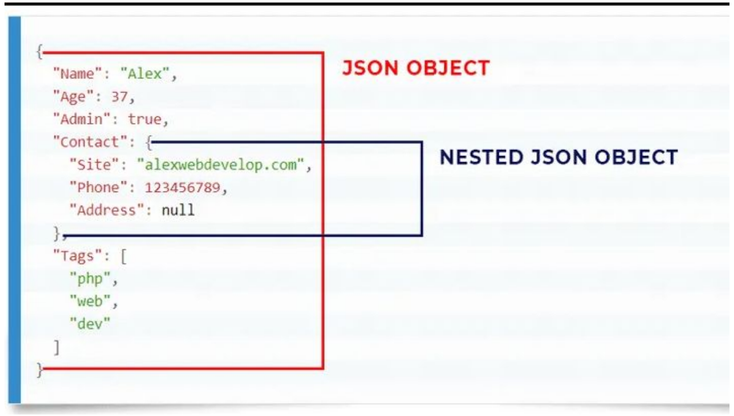
Introduction to Conda & File I/O Review

THE KNOWLEDGE HOUSE

Celebrating 10 Years of Diversifying Tech

Agenda - Schedule

1. Warm-Up
2. Software Tools
3. File I/O
4. Break
5. Conda Lab



```
{
  "Name": "Alex",
  "Age": 37,
  "Admin": true,
  "Contact": {
    "Site": "alexwebdevelop.com",
    "Phone": 123456789,
    "Address": null
  },
  "Tags": [
    "php",
    "web",
    "dev"
  ]
}
```

JavaScript Object Notation (JSON) is an open-standard data format or interchange for semi-structured data. It is text-based and readable by humans and machines. https://www.snowflake.com/guides/what-is-json

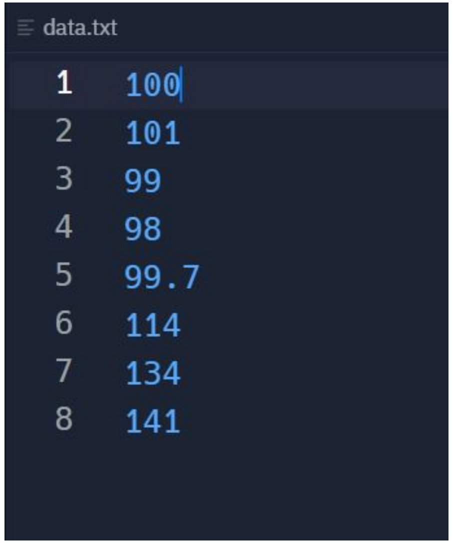
Agenda - Goals

- Ensure proper setup of all software tools for Phase 1
- Review usage of software tools
- Create conda environment
- Understand how to work with text files in your Python program
- Get familiar with opening projects in VSCode

Warm-Up

```python
def evaluate(obj) -> bool:
        accumulate = False
        for d in obj:
                val = float(d.strip())
                accumulate = accumulate or val > 140
        return accumulate

obj = open("data.txt")
print(evaluate(obj))
```


data.txt

```
100
101
99
98
99.7
114
134
141
```

Work together to figure out what will occur when we run this code.

Software Tools - VSCode, GitHub, Pip, & Conda

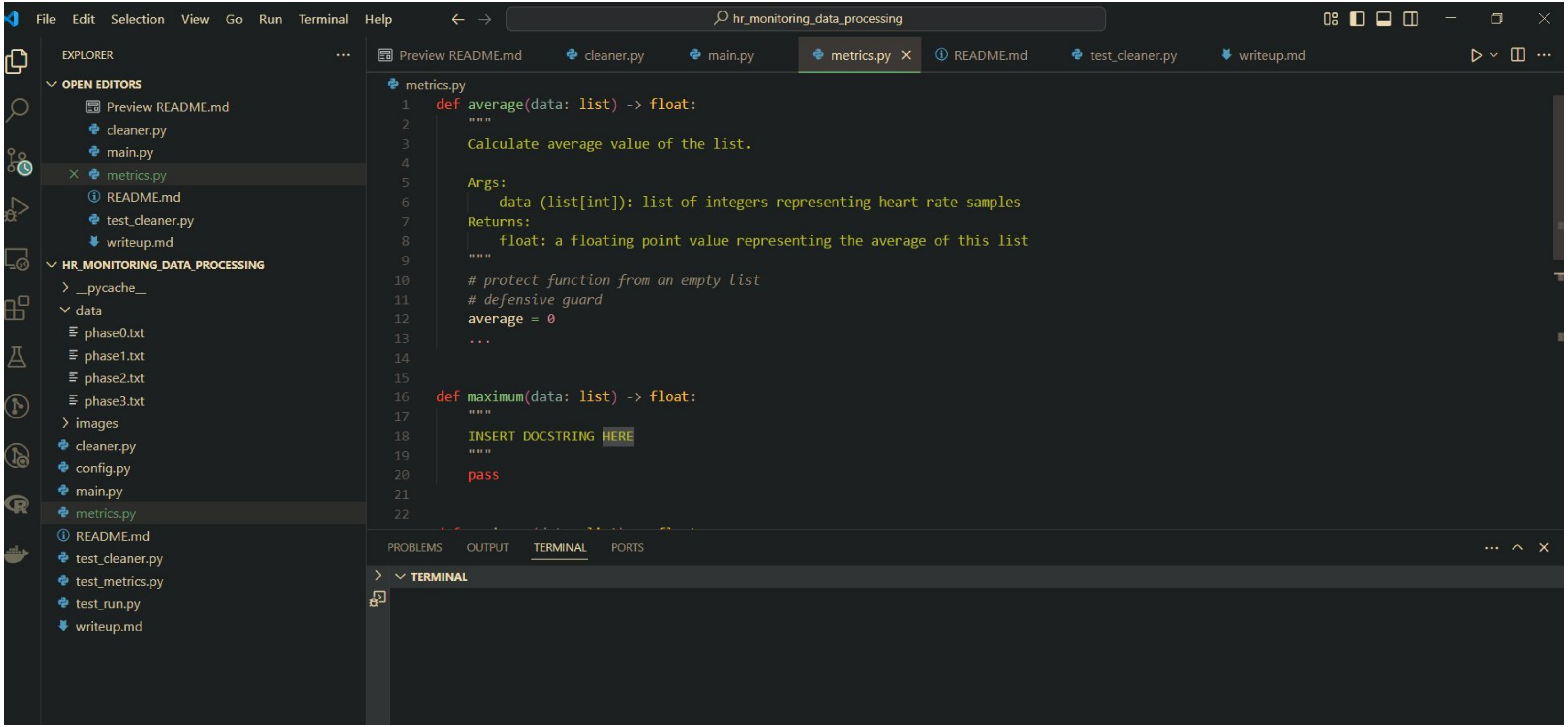
VS Code

Every programmer needs an integrated development environment (IDE) or a simple code editor.

We will be using **VSCode**.

- Great add-ons
- Easy to set up & use
- Great support

We will be demonstrating how to open a project with VSCode.



```python
def average(data: list) -> float:
    """
    Calculate average value of the list.

    Args:
        data (list[int]): list of integers representing heart rate samples
    Returns:
        float: a floating point value representing the average of this list
    """
    # protect function from an empty list
    # defensive guard
    average = 0
    ...


def maximum(data: list) -> float:
    """
    INSERT DOCSTRING HERE
    """
    pass

```

Demonstration of opening VSCode

The Terminal

All should technologists should be familiar with the process of navigating their computer using a **terminal.**

Reason's being:

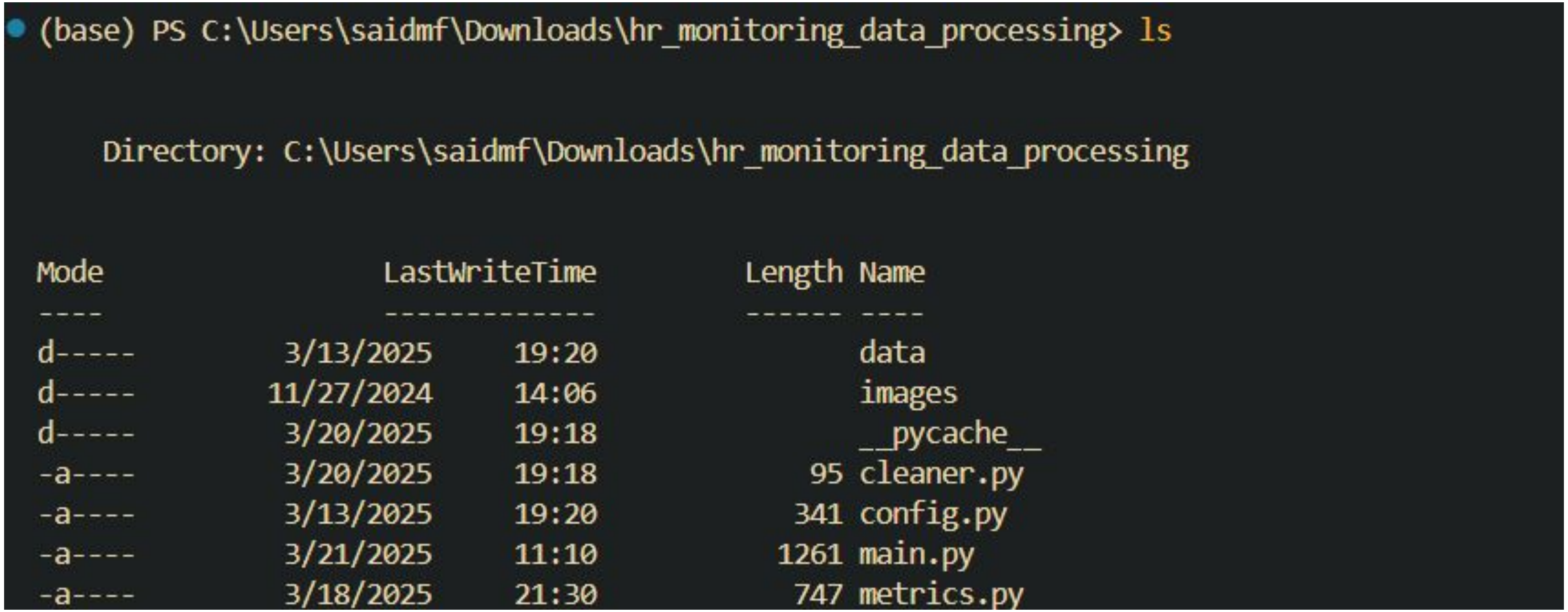
- It's accurate
- It's fast
- It's used by everyone

