# File Handling,
# Python Libraries and Modules

Distributed Network and Systems Laboratory,
Chonnam National University
Presenter : Shivani Kolekar

# *Recap*

- ✓ **Practice of what we learned last week**
- ✓ **Creating files with Python**
- ✓ **File related methods (read,write,append)**

# *Today's Goals*

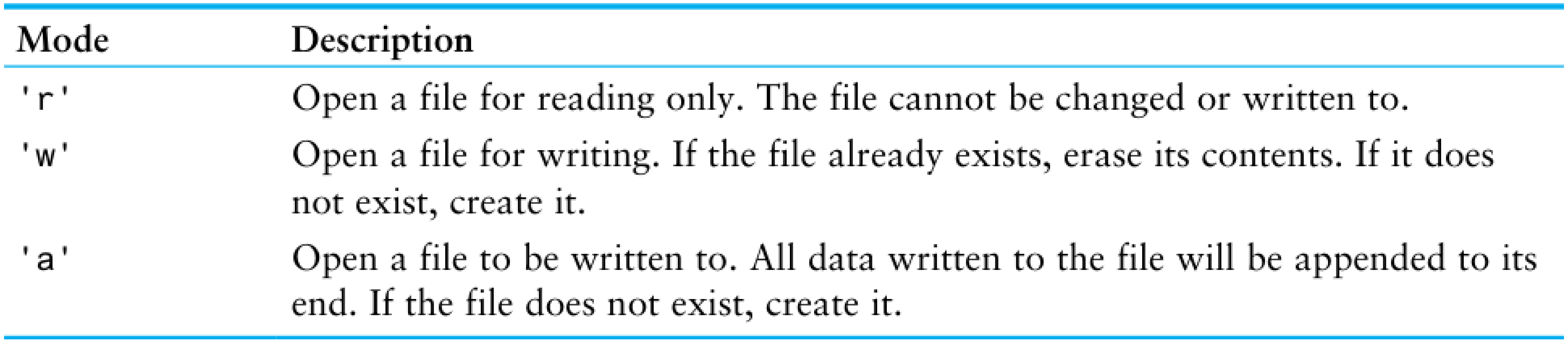
- ✓ **Introduction to Python libraries**
- ✓ **math, random etc.**
- ✓ **Installing libraries – pandas etc.**
- ✓ **Practice examples**
- ✓ **Q & A**

# *Python File Handling*

- ✓ File handling is an important part of any web application.
- ✓ Python has several functions for creating, reading, updating, and deleting files.

# *Python File Handling: open() function*

- Some of the Python File modes:



| Mode | Description |
|---|---|
| 'r' | Open a file for reading only. The file cannot be changed or written to. |
| 'w' | Open a file for writing. If the file already exists, erase its contents. If it does not exist, create it. |
| 'a' | Open a file to be written to. All data written to the file will be appended to its end. If the file does not exist, create it. |

✓ General Format:

**file_variable = open(filename, mode)**

✓ Example:

**new_file = open(my_first_file.txt, ‘r’)**

# *Python File Handling: close() function*

- Once a program is finished working with a file, it should close the file.
- Closing a file disconnects the program from the file.
- In some systems, failure to close an output file can cause a loss of data.

- This happens because the data that is written to a file is first written to a buffer, which is a small "holding section" in memory.

