

Assignment(ESHA IQBAL) ¶

```
In [ ]: #Q1
import numpy as np
```

```
In [4]: #Q2
L1=np.arange(0,10)
print(L1)
print(type(L1))

[0 1 2 3 4 5 6 7 8 9]
<class 'numpy.ndarray'>
```

```
In [5]: L2=10+np.arange(40)#Q3
```

```
In [6]: import numpy as np #Q6
print(np.__version__)
print(np.show_config())
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-6-f6c63daed2e0> in <module>
      1 import numpy as np
----> 2 print(np.__version__)
      3 print(np.show_config())

~\anaconda3\lib\site-packages\numpy\__init__.py in __getattr__(attr)
    218         else:
    219             raise AttributeError("module {!r} has no attribute "
--> 220                                     "{!r}".format(__name__, attr))
    221
    222     def __dir__():

AttributeError: module 'numpy' has no attribute '__version__'
```

```
In [9]: import numpy as np #Q7
x = np.arange(10, 49)
print("Array from 10 to 49:")
print(x)
```

```
Array from 10 to 49:
[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]
```

```
In [12]: #Q8
sample_arr = [True]

# Create a numpy array with random True or False of size 10
bool_arr = np.random.choice(sample_arr, size=10)
print('Numpy Array: ')
print(bool_arr)
```

Numpy Array:
[True True True True True True True True True True]

```
In [15]: #Q9
# 2D array
rows, cols = (5, 5)
arr = [[0]*cols]*rows
print(arr)
```

[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

```
In [18]: #Q10
#3D array
array = np.arange(27).reshape(3,3,3)
array
```

```
Out[18]: array([[[ 0,  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, 26]]])
```

```
In [19]: #Q11
import numpy as np

# initialising numpy array
ini_array = np.array([1, 2, 3, 6, 4, 5])

# printing initial ini_array
print("initial array", str(ini_array))

# printing type of ini_array
print("type of ini_array", type(ini_array))

# using shortcut method to reverse
res = ini_array[::-1]

# printing result
print("final array", str(res))

initial array [1 2 3 6 4 5]
type of ini_array <class 'numpy.ndarray'>
final array [5 4 6 3 2 1]
```

```
In [22]: #Q12
import numpy as np
Z = np.zeros(10)
Z[4] = 1
print(Z)

[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
```

```
In [23]: #Q13
import numpy as np
Z = np.eye(3)#create identity matrix
print(Z)

[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

```
In [30]: #Q14
import numpy as np
int_array=np.array([1,2,3,4,5])
float_array = int_array.astype(float)
print(float_array)

[1. 2. 3. 4. 5.]
```

```
In [33]: #Q15
import numpy as np
arr1=np.array([[1.,2.,3.],
               [4.,5.,6.,]])
arr2=np.array([[0.,4.,1.],
               [7.,2.,12.]])
print ("1st Input  number : ", arr1)
print ("2nd Input  number : ", arr2)

out_num = np.multiply(arr1,arr2)
print ("output number : ", out_num)
```

```
1st Input  number : [[1. 2. 3.]
 [4. 5. 6.]]
2nd Input  number : [[ 0.  4.  1.]
 [ 7.  2. 12.]]
output number : [[ 0.  8.  3.]
 [28. 10. 72.]]
```

```
In [35]: #Q16
import numpy as np
arr1=np.array([[1.,2.,3.],
               [4.,5.,6.,]])
arr2=np.array([[0.,4.,1.],
               [7.,2.,12.]])
arr1==arr2
```

```
Out[35]: array([[False, False, False],
 [False, False, False]])
```

```
In [41]: #Q17
import numpy as np
array=np.arange(1,10,2)
print("Array of all the odd integers from 0 to 9")
print(array)
```

```
Array of all the odd integers from 0 to 9
[1 3 5 7 9]
```

```
In [47]: #Q18
import numpy as np
a = np.array([1,10])
rm = np.array([5,6,7,8])
# np.in1d return true if the element of `a` is in `rm`
#idx = np.in1d(a, rm)
idx
```

