

## HW4 – Numpy와 Pandas 활용 과제

201723365 – 정회진

### 1 -> Numpy 기초 연습 문제

In [6]:

```
import numpy as np
```

- 1. "np.zeros"를 이용해, 행=3, 열=4인 ndarray를 생성합니다.
- 이 때 생성된 ndarray의 이름은 a가 되도록 합니다.
- zero()함수는 0으로 초기화 시켜 shape의 형태로 반환합니다.

In [7]:

```
a = np.zeros((3, 4))  
a
```

Out[7]:

```
array([[0., 0., 0., 0.],  
       [0., 0., 0., 0.],  
       [0., 0., 0., 0.]])
```

- 1. 생성한 ndarray a에 대해, 다중 for 문을 작성해서,아래와 같은 값을 갖도록 초기화하는 과정
- shape은 행열의 차원으로 보여줍니다.

In [48]:

```
for i in range(a.shape[0]):  
    for j in range(a.shape[1]):  
        a[i, j] = i*4 + j + 1  
a
```

Out[48]:

```
array([[1., 2., 3., 4.],  
       [5., 6., 7., 8.],  
       [9., 10., 11., 12.]])
```

In [49]:

```
num = 1
for i in range(a.shape[0]):
    for j in range(a.shape[1]):
        a[i, j] = num
        num += 1
a
```

Out[49]:

```
array([[ 1.,  2.,  3.,  4.],
       [ 5.,  6.,  7.,  8.],
       [ 9., 10., 11., 12.]])
```

- 1. 초기화한 a로 값 7을 꺼내도록 인덱싱하는 과정

In [51]:

a[1,2]

Out[51]:

7.0

- 1. a에서 아래와 같이 둘째 행을 꺼내 인덱싱 하는 과정입니다.
- 단 차원은 유지

In [53]:

a.shape

Out[53]:

(3, 4)

In [54]:

a[[1],:]

Out[54]:

array([[5., 6., 7., 8.]])

- 1. a에서 아래와 같이 둘째 열과 셋째 열이 나오도록 인덱싱 하는 과정입니다.

In [16]:

a[:, [0, 2]]

Out[16]:

```
array([[ 1.,  3.],
       [ 5.,  7.],
       [ 9., 11.]])
```

- 1. 아래와 같이 a에서 6보다 작거나 같은 값들이 나오도록 불리안 인덱싱을 하는 과정입니다.

In [18]:

```
result = np.asarray([a[i, j] for i in range(a.shape[0]) for j in range(a.shape[1]) if a[i, j] <= 6])  
result
```

Out[18]:

```
array([1., 2., 3., 4., 5., 6.])
```

In [19]:

```
a[a <= 6]
```

Out[19]:

```
array([1., 2., 3., 4., 5., 6.])
```

- 1. "np.random" 패키지를 이용하여 0에서 1사이의 값이 무작위로 생성 후
- 행이 3 열이 4인 ndarray를 생성합니다.
- 이 때 생성된 ndarray의 이름은 b가 되도록 합니다.

In [20]:

```
b = np.random.random((3, 4))  
b
```

Out[20]:

```
array([[0.09709642, 0.92996452, 0.66913222, 0.0742523 ],  
       [0.468209 , 0.21306448, 0.42794535, 0.77430731],  
       [0.72236693, 0.2219291 , 0.98473634, 0.3615838 ]])
```

- 1. a와 b의 요소곱을 하여 c에 저장하는 과정입니다.

In [22]:

```
c = a * b  
c
```

Out[22]:

```
array([[ 0.09709642,  1.85992903,  2.00739665,  0.29700921],  
       [ 2.34104499,  1.27838686,  2.99561742,  6.19445848],  
       [ 6.50130234,  2.21929104, 10.83209974,  4.33900559]])
```

- 1. b의 전치행렬을 구하여 d에 저장합니다.