- **General Format:**

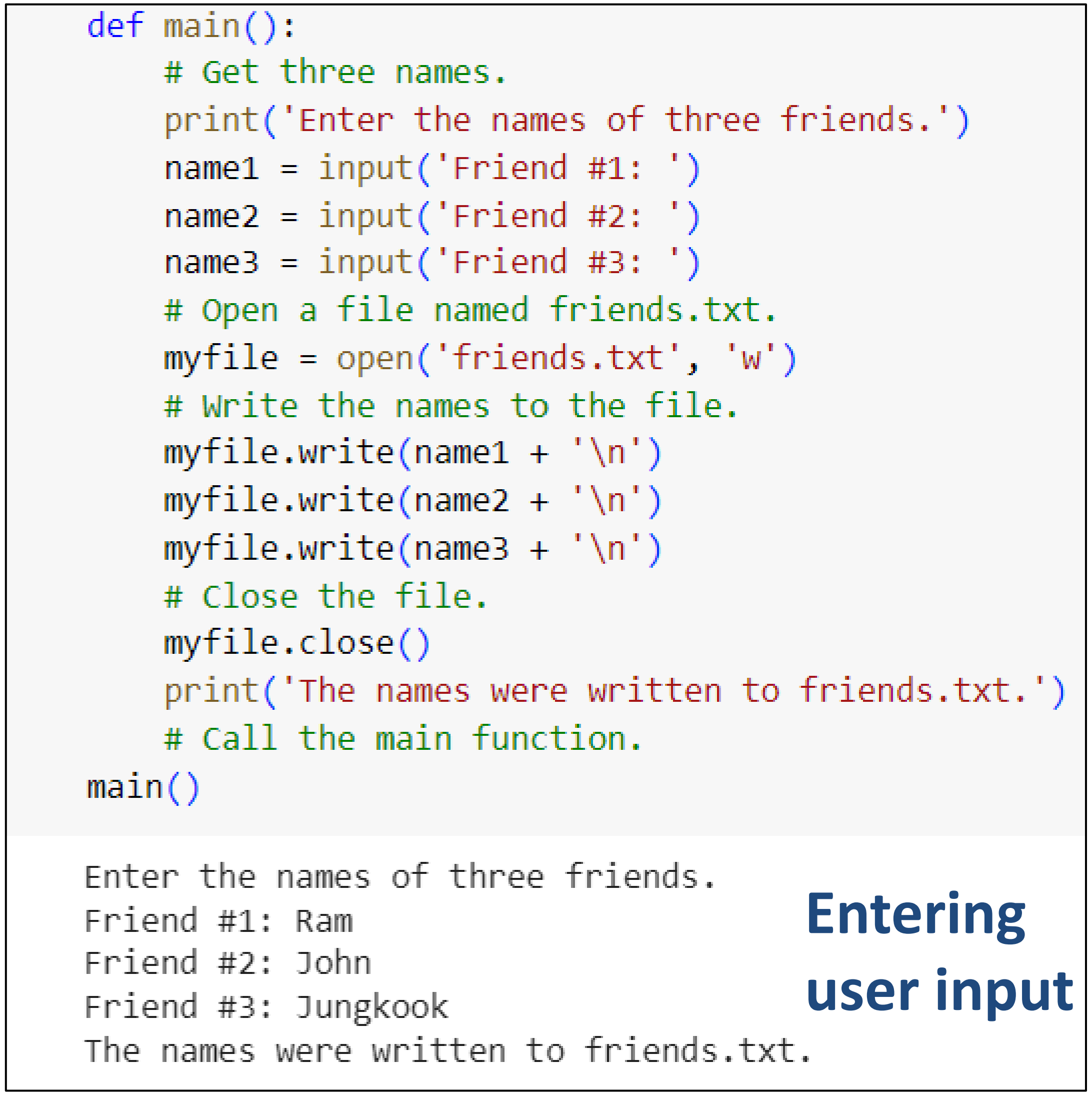
```
new_file.close()
```

# *write() method and Concatenating a newline to a string*

- When a program writes data that has been entered by the user to a file, it is usually necessary to concatenate a \n escape sequence to the data before writing it.

- This ensures that each piece of data is written to a separate line in the file.

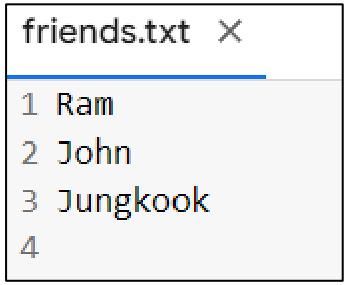
## Concatenating_newline.py



```
def main():
    # Get three names.
    print('Enter the names of three friends.')
    name1 = input('Friend #1: ')
    name2 = input('Friend #2: ')
    name3 = input('Friend #3: ')
    # Open a file named friends.txt.
    myfile = open('friends.txt', 'w')
    # Write the names to the file.
    myfile.write(name1 + '\n')
    myfile.write(name2 + '\n')
    myfile.write(name3 + '\n')
    # Close the file.
    myfile.close()
    print('The names were written to friends.txt.')
    # Call the main function.
main()
```

```
Enter the names of three friends.
Friend #1: Ram
Friend #2: John
Friend #3: Jungkook
The names were written to friends.txt.
```

