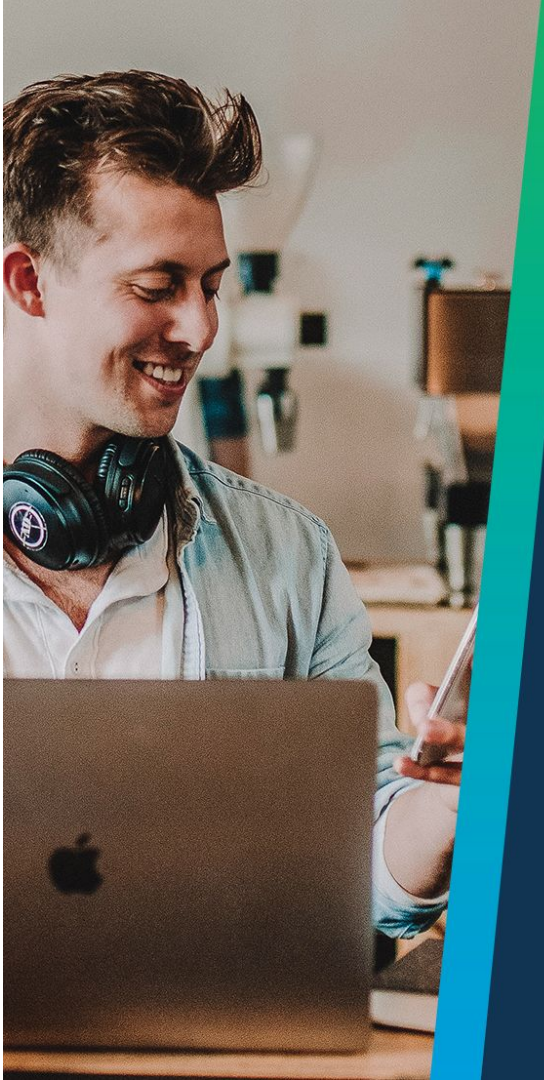


Software Engineering Bootcamp

Hyperiondev

External Sources and Error Handling

Lecture – Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- No question is daft or silly - ask them!
- There are Q/A sessions at the end of the session, should you wish to ask any follow-up questions.
- For all non-academic questions, please submit a query: www.hyperiondev.com/support
- Report a safeguarding incident: http://hyperiondev.com/safeguardreporting

Objectives

1. **File I/O:** Accurately read and write data to/from at least two different file formats (e.g., CSV, JSON, text) within a Python script.
2. **Data Extraction**: Extract specific information from various file formats and manipulate the extracted data.
3. **Error Identification**: Recognize common error types (e.g., syntax errors, runtime errors, exceptions) in Python code.
4. **Error Handling**: Implement a try-except block to handle a specified error in the code and prevent program crashes.
5. **Exception Handling**: Use different exception types to handle specific errors and provide informative error messages.

Poll

What is the output of the following code snippet?

```
numbers = [1, 2, 3, 4, 5]
result = 0
for number in numbers:
    if number % 2 == 0:
        continue
    result += number
print(result)
```

- 15
- 10
- 9
- 6

Poll

What will be the output of the following code snippet?

```
count = 0
while count < 5:
    print(count)
    count += 1
```

- 0 1 2 3 4 5
- 0 1 2 3 4
- 1 2 3 4 5
- 1 2 3 4

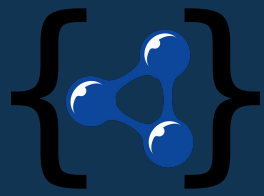
Introduction

Hyperiondev

Understanding External Sources

They couldn't play soccer because they couldn't see the ball. They didn't want to go inside and play a game because it was a beautiful springtime night and they liked being ...[1]


```
id,name,price
1,Laptop,1200
2,Smartphone,800
3,Headphones,200
4,Mouse,50
5,Monitor, 1000
6,Desk,600
7,Speakers,400
```




```
{
    "track_id": "3",
    "name": "Inde",
    "artist": "Bucie",
    "album": "Inde",
    "duration_ms":
373000,
    "popularity": 3
}
```


Hyperiondev

[1] HIDE AND SEEK | Free Children's book from Monkey Pen

Working with Files in Python: I/O

```
file = open("filename.txt", "access_mode")
content = file.read()
file.close()
```

Text file (.txt)

opening

Read Only	r
Read and Write	r+
Write Only	w
Write and Read	w+
Append Only	a
Append and Read	a+

