

데이터(Data) -> 반드시 숫자화가 되어야 한다.

종류

1. 소리 데이터
2. 이미지, 영상 데이터
3. 텍스트 데이터
4. 숫자 데이터 (날씨, 온도 등) -> 이걸 바로 이해할 수 있음

Q. 그렇다면, 숫자화 되지 않는 데이터들 (1~3번) 어떻게 숫자화할 수 있나?

A. 벡터, 행렬에 대한 설명이 필요!

숫자를 주르륵 나열 (숫자열)=> **벡터**

e.g. 1573628593252748949....

직사각형 형태 -> 행(세로), 열(가로) => 합쳐서 **행렬**

e.g. 컴퓨터 상의 이미지 0 = 검정, 10 = 하양색이면 1~9 그 중간색

10392748539

92738493828

28377563001

+색깔

직사각형 3개가 쌓임 (기본색깔 R, B, G) / 1개일 때 (흑백)

동영상?

색깔 + 시간

차원의 세계

흑백 이미지 -> 2차원 (행, 열)

색깔 이미지 -> 3차원 (행, 열, 색)

동영상 -> 4차원 (행, 열, 색, 시간)

소리?

➔ 파형에 각각 부분에 점을 찍고, 그 점 위치를 숫자로 바꿔서 벡터화

텍스트?

사전이 있다고 가정 / 5만개 단어

1번째 단어 벡터화 100000 (x5만) 00000

2번째 단어 벡터화 010000 (x5만) 00000

...

5만번째 단어 벡터화 000000 (x5만) 00001

Numpy -> 숫자처리 (사칙연산) 시키는 도구

a = [1,3,5]

b = [2,4,6]

c = a+b

c

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

➔ List 인 상태일 때는 사칙연산 자체가 불가하다 (수학적 처리가 필요하다)

여기서 사용되는 도구인 **numpy**

import numpy as np -> numpy 를 도입하고, 앞으로 모든 numpy는 np로 사용가능

/ 가장 큰 파일인 numpy

A = numpy.array(a) -> numpy 안에 있는 array 함수(배열 함수)가 계산가능한 요소로 전환

B = numpy.array(b)

A + B

array([3,7,11]) -> 위의 list 결과와 달리 2개의 값들이 서로 더해졌다

type(A) -> A의 type는?

numpy.ndarray -> 다차원 배열(n-dimensional array)

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

X.shape -> X의 모양은 어떻게 되는가?

(2,3) -> X는 2행 3열의 matrix이다

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

```
= from matplotlib import pyplot as plt
```

➔ Matplotlib이 pyplot을 포함함

```
np.empty([2,3], dtype='int')
```

➔ empty함수는 2행 3열의 행렬과 dtype을 int(정수)로 바꿔라

```
array([[          0, 1072168960,           0],
       [1072168960,           0,           0]])
```

-> 여기에 있는 숫자들은 모두 random한 숫자

```
np.zeros([2,3])
```

zeros 함수 2행 3열 행렬 -> 0으로 다 채워진 함수

```
np.ones([2,3])
```

ones 함수 2행 3열 행렬 -> 1로 다 채워진 함수

➔ float 상태인데(소수점 있음) 이걸 정수형태로 바꾸고 싶으면, 함수 내에 dtype = 'int' 추가

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

array 함수는 행렬화, matrix화 시켜주는 함수이다 -> "안의 값들을 행렬화시켜라!"

```
np.arange(5)
```

"5개의 index 만들어라"

```
array([0,1,2,3,4])
```

np 안에 있는 arange 함수 -> index를 형성해준다.

```
np.arange(0,10) -> index를 만들되, 0부터 9까지 (10 이전)
```

```
array([0,1,2,3,4,5,6,7,8,9])
```

```
np.arange(0,10,2) -> "0부터 9까지 2만큼씩 간격뒤서 숫자 index 만들어라"
```

```
array([0,2,4,6,8])
```

```
np.linspace(0,10,6) -> 0부터 10까지 6등분 해준다 (10 포함!) / linspace
```

```
array([ 0.,  2.,  4.,  6.,  8., 10.])
```

괄호 수에 따라서 차원수가 달라짐!

```
np.array([[1,2,3], [4,5,6]]) -> 2행 3열의 2차원 / 대괄호 2개
```

```
np.array([[1,2,3],[4,5,6]],[[2,5,7],[6,8,10]]) -> 2행 3열이 2개가 포개져 있는 3차원 / 대괄호 3개
```

```
x = np.array([[1,2,3],[4,5,6]],[[2,5,7],[6,8,10]])
```

x

```
array([[1,2,3],
```

```
      [4,5,6],
```

```
      [[2,5,7],
```

```
      [6,8,10]])
```

x.ndim -> x는 몇차원?

3

x.shape

(2,2,3)

➔ 행렬 2개가 포개져 있는 2행 3열의 데이터

x.dtype -> 숫자들의 type은?

dtype('int32')

x.astype(np.float64) -> 숫자들 type을 float64로 바꿔줘라

np.zeros_like(x) -> 모든 숫자값을 0으로 바꿔줘라

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

data = np.random.normal(0,1,100)

➔ numpy 라는 library package 에 random이라는 subpackage가 존재하며, 거기 안의 **normal 함수**를 사용

(0 = mean(평균), 1 = standard deviation(표준편차), 100 (100개 숫자 random하게 만들어))

print(data)

plt.hist(data, bins=10)

➔ plot에서 **hist 함수** -> histogram (히스토그램) / 거기서 바구니 (bins) 10개 있고, 랜덤하게 data에 있던 숫자들을 던져본다 (**y값**은 무조건 **정수값**이 나와야 한다 > 몇 개 던졌는데?)

plt.show()

data.ndim -> 몇 차원

1

data.shape -> 어떤 모양

(100,)

```
x = np.ones([2,3,4])
```

x

```
array([[[[1., 1., 1., 1.],
         [1., 1., 1., 1.],
         [1., 1., 1., 1.]],
       [[1., 1., 1., 1.],
         [1., 1., 1., 1.],
         [1., 1., 1., 1.]])])
```

```
y = x.reshape([-1,3,2])
```

➔ reshape 함수는 x 의 모양을 바꾼다 (곱이 24가 되는 한에서만 바꾸기 가능

/ 1, 6, 4 or 3, 2, 4 등등)

➔ -1 (니가 알아서 조정해라) 여기서 원래 값은 4인데 귀찮을 때

y

```
array([[[1., 1.],
        [1., 1.],
        [1., 1.]],
       [[1., 1.],
        [1., 1.],
        [1., 1.]])
```

```
[1., 1.]],
```

```
[[1., 1.],
```

```
[1., 1.],
```

```
[1., 1.]],
```

```
[[1., 1.],
```

```
[1., 1.],
```

```
[1., 1.]])
```

`np.allclose(x.reshape(-1,3,2),y) -> 둘이 유사하냐? 똑같은가?`

True

```
a = np.random.randint(0,10,[2,3])
```

➔ **randint 함수:** "입력된 0~9까지의 숫자들을 random하게 2행 3열로 만들어"

```
b = np.random.random([2,3])
```

➔ **random 함수:** "0부터 1사이의 float를 random하게 2행 3열로 만들어라"

```
np.savez('test', a,b)
```

savez 함수 -> 실제 파일로 저장을 해줌

a와 b를 저장해줌 'test' 라는 이름으로

test.npz로 만들어짐

`del a,b -> variable 을 지우자`

`%who -> assign되어있는 variable, available한 variable은 뭐가 있나?`

```
npzfiles = np.load("test.npz")
```

```
npzfiles.files
```

```
['arr_0', 'arr_1']
```

```
npzfiles['arr_0']
```

```
array([[5, 9, 1],  
       [8, 3, 0]])
```

```
arr = np.random.random([5,2,3]) /
```

```
type(arr)
```

```
numpy.ndarray
```

```
a
```

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

a.sum() -> a에 있는 것을 모두 더해라 / 이미 np에 있기 때문에 sum 이 가능함

sum 함수 ()

np.sum(a) -> np 속의 sum을 해야 되는데, 그 중에서 a를 sum하라

sum(axis=0) / 몇번째 차원에서 모조리 더할 것인가? 0=1차원 (세로차원(행)에서)

```
array([12,15,18])
```

```
arr = np. random.random([5,2,3])
```

len(arr) -> 길이가 어떻게 됨

5 -> 5개 포개져 있는만큼의 길이

`print(arr.shape)` -> 배열이 어떤 모양

(5,2,3) -> 5개의 2행 3열이 포개져있다

`print(arr.ndim)` -> 몇차원?

3 -> 3차원

`print(arr.size)` -> 배열의 사이즈

$5 \times 2 \times 3 = 30$

`print(arr.dtype)` -> 배열의 숫자값의 type

float64