

파이썬 자료형

[부성순] bakpak@empas.com

입력데이터

- **csv, excel,txt, json, xml등 파일데이터**
파일에 따라 전문 읽기 명령어로 처리

- (1) 내pc(로컬)자료
- (2) 네트워크로 연결된 다른pc자료

- **데이터베이스 서버의 데이터**
데이터베이스전문처리명령어 실행(예: SQL)

- (1) 내pc(로컬)자료
- (2) 네트워크로 연결된 다른 pc자료
- (3) 네트워크 데이터베이스 서버

- **웹페이지 자료**
크롤링(웹스크래핑)으로 필요한 자료를 가져오거나
API를 이용하여 허가받고 자료를 연결함

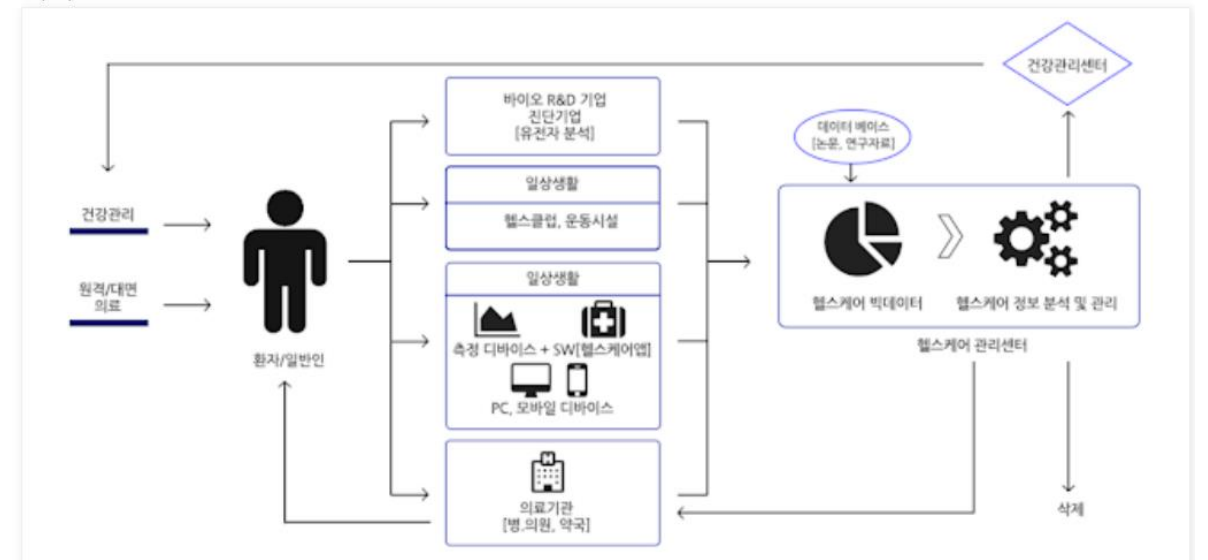
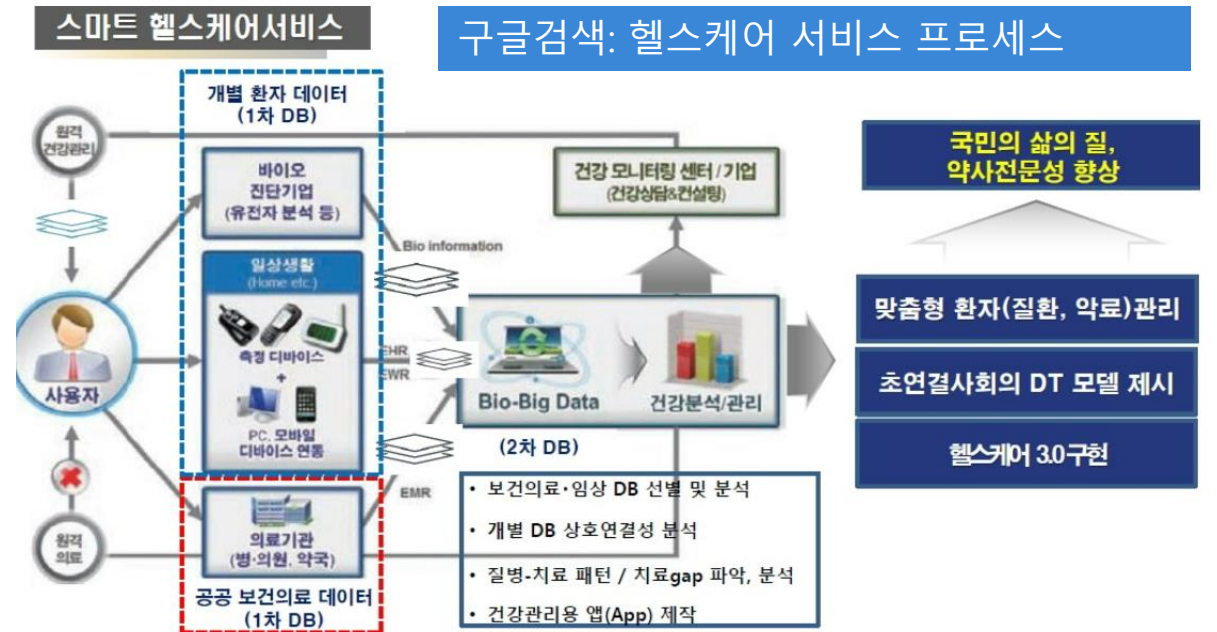
- (1) 인터넷상의 웹페이지 자료의 구조를 보고
- (2) 다른네트워크의 저장된 자료셋을 연결하여서

