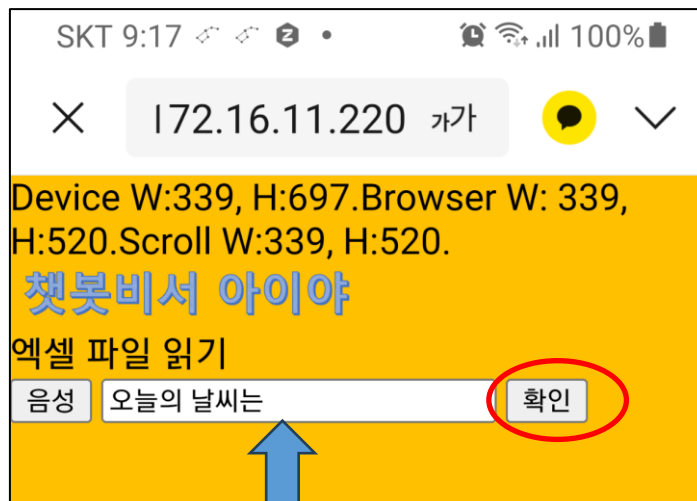
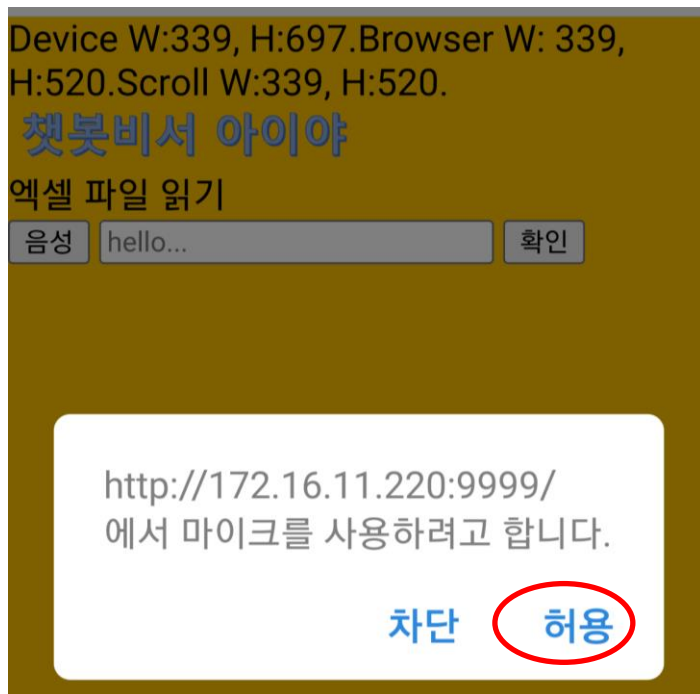




# \*. 전수업리뷰

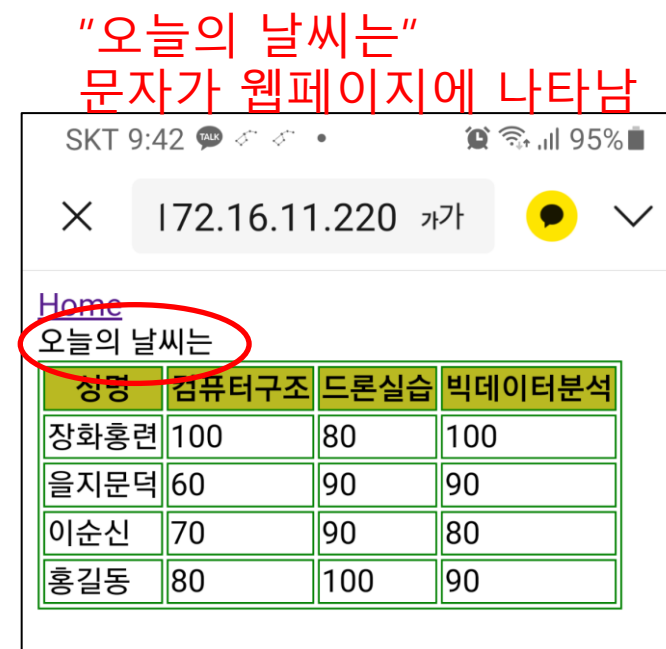
## ⌚ 웹서버 실행

- 스마트폰에서 접속(동일한 Wifi 존)



뽁~ 소리가 나면,  
음성으로 말함  
"오늘의 날씨는"

"오늘의 날씨는"  
문자가 입력이 됨



# \*. 전달 사항



## 주교재

- PowerPoint 로 만든 pdf 자료
- 데이터 과학 기반의 파이썬 빅데이터 분석 (이지영 지음, 한빛아카데미)

## 부교재

- 필요 시, 영상 공유



# \*. 전달 사항

## RoadMap

### Hadoop설치

- ✓ VM 셋업
- ✓ JDK
- ✓ Python
- ✓ Hadoop Engine
- ✓ Spark Engine
- ✓ Zeppelin

### 빅데이터분석

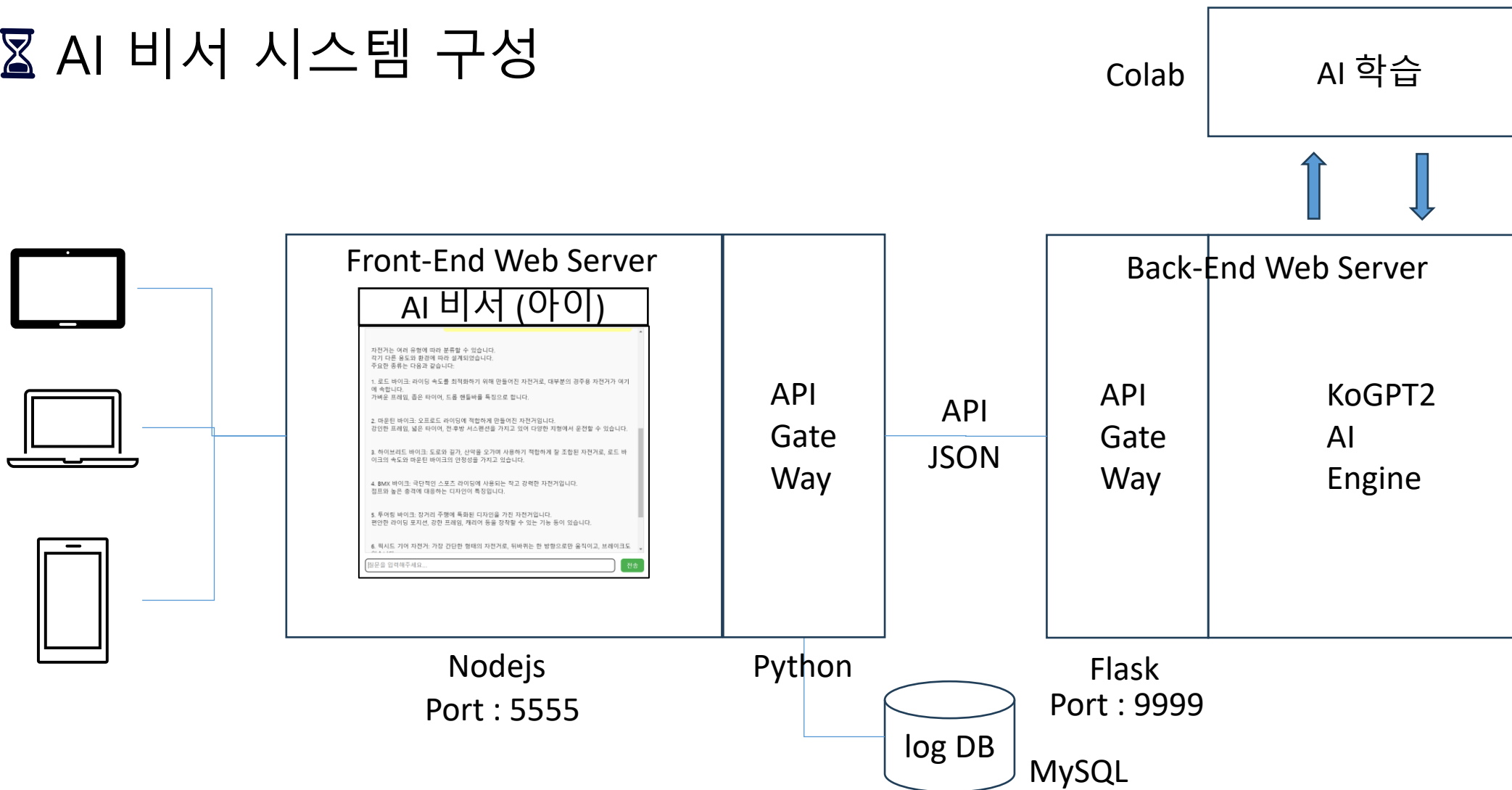
- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

### AI 비서학습

- ✓ 챗봇 데이터 수집
- ✓ Flask 웹서버
- ✓ Nodejs API 연동
- ✓ KoGPT2 환경구성
- ✓ Colab을 이용한 학습
- ✓ 말풍선생성기 활용
- ✓ MySQL
- ✓ 챗봇 비서 만들기

# \*. 전달 사항

## ⌚ AI 비서 시스템 구성



# 1. 크롤링

 크롤링 : 웹페이지를 그대로 가져와서 그곳으로 부터 데이터를 추출

## ■ 의의

- 일반적으로 검색엔진에서 사용하는 기술 요소
- 사람이 일일이 해당 사이트의 정보를 검색하는 것이 아니라 컴퓨터 프로그램의 미리 입력된 방식에 따라 끊임없이 새로운 웹페이지를 찾아 종합하고 색인하는 작업을 반복 수행
- 크롤링하는 행위를 하는 소프트웨어를 크롤러(crawler) 라고 부름

## ■ 종류

- 스크래핑(scraping) : 하나의 페이지를 수집함
- 크롤링(crawling) : 동적으로 웹페이지를 돌아다니면서 데이터 수집



# 1. 크롤링

※ API(Application Programming Interface) : 컴퓨터나 컴퓨터 사이의 연결

## ⌚ 웹 API

### ■ 의의

- 웹 API는 일반적으로 HTTP 통신을 사용하는데 사용
- 지도, 검색, 주가, 환율 등 다양한 정보를 가지고 있는 웹 사이트의 기능을 외부에서 쉽게 사용할 수 있도록 사용 절차와 규약을 정의한 것

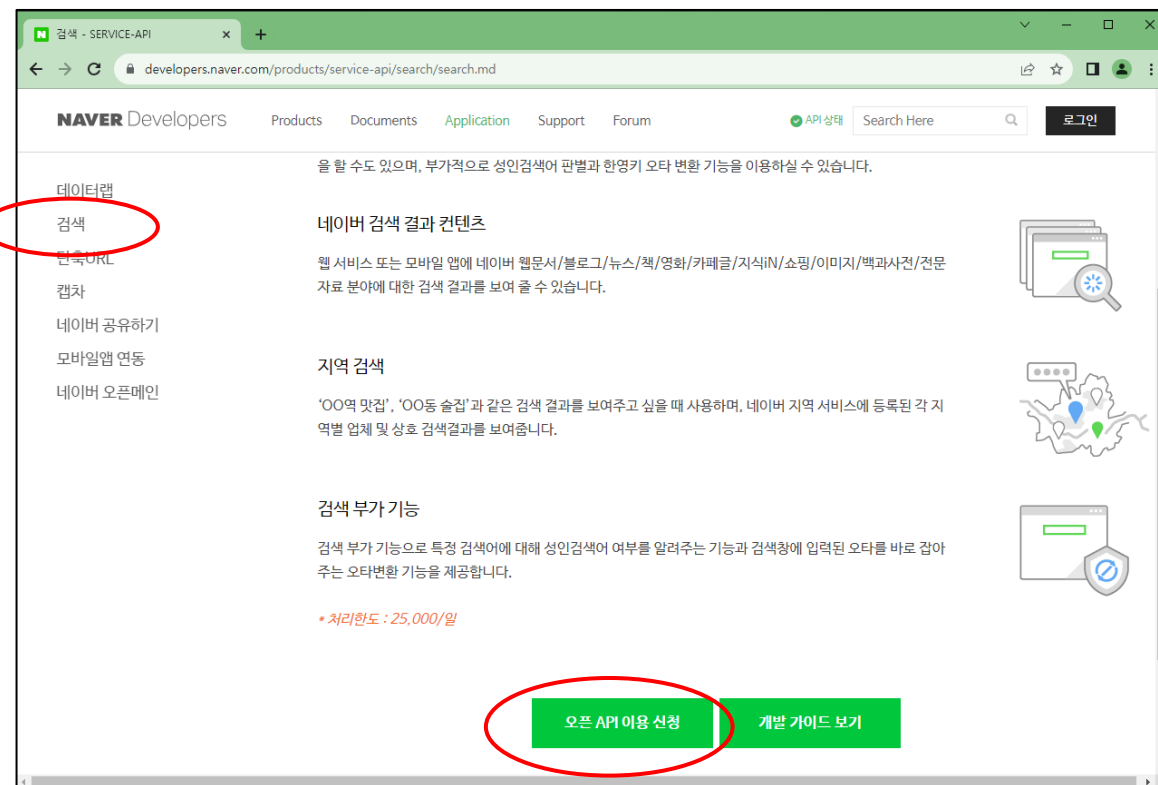
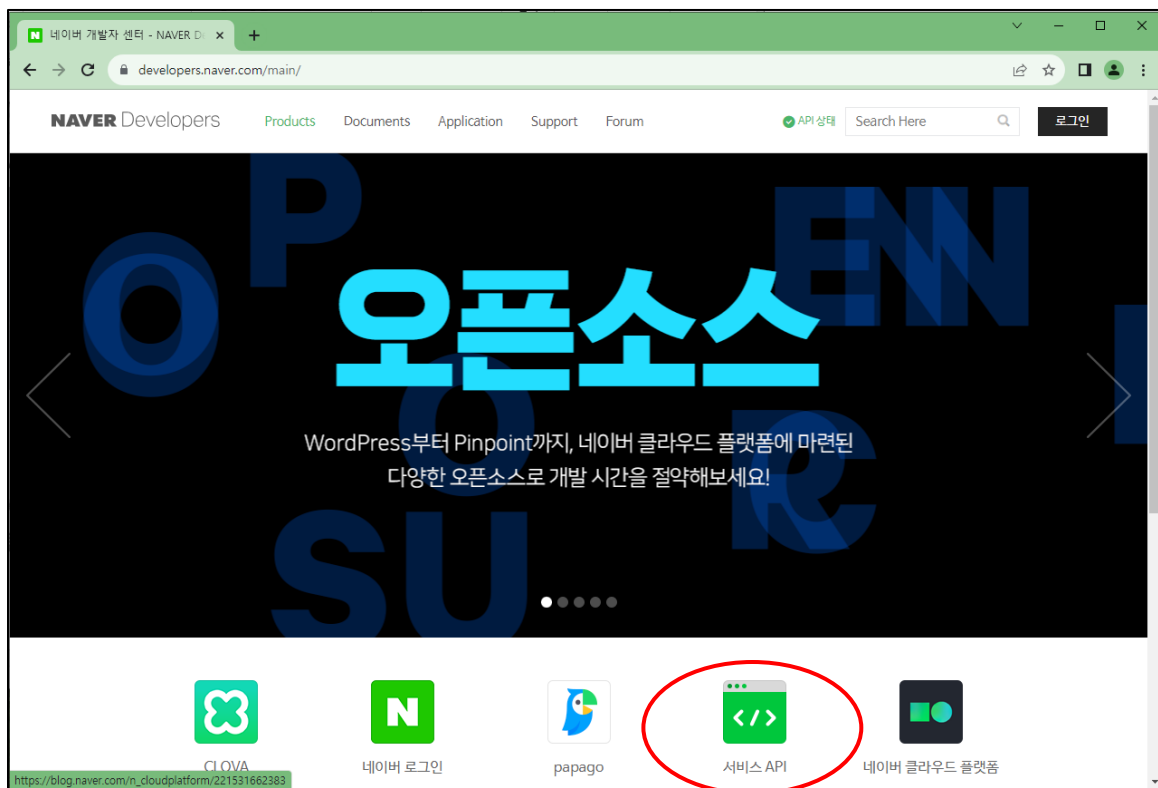
종류	주소
네이버 개발자 센터	<a href="https://developers.naver.com">https://developers.naver.com</a>
카카오 앱 개발 플랫폼 서비스	<a href="https://developers.kakao.com">https://developers.kakao.com</a>
페이스북 개발자 센터	<a href="https://developers.facebook.com">https://developers.facebook.com</a>
트위터 개발자 센터	<a href="https://developer.twitter.com">https://developer.twitter.com</a>

