Importing Text Files in Python

Importing Text Files

Xavier Morera

Helping developers create amazing applications

@xmorera / www.xaviermorera.com / www.bigdatainc.org

Different Types of Files

JPG

TXT

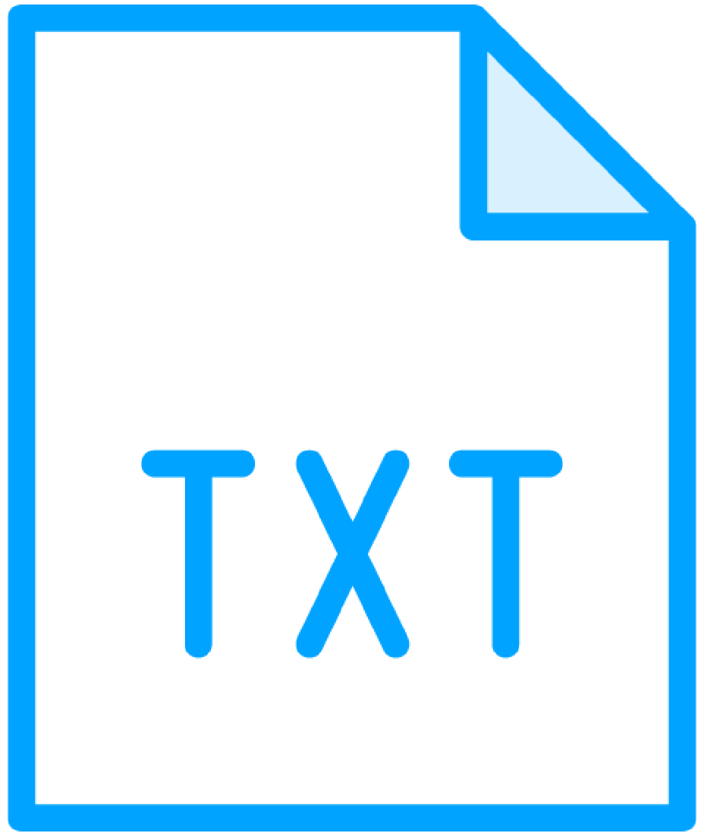
MP4

EXE

SQL

ZIP

Text Files



Text Files

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

Array
(Numpy)

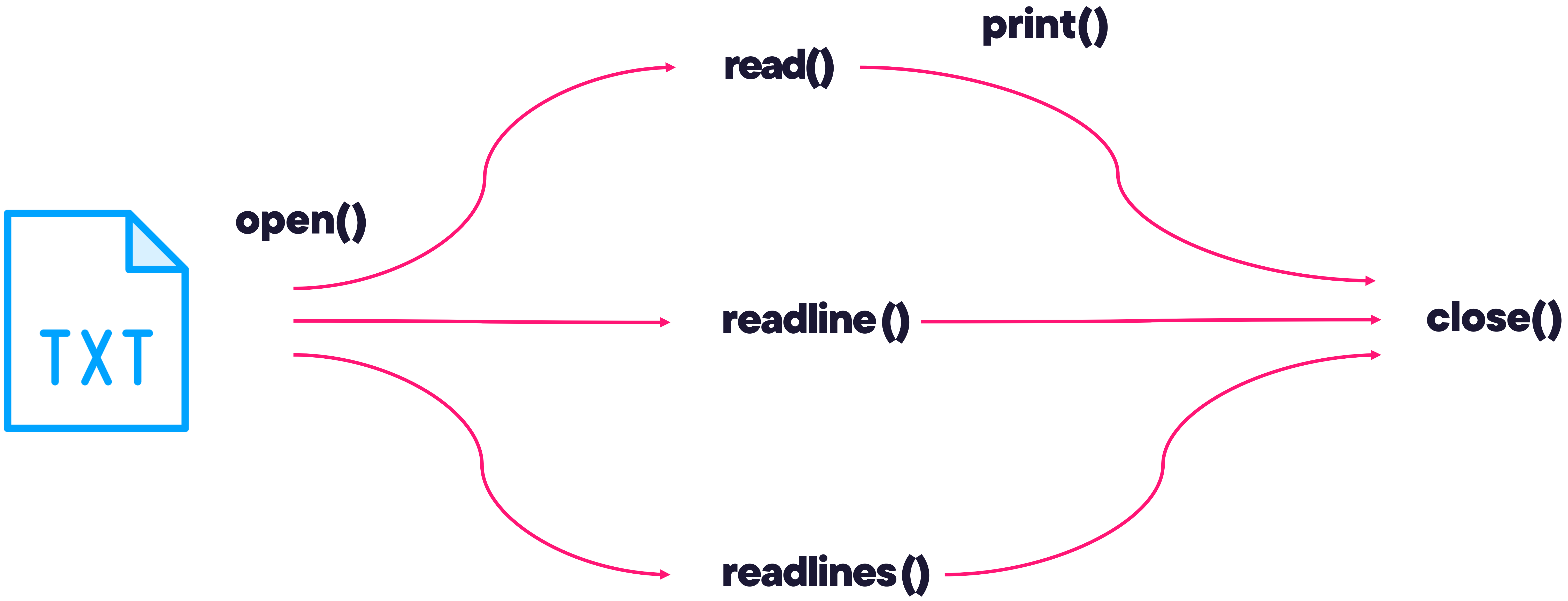
Text files in Python, simple and clean
Lines of code, words unseen
Open and close, read and write
A programmer's tool, day and night

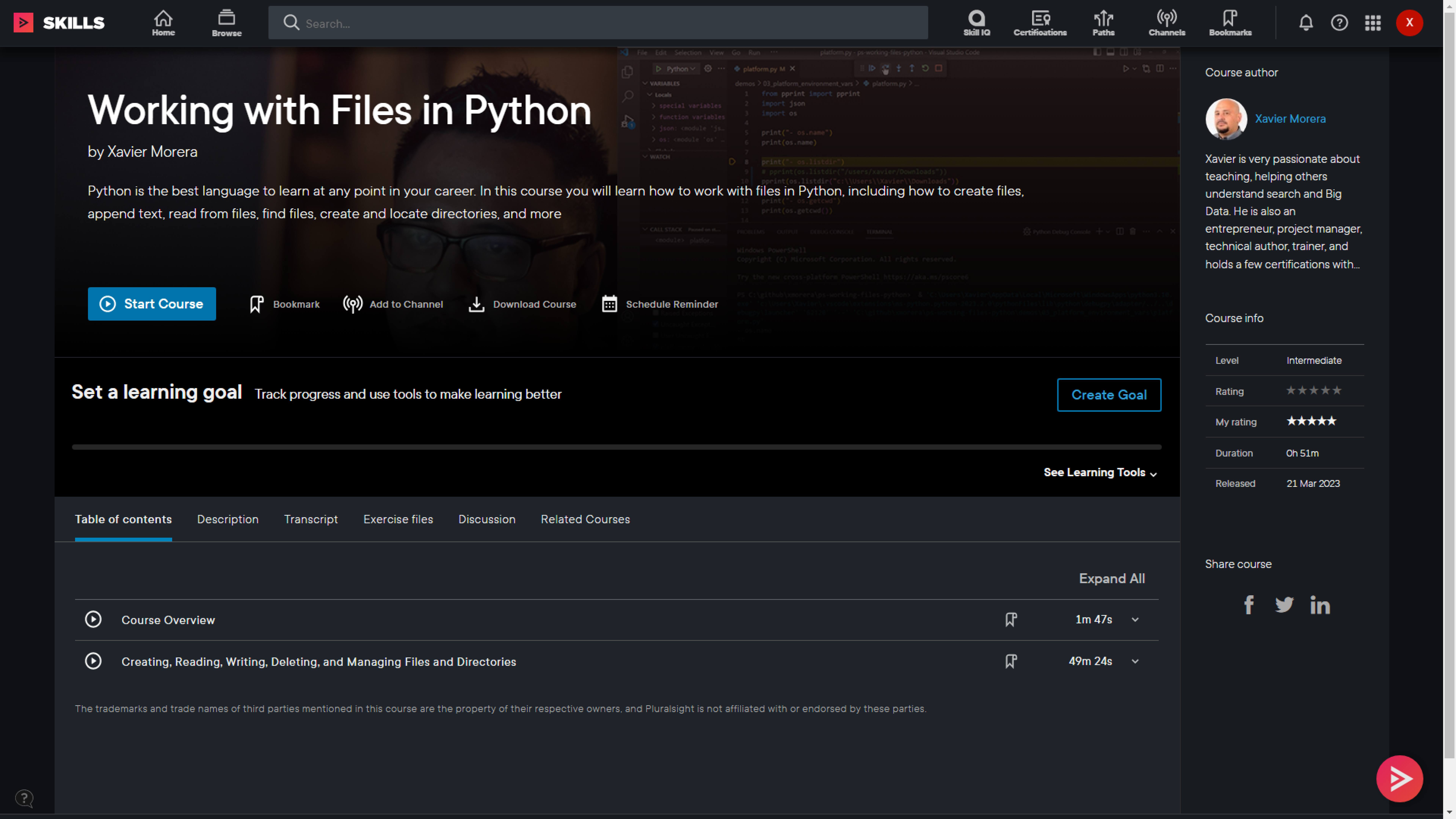
Text
(A poem)

```
Xavier, 40, SJO
Irene, 25, BAR
Carlos, 40, YYZ
Mario, 35, LIR
```

