numpy_array

January 2, 2019

In [5]:
```
# basic
import numpy as np
list_of_ints = [1, 2 ,3]
array = np.array(list_of_ints)
print(array)
print(type(array))
print(array.dtype)

array2 = np.array(list_of_ints, dtype ='int8')
print(array2)
print(type(array2))
print(array2.dtype)

array_float = array.astype('float32')
print(array_float)
print(type(array_float))
print(array_float.dtype)
```

```
[1 2 3]
<class 'numpy.ndarray'>
int32
[1 2 3]
<class 'numpy.ndarray'>
int8
[1. 2. 3.]
<class 'numpy.ndarray'>
float32
```

In [6]:
```
# 2D array
list_of_lists = [[1,2,3], [4,5,6], [7,8,9]]
array_2d = np.array(list_of_lists)
print(array_2d)
print(type(array_2d))
print(array_2d.dtype)
```

```
[[1 2 3]
 [4 5 6]
```

```
 [7 8 9]]
<class 'numpy.ndarray'>
int32
```

```
In [9]: # 3D array
        list_of_lists = [[[1,2], [4,5], [6,7]],[[8,9], [10,11], [12,13]]]
        array_3d = np.array(list_of_lists)
        print(array_3d)
        print(type(array_3d))
        print(array_3d.dtype)
```

```
[[[ 1  2]
  [ 4  5]
  [ 6  7]]

 [[ 8  9]
  [10 11]
  [12 13]]]
<class 'numpy.ndarray'>
int32
```

```
In [12]: # reschape (new array is pointer to old one)
         list = [1,2,3,4,5,6,7,8,9,10]
         array = np.array(list)
         reschaped_pointer = array.reshape((5,2))
         reschaped_pointer[0,0] = -1
         print(array)
         print(reschaped_pointer)
```

```
[-1  2  3  4  5  6  7  8  9 10]
[[-1  2]
 [ 3  4]
 [ 5  6]
 [ 7  8]
 [ 9 10]]
```

```
In [13]: # resize (crete new array in place)
         list = [1,2,3,4,5,6,7,8,9,10]
         array = np.array(list)
         array.resize((5,2))
         print(array)
```

```
[[ 1  2]
 [ 3  4]
 [ 5  6]
 [ 7  8]
```

```
 [ 9 10]]
```

```
In [21]: # generate data
         ordinal = np.arange(9)
         print(ordinal)
         zeros = np.zeros((3,3))
         print(zeros)
         ones = np.ones((6,6))
         print(ones)
         eye = np.eye(3)
         print(eye)
         # random data
         random = np.random.randint(low=1, high=10, size=(3,3))
         print(random)
         # normal distribution (Gaus)
         normal = np.random.normal(size=(3,3))
         print(normal)
```

```
[0 1 2 3 4 5 6 7 8]
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
[[1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1.]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[6 7 7]
 [7 4 2]
 [5 2 5]]
[[-0.08685524 -0.32995069  1.33801461]
 [ 0.99482601 -0.61976423 -1.00147939]
 [ 0.50925657  0.91864279  0.15143081]]
```

```
In [ ]: #data from file
        np.loadtxt() # file name and so on
```