- **의료정보 표준화**

<https://www.aitimes.com/news/articleView.html?idxno=140168>

<https://aifactory.space/task/2290/discussion/280>

구글검색: '의료데이터 표준화 json'



[무작정 해보기] csv, txt 와 json 파일을 파이썬에서 처리하고자함

csv는 몇 가지 필드를 쉼표(,)로 구분한 텍스트 데이터 및 텍스트 파일이며 row--> enter단위로 자료를 가져옴

```
1 data[:3]
```

```
[['년도', '시도', '시군구', '성별', '연령구분', '환자수', '입내원일수', '요양급여비용'],  
 ['2018', '서울', '강남구', '남', '0~9세', '57', '121', '9914070'],  
 ['2018', '서울', '강남구', '남', '10~19세', '63', '130', '9030550']]
```

행(row)

열(column)

1

2

3

4

5

6

[무작정 해보기] csv, txt 와 json 파일을 파이썬에서 처리하고자함

- [1] csv자료를 with 구문으로 읽을수 있음.
숫자,문자가 섞여있는 경우는 pandas 모듈이 더 편리함

```
1 import csv
2
3 path = '/content/시군구별 성별 연령별 비만진료 통계 2018~2022.csv'
4
5 data=[] #빈공간리스트 준비
6 # CSV 파일 읽기
7 with open(path, 'r', encoding='euc-kr') as file:
8     reader = csv.reader(file)
9
10    # 첫 번째 줄은 헤더로 처리
11    header = next(reader) # 이코드때문에 data[0]이 제목이 되지 않고, 자료가 됨
12
13    # 데이터 출력 (옵션)
14    for row in reader:
15        data.append(row) # 각 행 출력 예시
16
17 print('end')
```

end

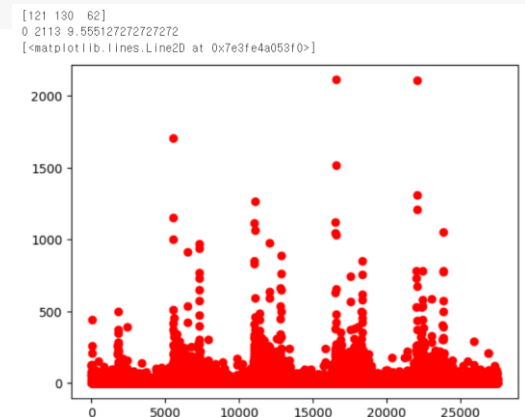
```
] 1 data[:3]
```

```
[[['2018', '서울', '강남구', '남', '0~9세', '57', '121', '9914070'],
  ['2018', '서울', '강남구', '남', '10~19세', '63', '130', '9030550'],
  ['2018', '서울', '강남구', '남', '20~29세', '34', '62', '5063090']]
```

0	1	2	3	4	5	6	7
년도	시도	시군구	성별	연령구분	환자수	입내원일수	요양급여비용

- [2] csv 자료중 '입내원일수' (6열) 자료로 기본적인 통계와,
차트 작성

```
1 import numpy as np
2 dataArray=np.array(data)
3 value=dataArray[:,6].astype('int') #숫자형변환
4
5 print(value[:3]) # 0,1,2 3개만
6
7 # 데이터의 최저, 최고, 평균값 출력
8 print(np.min(value), np.max(value), np.mean(value))
9
10 # 차트 출력
11 import matplotlib.pyplot as plt
12 plt.plot(value, 'ro') # r:빨강, o:동그라미 출력
```



(참고) [3] 환자수 (5번열), 입내원일수(6번열), 요양급여비용(7번열) 의 값을 반복하여서 처리하고자 할때는 함수로 작업하면 편리함

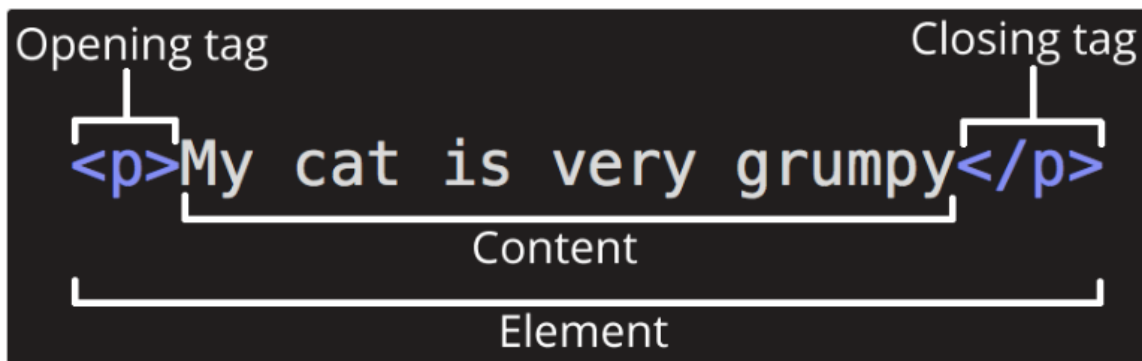
```
1 #####
2 ### [함수]
3 ### 자료의 열 값을 넣으면
4 ### 숫자로 변경하고, 기본 통계를 출력, 차트작성
5 #####
6 import numpy as np
7 import matplotlib.pyplot as plt
8 def my(columnBun):
9
10     dataArray=np.array(data)
11     value=dataArray[:,columnBun].astype('int') #숫자형변환
12
13     print(value[:3]) # 0,1,2 3개만
14
15     # 데이터의 최저, 최고, 평균값 출력
16     print(np.min(value), np.max(value), np.mean(value))
17
18     # 차트 출력
19
20     plt.plot(value, 'ro') # r:빨강, o:동그라미 출력
21
22 my(7)
```

Html

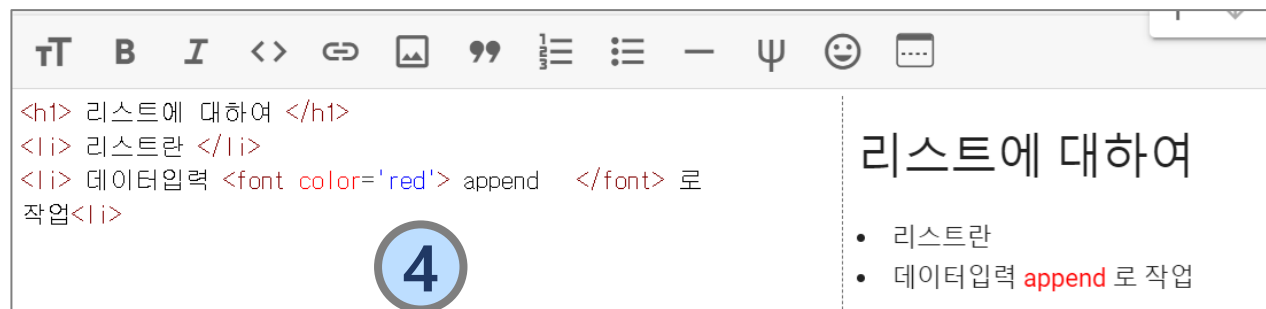
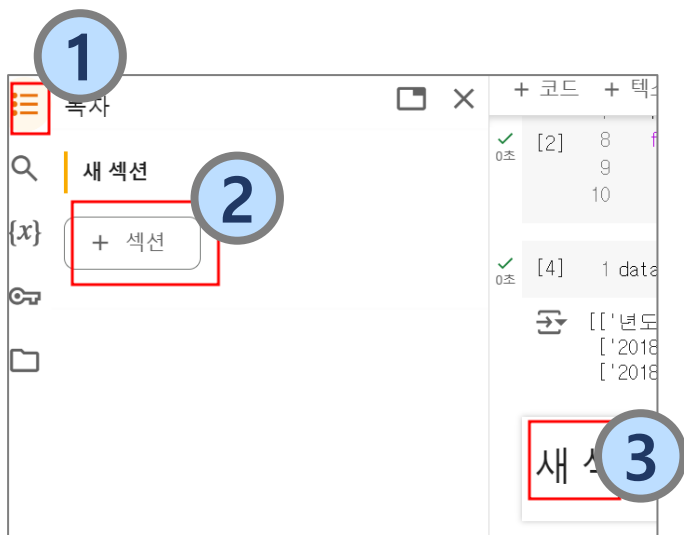
[https://developer.mozilla.org/ko/docs/Learn/HTML/Introduction to HTML/Getting started](https://developer.mozilla.org/ko/docs/Learn/HTML/Introduction_to_HTML/Getting_started)

HTML (Hypertext Markup Language, 하이퍼텍스트 마크업 언어)는 프로그래밍 언어는 아니고, 우리가 보는 웹페이지가 어떻게 구조화되어 있는지 브라우저로 하여금 알 수 있도록 하는 마크업 언어

마크업 언어란 태그 등을 이용해서 문서나 데이터의 구조를 표현하는 **언어**의 한 가지로서 '마크(Mark)'로 둘러싸인, 문서의 구조를 정의하는 **언어**임.



코랩에서 Markdown



강의내용

- 파이썬 List

<https://wikidocs.net/14>

- 파이썬 딕셔너리

<https://wikidocs.net/16>

- 파이썬 Numpy



파이썬에서 리스트

- ▶ 파이썬에서 리스트(list)란 대괄호([,])로 감싸서 나타내며, 안에는 0개 이상의 원소가 저장될 수 있음

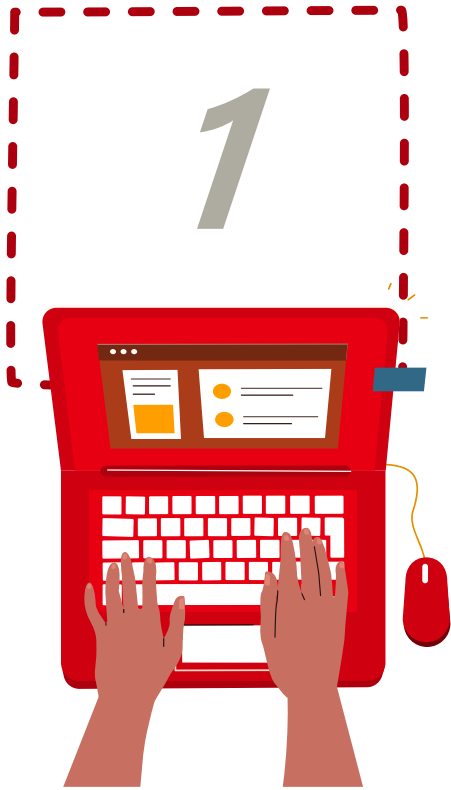
예) `a=[]` 또는 `a = list()`, `a=[1]`, `a=['마라']`, `a=[20,30,40]`

- ▶ 원소들이 연속적으로 저장되는 형태의 자료형

예) `a= [20,30]` , `a=[20,30, [1,2,3,]]`

- ▶ 이때 저장되는 요소들이 모두 같은 자료형일 필요는 없다
*참고 Array(행렬)은 반드시 갯수를 통일하고, 자료형을 통일한다.
숫자,문자가 들어가면 모두 문자 처리함

예) `a=[`
 `['서울',20,30]`
 `['부산',15]`
 `['제주',25,30,50]`
 `]`



파이썬 딕셔너리 https://wikidocs.net/16#_1

- ▶ 'key': value 값으로 구성됨 {} 안에 표현함
- ▶ .json 은 딕셔너리 형식임

[실전] 파이썬에서 서울시 공공데이터 json 가져오기

서울시 공공데이터 접속하여

```
import urllib.request
import json
url='http://openapi.seoul.go.kr:8088/sample/json/SeoulPublicLibraryInfo/1/5/'
plainText=urllib.request.urlopen(url).read().decode()
plainText
```

```
jsonData=json.loads(plainText)
jsonData
```

```
libData=jsonData.get('SeoulPublicLibraryInfo').get('row')
```

```
import pandas as pd
df=pd.DataFrame(libData)
df.head()
```

LBRRY_SEQ_NO	LBRRY_NAME	GU_CODE	CODE_VALUE	ADRES	TEL_NO	HMPG_URL	OP_TIME	FORM_CI
--------------	------------	---------	------------	-------	--------	----------	---------	---------

				서울 특별				
--	--	--	--	----------	--	--	--	--

평일 :

```
df.to_csv('공공자전거.csv', encoding='euc-kr', index=False)
```

[실전] 파이썬에서 서울시 공공데이터 json 가져오기



```
1 ### 만약 모든 자료가 아니라 필요한 열만 넘기고자 한다면
2 dataList=[]
3 for item in libData:
4     name=item.get('LBRRY_NAME')
5     tel=item.get('TEL_NO')
6     dataList.append({'name':name,
7                      'tel':tel})
8 pd.DataFrame(dataList)
```



	name	tel
0	강남구립못골도서관	02-459-5522
1	강남역삼푸른솔도서관	02-2051-1178
2	개포하늘꿈도서관	02-451-1511
3	논현도서관	02-3443-7650



[실전] 파이썬에서 서울시 공공데이터 json 가져오기

1

```
1 libData=jsonData['SeoulPublicLibraryInfo']['row']
2
3 import pandas as pd
4 df=pd.DataFrame(libData)
5 df.head() # = df.head(5)
```

-> df.head()는 5가 생략된 코
드임
df.head(3) 위로 3개의 자료보
기

✓ 0.0s

LBRRY_SEQ_NO	LBRRY_NAME	GU_CODE	CODE_VALUE	ADRES
--------------	------------	---------	------------	-------

서울특
별시

0	1925	강남구립못골	0008	가남구 가남구
---	------	--------	------	---------

3

```
1 #df[['TEL_NO','HMPG_URL']] # df.
2 df.iloc[:, [6,5]]
```

✓ 0.0s

HMP

0	http://library.gangnam.go.k
---	-----------------------------

1	https://library.gangnam.go.kr/ysplib/i
---	--

2

```
1 df['TEL_NO'][:3] # df.iloc[:, 5][:3]
```

✓ 0.0s

0	02-459-5522
---	-------------

1	02-2051-1178
---	--------------

2	02-451-1511
---	-------------

Name: TEL_NO, dtype: object

4

```
1 # 서울시 도서관 전체 갯수는?
2 len(df)
```

✓ 0.0s

206

[실전] 파이썬에서 서울시 공공데이터 json 가져오기

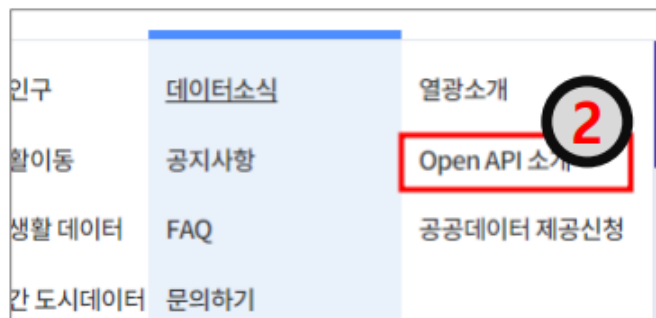
[사전지식] 공공데이터란? open API란?

(1) 서울시 공공데이터에 접속 <https://data.seoul.go.kr/>

(2) 회원가입 및 openAPI키 발급받기

서울시 공공데이터

- 공공데이터
- OpenAPI



(3) 일반 인증키 신청

(4) 인증키관리에서 본인인증키 복사



[실전] 파이썬에서 서울시 공공데이터 json 가져오기

- [1] 서울시 공공데이터
- [2] 검색어 입력 '공공도서관'
- [3] 공공데이터 클릭
- [4] 서울시 공공도서관 현황정보 클릭

The screenshot shows the '서울 열린데이터 광장' (Seoul Open Data Portal) search results for '공공도서관' (Public Library). The search results show 3 items. The first item, '서울시 공공도서관 현황정보' (Seoul Public Library Status Information), is highlighted with a red box and a circled '4'. The search bar contains '공공도서관' with a circled '2'. The '공공데이터(3)' button is circled with a '3'. The top navigation bar has '공공데이터' circled with a '1'.

- [5] OpenAPI
- [6] 샘플 URL 복사해서

The screenshot shows the '미리보기' (Preview) section for the '서울시 공공도서관 현황' (Seoul Public Library Status) dataset. The 'Open API' button is circled with a '5'. The '샘플 URL' (Sample URL) is circled with a '6' and contains the URL: `http://openapi.seoul.go.kr:8088/(인증키)/xml/SeoulPublicLibraryInfo/1/5/`. Below the URL, an XML sample is shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<SeoulPublicLibraryInfo>
<list_total_count>206</list_total_count>
<RESULT>
<CODE>INFO-000</CODE>
<MESSAGE>정상 처리되었습니다</MESSAGE>
</RESULT>
<row>
<LBRRY_SEQ_NO>1935</LBRRY_SEQ_NO>
<LBRRY_NAME>강남구립못골도서관</LBRRY_NAME>
<CU_CODE>0000</CU_CODE>
```

[실전] 파이썬에서 서울시 공공데이터 json 가져오기

제공되는 샘플

[http://openapi.seoul.go.kr:8088/\(인증키\)/xml/SeoulPublicLibraryInfo/1/5/](http://openapi.seoul.go.kr:8088/(인증키)/xml/SeoulPublicLibraryInfo/1/5/)

샘플인증키

78734b626c62616b35344947554c45

코드변환 url을 상황에 따라 나누어서 변환해야 하는 경우가 많으므로 아래와 같이 url을 나누어작업함

```
6 import urllib.request
7 import json
8
9 apiKey='78734b626c62616b35344947554c45'
10 url='http://openapi.seoul.go.kr:8088/'
11 url+=apiKey
12 url+=' /json/SeoulPublicLibraryInfo/1/5/'
13 print(url)
14
```

✓ 0.0s

전체데이터 갯수는 현재 206개이지만
샘플로 5개만 확인함

<http://openapi.seoul.go.kr:8088/78734b626c62616b35344947554c45/json/SeoulPublicLibraryInfo/1/5/>

url이 잘 출력되는지 확인
용

[실전] 파이썬에서 서울시 공공데이터 json 가져오기

```
{
  "SeoulPublicLibraryInfo": {
    "list total count": 206,
    "RESULT": {
      "CODE": "INFO-000",
      "MESSAGE": "정상 처리되었습니다"
    },
    "row": [
      {
        "LBRRY_SEQ_NO": "1935",
        "LBRRY_NAME": "강남구립못골도서관",
        "GU_CODE": "0008",
        "CODE_VALUE": "강남구",
        "ADRES": "서울특별시 강남구 자곡로 116"
```

```
import urllib.request
import json
```

```
api_Key='78734b626c62616b35344947554c45'
url='http://openapi.seoul.go.kr:8088/'
url+=api_Key
url+= '/json/SeoulPublicLibraryInfo/1/206/'
plainText=urllib.request.urlopen(url).read().decode()
jsonData=json.loads(plainText)
libData=jsonData.get('SeoulPublicLibraryInfo').get('row')
```

데이터셋의 갯수는 직접 확인하여서 206개로 변경해도 되고 계속 변동되는 자료라면 List_total_count값으로 다시 연결할수는 있지만 여기서는 갯수를 직접 입력하는 코드로 작업하고자 함.

```
import pandas as pd
df = pd.DataFrame(libData)
display(df)
```


[실전] 파이썬에서 서울시 공공데이터 json 가져오기

```
1 import pandas as pd
2 df = pd.DataFrame(libData)
3 display(df)
4 df.csv()
```

0.0s

field라고 함.

	LBRRY_SEQ_NO	LBRRY_NAME	GU_CODE	CODE_VALUE	ADRES	TEL_NO	HMPG_URL	OP_TIME	FDRM
0	1935	강남구립못골도서관	08	강남구	서울특별시 강남구 자곡로 116	02-459-5522	http://library.gangnam.go.kr/mglib	평일 : 09:00~22:00, 주말 : 09:00~22:00	매주 화요일

`iloc[:,0]` `iloc[:,1]` `iloc[:,2]`

`df['TEL_NO']` 하면 전화번호 필드 데이터만 출력됨

`df[['TEL_NO','ADDRES']]` 하면 전화번호와 주소 필드가 출력됨

`df.iloc[행번호,열번호]`, 행번호 열번호 위치에 : 이 있으면 모든 자료란 뜻

`df.iloc[:,5]` 는 전체행, 5열임 / 전화번호 필드 출력됨

`df.iloc[:, [5,4]]`는 전체행, 5열/4열 출력 (전화번호와 주소 필드 출력됨)

(참고) csv를 리스트로 읽은 자료를 json으로 변경

```
1 #####
2 ### 데이터를 dict로 구성하면 추후 자료 확인이 쉬움
3 #####
4 items=[]
5 for row in data:
6     val={'년도': row[0],
7         '시도': row[1],
8         '시군구': row[2],
9         '성별': row[3],
10        '연령구분': row[4],
11        '환자수': row[5],
12        '입내원일수': row[6],
13        '요양급여비용': row[7]}
14
15     items.append(val)
16 items[:2]
```

```
[{'년도': '2018',
  '시도': '서울',
  '시군구': '강남구',
  '성별': '남',
  '연령구분': '0~9세',
  '환자수': '57',
  '입내원일수': '121',
  '요양급여비용': '9914070'},
 {'년도': '2018',
  '시도': '서울',
  '시군구': '강남구',
  '성별': '남',
  '연령구분': '10~19세',
  '환자수': '63',
  '입내원일수': '130',
  '요양급여비용': '9030550'}]
```



파이썬에서 Numpy

- ▶ Array (행렬) 이며, 행(row)와 열(column)으로 이루어짐.
반드시 갯수와 성격이 같은 자료가 되어야함.
- ▶ 행렬을 만드는 이유는 계산을 빠르게 하기위함임.

자료의 형식은 좌측에 있는 엑셀자료와 같음을 인지해야함.

작업1) 자료불러와 제목과 자료 분리

```
신규회원=[['지점','1월','2월','3월'],  
          ['A',1,2,5],  
          ['B',3,4,1],  
          ['C',4,1,1]]
```

```
header=신규회원[0]  
data=신규회원[1:]  
data
```

header

data

	A	B	C	D
1	지점	1월	2월	3월
2	A	1	2	5
3	B	3	4	1
4	C	4	1	1

작업2) 작업하기 편리하게 인원수만 분리함

```
import numpy as np  
data=np.array(data)  
인원수=data[:,1:].astype(np.uint8)  
인원수
```

data[:, 1:]

모든행

1월이후

	A	B	C	D
1	지점	1월	2월	3월
2	A	1	2	5
3	B	3	4	1
4	C	4	1	1

작업3) 인원수 자료중 1월에만 100을 더하고자 함. Numpy 자료임으로 '브로드 캐스트' 가능함

*브로드캐스트: 각 인덱스마다 +100을 하지 않고 행,열 단위로 한번에 작업하는 기능을 의미
넘파이 자료만 브로드캐스트 가능함

	1월	2월	3월
	인원수[:,0]	인원수[:,1]	인원수[:,2]
	A	B	C
1	1	2	5
2	3	4	1
3	4	1	1

```
#####  
# 1월에만 +100을 하고자할때  
#####  
인원수[:,0]+=100  
인원수([[101, 2, 5],  
 [103, 4, 1],  
 [104, 1, 1]], dtype=uint8)
```

작업4) 2월자료에만 *2.3 을 하고자 함. dtype변경을 하지 않으면 정수자료임으로 정수만 출력됨

*형은 맨처음 입력할때 0~255의 정수만 입력하였으므로 자동 설정된 것임

구글검색 '넘파이 형선언', '넘파이 형변환'

```
신규회원=[['지점','1월','2월','3월'],  
          ['A',1,2,5],  
          ['B',3,4,1],  
          ['C',4,1,1]]
```

```
1 #####  
2 # 2월에만 *2.3을 하고자할때  
3 # dtype을 변경해야함  
4 #####  
5 인원수=인원수.astype(np.float32)  
6 인원수[:,1]=인원수[:,1]*2.5  
7 인원수  
✓ 0.0s  
  
array([[101. , 12.5,  5. ],  
       [103. , 25. ,  1. ],  
       [104. ,  5. ,  1. ]], dtype=float32)
```

```
1 #####  
2 # 2월에만 *2.3을 하고자할때  
3 # dtype이 int라 소수점이하 절사됨  
4 #####  
5 인원수[:,1]=인원수[:,1]*2.5  
6 인원수  
✓ 0.0s  
  
array([[101,  5,  5],  
       [103, 10,  1],  
       [104,  2,  1]], dtype=uint8)
```

정수형중 0~255까지 숫자

작업5) C지점에만 전체 -100을 하고자 할때

```
1 #####
2 # C지점에만 전체 -100을 하고자 할때
3 #####
4 인원수[2,:] = 인원수[2,:]-100
5 인원수
```

✓ 0.0s

```
array([[101. , 12.5,  5. ],
       [103. , 25. ,  1. ],
       [  4. , -95. , -99. ]], dtype=float32)
```

인원수[0, :]
인원수[1, :]
인원수[2, :]

	A	B	C
1	1	2	5
2	3	4	1
3	4	1	1

신규회원=[['지점','1월','2월','3월'],

['A',1,2,5],
['B',3,4,1],
['C',4,1,1]]

작업6) 인덱싱 마무리

```
1 #####
2 # B지점의 3월자료를 100으로 변경하고자할때
3 #####
4 인원수[1,2]=100
5 인원수
✓ 0.0s

array([[101. , 12.5,  5. ],
       [103. , 25. , 100. ],
       [  4. , -95. , -99. ]], dtype=float32)
```

```
1 #####
2 # 2월자료를 모두 50으로 변경하고자 할때
3 #####
4 인원수[:,1]=50
5 인원수
✓ 0.0s

array([[101., 50.,  5.],
       [103., 50., 100.],
       [  4., 50., -99.]], dtype=float32)
```

		1월	2월	3월
A		A	B	C
	1	0,0	0,1	0,2
	2	1,0	1,1	1,2
	3	2,0	2,1	2,2

작업7) Numpy 조건문

이 PT 자 보고난뒤에 별도로 'Numpy 조건문' 으로 검색하여 조건문을 인지합니다.
2개이상의 조건을 동시에 만족하는 And, 조건중 한 개만 만족하는 Or에 대한 문법도 확인합니다.

```
1 #####
2 # 자료중 50이상인 값만 999로 변경하고자할때
3 #####
4 인원수[인원수[:, :]>50]=999
5 인원수
```

✓ 0.0s

```
array([[999., 50., 5.],
       [999., 50., 999.],
       [ 4., 50., -99.]], dtype=float32)
```

```
1 #####
2 # 위의 공식은 np.where 로 대체될수 있음
3 #####
4 인원수[np.where(인원수>50)]=999
5
6 인원수
7
```

✓ 0.0s

```
array([[999., 50., 5.],
       [999., 50., 999.],
       [ 4., 50., -99.]], dtype=float32)
```

작업8) Numpy를 이용하면 편리한 계산식 (브로드캐스트와 동시와 계산식도 제공됨)

브로드 캐스트 및 연산식(합,평균,최대,최소등) 제공함

np.mean→평균

```
1  ### 총인원수 몇명인지 알고 싶을때
2  ### 인원수당*인센티브한자료의 합
3  ### 평균등 계산
4  np.sum(인원수), np.sum(인원수*5000), np.mean(인원수), np.mean(인원수*5000)
```

✓ 0.0s

(3057.0, 15285000.0, 339.66666, 1698333.4)

작업9) 월별인센티브를 넣어서 각 행에 곱셈 가능함.

```
1 #####3
2 ## 월별 인센티브 가 다를때
3 #####
4 월별인센티브=[20,
5 |   |   | 50,
6 |   |   | 15] #1월은 20, 2월은 50, 3월은 15일때
7 인원수, 월별인센티브, 인원수*월별인센티브
```

✓ 0.0s

```
(array([[999., 50., 5.],
       [999., 50., 999.],
       [ 4., 50., -99.]], dtype=float32),
 [20, 50, 15],
 array([[19980., 2500., 75.],
       [19980., 2500., 14985.],
       [ 80., 2500., -1485.])))
```

Array shape 가 통일되지 않으면 연산되지 않는 문제

작업10) 내적곱(np.dot) 에 대한 이해 : 곱셈한뒤 그 곱셈의 값을 행단위로 더하는것

A,B,C 지점에 월별 인센티브를 더해서 1,2,3월의 총합을 구하고자 할때
계산식을 나누어 작업해도 되지만 (곱셈후, 행단위의 합계) np.dot 기능을 이용하여 한번에 작업가능함

인원수

각월별인센티브

인원수*각월별인센티브

내적곱결과

1월 2월 3월

$\begin{bmatrix} 3, & 4, & 5 \\ 2, & 3, & 4 \end{bmatrix}$

2행*3열

$\begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$

3행*1열

1월에 곱할값
2월에 곱할값
3월에 곱할값

→

$\begin{bmatrix} 3*100, & 4*200, & 5*200 \\ 2*100, & 3*200, & 4*300 \end{bmatrix}$

2600

2000

np.dot(a,b)가 되려면 반드시
a의 열과 b의 행값이 같아야함.
결과는 a의 행* b의 열임

```
1 a,b=np.array(a), np.array(b)  
2 a*b, np.sum(a*b, axis=1), np.dot(a,b)
```

✓ 0.0s

```
(array([[ 300,  800, 1500],  
        [ 200,  600, 1200]]),  
 array([2600, 2000]),  
 array([2600, 2000]))
```

결과물은
2행*1열임

작업10) 내적곱(np.dot) 에 대한 이해 : 곱셈한뒤 그 곱셈의 값을 행단위로 더하는것

행열이 다르면 np.dot가 실행되지 않음.

```
> 1 a=[[3,4,5],
2     |   | [2,3,4]]
3 b=[[100,200,300]]
4
5 print(np.shape(a), np.shape(b))
6 np.dot(a,b)

[87] (x) 0.0s

.. (2, 3) (1, 3)

.. -----
ValueError                                Traceback (most recent call last)
Cell In[87], line 6
      3 b=[[100,200,300]]
      5 print(np.shape(a), np.shape(b))
----> 6 np.dot(a,b)

ValueError: shapes (2,3) and (1,3) not aligned: 3 (dim 1) != 1 (dim 0)
```

작업10) 내적곱(np.dot) 에 대한 이해 : 곱셈한뒤 그 곱셈의 값을 행단위로 더하는것

이때 reshape을 통해서 행열값을 변경하고 연산가능함.

```
1 a=[3,4,5],  
2   [2,3,4]  
3 b=[100,200,300]  
4 c=np.reshape(b, (3,1))  
5 print(np.shape(a), np.shape(b)), np.shape(c)  
6 np.dot(a,c)
```

[88] ✓ 0.0s

.. (2, 3) (1, 3)

.. array([[2600],
 [2000]])

작업11) array 끼리 추가(열단위, 행단위) 추가도 array의 shape이 일치해야함.

hstack: 열(세로로) 추가할때

```
1 #####
2 ### 4,5,6월자료가 추가될때
3 #####
4 import numpy as np
5 추가= [ [1,4,5],
6         [2,2,3],
7         [7,8,9] ]
8
9 인원수=np.hstack((인원수,추가))
10 인원수
```

✓ 0.0s

```
array([[999.,  50.,   5.,   1.,   4.,   5.],
       [999.,  50., 999.,   2.,   2.,   3.],
       [  4.,  50., -99.,   7.,   8.,   9.]])
```


작업11) array 끼리 추가(열단위, 행단위) 추가도 array의 shape이 일치해야함.

vstack: 열(세로로) 추가할때

```
1 #####
2 ## E지점의 1월~6월자료를 추가하고자 할때
3 #####
4
5 E지점=[[1,2,1,1,1,1]]
6 np.vstack((인원수,E지점))
```

✓ 0.0s

```
array([[999.,  50.,   5.,   1.,   4.,   5.],
       [999.,  50., 999.,   2.,   2.,   3.],
       [  4.,  50., -99.,   7.,   8.,   9.],
       [  1.,   2.,   1.,   1.,   1.,   1.]])
```