종류	주소
공공데이터포털	<a href="https://www.data.go.kr">https://www.data.go.kr</a>
세계 날씨	<a href="http://openweathermap.org">http://openweathermap.org</a>
유료/무료 API 스토어	<a href="http://mashup.or.kr">http://mashup.or.kr</a> <a href="http://www.apistore.co.kr/api/apiList.do">http://www.apistore.co.kr/api/apiList.do</a>

## 2. 네이버 API

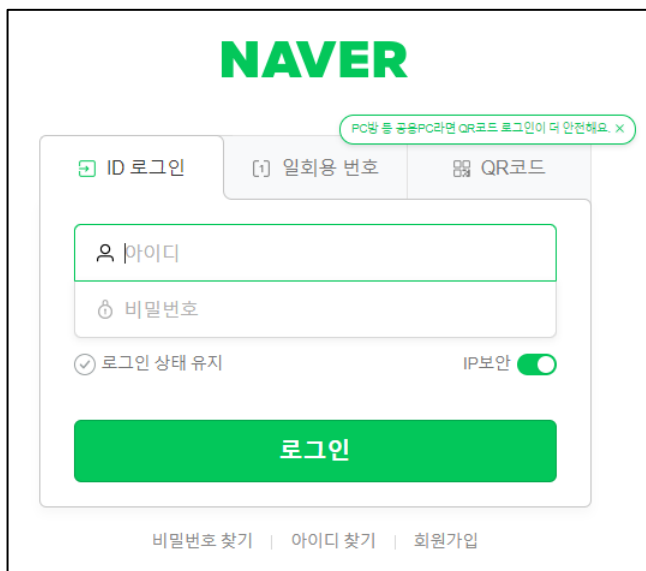
### ⌚ 가입

<https://developers.naver.com/main/>



## 2. 네이버 API

### ⌚ 가입



The image shows the Naver login page. At the top is the Naver logo. Below it are three tabs: 'ID 로그인' (selected), '일회용 번호', and 'QR코드'. Under 'ID 로그인', there are input fields for '아이디' (ID) and '비밀번호' (Password). There are checkboxes for '로그인 상태 유지' (Keep login state) and 'IP보안' (IP security, which is turned on). A large green '로그인' (Login) button is at the bottom. At the very bottom, there are links for '비밀번호 찾기' (Find password), '아이디 찾기' (Find ID), and '회원가입' (Sign up).

네이버 로그인  
가입이 안되어 있으면,  
<회원가입>을 함



The image shows the Naver Developers registration page. The header says 'NAVER Developers'. Below the header, there's a message: '애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서버 메뉴에 등록하신 애플리케이션 이름으로 서버 메뉴가 만들어집니다.' (If you register the basic information of the application, the server menu in the left 'My Application' menu will be created with the application name you registered.)

The main form has three sections:

- 애플리케이션 이름** (Application Name): A text input field containing 'myapp' with a green checkmark. Below it, there are instructions: '네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.' and '40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), "/" , "-" , "\_" 만 입력 가능합니다.'
- 사용 API** (Used API): A dropdown menu with '선택하세요.' (Select) and a search bar with a '검색' (Search) button highlighted by a red box.
- 환경 추가** (Add Environment): A dropdown menu with '환경 추가' (Add Environment) and a 'WEB 설정' (WEB Settings) button. Below this, there's a section for '웹 서비스 URL (최대 10개)' (Web service URL (up to 10)). A text input field contains 'http://localhost' with a green checkmark.

At the bottom, there are two buttons: '등록하기' (Register) and '취소' (Cancel). The '등록하기' button is highlighted with a red circle.

환경추가 - WEB설정

http://localhost



The image shows two buttons: a green '등록하기' (Register) button and a grey '취소' (Cancel) button. The '등록하기' button is circled in red.

## 2. 네이버 API

### ⌚ 가입

Application

myapp

개요 API 설정 멤버관리 로그인 통계

애플리케이션 정보

Client ID GG0sDqt6L7om\_xfa3WrR

Client Secret 보기

<보기>를 클릭한 다음,  
ID 와 Secret 은 메모장으로 복사해 둬

### API 호출 안내

지도 API 인증실때나 네이버 로그인 이용 제한이 걸렸다면 [API 설정] 탭에서 URL 관련 설정을 수정하시면 정상 이용 가능합니다 !!!

### 비로그인 오픈 API 당일 사용량

API호출량/일일허용량

검색

0/25000

하루에 25,000번 호출이 가능하고, 늘리고 싶으면, 별도 신청

NAVER Developers

Products Documents Application Support Forum

API 공통 가이드 CLOVA

네이버 로그인 개요 검수 가이드 개발 가이드 API 명세 튜토리얼 SDK 다운로드

파파고 Papago 번역 언어 감지 한글인명-로마자 변환

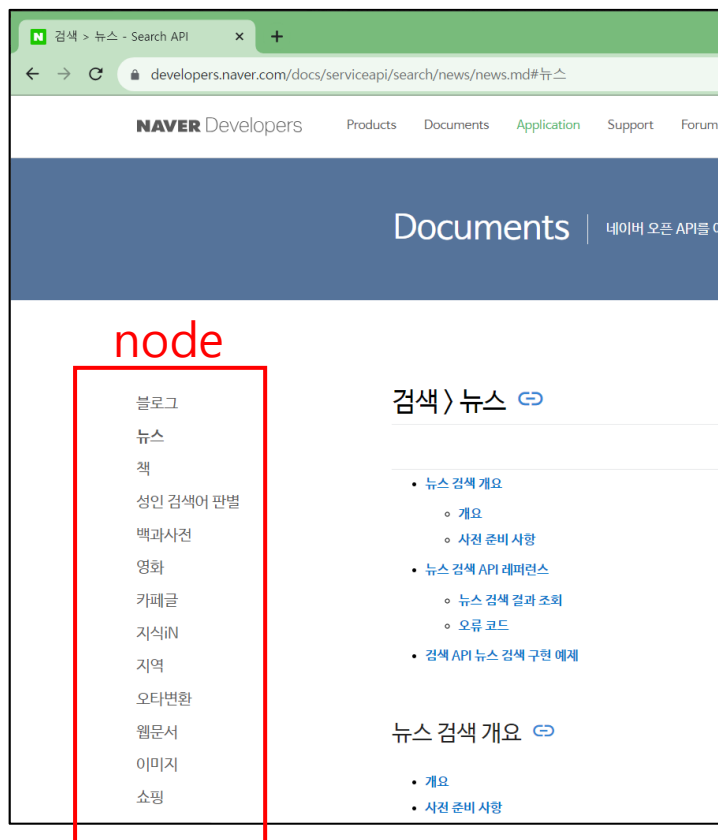
서비스 API 데이터랩 검색 단축 URL 이미지캡차 음성캡차 네이버 공유하기 네이버 앱 연동 네이버 오픈메인

도움이 필요할 땐, <Documents> -<서비스API> -<검색> 이용

## 2. 네이버 API

### ⌚ 요청과 응답

<https://developers.naver.com/docs/serviceapi/search/news/news.md#뉴스>



구분	내용 및 설명	
URL	뉴스	<a href="https://openapi.naver.com/v1/search/news.json">https://openapi.naver.com/v1/search/news.json</a>
	블로그	<a href="https://openapi.naver.com/v1/search/blog.json">https://openapi.naver.com/v1/search/blog.json</a>
	카페	<a href="https://openapi.naver.com/v1/search/cafearticle.json">https://openapi.naver.com/v1/search/cafearticle.json</a>
	영화	<a href="https://openapi.naver.com/v1/search/movie.json">https://openapi.naver.com/v1/search/movie.json</a>
	쇼핑	<a href="https://openapi.naver.com/v1/search/shop.json">https://openapi.naver.com/v1/search/shop.json</a>

데이터 요청 주소

## 2. 네이버 API

### ⌚ 요청과 응답

#### news 의 요청과 응답

요청 변수	query	검색을 원하는 문자열이며 UTF-8로 인코딩한다.
	start	검색 시작 위치로 최대 1000까지 가능하다. 1(기본값)~1000(최대값)
	display	검색 결과 출력 건수를 지정한다. 10(기본값)~100(최대값)
응답 변수	items	검색 결과로 title, originallink, link, description, pubDate를 포함한다.
	title	검색 결과 문서의 제목이다.
	link	검색 결과 문서를 제공하는 네이버의 하이퍼텍스트 link다.
	originallink	검색 결과 문서를 제공하는 언론사의 하이퍼텍스트 link다.
	description	검색 결과 문서의 내용을 요약한 정보다.
	pubDate	검색 결과 문서가 네이버에 제공된 시간이다.



## 2. 네이버 API

### ⌚ 요청과 응답

#### book 의 요청과 응답

파라미터	타입	필수 여부	설명
query	String	Y	검색어. UTF-8로 인코딩되어야 합니다.
display	Integer	N	한 번에 표시할 검색 결과 개수(기본값: 10, 최대값: 100)
start	Integer	N	검색 시작 위치(기본값: 1, 최대값: 1000)

요소	타입	설명
rss/channel/start	Integer	검색 시작 위치
rss/channel/display	Integer	한 번에 표시할 검색 결과 개수
rss/channel/item	-	개별 검색 결과. JSON 형식의 결과값에서는 <code>items</code> 속성의 JSON 배열로 개별 검색 결과를 반환합니다.
rss/channel/item/title	String	책 제목
rss/channel/item/link	String	네이버 도서 정보 URL
rss/channel/item/image	String	섬네일 이미지의 URL
rss/channel/item/author	String	저자 이름
rss/channel/item/discount	Integer	판매 가격. 절판 등의 이유로 가격이 없으면 값을 반환하지 않습니다.
rss/channel/item/publisher	String	출판사

## 2. 네이버 API

### ⌚ 요청과 응답

#### movie 의 요청과 응답

파라미터	타입	필수 여부	설명
query	String (필수)	Y	검색어. UTF-8로 인코딩되어야 합니다.
display	Integer	N	한 번에 표시할 검색 결과 개수(기본값: 10, 최대값: 100)
start	Integer	N	검색 시작 위치(기본값: 1, 최대값: 1000)
genre	String	N	<ul style="list-style-type: none"> <li>- 13 : 미스터리</li> <li>- 14 : 전쟁</li> <li>- 15 : 애니메이션</li> </ul>
country	String	N	<ul style="list-style-type: none"> <li>- GB : 영국</li> <li>- HK : 홍콩</li> <li>- JP : 일본</li> </ul>

요소	타입	설명
rss/channel/total	Integer	총 검색 결과 개수
rss/channel/start	Integer	검색 시작 위치
rss/channel/display	Integer	한 번에 표시할 검색 결과 개수
rss/channel/item	-	개별 검색 결과. JSON 형식의 결과값에서는 items 속성의 JSON 배열로 개별 검색 결과를 반환합니다.
rss/channel/item/title	String	영화 제목. 제목에서 검색어와 일치하는 부분은 <b> 태그로 감싸져 있습니다.
rss/channel/item/link	String	네이버 영화 정보 URL
rss/channel/item/image	String	썸네일 이미지의 URL
rss/channel/item/subtitle	String	영어 제목 또는 원제
rss/channel/item/pubDate	Date	제작 연도 (yyyy 형식)
rss/channel/item/director	String	감독

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 전체 작업 설계

작업 설계	사용할 코드
1. 검색어 지정하기	srcText = '월드컵'
2. 네이버 뉴스 검색하기	<b>getNaverSearch()</b>
2.1 url 구성하기	url = base + node + srcText
2.2 url 접속과 검색 요청하기	urllib.request.urlopen()
2.3 요청 결과를 응답 JSON으로 받기	json.load()
3. 응답 데이터를 정리하여 리스트에 저장하기	<b>getPostData()</b>
4. 리스트를 JSON 파일로 저장하기	json.dumps()

cmain.py

CODE2.py

CODE1.py

CODE3.py

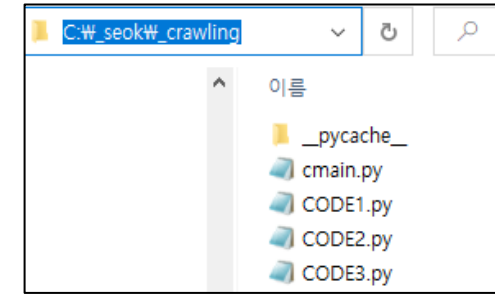
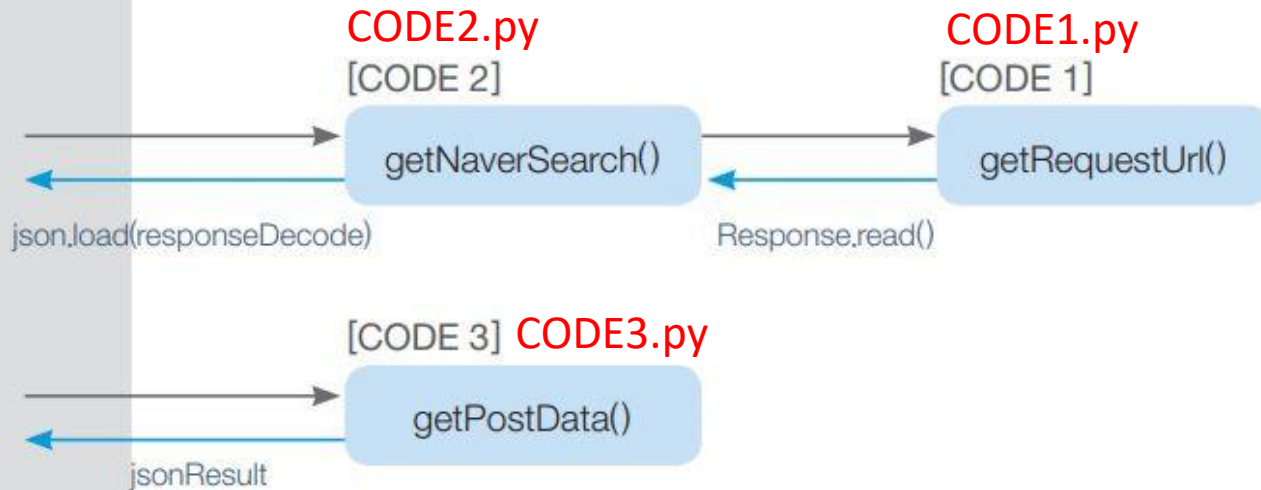
## 2. 네이버 API

