

Intro to Coding with Python—Working with Files

Dr. Ab Mosca (they/them)

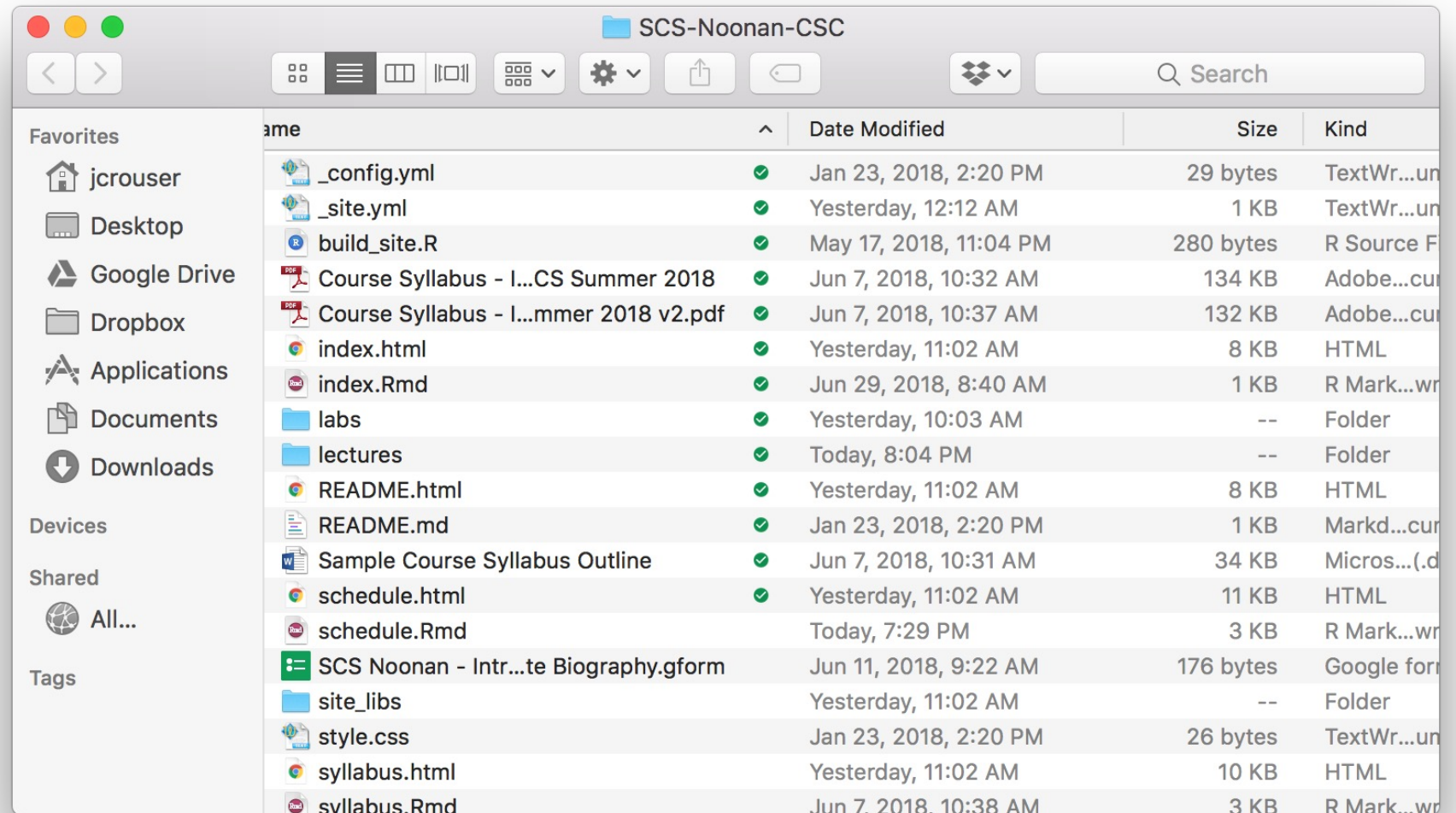
Plan for Today

- What is a file
- Read data in
- Writing data out

RECAP:
Computer
Drive Stores
Information

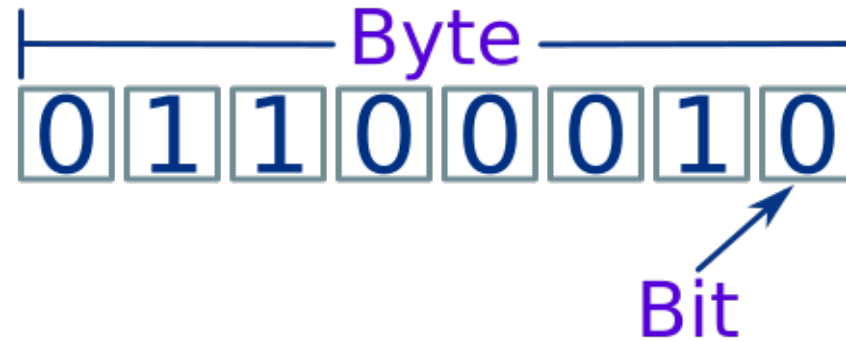


What is a file?



Working with files

- Containers of **bits**, organized into **bytes**



- Could represent text, images, music, movies, programs, applications, list of files (folders)... but underneath, they're all the same: 0s and 1s
- We'll start by playing with text files

Demo: looking inside a text file

The screenshot shows the HexEd.it web application in a browser window. The browser's address bar displays "Secure | https://hexed.it". The application's top navigation bar includes buttons for "New file", "Open file", "Reload", "Export", "Undo", "Redo", "Tools", "Settings", and "Help".

The main interface is divided into several sections:

- File Information:** Displays file details such as "File Name: -Untitled-", "File Size: 1,024 bytes (1 KiB)", and "New File Size: 970 bytes".
- Data Inspector (Little-endian):** Shows the data type as "Binary" with a row of eight radio buttons. It also includes "Unsigned (+)" and "Signed (±)" options.
- Data Inspector (Big-endian):** A section for viewing data in big-endian format.
- Go To:** A section for navigating to specific memory addresses, showing "Current Address: 0x00000000" and "Last Address: 0x000003C9".
- Search:** A section for searching through the file's content.

The central part of the interface is a hex editor displaying a grid of hexadecimal values. The first row shows "00 00 00 00 00 00 00 00" repeated across the grid, indicating that the file contains null bytes.

Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

```
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/l...
def main():

    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.read()
    print(text)

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

Ln: 22 Col: 6

Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

```
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/l...
def main():
    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.read()
    print(text)

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

"r" stands for READ-MODE

Ln: 22 Col: 6

Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

```
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/l...
def main():
    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.read()
    print(text)

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

as with email or texts
opening != reading

Ln: 22 Col: 6

Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

```
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/l...
def main():

    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.read()
    print(text)

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

file is now an **object** with a **.read()** method that returns a string

Ln: 22 Col: 6

Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

```
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/I...
def main():

    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.readlines()
    print(text)

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

it also has a **.readlines()** method that returns a **list** of strings

Ln: 22 Col: 6

Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

```
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/I...
def main():

    # Open file for reading
    file = open("test.txt", "r")

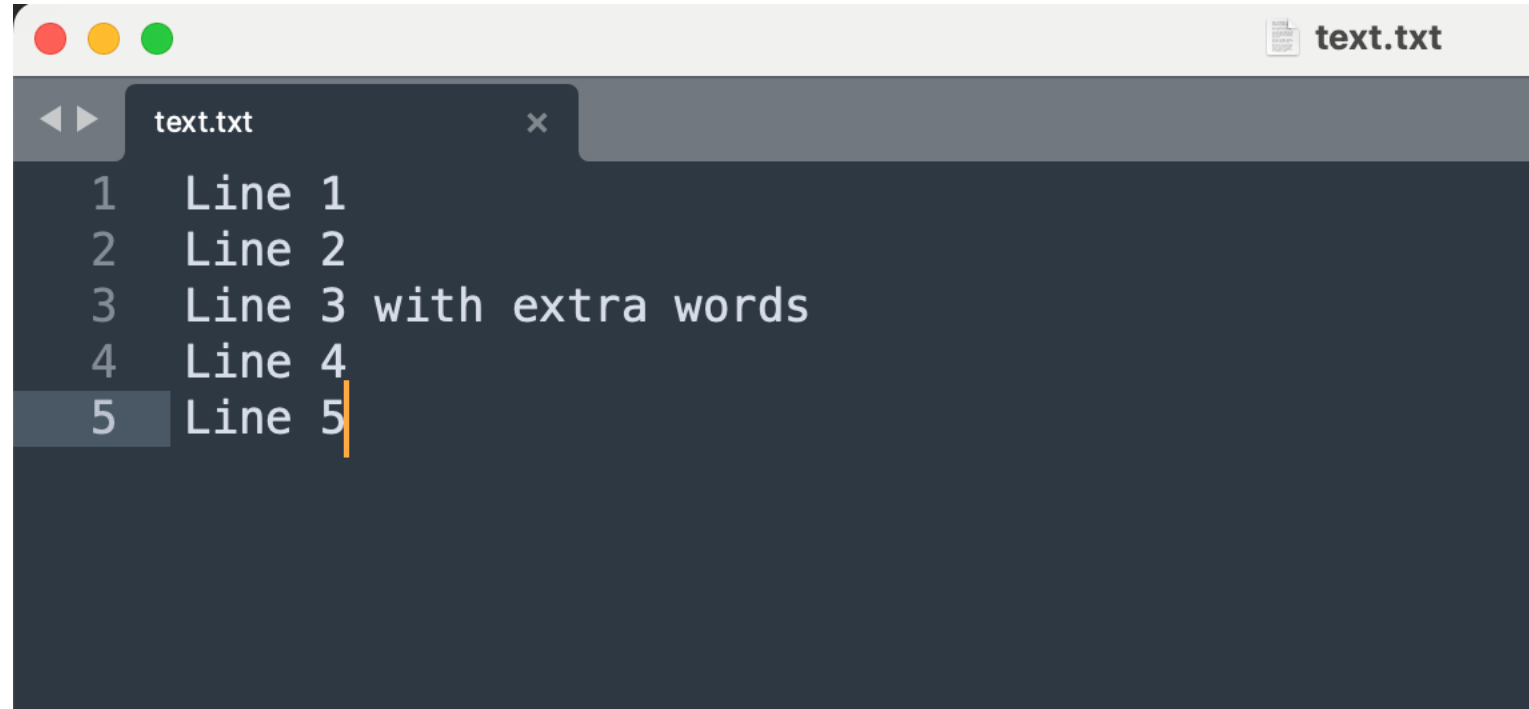
    # Read the file and print its contents
    text = file.read()
    print(text)