As we develop in this fellowship we should **shift to using the terminal**

The Terminal

Some commands you should **get used to include**:

- **ls** : list all files in current working directory
    - **dir** : if you’re on windows
- **pwd** : print working directory
    - **cd** : if you’re on windows
- **cd** [folder]: change directory to specified folder (same on windows)



```
● (base) PS C:\Users\saidmf\Downloads\hr_monitoring_data_processing> ls


    Directory: C:\Users\saidmf\Downloads\hr_monitoring_data_processing


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         3/13/2025     19:20                data
d-----        11/27/2024     14:06                images
d-----         3/20/2025     19:18                __pycache__
-a----         3/20/2025     19:18             95 cleaner.py
-a----         3/13/2025     19:20            341 config.py
-a----         3/21/2025     11:10           1261 main.py
-a----         3/18/2025     21:30            747 metrics.py
```

Demonstration of terminal interaction

Virtual Environments - Metaphor

You are a carpenter with a private business.

You get requests for numerous types of jobs:

- House framing
- Wooden furniture
- Cabinet-making

When leaving to your worksite, you **spend an hour gathering your tools** (inefficient).

Virtual Environments - Metaphor

Instead you set up **3 separate tool-boxes** with different sets of tools (depending on what kind of job).

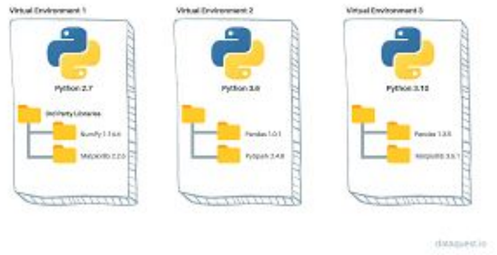
Now you can start your day by simply **grabbing the box you need** (efficient).

Virtual Environments - Real

You are a data scientist.

You get requests for numerous types of jobs:

- Exploratory Data Analysis
- Dashboard building
- Machine Learning



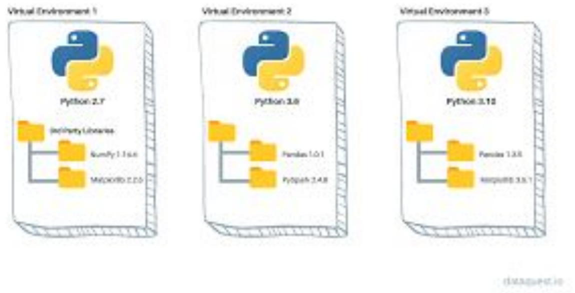
You will have sandboxes called “**virtual environments**” set up in your terminal that have **all the 3rd party packages necessary** to complete these jobs (efficient).

Virtual Environments

conda env create -f environment.yml	*: creates a venv*
conda activate ds	*: activates the venv*
conda deactivate	*: deactivates the venv*

We will not demonstrate how to work with conda environments, instead we want to challenge you to create your own environment in tonight's lab.

Pip

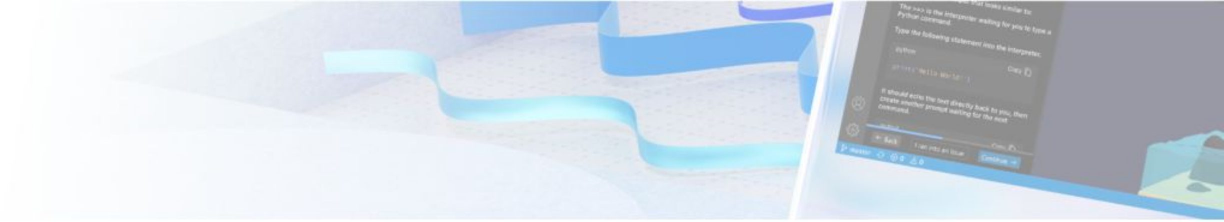


pip install <package>

Pip

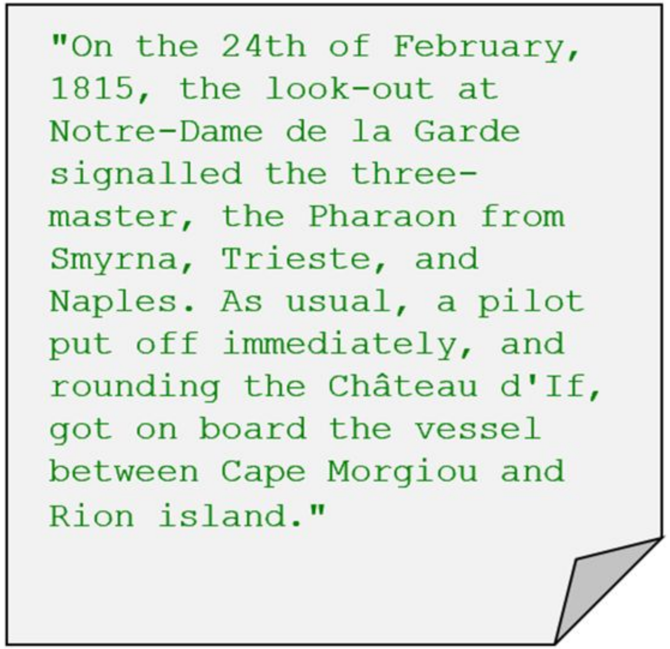
pip install pandas	: *install updated pkg*
pip install pandas==2.1.0	: *install specific pkg*
pip install -r requirements.txt	: *install all pckg’s*