### ⌚ 뉴스 크롤링

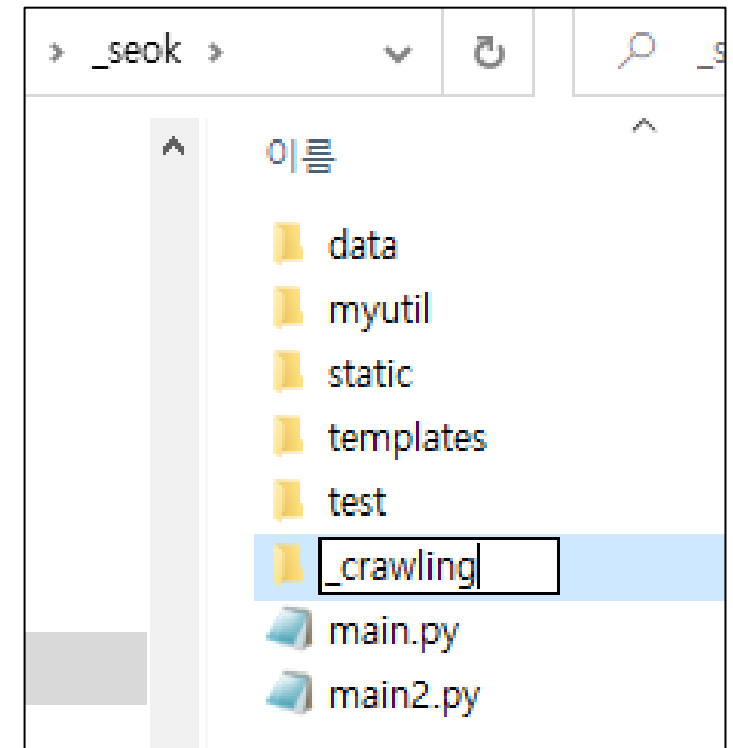
#### ■ 프로그램 구성 설계

[CODE 0] **cmain.py**

```
def main()
1. 검색어 지정
2. 네이버 뉴스 검색
3. 응답 데이터 정리 후
   리스트에 저장
4. 리스트를 JSON 파일로 저장
```



자신의 홈 디렉터리에



**\_crawling 디렉터리 만들기**

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 메인 프로그램 : cmain.py

```

4칸 8칸 12칸
#=====
# main program
#=====
import json
import pandas as pd
from CODE2 import getNaverSearch
from CODE3 import getPostData

def main():
    MAX_CNT = 50          #검색해 올 총 개수
    DISPLAY_CNT = 10      #한번에 표시할 개수
    MAX_START_POINT = 1000 #MAX 검색시작위치
    node = 'news'         #크롤링할 대상
    srcText = input('검색어를 입력하세요: ')
    #srcText = '중국음식'

    cnt = 0
    jsonResult = []
    
```

#### 주석 - 설명문

**getNaverSearch**: 1부터 100개의 검색 결과를 처리한다

**getPostData**: 검색 결과 한 개를 처리한다

**main()**: 메인함수

**MAX\_CNT**: 검색해 올 총 개수

**DISPLAY\_CNT**: 한번에 표시할 개수

**MAX\_START\_POINT = 1000**: MAX 검색시작위치

**node**: 네이버 검색 API에서 검색할 대상 노드

**srcText**: 사용자 입력으로 받은 검색어 저장

**input**: 사용자로부터 입력을 받는다.

**cnt**: 검색 결과 카운트

**jsonResult**: 검색 결과를 정리하여 저장할 리스트 객체

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 메인 프로그램 : cmain.py

4칸 8칸 12칸

```
jsonResponse = getNaverSearch(node, srcText, 1, DISPLAY_CNT) #[CODE 2]
total = jsonResponse['total']

while ((jsonResponse != None) and (jsonResponse['display'] != 0)):
    for post in jsonResponse['items']:
        cnt += 1
        getPostData(post, jsonResult, cnt)                               #[CODE 3]

    start = jsonResponse['start'] + jsonResponse['display']
    if start > MAX_START_POINT: start = 1
    if start > MAX_CNT: break
    jsonResponse = getNaverSearch(node, srcText, start, DISPLAY_CNT)

print('전체 검색 : %d 건' %total)

file_name = ("naver_%s_%s(%05d)" % (node, srcText, cnt))
```

**jsonResponse**: 네이버 뉴스검색에 대한  
응답을 저장하는 객체

**total**: 전체 검색 결과 개수

**post**: 응답받은 검색 결과 중에서 한  
개를 저장한 객체

**items**: 전체 응답 검색 결과

**file\_name**: 저장할 파일 이름

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 메인 프로그램 : cmain.py

4칸 8칸 12칸

```
with open(file_name + '.json', 'w', encoding='utf8') as outfile:
    jsonFile = json.dumps(jsonResult, indent = 4, ensure_ascii = False)
    outfile.write(jsonFile)

out_df = pd.DataFrame(jsonResult)
out_df.to_excel(file_name + '.xlsx', index=False)

print("가져온 데이터 : %d 건" %(cnt))
print('%s SAVED' % (file_name))

if __name__ == '__main__':
    main()
```

**json.dumps():** 객체를 JSON 형식으로 변환

**jsonFile:** JSON 파일에 저장할 데이터를 담은 객체

**out\_df:** 수집된 데이터 객체인 jsonResult 를 판다스 데이터프레임으로 변환

파이썬 최초로 시작되는 위치

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 네이버주소 호출 프로그램 : CODE2.py

```

4칸 8칸 12칸
#=====
# CODE2 program
#=====
import urllib.request
import json
from CODE1 import getRequestUrl

def getNaverSearch(node, srcText, start, display):
    base = "https://openapi.naver.com/v1/search"
    node = "/%s.json" % node
    parameters = ("?query=%s&start=%s&display=%s"
                  % (urllib.parse.quote(srcText), start, display))

    url = base + node + parameters
    #print('url = %s' % url)
    responseDecode = getRequestUrl(url)          #[CODE 1]
    #print('responseDecode = %s' % responseDecode)

```

주석 - 설명문

**getRequestUrl**: 호출하여 반환받은  
응답 데이터를 파이썬  
json 형식으로 반환

**node**: 네이버 검색 API를 이용하여 검색할  
대상 노드

**srcText**: 검색어

**page\_start**: 검색 시작 위치(1~1000)

**display**: 출력 건수(10~100)

**base**: 검색 url의 기본 주소

**parameter**: url에 추가할 검색어와 검색 시작  
위치, 출력 건수 등의 매개변수



## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 네이버주소 호출 프로그램 : CODE2.py

4칸 8칸 12칸

```
if (responseDecode == None):
    return None
else:
    return json.loads(responseDecode)
```

**responseDecode**: getRequestId(url)을 호출하여 반환받은 응답 객체(utf-8로 디코드)

**json.loads(responseDecode)**: 응답 객체를 파이썬이 처리할 수 있는 JSON 형식으로 변환

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 데이터 요청/응답 프로그램 : CODE1.py

```

4칸 8칸 12칸
#=====
# CODE1 program
#=====
import urllib.request
from datetime import datetime
client_id = '본인이 발급받은 ID'
client_secret = '본인이 발급받은 secret 코드'

def getRequestUrl(url):
    req = urllib.request.Request(url)
    req.add_header("X-Naver-Client-Id", client_id)
    req.add_header("X-Naver-Client-Secret", client_secret)

    print('----call url = ', url)

```

주석 - 설명문

**client\_id**: 네이버 개발자로 등록하고 받은 Client ID

**client\_secret**: 네이버 개발자로 등록하고 받은 Client Secret

**url**: 네이버 뉴스 검색('월드컵')에 대한 url

**req**: url 접속 요청(request) 객체

**urllib.request.Request()**:

urllib 패키지의 request 모듈에 있는 Request() 함수로 네이버 서버에 보낼 요청(request) 객체를 생성

**Request.add\_header()**: 서버에 보내는 요청 객체에 헤더 정보를 추가

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 데이터 요청/응답 프로그램 : CODE1.py

```

4칸 8칸 12칸
try:
    response = urllib.request.urlopen(req)
    if response.getcode() == 200:
        print("----[%s] Url Request Success" % datetime.now())
        return response.read().decode('utf-8')
except Exception as e:
    print(e)
    print("[%s] Error for URL : %s" % (datetime.now(), url))
    return None

```

**response**: 네이버 서버에서 받은 응답을 저장하는 객체

**urllib.request.urlopen()**: 서버에서 받은 응답을 변수에 저장하기 위해 메모리로 가져오는 urllib 패키지의 request 모듈에 있는 함수

**response.getcode()**: 요청 처리에 대한 응답 상태를 확인하는 response 객체의 멤버 함수로 상태 코드가 200이면 요청 처리 성공을 나타냄

**datetime.now()**: 현재 시간을 구하는 함수

**response.read().decode()**: utf-8 형식으로 문자열을 디코딩

## 2. 네이버 API

### ⌚ 뉴스 크롤링

- 응답을 변수에 저장하는 프로그램 : CODE3.py

4칸 8칸 12칸

```
#=====
# CODE3 program
#=====
from datetime import datetime

def getPostData(post, jsonResult, cnt):
    title = post['title']
    description = post['description']
    org_link = post['originallink']
    link = post['link']
```

주석 - 설명문

**post**: 응답으로 받은 검색 결과 데이터 중에서 결과 한 개를 저장한 객체

**jsonResult**: 필요한 부분만 저장하여 반환할 리스트 객체

**cnt**: 현재 작업 중인 검색 결과의 번호

**post['title']**: post 객체의 title 항목에 저장된 값

**post['description']**: post 객체의 description 항목에 저장된 값

**post['originallink']**: post 객체의 originallink 항목에 저장된 값

**post['link']**: post 객체의 link 항목에 저장된 값

## 2. 네이버 API

### ⌚ 뉴스 크롤링

- 응답을 변수에 저장하는 프로그램 : CODE3.py

4칸 8칸 12칸

```
pDate = datetime.strptime(post['pubDate'],
                             '%a, %d %b %Y %H:%M:%S +0900')
pDate = pDate.strftime('%Y-%m-%d %H:%M:%S')

jsonResult.append({'cnt':cnt, 'title':title,
                   'description': description,
                   'org_link':org_link, 'link': org_link,
                   'pDate':pDate})

return
```

**datetime.datetime.strptime()**: 문자열을 날짜 객체  
형식으로 변환

네이버에서 제공하는 시간인 pubDate는  
문자열 형태이므로 날짜 객체로 변환  
pubDate는 그리니치 평균시 형식을  
사용하는데 한국 표준시보다 9시간 느리므로  
+0900 을 사용해 한국 표준시로 맞춤

**pDate.strftime()**: 날짜 객체의 표시 형식을 지정  
수정된 날짜를 '년-월-일 시:분:초' 형식으로  
나타냄

**jsonResult.append()**: 리스트 객체인 jsonResult에  
원소를 추가

## 2. 네이버 API

### ⌚ 뉴스 크롤링

#### ■ 실행

```

C:\> 명령 프롬프트
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\> cd W_seok

C:\W_seok> cd _crawling

C:\W_seok\W_crawling> python cmain.py
검색어를 입력하세요: BTS
----call url = https://openapi.naver.com/v1/search/news.json?query=BTS&
----[2023-10-15 17:27:54.046335] Url Request Success
----call url = https://openapi.naver.com/v1/search/news.json?query=BTS&
----[2023-10-15 17:27:54.140085] Url Request Success
----call url = https://openapi.naver.com/v1/search/news.json?query=BTS&
----[2023-10-15 17:27:54.249461] Url Request Success
----call url = https://openapi.naver.com/v1/search/news.json?query=BTS&
----[2023-10-15 17:27:54.343211] Url Request Success
----call url = https://openapi.naver.com/v1/search/news.json?query=BTS&
----[2023-10-15 17:27:54.436963] Url Request Success
전체 검색 : 436273 건
가져온 데이터 : 50 건
naver_news_BTS(00050) SAVED

C:\W_seok\W_crawling>

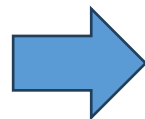
```

cd W\_seok

cd \_crawling

python cmain.py

파일이 생성 됨



naver\_news\_BTS(00050).json



naver\_news\_BTS(00050).xlsx

## 2. 네이버 API

### ⌚ 뉴스 크롤링


#### ■ 확인


메모장이나  
vsc 로 확인

```

naver_news_BTS(00050).json - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

[
{
  "cnt": 1,
  "title": "\"누님들 감사합니다\" 한덕수 총리에 도착한 '깜짝선물' 정체",
  "description": "그러면서 \"어르신들은 이번 엑스포 응원을 만드느라 노랏말도 직접 쓰시고, 뮤직비디오도 하셨다고 한다\"며 \"<b>BTS</b>와 블랙핑크만 엑스포를 홍보할 수 있는 게 아니면서 수니와...
  "org_link": "https://www.seoul.co.kr/news/newsView.php?id=20231015500088&wlog_tag3=naver",
  "link": "https://www.seoul.co.kr/news/newsView.php?id=20231015500088&wlog_tag3=naver",
  "pDate": "2023-10-15 17:23:00"
},
]
    
```


