In [1]:

```
import numpy as np
```

Q1:Create array of 10 zeros

In [2]:

```
array_of_zeros = np.zeros(10)
print(array_of_zeros)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

In [3]:

```
array_of_zeros
```

Out[3]:

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

In [ ]:

```

```

Q2:Create array of 10 ones

In [5]:

```
array_of_ones = np.ones(10)
print(array_of_ones)
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [6]:

```
array_of_ones
```

Out[6]:

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

In [ ]:

```

```

Q:Create an array of 10 fives

In [84]:

```
array_of_fives = np.ones(10,dtype=np.int32)*5
```

In [85]:

```
array_of_fives
```

Out[85]:

```
array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5], dtype=int32)
```

In [ ]:

```

```

Q3:create array of integers from 10 to 50

In [81]:

```
array_of_integers = np.arange(10,51,dtype=np.int32)
```

In [11]:

```
print(array_of_integers)
```

```
[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]
```

In [12]:

```
array_of_integers
```

Out[12]:

```
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], dtype=int32)
```

In [ ]:

```

```

Q4: Create an array of all even integers from 10 to 50

In [13]:

```
array_of_even_integers = np.array([number for number in range(10,51) if number%2==0
```

In [14]:

```
array_of_even_integers
```

Out[14]:

```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40,
42,
       44, 46, 48, 50], dtype=int32)
```

In [ ]:

```

```

Q5: Create a 3x3 matrix with values ranging from 0 to 8

In [15]:

```
matrix_1 = np.arange(0,9).reshape(3,3)
```

In [16]:

```
matrix_1
```

Out[16]:

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

In [ ]:

```

```

Q6: create a 3x3 identity matrix

In [19]:

```
identity_matrix = np.identity(3,dtype=np.int32)
```

In [20]:

```
identity_matrix
```

Out[20]:

```
array([[1, 0, 0],
       [0, 1, 0],
       [0, 0, 1]], dtype=int32)
```

In [ ]:

```

```

Q7: use numpy to generate random number between 0 and 1

In [39]:

```
random_number = np.random.rand(1)
```

In [40]:

```
random_number
```

Out[40]:

```
array([0.42730633])
```

In [ ]:

```
```

Q8: Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

In [64]:

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

In [65]:

```
random_numbers
```

Out[65]:

```
array([0.13258112, 0.39716685, 0.72457476, 0.8741997 , 0.33762189,
       0.26428672, 0.82955148, 0.66635311, 0.95781147, 0.06233736,
       0.0334605 , 0.29949276, 0.81519357, 0.36598762, 0.55738495,
       0.2054505 , 0.85261365, 0.36480802, 0.815829  , 0.24847958,
       0.24407297, 0.73746361, 0.26209342, 0.28023354, 0.42664918])
```

In [43]:

```
len(random_numbers)
```

Out[43]:

```
25
```

In [ ]:

```
```

Q9: Create following matrix

In [57]:

```
matrix_2 = np.linspace(0.01,1,100).reshape(10,10)
```

In [58]:

```
matrix_2
```

Out[58]:

```
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.  ]])
```

In [ ]:

```

```

Q10: Create an array of 20 linearly spaced points between 0 and 1

In [59]:

```
matrix_3 = np.linspace(0,1,20)
```

In [60]:

```
matrix_3
```

Out[60]:

```
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.        ])
```

In [61]:

```
len(matrix_3)
```

Out[61]:

```
20
```

In [ ]:

```

```

In [ ]:

```

```

Nump indexing and selection

In [66]:

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

In [67]:

```
matrix
```

Out[67]:

```
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]])
```

Q1

In [70]:

```
matrix[2:,1:]
```

Out[70]:

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

Q2

In [72]:

```
matrix[:3,1].reshape(3,1)
```

Out[72]:

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

Q3

In [73]:

```
matrix[4]
```

Out[73]:

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

Q4

In [74]:

```
matrix[3:]
```

Out[74]:

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

Q5

In [76]:

```
np.sum(matrix)
```

Out[76]:

```
325
```

Q6

In [77]:

```
np.std(matrix)
```

Out[77]:

```
7.211102550927978
```

Q7

In [80]:

```
np.sum(matrix,axis=0)
```

Out[80]:

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

In [ ]:

```

```