

NumPy – Exercise – Assignment - 1

```
import numpy as np
```

Create an array of 10 zeros

```
np.zeros(10)  
  
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

Create an array of 10 ones

```
np.ones(10)  
  
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Create an array of 10 fives

```
np.ones(10) * 5  
  
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

Create an array of the integers from 10 to 50

```
np.arange(10, 51)  
  
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,  
       27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,  
       44, 45, 46, 47, 48, 49, 50])
```

Create an array of all the even integers from 10 to 50

```
print(np.arange(10, 51, 2))  
  
[10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50]  
  
x = np.arange(10, 51)  
y = (x%2 == 0)  
z = x[y]  
print(z)  
  
[10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50]
```

Create a 3x3 matrix with values ranging from 0 to 8

```
np.arange(0, 9).reshape((3, 3))
```

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

Create a 3x3 identity matrix

```
np.eye(3)

array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

Generate a random number between 0 and 1

```
np.random.randint(0,1)

0
```

Create an array of 20 linearly spaced points between 0 and 1

```
np.linspace(0,1,20)

array([0.          , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
       0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
       0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
       0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.          ])
```

Numpy Indexing and Selection

Now you will be given a few matrices, and be asked to replicate the resulting matrix outputs:

```
mat = np.arange(1,26).reshape(5,5)
mat

array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

```
mat[2:,1:]  
  
array([[12, 13, 14, 15],  
       [17, 18, 19, 20],  
       [22, 23, 24, 25]])
```

```
mat[3,4]
```

```
20
```

```
mat[0:3,1:2]  
  
array([[ 2],  
       [ 7],  
       [12]])
```

```
mat[4]
```

```
array([21, 22, 23, 24, 25])
```

```
mat[3:]
```

```
array([[16, 17, 18, 19, 20],  
       [21, 22, 23, 24, 25]])
```

Get the sum of all the values in mat

```
mat.sum()
```

```
325
```

Get the standard deviation of the values in mat

```
mat.std()
```

```
7.211102550927978
```

Get the sum of all the columns in mat

```
mat.sum(axis=1)
```

```
array([ 15,  40,  65,  90, 115])
```

Get the sum of all the rows in mat

```
mat.sum(axis=0)
```

```
array([55, 60, 65, 70, 75])
```