naver\_news\_BTS(00050).json


naver\_news\_BTS(00050).xlsx

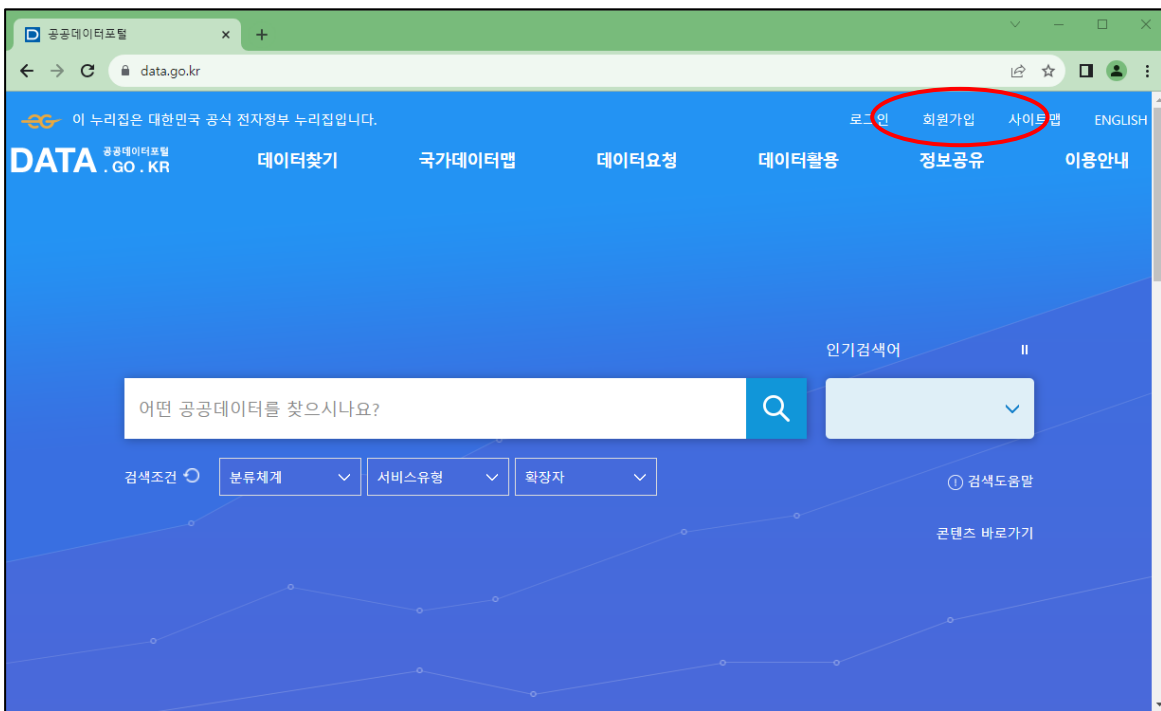
excel 로 확인

	A	B	C	D	E	F
	cnt	title	description	org_link	link	pDate
1			그러면서 "어르신들은 이번 엑스포 응원을 만드느라 노랏말도 직접 쓰시고, 뮤직비디오 찍기 전에 일주일간 안무 연습도 하셨다고 한다"며 "<b>BTS</b>와 블랙핑크만 엑스포를 홍보할 수 있는 게 아니면서 수니와...	https://www.seoul.co.kr/news/newsView.php?id=20231015500088&wlog_tag3=naver	https://www.seoul.co.kr/news/newsView.php?id=20231015500088&wlog_tag3=naver	2023-10-15 17:23:00
2		"누님들 감사합니다" 한덕수 총리에 도착한 '깜짝선물' 정체	사진=수원시청 '방탄소년단(<b>BTS</b>) 뷔'와 '생태교통 수원 뉴페스타 홍보 그림'이 행궁동 '화성사업소'에 벽화... 이 작가는 도산 안창호, 안중근 의사, <b>BTS</b> 등의 그래피티를 국내와 해외에 선보여 대중에게 알려졌다. 이재준...	http://www.joongboo.com/news/articleView.html?idxno=363615554	http://www.joongboo.com/news/articleView.html?idxno=363615554	2023-10-15 17:14:00
3		행궁동 화성사업소에 '<b>BTS</b>' '생태교통 수원 뉴페스타' 벽화 제작	이날 촬영은 <b>BTS</b> RM, 이효리, 조인성, 배우나, 공효진 등 톱스타들과 작업했던 유명 포토그래퍼와 함께해 눈길을 끌었다. 촬영이 시작되자 흥현희와 준범이는 금세 눈빛이 돌변, 웃음기를 속 빼고 치명적인 매력을 발산했다....	https://www.newsen.com/news_view.php?uid=202310151649596710	https://www.newsen.com/news_view.php?uid=202310151649596710	2023-10-15 16:53:00
4		'전참시' 이국주 송국영고 이효리 4 중가 영상 공개..최고 시청률 4.8%				

# 3. 공공데이터 API

## ⌚ 가입

<https://www.data.go.kr/>



사이트	설명
<a href="http://data.go.kr">http://data.go.kr</a>	한국 정부에서 제공하는 공공데이터
<a href="http://kostat.go.kr">http://kostat.go.kr</a>	한국 통계청에서 공개하는 데이터
<a href="http://opendata.hira.or.kr">http://opendata.hira.or.kr</a>	한국 보건 의료 빅데이터 개방 시스템
<a href="http://www.localdata.kr">http://www.localdata.kr</a>	한국 지방행정 인허가 데이터
<a href="https://www.mcst.go.kr">https://www.mcst.go.kr</a>	한국 문화체육관광부 문화 데이터

사이트	설명
<a href="http://data.seoul.go.kr">http://data.seoul.go.kr</a>	서울시 열린데이터 광장
<a href="https://data.gg.go.kr">https://data.gg.go.kr</a>	경기도 공공데이터 개방 포털
<a href="http://data.gov">http://data.gov</a>	미국 정부의 공공데이터
<a href="http://data.worldbank.org">http://data.worldbank.org</a>	세계 은행에서 제공하는 개방 데이터
<a href="http://open.fda.gov">http://open.fda.gov</a>	미국 식약청의 개방 데이터

데이터과학 → 데이터수집

일반회원

만 14세 이상 내국인

회원가입 여부 확인

이름

정석일

① 실명으로 가입하지 않은 경우 공공데이터 제공신청, 분쟁조정신청 등 일부 서비스에 대해 불이익이 있을 수 있습니다.

이메일

dossacc@naver.com

① 이메일 주소 입력은 영문키 상태에서 기재해주시기 바랍니다.

가입확인



# 3. 공공데이터 API

## ⌚ 가입

개인정보 수집 및 이용 ☒ 동의

제1조. 개인정보의 처리목적

④ 행정안전부가 개인정보 보호법 제32조에 따라 등록·공개하는 개인정보파일의 처리목적은 다음과 같습니다.

1. 개인정보 파일명 : 공공데이터포털회원정보

가. 개인정보의 처리목적 : 공공데이터포털 이용 및 공공데이터 활용 관리, 공공데이터 정책 반영, 공공데이터 제공 운영실태 평가 수행 등의 목적

나. 수집방법 : 공공데이터포털 누리집

공공데이터포털 서비스 이용약관과 개인정보 수집 및 이용에 동의하지 않을 시에 회원가입이 되지 않습니다.

☒ 전체 약관에 동의합니다.

이전 단계 다음 단계

기본정보 입력

아이디  중복확인

비밀번호

비밀번호 확인

개인정보 입력

이름

이메일  이메일인증

☒ 이메일 수신 동의

① 수신거부 시 일부 서비스에 제한이 있을 수 있습니다. 비밀번호 초기화 메일 수신 등에 반드시 필요한 정보이므로 정확히 이메일 주소를 입력해주세요.

이메일 인증번호  인증확인

휴대전화번호

☒ SMS 수신 동의

① 수신거부 시 일부 서비스에 제한이 있을 수 있습니다.

이전 단계 다음 단계

인증방법 선택

☒ 휴대폰 인증  
본인 명의의 휴대폰으로 인증번호를 받은 후 가입하실 수 있습니다.

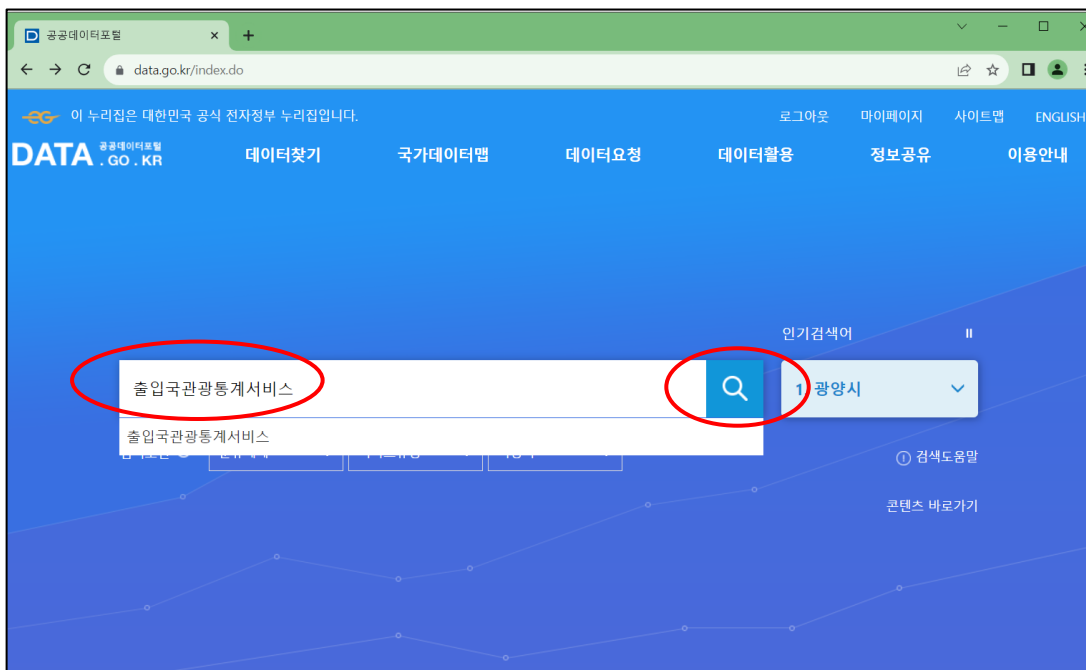
☐ 아이핀(I-PIN) 인증  
본인 명의의 아이핀(I-PIN) 계정으로 인증 후 가입하실 수 있습니다.

● 본인인증 필요  
본인인증이 필요한 서비스입니다.

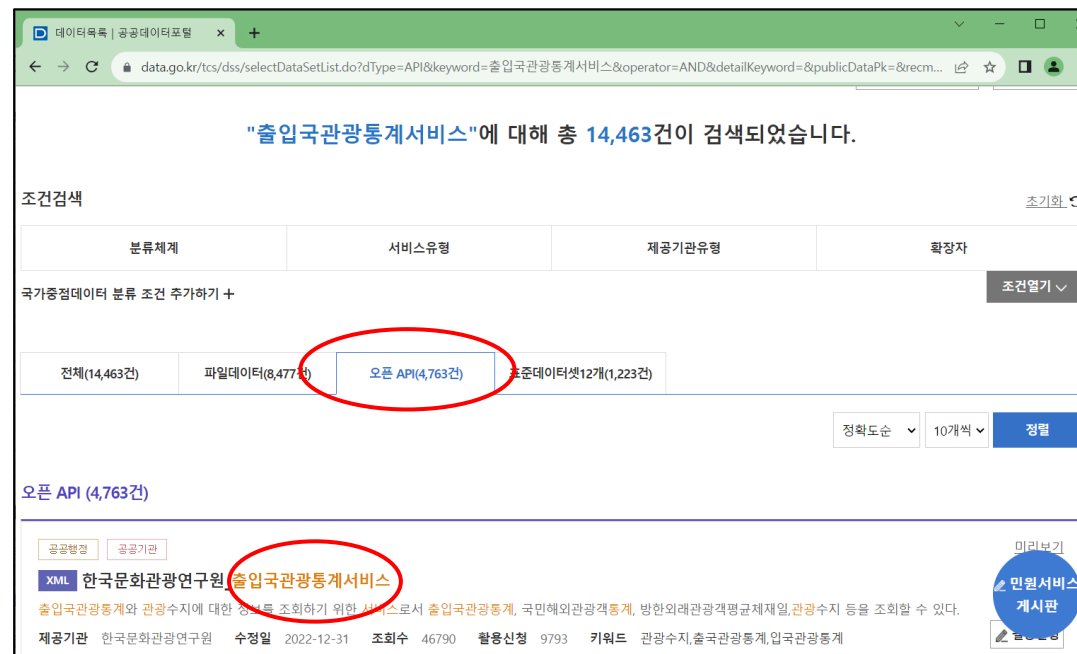
이전 단계 가입 완료

# 3. 공공데이터 API

## ⌚ 검색



출입국관광통계서비스 를 입력하고 검색



# 3. 공공데이터 API

## ⌚ 활용신청

www.data.go.kr 내용:  
신청이 완료되었습니다. 신청된 API는 1~2시간 후 호출 가능합니다.  
※ 일부 API에서 24시간 후 인증이 가능한 API가 있습니다.  
빠른 시일내에 모든 API가 1시간 이내 인증 가능하도록 개선하겠습니다

확인



활용목적 선택 \*표시는 필수 입력항목입니다.

\*활용목적

☐ 웹 사이트 개발
 ☐ 앱개발 (모바일,솔루션등)
 ☐ 기타
 ☐ 참고자료
 ☒ 연구(논문 등)

공공데이터 활용 학습

11/250

첨부파일 파일 선택

공공데이터 활용 학습 이라고 입력

상세기능정보 선택

<input type="checkbox"/>	상세기능	설명	일일 트래픽
<input checked="" type="checkbox"/>	출입국관광통계조회	기간, 국가의 검색조건에 따라 관광출입국자수를 제공하는 기능	1000
<input type="checkbox"/>	국민해외관광객통계조회	월별, 성별, 연령대, 출국항의 검색 조건에 따라 국민해외관광객수를 제공하는 기능	1000
<input type="checkbox"/>	방한외래관광객세부통계조회	월별, 국적, 성별, 연령대, 여행목적, 입국항의 검색 조건에 따라 방한외래관광객수를 제공하는 기능	1000
<input type="checkbox"/>	방한외래관광객평균체재일조회	월별, 국적의 검색 조건에 따라 방한외래관광객 평균체재일수를 제공하는 기능	1000
<input type="checkbox"/>	관광수지조회	월별 관광 수입과 지출, 1인당 평균 소비액을 제공하는 기능	1000

라이선스 표시

\*이용허락범위

저작자표시 ☒ 동의합니다.

취소 활용신청

# 3. 공공데이터 API

## ⌚ 인증키 확인

The screenshot shows the '마이페이지' (My Page) section of the data.go.kr portal. The '마이페이지' link in the top navigation bar is circled in red. On the left sidebar, '마이페이지' is selected. The main content area shows a welcome message and statistics for file data, API requests, and interest data. Below this, the '개인 API인증키' (Personal API Authentication Key) section is expanded, showing a table with one entry. The entry's '구분' (Category) is '일반' (General), '발급일자' (Issuance Date) is '2023-10-15', and '인증키' (Authentication Key) is a long alphanumeric string. The '인증키' column has two buttons: '인증키 복사(Encoding)' and '인증키 복사(Decoding)', both of which are circled in red. At the bottom, there is an 'API공지' (API Notice) section with a '더보기 >' link.

구분	발급일자	인증키
일반	2023-10-15	dO62WvSYyf5rFu+W+oD6FDfQ2GRN0dfCHn97fzgb... <div>인증키 복사(Encoding)</div> <div>인증키 복사(Decoding)</div>

메모장에 복사해 둬

# 3. 공공데이터 API

## ⌚ 서비스 확인

이 누리집은 대한민국 공식 전자정부 누리집입니다.

로그아웃 마이페이지 사이트맵 ENGLISH

DATA GO . KR 데이터찾기 국가데이터맵 데이터요청 데이터활용 정보공유 이용안내

**마이페이지**

- 데이터 활용 >
- 데이터 요청 >
- 나의 문의 >
- 회원정보 수정 >

정석일님, 반갑습니다.  
회원정보 수정 >

파일데이터 0건 API신청 1건 관심데이터 0건

개인 API인증키 더보기 >

구분	발급일자	인증키
일반	2023-10-15	d062Wv5Yyf5rFu+W+oD6FDFq2GRN0dfCHn97fzgb... 인증키 복사(Encoding) 인증키 복사(Decoding)

API공지 더보기 >

API신청 확인

# 3. 공공데이터 API

## ⌚ 서비스 확인

### 개발계정 상세보기

#### 기본정보

데이터명	한국문화관광연구원_출입국관광통계서비스	상세설명	
서비스유형	REST	심의여부	자동승인
신청유형	개발계정   활용신청	처리상태	승인
활용기간	2023-10-15 ~ 2025-10-15		

승인이 된 것 확인하고 <상세설명> 클릭

# 3. 공공데이터 API

## ⌚ 요청과 응답

요청주소 : <http://openapi.tour.go.kr/openapi/service/EdrcntTourismStatsService/getEdrcntTourismStatsList>

요청변수(Request Parameter)					
항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
연월	YM	12	필수	201201	연월
국가코드	NAT_CD	6	옵션	112	국가코드
출입국구분코드	ED_CD	2	옵션	E	출입국구분코드

# 3. 공공데이터 API

## ⌚ 요청과 응답

출력결과(Response Element)					
항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	4	필수	0000	결과코드
결과메시지	resultMsg	50	필수	OK	결과메시지
한 페이지 결과 수	numOfRows	2	옵션	10	한 페이지 결과 수
페이지 번호	pageNo	5	옵션	1	페이지 번호
전체 결과 수	totalCount	7	옵션	12334	전체 결과 수
출입국 구분	ed	14	필수	방한외래관광객	출입국구분
출입국 구분코드	edCd	2	필수	E	출입국 구분코드
국가코드	natCd	6	필수	112	국가코드
국가	natKorNm	80	필수	중국	국가
출입국자수	num	10	옵션	179508	출입국자수
연월	ym	12	필수	201206	연월
결과값 연번	rnum	2	필수	1	결과값 나열순서



# 3. 공공데이터 API

## ⌚ 전체 작업 설계

작업 설계	사용할 코드
1. 데이터를 수집할 국가코드와 연도 입력하기	national_code, nStartYear, nEndYear
2. 데이터 수집 요청하기	<b>getTourismStatsService()</b>
2.1 url 구성하여 데이터 요청하기	<b>getTourismStatsItem()</b>
2.2 url 접속하고 요청하기	<b>getRequestUrl()</b>
2.3 응답 데이터를 리스트로 구성하기	jsonResult, result
3. 데이터를 JSON 파일과 CSV 파일로 저장하기	json.dumps(), to_csv()

# 3. 공공데이터 API

## ⌚ 프로그램 구성 설계

gmain.py

[CODE 0]

def main()

1. 수집할 국가 코드와  
연도 입력

2. 데이터 수집 요청

데이터 리스트

3. 파일 저장

[CODE 3]

getTourismStatsService()

월 데이터 요청

json.loads()

[CODE 1]

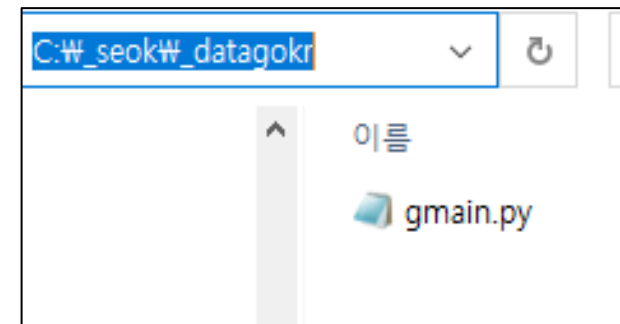
getRequestUrl()

접속할 url

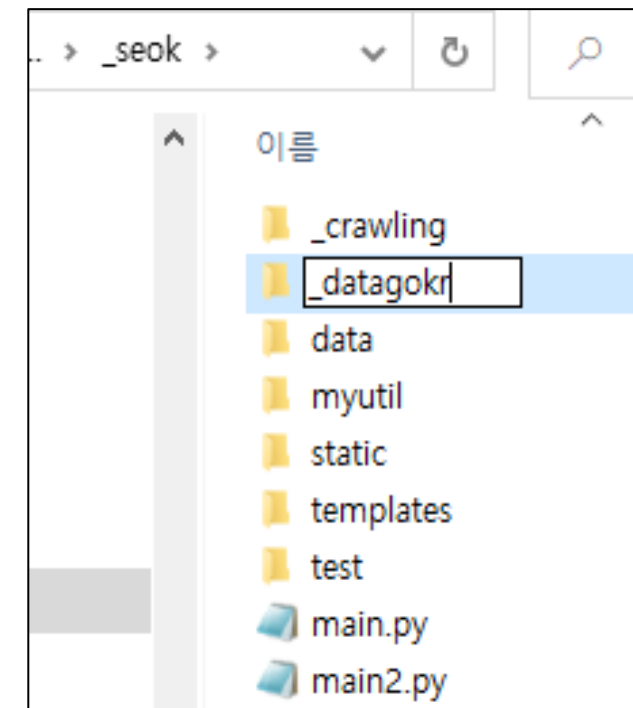
response.read()

[CODE 2]

getTourismStatItem()



자신의 홈 디렉터리에



\_datagokr 디렉터리 만들기

# 3. 공공데이터 API

## ⌚ 프로그램 : gmain.py

4칸 8칸 12칸

```
#=====
# 공공데이터 API 크롤링 프로그램
#=====
import os
import sys
import urllib.request
import datetime
import time
import json
import pandas as pd

ServiceKey = "본인이 발급받은 Encoding Key"

#[CODE 1]
def getRequestUrl(url):
    req = urllib.request.Request(url)
    try:
        response = urllib.request.urlopen(req)
```

**url**: 출입국관광통계서비스의 오픈 API를 사용하는 데이터를 요청하는 url

**req**: url 접속을 요청하는 객체

**response**: 서버에서 받은 응답을 저장하는 객체

**urllib.request.Request()**: urllib 패키지의 request 모듈에 있는 Request() 함수로 요청 객체를 생성

**urllib.request.urlopen()**: 서버에 요청을 보내고 받은 응답을 객체로 반환

# 3. 공공데이터 API

## ⌚ 프로그램

4칸 8칸 12칸

```
if response.getcode() == 200:
    print("[%s] Url Request Success" % datetime.datetime.now())
    return response.read().decode('utf-8')
except Exception as e:
    print(e)
    print("[%s] Error for URL : %s" % (datetime.datetime.now(), url))
    return None
```

**response.getcode():** 요청 처리에 대한 응답 상태를 확인하는  
response 객체의 멤버 함수, 상태 코드가 200이면 요청  
처리 성공을 나타냄  
**datetime.datetime.now():** 현재 시간을 구함  
**response.read().decode('utf-8'):** 문자열을 utf-8 형식으로 디코딩

# 3. 공공데이터 API

## ⌚ 프로그램

4칸 8칸 12칸

```
#[CODE 2]
def getTourismStatsItem(yyyymm, national_code, ed_cd):
    service_url = "http://openapi.tour.go.kr/openapi/service/EdrcntTourismStatsService/getEdrcntTourismStatsList"

    parameters = "?_type=json&serviceKey=" + ServiceKey #인증키
    parameters += "&YM=" + yyyymm
    parameters += "&NAT_CD=" + national_code
    parameters += "&ED_CD=" + ed_cd

    url = service_url + parameters
    print(url) #엑세스 거부 여부 확인용 출력
    responseDecode = getRequestUrl(url) #[CODE 1]

    if (responseDecode == None):
        return None
    else:
        return json.loads(responseDecode)
```

**yyyymm**: 수집할 연월(예: 202003)  
**nat\_cd**: 수집 대상 국가의 코드(예: 중국 = 112)  
**ed\_cd**: 수집할 데이터 종류(방한외래관광객 = "E")  
**service\_url**: 출입국관광통계서비스의  
 공공데이터에 접속할 앤드 포인트 주소  
**parameters**: url에 추가할 매개변수  
**url**: service\_url과 parameters를 연결하여 완성한 url  
**responseDecode**: getRequestUrl(url)을 호출하여  
 반환받은 응답 객체  
**getRequestUrl()**: [CODE1]을 호출하여 url 요청에  
 대한 응답 데이터를 받음  
**json.loads()**: json 형식으로 받은 응답 데이터인  
 responseDecode를 파이썬 객체로 읽음

# 3. 공공데이터 API

## ⌚ 프로그램

4칸 8칸 12칸 16칸 20칸

```
#[CODE 3]
def getTourismStatsService(nat_cd, ed_cd, nStartYear, nEndYear):
    jsonResult = []
    result = []
    natName = ''
    ed = ''
    dataEND = "{0}{1:0>2}".format(str(nEndYear), str(12))
    isDataEnd = 0
    for year in range(nStartYear, nEndYear+1):
        for month in range(1, 13):
            if(isDataEnd == 1): break
            yyyyymm = "{0}{1:0>2}".format(str(year), str(month))
            jsonData = getTourismStatsItem(yyyyymm, nat_cd, ed_cd) #[CODE 2]
            print('=====yyyyymm', yyyyymm, str(jsonData))
            if (jsonData['response']['header']['resultMsg'] == 'OK'):
                #데이터가 없는 마지막 항목인 경우 -----
                if jsonData['response']['body']['items'] == '':
                    isDataEnd = 1
                    dataEND = "{0}{1:0>2}".format(str(year), str(month-1))
                    print("데이터 없음.... \n 제공되는 통계 데이터는 %s년 %s월까지 입니다." %(str(year), str(month-1)))
                    break
```

**nat\_cd**: 수집 대상 국가의 코드(예: 중국 = 112)  
**ed\_cd**: 수집 데이터의 종류(방한외래관광객 = "E")  
**nStartYear**: 데이터 수집 시작 연도  
**nEndYear**: 데이터 수집 끝 연도  
**jsonResult**: 수집한 데이터를 JSON 저장용으로  
 구성할 딕셔너리의 리스트 객체  
**result**: 수집한 데이터를 CSV 저장용으로 구성할  
 리스트 객체  
**jsonData**: [getTourismStatsItem()]을 호출하여  
 반환받은 응답 객체  
**dataEND**: 마지막 데이터의 연월  
**isDataEnd**: 데이터의 끝 확인용 플래그  
**natName**: 수집한 국가 이름 데이터  
**num**: 수집한 방문객 수 데이터  
**ed**: 수집한 출입국 구분 데이터  
**getTourismStatsItem()**: [CODE2]를 호출하여  
 응답으로 받은 월 데이터를 반환

# 3. 공공데이터 API

## ⌚ 프로그램

`json.dumps()`: 객체를 JSON 형식으로 변환

4칸 8칸 12칸 16칸

```
#jsonData를 출력하여 확인.....
print(json.dumps(jsonData, indent = 4, sort_keys = True, ensure_ascii = False))
natName = jsonData['response']['body']['items']['item']['natKorNm']
natName = natName.replace(' ', '')
num = jsonData['response']['body']['items']['item']['num']
ed = jsonData['response']['body']['items']['item']['ed']
print('[ %s_%s : %s ]' %(natName, yyyyymm, num))
print('-----')
jsonResult.append({'nat_name': natName, 'nat_cd': nat_cd, 'yyyyymm': yyyyymm, 'visit_cnt': num})
result.append([natName, nat_cd, yyyyymm, num])
else:
    print("*****jsonData error = ", jsonData['response']['header']['resultMsg'])

return (jsonResult, result, natName, ed_cd, dataEND)
```

# 3. 공공데이터 API

## ⌚ 프로그램

4칸

```
#[CODE 0]
def main():
    jsonResult = []
    result = []

    print("<< 국내 입국한 외국인의 통계 데이터를 수집합니다. >>")
    nat_cd = input('국가 코드를 입력하세요(중국: 112 / 일본: 130 / 미국: 275) : ')
    nStartYear = int(input('데이터를 몇 년부터 수집할까요? : '))
    nEndYear = int(input('데이터를 몇 년까지 수집할까요? : '))

    ed_cd = "E" #E : 방한외래관광객, D : 해외 출국
    jsonResult, result, natName, ed, dataEND = getTourismStatsService(nat_cd, ed_cd, nStartYear, nEndYear)

    print('jsonResult = ', jsonResult)
```

**jsonResult**: 수집한 데이터를 저장할 리스트 객체로  
JSON 파일 저장용  
**result**: 수집한 데이터를 저장할 리스트 객체로 CSV  
파일 저장용  
**nat\_cd**: 데이터를 수집할 국가 코드  
**natName**: 데이터를 수집할 국가 이름  
**ed\_cd**: 입국/출국 코드('E' 또는 'D')  
**nStartYear**: 데이터 수집 시작 연도  
**nEndYear**: 데이터 수집 끝 연도  
**dataEND**: 마지막 데이터의 연월  
**input()**: 사용자로부터 입력을 받는다.



# 3. 공공데이터 API

## ⌚ 프로그램

```

4칸      8칸
#파일저장 1 : json 파일
with open('./%s_%s_%d_%s.json' % (natName, ed, nStartYear, dataEND), 'w', encoding='utf8') as outfile:
    jsonFile = json.dumps(jsonResult, indent = 4, sort_keys = True, ensure_ascii = False)
    outfile.write(jsonFile)
#파일저장 2 : csv 파일
columns = ["입국자국가", "국가코드", "입국연월", "입국자 수"]
result_df = pd.DataFrame(result, columns = columns)
result_df.to_csv('./%s_%s_%d_%s.csv' % (natName, ed, nStartYear, dataEND), index=False, encoding='cp949')

if __name__ == '__main__':
    main()

```

**jsonFile**: JSON 파일에 저장할 데이터를 담은 객체  
**getTourismStatsList()**: 방한외래관광객 데이터를 요청  
**json.dumps()**: 객체를 JSON 형식으로 변환  
**pd.DataFrame()**: 리스트를 데이터프레임 형식으로 변환  
**to\_csv()**: 데이터프레임을 CSV 파일로 저장

# 3. 공공데이터 API

## ⌚ 실행

명령 프롬프트 - python gmain.py

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Wdossa>cd W_seok

C:\W_seok>cd _datagokr

C:\W_seok\W_seok_datagokr>python gmain.py
<< 국내 입국한 외국인의 통계 데이터를 수집합니다. >>
국가 코드를 입력하세요(중국: 112 / 일본: 130 / 미국: 275) : 112
데이터를 몇 년부터 수집할까요? : 2019
데이터를 몇 년까지 수집할까요? : 2021
```

cd W\_seok

cd \_datagokr

python gmain.py

112

2019

2021

## 4. nodejs 와 Flask API 연동

### nodejs

#### ■ nodejs 란

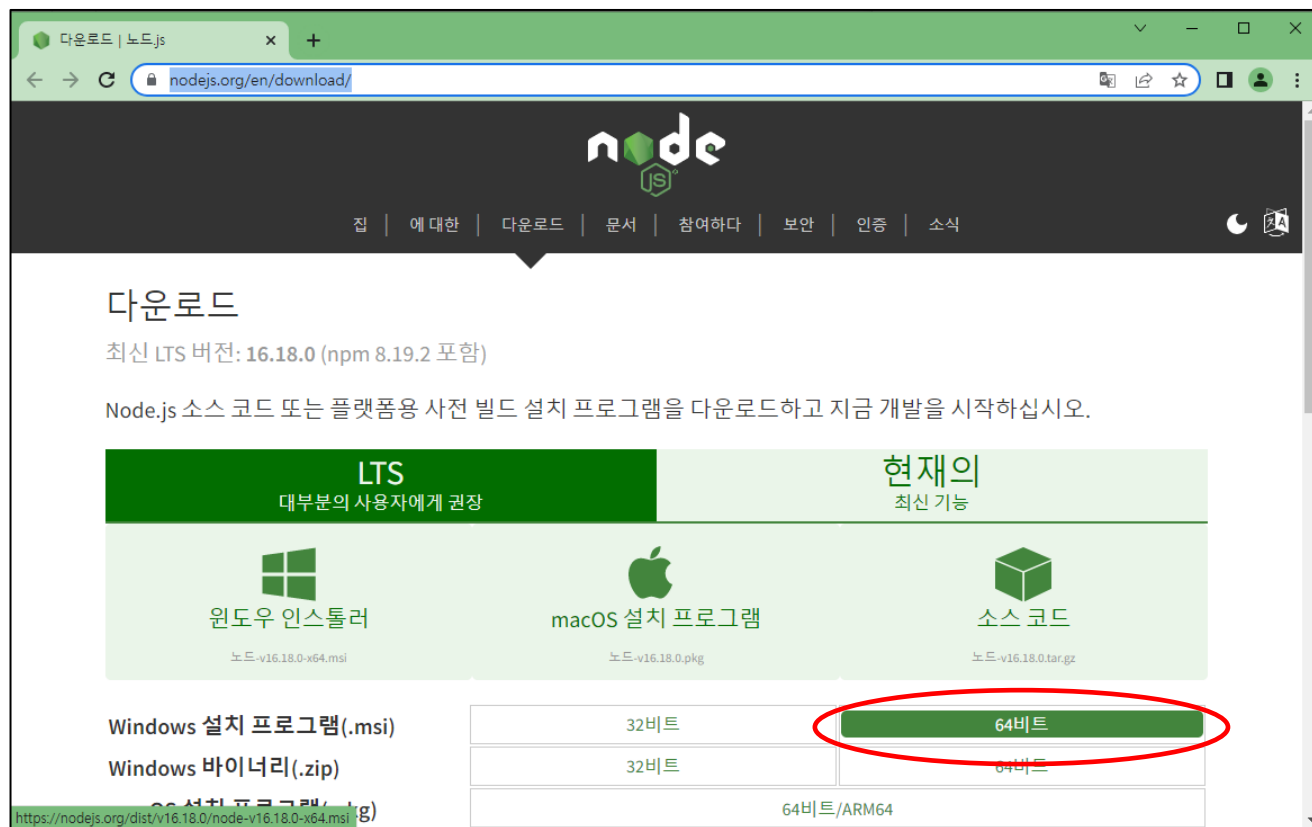
- 오픈 소스 JavaScript 엔진인 크롬 V8에 비동기 이벤트 처리 라이브러리인 libuv를 결합한 플랫폼
- JavaScript로 브라우저 밖에서 서버를 구축하는 등의 코드를 실행할 수 있게 해주는 런타임 환경
- 빈번한 I/O처리에 있어서의 우수한 성능, 서버 확장의 용이성
- JavaScript라는 프론트엔드 필수 언어로 백엔드까지 작성할 수 있다는 엄청난 장점
- 특히 넷플릭스처럼 엄청나게 많은 양의 인풋 아웃풋 데이터를 처리해야 하는 서비스에 있어서 강점

#### ■ npm(Node Package Manager)

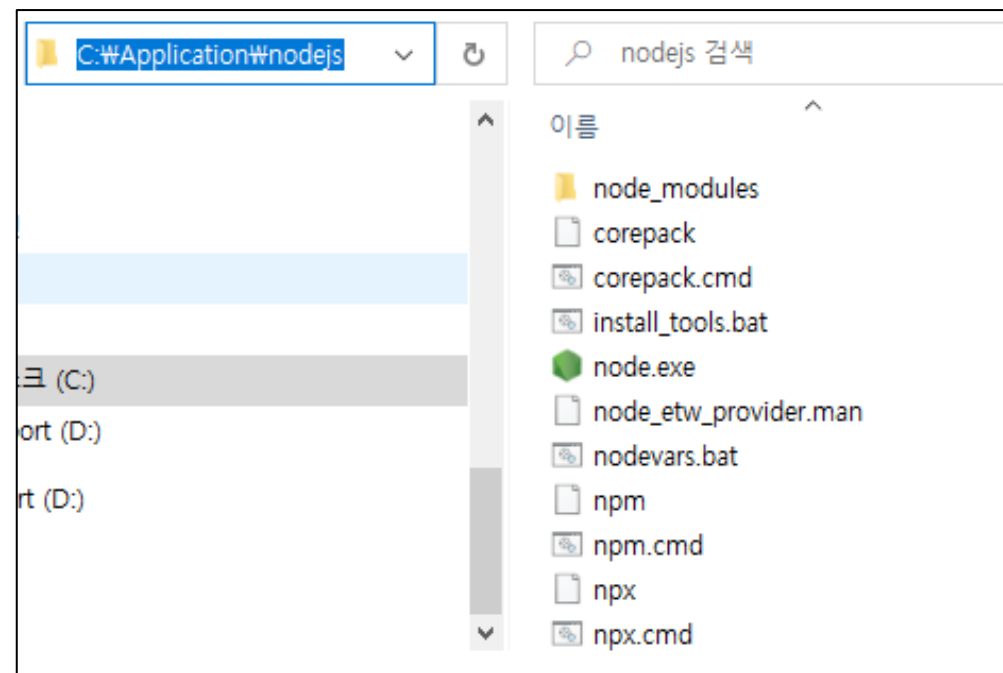
- node.js 를 위한 패키지 매니저이면서 오픈소스 생태계 관리자
- package.json 문서를 활용하여 패키지 생태계의 명세서(패키지의 종류들과 버전) 를 관리
- package-lock.json 은 패키지의 의존성에 대한 정확하고 구체적인 정보를 가지고 있음

# 4. nodejs 와 Flask API 연동

## ⌚ 설치



<https://nodejs.org/en/download/>

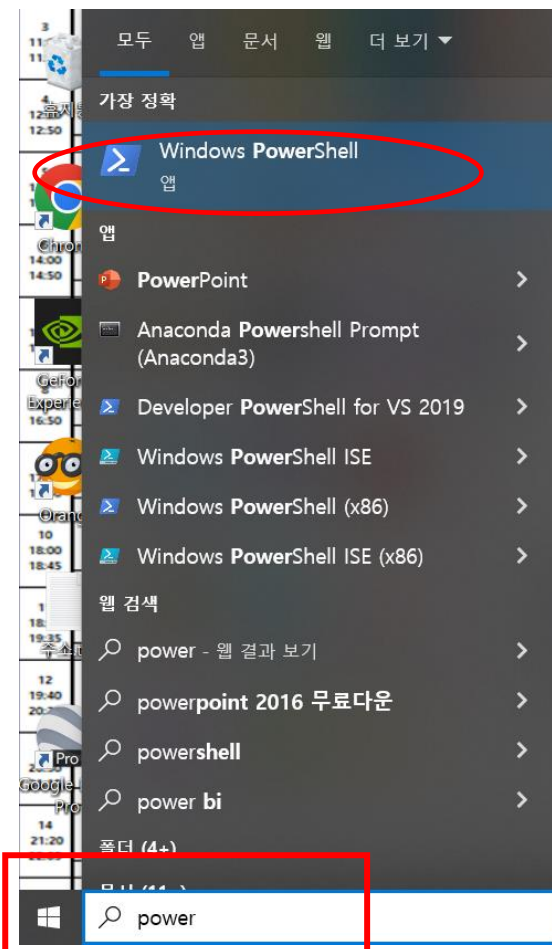


- node-v18.18.2-x64.msi  
파일 다운로드 및 설치

- 프로그램은,  
- C:\Application\nodejs 에 설치

# 4. nodejs 와 Flask API 연동

## ⌚ 설치



- [power]  
입력

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\wdossa> cd W_seok
PS C:\W_seok> npm -version
8.19.2
PS C:\W_seok> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (_seok) myai
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
    
```

cd W\_seok

npm -version

npm init

myai

나머지는  
<Enter Key>

# 4. nodejs 와 Flask API 연동

## ⌚ 설치

```

Windows PowerShell
git repository.
keywords:
author:
license: (ISC)
About to write to C:\₩_seok₩package.json:
{
  "name": "myai",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "directories": {
    "test": "test"
  },
  "scripts": {
    "test": "echo ₩"Error: no test specified₩" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
PS C:\₩_seok>

```

나머지는  
모두

<Enter Key>

## 4. nodejs 와 Flask API 연동

### ⌚ 설치 – 관련 패키지 설치

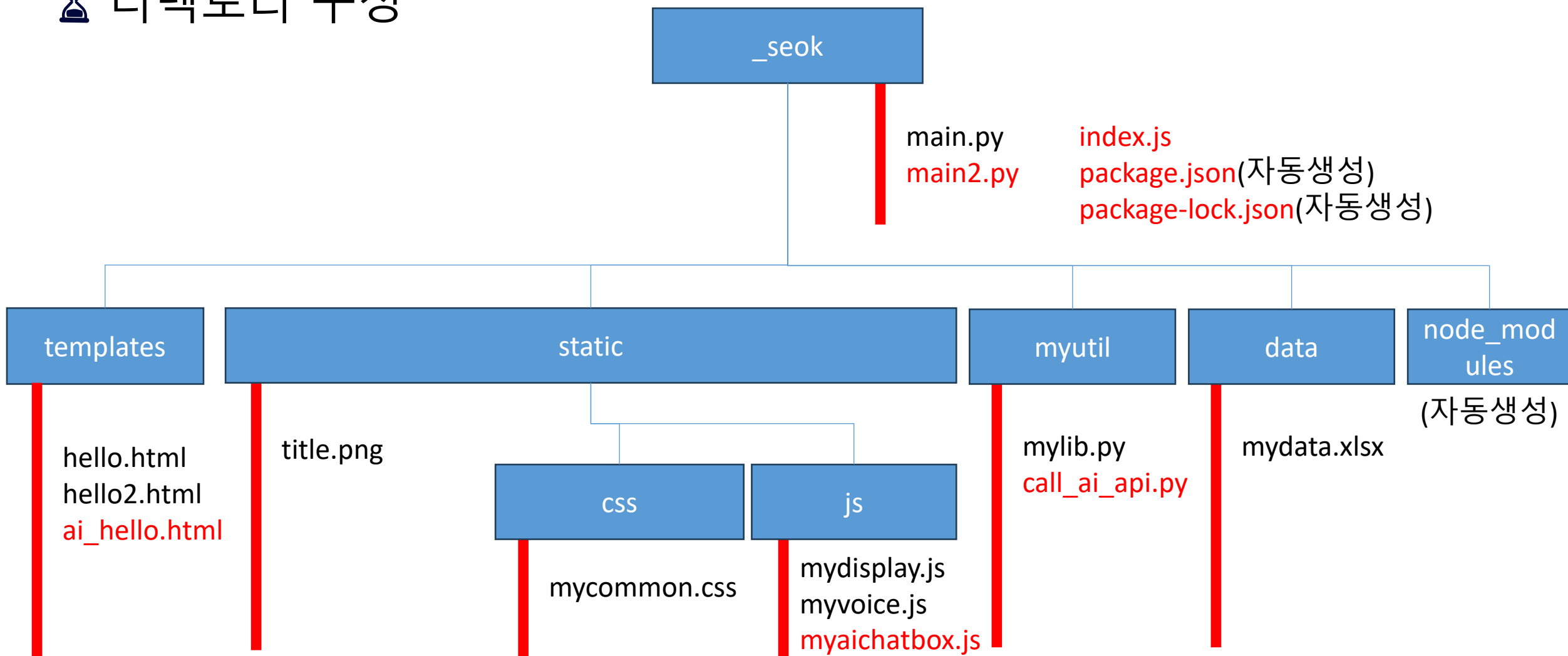
```
npm install express --save
npm install express-generator -g --save
```

```
npm install fs --save
npm install mysql --save
npm install path --save
npm install body-parser --save
npm i dotenv
npm i cors
```

```
npm install mobile-detect
npm install request --save
npm install jquery
npm install request-ip --save
npm install jsonwebtoken --save
npm install cookie-parser --save
npm install moment --save
npm install winston --save
npm install winston-daily-rotate-file --save
npm install ip --save
```

# 4. nodejs 와 Flask API 연동

## ⌚ 디렉토리 구성





# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ index.js 파일 생성

```
//=====
//nodejs Main
//=====
var fs = require('fs');
var express = require('express');
var app = express();
//=====
app.use(express.static("static"));
app.use(express.json()); //json 형태로 parsing
app.use(express.urlencoded({ extended: false }));
//=====
app.get('/', function (req, res) {
  var url = "";
  if(req.url == '/favicon.ico') {
    return res.writeHead(404);
  }
  else if (req.url == '/') {
    url = '/templates/ai_hello.html';
  }
  else {
    url = req.url;
  }
}
```

```
res.writeHead(200);
msg = `html의 위치는:${__dirname}${url}`;
console.log(msg);
res.end(fs.readFileSync(__dirname + url));
});
//=====
app.get('/html/:htmlname', function (req, res) {
  var url = "";
  var pagename = '/templates/' + req.params.htmlname;

  url = pagename;
  res.writeHead(200);
  msg = `html의 위치는:${__dirname}${url}`;
  console.log('html의 위치는 ' + __dirname + url);

  res.end(fs.readFileSync(__dirname + url));
});
//=====
app.post('/get_answer', function (req, res) {
  var htmlStr = "";
  var quest1 = req.body.question;

  if(quest1 == "") quest1 = "질문을 하세요~";
```

## 4. nodejs 와 Flask API 연동

### ⌚ 프로그램코딩

#### ■ index.js 파일 생성

```
const { spawn } = require("child_process");
const Python = spawn("python", ["myutil/call_ai_api.py", quest1]);

Python.stdout.on("data", (data) => {
  htmlStr += data;
});

Python.stderr.on("data", (data) => {
  htmlStr += data;
});

Python.on("close", (code) => {
  var resData = {};
  resData.answer = htmlStr;
  res.json(resData);
});
//=====
const port = 5555;
app.listen(port, ()=>{
  console.log('Server listening on port ', port);
});
```