File I/O Review



Files

- Files are strings of data.
- If you know the path and filename of the file, you can use Python to access it.
- You can then process the data as if it were a string or a list of strings.



```
"On the 24th of February,
1815, the look-out at
Notre-Dame de la Garde
signalled the three-
master, the Pharaon from
Smyrna, Trieste, and
Naples. As usual, a pilot
put off immediately, and
rounding the Château d'If,
got on board the vessel
between Cape Morgiou and
Rion island."
```

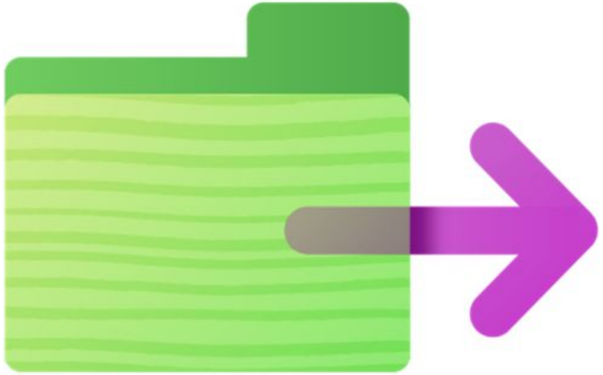
Count-of-monte-cristo.txt

Files

- The `open` function consumes the path to the file as a string and returns a `File` object.
- Typically, you store this `File` object in a variable.
- **Example**: Until you tell Python to read data from the file, the only information you have is that the file is open and ready.
- The `File` object isn't the same thing as the data inside the file.

```
book_path = "Count-of-monte-cristo.txt"
book_file = open(book_path)

# Boring!
print(book_file)
```

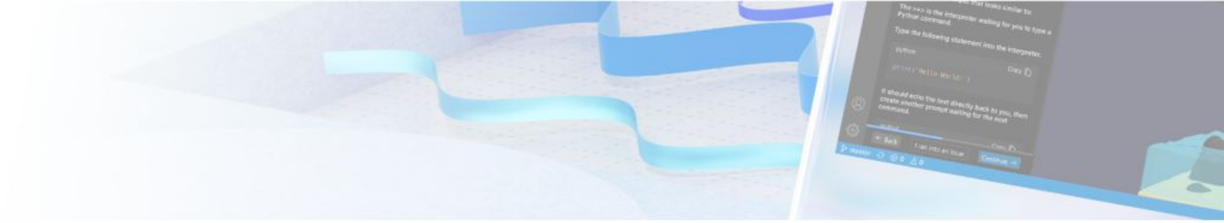


Reading Characters

```
book_path = "Count-of-monte-cristo.txt"
book_file = open(book_path)

# Use the read() method to get the file as a string
book_text = book_file.read()
print(book_text)
```

- Get data from a `File` object by using the `.read()` method, which returns the file contents as a string.
- **Example**: open the file, read the `File` object, and then print the file's text.
- This is a multi-step process:
  1. Use the path to open the file.
  2. Read from that open file.

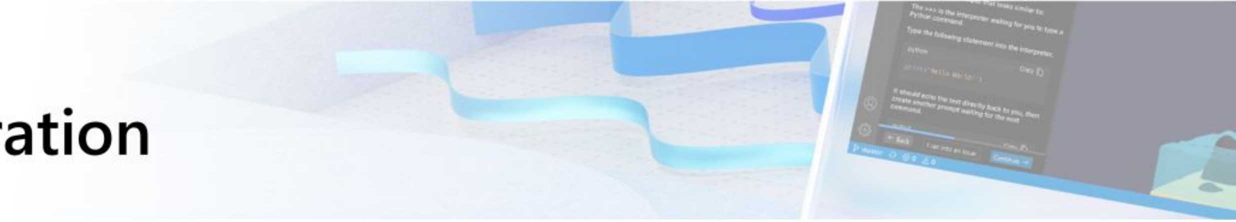


Reading Characters

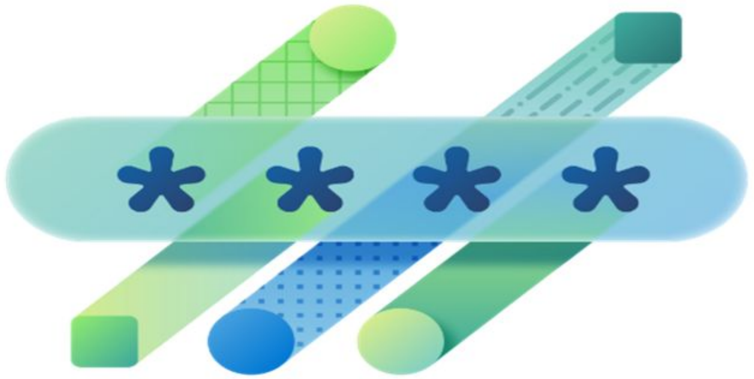
- With the string loaded from the file, you can process the file character by character.
- **Example**: Open the file again and count the number of characters by using the loop pattern.

```python
book_path = "Count-of-monte-cristo.txt"
book_file = open(book_path)

# Use the read() method to get the file as a string
book_text = book_file.read()
count = 0
for character in book_text:
    count += 1
print(count)
```



