

Operataing element by element	
Operate n to all components	<code>vector + n</code> <code>matrix * n</code> <code>vector * n</code>
Operation using another vector	<code>vector + vector</code> <code>matrix * matrix</code>
Matrix : Transpose	<code>matrix.T</code>
Matrix multiplication	<code>np.dot(matrix, matrix.T)</code>

Matrix & Vectors Operations	
Sum of all components	<code>np.sum(vector)</code> <code>vector.sum()</code>
Maximum	<code>np.max(vector)</code> <code>vector.max()</code>
Minimum	<code>np.min(vector)</code> <code>vector.min()</code>
Product	<code>np.prod(vector)</code> <code>vector.prod()</code>
Operation over column	<code>matrix.sum(axis=0)</code>
Operation over row	<code>matrix.sum(axis=1)</code>
Absolute	<code>np.abs(matrix)</code>

Indexing	
Especific element	<code>matrix[row, column]</code>
Range of row / columns	<code>matrix[rI : rE, cI : cE]</code>
Submatrix (same matrix)	<code>submatrix = matrix[1:3, 1:3]</code>
Submatrix (Different matrix)	<code>submatrix = submatrix.copy()</code>
Boolean Indexing	
Finding negative values	<code>negativus = example < 0</code>
Selecting negative values	<code>example[negativus]</code>
Changing values directly	<code>example[example==0] = 0</code>

Vectors & Matrix	
------------------	--

Broadcasting	
Normalize a matrix by columns	<code>np.random.uniform(size=(3,4))</code> <code>sumes = matrix.sum(axis=0)</code> <code>norm = matrix/sumes</code>
Normalize a matrix by rows**	<code>norm=(matrix.T/sumes).T</code> <code>norm=matrix/sumes.reshape((3,1))</code>
Not-a-Number (Nan)	
Initialize as NaN	<code>`a = np.nan</code>
Test if is NaN	<code>np.isnan(a)</code>
Test also with vectors	<code>np.isnan(vector)</code>

Extra	
Random uniform matrix	<code>np.random.uniform(size=(3,4))</code>
Another type of random matrix	<code>np.random.uniform(-1,1,size=(3,4))</code>
Copying a matrix	<code>matrix = matrix.copy()</code>
Shape of a matrix	<code>matrix.shape</code>
Data types	<code>dtype = np.float32</code> <code>dtype = np.float64</code> <code>dtype = np.int32</code> <code>dtype = uint32</code> <code>dtype = np.int8</code> <code>dtype = np.int8</code> <code>dtype = np.uint8</code>
Importing	<code>import numpy as np</code>

Create vector from list	<code>x = np.array(x)</code>
Create vector of n zeros	<code>zeros = np.zeros(n)</code>
Create vector of n ones	<code>ones = np.ones(n)</code>
Create vector of type float32	<code>np.array([1, 0.3, 2], dtype=np.float32)</code>
Change type of a vector	<code>vector = vector.astype(np.uint32)</code>
Create matrix of ones	<code>m1 = np.ones((5, 4))</code>
Create matrix from other matrix shape	<code>m2 = np.zeros_like(m1)</code>
Reshape a matrix (constant size)	<code>mat = mat.reshape((6, 3))</code>



By **hector**
cheatography.com/hector/

Not published yet.
 Last updated 17th September, 2019.
 Page 1 of 2.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>