(AP)
Advanced Programming

Week 2

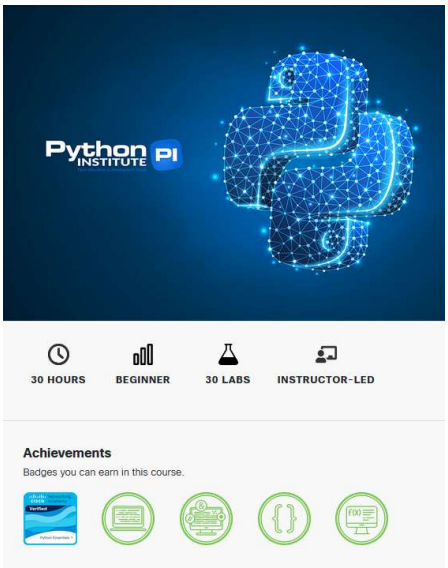
Spring 25-26

Feb. 19, 2026

Lect. Malik MALKAWI

1

Cisco #1



- https://www.netacad.com/courses/python-essentials-1?courseLang=en-US&instance_id=46a69b3a-c3d8-48eb-85ab-f00fe08f742c

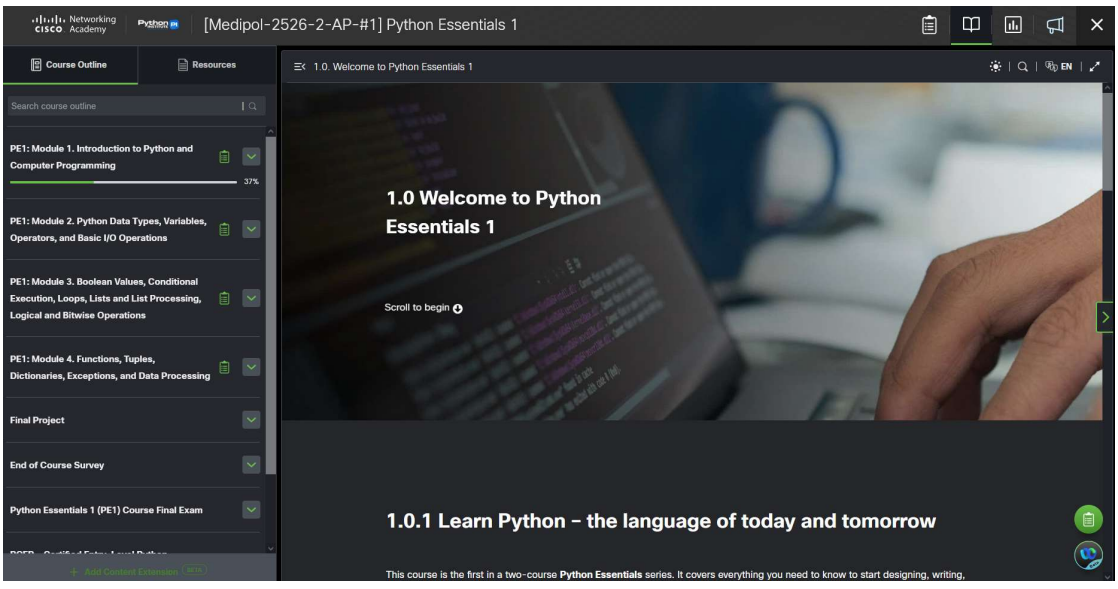
Deadline: Feb. 26, 2026

2

Cisco #1

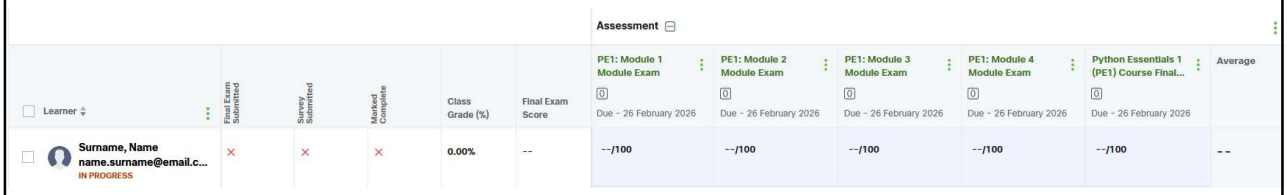
3

Cisco #1



4

Cisco #1



Learner	Final Exam Submitted	Survey Submitted	Marked Complete	Class Grade (%)	Final Exam Score	Assessment: PE1: Module 1 Module Exam	PE1: Module 2 Module Exam	PE1: Module 3 Module Exam	PE1: Module 4 Module Exam	Python Essentials 1 (PE1) Course Final...	Average
						Due - 26 February 2026	Due - 26 February 2026	Due - 26 February 2026	Due - 26 February 2026	Due - 26 February 2026	
Surname, Name name.surname@email.c... IN PROGRESS	×	×	×	0.00%	--	--/100	--/100	--/100	--/100	--/100	--