**Entering user input**

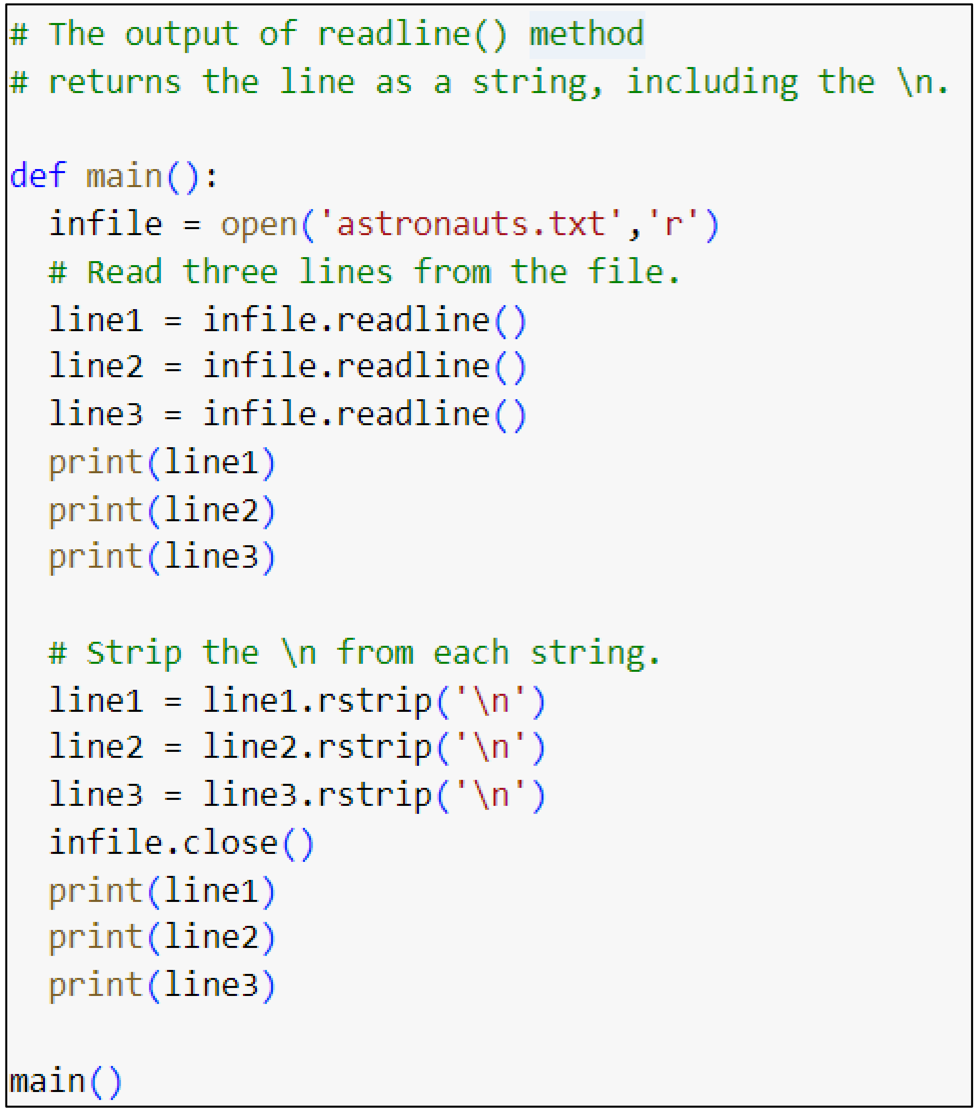


```
Ram
John
Jungkook

```

**Output_file**

# Stripping the newline from the string: rstrip() method

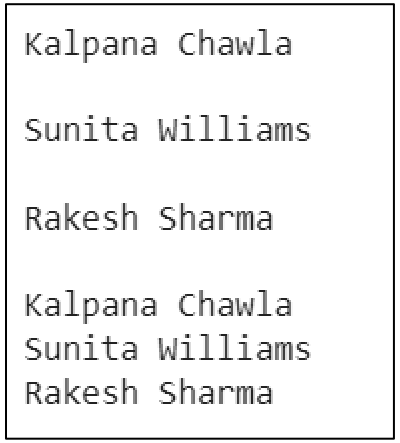
## Stripping_newline.py



```python
# The output of readline() method
# returns the line as a string, including the \n.

def main():
  infile = open('astronauts.txt','r')
  # Read three lines from the file.
  line1 = infile.readline()
  line2 = infile.readline()
  line3 = infile.readline()
  print(line1)
  print(line2)
  print(line3)

  # Strip the \n from each string.
  line1 = line1.rstrip('\n')
  line2 = line2.rstrip('\n')
  line3 = line3.rstrip('\n')
  infile.close()
  print(line1)
  print(line2)
  print(line3)

main()
```

- The \n → separates the items that are stored in the file.
- However, in many cases, you want to remove the \n from a string after it is read from a file.

## Output

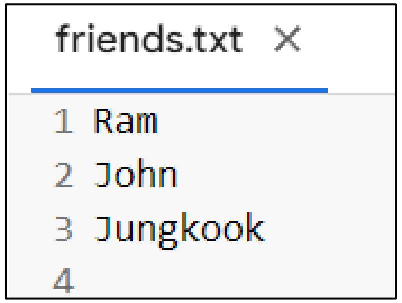


```
Kalpana Chawla

Sunita Williams

Rakesh Sharma

Kalpana Chawla
Sunita Williams
Rakesh Sharma
```

# Appending new data to an existing file

- ✓ For example, assume the file friends.txt contains the following names, each in a separate line:

Ram
John
Jungkook

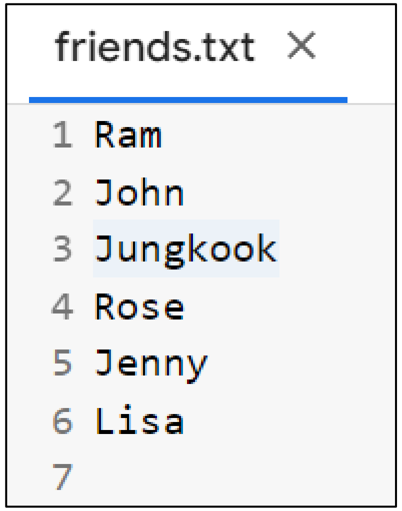


```
Ram
John
Jungkook

```

*Existing file before appending new data*

- ✓ The following code opens the file and appends additional data to its existing contents.

```