파일 - 다른이름으로저장

\_seok밑에 index.js 로 저장

# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ package.json 파일 수정

```
{
  "name": "myai",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "directories": {
    "test": "test"
  },
  "scripts": {
    "start": "nodemon index.js",
    "test": "echo W\"Error: no test specifiedW\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "express": "^4.18.2",
    "fs": "^0.0.1-security",
    "mysql": "^2.18.1",
    "path": "^0.12.7"
  }
}
```

파일 – 저장

\_seok 밑에 package.json 으로 저장

붉은 색 줄만 추가

# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ html 파일 생성

```
<!DOCTYPE html>
<html>
<head>
  <title>챗봇 비서</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src='/js/mydisplay.js'></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <link rel="stylesheet" href="/css/mycommon.css">
  <script>
    var a = return_screen_size();
    document.write(a);
  </script>
</head>
<body>
  <div class="container">
    <div></div>
    <div class="chatbox" id="chatbox">
      <div class="bot-message">안녕하세요~</div>
    </div>
    <form id="input-form" method="post">
      <button id="start-btn" type="button"><span class="blind">음성</span></button>
      <input type="text" name="input_data" id="input_data" placeholder="hello...">
    </form>
  </div>
  <div></div>
  <div class="chatbox" id="chatbox">
    <div class="bot-message">안녕하세요~</div>
  </div>
  <form id="input-form" method="post">
    <button id="start-btn" type="button"><span class="blind">음성</span></button>
    <input type="text" name="input_data" id="input_data" placeholder="hello...">
  </form>
</body>
</html>
```

```
    <button type="submit">전송</button>
  </form>
</div>
<script src='/js/myaichatbox.js'></script>
<script src='/js/myvoice.js'></script>
</body>
</html>
```

파일 - 다른이름으로 저장

\_seok\templates\에 ai\_hello.html 로 저장

# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ javascript 파일 생성

```
//=====
// AI 채팅 박스
//=====
function typeWriter(element, text, i, fnCallback) {
    if (i < text.length) {
        if (text.charAt(i) === '<') {
            var tag = '';
            while (text.charAt(i) !== '>') {
                tag += text.charAt(i);
                i++;
            }
            tag += '>';
            element.innerHTML += tag;
            i++;
        } else {
            element.innerHTML += text.charAt(i);
            i++;
        }
        setTimeout(function() { typeWriter(element, text, i, fnCallback);}, 10);
    } else if (typeof fnCallback === 'function') {
        setTimeout(fnCallback, 700);
    }
}
```

```
function animateBotResponse(response) {
    var botMessage = $('<div class="bot-message"></div>');
    $('#chatbox').append(botMessage);
    typeWriter(botMessage[0], response, 0, function() {
        scrollToBottom();
    });
}
```

뒷페이지 계속

# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ javascript 파일 생성

```
$(function() {
  $('#input-form').submit(function(event) {
    event.preventDefault();
    var user_input = $('#input_data').val();
    if (!user_input) return;
    $('#chatbox').append('<div class="user-message">' + user_input + '</div>');
    $('#input_data').val('');
    $.ajax({
      url: '/get_answer',
      type: 'POST',
      data: JSON.stringify({question: user_input}),
      contentType: 'application/json',
      success: function(data) {
        var bot_response = data.answer;
        bot_response = bot_response.replace(/\n/g, '<br>');
        animateBotResponse(bot_response);
      }
    });
  });
});
```

파일 - 다른이름으로 저장

\_seok\static\js 밑에 myaichatbox.js 로 저장

# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ call\_ai\_api.py 파일 생성

```

4칸 8칸 12칸
#=====
#AI API
#=====
import sys, io
import requests
import json
sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding='utf-8')

ai_band = "myai"
ai_model = "aiya"
AI_URL = "http://172.16.11.220:9999/api/get_data"
SECRET_KEY = "AAAAAAAAAAAAAABBBC111"
#-----
class ChatbotMessageSender:
    ep_path = AI_URL

    def req_message_send(self):
        self.ep_path = AI_URL
        request_body = {
            'key': SECRET_KEY,
            'question': myquery,
            'event': 'send'
        }

```

# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

### ■ call\_ai\_api.py 파일 생성

4칸 8칸 12칸

```
custom_headers = {
    'Content-Type': 'application/json;UTF-8'
}
encode_request_body = json.dumps(request_body).encode('UTF-8')
response = ""
try:
    response = requests.post(headers=custom_headers, url=self.ep_path, data=encode_request_body)
except Exception as ee:
    _msg = '(', self.ep_path, ') AI API 접속 주소를 확인하세요'
    response = requests.models.Response.text = str(_msg)

return response
```

4칸 8칸 12칸

```
#-----
if __name__ == '__main__':
    myquery = ""
    quest1 = sys.argv[1]

    myai = ChatbotMessageSender()
    myquery = quest1
    res1 = myai.req_message_send()

    json_dict = json.loads(res1.text)

    sResult = json_dict['answer']
    print(sResult)
```

파일 - 다른이름으로 저장

\_seok\myutil 밑에 call\_ai\_api.py 로 저장



# 4. nodejs 와 Flask API 연동

## ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 붉은색 내용 추가

```
#=====
# Flask 웹서버 메인 프로그램
#=====
import socket
import pandas as pd
import requests
from flask import Flask, render_template, request, jsonify
from myutil.mylib import mylib_Read_xlsx_Data, mylib_ViewPage
```

파일 – 저장

\_seok 밑에 main2.py 로 저장

```
4칸 8칸 12칸
#=====
@app.route('/api/get_data', methods=['POST'])
def api_page():

    question = request.json['question']
    print('***question : ', question)
    apikey = request.json['key']
    if apikey != 'AAAAAAAAAAAAAABBBCCC111':
        return jsonify({'answer': 'not supported'})

    answer = ""
    try:
        answer = question + " : 학습이 된 후에 답변 드리겠습니다~"
    except Exception as ee:
        answer = "오류가 발생했습니다~" + ee

    print('***answer : ', answer)
    return jsonify({'answer': answer})
#=====
if __name__ == '__main__':
    #_myip = socket.gethostbyname(socket.gethostname())
    app.run(host='172.16.11.220', port=9999, debug=False)
```

# 4. nodejs 와 Flask API 연동

## ⌚ 실행

### ■ nodejs 실행

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dossa>cd \_seok

C:\_seok>npm start

> myai@1.0.0 start
> nodemon index.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node index.js'
Server listening on port 5555
```

cd \\_seok

npm start

### ■ Flask 실행

```
C:\_seok>명령 프롬프트 - python main2.py

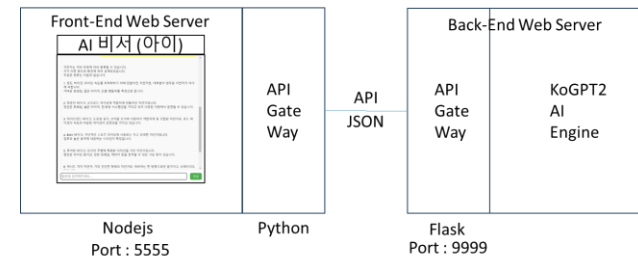
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dossa>cd \_seok

C:\_seok>python main2.py
* Serving Flask app 'main2'
* Debug mode: off
WARNING: This is a development server. Do not use without logging in!
* Running on http://172.16.11.220:9999
Press CTRL+C to quit
```

cd \\_seok

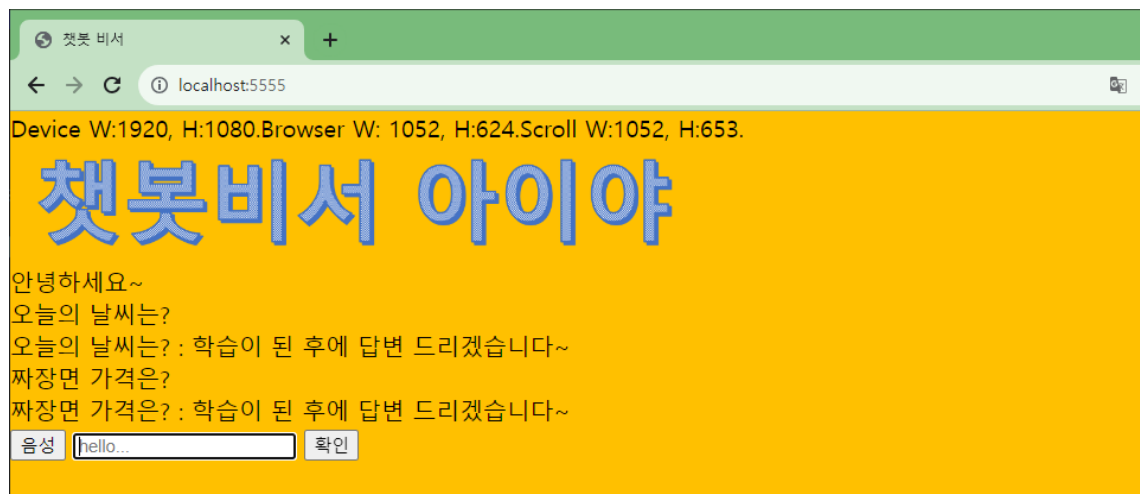
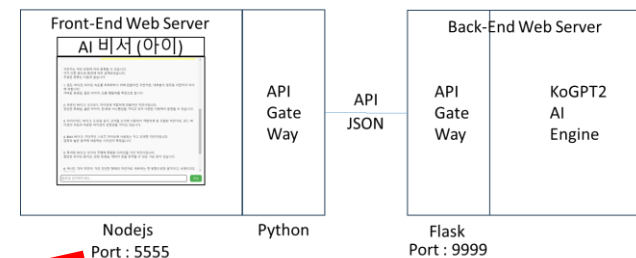
python main2.py



# 4. nodejs 와 Flask API 연동

⌚ 접속

http://localhost:5555/



```

(c) 명령 프롬프트 - python main2.py

(c) Microsoft Corporation. All rights reserved.