File Access Modes

closing

Must be closed to avoid issues like memory leaks

Hyperiondev

Error Handling in Python

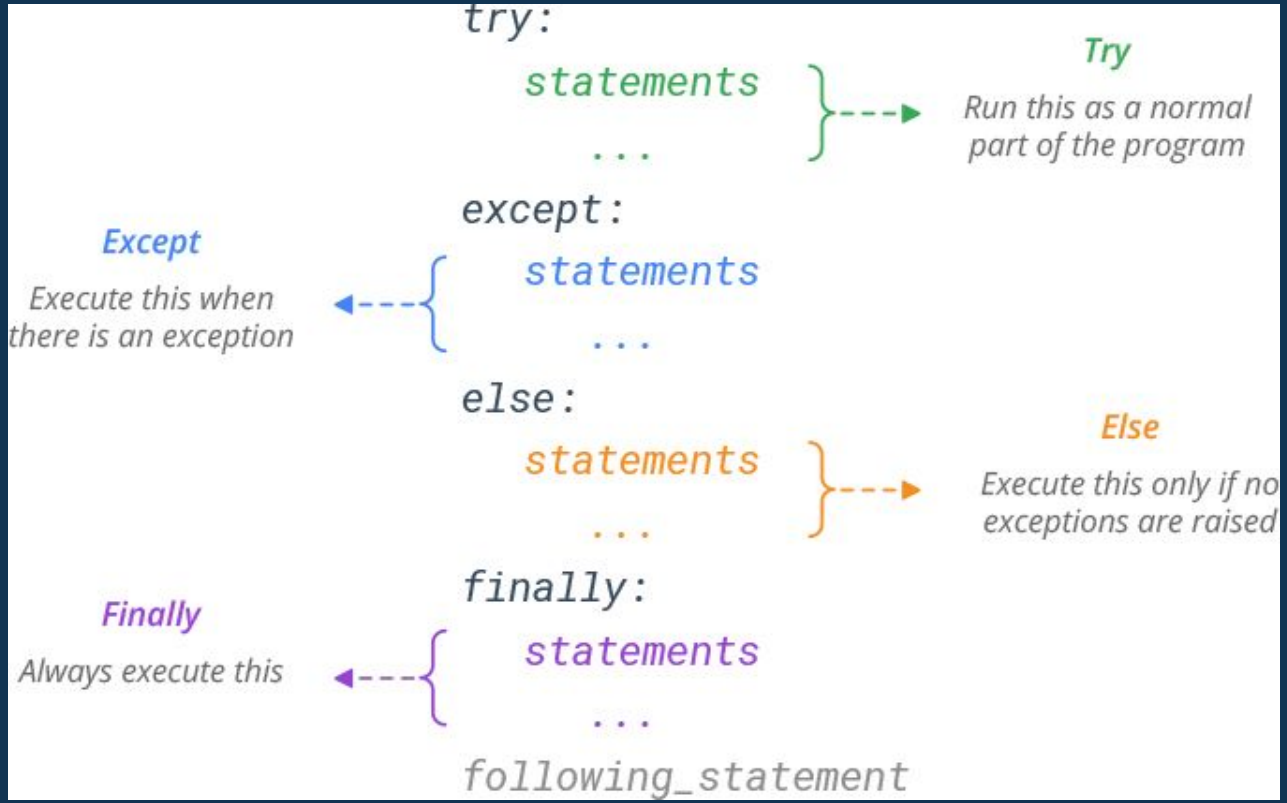
- An **error/exception** is an unexpected event that interrupts the normal execution of a computer program, preventing it from achieving its intended outcome.

TypeError
number = 5 text = "hello" result = number + text

IndexError
numbers = [1, 2, 3] index = 5 print(numbers[index])

ZeroDivisionError
x = 10 y = 0 result = 10 / 0

Implementing Error Handling



Lesson Conclusion and Recap

- File Operations
  - Open/close, read/write files (text, CSV, JSON)
  - File modes: read, write, append, binary
  - open(), read(), write(), with statement
- Working with External Data Formats
  - Handling text
- Error Handling with Exceptions
  - Importance of error handling
  - Common exceptions: FileNotFoundError, IOError, ValueError, KeyError
  - try, except, else, finally blocks

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any

Hyperiondev

Thank you for joining us

Take regular breaks.
Stay hydrated.
Avoid prolonged screen time.
Remember to have fun :)

Some useful links

Python: https://www.python.org/downloads/

VS Code: https://code.visualstudio.com/download