

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.

Ans-Yes, there is a difference in data types. `list_` contains strings, whereas `array_list` contains elements of data type `numpy.str_`. You can print the data types as follows:

```
print(type(list_))          # Output: <class 'list'>
print(type(array_list))     # Output: <class 'numpy.ndarray'>
```

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

Ans-You can use a loop to print the data type of each element in both `list_` and `array_list`:

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

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

Q3. Considering the following changes in the variable, `array_list`: `array_list = np.array(object = list_, dtype = int)`

Ans-After changing the data type to `int` when creating `array_list`, the data type of the elements in `array_list` will be integers, while `list_` will still contain strings. You can print the data types as follows:

```
for item in list_:
    print(type(item))          # Output: <class 'str'>
```

```
for item in array_list:
    print(type(item))          # Output: <class 'numpy.int64'>
```

Now, let's move on to the second code snippet:

Code Snippet 2:

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

In this code snippet, `num_list` is a list of lists, and `num_array` is a NumPy array created from `num_list`. The elements of `num_array` will have the same data type as the elements of `num_list`, which are integers.

If you want to print the data type of each element in `num_array`, you can do it using a loop in a similar way as in Q2 from the previous set of code:

```
for row in num_array:
    for item in row:
        print(type(item))
```

This code will print the data type of each element in the `num_array`.

Q4. Write a code to find the following characteristics of variable, `num_array`:
(i)`arr.shape` (ii)`arr.size`

Ans-import numpy as np

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

```
# (i) arr.shape
shape = num_array.shape
```

```
# (ii) arr.size
size = num_array.size
```

```
print("Shape of num_array:", shape)
print("Size of num_array:", size)
```

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

Ans-import numpy as np

```
# Create a 3x3 array filled with zeros
zeros_array = np.zeros((3, 3))
```

```
print(zeros_array)
```

In this code, we use np.zeros((3, 3)) to create a 3x3 array filled with zeros, and then we print the resulting zeros_array.

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

```
import numpy as np
```

```
# Create a 5x5 identity matrix
identity_matrix = np.eye(5)
```

```
print(identity_matrix)
```

```
[[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.]]
```