```
File "<ipython-input-47-27d84aee0d6e>", line 4
```

```
del = np.array([5,6,7,8])
```

```
^
```

```
SyntaxError: invalid syntax
```

```
In [50]: #Q18
import numpy as np
arr = np.array([1,2,3,4,5,6,7,8,9,10])
print(arr)
```

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

```
In [51]: new_arr = np.delete(arr, [5,6,7,8])
print(new_arr)
```

```
[ 1  2  3  4  5 10]
```

```
In [52]: #Q19
import numpy as np
x = np.ones((5,5))
print("Original array:")
print(x)
print("1 on the border and 0 inside in the array")
x[1:-1,1:-1] = 0
print(x)
```

Original array:

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

1 on the border and 0 inside in the array

```
[[1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1.]]
```

```
In [54]: #Q20
import numpy as np
arr2d = np.array([[1,2,3],
                  [4,5,6],
                  [7,8,9]])
print(arr2d)
```

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

```
In [56]: new_arr2d = np.delete(arr2d, [5,6,7,8,9])
print(new_arr2d)
```

```
[1 2 3 4 5]
```

C:\Users\Zain\anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: in the future out of bounds indices will raise an error instead of being ignored by `numpy.delete`.

"""Entry point for launching an IPython kernel.

```
In [62]: #Q21
import numpy as np
array1 = np.arange(9).reshape(3,3)
array1
```

```
Out[62]: array([[0, 1, 2],
               [3, 4, 5],
               [6, 7, 8]])
```

```
In [69]: array1[1:5]
```

```
Out[69]: array([[3, 4, 5],
               [6, 7, 8]])
```

```
In [70]: #Q22
import numpy as np
array2 = np.arange(9).reshape(3,3)
array2
```

```
Out[70]: array([[0, 1, 2],
               [3, 4, 5],
               [6, 7, 8]])
```

```
In [71]: array2[2:8]
```

```
Out[71]: array([[6, 7, 8]])
```

```
In [73]: #Q23
import numpy as np
x = np.random.random((10,10))
print("Original Array:")
print(x)
xmin, xmax = x.min(), x.max()
print("Minimum and Maximum Values:")
print(xmin, xmax)
```

Original Array:

```
[[0.63038918 0.37878346 0.06849832 0.901138    0.58819554 0.471852
  0.30964996 0.77044709 0.39526611 0.43884429]
 [0.06603159 0.24079548 0.88305896 0.92505946 0.69762975 0.67895085
  0.20166234 0.47807753 0.55737588 0.49134451]
 [0.72733166 0.16268    0.62354061 0.44188358 0.33220825 0.47320975
  0.35405752 0.8414911  0.64509317 0.81621156]
 [0.63024727 0.10639806 0.73663837 0.28111657 0.70472624 0.74320296
  0.53091408 0.75529557 0.6080081  0.79622208]
 [0.96279317 0.97811767 0.08930825 0.69027462 0.7523787  0.71043371
  0.23008325 0.73450287 0.09252509 0.25080269]
 [0.14291689 0.26372442 0.42378707 0.31620763 0.77316915 0.48477203
  0.73114836 0.3741015  0.03520913 0.53524806]
 [0.74373669 0.15823295 0.62669731 0.98548471 0.94724022 0.29946025
  0.45423032 0.15681807 0.20330654 0.16149493]
 [0.43089829 0.56190632 0.56754505 0.66182601 0.70225477 0.105122
  0.18921858 0.58170417 0.71884213 0.20787184]
 [0.09814111 0.28461076 0.51810039 0.07155333 0.73541944 0.26772805
  0.19442098 0.53521671 0.82201156 0.64744445]
 [0.64689585 0.7765091  0.53586477 0.98309318 0.18944503 0.84668571
  0.15693185 0.73304812 0.91179535 0.31955677]]
```

Minimum and Maximum Values:

```
0.03520912870741433 0.985484712691807
```

In []: