

## NumPy Lab Assignment

### Exercise 1: Cast a python list to a numpy array

```
In [1]: #import numpy
import numpy as np

#create a list
my_list = [10, 20, 30, 40, 50]

#display the list
print(my_list)
print("This is done by Bandish Patel")
```

```
[10, 20, 30, 40, 50]
This is done by Bandish Patel
```

```
In [2]: #cast that list as an array
arr = np.array(my_list)
#we get back an array as the container for that list object
# note this is one dimensional array
print(arr)
print("This is done by Bandish Patel")
```

```
[10 20 30 40 50]
This is done by Bandish Patel
```

```
In [3]: # assign this to an array
arr = [10, 20, 30, 40, 50]
# display arr
print(arr)
print("This is done by Bandish Patel")
```

```
[10, 20, 30, 40, 50]
This is done by Bandish Patel
```

### Exercise 2: Cast a list of lists to a two dimensional numpy array

```
In [6]: #mat is a list of four lists
mat1=[[10,20,30], [41,52,62], [71,18,19],[100,200,230]]

# Let us display it
mat1
```

```
Out[6]: [[10, 20, 30], [41, 52, 62], [71, 18, 19], [100, 200, 230]]
```

```
In [4]: #Let us cast this to an array
mat1 = [[10, 20, 30], [41, 52, 62], [71, 18, 19], [100, 200, 230]]
arr = np.array(mat1)
print(arr)
print("This is done by Bandish Patel")
# I will get back two dimensional array. The two sets of brackets and the way it is
# that it is a two dimensional array
```

```
[[ 10  20  30]
 [ 41  52  62]
 [ 71  18  19]
 [100 200 230]]
```

This is done by Bandish Patel

### Exercise 3: Using NumPy generation methods arrange to create arrays

In [5]: *#use arrange method to create the following numpy array [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]*

```
import numpy as np
arr = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
arr = np.arange(16)
print(arr)
print("This is done by Bandish Patel")
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

This is done by Bandish Patel

In [6]: *#np.arange using start, stop, and step arguments to create the following numpy arr*

```
arr = np.arange(0, 25, 4)
print(arr)
print("This is done by Bandish Patel")
```

```
[ 0  4  8 12 16 20 24]
```

This is done by Bandish Patel

### Exercise 4:

In [7]: *#create the two dimensional 4 x 4 numpy array of zeros*

```
array = np.zeros((4,4))
print(array)
print("This is done by Bandish Patel")
```

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

This is done by Bandish Patel

In [8]: *#create the two dimensional 5 x 5 numpy array of ones*

```
array_ones = np.ones((5,5))
print(array_ones)
print("This is done by Bandish Patel")
```

```
[[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.]]
```

This is done by Bandish Patel

In [9]: *#create a two dimensional identity matrix of size 6 x6*

```
arr = np.eye(6)
print(arr)
print("This is done by Bandish Patel")
```

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

This is done by Bandish Patel

```
In [10]: #Create numpy array from 0 to 19 elements with a Length of 20
arr = np.arange(20)

#display arr
print(arr)
print("This is done by Bandish Patel")
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

This is done by Bandish Patel

*\*Exericse 5*

```
In [11]: #consider the following numpy array
mylist = [[10, 21, 32, 40],[3, 4, 5, 6], [50, 6, 70, 8]]
#cast it to an array arr2
arr2 = np.array(mylist)

#find max
maximum = np.max(arr2)
print(maximum)
print("This is done by Bandish Patel")
```

```
70
```

This is done by Bandish Patel

```
In [12]: #find min
minimum = np.min(arr2)
print(minimum)
print("This is done by Bandish Patel")
```

```
3
```

This is done by Bandish Patel

```
In [13]: #Find the location of the min value
min_location = np.unravel_index(np.argmin(arr2), arr2.shape)
print(min_location)
print("This is done by Bandish Patel")
```

```
(1, 0)
```