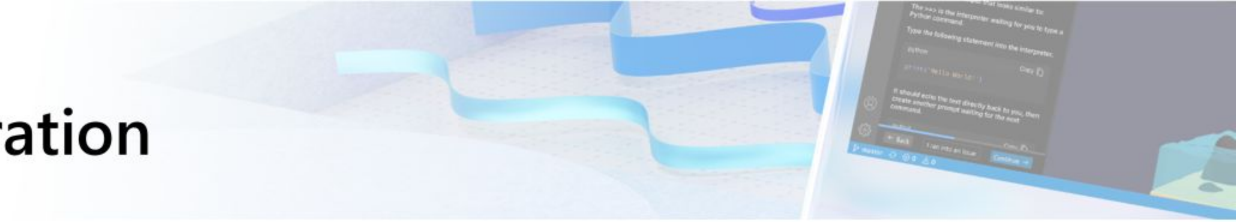
Line-by-Line File Iteration



```
book_path = "Count-of-monte-cristo.txt"
book_file = open(book_path)

for line in book_file:
    print(line)
```

- Because a `File` is a sequence of strings (each separated by a new line), you can process it by using a `for` loop.
- **Example**: Process the file line by line.
- Using the `for` loop, you no longer need to use the `.read()` method.
- Combining the `read` and `for` loop results in string iteration.

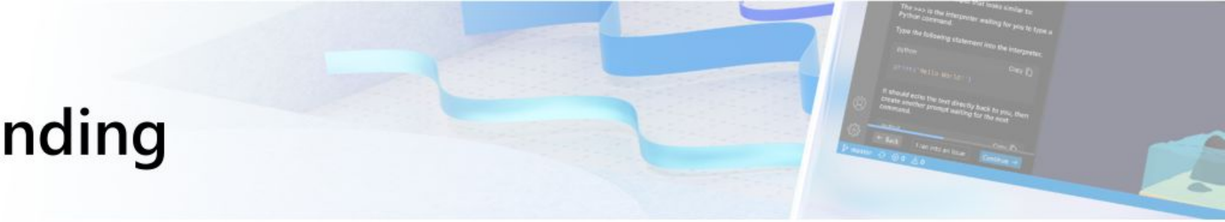


Line-by-Line File Iteration

- You can break up lines of a file by using new line characters (usually `\n`).
- When a file is read line by line, the new line characters are included in each line.
- Use the `strip` method to remove the extra whitespace from the end of the line.

```python
book_path = "Count-of-monte-cristo.txt"
book_file = open(book_path)

for line in book_file:
    print(line)
```



Check Your Understanding

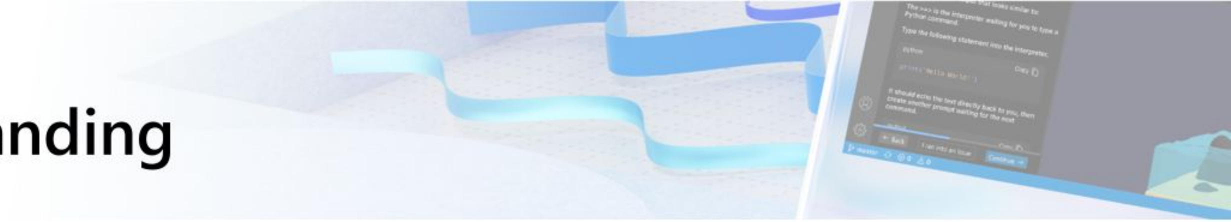
Question 1

The `open` function consumes a string representing a path.
What does the `open` function return?

❑ A string representing the file contents.

❑ A list of strings representing the file contents.

❑ A `File` object that you can use to access the file contents.



Check Your Understanding

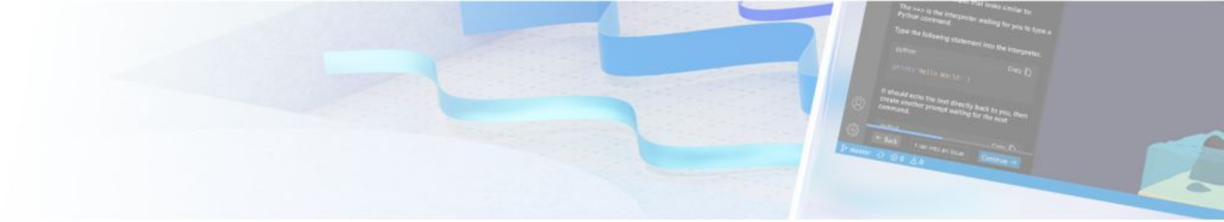
Question 1

The `open` function consumes a string representing a path. What does the `open` function return?

❑ A string representing the file contents.

❑ A list of strings representing the file contents.

✓ A `File` object that you can use to access the file contents.
The open function returns a `File` object, which is a new type of value with special methods for accessing the actual file data.



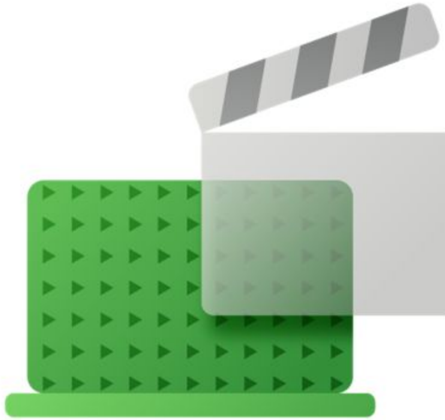
Closing Files

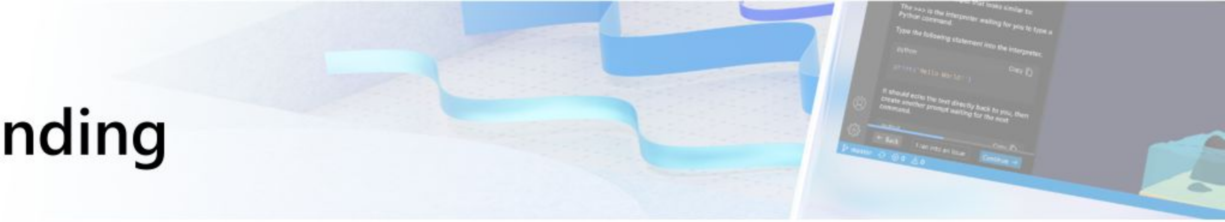
- When you're done with a file, close it by using the `close` method.
- Forgetting to close a file can leak memory resources on older devices and possibly cause data loss.
- The `close` indicates to anyone reading the program that the file-reading phase is finished.
- After a file is closed, you can't use the `read` method on the file or iterate through it with a `for` loop.

```python
book_path = "Count-of-monte-cristo.txt"
book_file = open(book_path)

print(book_file.read())

# This is critical!!!
book_file.close()
```





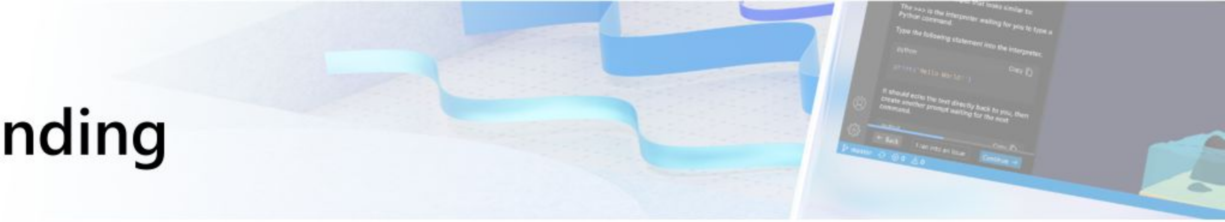
Check Your Understanding

Question 2

Given a file named `grades.txt` with the contents: `90, 85, 100` and the following Python code, what is the type of `grade_data`?

```
grade_file = open("grades.txt")
grade_data = grade_file.read()
```

- ❑ `list[int]`
- ❑ `str`
- ❑ `list[str]`



Check Your Understanding

Question 2

Given a file named `grades.txt` with the contents: `90, 85, 100` and the following Python code, what is the type of `grade_data`?

```
grade_file = open("grades.txt")
grade_data = grade_file.read()
```

❑ `list[int]`

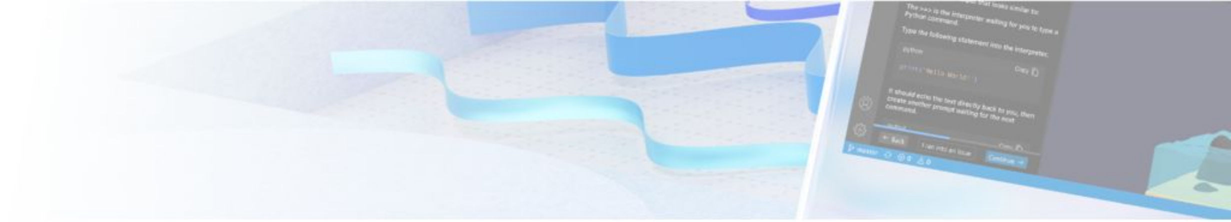
✓ `Str`
The `read` method always returns a string no matter what the contents of the file are (even if the file might look like a list or integers.)

❑ `list[str]`

File Objects

- When you call the `open` function from before, you’re given a `File` object.
- The `File` type has its own unique methods (`read`, `close`) and can be iterated by using a `for` loop.
- You can’t use operators like addition (`+`) or subscripting (square brackets like `[` with an index).

File Objects

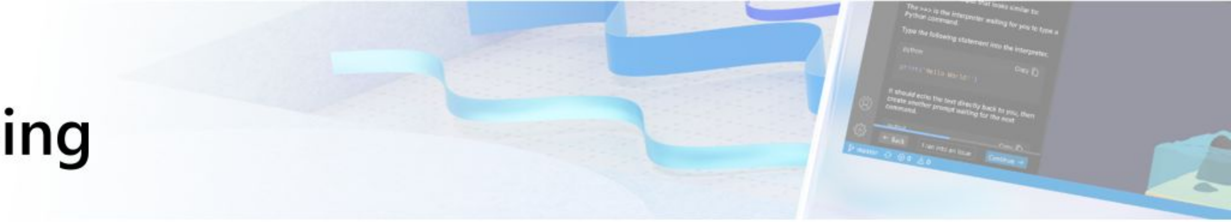


Python has many special built-in type values. For now, just remember the operations and methods for files:

- The `open` function that takes a string path and returns an open `File` object
- The `close` method of `File` objects that frees up the resource
- The `read` method of `File` objects that returns the contents of the file as a string
- The `for` loop iteration over the `File` object as a sequence of strings (separated by new lines)

FileNotFoundError

- File systems are tricky because everyone has a different setup.
- When you try to open a file that doesn't exist, Python raises a `FileNotFoundError` and suggests that the file doesn't exist.
- Ask yourself:
  - Do I have the right file name?
  - Do I have the right path?
  - Is the file where I think it is?
  - Is my program where I think it is?



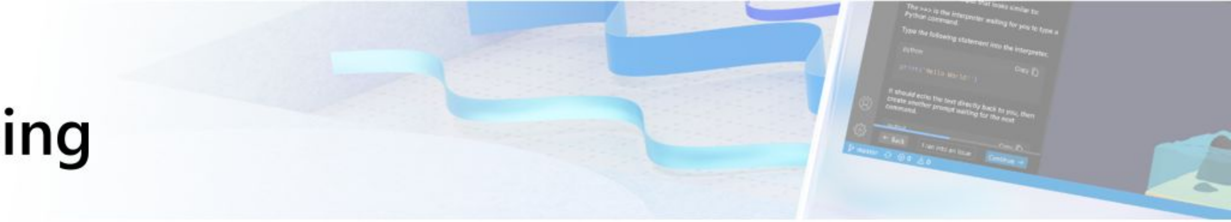
Example File Processing

```
score_sum = 0
data_file = open('scores.txt')

for line in data_file:
    score_sum = score_sum + int(line.strip())

data_file.close()
print(score_sum)
```

- **Example**: Process a list of numbers in a file. Each number represents a score.
- The code shows the sum pattern to add each of the scores together.
- Strip off the new lines at the end of each line, then convert that line to a number. When you read data from a file, it comes in as a string.



Example File Processing

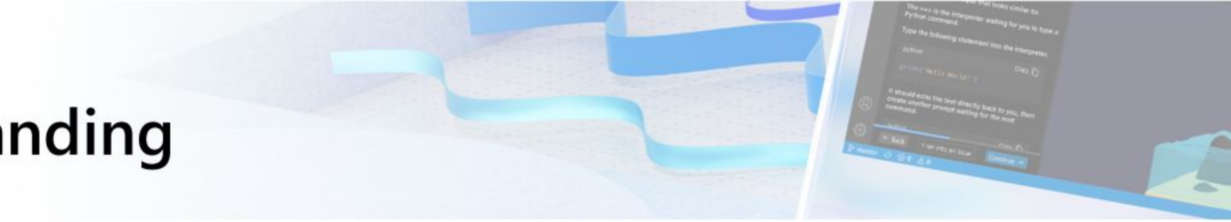
- Even if a file contains only numbers, the values returned by the `read` method and line-by-line iteration will still just be strings.
- Strings can contain numeric characters, but that doesn't make those values integers.
- Until you explicitly convert the contents by using the `int` or `float` function, you have string values.

```
score_sum = 0
data_file = open('scores.txt')

for line in data_file:
    score_sum = score_sum + int(line.strip())

data_file.close()
print(score_sum)
```

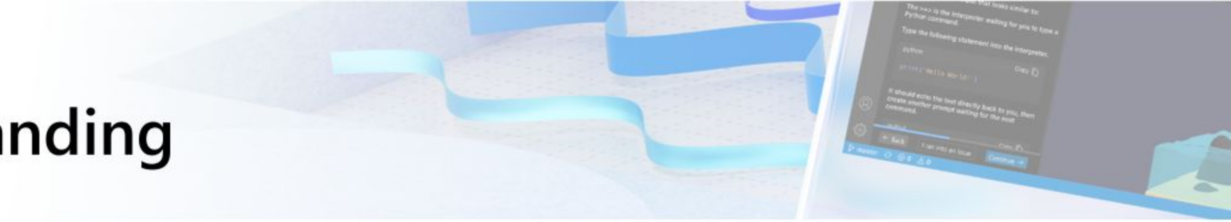
Check Your Understanding



Question 3

When you iterate through a `File` object with a `for` loop, how do you go through the file?

