넘파이 **(NumPy)**

Numerical Python을 의미하는 넘파이(NumPy)는 파이썬에서 선형대수 기반의 프로그램을 쉽게 만들 수 있도록 지원하는 대표적인 패키지이다.

- 루프를 사용하지 않고 대량 데이터의 배열 연산을 가능하게 하므로 빠른 배열 연산 속도를 보장한다.
- 기존 C/C++ 기반의 타 프로그램과 데이터를 주고받거나 API를 호출해 쉽게 통합할 수 있는 기능을 제공한다.

→ 구글의 대표적인 딥러닝 프레임워크인 텐서플로는 이러한 방식으로 배열 연산 수행 속도를 개선하고 다른 프로그램들과도 호환할 수 있게 작성됐다.

- 넘파이는 배열 기반의 연산은 물론이고 다양한 데이터 핸들링 기능을 제공한다.
  → 판다스에 비해 불편하다.

많은 머신러닝 알고리즘이 넘파이 기반으로 작성 돼 있음은 물론이고, 이들 알고리즘의 입력 데이터와 출력 데이터를 넘파이 배열 타입으로 사용하기 때문에 머신러닝에서 넘파이를 이해하는 것은 매우 중요하다.

넘파이의 기반 데이터 타입은 ndarray (배열)이다.

- N차원 배열을 뜻하는 numpy.ndarray 클래스를 사용한다.
- 다차원(Multi-dimension) 배열을 쉽게 생성하고 다양한 연산을 수행할 수 있다.

ndarray 내의 데이터값은 숫자 값, 문자열 값, bool 값 등이 모두 가능하다.
다만 그 연산의 특성상 같은 데이터 타입만 가능하다.

파이썬 기반의 머신러닝 알고리즘은 대부분 메모리로 데이터를 전체 로딩한 다음 이를 기반으로 알고리즘을 적용하기 때문에 대용량의 데이터를 로딩할 때는 수행속도가 느려지거나 메모리 부족으로 오류가 발생할 수 있다.

- ndarray 내 데이터값의 타입 변경을 한다.
  - int 형으로 충분한 경우인데, float 형이라면 int 형으로 바꿔서 메모리를 절약할 수 있다.
  - astype()을 이용한다.

array1.reshape(-1, 5)

- 이것은 array1과 호환될 수 있는 2차원 ndarray 로 변환하되, 고정된 5개의 컬럼에 맞는 로우를 자동으로 새롭게 생성해 변환하라는 의미이다.

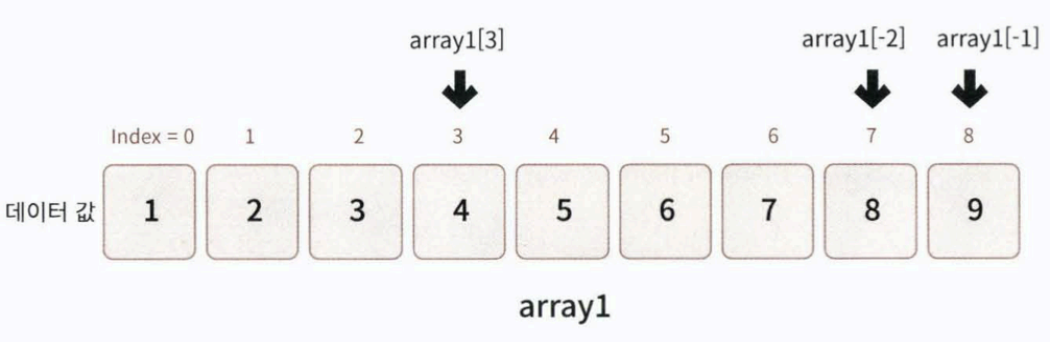
-1 인자는 reshape(-1, 1) 와 같은 형태로 자주 사용된다.
reshape(-1, 1) 은 원본 ndarray 가 어떤 형태라도 2차원이고, 여러 개의 로우를 가지되 반드시 1개의 컬럼을 가진 ndarray 로 변환됨을 보장한다.

