



DAS 사전교육

- Python 기초 -



Numpy 기초



- ▶ 1. 넘파이 배열의 개념 및 특징
- ▶ 2. 넘파이 배열 생성
- ▶ 3. 넘파이 배열 조회
- ▶ 4. 배열의 연산
- ▶ 5. 배열 변환

○○○ 1. 넘파이 배열의 개념 및 특징 ○○○

- ▶ 배열(array): 적은 메모리로 많은 데이터를 빠르게 처리 가능
- ▶ 넘파이(Numpy, Numerical Python): 벡터 및 행렬 연산 등 고성능 수치계산을 위해 제작
- ▶ 특징
 - ▶ 고성능 과학 계산을 위한 패키지로 강력한 N차원 배열 객체
 - ▶ 정교한 브로드 캐스팅 기능
 - ▶ 파이썬 자료형 리스트와 비슷하지만, 더 빠르고 메모리를 효율적으로 관리
 - ▶ 반복문 없이 데이터 배열에 대한 처리를 지원하여 빠르고 편리
 - ▶ 데이터 과학 도구에 대한 생태계의 핵심을 이루고 있음
 - ▶ 유용한 선형 대수학, 푸리에 변환 및 난수 기능
 - ▶ 범용적 데이터 처리에 사용 가능한 다차원 컨테이너



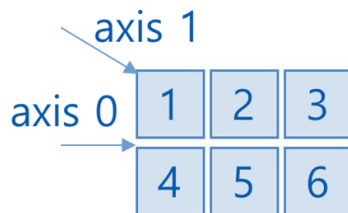
2. 넘파이 배열 생성



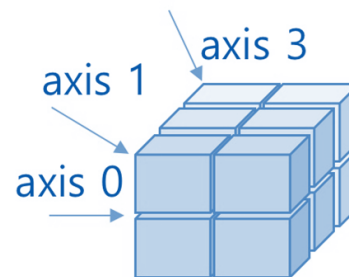
- ▶ 넘파이 배열은 다차원 배열을 지원한다.
- ▶ 여러 갈래의 입체적인 데이터 구조
- ▶ 데이터 방향을 axis로 표현
- ▶ 행방향(높이) axis = 0, 열방향(폭) axis = 1, 채널 방향 axis = 2



1D array



2D array



3D array

2. 넘파이 배열 생성

▶ 배열 생성 및 초기화 함수

함수	설명
<code>zeros()</code>	주어진 형태와 타입을 갖는 0으로 채워진 배열을 반환 <code>np.zeros(shape, dtype=float, order='C')</code> 지정된 shape의 배열을 생성하고, 모든 요소를 0으로 초기화
<code>ones()</code>	주어진 형태와 타입을 갖는 1로 채워진 배열을 반환 <code>np.ones(shape, dtype=None, order='C')</code> 지정된 shape의 배열을 생성하고, 모든 요소를 1로 초기화
<code>full()</code>	모든 요소를 지정한 값으로 초기화 <code>np.full(shape, fill_value, dtype=None, order='C')</code> 지정된 shape의 배열을 생성하고, 모든 요소를 "fill_value"로 초기화
<code>eye()</code>	단위행렬(정사각행렬) 생성 주대각선의 원소가 모두 1이고, 나머지 원소는 모두 0인 정사각행렬 <code>np.eye(N, M=None, k=0, dtype=<class 'float'>)</code> (N, N) shape의 단위 행렬(Unit Matrix)을 생성 예) <code>np.eye(n크기)</code>

