

# ***DOT PRODUCT IN NUMPY***

*Specifically, there are two rules to consider in dot product in python:*

- 1. If both vector “A” and vector “B” are 1-dimensional arrays, it is an inner product of vector (without complex conjugation).*
- 2. If both vector “A” and vector “B” are 2-dimensional arrays, it is a matrix multiplication.*

*You can only multiply two matrices if the number of columns of the first matrix is equal to the number of rows of the second matrix.*

*If the shape of one matrix is (m\*n) and the shape of the other one should be (n\*t).*

## ***DOT PRODUCT OF TWO MATRICES.***

*We can then go ahead and multiply 2 matrices, here is an example of a 1\*1 matrix “A” multiply with 1\*1 matrix “B”.*

*That is:  $A * B = C$*

$$A = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = 7$$

$$(-1*0) + (2*2) + (3*1)$$

$$0 + 4 + 3 = 7$$

*Firstly, import the Numpy package in your workspace and then create a 2D matrix as discussed in the example above or any other matrix structure you want to try it out with. Then create a vector ensuring the number of rows being equal to the number of columns in the matrix.*

*You can then use the “dot()” method in the matrix and vector. It will return the result vector for you as shown.*

*You can check out the entire notebook and play around with the code using.*

*Let’s code them using Numpy:*

***import numpy as np***

*#we are importing the library*

***a = np. array ([-1, 2, 3])***

***b = np. array([-0, 2, 1])***

*#we are using vector “a” and vector “b”*

***print(a)***

***output:*** `[ [-1 2 3] ]`

***print(b)***

***output:*** `[ [0 2 1] ]`

***np.inner(a, b)***

*#we are finding the inner product*

***Output:*** `array ( [ [7] ] )`