



## NumPy Exercises

Now that we've learned about NumPy let's test your knowledge. We'll start off with a few simple tasks, and then you'll be asked some more complicated questions.

**Import NumPy as np**

```
In [2]: import numpy as np
```

```
In [4]: list=[1,2,3,4,5]
```

```
In [5]: array=np.array(list)
```

```
In [7]: array
```

```
Out[7]: array([1, 2, 3, 4, 5])
```

### Create an array of 10 zeros

```
In [17]: x=np.zeros(10)
```

```
In [18]: x
```

```
Out[18]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

### Create an array of 10 ones

```
In [19]: y=np.ones(10)
```

```
In [20]: y
```

```
Out[20]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Create an array of 10 fives

```
In [5]: one=np.ones(10)*5
```

```
In [6]: one
```

```
Out[6]: array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

Create an array of the integers from 10 to 50

```
In [73]: a=np.arange(10,50)
```

```
In [74]: a
```

```
Out[74]: 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])
```

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

```
In [31]: c=np.arange(10,50,2)
```

```
In [32]: c
```

```
Out[32]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,  
              44, 46, 48])
```

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

```
In [37]: x=np.arange(0,9).reshape(3,3)
```

```
In [38]: x
```

```
Out[38]: array([[0, 1, 2],  
               [3, 4, 5],  
               [6, 7, 8]])
```

#### Create a 3x3 identity matrix

```
In [42]: x=[[1,0,0],[0,1,0],[0,0,1]]
```

```
In [44]: arr2d=np.array(x)
```

```
In [45]: arr2d
```

```
Out[45]: array([[1, 0, 0],  
               [0, 1, 0],  
               [0, 0, 1]])
```

#### Use NumPy to generate a random number between 0 and 1

```
In [56]: a=np.random.rand(1)
```

```
In [57]: a
```

```
Out[57]: array([0.4969775])
```

Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

```
In [105]: b=np.arange(25,50)
```

```
In [106]: flatten=b.flatten()
```

```
In [107]: flatten
```

```
Out[107]: array([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])
```

Create the following matrix:

```
In [97]: a=np.arange(1,17).reshape(4,4)
```

```
In [98]: a
```

```
Out[98]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12],
                [13, 14, 15, 16]])
```

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

```
In [65]: np.linspace(0,1,20)
```

```
Out[65]: 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:

```
In [4]: mat = np.arange(1,26).reshape(5,5)
        mat
```

```
Out[4]: 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]])
```

```
In [ ]: # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
        # BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
        # BE ABLE TO SEE THE OUTPUT ANY MORE
```

```
In [6]: list=[12,13,14,15],[17,18,19,20],[22,23,24,25]]
        arr=np.array(list)
        arr
```

```
Out[6]: array([[12, 13, 14, 15],
               [17, 18, 19, 20],
               [22, 23, 24, 25]])
```



```
In [ ]: # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

```
In [7]: np.random.randint(20,21)
```

```
Out[7]: 20
```

```
In [ ]: # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

```
In [8]: list1=[[2],[7],[12],[17],[22]]
arr1=np.array(list1)
arr1
```

```
Out[8]: array([[ 2],
               [ 7],
               [12],
               [17],
               [22]])
```

```
In [ ]: # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

```
In [9]: m=np.arange(21,26)
m
```

```
Out[9]: array([21, 22, 23, 24, 25])
```

```
In [ ]: # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

```
In [10]: a=np.arange(16,26).reshape(2,5)
a
```

```
Out[10]: array([[16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25]])
```