LESSON 7

main.py

```
with open("app.log") as f:
  for line in f:
    print(line)
```

Py

Files & Exceptions

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

PYTHON SCRIPTING & OOP

LESSON 7

Files, JSON, CSV and Exceptions

with open, pathlib, json, csv, try/except/finally and raising errors

Reading and Writing Files

files.py

```
with open("notes.txt", "w", encoding="utf-8") as f:
    f.write("first line\n")

with open("app.log", encoding="utf-8") as f:
    for line in f:
        if "ERROR" in line:
            print(line.strip())
```

- with open opens a file and closes it automatically, even if an error happens. Mode r reads, …
- Loop over the file object to read line by line, which works even for files larger than your …

01

pathlib

paths.py

```python
from pathlib import Path
logs = Path.home() / "logs"
logs.mkdir(exist_ok=True)
for p in logs.glob("*.log"):
    print(p.name, p.stat().st_size)
text = Path("config.ini").read_text(encoding="utf-8")
```

- pathlib treats paths as objects. The slash operator joins paths, and methods like exists, …
- It works the same on Windows, Linux and Mac.

02

JSON and CSV

data.py

```
import json, csv
with open("config.json") as f:
    cfg = json.load(f)
print(cfg["database"]["port"])

with open("students.csv", newline="") as f:
    for row in csv.DictReader(f):
        print(row["name"], row["score"])
```

- json dot load reads JSON into dictionaries and lists, and json dot dump writes them back. …
- With these two modules Python talks to almost every system and spreadsheet.

03

try, except, else and finally

safe.py

```
try:
    with open("scores.txt") as f:
        scores = [int(line) for line in f]
except FileNotFoundError:
    print("file missing"); scores = []
except ValueError as e:
    print("bad number:", e)
else:
    print("average", sum(scores) / len(scores))
finally:
    print("done")
```

- When something goes wrong, Python raises an exception. Catch it with try and except, naming …
- else runs if no error happened, and finally always runs. Catch specific errors, never hide …

04

Raising Your Own Errors

raise.py

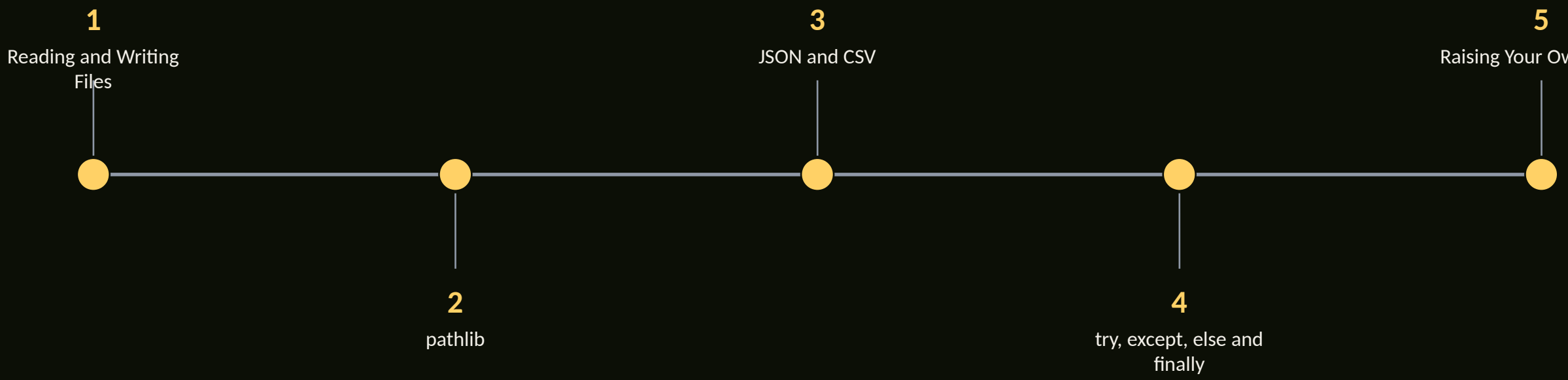
```python
class InsufficientFunds(Exception):
    pass

def withdraw(balance, amount):
    if amount <= 0:
        raise ValueError("amount must be positive")
    if amount > balance:
        raise InsufficientFunds(f"need {amount - balance} more")
    return balance - amount
```

- raise creates an exception when your function receives bad input. You can also define your …
- Clear errors at the right place make bugs easy to find.

05

Lesson Map


1
Reading and Writing Files
2
pathlib
3
JSON and CSV
4
try, except, else and finally
5
Raising Your Own Errors

Sources and Further Reading

files.py

```python
with open("notes.txt", "w", encoding="utf-8") as f:
    f.write("first line\n")

with open("app.log", encoding="utf-8") as f:
    for line in f:
        if "ERROR" in line:
            print(line.strip())
```

PYTHON SCRIPTING & OOP · 7

data.py

```python
import json, csv
with open("config.json") as f:
    cfg = json.load(f)
print(cfg["database"]["port"])

with open("students.csv", newline="") as f:
    for row in csv.DictReader(f):
        print(row["name"], row["score"])
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari