**NumPy**

**Travis Oliphant**

For python numerical values, Easily control the large amount of data using array. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

* Help:

i)Speed

ii) Less memory

iii) easy to operate

iv) using function for calculation

* Uses:

i)Data science

ii) ML (AI)

iii) Stock market analysis

iv) Medical rech.

v) Image processing

Use for make matrix:   
 import numpy as np

Array = np.array([1,2,3..])

Print(array)

Output: [1 2 3 . . . . . ]

**Using default function:**

* Fill with Zero

import numpy as np

Array = np.zeros((shape)) # shape: input row and column

Print(array)

Output: [0 0 0 0 . . . . . . . . . ]

* Fill with one(1)

import numpy as np

Array = np.ones((shape)) # shape: input row and column

Print(array)

Output: [1 1 1 1 . . . . . . . . . ]

* Fill with specific value

import numpy as np

Array = np.full((shape), value) # shape: input row and column

Print(array)

Output: [Automatically fill in the blank cell, set your specific value]

**Use arange function to create Sequence of number:**

Function: arrange(start, stop, step)

Import numpy as np

Array = np.arange(start, stop, step)

Print(Array)

**Array Properties and operations**

check

1. .shape (when working with multidimensional data shape used to see the structure- row & column)
2. .size (show total number of elements in an array)
3. .ndim (show the array dimension- 1D , 2D, 3D)
4. .type (show the variable type- int, float, char, etc)

Change data type:

Using .astype(new type) function to change the data type

Mathematical operation:

Print (array + or – or \* or \*\* or / or // or % number)

Aggregate function:

|  |  |
| --- | --- |
| Function | What it do |
| Np.sum(array) | Add all value |
| Np.mean(array) | Provide us array |
| Np.min(array) | Minimum number |
| Np.max(array) | Maximum number |
| No.std(array) | Standard deviation |
| Np.var(array) | Variance calculate |

**Indexing and Slicing**

Indexing- when we need to take one value in our data

Slicing- when we need to take a list of data

Fancy indexing- select multiple elements in one time using list

Boolean masking- filter elements using certain condition

Indexing

For 1D: print(arr(index))

For 2D: print(arr(row, column))

Slicing

Print(array[start:end:step]) # take only index number

Fancy indexing

Help to select non sequential elements.

Print(array[[1,2,3,4]] #take index value

*Boolean masking*

*Provide us that values which are satisfied the condition.*

*Print(array[array “use condition”])*

**Re-shaping and manipulation data**

Re-shaping

Change the dimension of array but don’t modify the data and total quantity of data. Need to match the dimension.

Code: array.reshape(row, col)

Flattering

When we try to create a 1D array using multidimensional array. This two things are use to make this array:

1. .ravel() – returns the new array and modify the primary array.
2. .flatten() – return new array (copy) but don’t modify the primary array.

Array Modification

Insert

We don’t modify a array directily in python. But in numpy we are use a function ( .insert()) use to modify array. *Make a new array.*

Code: np.insert(array, index, value, axis=none)

For 2D array

Axis= 0; row wise insert data

Axis= 1; column wise insert data

Code: np.insert(array, index, value([1,2 . .]), axis= )

Append

Add values in last.

Code: np.append(array, [value])

Concatenate

When we need to add two or more array that time use concatenate function. Provide us a 1D array.

Code: np.concat(array\_1, array\_2, .. .. .. .. , axis = ?)

Axis = 0; add arrays in row wise

Axis = 1; add arrays in column wise

Stacking

Use different array to make a new dimensional array.

1. .vstack() – add array in row wise
2. .hstack() – add array in column wise

Code: np.v/hstack(array, array, .. .. .. )

Splitting

Divided primary array in some sub array, based on the condition.

Code: np.split(array, koto bar)

There are three types of splitting:

1. Np.split() – divided array in equal parts
2. Np.vspllit() – divided array in vertically (work on two or more dimensional array)
3. Np.hsplit()- divided array in horizontally

**Broadcasting & vectorization**

Broadcasting is a numpy array, where we are perform many operation using different shape of array. [সোজা কথায় ছোট array কে সে বড় বানায় দেয়]

There are 3 portion to handel different shape:

1. Matching dimension
2. Expending single element of array
3. Incompatible shape (if array shape don’t match, if provide us error)

Vectorization

Perform a operation above all the array don’t using loop.

**Handling missing value and special case**

When a calculation fails because some data is missing in the table, Numpy provide a Boolean array to identify the missing values. Where TRUE character is our missing value. Numpy have 3 function to handle missing value. These are:

1. Np.isnan(array); detect missing value, not number.
2. Np.isinf(array); detect infinite value.
3. Np.nan\_to\_num(array, value); replace the missing values.

Or, if we don’t provide the value this time it will automatically replace the missing value to zero (0).

Change infinite number

Code: np.nan\_to\_num(array, posinf=value, neginf=value)

posinf=value; replace the positive infinite number

neginf=value; replace the negative infinite number