

In [2]:

```
import numpy as np
list_ = [ '1' , '2' , '3' , '4' , '5' ]
array_list = np.array(object = list_)
array_list
```

Out[2]:

```
array(['1', '2', '3', '4', '5'], dtype='<U1')
```

Q1. Is there any difference in the data type of variables list_ and array_list? If there is then write a code to print the data types of both the variables.

In [4]:

```
print(type(list_))
print(type(array_list))
```

```
<class 'list'>
<class 'numpy.ndarray'>
```

Q2. Write a code to print the data type of each and every element of both the variables list_ and arra_list.

In [5]:

```
for i in list_:
    print(type(i))
```

```
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'str'>
```

In [6]:

```
for i in array_list:
    print(type(i))
```

```
<class 'numpy.str_'>
<class 'numpy.str_'>
<class 'numpy.str_'>
<class 'numpy.str_'>
<class 'numpy.str_'>
```

Q3. Considering the following changes in the variable, array_list:

```
array_list = np.array(object = list_, dtype = int)
```

Will there be any difference in the data type of the elements present in both the variables, list_ and arra_list? If so then print the data types of each and every element present in both the variables, list_ and arra_list.

Consider the below code to answer further questions:

```
import numpy as np

num_list = [ [ 1 , 2 , 3 ] , [ 4 , 5 , 6 ] ]

num_array = np.array(object = num_list)
```

In [8]:

```
num_list = [ [ 1 , 2 , 3 ] , [ 4 , 5 , 6 ] ]
num_array = np.array(object = num_list)
```

In [9]:

```
for i in num_list:
    print(type(i))
```

```
<class 'list'>
<class 'list'>
```

In [10]:

```
for i in num_array:
    print(type(i))
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

Q4. Write a code to find the following characteristics of variable, num_array:

(i) shape

(ii) size

In [14]:

```
num_array.shape
```

Out[14]:

```
(2, 3)
```

In [15]:

```
num_array.size
```

Out[15]:

```
6
```

Q5. Write a code to create numpy array of 3*3 matrix containing zeros only, using a numpy array creation function.

In [18]:

```
arr = np.zeros((3,3))  
arr
```

Out[18]:

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

Q6. Create an identity matrix of shape (5,5) using numpy functions?

In [20]:

```
arr2 = np.eye(5)  
arr2
```

Out[20]:

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

In []: