NumPy (3) 배열 연산

1차원 배열 : 벡터

2차원 배열 : 행렬

""" 벡터화 연산 """

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ 10000 \end{bmatrix} + \begin{bmatrix} 10001 \\ 10002 \\ 10003 \\ \vdots \\ 20000 \end{bmatrix} = \begin{bmatrix} 1+10001 \\ 2+10002 \\ 3+10003 \\ \vdots \\ 10000+20000 \end{bmatrix} = \begin{bmatrix} 10002 \\ 10004 \\ 10006 \\ \vdots \\ 30000 \end{bmatrix}$$

x + y

```
import  numpy as np

x = np.arange(1, 10001)
y = np.arange(10001, 20001)

"""(사칙 연산) """

z = x + y
print(z)
```

실행결과

```
[10002 10004 10006 ... 29996 29998 30000]
```

```
""" """
a = np.array([1, 2, 3, 4])
print(10 ** a )  #10의 a 승
```

실행결과

[10 100 1000 10000]

```
"""(비교 연산,  논리 연산)"""
a = np.array([1, 2, 3, 4])
b = np.array([4, 2, 2, 4])

print((a >= b)  & (a == 4))
print((a >= b ) | (a == b ))
```

실행결과

[False False False True]
[False True True True]

```
""" 배열의 모든 원소가 다 같은지 알고 싶다면 all 명령 ."""

print(  np.all(a == b)  )
```

실행결과

False

```
"""지수 함수, 로그 함수 """
print( np.exp(a  )  )
print( np.log(a + 1) )
```

실행결과

[2.71828183 7.3890561 20.08553692 54.59815003]
[0.69314718 1.09861229 1.38629436 1.60943791]

```
"""
정렬

sort메소드: 배열 안의 원소를 크기에 따라 정렬
2차원 이상인 경우 :
 axis=0 이면 각각의 행을 따로따로 정렬
 axis=1이면 각각의 열을 따로따로 정렬 .
 디폴트 값은 -1 즉 가장 안쪽(나중)의 차원
"""
a = np.array([[4,  3,  5,  7],
              [1, 12, 11,  9],
              [2, 15,  1, 14]])

print( np.sort(a) )
print( np.sort(a, axis=0)  ) #행 정렬

""" 순서만 알고 싶다면 argsort """
a = np.array([42, 38, 12, 25])
j = np.argsort(a)
print ( j  )
print (a[j])   #정렬된 순서로 출력됨

"""
```

실행결과

[[3 4 5 7]
 [1 9 11 12]
 [1 2 14 15]]

[[1 3 1 7]
 [2 12 5 9]
 [4 15 11 14]]

[2 3 1 0]
[12 25 38 42]