Date / /
Page:

Unit-4 Working with data in Python

(i) Reading files with open

(ii) Create file

content ↓ overwrite (w)

```
In [1]  f = open ("file.txt", 'x')
In [2]  f = open ("file.txt", 'w')
        f = write ("hi")
out [2]  2
```

Four types of mode

1. r = read
2. w = write
3. x = create
4. a = append

→
```
f = open ("file.txt", 'a')
f.write ('bye')
```
out [3] 3

→
```
f = open ("file.txt", 'r')
f.read ()
```
out [4] 'hibye'

→
```
f = open ("file.txt", 'r')
f.read (2)
```
→ Return the character acc. to no.

out [5] 'hi'

Teacher Signature .....

Parents Signature .....

Date / /
Page:

Working with pandas
Pandas is a python library used for working with data sets.
Python data analysis.

1 → Installation of Pandas
pip install Pandas

2 → import Pandas as Pd (Alice is not fixed)
(keyword, Python env, Alice (Nickname))

3 → Working on Pandas

```
import Pandas as Pd
mydataset = {'Cars' : ['BMW', 'Volvo', 'Ford'], 'Passing':
                [3,7,2]}
myvar = Pd.DataFrame(mydataset)
print(myvar)
```

Output

	Cars	Passing
0	BMW	3
1	Volvo	7
2	Ford	2

Data Frame is a collection of data which is represented in the form of rows & columns

Teacher Signature ..... Parents Signature .....

→ df.head = It is a operation which pics 1 to 5 rows

→ df.tail = last 5 rows

Panda Series = It is like a table column.
It is one dimensional array holding data of any type.

```
import Pandas as Pd
a = [1, 7, 2]
myvar = Pd.Series(a)
print(myvar)
```

output :

0	1
1	7
2	2

data type : int64

→ Accessing series

```
print(myvar[2])
```

output : 2

→ Create the tables

```
myvar = Pd.Series(a, index = ['x', 'y', 'z'])
print(myvar)
```

output :

x	1
y	7
z	2

→ To access the value of dataframe

```
print(myvar.loc[[0]])
```

	cars	passing
# 0	BMW	3
# 1	Volvo	7

```
[[0,1]]
```

→ Loading data using pandas

```
import pandas as Pd
df = Pd.read_csv('data.csv')
df.head(2)
```

file name

Teacher Signature ..... Parents Signature .....

→ df.info() ⇒ give information related to columns
→ df ⇒ All the data set
→ df['Name'] ⇒ give 'Name Column' all entries

- Saving data with pandas

```
import pandas as Pd
data = {'calories': [420,380,390], 'duration': [50,40,45]}
print(data)
```

Step 1: Make the raw data in the form of dictionary.
Step 2: Convert the raw data into data frame

```
# load data into a data frame object.
df = Pd.DataFrame(data)
print(df)
```

Step 3: `# Saving the dataframe`
`df.to_CSV('file1.CSV')`

Numpy ⇒ It is a python library (Numerical Python)
It is invented by Travis Oliphant in 2005
It is helpful to perform numerical Computation as well as helpful to work with array

Teacher Signature ..... Parents Signature .....

i = integer b = boolean
c = complex
f = float

Date / /
Page

arr = [1, 2, 3] (Same data type)

Installation
pip install numpy

# import numpy
import numpy as np.

Working with numpy arrays:
1) creation of 1D array:

```
import numpy as np
l1 = [1,2,3,4,5]
arr = np.array(l1)          # [1 2 3 4 5]
print(arr)                    int 64
print(arr.dtype)
```

l1 = [1,2,3,4,5.6] # [1. 2. 3. 4. 5.6]
float 64

l1 = [1,'h',3,4] # ['1', 'h', '3', '4']
<U21

# Convert
arr = np.array(l1, dtype = 'f')
# [1. 2. 3. 4. 5.]
float 32

Teacher Signature ..... Parent Signature .....

STAR SHINE
Date / /
Page:

Creation of 2D array. [Matrix Row, Column]

```
l1 = [[1,2,3],[4,5,6],[7,8,9]]
arr = np.array(l1)
print(arr)
```

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

Arrange () fxn 1-D array.

```
import numpy as np
arr = np.arange(1,8)
print(arr)
```

[1 2 3 4 5 6 7]

Arrange with 2-D array.

```
import numpy as np
arr = np.arange(1,9).reshape((4,2))
print(arr)
```

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

Creation of 1D zeros array

```
import numpy as np
arr = np.zeros(4)
print(arr)
```

[0. 0. 0. 0.]

→ arr = np.ones(4) [1. 1. 1. 1.]

2D array

```
import numpy as np
arr = np.zeros((4,2))
print(arr)
```

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

Teacher Signature .....

Parents Signature .....

→ whether the array is 1D or 2D

(i) ndim [tell the dimension] # 2

```
import numpy as np
arr = np.zero ((4,2))
print (arr.ndim)
```

ii Shape (no. of rows & columns)

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

iii Size (total no. of elements)

```
print (arr.size)                # 6
```

iv dtype

```
print (arr.dtype)               # int64
```

v Sort

```
import numpy as np              # [0 2 3 9]
arr = np.array ([9,0,3,2])
print (np.sort (arr))
```

→ 2D array Sort)

```
arr = np.array ([[9,0,3,],[4,7,2,9]])
                # [[0 3 9]
                  [2 4 7 9]]
```

Teacher Signature ..... Parents Signature .....

Date / /
Page:

Numpy array indexing (Access)

```
import numpy as np                    #2
arr = np.array([4,7,2,9])
print(arr[2])
```

→ 2D array

```
import numpy as np                    #7
arr = np.array([[4,7,2,9],[1,2,3,4]])
print(arr[0,1])
```

→ Negative Indexing

```
print(arr[-1,-2])                     #3
```

Numpy array slicing:

```
arr = np.array([1,2,3,4])             # [2,3]
print(arr[1:3])
```

→ Slicing of 2D array

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

Print the element at 1st index of 2D array

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

(0:2 → index of array; 2:3 → index of elements of array)

Teacher Signature ..... Parents Signature .....

Date / /
Page

* Searching Arrays [Return Index]

```
arr1 = np.array([1,2,3,4,4])
x = np.where(arr1==4)          # (array([3,4]),)
print(x)
```

* Joining Arrays

```
arr2 = np.array([5,6])
x = np.concatenate((arr1, arr2))
print(x)                       # [1 2 3 4 4 5 6]
```

* Converting array into other data-type

```
arr = np.array([1.1, 2.2, 3.4])
newarr = arr.astype('i')
print(newarr)                  # [1 2 3]
print(newarr.dtype)            # int32
```

* Numpy Basic Operations :

- ```
  arr = np.arr([1,2,3])
  print(arr+1)                 # [2 3 4]
  ```

- ```
  arr = np.arr([1,2,3])
  print(arr-1)                 # [0 1 2]
  print(arr*10)                # [10 20 30]
  print(arr**2)                # [1 4 9]
  ```

Teacher Signature ..... Parents Signature .....

Date / /
Page

* Transpose

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

⇨ Numpy Unary Operations

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

* Max from row

```
print(arr.max(axis=1))   # [3 6]
```

* Max from column

```
print(arr.max(axis=0))   # [4 5 6]
```

* Sum of all elements in array

```
print(arr.sum())      # 21
```

* Cummulative Sum row wise

```
arr = np.array([[1,2,3],[4,5,6]])
print(arr.cumsum(axis=1))   # [[1 3 6]
                            #  [4 9 15]]
```

* Column wise

```
print(arr.cumsum(axis=0))   # [[1 2 3]
                            #  [5 7 9]]
```

Teacher Signature ..... Parents Signature .....

Date / /
Page

⇨ Numpy Binary Operations

```
arr1 = np.array([[1,3],[3,4]])
arr2 = np.array([[7,3],[2,1]])
print(arr1 + arr2)        # [[8 6]
                          #  [5 5]]
```

* Multiplication with element

```
print(arr1 * arr2)        [[7 9]
                           [6 4]]
```

* Matrix Multiplication

```
print(arr1.dot(arr2))     [[13 6]
                           [29 13]]
```

⇒ Append in numpy

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

Teacher Signature ..... Parents Signature .....