Welcome Everyone!

We will wait for others to join in!

KNOW ABOUT ME:

We Will start in 10

6

Python Basics

Lecture 6

With, Aaryan Kapur

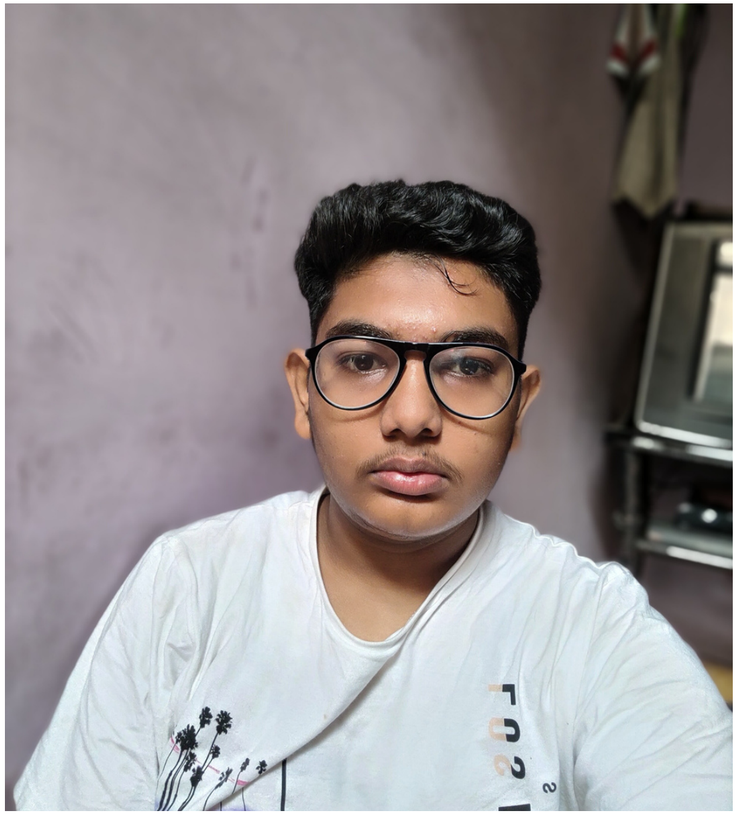
Discussion Points

Overview

File Handling in Python
OS in Python
PIP in Python
Pandas in Python
Numpy in Python
MatplotLib in Python

3

Top Performer



Akshat Tiwari

Week 5

04

File Handling

File Handling in Python

File handling is an important part of any web application.

Python has several functions for creating, reading, updating, and deleting files.

File Methods in Python

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

File Handling Methods in Python

"t" - Text - Default value. Text mode

"b" - Binary - Binary mode (e.g. images)

File methods

01 Read Only

```
f = open("demo.txt")
f = open("demo.txt", "r")
```

02 Write Only

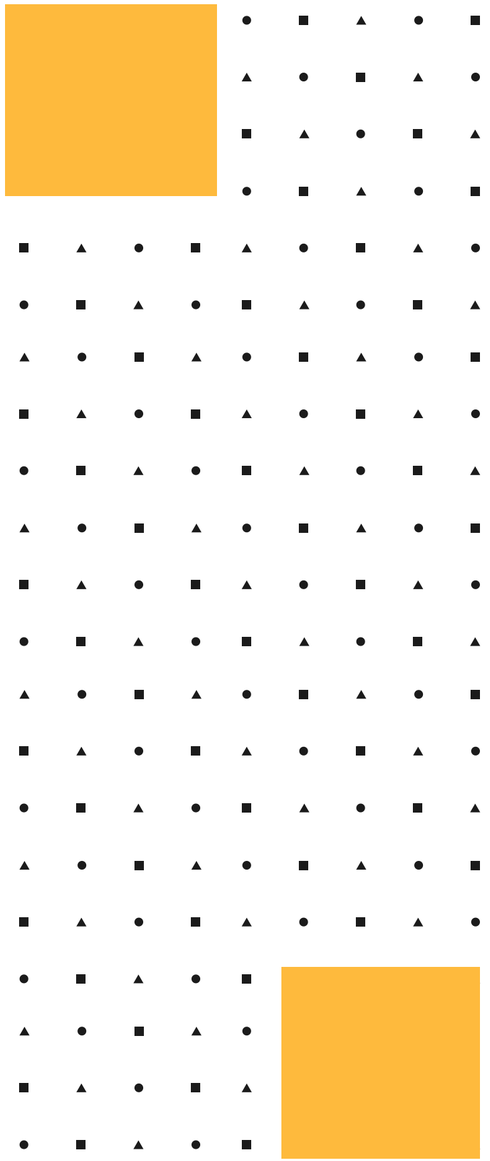
```
f = open("demo.txt", "w")
```

03 Append Only

```
f = open("demo.txt", "a")
```

04 Create File

```
f = open("demo.txt", "x")
```



05 Read and Write File

```
f = open("demo.txt", "r+")
```

06 Closing file Only

```
f.close()
```

07 Write to files

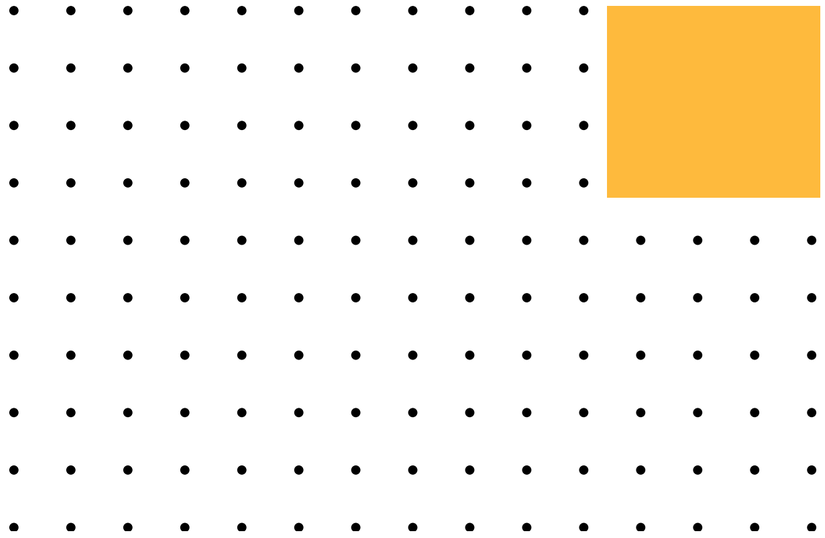
```
f.write("Hello")
```

```
ff = open("demo.txt","r+")
L = ["Hello","bye","Go"]
f.writelines(L)
```

08 Reading from our file

```
f.read()
f.readline()
f.readlines()
```

Ways to open files



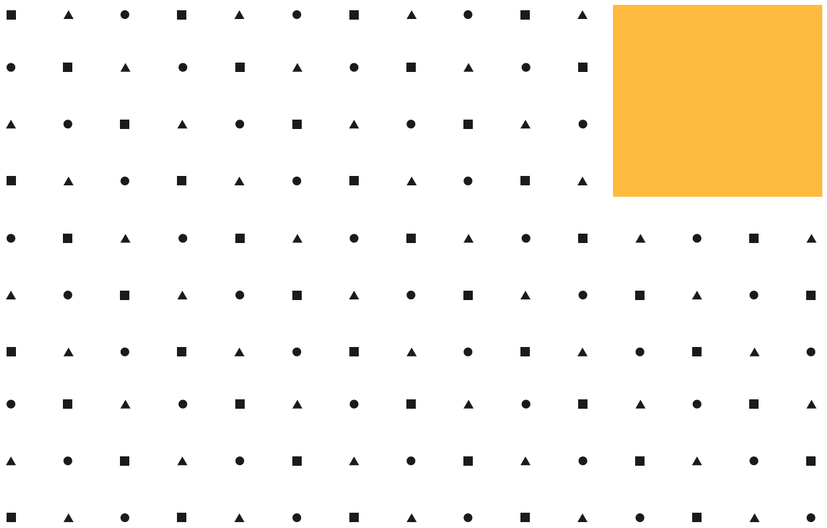
```
with open("file.txt", "w") as f:
    f.write("Hello World!!!")
```

Same as

```
f = open("demo.txt", "w")
f.write("Hello World")
```

OS in Python

import os



01 print(os.listdir())

list the folders/files in the current directory

02 os.mkdir('new')

create a new directory

03 os.rmdir('new')

delete directory

04 os.rename('demo.txt','New.txt')

rename file

05 os.remove('new.txt')

remove file

06 os.path.exists("new.txt")

Tells us if file exists

07 os.path.getsize("new.txt")

find size of file

08

PIP

PIP is a package manager for Python packages, or modules if you like.

To run in terminal:

pip ....

To run terminal commands in Notebook:

!pip ....

pip

Find pip version

pip --version

Installing a package

pip install <PACKAGE>

Find list of installed packages

pip list

Install special version of package

pip install <PACKAGE>==<VERSION>

Install from requirements.txt

pip install -r requirements.txt

Install pip

https://pypi.org/project/pip/

Uninstalling a package

pip uninstall <PACKAGE>

show package details

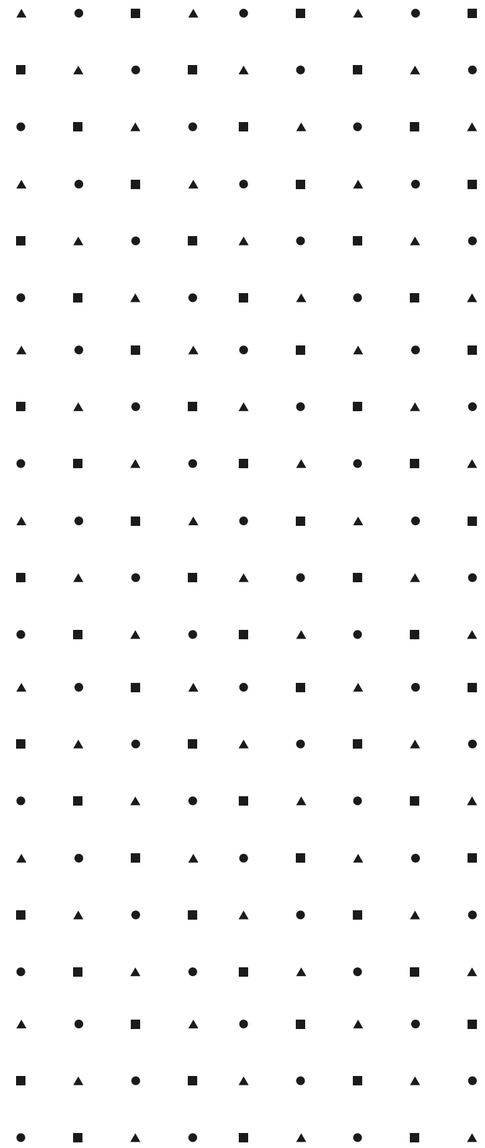
pip show <PACKAGE>

Save all installed packages

pip freeze > requirements.txt

Search packages

pip search <PACKAGE>



Pandas in Python

Pandas

11

pip install pandas
import pandas

pip install pandas
import pandas as pd

12

Pandas Series

A Pandas Series is like a column in a table.
It is a one-dimensional array holding data of any type.

Pandas series from dictionary

```
calories = {"day1": 420, "day2": 380, "day3": 390}
myvar = pd.Series(calories)
```

Pandas series with index

```
a = [1, 7, 2]
myvar = pd.Series(a, index = ["x", "y", "z"])
print(myvar)
```

Pandas series with index

```
a = [1, 7, 2]
myvar = pd.Series(a, index = ["x", "y", "z"])
print(myvar)
```

Check Pandas version

```
print(pd.__version__)
```

Pandas series

```
a = [1, 7, 2]
myvar = pd.Series(a)
print(myvar)
```

Return Value in series

```
print(mySeries[3])
```

Pandas Dataframes

13

use pandas to create DF with data

```
import pandas as pd
mydataset = {
  'para1': ["a", "b", "c"],
  'para2': [3, 7, 2]
}
myvar = pd.DataFrame(mydataset)
print(myvar)
```

Locate row in DF by index

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

Locate row in DF by row name

```
print(df.loc["a"])
```

print DF

```
print(DF)
```

print complete DF

```
print(DF.to_string())
```

Use to_string() to print the entire DataFrame.

14

Load CSV,JSON etc.

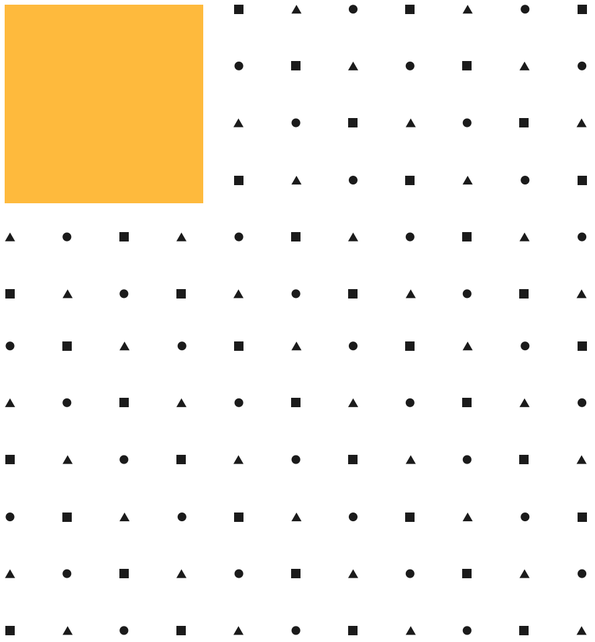
Load CSV in DF

```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.to_string())
```

Load JSON in DF

```
import pandas as pd
df = pd.read_json('data.json')
print(df.to_string())
```

Analysing data



View Dataframes

Print top n rows in DF
{by default n is 5}

print(df.head(n))

Print bottom n rows in DF
{by default n is 5}

print(df.tail(n))

Print info about DF

print(df.info())

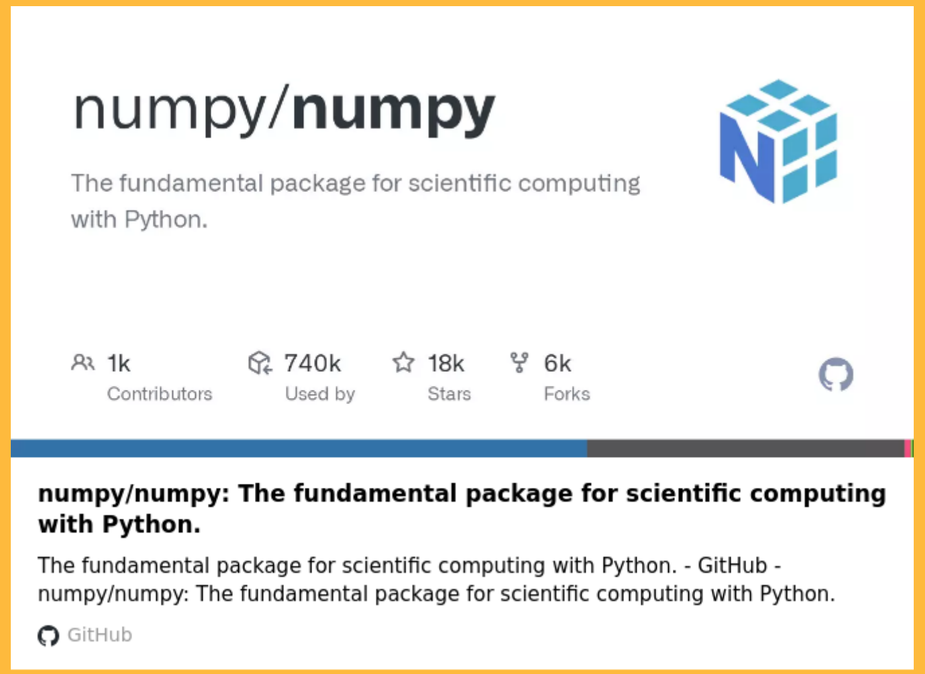
Numpy in Python

Numpy

18

pip install numpy
import numpy

pip install pandas
import numpy as np



19

Numpy Arrays

NumPy is used to work with arrays.
The array object in NumPy is called ndarray

Create ndarray

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

Print Number of dimensions in Array

```
print(a.ndim)
```

Make copy of ndarray

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

Make copy of ndarray

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

0-D array

```
arr = np.array(42)
print(arr)
```

1-D array

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

2-D Array [Matrices]

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

3-D Array [Tensors]

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

Numpy(cont.)

20

Find shape of ndarray

```
print(arr.shape)
```

Joining Arrays

```
arr1 = np.array([1, 2, 3])

arr2 = np.array([4, 5, 6])

arr = np.concatenate((arr1, arr2))
```

Find in arr

```
x = np.where(arr == 4)
```

reshape Array

```
newarr = arr.reshape(4, 3)
```

Split array

```
newarr = np.array_split(arr, 3)
```

Find in arr

```
x = np.where(arr == 4)
```

Sort Array

```
arr = np.array([3, 2, 0, 1])
print(np.sort(arr))
```

Matplotlib in Python

Matplotlib

23

pip install matplotlib
import matplotlib

24

Matplotlib pyplot

Used to plot / draw!

import

```
import matplotlib.pyplot as plt
```

Draw a line from 0,0 to 10,10

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([0, 10])
ypoints = np.array([0, 10])

plt.plot(xpoints, ypoints)
plt.show()
```

Draw points of line

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints, 'o')
plt.show()
```

25

Matplotlib pyplot

Used to plot / draw!

Draw a scatter plot

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)
plt.show()
```

Draw Bar chart

```
import matplotlib.pyplot as plt
import numpy as np

x =
np.array([5,7,8,7,2,17,2,9,4,11,1
2,9,6])
y =
np.array([99,86,87,88,111,86,10
3,87,94,78,77,85,86])
```

Draw Horizontal Bar chart

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.barh(x, y)
plt.show()
```

26

Matplotlib pyplot

Used to plot / draw!

Draw Histogram

```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)

plt.hist(x)
plt.show()
```

Draw pie chart

```
import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```

Draw pie chart with labels

```
import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]

plt.pie(y, labels = mylabels)
plt.show()
```

LET'S CODE!