



SOMAIYA
VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

Department of Science and Humanities



Batch: C5-1

Roll No.: 16010123293

Experiment / assignment / tutorial No.: Exp7

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title : NumPy library of Python

AIM: To explore the Numpy library of Python

Expected OUTCOME of Experiment:

CO2: Use Numpy Library functions

Resource Needed: Python IDE

Theory:

NumPy :A Python library used for working with arrays.

- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy stands for Numerical Python.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

For Installation of NumPy:

- pip install numpy

Example

```
import numpy
```

```
arr=numpy.array([1, 2, 3, 4, 5])
```

```
print(arr)
```

output:?

For Creation of NumPy ndarray Object:

- NumPy is used to work with arrays. The array object in NumPy is called ndarray.
- We can create a NumPy ndarray object by using the array() function.
Example:

```
import numpy as np
arr=np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

Creating ndarrays:

```
array = np.array([[0,1,2],[2,3,4]])
```

output:

```
[[0 1 2]
 [2 3 4]]
```

```
array = np.zeros((2,3))
```

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

```
array = np.ones((2,3))
```

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

```
array = np.eye(3)
```

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

```
array = np.arange(0, 10, 2)
```

```
[0, 2, 4, 6, 8]
```

```
array = np.random.randint(0, 10, (3,3))
```

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

Slicing arrays

Slicing in python means taking elements from one given index to another given index.

We pass slice instead of index like this: [start:end].

We can also define the step, like this: [start:end:step].

Arithmetic with NumPy Arrays:

Any arithmetic operations between equal-size arrays applies the operation element-wise

```
arr = np.array([[1., 2., 3.], [4., 5., 6.]])
```

```
print(arr)
```

```
[[1. 2. 3.]
 [4. 5. 6.]]
```

```
print(arr * arr)
```

```
[[ 1.  4.  9.]  
 [16. 25. 36.]]
```

```
print(arr - arr)  
[[0.  0.  0.]  
 [0.  0.  0.]]
```

- **Shape of an Array**

The shape of an array is the number of elements in each dimension.

- **Reshaping arrays**

Reshaping means changing the shape of an array.

The shape of an array is the number of elements in each dimension.

By reshaping we can add or remove dimensions or change number of elements in each dimension.

- **Iterating Arrays**

Iterating means going through elements one by one.

As we deal with multi-dimensional arrays in numpy, we can do this using basic for loop of python. If we iterate on a 1-D array it will go through each element one by one.

- **Joining NumPy Arrays**

Joining means putting contents of two or more arrays in a single array.

In SQL we join tables based on a key, whereas in NumPy we join arrays by axes.

We pass a sequence of arrays that we want to join to the concatenate() function, along with the axis. If axis is not explicitly passed, it is taken as 0.

- **Splitting NumPy Arrays**

Splitting is reverse operation of Joining.

Joining merges multiple arrays into one and Splitting breaks one array into multiple.

We use array_split() for splitting arrays, we pass it the array we want to split and the number of splits.

- **NumPy Searching Arrays**

You can search an array for a certain value, and return the indexes that get a match.

To search an array, use the where() method.

- **Sorting Arrays**

Sorting means putting elements in an ordered sequence.

Ordered sequence is any sequence that has an order corresponding to elements, like numeric or alphabetical, ascending or descending.

The NumPy ndarray object has a function called sort(), that will sort a specified array.

- **NumPy Filter Array**

Getting some elements out of an existing array and creating a new array out of them is called filtering. In NumPy, you filter an array using a boolean index list.

1.Problem statement:



Python Code	Output
<pre>import numpy as np arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8]]) print(arr.shape)</pre>	(2,4)
<pre>import numpy as np arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]) newarr = arr.reshape(4, 3) print(newarr)</pre>	<pre>[[1 2 3] [4 5 6] [7 8 9] [10 11 12]]</pre>
<pre>import numpy as np arr = np.array([1, 2, 3]) for x in arr: print(x)</pre>	<pre>1 2 3</pre>
<pre>import numpy as np arr1 = np.array([1, 2, 3]) arr2 = np.array([4, 5, 6]) arr = np.concatenate((arr1, arr2)) print(arr)</pre>	[1 2 3 4 5 6]
<pre>import numpy as np arr = np.array([1, 2, 3, 4, 5, 6]) newarr = np.array_split(arr, 3) print(newarr)</pre>	[array([1, 2]), array([3, 4]), array([5, 6])]
<pre>import numpy as np arr = np.array([1, 2, 3, 4, 5, 4, 4]) x = np.where(arr == 4) print(x)</pre>	(array([3, 5, 6], dtype=int64),)
<pre>import numpy as np arr = np.array([3, 2, 0, 1]) print(np.sort(arr))</pre>	[0 1 2 3]
<pre>import numpy as np arr = np.array([41, 42, 43, 44]) x = [True, False, True, False] newarr = arr[x] print(newarr)</pre>	[41 43]

2. Write a python program to calculate the sum of all columns in a 2D NumPy array.


3. Create two NumPy arrays representing monthly high and low temperatures for a year. Calculate the monthly average temperatures, the overall average high and low temperatures, and identify the months with the highest and lowest average temperatures.

Books/ Journals/ Websites referred:

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018, India

Implementation details:

```
4 #Q.2
5 import numpy as np
6 arr = np.array([[1,2],
7                [3,7],[6,5]])
8 def add(arr):
9     x = 0
10    y = 0
11    for i in arr:
12        x+=i[0]
13        y+=i[1]
14    print("sum of all the columns: ",x+y)
15 add(arr)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ...  Py

```
PS C:\Users\abhis> & C:/Users/abhis/AppData/Local/M
hon3.10.exe "c:/Users/abhis/OneDrive/Desktop/progra
ab.py"
sum of all the columns: 24
PS C:\Users\abhis> █
```

```

31 #Q.3
32 import numpy as np
33 high_c = np.array([32, 30, 33, 34, 35, 34, 34, 33, 33, 34, 33, 32, 31])
34 low_c = np.array([17, 17, 18, 18, 17, 17, 17, 18, 19, 18, 16, 15, 15])
35 monthly_avg_c = (high_c + low_c) / 2
36 overall_avg_high_c = np.mean(high_c)
37 overall_avg_low_c = np.mean(low_c)
38 max_index_c = np.argmax(monthly_avg_c)
39 min_index_c = np.argmin(monthly_avg_c)
40 month_names = {0: "Jan", 1: "Feb", 2: "Mar", 3: "April", 4: "May", 5: "June", 6: "July", 7: "Aug", 8: "Sept", 9: "Oct", 10: "Nov", 11: "Dec"}
41 min_index_c %= 12
42 print("Monthly Average Temperatures (Celsius):", monthly_avg_c)
43 print("Overall Average High Temperature (Celsius):", overall_avg_high_c)
44 print("Overall Average Low Temperature (Celsius):", overall_avg_low_c)
45 print("Month with the Highest Average Temperature:", month_names[max_index_c])
46 print("Month with the Lowest Average Temperature:", month_names[min_index_c])

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + v

PS C:\Users\abhis> & C:/Users/abhis/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/abhis/OneDrive/Desktop/programming in python/exp7-pplab

Monthly Average Temperatures (Celsius): [24.5 23.5 25.5 26. 26. 25.5 25.5 25.5 26. 26. 24.5 23.5 23.]

Overall Average High Temperature (Celsius): 32.92307692307692

Overall Average Low Temperature (Celsius): 17.076923076923077

Month with the Highest Average Temperature: April

Month with the Lowest Average Temperature: Jan

PS C:\Users\abhis>

Conclusion: In this experiment we learnt how to install numpy and use it to create arrays and perform different tasks with them

Post Lab Descriptive Questions

1. Generate a random integer from 0 to 100 using NumPy random function?

Ans:

```

82 import numpy as np
83 n = np.random.randint(0,100)
84 print(n)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

PS C:\Users\abhis> & C:/Users/abhis/

74

PS C:\Users\abhis>

2. Explain the slicing of 2-D Array

Ans: In numpy slicing a 2D array involves specifying ranges for rows and columns using array[start:stop:step, start:stop:step]. The first combination of start stop is for rows and the second for columns.

Date: _____

Signature of faculty in-charge