

Numpy 이해

강사: 부성순(bakpak@empas.com)

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
```

0s

```
([[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.]])
```