In [1]:
```
import sys
import numpy as np

print("Python {}".format(sys.version))
print("NumPy {}".format(np.__version__))
```

```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
NumPy 1.18.1
```

In [2]:
```
# defining a scalar
x = 6
x
```

Out[2]:
```
6
```

In [3]:
```
# defining a vector
x = np.array((1,2,3))
x
```

Out[3]:
```
array([1, 2, 3])
```

In [5]:
```
print("Vector Dimensions: {}".format(x.shape))
print("Vector Size: {}".format(x.size))
```

```
Vector Dimensions: (3,)
Vector Size: 3
```

In [6]:
```
# defining a matrix
x = np.matrix([[1,2,3],[4,5,6],[7,8,9]])
x
```

Out[6]:
```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

In [7]:
```
print("Matrix Dimensions: {}".format(x.shape))
print("Matrix Size: {}".format(x.size))
```

```
Matrix Dimensions: (3, 3)
Matrix Size: 9
```

In [8]:
```
# define a matrix of given dimension
x = np.ones([3,3])
x
```

Out[8]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

In [9]:
```
print("Matrix Dimensions: {}".format(x.shape))
print("Matrix Size: {}".format(x.size))
```

```
Vector Dimensions: (3, 3)
Vector Size: 9
```

In [10]:
```
# example of a three dimensional tensor
x = np.ones([3,3,3])
x
```

Out[10]:
```
array([[[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.]]])
```

In [11]:
```
print("Matrix Dimensions: {}".format(x.shape))
print("Matrix Size: {}".format(x.size))
```

```
Matrix Dimensions: (3, 3, 3)
Matrix Size: 27
```

In [17]:
```
# indexing
A = np.ones([5,5], dtype = np.int)
A
```

Out[17]:
```
array([[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]])
```

In [18]:
```
# indexing starts at 0
A[0,1] = 2
A
```

Out[18]:
```
array([[1, 2, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

In [19]:
```
A[:,0] = 3
A
```

Out[19]:
```
array([[3, 2, 1, 1, 1],
       [3, 1, 1, 1, 1],
       [3, 1, 1, 1, 1],
       [3, 1, 1, 1, 1],
       [3, 1, 1, 1, 1]])
```

In [20]:
```
A[:,:] = 5
A
```

Out[20]:
```
array([[5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5],
       [5, 5, 5, 5, 5]])
```

In [25]:
```
# for higher dimensions, simply add an index
A = np.ones([5,5,5], dtype = np.int)

# assign first row a new value
A[:,0,0] = 6
A
```

Out[25]:
```
array([[[6, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1]],

       [[6, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1]],

       [[6, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1]],

       [[6, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1]],

       [[6, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1]]])
```

In [27]:
```
# matrix operations
A = np.matrix([[1,2],[3,4]])
B = np.ones([2,2], dtype = np.int)
```

In [28]:
```
A
```

Out[28]:
```
matrix([[1, 2],
        [3, 4]])
```

In [29]:
```
B
```

Out[29]:
```
array([[1, 1],
       [1, 1]])
```

In [30]:
```
# Element wise addition
C= A + B
C
```

Out[30]:
```
matrix([[2, 3],
        [4, 5]])
```

In [31]:
```
# Element wise subtraction
C = A - B
C
```

Out[31]:
```
matrix([[0, 1],
        [2, 3]])
```

In [32]:
```
# Element wise multiplication
C = A * B
C
```

Out[32]:
```
matrix([[3, 3],
        [7, 7]])
```

In [33]:
```
# matrix transpose
A = np.array(range(9))
A = A.reshape(3,3)
A
```

Out[33]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [34]:
```
B = A.T
B
```

Out[34]:
```
array([[0, 3, 6],
       [1, 4, 7],
       [2, 5, 8]])
```

In [35]:
```
C = B.T
C
```

Out[35]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [36]:
```
A = np.array(range(10))
A = A.reshape(2,5)
A
```

Out[36]:
```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

In [37]:
```
B = A.T
B
```

Out[37]:
```
array([[0, 5],
       [1, 6],
       [2, 7],
       [3, 8],
       [4, 9]])
```

In [38]:
```
print(B.shape)
```

```
(5, 2)
```

In [39]:
```
print(A.shape)
```

```
(2, 5)
```

In [41]:
```
# tensors
A = np.ones((3,3,3,3,3,3,3,3,3,3))
print(A.shape)
print(len(A.shape))
print(A.size)
```

```
(3, 3, 3, 3, 3, 3, 3, 3, 3, 3)
10
59049
```

In [ ]:
```

```