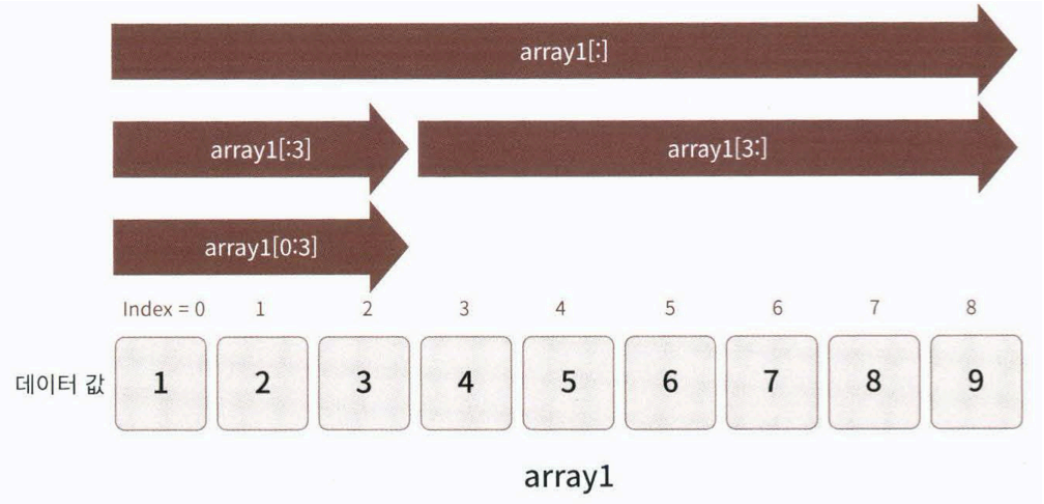
넘파이의 ndarray 의 데이터 세트 선택하기 - 인덱싱 (indexing)

1. 특정한 데이터만 추출
2. 슬라이싱(Slicing)
3. 팬시 인덱싱(Fancy Indexing)
4. 불린 인덱싱(Boolean Indexing)

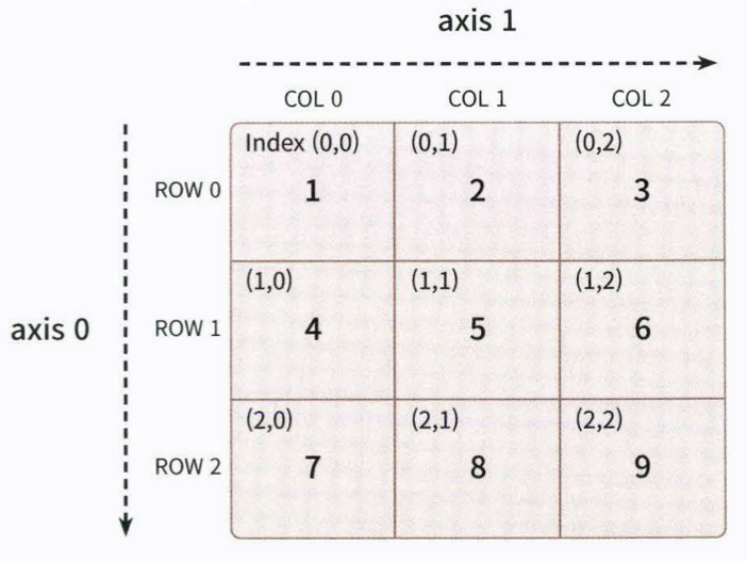
ndarray 에서 데이터를 가져오는 방식

array1 = np.arange(start=1, stop=10)