In [25]:

```
d = b.T  
d
```

Out[25]:

```
array([[0.09709642, 0.468209 , 0.72236693],  
       [0.92996452, 0.21306448, 0.2219291 ],  
       [0.66913222, 0.42794535, 0.98473634],  
       [0.0742523 , 0.77430731, 0.3615838 ]])
```

In [26]:

```
d.shape
```

Out[26]:

(4, 3)

- 1. a와 d에 대하여 각 차원의 크기를 출력합니다.

In [28]:

```
print(a.shape, d.shape)
```

(3, 4) (4, 3)

- 1. a와 d의 행렬곱을 계산하여 e에 저장합니다.

In [30]:

```
e = a.dot(d)  
e
```

Out[30]:

```
array([[ 4.26143131,  5.27540323,  5.56676935],  
       [11.34321312, 12.80950775, 14.72923403],  
       [18.42499492, 20.34361227, 23.89169871]])
```

- 1. e에 대하여 각 열의 합을 계산합니다.

In [32]:

```
np.sum(e, axis=1)
```

Out[32]:

```
array([15.10360389, 38.8819549 , 62.6603059 ])
```

## 2 -> Pandas 기초연습문제

In [2]:

```
import pandas as pd
```

- 1. 딕셔너리를 이용해 data frame을 생성합니다.

In [3]:

```
score={'학번':[201524213,201621223,201621233, 201611111, 201722222],
      '이름':['오지환','박용택','정성훈', '다니엘', '지민'],
      '학과':['경영정보','경영정보','경영정보','컴퓨터공학','컴퓨터공학'],
      'Quiz':[15,20,10,17,18],
      '중간고사':[20,20,15,18,16]}
```

```
df= pd.DataFrame(score)
df
```

Out[3]:

	학번	이름	학과	Quiz	중간고사
0	201524213	오지환	경영정보	15	20
1	201621223	박용택	경영정보	20	20
2	201621233	정성훈	경영정보	10	15
3	201611111	다니엘	컴퓨터공학	17	18
4	201722222	지민	컴퓨터공학	18	16

In [4]:

```
df['학번']
```

Out[4]:

```
0    201524213
1    201621223
2    201621233
3    201611111
4    201722222
Name: 학번, dtype: int64
```

- 1. 생성한 데이터 프레임으로 인덱싱하여 이름과 중간고사를 추출하는 과정입니다.

In [5]:

```
df[['이름', '중간고사']]
```

Out[5]:

	이름	중간고사
0	오지환	20
1	박용택	20
2	정성훈	15
3	다니엘	18
4	지민	16

- 1. re index를 이용하여 2, 0, 4, 1, 3 순으로 순서를 변경
- index 자체가 객체를 변경 할 수 없어서, 재색인(reindex를 이용)
- reindex는 새로운 index에 맞게, 새로운 객체를 생성합니다. (행과열 모두 변경 가능합니다)

In [6]:

```
df = df.reindex([2, 0, 4, 1, 3])
df
```

Out[6]:

	학번	이름	학과	Quiz	중간고사
2	201621233	정성훈	경영정보	10	15
0	201524213	오지환	경영정보	15	20
4	201722222	지민	컴퓨터공학	18	16
1	201621223	박용택	경영정보	20	20
3	201611111	다니엘	컴퓨터공학	17	18

- 1. iloc과 loc을 각각 이용하여 박용택 학생의 record를 추출하시오.
- iclo = 위치를 이용한 정수 인덱싱을 의미합니다.
- loc= 레이블을 베이스로 이용하여 인덱싱합니다.

In [8]:

```
print(df.iloc[3])
print(df.loc[1])
```

```
학번    201621223
이름      박용택
학과      경영정보
Quiz      20
중간고사    20
Name: 1, dtype: object
학번    201621223
이름      박용택
학과      경영정보
Quiz      20
중간고사    20
Name: 1, dtype: object
```

- 1. iloc과 loc을 각각 이용하여 모든 학생의 이름과 중간고사 점수를 추출하시오.
- iclo = 위치를 이용한 정수 인덱싱을 의미합니다.
- loc= 레이블을 베이스로 이용하여 인덱싱합니다.

In [10]:

```
print(df.iloc[:, [1, 4]])
print(df.loc[:, ['이름', '중간고사']])
```

```
이름 중간고사
2 정성훈 15
0 오지환 20
4 지민 16
1 박용택 20
3 다니엘 18
이름 중간고사
2 정성훈 15
0 오지환 20
4 지민 16
1 박용택 20
3 다니엘 18
```

- 1. 중간고사 점수의 평균 계산

In [12]:

```
df['중간고사'].mean()
```

Out[12]:

17.8

- 1. 중간고사 점수가 가장 낮은 학생의 index 계산 과정

In [14]:

```
df['중간고사'].idxmin()
```

Out[14]:

2

- 1. 퀴즈 점수가 가장 높은 학생의 학번 계산

In [16]:

```
df.loc[df['Quiz'].idxmax(), '학번']
```

Out[16]:

201621223

- 1. 위 데이터 프레임의 각 열에 대해서 개수, 평균, 표준편차, 최소최대값 등의 다양한 통계량을 출력

In [17]:

df.describe()

Out[17]:

	학번	Quiz	중간고사
count	5.000000e+00	5.000000	5.000000
mean	2.016200e+08	16.000000	17.800000
std	7.018995e+04	3.807887	2.280351
min	2.015242e+08	10.000000	15.000000
25%	2.016111e+08	15.000000	16.000000
50%	2.016212e+08	17.000000	18.000000
75%	2.016212e+08	18.000000	20.000000
max	2.017222e+08	20.000000	20.000000

- 1. 위 데이터프레임을 중간고사 점수에 따라 내림차순으로 정렬합니다.

In [19]:

df.sort\_values('중간고사', ascending=False)

Out[19]:

	학번	이름	학과	Quiz	중간고사
0	201524213	오지환	경영정보	15	20
1	201621223	박용택	경영정보	20	20
3	201611111	다니엘	컴퓨터공학	17	18
4	201722222	지민	컴퓨터공학	18	16
2	201621233	정성훈	경영정보	10	15

- 1. 학생 중에서 퀴즈 점수가 평균 이상인 학생들의 이름을 출력 과정

In [20]:

```
quiz_mean = df['Quiz'].mean()
df[df['Quiz'] > quiz_mean]['이름']
```

Out[20]:

```
4 지민
1 박용택
3 다니엘
Name: 이름, dtype: object
```

- 1. 학생 중에서 퀴즈와 중간고사 점수 모두 평균 이상인 학생들의 record를 출력합니다.



In [22]:

```
quiz_mean = df['Quiz'].mean()
mid_mean = df['중간고사'].mean()
df[(df['Quiz'] > quiz_mean) & (df['중간고사'] > mid_mean)]
```

Out[22]:

	학번	이름	학과	Quiz	중간고사
1	201621223	박용택	경영정보	20	20
3	201611111	다니엘	컴퓨터공학	17	18

- 1. value\_counts를 이용하여 학과 별로 학생의 수를 계산합니다.

In [24]:

```
df['학과'].value_counts()
```

Out[24]:

```
경영정보    3
컴퓨터공학  2
Name: 학과, dtype: int64
```

- 1. groupby를 이용하여 학과 별로 중간고사의 평균 계산하는 과정입니다.

In [25]:

```
df.groupby('학과')['중간고사'].mean()
```

Out[25]:

```
학과
경영정보    18.333333
컴퓨터공학  17.000000
Name: 중간고사, dtype: float64
```

In [26]:

```
df.mean()
```

Out[26]:

```
학번    201620000.4
Quiz      16.0
중간고사    17.8
dtype: float64
```

In [27]:

```
list(df.groupby('학과'))
```

Out[27]:

```
[('경영정보',  
  학번 이름   학과 Quiz 중간고사  
  2 201621233 정성훈 경영정보 10 15  
  0 201524213 오지환 경영정보 15 20  
  1 201621223 박용택 경영정보 20 20),  
 ('컴퓨터공학',  
  학번 이름   학과 Quiz 중간고사  
  4 201722222 지민  컴퓨터공학 18 16  
  3 201611111 다니엘 컴퓨터공학 17 18)]
```