# **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 [ ]: import numpy as np
```

### Create an array of 10 zeros

```
In [8]: import numpy as np
my_array = np.zeros(10)
print(repr(my_array))
array([0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

### Create an array of 10 ones

```
In [7]: import numpy as np
    my_array = np.ones(10)
    print(repr(my_array))

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

#### Create an array of 10 fives

```
In [33]: import numpy as np
    my_array = np.full(10, 5.0)
    print(repr(my_array))
    array([5., 5., 5., 5., 5., 5., 5., 5.])
```

#### Create an array of the integers from 10 to 50

44, 45, 46, 47, 48, 49, 50])

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

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

#### Create a 3x3 identity matrix

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

```
In [13]: import numpy as np
    random_number = np.random.rand()
    print(repr(random_number))
```

0.19228912584602065

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

#### Create the following matrix:

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

# **Numpy Indexing and Selection**

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

```
In [3]: mat = np.arange(1,26).reshape(5,5)
         mat
 Out[3]: 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 [15]: import numpy as np
         subarray = mat[2:, 1:]
         print(repr(subarray))
         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 [29]: import numpy as np
         subarray = mat[3:4, 4:5]
         print(subarray)
         [[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 [27]: import numpy as np
         subarray = mat[0:3, 1:2]
         print(repr(subarray))
         array([[ 2],
                [7],
                [12]])
```

```
In [24]: # 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
         import numpy as np
In [22]:
         subarray= mat[4:5, :]
         print(repr(subarray))
         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 [20]: import numpy as np
         subarray=mat[3:5, :]
         print(repr(subarray))
         array([[16, 17, 18, 19, 20],
                [21, 22, 23, 24, 25]])
```

# Now do the following

#### Get the sum of all the values in mat

```
In [4]: import numpy as np
total_sum = np.sum(mat)
print(total_sum)
```

#### Get the standard deviation of the values in mat

```
In [5]: import numpy as np
std_deviation = np.std(mat)
print(std_deviation)
```

7.211102550927978

#### Get the sum of all the columns in mat

```
In [6]: import numpy as np
  column_sums = np.sum(mat, axis=0)
  print(column_sums)
```

[55 60 65 70 75]

Type  $\mathit{Markdown}$  and  $\mathit{LaTeX}$ :  $\alpha^2$