12-08-2022

August 12, 2022

0.1 numpy version and initialisaion

[3]:

```
import numpy
import numpy as np
print(np.__version__)
```

```
1.21.5
```

0.2 new array using arange and converting it to integer

[3]:

```
arr=np.zeros(5, dtype=np.int64 )
print(arr)
```

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

0.3 array shape

[4]:

```
ones_array = np.ones( (2, 4), dtype=np.int64 )
print(ones_array)
print(ones_array.shape)
```

```
[[1 1 1 1]
 [1 1 1 1]]
(2, 4)
```

0.4 converting an array from 1D to 2D and D

[16]:

```
array1 = np.arange( 12 )
print(array1)
new = array1.reshape(4, 3)shape
print("array in 2D is: \n",new)
new1 = array1.reshape(2, 3, 2)
print("array in 3D is:\n",new1)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11]
array in 2D is:
 [[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
array in 3D is:
 [[[ 0  1]
  [ 2  3]
  [ 4  5]]

 [[ 6  7]
  [ 8  9]
  [10 11]]]
```

0.5 resizing an array

[17]:

```
a = np.arange( 12 )
print(a)
a.resize(4,4)
print(a)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [ 0  0  0  0]]
```

0.6 Slicing

[26]:

```
arra = np.arange(15)
print(arra)
print(arra[1:5])
print(arra[2:])
print(arra[:10])
print(arra[-3:-1])
print(arra[0:10:2])
print(arra[1:-3])
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
[1 2 3 4]
[ 2  3  4  5  6  7  8  9 10 11 12 13 14]
[0 1 2 3 4 5 6 7 8 9]
[12 13]
[0 2 4 6 8]
[ 1  2  3  4  5  6  7  8  9 10 11]
```

0.7 1. Write a mumpy program to create an array with values ranging from 1 to 15 and do the following operations.

0.7.1 a) Find size of the array

0.7.2 b) Update its 8th element

0.7.3 c) Reverse the array

0.7.4 d) Convert it into 7 x 2 matrix

[7]:

```
ar = np.arange(1,16)
print(ar)
#a
print("size of the array is: ",ar.size)
#b
ar[7]=19
print(ar)
#cshape
print("array in reverse order:  ",ar[::-1])
#d
ar.resize(7,2)
print("array after resizing: \n",ar)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
size of the array is:  15
[ 1  2  3  4  5  6  7 19  9 10 11 12 13 14 15]
array in reverse order:   [15 14 13 12 11 10  9 19  7  6  5  4  3  2  1]
array after resizing:
 [[ 1  2]
 [ 3  4]
 [ 5  6]
 [ 7 19]
 [ 9 10]
 [11 12]
 [13 14]]
```

0.8 2. Write a numpy program to create a 5 x 4 matrix with values ranging from 12 to 32. Print its shape, type and data type

[8]:

```
a1 = np.arange(12,32)
a1.resize(5,4)
print("array is: \n",a1)
print("array shape is: ",a1.shape)
print("array type is: ",type(a1))
print("array data type is: ",a1.dtype)
```

```
array is:
 [[12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]
 [24 25 26 27]
 [28 29 30 31]]
array shape is:  (5, 4)
array type is:  <class 'numpy.ndarray'>
array data type is:  int64
```