NumPy Tutorial

https://docs.scipy.org/doc/numpy-dev/user/quickstart.html

```
#importing with alias
import numpy as np
#create an array by passing a list
a = np.array([1,2,3,4,5])
#array of consecutive numbers
a = np.arange(12)
```

```
#add arrays
a = np.arange(12)
b = np.arange(12)
a + b
#multiply by scalar
a * 3
# * does not do dot product in numpy
```

```
#get the type of an array
np.type(a)
#get types of elements of array
a.dtype
#get shape
a.shape, np.shape(a)
#get size
a.size, np.size(a)
```

```
#fill all values in an array
a.fill(0), a[:] = 1, np.zeros((3,2)),
np.ones((3,2)), np.empty((4,5))
#beware of type conversion: a float will be
#coerced into an int32 if that’s the dtype
#a random matrix
r np.random.rand(3, 3)
```

```
#fill all values in an array (contd)
#200 numbers from 0 to 50
x = np.linspace(0, 50, 200)

#use a math function to transform
np.sin(x)
```

Slicing

```
#same as for lists
#last 2 elements
a[-2:]

#every other element
a[::2]
```

Dot and element-wise products

```
A = np.array( [[1,1],
               [0,1]] )
B = np.array( [[2,0],
               [3,4]] )
A*B                          # element-wise product
result: array([[2, 0],
               [0, 4]])
A.dot(B)                     # matrix product
result: array([[5, 4],
               [3, 4]])
```

```
#upcasting to more general type
a = np.ones(3, dtype=np.int32) #dtype int32
b = np.linspace(0,np.pi,3)
b.dtype.name #'float64'

c = a+b
c.dtype.name #'float64'
```

For a list of numpy datatypes, see
http://docs.scipy.org/doc/numpy/user/basics.types.html

```
#get min, max, sum
a.sum()
a.min()
a.max()
np.sum(a)
np.min(a)
np.max(a)
```

```
#summing, getting max or mins from columns
or rows in a matrix
b.sum(axis=0)  # sum of each column
b.min(axis=1)  # min of each row

b.cumsum(axis=1) # cumulative sum along each
row
```

Universal Fuctions

```
#apply element-wise on an array --- see
#https://docs.scipy.org/doc/numpy-dev/user/#
quickstart.html
```

all, any, apply_along_axis, argmax, argmin, argsort, average, bincount, ceil, clip,conj, corrcoef, cov, cross, cumprod, cumsum, diff, dot, floor, inner, *inv*, lexsort,max, maximum, mean, median, min, minimum, nonzero, outer, prod, re,round, sort, std, sum, trace, transpose, var, vdot, vectorize, where

Multidimensional arrays

```
#getting rows and columns
#the value in the third row and 6th column
b[2,5]
#the second row
b[2, :]
#the second column
b[:, 2]
#iterations are over the first axis
```

Change array shapes

```
#flatten
a.ravel()
#set shape
a.shape = (6,2)
#transpose
a.T
#resize: missing indices filled with zeroes
a.resize(1,2)
```

Change array shapes

```
#You can’t resize an array that is
referenced
```

Stacking arrays

```
#stack vertically (as rows)
np.vstack((a,b))
#stack horizontally (as columns)
np.hstack((a,b))
#to stack a 1d column
np.column_stack((a,b))
```

Splitting arrays

```
#horizontally into 3
np.hsplit(a,3)
#horizontally after 3rd and 4th col
np.hsplit(a, (3,4))
#np.vsplit splits along vertical axis
#np.array_split --- can specify axis
```

Using Arrays as indices

```
a = np.arange(12)**2                                    #
the first 12 square numbers
i = np.array( [ 1,1,3,8,5 ] )                           #
an array of indices
a[i]                                                    #
the elements of a at the positions i
j = np.array( [ [ 3, 4], [ 9, 7 ] ] )                   #
a bidimensional array of indices
a[j]      # the same shape as j
```

Indexing with boolean arrays

```
a = np.arange(12).reshape(3,4)
b = a > 4
b                                                    #
b is a boolean with a's shape
array([[False, False, False, False],
       [False,  True,  True,  True],
       [ True,  True,  True,  True]],
dtype=bool)
a[b]                                                 #
1d array with the selected elements
array([ 5,  6,  7,  8,  9, 10, 11])
```

Using boolean indices in assignments

```
#change the zeros to -1
a_list = [[0,2,3,0], [1,3,0,0], [0, 2, 0, 4]]
a = np.array(a_list)
b = a==0
a[b] = -1
```

References

https://docs.scipy.org/doc/numpy-dev/user/quickstart.html