

Operataing element by element

Operate n to all components

```
vector + n
matrix * n
vector * n
```

Operation using another vector

```
vector + vector
matrix * matrix
```

Matrix : Transpose

```
matrix.T
```

Matrix multiplication

```
np.dot (ma tri x, m atr ix.T)
```

Vectors & Matrix

Create vector from list

```
x = np.arr ay(x)
```

Create vector of n zeros

```
zeros = np.zer os(n)
```

Create vector of n ones

```
ones = np.ones(n)
```

Create vector of type float32

```
np.arr ay( [1, -0.3 ,2], dtype=np.f l o at32)
```

Change type of a vector

```
vector = vector.as typ e(n p p r int32)
```

Create matrix of ones

```
m1 = np.one s(( 5,4))
```

Create matrix from other matrix shape

```
m2 = np.zer os_ lik e(m1)
```

Reshape a matrix (constant size)

```
mat=ma t.r esh ape ((6,3))
```

Matrix & Vectors Operations

Sum of all components

```
np.sum (ve ctor)
vector.sum()
```

Maximum

```
np.max (ve ctor)
vector.max()
```

Minimum

```
np.min (ve ctor)
vector.min()
```

Product

```
np.pro d(v ector)
vector.prod()
```

Operation over column

```
matrix.sum(a xis=0)
```

Operation over row

```
matrix.sum(a xis=1)
```

Absolute

```
np.sum (ma triu)
```

Indexing

Especific element

```
matrix[row,column ]
```

Range of row / columns

```
matrix[ rI : rE, cI :cE]
```

Submatrix (same matrix)

```
submatrix = matrix[1:3, 1:3]
```

Submatrix (Different matrix)

```
submatrix = submatrix.copy()
```

Boolean Indexing

Finding negative values

```
negatius = example < 0
```

Selecting negative values

```
example[negatius]
```

Changing values directly

```
example[example==0] = 0
```

Broadcasting

Normalize a matrix by columns

```
np.random.uniform(size=(3,4))
sumes = matrix.sum(axis=0)
norm = matrix/sumes
```

Normalize a matrix by rows**

```
norm=(matrix.T/sumes).T
norm=matrix/sumes.reshape((3,4))
```

Not-a-Number (NaN)

Initialize as NaN

```
`a = np.nan
```

Test if is NaN

```
np.isn an(a)
```

Test also with vectors

```
np.isn an( vector)
```

Extra

Random uniform matrix

```
np.random.uniform(size=(3,4))
```

Another type of random matrix

```
np.random.uniform(-1,1,size=(3,4))
```

Copying a matrix

```
matrix = matrix.copy()
```

Shape of a matrix

```
matrix.shape
```

Data types

```
dtype = np.float32
dtype = np.float64
dtype = np.int32
dtype = np.int64
dtype = np.uint32
dtype = np.uint64
dtype = np.int8
dtype = np.uint8
```

Importing

```
import numpy as np
```