    # Close the file
    file.close()

Ln: 22 Col: 6
```

after you `.read()` or `.readlines()`
a file, remember to `.close()` it

Demo: Reading a text file



```
1 Line 1
2 Line 2
3 Line 3 with extra words
4 Line 4
5 Line 5
```

The image shows a screenshot of a text editor window. The window title bar at the top right says "text.txt". Below the title bar, there is a tab labeled "text.txt" with a close button (X) on the right. The main content area of the window displays five lines of text, each numbered from 1 to 5. The text is as follows: "1 Line 1", "2 Line 2", "3 Line 3 with extra words", "4 Line 4", and "5 Line 5". The fifth line is highlighted with a dark blue background, and a vertical orange cursor is positioned at the end of the text on that line.

Key points for reading files

- Three-step process:
 1. `.open()`
 2. `.read()` or `.readlines()`
 3. `.close()`
- All three steps, always in that order
- If you want to `.read()` a file multiple times, you have to repeat the whole process

```
file = open("test.txt", "r")
text1 = file.read()
text2 = file.read() # Empty!
file.close()
```

Writing data to a text file

- The process looks very similar when we want to **write** data to a file:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...
def main():

    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

Ln: 15 Col: 21

Writing data to a text file

- The process looks very similar when we want to **write** data to a file:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...
def main():
    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

exactly the same function to .open() a file

Ln: 15 Col: 21

Writing data to a text file

- The process looks very similar when we want to **write** data to a file:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...  
def main():  
  
    # Open file for writing  
    file = open("test2.txt", "w")  
  
    # Write a string to the file  
    text = "Put this string into a file."  
    file.write(text)  
  
    # Close the file  
    file.close()
```

“w” stands for **WRITE-MODE**

Ln: 15 Col: 21

Writing data to a text file

- The process looks very similar when we want to **write** data to a file:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...
def main():

    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

file is now an **object** with a **.write()** method that takes in a string

Ln: 15 Col: 21

Writing data to a text file

- The process looks very similar when we want to **write** data to a file:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...
def main():

    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

Ln: 15 Col: 21

as before, after you `.write()` to a file remember to `.close()` it

Writing data to a text file

- Some quirks to be aware of when **writing** to files:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...
def main():
    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

if this file **doesn't exist**
Python will create it

Ln: 15 Col: 21

Writing data to a text file

- Some quirks to be aware of when **writing** to files:

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...
def main():
    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

if this file **does exist**
Python will overwrite it

Ln: 15 Col: 21

Writing data to a text file

- Some quirks to be aware of when **writing** to files:

if you want to
add to an existing file
instead of overwrite it

```
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Classes/Python/Scripts/Scripts.py
def main():

    # Open file for writing
    file = open("test2.txt", "a")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

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

"a" stands for
APPEND-MODE

Ln: 15 Col: 21

Key points for writing to files

- Three-step process:
 1. `.open()`
 2. `.write()`
 3. `.close()`
- Unlike `.read()`, you can `.write()` to an `.open()` file as many times as you want (appending each time)

```
file = open("test2.txt", "w")
file.write("Hello")
file.write("there!")
file.close()
```

- If you want a new line, you have to add it yourself! (`\n`)

15-minute exercise: read and write

Write a program that:

1. reads the file **horizontal.txt** (under Demos on the course website; you will need to download it and save it in the same folder as your .py file)
2. breaks it into individual words
3. writes the words to a new file **vertical.txt**, each one on its own line

```
horizontal.txt
Scene 1
Synopsis:
At his court, Orsino, sick with love for the Lady Olivia, learns from his messenger that she is
grieving for her dead brother and refuses to be seen for seven years.
Enter Orsino, Duke of Illyria, Curio, and other Lords,
with Musicians playing.
```

```
vertical.txt
Scene
1
Synopsis:
At
his
court,
Orsino,
sick
with
love
for
the
```


Discussion

What did you come up with?