CSV
(Pandas)

Reading Text Files





Working with Files in Python

by Xavier Morera

Python is the best language to learn at any point in your career. In this course you will learn how to work with files in Python, including how to create files, append text, read from files, find files, create and locate directories, and more

Set a learning goal

Track progress and use tools to make learning better

Course Overview	1m 47s
Creating, Reading, Writing, Deleting, and Managing Files and Directories	49m 24s

The trademarks and trade names of third parties mentioned in this course are the property of their respective owners, and Pluralsight is not affiliated with or endorsed by these parties.

Course author

Xavier Morera

Xavier is very passionate about teaching, helping others understand search and Big Data. He is also an entrepreneur, project manager, technical author, trainer, and holds a few certifications with...

Course info

Level	Intermediate
Rating	★★★★★
My rating	★★★★★
Duration	0h 51m
Released	21 Mar 2023

Share course

Opening and Closing Files

Open

Use the open( ) **function to access a file**

- Reading, writing, or for appending

Function takes several arguments

- Name of the file to be opened
- In which mode the file should be opened
  - Specify type of operation to perform
- Other possible parameters

Returns a file object on success

- An exception is raised if it does not exist

Opening Files

file

mode

buffering
encoding
errors
newline
closed
opener

```
open("file.txt", "r",  )
```

Opening Files

```
file = open("file.txt", "r")
```

Opening Files

```
file
```

Reading from Files

```
file.read()
```

Reading from Files

```
file.readline()
```

Reading from Files

```
file.readlines()
```

```
* I'll come back to reading in a moment...
```

```python
# Open the file
file = open("file.txt", "r")

# Perform some operations...

# Close the file
file.close()
```

Close

When done reading from a file, it is necessary to close the file

- Using the `close()` method

Any operation on a closed file will raise a ValueError

Do not leave files open unnecessarily

But there is a potential problem…

#1 You may forget to call the close method…

#2 Or an exception may be raised…

#3 Code does not execute…

```python
with open ("file.txt", "r") as file:
    # Perform some operations...
```

Using a Context Manager

Use the with keyword to create a context

No need to call close()

Resources freed up automatically on exit

File is closed

Opening and closing files

Reading and Printing Lines from a Text File

Reading Files

read

readline

readlines

read

Reads the entire contents of the file

```
with open('example.txt', 'r') as file:
    content = file.read()
```

readline

Reads one line at a time

```python
with open('example.txt', 'r') as file:
    line = file.readline()
```

readlines

Reads all lines into an array object

```python
with open('example.txt', 'r') as file:
    lines = file.readlines()
```

Other Ways of Reading Files

line in file

mmap

line in file

Iterate over the file object

```
with open('example.txt', 'r') as file:
    for line in file:
        # Perform some operations
```

mmap

Memory-map a file for random access

```python
import mmap

with open('file.txt', 'r') as file:
    with mmap.mmap(file.fileno(), 0, access=mmap.ACCESS_READ) as mem:
        content = mem[:].decode('utf-8')
```

Printing lines

print

pprint

Printing Lines From a File

```python
from pprint import pprint

# Print one line at a time
with open("file.txt", "r") as file:
    for line in file:
        print(line)

# Print many lines using pprint
with open("file.txt", "r") as file:
    lines = file.readlines()
    pprint(lines)
```

Reading and printing lines from a text file

Takeaway

Different types of files available

- Text, music, images, data…

Text files can hold different types of information

- Plain text
- Numeric data
- Tabular data

Takeaway

Read text in a file

- Open the file

Read the file contents

- All text
- One line at a time
- All lines

Print the contents of the file

- print or pprint methods

Up Next:

Loading and Reading Flat Files