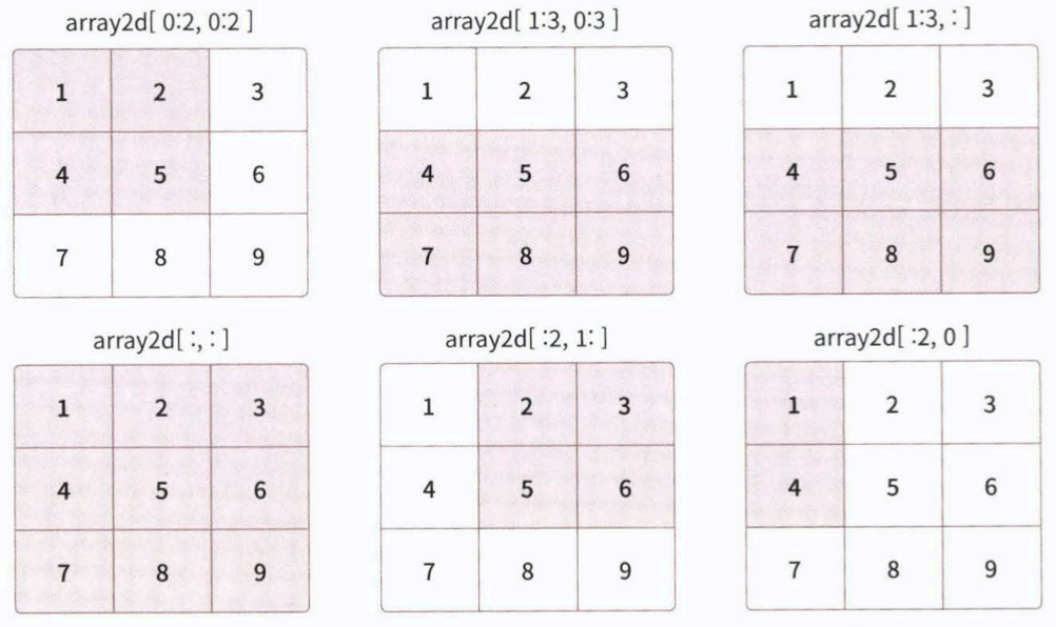




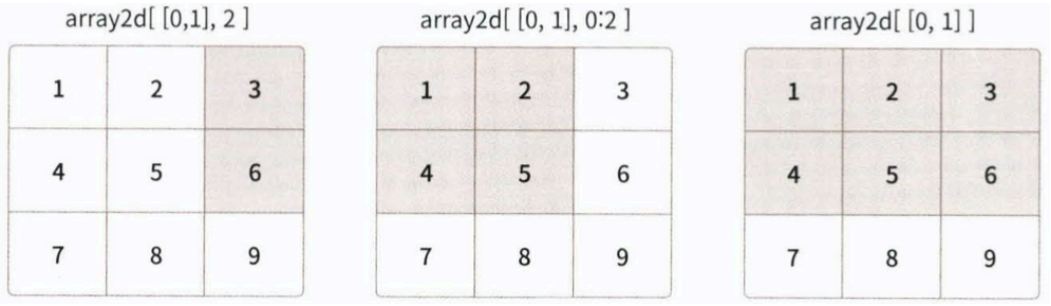
array1d = np.arange(start=1, stop=10)
array2d = array1d.reshape(3,3)



슬라이싱

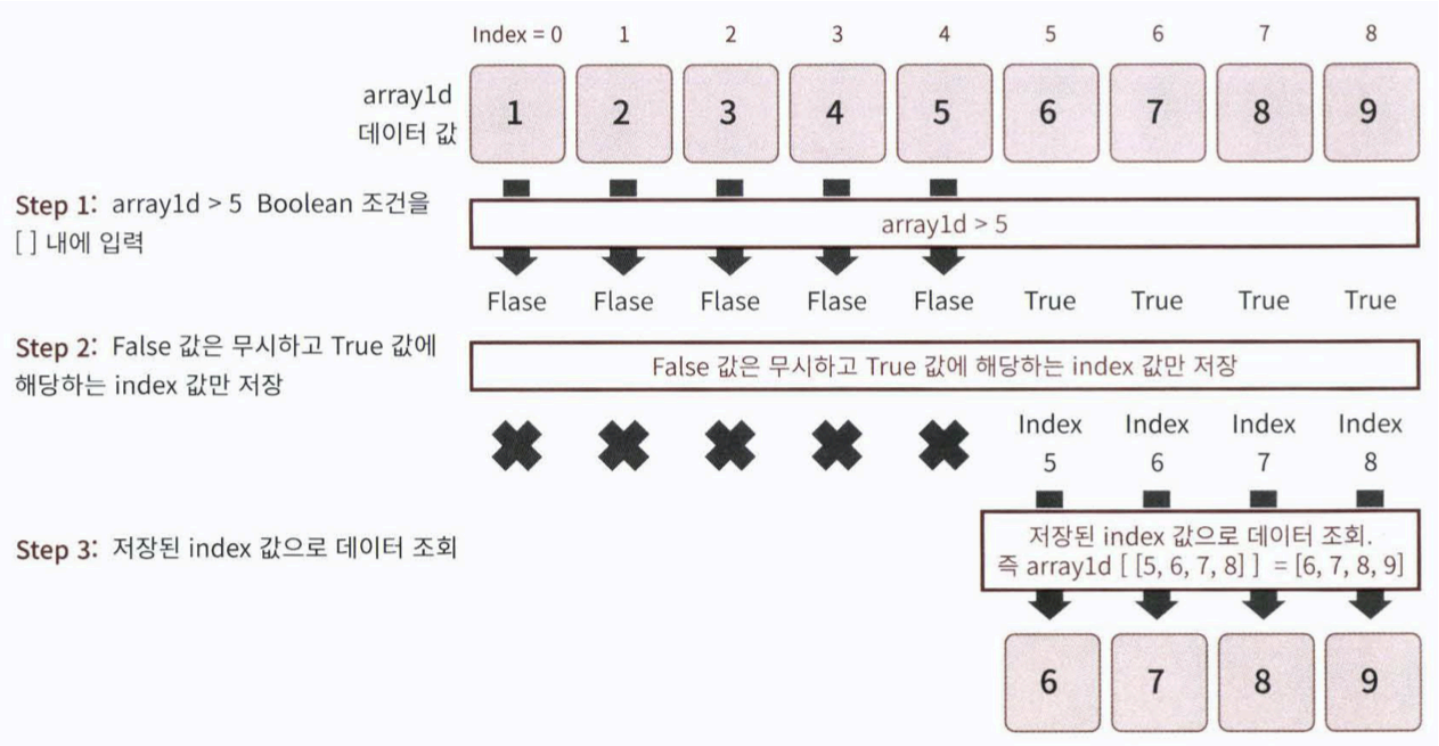


팬시 인덱싱



불린 인덱싱

arrayB1 = array1d[array1d > 5]



행렬의 정렬

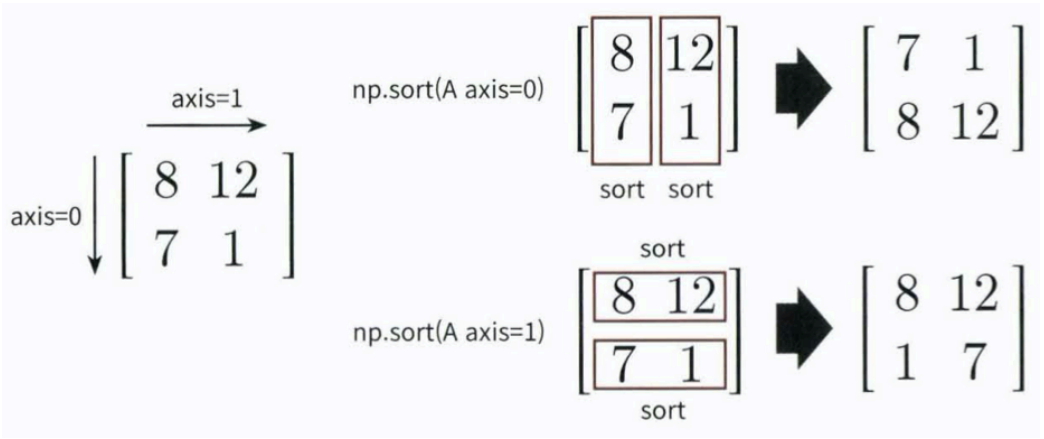
넘파이에서 행렬을 정렬

- 넘파이에서 호출 : np.sort()
    - 원 행렬은 그대로 유지한 채 원 행렬의 정렬된 행렬을 반환
- 행렬 자체에서 호출 : ndarray.sort()
    - 원 행렬 자체를 정렬한 형태로 반환하며 반환 값은 None 이다.

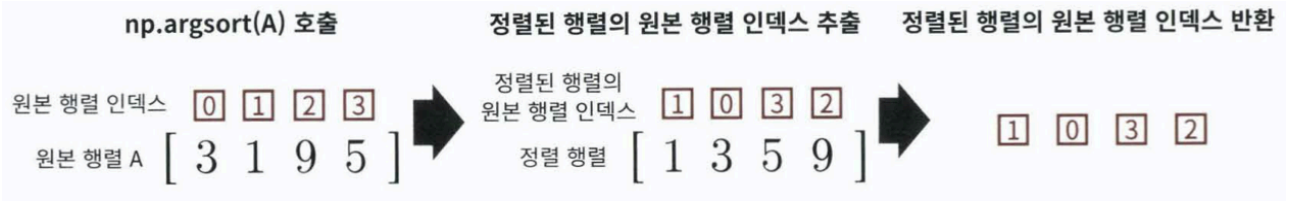
정렬된 행렬의 인덱스를 반환

- argsort()

행렬이 2차원 이상일 경우 정렬



원본 행렬이 정렬되었을 때 기존 원본 행렬의 원소에 대한 인덱스를 필요로 할 때 np.argsort() 를 이용한다. np.argsort() 는 정렬 행렬의 원본 행렬 인덱스를 ndarrary 형으로 반환한다.



넘파이의 ndarray 는 RDBMS의 TABLE 컬럼이나 판다스 DataFrame 컬럼과 같은 메타 데이터를 가질 수 없다. 따라서 실제 값과 그 값이 뜻하는 메타 데이터를 별도의 ndarray 로 각각 가져야만 한다.

- 학생별 시험 성적
    - 학생의 이름과 시험 성적을 각각 ndarray 로 가져야 한다.
    - 이 때 성적순으로 학생 이름을 출력하고자 할 때 사용할 수 있다.

```
score_array= np.array([78, 95, 84, 98, 88])
sort_indices_asc = np.argsort(score_array)
```

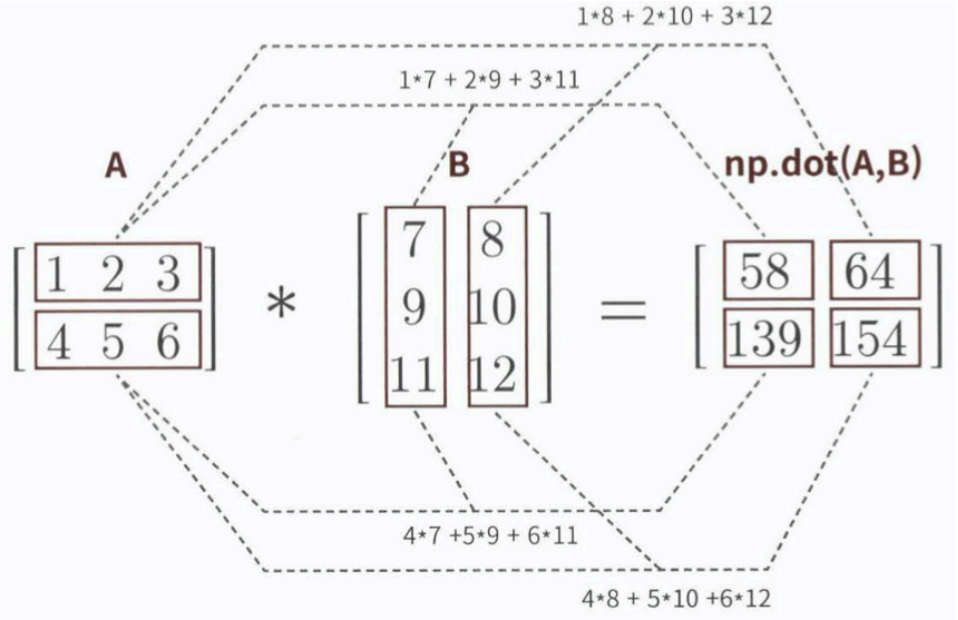
```
In [252]: print('성적 오름차순 정렬 시 score_array의 인덱스:',
sort_indices_asc)
성적 오름차순 정렬 시 score_array의 인덱스: [0 2 4 1 3]
```

선형 대수 연산

넘파이는 매우 다양한 선형대수 연산을 지원한다.

행렬 내적 **(**행렬 곱**)**

- 두 행렬 A와 B의 내적은 np.dot()을 이용해 계산이 가능하다.



전치 행렬

- 원 행렬에서 행과 열 위치를 교환한 원소로 구성한 행렬을 그 행렬의 전치행렬 이라고 한다.

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \quad A^T=\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$$

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix} \quad A^T=\begin{bmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{bmatrix}$$