This is done by Bandish Patel

### Exercise 6-NumPy Scalar Operations

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

```
In [26]: #to hide warnings from jupyter notebook
```

```
In [14]: #consider the following numpy array
```

```
list2 = [[11, 21, 3, 4],[3, 41, 5, 6], [51, 6, 71, 8]]

#cast it to a numpy array arr2
arr2 = np.array(list2)
print(arr2)
print("This is done by Bandish Patel")
```

```
[[11 21  3  4]
 [ 3 41  5  6]
 [51  6 71  8]]
This is done by Bandish Patel
```

```
In [15]: #multiply each element by 6
arr_multiplied = arr2 * 6
print(arr_multiplied)
print("This is done by Bandish Patel")
```

```
[[ 66 126  18  24]
 [ 18 246  30  36]
 [306  36 426  48]]
This is done by Bandish Patel
```

```
In [16]: #raise every element to the power 4
arr_powered = arr2 ** 4
print(arr_powered)
print("This is done by Bandish Patel")
```

```
[[ 14641  194481      81    256]
 [      81 2825761    625   1296]
 [6765201    1296 25411681  4096]]
This is done by Bandish Patel
```

### Exercise 6 descriptive statistics

```
In [17]: # get the average of numbers in a numpy array arr2
average = np.mean(arr2)
print(average)
print("This is done by Bandish Patel")
```

```
19.166666666666668
This is done by Bandish Patel
```

```
In [18]: # get the median of numbers in a numpy array aar2
med = np.median(arr2)
print(med)
print("This is done by Bandish Patel")
```

```
7.0
This is done by Bandish Patel
```

- Let us create another numpy array that has elements repeated more than once
- arr3=np.append([10, 20, 30], [[4, 5, 6], [40, 40, 60]])

```
In [19]: #create the array mentioned above
#compute the mode for arr3
arr3=np.append([10, 20, 30], [[4, 5, 6], [40, 40, 60]])
```

```

#get the mode of number in a numpy array
#import the required library
import numpy as np
from scipy.stats import mode

# get the mode of arr3
arr_mode = mode(arr3)

#print the mode
print(arr_mode)
print("This is done by Bandish Patel")

```

ModeResult(mode=40, count=2)  
This is done by Bandish Patel

### Exercise 7

- Get the min max mean variance skewness and kurtosis for the following numpy array
- b= np. arange (15)

```

In [20]: #create a numpy array using arange
b = np.arange(15)

#import required library
import numpy as np
from scipy.stats import describe

#Get all the required values in one statement
min_val, max_val, mean_val, variance_val, skewness_val, kurtosis_val = describe(b).
print(min_val)
print(max_val)
print(mean_val)
print(variance_val)
print(skewness_val)
print(kurtosis_val)
print("This is done by Bandish Patel")

```

0  
14  
7.0  
20.0  
0.0  
-1.210714285714286  
This is done by Bandish Patel

### Exercise 8: Slicing Two Dimensional Array ---Matrix

- Consider the following two dimensional array
- arr4 = np.array([[35,20,15],[40,115,25],[15,20,75],[30,10,35]])

```

In [21]: #create the array arr4 mentioned above

```

```
arr4 = np.array([[35, 20, 15], [40, 115, 25], [15, 20, 75], [30, 10, 35]])

#print it
print(arr4)
print("This is done by Bandish Patel")
```

```
[[ 35  20  15]
 [ 40 115  25]
 [ 15  20  75]
 [ 30  10  35]]
This is done by Bandish Patel
```

```
In [22]: ##to get the second and third row of arr4
sec_row = arr4[1]
third_row = arr4[2]
print(sec_row)
print(third_row)
print("This is done by Bandish Patel")
```

```
[ 40 115  25]
[15 20 75]
This is done by Bandish Patel
```

```
In [23]: #if you want submatrices of a matrix not elements you can use : for slice notations
#let us say that I want to grap 20,15,115,25,20,75 i.e. the submatrix at the top ri
submatrix_top_right = arr4[:2, 1:]
print(submatrix_top_right)
print("This is done by Bandish Patel")
```

```
[[ 20  15]
 [115  25]]
This is done by Bandish Patel
```

```
In [24]: #Let us say that we want to know all the elements of the array arr4 that are > 30
greater = arr4[arr4 > 30]
print(greater)
print("This is done by Bandish Patel")
```

```
[ 35  40 115  75  35]
This is done by Bandish Patel
```

```
In [ ]:
```