Numpy

NumPy란 무엇인가?

- NumPy는 Numerical Python의 약자
- 다차원 배열 객체와 배열을 처리하는 루틴 모음으로 구성된 라이브러리
- NumPy를 사용하면 배열에 대한 수학적 연산과 논리적 연산을 수행할 수 있음
- NumPy는 Python 패키지임

- Jim Hugunin에 의해 개발되었음. 2005 년에 Travis Oliphant는 Numarray의 기능을 Numeric 패키지에 통합하여 NumPy 패키지를 만들었음

NumPy로 할 수 있는 기능은?

- 배열에 대한 수학적 및 논리적 연산
- 푸리에 변환 및 루틴
- 선형 대수와 관련된 연산
- 선형 대수와 난수 생성을 위한 내장 함수 제공

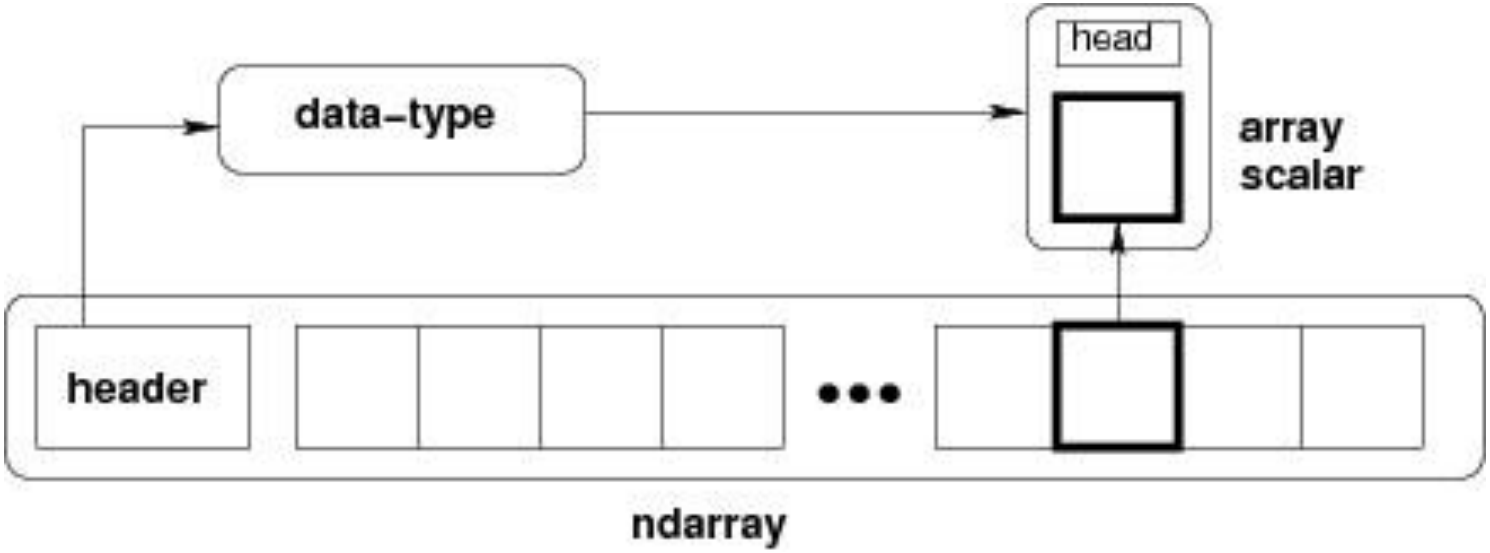
- SciPy (Scientific Python) 및 Mat-plotlib (플로팅 라이브러리) 와 함께 사용됨. 기타 등등

Ndarray Object

- ndarray 라고하는 N 차원 배열. 동일한 유형의 항목 모음. 컬렉션의 항목을 0부터 시작하는 인덱스를 사용하여 엑세스함.

- ndarray의 모든 항목은 메모리에서 동일한 크기의 블록을 사용함. ndarray의 각 요소는 데이터 유형 객체의 객체임 (dtype 이라고 함).

- ndarray 객체에서 추출된 항목은 배열 스칼라 유형 중 하나의 파이썬 객체로 표현됨.



Data Types

- bool_ : 부울(True 또는 False)을 바이트로 저장
- int_ : 기본 정수 유형 (C long과 동일, 일반적으로 int64 또는 int32)
- intc : C int (일반적으로 int32 또는 int64)와 동일
- intp : 인덱싱에 사용되는 정수 (C ssize_t와 동일, 일반적으로 int32 또는 int64)
- int8 : 바이트 (-128 ~ 127)
- int16 : 정수 (-32768 ~ 32767)
- int32 : 정수 (-2147483648 ~ 2147483647)
- int64 : 정수 (-9223372036854775808 ~ 9223372036854775807)
- uint8 : 부호없는 정수 (0에서 255)
- uint16 : 부호없는 정수 (0 ~ 65535)

Data Types

- uint32 : 부호없는 정수 (0 ~ 4294967295)
- uint64 : 부호없는 정수 (0 ~ 18446744073709551615)
- float_ : float64
- float : 16반 정밀도 부동 소수점 : 부호 비트, 5 비트 지수, 10 비트 가수
- float : 32단 정밀도 부동 소수점 : 부호 비트, 8 비트 지수, 23 비트 가수
- float : 64배정도 부동 소수점 : 부호 비트, 11 비트 지수, 52 비트 가수
- complex_ : complex128의 약자
- complex64 : 두 개의 32 비트 부동 소수점으로 표시되는 복소수 (실수 및 허수)
- complex128 : 두 개의 64 비트 부동 소수점으로 표시되는 복소수 (실수 및 허수)

Array 속성

- ndarray.shape : 튜풀로 반환함
- ndarray.ndim : 배열의 차원의 수로 반환함
- numpy.itemsize: 배열의 각 요소를 바이트 단위로 반환함
- numpy.flags : ndarray객체의 속성
  - C_CONTIGUOUS(C) : 데이터는 단일 C 스타일 연속 세그먼트임.
  - F_CONTIGUOUS(F) : 데이터는 단일, 포트란 스타일 연속 세그먼트에 임.
  - OWNDATA(O) : 배열은 사용하는 메모리를 소유하거나 다른 객체에서 가져옴.
  - WRITEABLE(W) : 데이터 영역에 쓸 수 있음. False로 설정하면 데이터가 잠겨 읽기 전용이됨.
  - ALIGNED(A): 데이터 및 모든 요소는 하드웨어에 맞게 적절하게 정렬됨.
  - UPDATEIFCOPY(U) : 이 배열은 다른 배열의 복사본임. 이 배열의 할당이 해제되면, 기본 배열은이 배열의 내용으로 됨.

Array 생성 루틴

- ndarray 객체는 다음 배열 생성 루틴이나 저수준 ndarray 생성자를 사용하여 만들 수 있음
- numpy.empty : 초기화 되지 않은 상태의 배열 생성
  - Shape – int또는 int의 tuple로 empty 배열형식
  - Dtype – 데이터 타입
  - Order – C스타일의 배열인 경우'C', Fortran스타일의 배열인 경우 'F'
- numpy.zeros : 지정된 크기의 배열을 0으로 채움
- numpy.ones : 지정된 크기의 배열을 채움(default 값 : 1)
- numpy.as array : numpy.array와 비슷하지만 매개 변수가 적다는 점만 다름.
- numpy.frombuffer : 버퍼를 1차원 배열로 정의하고 ndarray로 반환함
- numpy.fromiter : 모든 반복 가능한 객체로 부터 ndarray를 생성하고 1차원 배열로 반환함

Numerical Ranges Array

- numpy.arrange : 파이선의 range함수의 배열버전
  - numpy.arange(start, stop, step, dtype)
- numpy.linspace : arrange함수와 유사함. 함수의 크기 대신 간격 사이의 수로 정의됨
  - numpy.linspace(start, stop, num, endpoint, retstep, dtype)
- numpy.logspace : log scale에 균등하게 간격을 정의한 숫자를 포함한 ndarray 객체를 반환함

Indexing & Slicing

- ndarray 객체의 내용은 Python의 내장 컨테이너 객체와 마찬가지로 인덱싱 또는 슬라이싱을 통해 액세스하고 수정할 수 있음

- ndarray 객체의 항목은 0부터 시작하는 인덱스를 사용함. 필드 액세스, 기본 슬라이싱 및 고급 인덱싱의 세 가지 유형의 인덱싱 방법을 사용할 수 있음