C:\Users\wdossa>cd \_seok

C:\_seok>python main2.py
* Serving Flask app 'main2'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a prod
* Running on http://172.16.11.220:9999
Press CTRL+C to quit
172.16.11.220 - - [16/Oct/2023 18:44:03] "GET / HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:03] "GET /static/css/mycommon.css HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:03] "GET /static/title.png HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:03] "GET /static/js/myvoice.js HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:03] "GET /static/js/mydisplay.js HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:03] "GET /favicon.ico HTTP/1.1" 404 -
***input_data :   aaa
172.16.11.220 - - [16/Oct/2023 18:44:18] "POST /get_data HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:20] "GET / HTTP/1.1" 200 -
172.16.11.220 - - [16/Oct/2023 18:44:20] "GET /static/js/mydisplay.js HTTP/1.1" 304 -
172.16.11.220 - - [16/Oct/2023 18:44:20] "GET /static/css/mycommon.css HTTP/1.1" 304 -
172.16.11.220 - - [16/Oct/2023 18:44:20] "GET /static/title.png HTTP/1.1" 304 -
172.16.11.220 - - [16/Oct/2023 18:44:20] "GET /static/js/myvoice.js HTTP/1.1" 304 -
***question :   오늘의 날씨는?
***answer :   오늘의 날씨는? : 학습이 된 후에 답변 드리겠습니다~
172.16.11.220 - - [16/Oct/2023 18:44:37] "POST /api/get_data HTTP/1.1" 200 -
***question :   짜장면 가격은?
***answer :   짜장면 가격은? : 학습이 된 후에 답변 드리겠습니다~
172.16.11.220 - - [16/Oct/2023 18:44:55] "POST /api/get_data HTTP/1.1" 200 -
  
```

# 참고 자료

- 자바와 파이썬으로 만드는 빅데이터시스템(제이펍, 황세규)
- 위키독스(<https://wikidocs.net/22654>)
- 네이버블로그(<https://blog.naver.com/classmethodkr/222822485338>)
- 데이터분석과 인공지능 활용 (NOSVOS, 데이터분석과인공지능활용편찬위원회 편)

## 참고 사이트

유튜버 : 빅공잼 : <https://www.youtube.com/watch?v=bnYxO2XRCQ0>

네이버 블로그 : 빅공잼

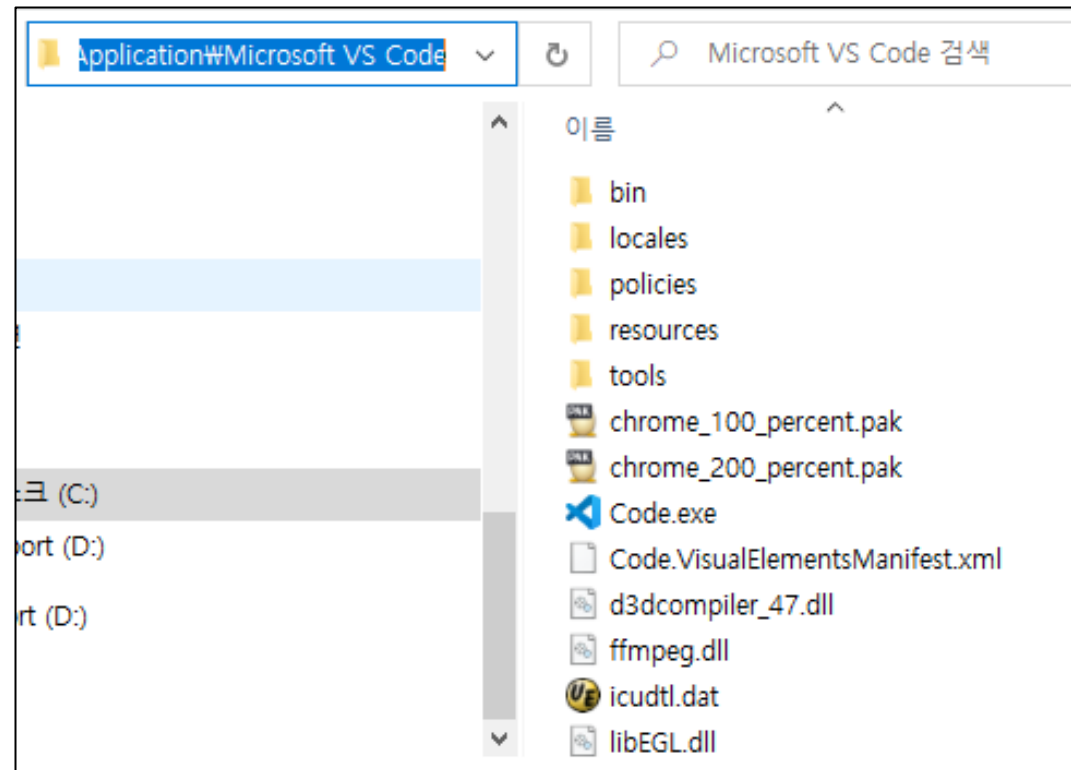
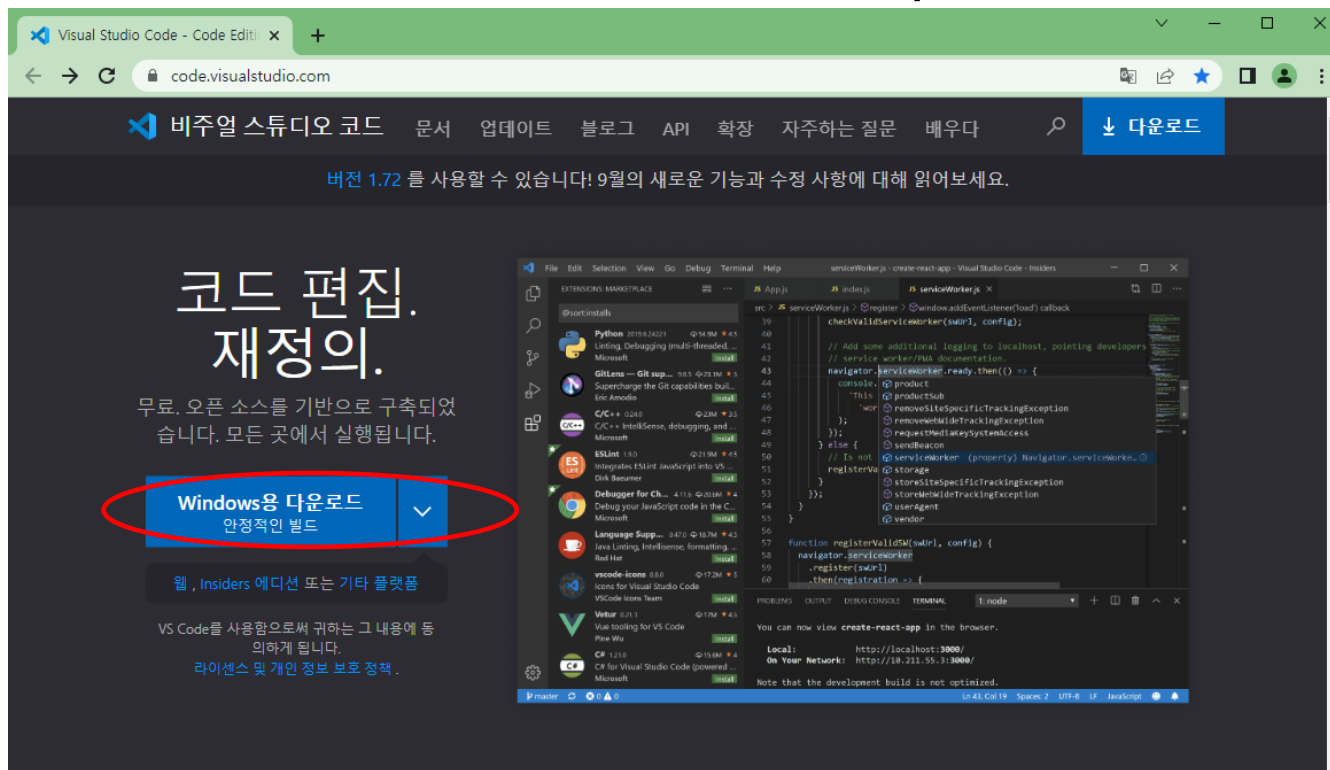
<https://biggongjam.notion.site/3-Hadoop-cd6944182da74edf8d2339b654e0bfb9>

<https://biggongjam.notion.site/4-Spark-2c341ddc8715411484cb2f0254b60126>

Q n A

# \* TOOL설치

## Visual Studio Code 설치

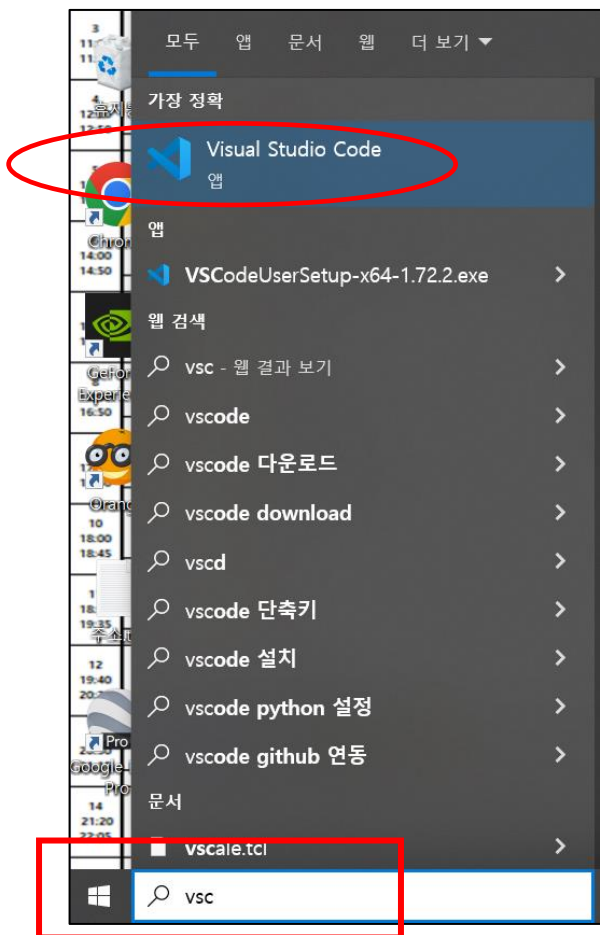


<https://code.visualstudio.com/>

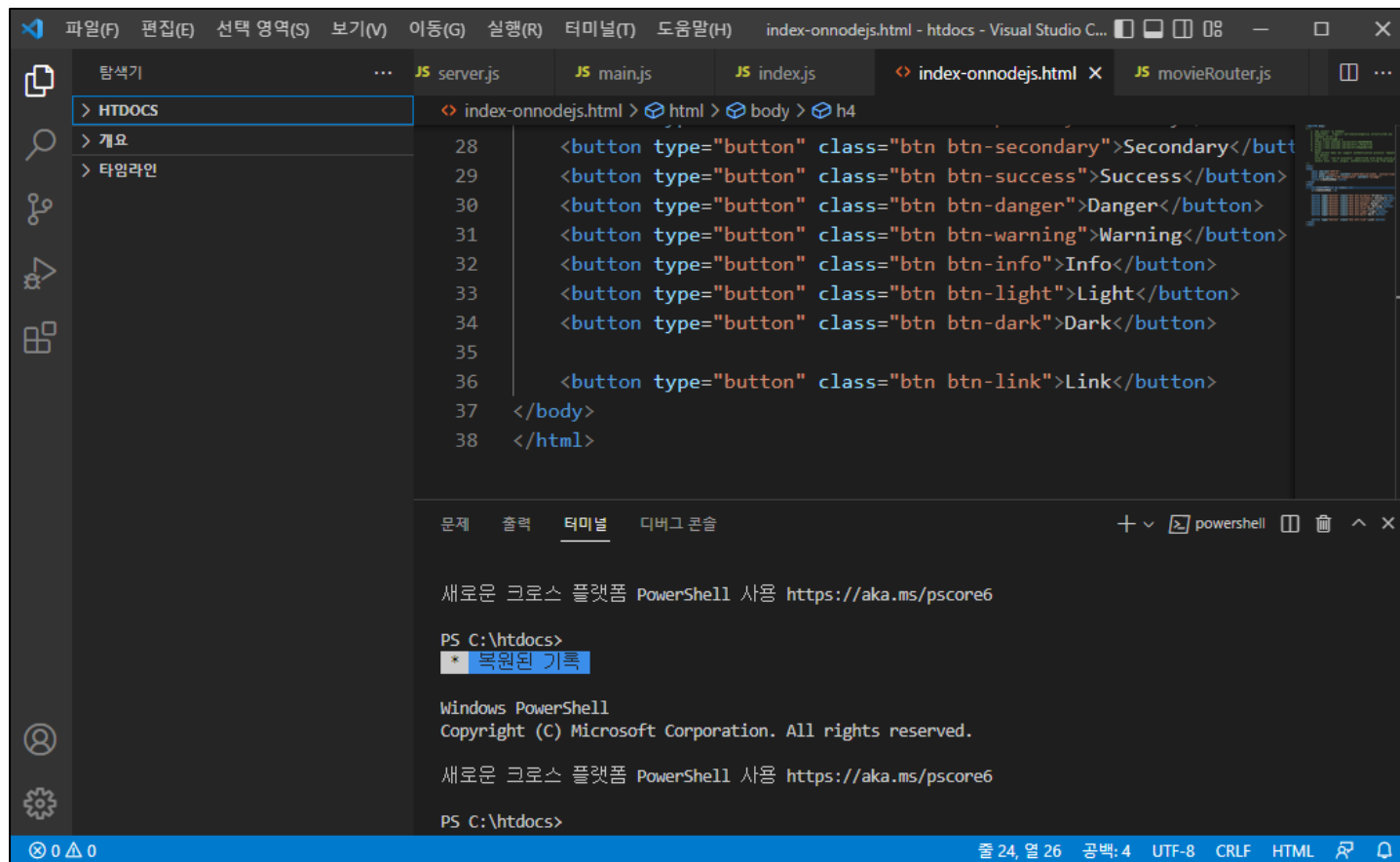
- VSCodeUserSetup-x64-1.83.1.exe  
파일 다운로드 및 설치
- C:\Application 하단에 설치함
- C:\Application\Microsoft VS Code 에 설치됨

# \* TOOL설치

## Visual Studio Code 설치 - 확인

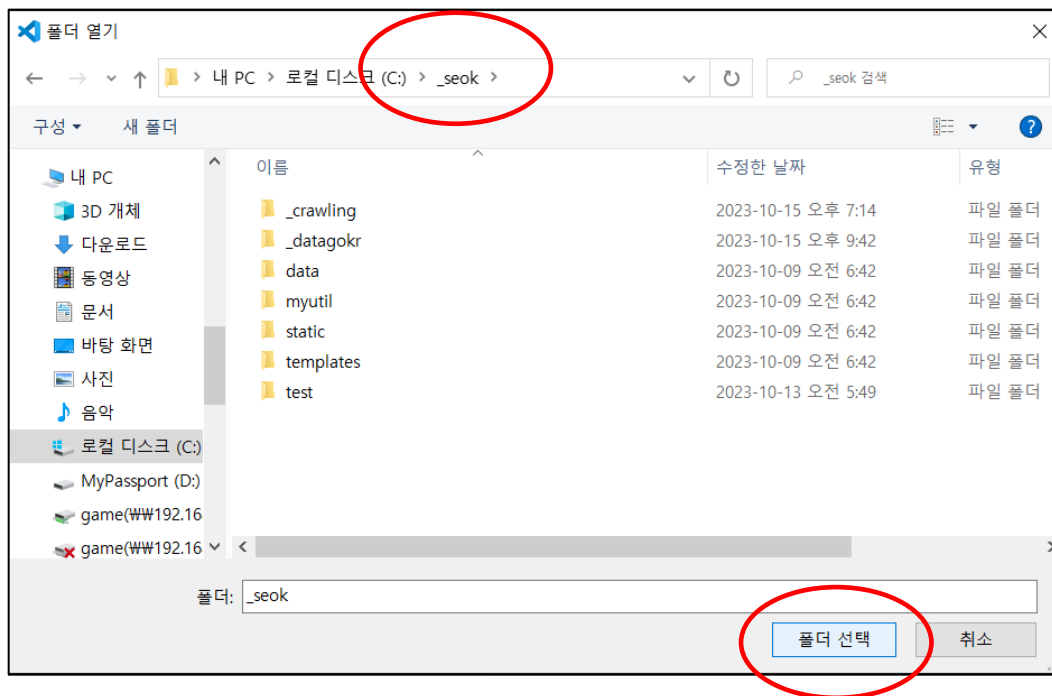
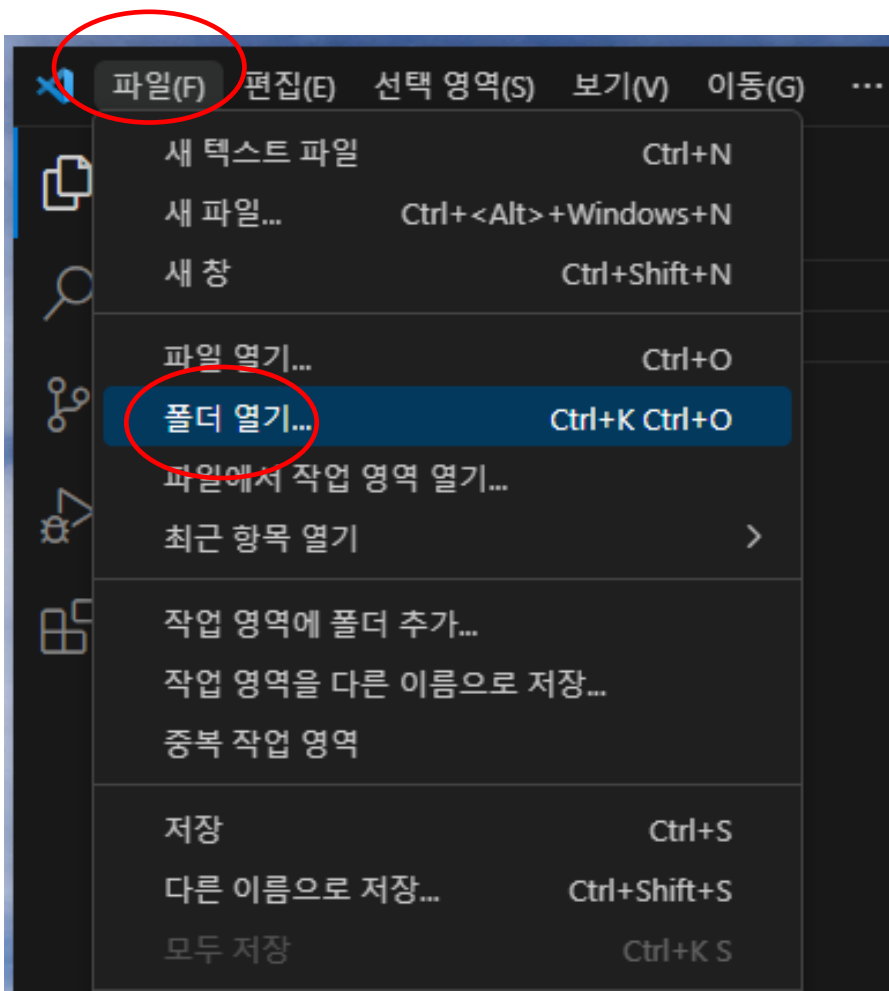


- [vsc] 입력



# \* TOOL설치

## ☁ Visual Studio Code - 프로젝트설정



- 본인의 프로젝트 폴더 \_seok 선택



# \* PC 부팅 시, nodejs 자동 실행

## ⌚ 환경 설정

- 메모장으로 작성 (autostart\_nodejs5555.bat 의 내용)

```
@echo off
echo -----by seokill
echo C:\Users\dossa\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\autostart_nodejs5555.bat
cd c:\_seok
npm start
pause
```

- autostart\_nodejs5555.bat 파일을 아래의 디렉토리로 복사

C:\₩ → 사용자 → dossa → AppData → Roaming → Microsoft → Windows → 시작 메뉴 → 프로그램 → 시작프로그램

또는

[윈도우] + [r] → shell:startup 입력 후 창을 열어도 됨