Week - 9:

Exploratio

n of

NumPy

Package

1. Import numpy, and explore their functionalities.

Import NumPy:

```
Impot
numpy as
np
```

Functionalities:

Creating Arrays:

Program:

```
import numpy as np
arr1 =
np.array([10,20,30,40,
50])
print(arr1)
print(type(arr1)
)
```

Output:

```
[10 20 30 40
50] <class
‘numpy.ndarra
y’>
```

1-D Arrays

Program:

```
import numpy as np
 arr = np.array([1,
2, 3, 4, 5])
print(arr)
print(arr[0]
```

Output:

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

2-D Arrays

Program:

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

Output:

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

Dimension **Program:**

```
import numpy as np
arr = np.array([[1, 2, 3],
[4, 5, 6]])
print(arr.ndim)
```

Output:

```
 2
```

Size of each element (in bytes) **Program:**

```
import numpy as np
arr = np.array([[1, 2, 3],
[4, 5, 6]])
print(arr.itemsize)
```

Output:

```
4
```

Datatype

Program:

```
import numpy as np  arr
= np.array([[1, 2, 3], [4,
5, 6]])  print(arr.dtype)
```

Output:

```
Int32
```

Size and Shape

Program:

```
import numpy as np  arr
= np.array([[1, 2, 3], [4,
5, 6]])  print(arr.size)
print(arr.shape)
```

Output:

```
6
(2, 3)
```

Reshape

Program:

```
import numpy as np
arr = np.array([[1, 2, 3], [4, 5, 6]])
a=arr.reshape(3,2)
print(a)
```

Output:

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

Slicing

Program:

```
import numpy as np
arr = np.array([[1, 2, 3],
[4, 5, 6]])
print(arr[0:,2])
```

Output:

```
[3 6]
```

Min/Max/Sum Program:

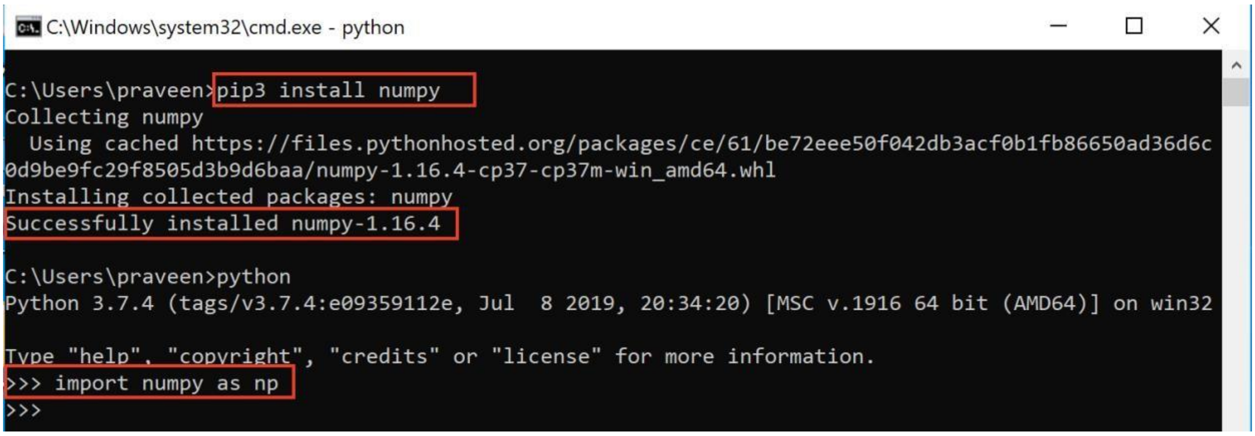
```
import numpy as np
arr = np.array([[1, 2, 3],
[4, 5, 6]])
print(arr.min())
print(arr.max())
print(arr.sum())
```

Output:

```
1
6
21
```

2. a) Install NumPy package with pip and explore it.