myfile = open('friends.txt', 'a’)
myfile.write(‘Rose\n’)
myfile.write(‘Jenny\n’)
myfile.write(‘Lisa\n’)
myfile.close()
```



```
Ram
John
Jungkook
Rose
Jenny
Lisa

```

*Existing file after appending new data*

# *Python File Handling: remove() function*

We need to import os for this operation:

```
import os
os.remove("demofile.txt")
```

This program will remove the existing file - demofile.txt

# Lets Code!

# *Python Libraries and modules*

- The Python standard library provides classes and functions that your programs can use to perform basic operations, as well as many advanced tasks.

- There are operations, however, that the standard library cannot perform.

- When you need to do something that is beyond the scope of the standard library, you have two choices:

1. write the code yourself, or
2. use code that someone else has already written. *(third party modules)*

Modules can be found at*: pypi. python.org*

# *Modules*

- A module is a Python source code file that contains functions and/or classes.
- Ex. – math, random
- To import a module, you write an import statement at the top of your program.
- General Format with example of math module :

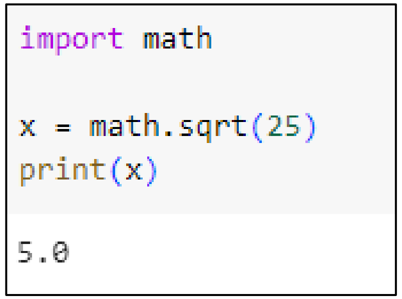
**import math**

*This statement causes the Python interpreter to load the contents of the math module into memory, making the functions and/or classes that are stored in the math module available to the program.*

# *math module*

- ✓ importing math module from Python:

## Square_root1.py



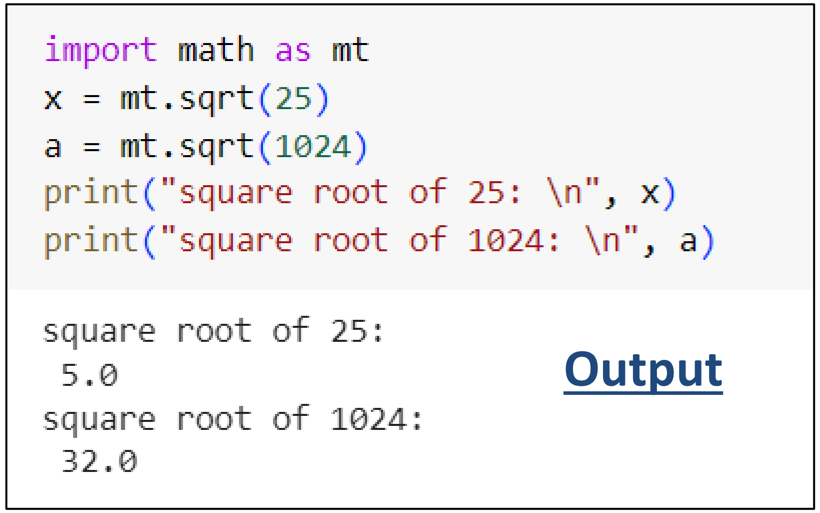
```
import math

x = math.sqrt(25)
print(x)
```

```
5.0
```

Output

- ✓ Using alias for importing math:
  **import math as mt**

## Using_alias.py



```
import math as mt
x = mt.sqrt(25)
a = mt.sqrt(1024)
print("square root of 25: \n", x)
print("square root of 1024: \n", a)
```

```
square root of 25: 
 5.0
square root of 1024: 
 32.0
```

Output

# *Random module*

- Python has a built-in module that you can use to make random numbers.

- The random module has a set of methods:

| | |
|---|---|
| randint() | Returns a random number between the given range |
| choice() | Returns a random element from the given sequence |

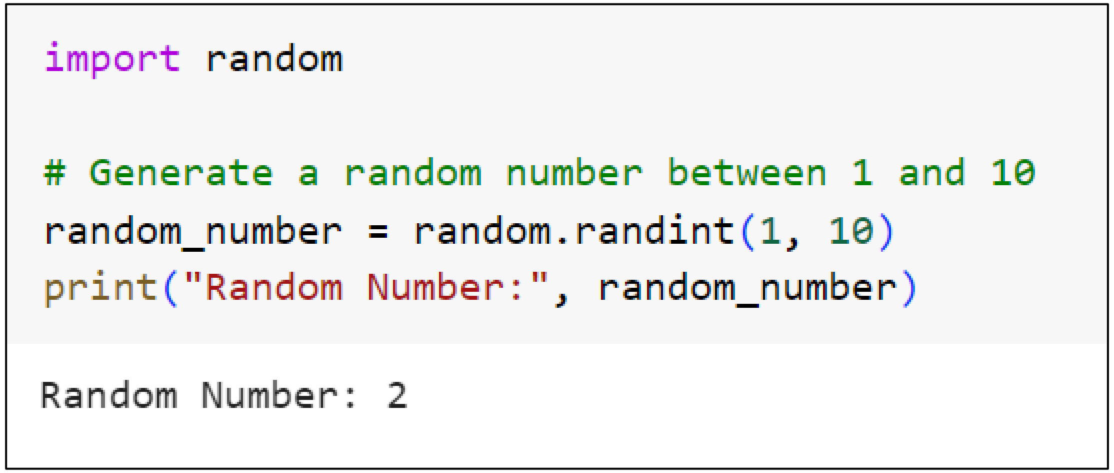
- General Format with example of math module :

  **import random**

# *Random module*

- Python has a built-in module that you can use to make random numbers.

