김재현

Numpy(Numerical Python)란

- 데이터 분석시 기초 라이브러리로 사용
- Numpy는 C언어로 구현된 파이썬 라이브러리
- 고성능의 수치계산을 위해 제작
- 벡터 및 행렬 연산에 있어서 매우 편리한 기능을 제공
- Pandas와 함께 사용하여 고수준의 데이터 분석을 수행

Numpy를 사용하는 이유

- 성능 : 파이썬 리스트보다 빠름
- 메모리 사이즈 : 파이썬 리스트보다 적은 메모리 사용
- 빌트인 함수 : 선형대수, 통계관련 여러 함수 내장

Numpy

```
import numpy as np
```

ndarray

1	2	3	4

1차원 배열

numpy.array([1, 2, 3, 4])

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

2차원 배열

```
numpy.array([[1, 2, 3, 4],
             [5, 6, 7, 8],
             [9, 10, 11, 12],
             [13, 14, 15, 16]])
```

ndarray

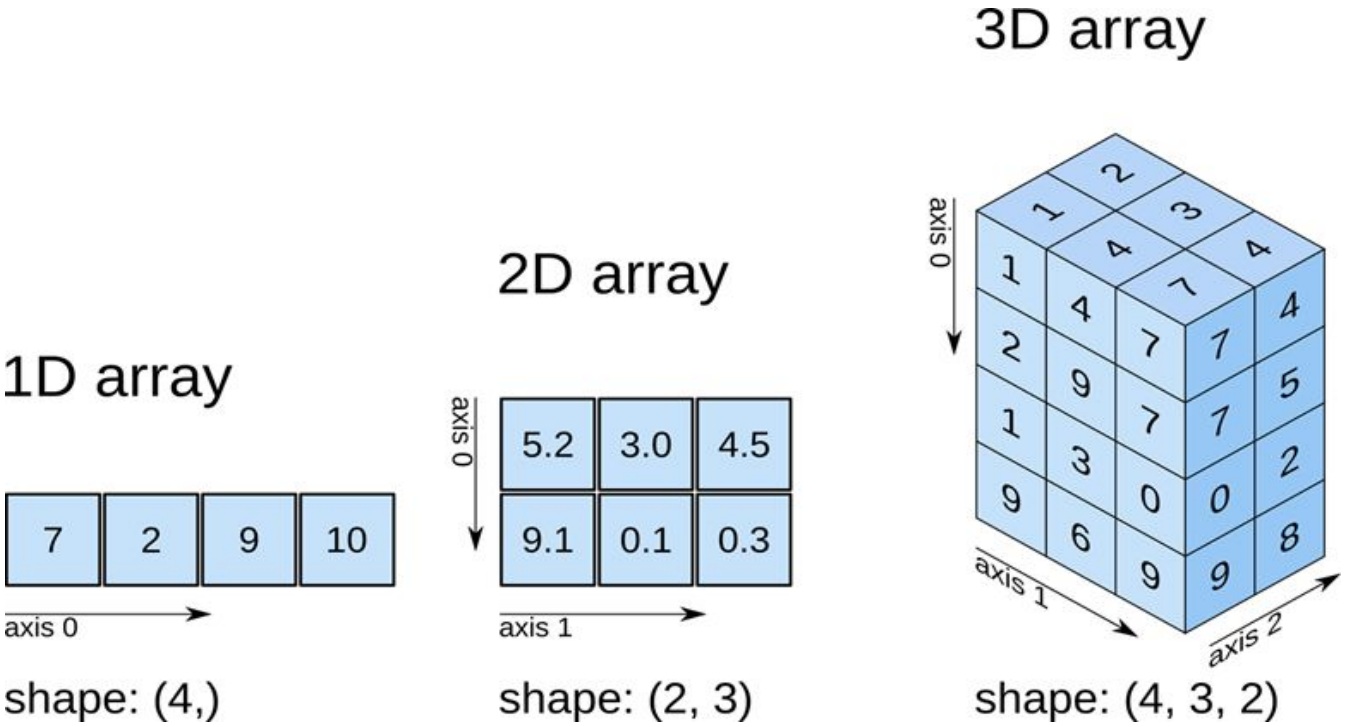
ndarray → n dimensionarray

ndarray

numpy에서 사용되는 다차원 리스트를

표현할 때 사용되는 데이터 타입

ndarray


1D array
7
2
9
10
axis 0
shape: (4,)
2D array
axis 0
5.2
3.0
4.5
9.1
0.1
0.3
axis 1
shape: (2, 3)
3D array
axis 0
axis 1
axis 2
shape: (4, 3, 2)

ndarray

ndarray에서 shape와 axis개념은

명확히 이해할 것

shape(차원)

차원의 수를 확인할 때 사용

shape(차원)

(3,)　　:　3 개의 배열

(4, 3)　　:　4 X 3의 배열

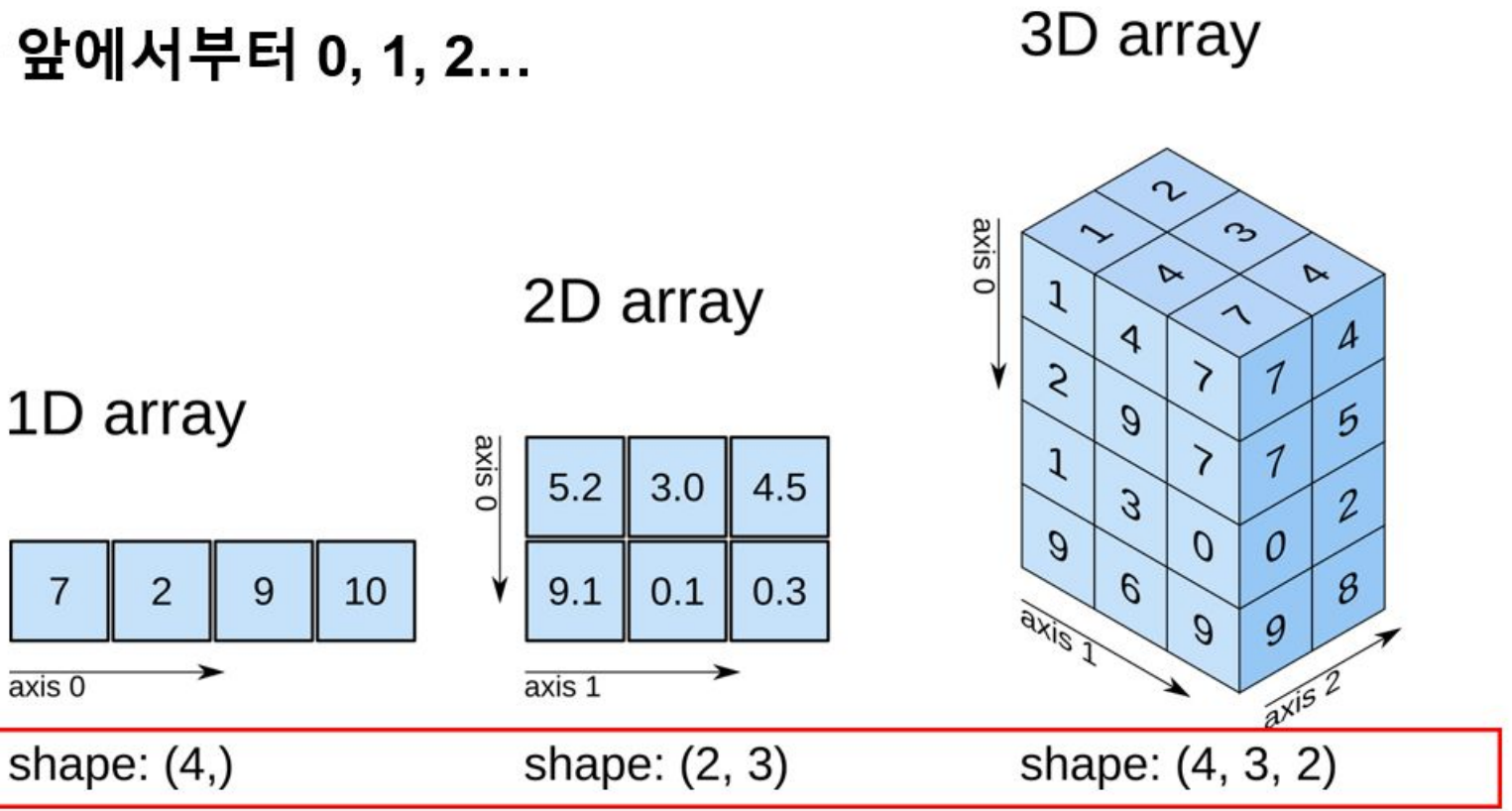
(2, 5, 3)　:　2 X 5 X 3의 배열

axis

기준이 되는 축을 의미

shape & axis

axis는 앞에서부터 0, 1, 2…



Matrix

Matrix

행(row) + 열(column)

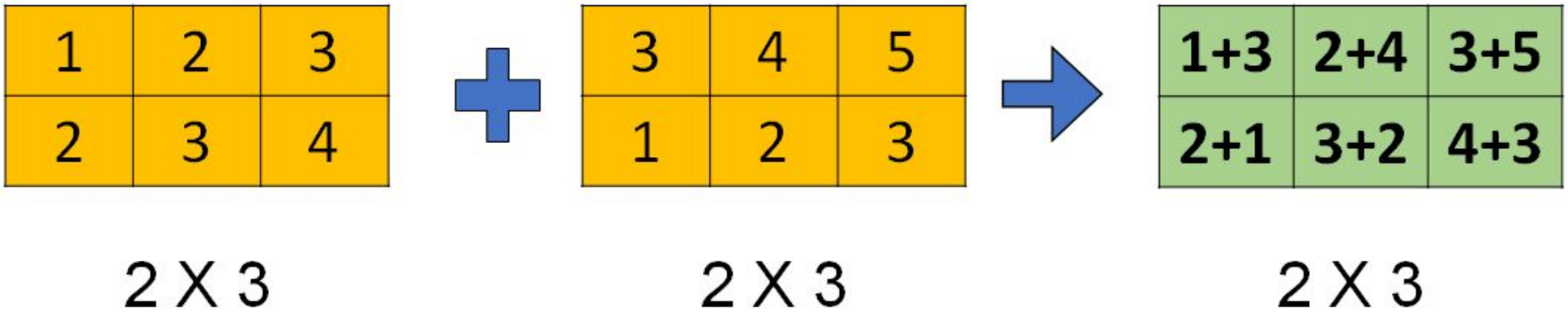
Matrix

Matrix 없이는

데이터 분석, 머신러닝(딥러닝)을 할 수 없다

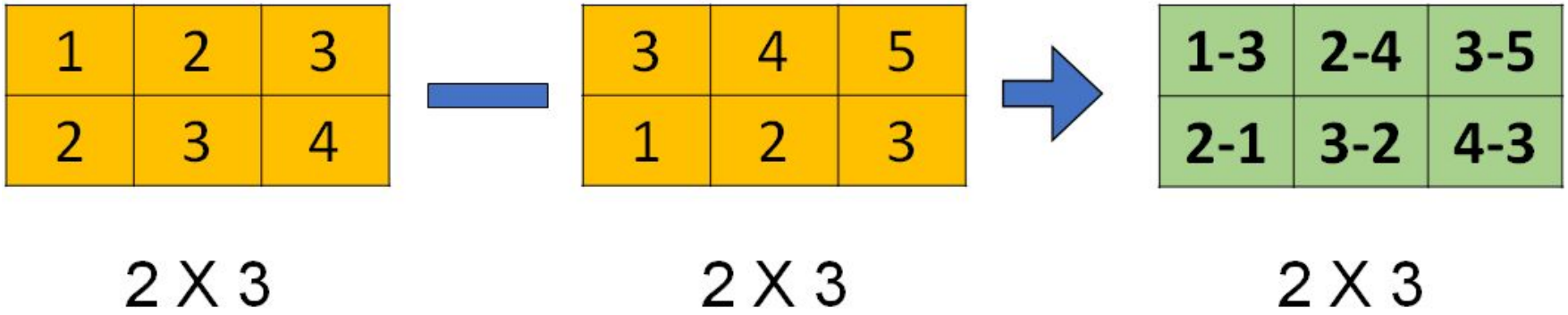
Matrix(행렬)의 연산 : 덧셈

- shape이 같아야 한다.
- 같은 position끼리 연산



Matrix(행렬)의 연산 : 뺄셈

- shape이 같아야 한다.
- 같은 position끼리 연산



Matrix(행렬)의 연산 : 곱셈(dot product)

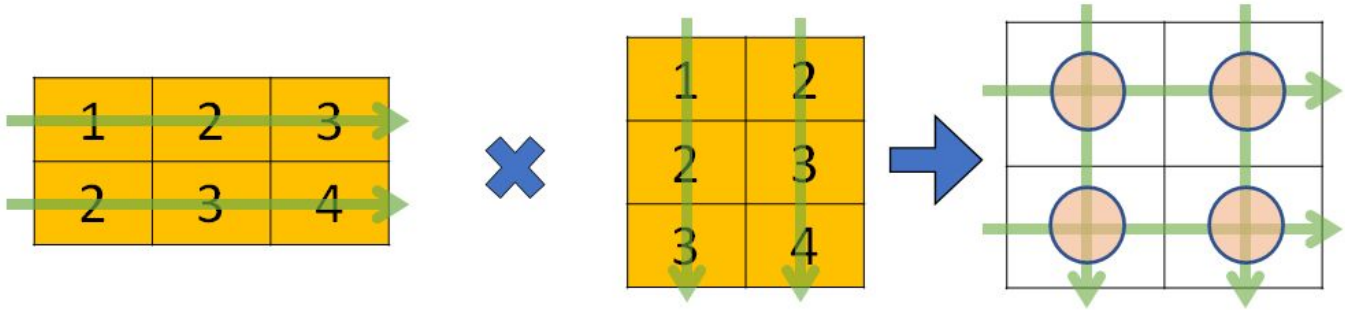
- **맞닿는 shape**이 같아야 곱셈 연산이 가능하다.

연산 불가

2 X **3** **2** X 3

← 단순 곱셈은 연산이 이루어짐

연산 가능



2 X **3** **3** X 2

Matrix(행렬)의 연산 : 곱셈 연산의 결과