

Name: Shravani Satish Ingle
Roll No.: 315 Class : C1
PRN No.: 202201070028

ASSIGNMENT 3

#ARRAY OPERATIONS IN PYTHON

```
import pandas as pd
import numpy as np

df = pd.read_excel('/content/drive/MyDrive/C1 Class.xlsx')
df1 = pd.read_excel('/content/drive/MyDrive/rollno.xlsx')

array = df.to_numpy()
array1 = df1.to_numpy()

print(array)
print(array1)
```

OUTPUT

```
[[301  32]
 [302  21]
 [303  36]
 [304  14]
 [305  25]
 [306  41]
 [307  12]
 [308  13]
 [309  43]
 [310  38]]
[['Shravani' 15]
 ['Divya' 13]
 ['Aditi' 28]
 ['Nandini' 84]
 ['Devyani' 8]
 ['Anushka' 5]
 ['Nikita' 19]]
```

```
resarray = np.vstack((array, array1))
print(resarray)
```

#ADITION OF ARRAYS

```
Of only one csv file (/content/drive/MyDrive/C1 Class.xlsx)
print("Addition of matrix")
print(np.add(array,array1))
```

OUTPUT

Addition of matrix

```
[[602  64]
 [604  42]
 [606  72]
 [608  28]
 [610  50]
 [612  82]
 [614  24]
 [616  26]
 [618  86]
 [620  76]]
```

#SHALLOW COPY

#in the case of shallow copy, a reference of the object is copied in another object.

#it means that any changes made to a copy of the object do reflect in the original object.

#We will be implementing shallow copy using the view() function.

```
array3 =array1.view()
```

```
print(array3)
```

OUTPUT

```
[[301  32]
 [302  21]
 [303  36]
 [304  14]
 [305  25]
 [306  41]
 [307  12]
 [308  13]
 [309  43]
 [310  38]]
```

#STACKING

```
resarray = np.vstack((array, array1))
```

```
print(resarray)
```

OUTPUT

```
[[301  32]
 [302  21]
 [303  36]
 [304  14]
 [305  25]
 [306  41]
 [307  12]
 [308  13]
 [309  43]
 [310  38]
 ['Shravani' 15]
 ['Divya' 13]]
```

```
['Aditi' 28]
['Nandini' 84]
['Devyani' 8]
['Anushka' 5]
['Nikita' 19]]
```

#TRANSPOSE

```
resarray = np.transpose(array2)
print(resarray)
```

OUTPUT

```
[[11 14 17]
 [12 15 18]
 [13 16 19]]
```

#SUBTRACTION

```
res = array1/array2
print(res)
```

OUTPUT

```
[[0.09090909 0.16666667 0.23076923]
 [0.28571429 0.33333333 0.375       ]
 [0.41176471 0.44444444 0.47368421]]
```

#

```
res = array1%array2
print(res)
```

OUTPUT

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

#HSTACK

```
resarray = np.hstack((array1,array2))
print(resarray)
```

OUTPUT

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

#SORTING

```
arr = [5, 2, 8, 7, 1];
temp = 0;
```

```
print("Elements of original array: ");
for i in range(0, len(arr)):
    print(arr[i], end=" ");
```

```
#Sort the array in ascending order
```

```
for i in range(0, len(arr)):
    for j in range(i+1, len(arr)):
```

```

        if(arr[i] > arr[j]):
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;

print();

#Displaying elements of the array after sorting

print("Elements of array sorted in ascending order: ");
for i in range(0, len(arr)):
    print(arr[i], end=" ");

```

OUTPUT

```

Elements of original array:
5 2 8 7 1
Elements of array sorted in ascending order:
1 2 5 7 8

```

#BROADCASTING

```

import numpy as np
x = np.array([11,12,13,14,15,16])
y = np.array([6,7,8])
X=x.reshape(2,3)
print("X =",X)
print("\n y =",y)
z=X+y
print("X +y =",z)

```

OUTPUT

```

X = [[11 12 13]
      [14 15 16]]

y = [6 7 8]
X +y = [[17 19 21]
        [20 22 24]]

```

#LINEAR SEARCH

```

def linearsearch(arr, x):
    for i in range(len(arr)):
        if arr[i] == x:
            return i
    return -1

arr = ['t','u','t','o','r','i','a','l']
x = 'o'
print("element found at index "+str(linearsearch(arr,x)))

```

OUTPUT

```
element found at index 3
```

#LENGTH

```
arr = [0, 1, 2, a, 4]  
print ("The given array is: ")  
print (arr)
```

Finding length of the given array

```
size = len(arr)  
print ("The length of array is: ")  
print (size)
```

OUTPUT

```
The given array is:  
[0, 1, 2, a, 4]  
The length of array is:  
5
```