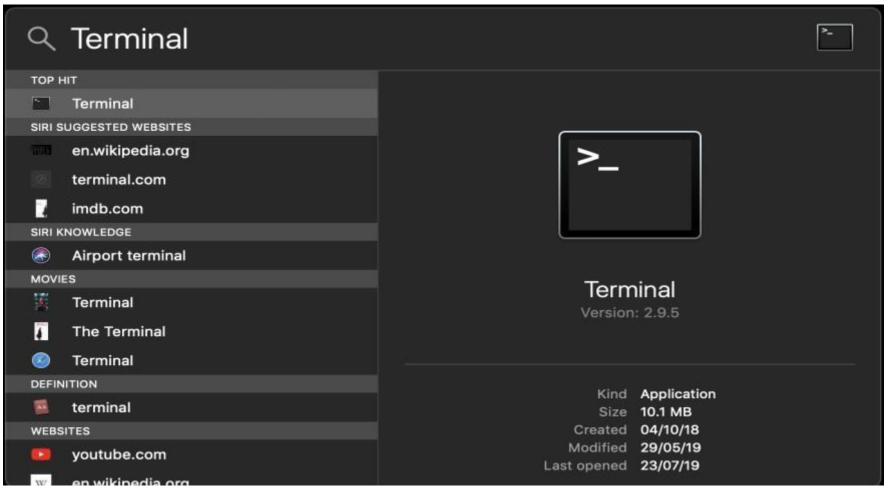
Python is not installed by default in windows operating system. You can download the required version of python from python.org. Once python is installed successfully, open command prompt and use pip to install numpy.

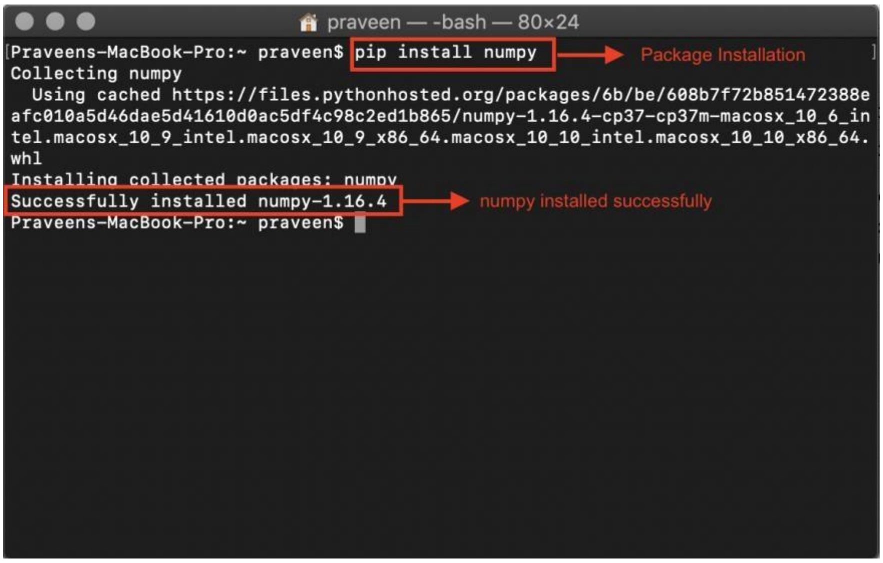


```
C:\Users\praveen>pip3 install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/ce/61/be72eee50f042db3acf0b1fb86650ad36d6c
0d9be9fc29f8505d3b9d6baa/numpy-1.16.4-cp37-cp37m-win_amd64.whl
Installing collected packages: numpy
Successfully installed numpy-1.16.4

C:\Users\praveen>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>>
```

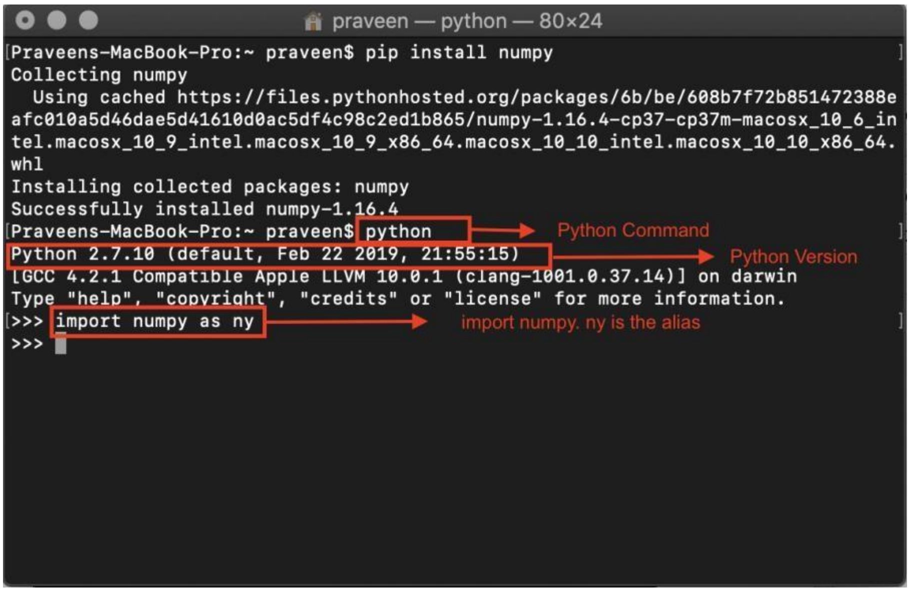




```
Praveens-MacBook-Pro:~ praveen$ pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/6b/be/608b7f72b851472388e
afc010a5d46dae5d41610d0ac5df4c98c2ed1b865/numpy-1.16.4-cp37-cp37m-macosx_10_6_in
tel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.
whl
Installing collected packages: numpy
Successfully installed numpy-1.16.4
Praveens-MacBook-Pro:~ praveen$
```

Package Installation

numpy installed successfully



```
Praveens-MacBook-Pro:~ praveen$ pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/6b/be/608b7f72b851472388e
afc010a5d46dae5d41610d0ac5df4c98c2ed1b865/numpy-1.16.4-cp37-cp37m-macosx_10_6_in
tel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.
whl
Installing collected packages: numpy
Successfully installed numpy-1.16.4
Praveens-MacBook-Pro:~ praveen$ python
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as ny
>>>
```

Python Command

Python Version

import numpy. ny is the alias

3. Write a program for slicing arrays using numpy .

Program:

```
import numpy as
np arr = np.
array([1, 2, 3, 4, 5,
6, 7])
print(arr[1:5])
print(arr[4:])
print(arr[:4])
print(arr[-3:-1])
print(arr[1:5:2])
print(arr[::2])
```

Output:

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

4. Write a program for Math operations on array using numpy.

- **Square Root and Standard Deviation**

Program:

```
                import numpy as np
arr=np.array([[1,2,3],[4,5,6]])
                print(np.sqrt(arr))

print(np.std(arr))
```

Output:

```
              [[1.         1.41421356 1.73205081]
                     [2.         2.23606798 2.44948974]]
                   1.707825127659933
```

- **Addition, subtraction, multiplication and division of the two matrices**

Prog ram:

```
import numpy as np
arr1=np.array([[1,2,
3],[4,5,6]])
arr2=np.array([[1,2,
3],[4,5,6]])
print(arr1+arr2)
print(arr1-arr2)
print(arr1*arr2)
print(arr1/arr2)
```

Output:

```
[[ 2  4  6]
 [ 8 10 12]]

[[0 0 0]
 [0 0 0]]

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

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

- **Vertical and Horizontal Stacking**

Program:

```
import numpy as np
arr1=np.array([[1,2,3],[4,5,6]])
arr2=np.array([[7,8,9],[10,11,12]])
print("vstack:",np.vstack((arr1,arr2)
))
print("hstack:",np.hstack((arr1,arr2)
))
```

Output:

```
vstack: [[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]  [10
11          12]]
hstack: [[ 1  2
3  7  8  9]  [ 4
5  6 10 11 12]]
```

- **Ravel:**

Program:

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

```
print(np.ravel(arr))
```

Output:

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

5. Write a program for searching . Program:

```
import numpy as np
arr = np.array([10, 32, 30, 50, 20,
82, 91, 45]) i = np.where(arr == 30)
print(i)
```

Output:

```
(array([2], dtype=int64),)
```

6. Write a program for sorting. Program:

```
import numpy as np
arr=np.array([[1,4,2,3],[9,13,61,1],[43,2
4,88,22]])   sort_arr=np.sort(arr)
print(sort_arr)
```

Output:

```
[[ 1  2  3  4]
 [ 1  9 13 61]
 [22 24 43 88]]
```