MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Dr. Vishwanath Karad

MIT WORLD PEACE UNIVERSITY | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering and Technology

Lab Assignment-08

08-Using a numpy module create an array and check the following: 1. Type of array 2. Axes of array 3. Shape of array 4. Type of elements in array.

What is NumPy ?

- It is a library consisting of multidimensional array objects and a collection of routines for processing of array.
- NumPy is an open-source project that can be used freely.
- NumPy was created in 2005 by **Travis Oliphant.**
- NumPy stands for **Numerical Python.**
- We use python NumPy array instead of a list because of the below three reasons
    - NumPy arrays are multi-dimensional, while lists are one-dimensional
    - NumPy arrays are homogeneous, while lists are heterogeneous
    - NumPy arrays are more efficient, required less memory and fast than Python list

Creating Arrays

- We can create a NumPy ndarray object by using the array() function.
- To create an ndarray, we can pass a list, tuple or any array-like object into the array() method, and it will be converted into an ndarray
- The shape of an array is the number of elements in each dimension.

```
import numpy
arr = numpy . array ([1 , 2, 3, 4, 5])
print ( arr )
```

```
import numpy as np
a = np.array([1, 2, 3, 4, 5])
print(a)
print(type(a))
```

Output:

```
[1 2 3 4 5]
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Dimensions- Arrays (1-D,2-D…)

```
import numpy as np
# 0-D Arrays : scalar
arr = np. array (42)
print ( arr )

arr1 = np.array([1, 2, 3, 4, 5])
print(arr1)

arr2 = np.array([[1, 2, 3], [4, 5,
6]])
print(arr2)

arr3 = np.array([[[0, 1], [2, 3]],
[[4, 5], [6, 7]]])
```

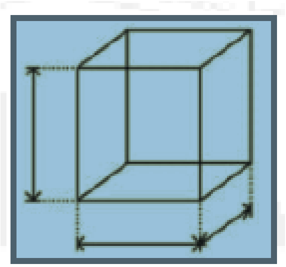
Output:

```
42

[1 2 3 4 5]

[[1 2 3] [4 5 6]]

[[[0, 1], [2, 3]], [[4, 5],
[6, 7]]]
```

ndim

- You can find the dimension of the array, whether it is a two-dimensional array or a single dimensional array.

```
import numpy as np
a = np. array ([(1 ,2 ,3) ,(4 ,5 ,6) ])
print (a. ndim )
```



dtype

- You can find the data type of the elements that are stored in an array
- we can find the size and shape of the array using size and shape function

```
import numpy as np
a = np. array ([(1 ,2 ,3) ])
print (a. dtype )

arr = arr.astype('float64')

# Print the array after changing the data type
print(arr)
print(arr.dtype)

s = np.array(['Ram', 'Robert', 'Rahim'])
s.dtype
```

Output :

```
Int32
[[1. 2. 3.]]
float64
```

shape

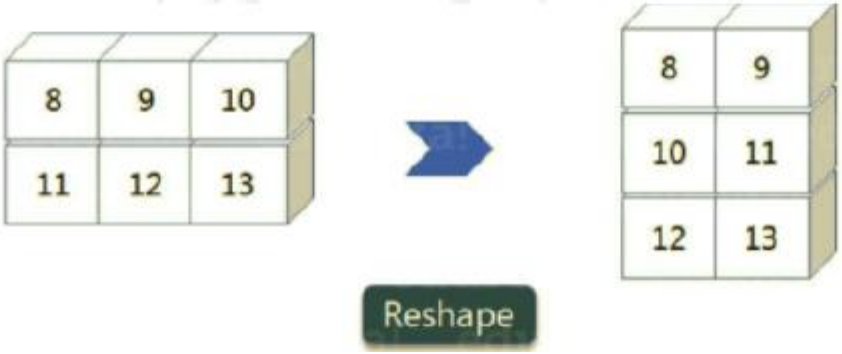
- The shape of an array is the number of elements in each dimension.
- NumPy arrays have an attribute called shape that returns a tuple with each index having the number of corresponding elements.

```
import numpy as np

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

print(arr.shape)


arr = np.array([1, 2, 3, 4], ndmin=5)

print(arr)
print('shape of array :', arr.shape)
```

reshape

- Reshape is when you change the number of rows and columns which gives a new view to an object.

```
import numpy as np
a = np. array ([(8 ,9 ,10) ,(11 ,12 ,13) ])
print (a)
a=a. reshape (3 ,2)
print (a)
 a = np.array([1,2,3,4,5,6])
 a = a.reshape(3,2)
```



Basic operations on Arrays

```
a = np.array([1, 2, 3, 4])
print(a + 1)
print(a**2)
b = np.ones(4) + 1
print(b)
b = np.zeros((3,3)) + 1
print(b)
c = np.eye(3)
d = np.eye(3, 2)
a = np.diag([1, 2, 3, 4])
a
print(a[2, 2])
```

Output :

```
[2 3 4 5]
[ 1  4  9 16]
[2. 2. 2. 2.]
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
array([[1., 0.],
       [0., 1.],
       [0., 0.]])
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
```

Basic operations on Arrays

- arange is an array-valued version of the built-in Python range function

```
a = np.arange(10) # 0.... n-1
a
print(a[5])
b = np.arange(1, 10, 2)
b
a = np.linspace(0, 1, 6)
a
```

Output :

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
array([1, 3, 5, 7, 9])
array([0. , 0.2, 0.4, 0.6, 0.8, 1. ])
```

Basic operations on Arrays

```
import numpy as np
x = np. array ([(8 ,9 ,10) ,(11 ,12 ,13) ])
y =np. array ([(1 ,2 ,3) ,(4, 5 ,6) ])
print(x+y)
print(np.add(x,y))
print(np.subtract(x,y))
print(np.sum(x))
print(np.sum(x,axis=0))
print(np.sum(x,axis=1))
```

References

- https://www.javatpoint.com/python-features
- https://www.w3schools.com/python/python_intro.asp
- https://www.geeksforgeeks.org/python-data-types/
- https://www.tutorialsteacher.com/python/python-data-types
- https://www.programiz.com/python-programming/variables-datatypes
- http://sthurlow.com/python/lesson06/
- https://subscription.packtpub.com/book/application_development/9781789800111/1/ch01lvl1sec04/lists-sets-strings-tuples-and-dictionaries
- https://www.edureka.co/blog/data-structures-in-python/
- https://www.guru99.com/if-loop-python-conditional-structures.html
- https://beginnersbook.com/2018/01/python-functions/