Inclass - Lab (Day 1)

Table of Content

Import the NumPy library before solving the exercises

Let's begin with some hands-on practice exercises

1. One Dimensional Array

1. Write a NumPy program to convert a list of numeric values into a one-dimensional NumPy array

Use the list given below: (or create your own list)

```
list_temperature = [3.04, 1.65, 5.76, 1.66, 1.54, 4.69, 2.68]
```

In [15]:

```
# type your code here
import numpy as np
list_temperature = [3.04, 1.65, 5.76, 1.66, 1.54, 4.69, 2.68]
n = np.array(list_temperature)
print(n)
print(type(n))
```

```
[3.04 1.65 5.76 1.66 1.54 4.69 2.68]
<class 'numpy.ndarray'>
```

2. Create a 1D array of weights of 10 students and retrieve all the weights greater than 68

Use the weights given below: (or create your own list)

```
weights = [74.2, 85, 74, 67.9, 52, 70.5, 86, 51.8, 64, 82]
```

In [4]:

```
# type your code here
weights = [74.2, 85, 74, 67.9, 52, 70.5, 86, 51.8, 64, 82]
n_w = np.array(weights)
n_w[n_w>68]
```

Out[4]:

```
array([74.2, 85. , 74. , 70.5, 86. , 82. ])
```

3. Write a code to generate 10 random integers between 20 to 40 (Is it possible to generate same random numbers everytime? If yes, describe the function)

In [6]:

```
# type your code here
n1 = np.arange(20,40)
n1
```

Out[6]:

```
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
       37, 38, 39])
```

4. Write a program to get the fifth power of the first five positive integers

In [11]:

```
# type your code here
n4 = np.arange(6)
print(n4**5)
```

```
[   0    1   32  243 1024 3125]
```

5. Write a program to create an array of all the odd integers from 50 to 100

In [9]:

```
# type your code here
n5 = np.arange(50,101)
n5[n5%2!=0]
```

Out[9]:

```
array([51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83,
       85, 87, 89, 91, 93, 95, 97, 99])
```

6. Write a program to calculate the sum of each column

Use the array given below: (or create your own array)

```
A = [[1, 4, 8], [5, 7, 3], [9, 14, 2]]
```

In [3]:
```
# type your code here
A = [[1, 4, 8], [5, 7, 3], [9, 14, 2]]
n33 = np.array(A)
result = n33.sum(axis=0)
print(result)
```

```
[15 25 13]
```

7. Write a code to get the element-wise remainder of an array after division by 8

Use the list given below:

```
num = [42, 87, 90, 14, 32, 75, 61, 80, 92]
```

In [7]:
```
# type your code here
num = [42, 87, 90, 14, 32, 75, 61, 80, 92]
n7_num = np.array(num)
n7_num
for item in n7_num:
    print(item,' remainder after dividing by 8 is: ',item//8)
```

```
42  remainder after dividing by 8 is:  5
87  remainder after dividing by 8 is:  10
90  remainder after dividing by 8 is:  11
14  remainder after dividing by 8 is:  1
32  remainder after dividing by 8 is:  4
75  remainder after dividing by 8 is:  9
61  remainder after dividing by 8 is:  7
80  remainder after dividing by 8 is:  10
92  remainder after dividing by 8 is:  11
```

2. Two Dimensional Array

8. Create a 2D numpy array from a 1D array of 20 elements of your choice. Further, subset the first 5 elements from the second element of a resulting 2D array

In [14]:
```
# type your code here
n8 = np.random.randint(low=10,high=100,size=20)
n8.reshape(5,4)
```

Out[14]:
```
array([[67, 40, 93, 62],
       [78, 25, 85, 73],
       [83, 20, 49, 44],
       [53, 27, 16, 68],
       [95, 81, 89, 37]])
```

9. Write a program to find common values between two arrays

Use the lists given below:

```
prices_monday = [10, 30, 50, 60, 90]

prices_tuesday = [50, 20, 80, 125, 10, 70]
```

In [10]:

```
# type your code here
prices_monday = [10, 30, 50, 60, 90]
prices_tuesday = [50, 20, 80, 125, 10, 70]
n9_a = np.array(prices_monday)
n9_b = np.array(prices_tuesday)
print(np.intersect1d(n9_a, n9_b))
```

```
[10 50]
```

10. Write a program to find BMI of the patients whose height and weight is provided in the list given below

Use the lists given below:

```
list of weights (in kg)
weights = [57, 69, 54, 65 ,71, 62, 56]

list of heights (in m)<br>
heights = [1.59, 1.75, 1.66, 1.74, 1.64, 1.72, 1.53]
```

In [55]:

```
# type your code here
weights = [57, 69, 54, 65 ,71, 62, 56]
heights = [1.59, 1.75, 1.66, 1.74, 1.64, 1.72, 1.53]
np_w = np.array(weights)
np_h = np.array(heights)

def bmi(x,y):
    print(x/y*y)

bmi(np_w,np_h)
```

```
[57. 69. 54. 65. 71. 62. 56.]
```

11. Write a program to add a new column ([5,7]) at the last position in the array given below

Use the array given below:

```
array1 = [[[3, 4], [8, 2],[5, 9]]]
```

In [56]:

```
# type your code here
array1 = np.array([[[3, 4], [8, 2],[5, 9]]])
x = np.array([[[5,7]]])
np.append(array1,x,axis=1)
```

Out[56]:

```
array([[[3, 4],
        [8, 2],
        [5, 9],
        [5, 7]]])
```

12. Write a code to print the identity matrix of order 5

In [26]:

```
# type your code here
n12 = np.identity(5)
n12
```

Out[26]:

```
array([[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.]])
```

13. Write a program to transpose a matrix

Use the matrix given below:

```
M = [[15, 17, 45, 56], [7, 42, 15, 63], [54, 3, 61, 41], [0, 87, 16, 20]]
```

In [31]:

```
# type your code here
M = [[15, 17, 45, 56], [7, 42, 15, 63], [54, 3, 61, 41], [0, 87, 16, 20]]
np13 = np.array(M)
print("The original matrix is :")
print(np13)
res_np13 = np.transpose(np13)
print("The matrix after transpose is : ")
print(res_np13)
```

```
The original matrix is :
[[15 17 45 56]
 [ 7 42 15 63]
 [54  3 61 41]
 [ 0 87 16 20]]
The matrix after transpose is :
[[15  7 54  0]
 [17 42  3 87]
 [45 15 61 16]
 [56 63 41 20]]
```

3. Stacking

14. Write a program to combine the two arrays vertically

Use the arrays given below: (or create your own array)

```
A1 = [[5, 9], [7, 6]]
```

```
A2 = [[8,-7], [0,4]]
```

Combine the array A2 with A1

In [42]:

```
# type your code here
A1 = [[5, 9], [7, 6]]
A2 = [[8,-7], [0,4]]
nA1 = np.array(A1)
print(nA1)
nA2 = np.array(A2)
print(nA2)
nvs = np.vstack(tup=(nA2,nA1))
print(nvs)
```

```
[[5 9]
 [7 6]]
[[ 8 -7]
 [ 0  4]]
[[ 8 -7]
 [ 0  4]
 [ 5  9]
 [ 7  6]]
```

15. Write a code to combine the two arrays horizontally

Use the arrays given below: (or create your own array)

```
A1 = [[-81, 75, 40], [27, 67, 52]]
```

```
A2 = [[15, 54], [39, 56]]
```

Combine the array A2 with A1

In [41]:

```
# type your code here
A1 = [[-81, 75, 40], [27, 67, 52]]
A2 = [[15, 54], [39, 56]]
n15_A1 = np.array(A1)
print(n15_A1)
n15_A2 = np.array(A2)
print(n15_A2)
nhs = np.hstack(tup=(n15_A2,n15_A1))
print(nhs)
```

```
[[-81  75  40]
 [ 27  67  52]]
[[15 54]
 [39 56]]
[[ 15  54 -81  75  40]
 [ 39  56  27  67  52]]
```

4. Splitting

16. Write a code to divide the array into three sub-arrays containing equal number of rows

Use the array given below:

```
A = [[4, 7], [2, -3], [8, 1], [0, 9], [5,-1], [8, 3]]
```

In [37]:

```
# type your code here
A = [[4, 7], [2, -3], [8, 1], [0, 9], [5,-1], [8, 3]]
n16 = np.array(A)
np.vsplit(n16,3)
```

Out[37]:

```
[array([[ 4,  7],
        [ 2, -3]]),
 array([[8, 1],
        [0, 9]]),
 array([[ 5, -1],
        [ 8,  3]])]
```

17. Write a program to divide the array into two sub-arrays containing equal number of columns

Use the array given below:

```
B = [[8, 7, 0, 9], [2, -3, 5, 10], [8, 5, -1,3]]
```

In [38]:

```
# type your code here
B = [[8, 7, 0, 9], [2, -3, 5, 10], [8, 5, -1,3]]
n17 = np.array(B)
np.hsplit(n17,2)
```

Out[38]:

```
[array([[ 8,  7],
        [ 2, -3],
        [ 8,  5]]),
 array([[ 0,  9],
        [ 5, 10],
        [-1,  3]])]
```

5. For Loop

18. Write a program to perform one-to-one mapping of the two lists

Use the lists given below: (or create your own lists)

```
L1 = [45, 67, 56, 71, 54, 62, 56]

L2 = [21, 52, 64, 10, 49, 74, 26]
```

In [49]:

```
# type your code here
L1 = [45, 67, 56, 71, 54, 62, 56]
L2 = [21, 52, 64, 10, 49, 74, 26]
np_l1 = np.array(L1)
np_l2 = np.array(L2)
list(zip(map(lambda x:x ,(np_l1,np_l2))))
```

Out[49]:

```
[(array([45, 67, 56, 71, 54, 62, 56]),),
 (array([21, 52, 64, 10, 49, 74, 26]),)]
```

19. Define a for loop to apply the function (2x-1)/(x+5) on every element of an array

Use the list given below:

```
numbers = [6, 5, 3, 15]
```

In [50]:

```
# type your code here
numbers = [6, 5, 3, 15]
np19 = np.array(numbers)
for i in np19:
    print(((2*i)-1)/(i+5))
```

```
1.0
0.9
0.625
1.45
```

20. Create an identity matrix with n dimensions (take input from the user). Fill the diagonals of that matrix with the multiples of the number provided by the user; take another input from the user. Consider the output below as an example

In [51]:

```
# type your code here
x = int(input('enter size of matrix: '))
y = int(input('enter a number: '))
ls  = [i*y for i in range(1,x+1)]
id_array = np.identity(x)
for i in range(0,len(id_array)):
    id_array[i][i] = ls[i]
    id_array[len(id_array)-1-i][i]=ls[i]
id_array
```

```
enter size of matrix: 6
enter a number: 3
```

Out[51]:

```
array([[ 3.,  0.,  0.,  0.,  0., 18.],
       [ 0.,  6.,  0.,  0., 15.,  0.],
       [ 0.,  0.,  9., 12.,  0.,  0.],
       [ 0.,  0.,  9., 12.,  0.,  0.],
       [ 0.,  6.,  0.,  0., 15.,  0.],
       [ 3.,  0.,  0.,  0.,  0., 18.]])
```