2. 넘파이 배열 생성

▶ 생성한 값으로 배열 생성

함수	설명
arange()	<code>np.arange([start,] stop[, step,], dtype=None)</code> start부터 stop 미만까지 step 간격으로 데이터 생성한 후 배열을 만들 범위 내에서 간격을 기준 균등 간격의 배열 요소의 개수가 아닌 데이터의 간격을 기준으로 배열 생성
linspace()	<code>np.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)</code> start부터 stop의 범위에서 num개를 균일한 가격으로 데이터를 생성하고 배열 을 만드는 함수 요소 개수를 기준으로 균등 간격의 배열을 생성
logspace()	<code>np.logspace(start, stop, num=50, endpoint=True, base=10.0, dtype=None)</code> 로그 스케일의 linspace 함수 로그 스케일로 지정된 범위에서 num 개수만큼 균등 간격으로 데이터 생성한 후 배열 만들



2. 넘파이 배열 생성



▶ 난수 기반으로 배열 생성

함수	설명
seed	난수 발생을 위한 시드(seed) 지정
shuffle	리스트나 배열의 순서를 섞음
random	랜덤한 수의 배열 생성
randint	일정 구간의 랜덤 정수 배열 생성
rand	균등분포에서 표본 추출
randn	표준편차가 1, 평균값 0인 정규분포의 표본 추출
normal	정규분포를 고려한 랜덤한 수의 배열 생성



3. 넘파이 배열 조회



▶ 넘파이 표준 데이터 타입

데이터 타입	설명
np.int32/64	32/64비트 정수 타입
np.float32/64	32/64비트 부동 소수 타입
np.bool	불린 타입(True, False)
np.object	파이썬 객체 타입
np.complex	복소수 (128 float)



3. 넘파이 배열 조회



▶ 넘파이 배열 속성 정보

속성	설명
ndim	배열 차원 수 또는 배열의 축 수
shape	배열 각 차원의 크기를 튜플 형태로 표현
size	배열 원소의 개수
dtype	배열 내의 원소의 데이터형
itemsize	배열 내의 원소의 크기를 바이트 단위로 기술
nbytes	배열 전체 바이트



3. 넘파이 배열 조회



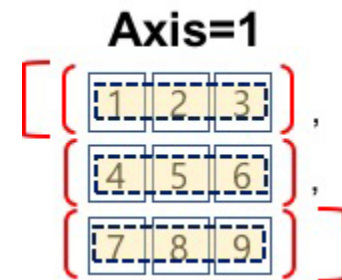
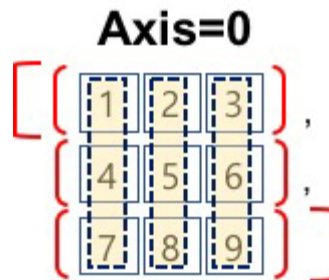
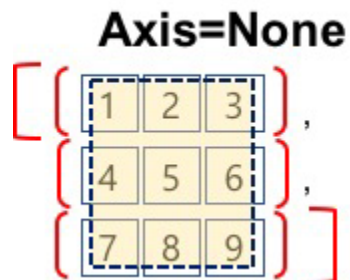
- ▶ 배열의 인덱싱과 슬라이싱
- ▶ 논리 인덱싱
 - ▶ 비교 연산자 사용
 - ▶ 논리 연산자 사용



4. 배열의 연산



- ▶ 일반 연산
 - ▶ 산술 연산: 사칙연산 연산자와 함수 활용
 - ▶ 기타 연산: 지수/제곱근/sin/cos/tan/log
 - ▶ 집계 함수

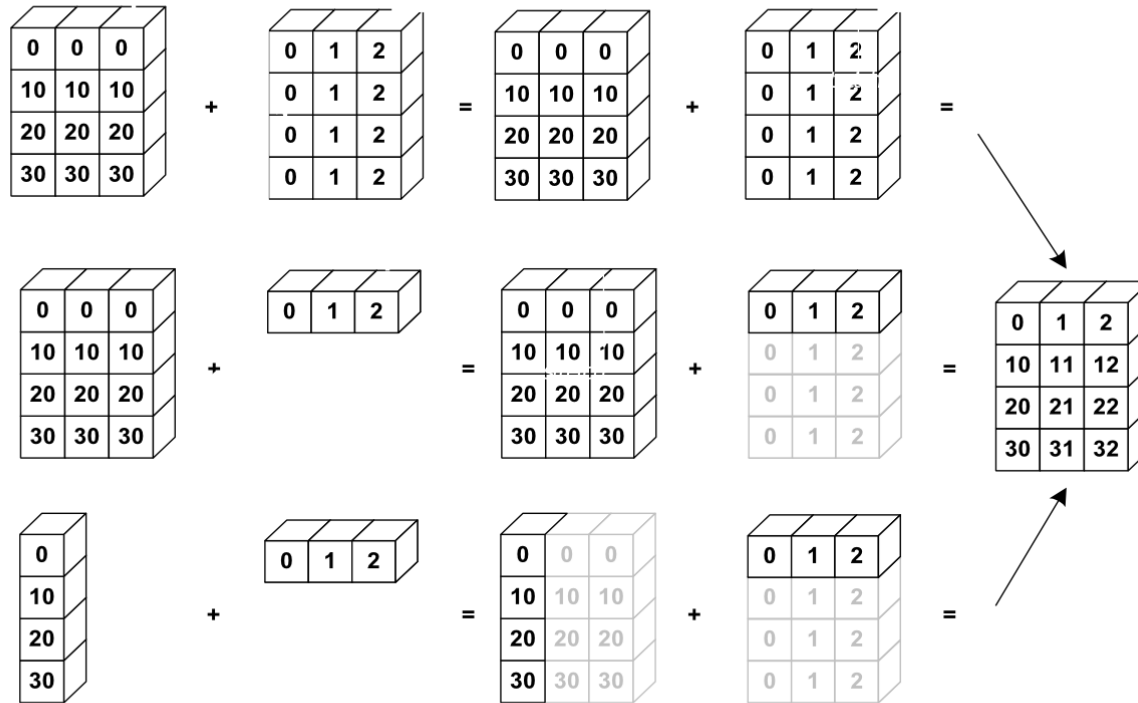




4. 배열의 연산



▶ 브로드캐스팅 연산과 벡터 연산





5. 배열 변환



▶ 전치(Transpose)

$$\begin{bmatrix} 1 & 2 \end{bmatrix}^T = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

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

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



5. 배열 변환



▶ 배열 형태 변경

[numpy.ndarray 객체]는 배열의 Shape을 변경하는 ravel 메서드와 reshape 메서드를 제공

- ▶ ravel은 배열의 shape을 1차원 배열로 만드는 메서드
- ▶ reshape은 데이터 변경없이 지정된 shape으로 변환하는 메서드

▶ [numpy.ndarray 객체].ravel()

배열을 1차원 배열로 반환하는 메서드

numpy.ndarray 배열 객체의 View를 반환

▶ [numpy.ndarray 객체].reshape()

[numpy.ndarray 객체]의 shape을 변경



5. 배열 변환



▶ 배열 요소 추가/삭제

배열의 요소를 변경, 추가, 삽입 및 삭제하는 `resize`, `append`, `insert`, `delete` 함수를 제공

- ▶ `[numpy.ndarray 객체].resize()`
- ▶ `[numpy.ndarray 객체].append()`
- ▶ `[numpy.ndarray 객체].insert()`
- ▶ `[numpy.ndarray 객체].delete()`



5. 배열 변환



▶ 배열 결합

배열과 배열을 결합하는 `np.concatenate`, `np.vstack`, `np.hstack` 함수를 제공

- ▶ `[numpy.ndarray 객체].concatenate()`
- ▶ `[numpy.ndarray 객체].vstack()`
- ▶ `[numpy.ndarray 객체].hstack()`



5. 배열 변환



▶ 배열 분리

NumPy는 배열을 수직, 수평으로 분할하는 함수를 제공

- ▶ `np.hsplit()`: 지정한 배열을 수평(행) 방향으로 분할
- ▶ `np.vsplit()`: 지정한 배열을 수직(열) 방향으로 분할