- 기본 슬라이스는 Python의 n차원으로 슬라이싱하는 기본 개념을 확장한 것임. 파이썬 슬라이스 객체는 내장 된 슬라이스 함수에 시작, 정지 및 단계 매개 변수를 제공하여 구성됨. 이 슬라이스 객체는 배열의 일부를 추출하기 위해 배열로 전달됨

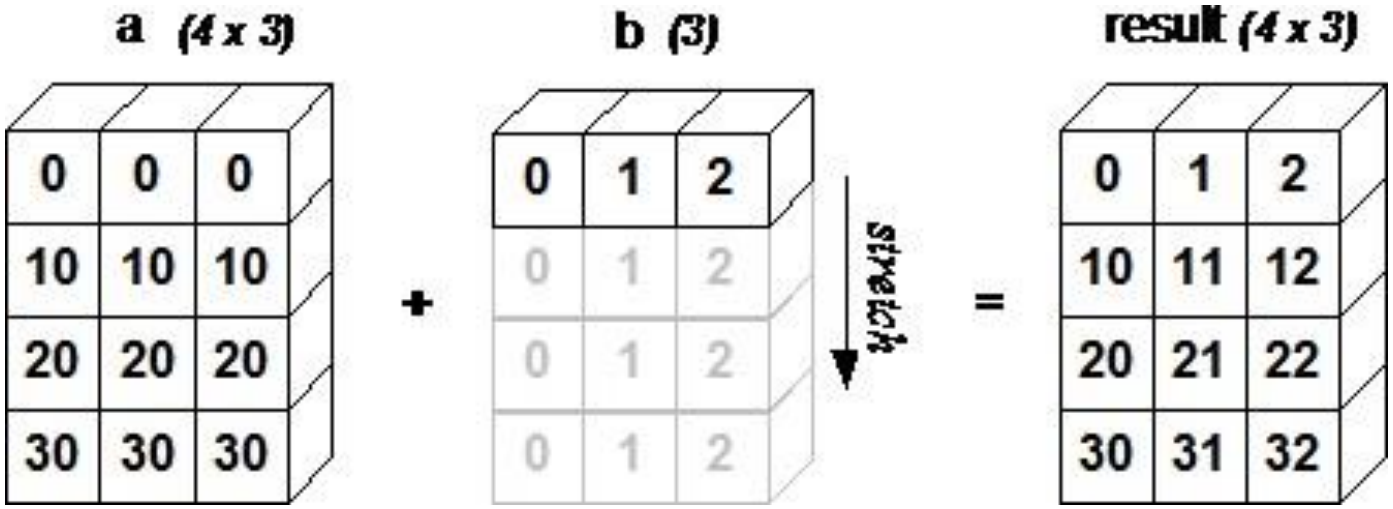
Advanced Indexing

- 비 튜플 시퀀스인 ndarray, 정수 또는 부울 데이터 유형의 ndarray 객체 또는 적어도 하나의 항목이 시퀀스 객체 인 튜플을 선택하는 것이 가능함.
- 고급 색인 생성은 항상 데이터 복사본을 반환함.
- 고급 인덱싱에는 정수와 부울 의 두 가지 유형있음

- 정수형 인덱싱
  - Ndimensional 인덱스를 기반으로 배열에서 임의의 항목을 선택하는데 도움됨. 각 정수 배열은 해당 차원에 대한 인덱스 수를 나타냄. 인덱스가 대상 ndarray의 크기만큼의 정수 배열로 구성되면 간단해짐.
- 부울 배열 인덱싱
  - 결과 객체가 비교 연산자와 같은 부울 연산의 결과가 될 때 사용됨.

Broadcasting

- 브로드 캐스팅이라는 용어는 NumPy가 산술 연산 중에 다른 모양의 배열을 처리 할 수있는 기능을 말함.
- 두 개의 배열이 정확히 같은 모양인 경우 이러한 작업이 원활하게 수행됨.



Iterating Over Array

- NumPy 패키지에는 iterator 객체 numpy.nditer 가 포함되어 있음 .
- 배열을 반복 할 수있는 효율적인 multidimensional iterator object임.
- 배열의 각 요소는 Python의 표준 Iterator 인터페이스를 함.

- Iteration Order
  - F-스타일 순서를 사용하여 동일한 요소가 저장되면 iterator는 배열에 대해보다 효율적인 반복 방법을 선택함

Array Manipulation

- Ndarray객체의 항목을 조작하기 위해서 사용하는 루틴
- Shapre 변경
  - reshap
    - 데이터를 변경하지 않고 배열에 새료운 shape를 부여함
  - Flat
    - 배열에 대한 1차원 iterator
  - flatten
    - 한 차원으로 축소된 배열 복사복을 반환함
  - ravel
    - 인접한 병합된 배열을 반환함

Array Manipulation

- Operations 변경하기(Transpose)
  - transpose
    - 배열의 크기 허용
  - ndarry.T
    - Self.transpose()와 동일함
  - rollaxis
    - 지정된 축으로 취로 움직임
  - swapaxes
    - 배열의 두축을 변경함

Array Manipulation

- 차원 변경(Changing Dimensions)
  - boradcast
    - Mimics broadcasting객체를 생성함
  - broadcast_to
    - 배열을 새로운 모앙으로 broadcastin함
  - expand_dims
    - 배열 모양을 확장함
  - squeeze
    - 배열 모양에서 1차원 항목을 제거함
- 배열합치기(Joining Arrays)
  - concatenate
    - 기존 축을 따라서 배열의 순서를 조인함
  - stack
    - 새축을 따라서 배열 순서를 조인함
  - hstack
    - 배열을 가로로 순서대로 배열함
  - vtack
    - 배열을 세로로 순서대로 배열함

Array Manipulation

- 배열 분할(Splitting Arrays)
  - split
    - 배열을 여러 개의 하위 배열로 나눔
  - Hsplit
    - 배열을 가로로 여러 개의 하위 배열로 나눔
  - Vsplit
    - 배열을 세로로 여러 개의 하위 배열로 나눔
- 요소 추가/제거(Adding/Removing Elements)
  - resize 지정된 모양으로 새배열을 반환함
  - apend 값을 배열 끝에 추가함
  - insert 지정된 인덱스의 전에 지정된 축으로 값을 삽입함
  - delete 삭제된 축을 따라서 배열이 있는 새 배열을 반환함
  - unique 배열의 고유 요소를 찾음

String Functions

- dtype의 numpy.string_ 또는 numpy.unicode_의 배열에 대해 벡터화된 문자열 연산을 수행하는데 사용됨

- add(), muliply(), center, capitalize(), title(), lower(), upper(), split(), splitlines(), strip(), join(), replace(), decode(), encode()

Mathermatical Functions

- NumPy에는 다양한 수학 연산함수가 포함됨.
- 표준 삼각함수, 산술 연산 기능, 복소수 처리 기능들을 지원함

- arcsin, arcos, arctan,
- numpy.degrees()
  - 라디안을 소수로 변환하는 함수
- nmpy.around()
  - 반올림함수
- nmpy.floor()
  - 입력 매개 변수보다 크지 않은 가장 큰 정수를 반환함
- numpy.ceil()
  - 입력 값의 한도를 반환함

Arithmetic Operations

- add(), subtract(), multiply(), divide()와 같은 산술 연산을 수행하기 위한 함수 제공
- numpy.reciprocal()
  - 인자의 역수를 원소 단위로 반환함
- numpy.power()
  - 첫 번째 입력 배열의 요소를 기본 요소루 처리하고 두번째 입력 배열의 해당 요소를 거듭 제곱으로 반환함
- Numpy.mod()
  - 입력 배열의 해당 요소를 나눈 나머지를 반환함.
  - Numpy.remainder()함수와 동일한 결과를 생성함.

Statistical Functions

- 배열의 지정된 요소에서 최소, 최대, 백분위 표준 편차 및 분산등의 함수를 제공함.
- numpy.amin(), numpy.amax()
  - 지정된 축을 따라서 지정된 배열의 요소에서 최소값과 최대 값을 반환함
- numpy.ptp()
  - 축을 따라 값의 범위(최대-최소)를 반환함.
- numpy.percentile()
  - 백분위(또는 센티미터)는 주어진 관측 백분율이 그 이하가 되는 값을 나타내는 통계에 사용되는 측정임
- numpy.median()
  - 중앙값은 데이터 샘플의 상위 절반을 하위 절반에서 분리하는 값으로 정의됨
- Numpy.mean()
  - 산술 평균은 축의 요소 합계를 항목의 수로 나눈값임
- Numpy.average()
  - 가중 평균은 각 구성 요소의 중요도를 반영하는 요소로 각 구성요소를 곱한 결과의 평균임