**random_number.py**



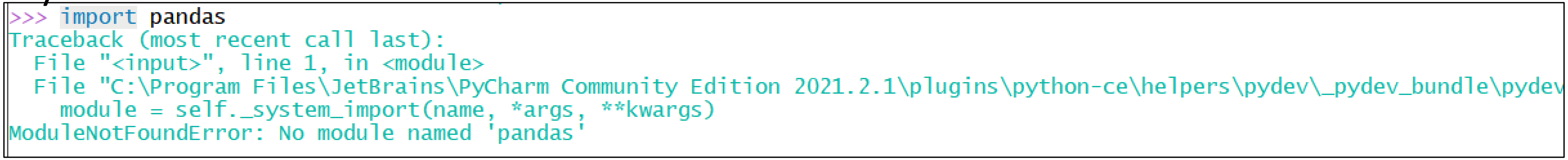
```
import random

# Generate a random number between 1 and 10
random_number = random.randint(1, 10)
print("Random Number:", random_number)
```

**Output**

```
Random Number: 2
```

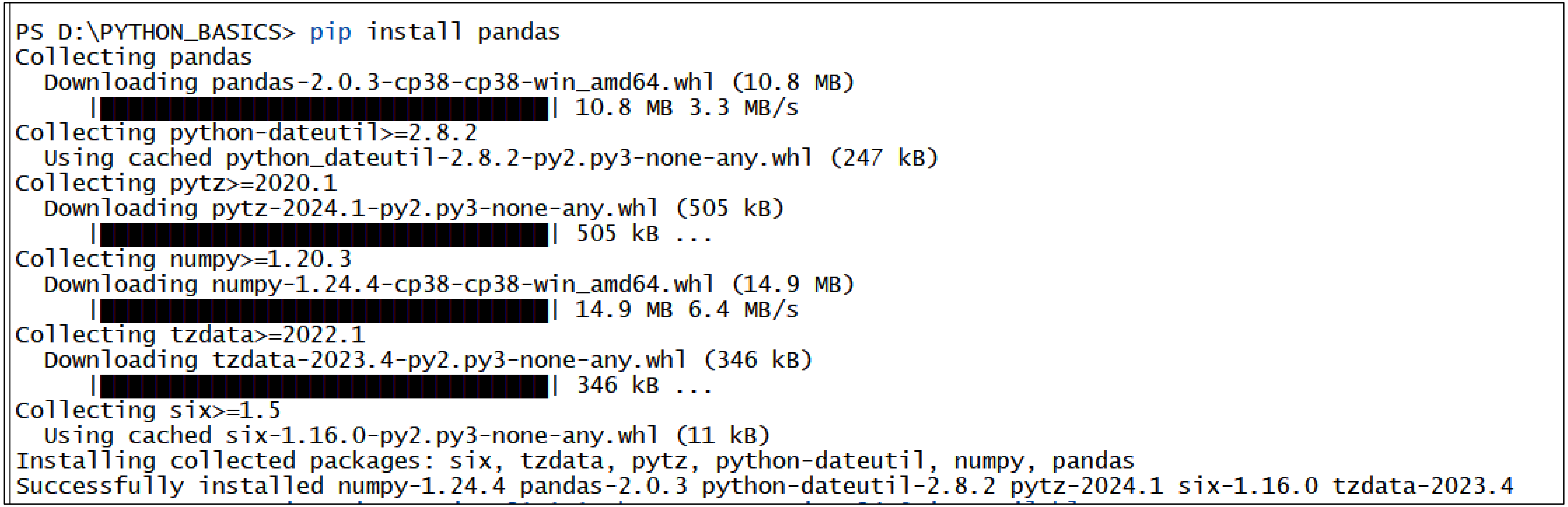
# *Installing python libraries*

Python interactive mode →



```
>>> import pandas
Traceback (most recent call last):
  File "<input>", line 1, in <module>
  File "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.1\plugins\python-ce\helpers\pydev\_pydev_bundle\pydev
    module = self._system_import(name, *args, **kwargs)
ModuleNotFoundError: No module named 'pandas'
```

