NumPy Exercises

Jithu Joji 21BCE1451

Import NumPy as np

```
import numpy as np
```

Create an array of 10 zeros

```
zeros_array = np.zeros(10)
zeros_array
```

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

Create an array of 10 ones

```
ones_array = np.ones(10)
ones_array
```

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

Create an array of 10 fives

```
fives_array = np.full(10, 5)
fives_array
```

```
array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
```

Create an array of the integers from 10 to 50

```
array_values = np.arange(10, 51)
array_values
```

```
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

```
even_array = np.arange(10, 51, 2)
even_array
```

```
array([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

```
array_2d = np.array([[0, 1, 2],[3, 4, 5],[6, 7, 8]])
array_2d
```

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

Create a 3x3 identity matrix

```
identity_matrix = np.eye(3)
identity_matrix
```

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

Use NumPy to generate a random number between 0 and 1

```
random_number = np.random.rand()
random_number
```

```
0.8524420533862203
```

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

```
random_numbers = np.random.randn(25)
random_numbers
```

```
array([-0.67953381,  1.31529854, -0.55425097, -1.08043569,
1.09293202,
        0.36572518,  0.32674228, -0.26012009,  1.88456236,
0.14542627,
        0.99828867,  0.80102729, -1.17634645, -1.62719473,
0.33654313,
       -0.2619102 , -0.31200345,  1.53384079, -0.22727076, -
0.15361238,
        0.11423855, -1.70525254, -1.53048734,  0.17419219,
0.12779738])
```

Create the following matrix:

```
values = np.arange(0.01, 1.01, 0.01)
matrix = values.reshape(10, 10)
matrix
```

```
array([[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 ],
       [0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 ],
       [0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 ],
       [0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 ],
       [0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5 ],
       [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6 ],
       [0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7 ],
       [0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8 ],
       [0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9 ],
       [0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1.  ]])
```

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

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

```
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]])
```

```
result_matrix2 = mat[2:5, 1:5]
result_matrix2
```

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

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

```
result = mat[3, 4]
result
```

```
20
```

```
20
```

```
result_matrix = mat[0:3, 1:2]
result_matrix
```

```
array([[ 2],
       [ 7],
       [12]])
```

```
array([[ 2],
       [ 7],
       [12]])
```

```
result_array = mat[4, ]
result_array
```

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

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

```
result_matrix = mat[3:  ]
result_matrix
```

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

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

Now do the following

Get the sum of all the values in mat

```
total_sum = np.sum(mat)
total_sum
```

```
325
```

Get the standard deviation of the values in mat

```
std_deviation = np.std(mat)
std_deviation
```

```
7.211102550927978
```

Get the sum of all the columns in mat

```
column_sum = np.sum(mat, axis=0)
column_sum
```

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