Eymen Kuru 98.9 Yağmur Önal

5

LABS

6

Menu

1 – Files *(Open and TXT)*
2 – JSON *(JavaScript Object Notation)*
3 – Excel *(Your Task :)*

Files
(Open and TXT)

Persistence

- **Transient**: a program runs for a short time and produce some output, but when it ends, its data disappears. If you run the program again, it starts with a clean state.
- **Persistent**: a program runs for a long time (or all the time); it keeps at least some of their data in permanent storage (a hard drive, for example); and if it shuts down and restarts, it picks up where it left off.

Writing

- To write to a file, you have to open it with mode 'w' as a second parameter:

```
fout = open("output.txt", "w")
```

- If the file already exists, opening it in write mode clears out the old data and starts fresh, so be careful!
- If the file doesn't exist, a new one is created.

Writing

- The write method puts data into the file

```
line1 = "Istanbul Medipol\n"
fout.write(line1)
```

- Again, the file object keeps track of where it is, so if you call write again, it adds the new data to the end.

```
line2 = "University.\n"
fout.write(line2)
```


Closing

- When you are done writing, you have to close the file.

```
fout.close()
```

Reading – Method 1.1

```
file = open('newfile.txt', 'r')
for line in file:
    print(line)
file.close()
```


Reading – Method 1.2

```
for line in open('newfile.txt', 'r'):
    print(line)
file.close()
```

Reading – Method 2

```
with open('newfile.txt', 'r') as file:
    for line in file:
        print(line)
```

15

```
for line in open('newfile.txt', 'r') :
    print(line)
file.close()
```

```
file = open('newfile.txt', 'r')
for line in file:
    print(line)
file.close()
```

```
with open('newfile.txt', 'r') as file:
    for line in file:
        print(line)
```

16

Modes

Modes	Description
r	Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer placed at the beginning of the file. Does not create the file if it does not exist.
w	Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
w+	Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
a	Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
a+	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and appending.

17

Modes and Pointer

Mode	Description	File Pointer Position	Creates File if Not Exists	Truncates Existing File
r	Read-only	Beginning of the file	No	No
r+	Read and write (updating)	Beginning of the file	No	No
w	Write-only (overwrite or create)	Beginning of the file	Yes	Yes
w+	Write and read (overwrite or create)	Beginning of the file	Yes	Yes
a	Append-only (append or create)	End of the file	Yes	No
a+	Append and read (append or create)	End of the file	Yes	No

18

Seek Function

- The seek function is a built-in function that is used to set the current position of the file pointer within a file.

Syntax

```
file.seek(offset, whence)
```

- The first argument, offset, is the number of bytes we want to move the file pointer.
- The second argument, whence, specifies the reference position from where we want to move the file pointer. The possible values of whence are
  - 0 (default): refers to the beginning of the file
  - 1: refers to the current position of the file pointer
  - 2: refers to the end of the file

Seek Function

```
file = open("data.txt", "r")
file.seek() # refers to the beginning of the file
data = file.read(5) # Read the next 5 bytes from the file
print(data)
file.close()
```

Seek Function

```
file = open("data.txt", "r")
file.seek(10, 1) #position of the pointer to 10 bytes from current position
data = file.read(5) # Read the next 5 bytes from the file
print(data)
file.close()
```


Seek Function

```
file = open("data.txt", "r")
file.seek(10) # Set the position of the file pointer to byte 10
data = file.read(5) # Read the next 5 bytes from the file
print(data)
file.close()
```

Binary Mode

- To read a binary file in Python, first, we need to open it in **binary mode**

Read Mode	Write Mode
`mode="rb"`	`mode="wb"`
`open('newfile.txt', 'rb')`	`open('newfile.txt', 'wb')`

Binary Mode (Read)

```
f = open("files.zip", mode="rb")
data = f.read()
print(type(data))
print(data)
f.close()
```

```
b'PK\x03\x04\x14\x00\x00\x00\x08\x00U\xbd\xebV\xc2=j\x87\x1e\x00\x00\x00!
\x00\x00\x00\n\x00\x00\x00TODO11.txt\xe3\xe5\N,\xceH-/
\xe6\xe5\x82\xc0\xcc\xbc\x92\xd4\x9c\x9c\xcc\x82\xc4\xc4\x12^.w7w\x00PK\x01\x02
\x14\x00\x14\x00\x00\x00\x08\x00U\xbd\xebV\xc2=j\x87\x1e\x00\x00\x00!
\x00\x00\x00\n\x00\x00\x00\x00\x00\x00\x00\x01\x00
\x00\x00\x00\x00\x00\x00\x00TODO11.txtPK\x05\x06\x00\x00\x00\x01\x00\x01
\x008\x00\x00\x00Fx00\x00\x00\x00'
```

Formatting

- The argument of write has to be a string, so if we want to put other values in a file, we have to convert them to strings. The easiest way to do that is with str:

```
x = 52
f.write(str(x))
```

```
x = 52
f.write(f"{x}")
```


Example

- Assume that you have a txt file containing the following information. Find the average of each student.

```
File.txt
    STD1:90,80,2,100
    STD2:100,1,50,45.5
    STD3:50,1.1,70,2
```

Example

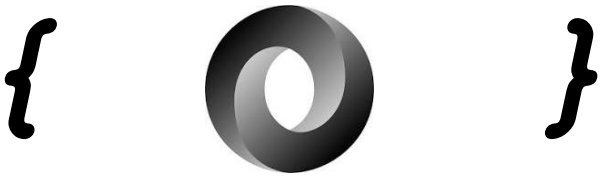
```
file = open("grades.txt", "r")
for i in file.readlines():
    TOTAL = 0
    COUNT = 0
    i = i.strip()
    i = i.split(":")
    STD_NAME = i[0]
    STD_GRADES = i[1]
    for i in STD_GRADES.split(","):
        TOTAL += float(i)
        COUNT += 1
    AVG = TOTAL/COUNT
    print(f"STUDENT NAME: {STD_NAME},\t AVERAGE: {AVG}")
file.close()
```


JSON

JSON

- JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays

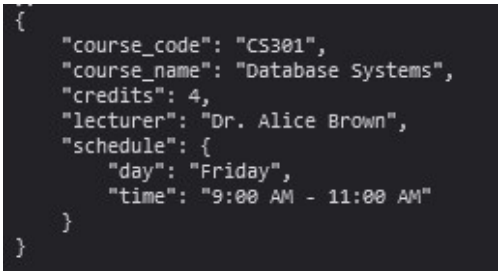
```
import json
```



JSON


```
{"university_name": "University of Example","location": {"country": "United States","city": "Example City","state": "Example
State"},"faculties": [{ "faculty_name": "Faculty of Science","departments": [{"department_name": "Computer Science","courses":
[{"course_code": "CS101","course_name": "Introduction to Computer Science","credits": 3,"lecturer": "Dr. John Smith","schedule": {"day":
"Monday","time": "10:00 AM - 12:00 PM"}},{"course_code": "CS201","course_name": "Data Structures and Algorithms","credits": 4,"lecturer":
"Prof. Emily Johnson","schedule": {"day": "Wednesday","time": "2:00 PM - 4:00 PM"}},{"course_code": "CS301","course_name": "Database
Systems","credits": 4,"lecturer": "Dr. Alice Brown","schedule": {"day": "Friday","time": "9:00 AM - 11:00 AM"}}]},{"department_name":
"Mathematics","courses": [{"course_code": "MATH101","course_name": "Calculus I","credits": 4,"lecturer": "Dr. Michael Brown","schedule":
{"day": "Tuesday","time": "9:00 AM - 11:00 AM"}},{"course_code": "MATH201","course_name": "Linear Algebra","credits": 3,"lecturer": "Prof.
Sarah Lee","schedule": {"day": "Thursday","time": "1:00 PM - 3:00 PM"}},{"course_code": "MATH301","course_name": "Probability
Theory","credits": 4,"lecturer": "Dr. Robert Taylor","schedule": {"day": "Wednesday","time": "10:00 AM - 12:00 PM"}}]}]},{"faculty_name":
"Faculty of Engineering","departments": [{"department_name": "Electrical Engineering","courses": [{"course_code": "EE101","course_name":
"Circuit Analysis","credits": 3,"lecturer": "Dr. David Wilson","schedule": {"day": "Monday","time": "1:00 PM - 3:00 PM"}},{"course_code":
"EE201","course_name": "Digital Signal Processing","credits": 4,"lecturer": "Prof. James Miller","schedule": {"day": "Wednesday","time":
"4:00 PM - 6:00 PM"}},{"course_code": "EE301","course_name": "Power Systems Engineering","credits": 4,"lecturer": "Dr. Lisa
Johnson","schedule": {"day": "Friday","time": "1:00 PM - 3:00 PM"}}]},{"department_name": "Mechanical Engineering","courses":
[{"course_code": "ME101","course_name": "Statics","credits": 3,"lecturer": "Dr. Elizabeth Clark","schedule": {"day": "Tuesday","time":
"11:00 AM - 1:00 PM"}},{"course_code": "ME201","course_name": "Thermodynamics","credits": 4,"lecturer": "Prof. Andrew White","schedule":
{"day": "Thursday","time": "3:00 PM - 5:00 PM"}},{"course_code": "ME301","course_name": "Fluid Mechanics","credits": 4,"lecturer": "Dr.
William Turner","schedule": {"day": "Wednesday","time": "2:00 PM - 4:00 PM"}}]}]},{"faculty_name": "Faculty of Business","departments":
[{"department_name": "Finance","courses": [{"course_code": "FIN101","course_name": "Financial Management","credits": 3,"lecturer": "Dr.
Susan Roberts","schedule": {"day": "Monday","time": "9:00 AM - 11:00 AM"}},{"course_code": "FIN201","course_name": "Investments","credits":
3,"lecturer": "Prof. Charles Brown","schedule": {"day": "Wednesday","time": "1:00 PM - 3:00 PM"}}]},{"department_name":
"Marketing","courses": [{"course_code": "MKT101","course_name": "Principles of Marketing","credits": 3,"lecturer": "Dr. Jennifer
Davis","schedule": {"day": "Tuesday","time": "10:00 AM - 12:00 PM"}},{"course_code": "MKT201","course_name": "Consumer Behavior","credits":
3,"lecturer": "Prof. Michael Wilson","schedule": {"day": "Thursday","time": "2:00 PM - 4:00 PM"}}]}]}]}
```

JSON - Pretty



```
{
    "course_code": "CS301",
    "course_name": "Database Systems",
    "credits": 4,
    "lecturer": "Dr. Alice Brown",
    "schedule": {
        "day": "Friday",
        "time": "9:00 AM - 11:00 AM"
    }
}
```

```
{
    "university_name": "University of Example",
    "location": {
        "country": "united states",
        "city": "Example City",
        "state": "Example State"
    },
    "faculties": [
        {
            "faculty_name": "Faculty of Science",
            "departments": [
                {
                    "department_name": "Computer Science",
                    "courses": [
                        {
                            "course_code": "CS101",
                            "course_name": "Introduction to Computer Science",
                            "credits": 3,
                            "lecturer": "Dr. John Smith",
                            "schedule": {
                                "day": "Monday",
                                "time": "10:00 AM - 12:00 PM"
                            }
                        },
                        {
                            "course_code": "CS201",
                            "course_name": "Data Structures and Algorithms",
                            "credits": 4,
                            "lecturer": "Prof. Emily Johnson",
                            "schedule": {
                                "day": "Wednesday",
                                "time": "2:00 PM - 4:00 PM"
                            }
                        },
                        {
                            "course_code": "CS301",
                            "course_name": "Database Systems",
                            "credits": 4,
                            "lecturer": "Dr. Alice Brown",
                            "schedule": {
                                "day": "Friday",
                                "time": "9:00 AM - 11:00 AM"
                            }
                        }
                    ]
                },
```


JSON - Write

```
data = {"Candidate": [{
    "name": "John Doe",
    "age": 30,
    "skills": ["Python", "JavaScript"],
    "is_student": False}
]}

with open("data.json", "w") as json_file:
    json.dumps(data, json_file, indent=4)
```

JSON - Read

```
def read_json(file_path):
    try:
        with open(file_path, "r") as json_file:
            data = json.load(json_file)
            return data
    except FileNotFoundError:
        print("Error: File not found.")
        return None
    except json.JSONDecodeError:
        print("Error: Invalid JSON format.")
        return None
```


JSON – Example 1

- Using the given file “University Data” find the total number of faculties in the university

JSON – Example 1

```
num_faculties = len(university_data["faculties"])
print("Total number of faculties:", num_faculties)
```


JSON – Example 2

- Using the given file “University Data”, list all the course codes and their corresponding course names.

JSON – Example 2

```
for faculty in university_data["faculties"]:
    for department in faculty["departments"]:
        for course in department["courses"]:
            print(course["course_code"], course["course_name"])
```


JSON - Example

- Using the given file “University Data” find the total number of faculties in the university
- List all the course codes and their corresponding course names.

Excel Files

39

Excel Files

- Its you task now to learn it, don't forget that you will need it in MP1

```
Hint: "openpyxl"
```

40

Additional Resources

- https://www.programiz.com/python-programming/file-operation
- https://runestone.academy/ns/books/published/thinkcspy/Files/toctree.html
- https://www.w3schools.com/python/python_file_handling.asp
- https://www.geeksforgeeks.org/reading-binary-files-in-python/
- https://www.tutorialsteacher.com/python/python-read-write-file
- https://www.prepbytes.com/blog/python/seek-function-in-python/