Python Terminal mode →



```
PS D:\PYTHON_BASICS> pip install pandas
Collecting pandas
  Downloading pandas-2.0.3-cp38-cp38-win_amd64.whl (10.8 MB)
     |████████████████████████████████| 10.8 MB 3.3 MB/s
Collecting python-dateutil>=2.8.2
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting pytz>=2020.1
  Downloading pytz-2024.1-py2.py3-none-any.whl (505 kB)
     |████████████████████████████████| 505 kB ...
Collecting numpy>=1.20.3
  Downloading numpy-1.24.4-cp38-cp38-win_amd64.whl (14.9 MB)
     |████████████████████████████████| 14.9 MB 6.4 MB/s
Collecting tzdata>=2022.1
  Downloading tzdata-2023.4-py2.py3-none-any.whl (346 kB)
     |████████████████████████████████| 346 kB ...
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, tzdata, pytz, python-dateutil, numpy, pandas
Successfully installed numpy-1.24.4 pandas-2.0.3 python-dateutil-2.8.2 pytz-2024.1 six-1.16.0 tzdata-2023.4
```

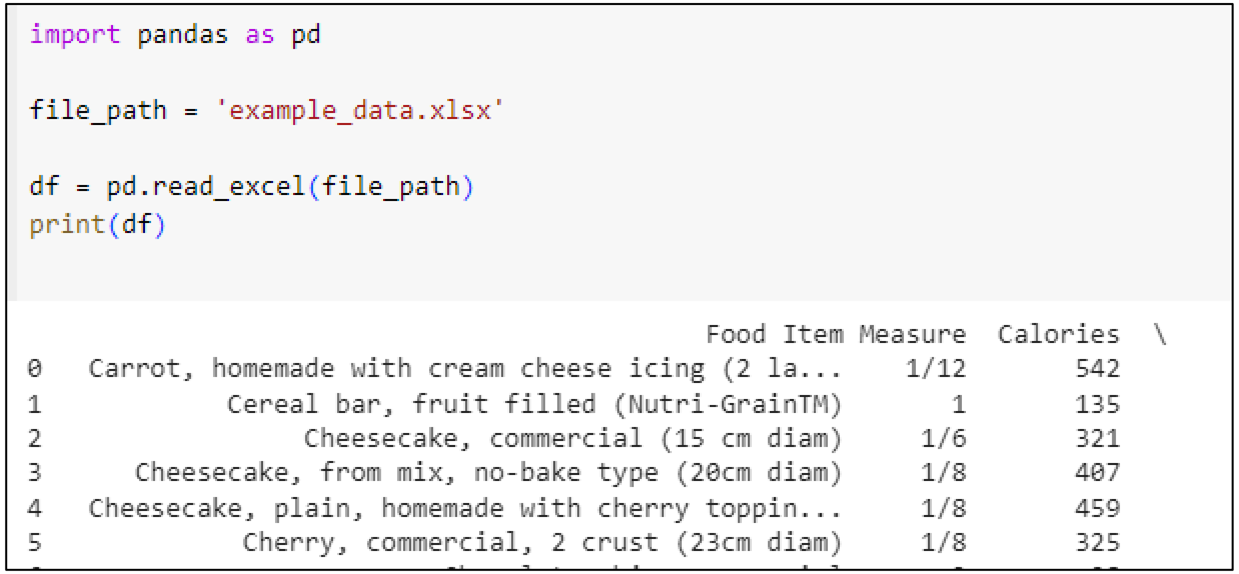
# *Using pandas*

- Pandas is a Python library used for working with data sets.

- It has functions for analyzing, cleaning, exploring, and manipulating data.

## Pandas_example.py

Output



```python
import pandas as pd

file_path = 'example_data.xlsx'

df = pd.read_excel(file_path)
print(df)
```

```
                                           Food Item Measure  Calories  \
0   Carrot, homemade with cream cheese icing (2 la...    1/12       542
1                 Cereal bar, fruit filled (Nutri-GrainTM)       1       135
2                     Cheesecake, commercial (15 cm diam)      1/6       321
3          Cheesecake, from mix, no-bake type (20cm diam)      1/8       407
4   Cheesecake, plain, homemade with cherry toppin...      1/8       459
5                  Cherry, commercial, 2 crust (23cm diam)      1/8       325
```

# Lets Code!