

Numpy - general purpose array-processing package

Numerical Python - powerful N-dimensional array object

Install - pip install numpy

Import - import numpy as np.

Arrays - table of elements (numbers), all of the same datatype, indexed.

↳ ndarray

Numpy Array - np.array() - Create N-dimensional array

- elements inside = same data type
- element wise mathematical operation
- 1D Array (list, tuple)
- 2D Array ----

Python list - Ordered collection of items/elements

- put any datatype of elements
- element wise operation = not possible
- (1D)

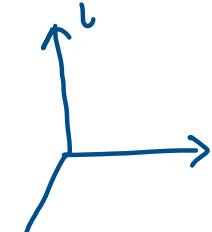
Memory management -

May not store elements in contiguous memory

Cross product b/w two vectors

$$\vec{a} = a_1 \hat{i} + a_2 \hat{j} + a_3 \hat{k}, \quad \vec{b} = b_1 \hat{i} + b_2 \hat{j} + b_3 \hat{k}$$

$$\vec{c} = \vec{a} \times \vec{b}$$

$$\vec{c} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}$$


$$\begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \quad k$$

$$\begin{aligned}\vec{C} &= i \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} - j \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix} + k \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \\ &= i(a_2b_3 - a_3b_2) - j(a_1b_3 - a_3b_1) + k(a_1b_2 - a_2b_1)\end{aligned}$$