- ❑ Character by character
- ❑ Sentence by sentence
- ❑ Line by line



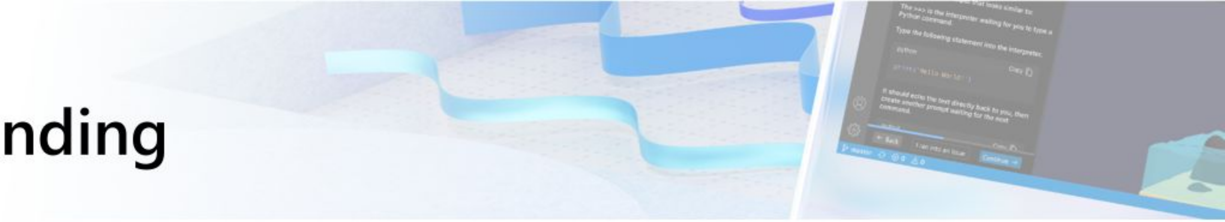
Check Your Understanding

Question 3

When you iterate through a `File` object with a `for` loop, how do you go through the file?

- ❑ Character by character
- ❑ Sentence by sentence
- ✓ Line by line

  A file is organized into lines separated by new line characters (`\n`). When you iterate through the file with a `for` loop, you get each line as a string value (including the new line.)



Check Your Understanding

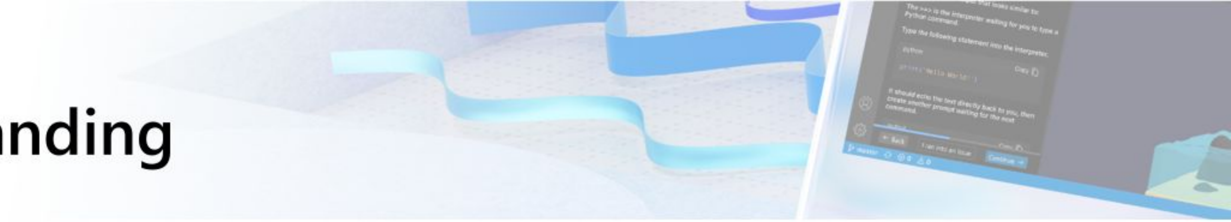
Question 4

When should you close a file?

❑ Immediately after opening the file

❑ After you finish reading the file

❑ At the very end of the program, on the last line



Check Your Understanding

Question 4

When should you close a file?

❑ Immediately after opening the file

✓ After you finish reading the file
Once a file has been read, you have no further use for it. That's the best time to close the file.

❑ At the very end of the program, on the last line

Lab

Lab - Conda Module

For the remaining lab time, break into your pod groups and complete the **Conda Installation Lab**

If you encounter an error, do not give up!

An expert is someone who has failed 1000s of more times than the beginner. **No pain no gain.**

Wrap-Up

Lab (Due 03/28)

Taipei City, Taiwan

The company you work for, Seng-Links, aims to identify periods when a user sleeps or exercises using their varying recorded heart rates.

Your company has provided you a data folder (*data/*) of **4 files** that contain heart-rate samples from a participant. The participants device records heart rate data every 5 minutes (aka *sampling rate*).

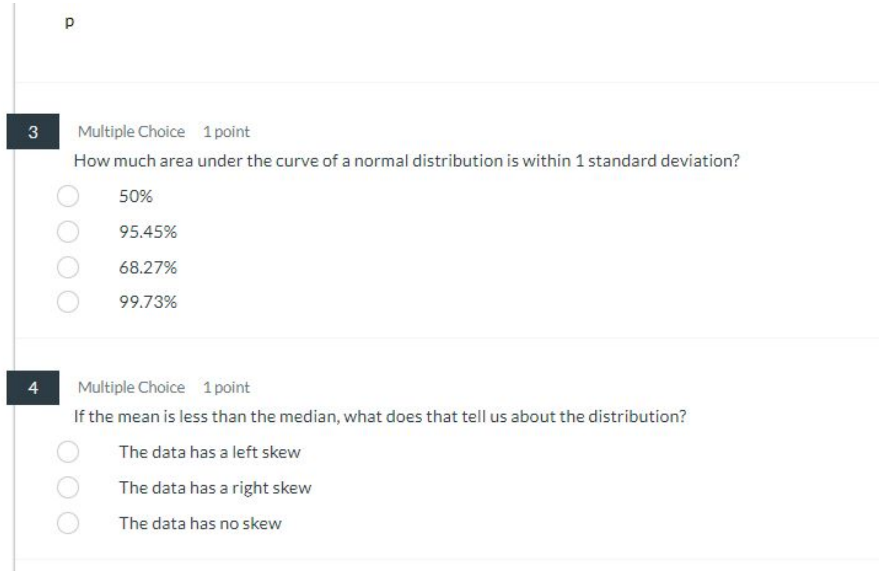
You are tasked with writing code that **processes each data file**. You will utilize test-driven development in order to complete this project.

Stats Quiz (Due 03/28)

Please complete this quiz by 03/28.

This is a 10-question quiz that will test your knowledge of statistics concepts.

2 attempts allowed.



p

3 Multiple Choice 1 point

How much area under the curve of a normal distribution is within 1 standard deviation?

- 50%
- 95.45%
- 68.27%
- 99.73%

4 Multiple Choice 1 point

If the mean is less than the median, what does that tell us about the distribution?

- The data has a left skew
- The data has a right skew
- The data has no skew

Tuesday

Tuesday will entail:

- Introduction to different data formats
- A review of JSON data.

Jupyter: scratchpad of the data scientist

If you understand what you're doing, you're not learning anything. - Anonymous