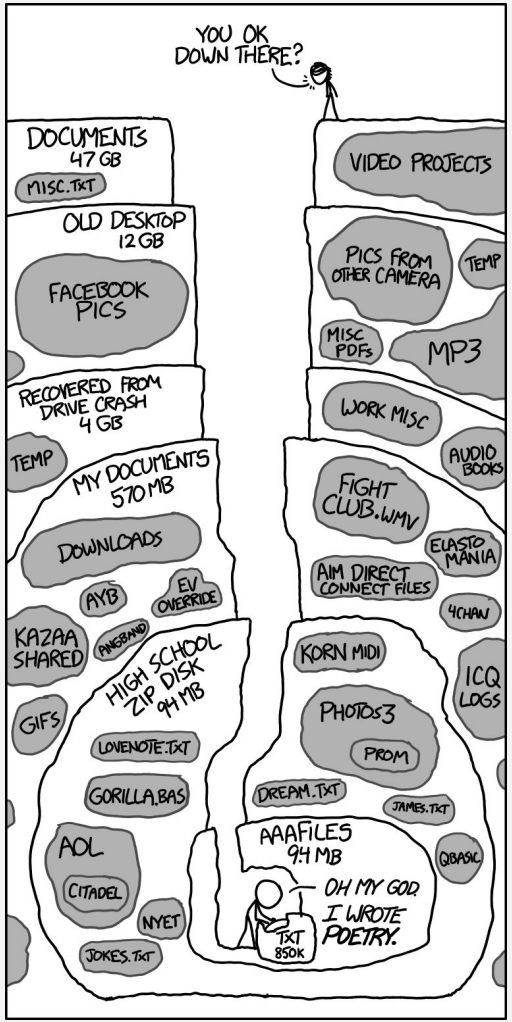
CSc 110
File Reading and writing

Adriana Picoral (she/her/hers)



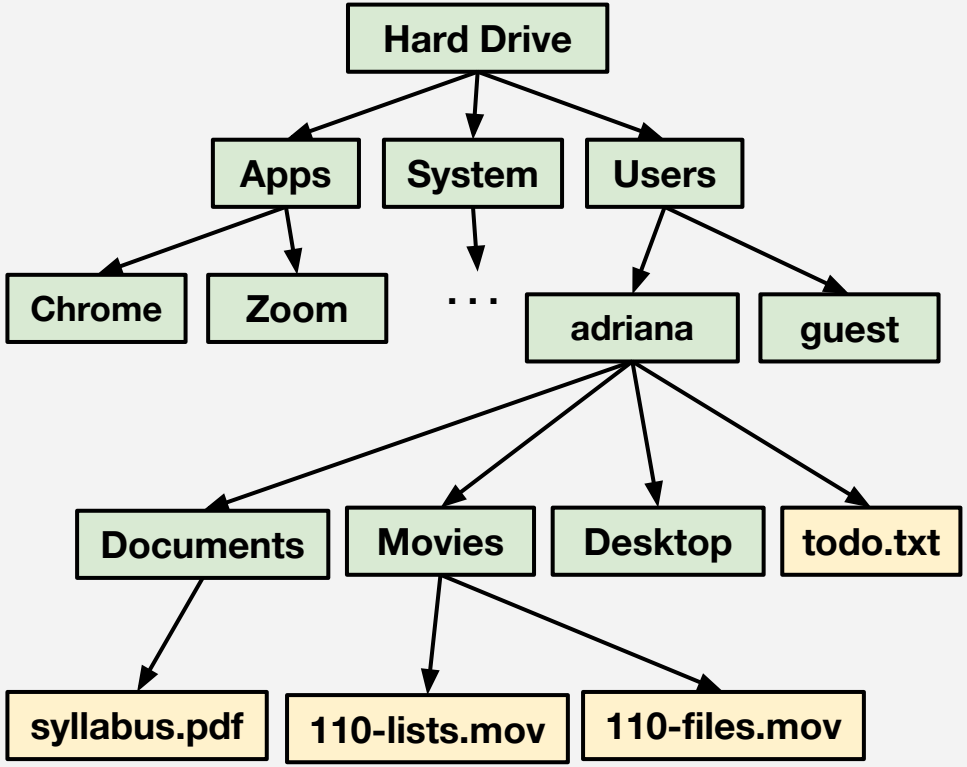
https://xkcd.com/1360/

Announcements

- Exams this week
- Review session: MCPRK 105 5pm to 6:50pm tomorrow
- D2L grades
- Videos on schedule

Files and File Systems

- On (at least most of) our computers, there is a **file system** via which we can create, save, modify, and remove files
  - On Mac: can browse with Finder
  - On Windows: can browse with windows explorer
- File systems often hierarchical



Opening a file

- To open a file in a python program:

```
a_file = open(file_name, mode)
```

- **`file_name`** should be the name of the file to open
  - It can also be a path
- **`mode`** should be the mode in which to open in
  - `'a'` `'r'` `'w'`
- After opening, use the **`a_file`** object to read from or write to the file

Reading a file

- Two functions
    - **`file.readline()`** reads one line from a file, returns a string. Calling it repeatedly will read a sequence of lines.
        - Returns an empty string if it has reached the end of the content
    - **`file.readlines()`** reads all of the lines, returns a list of strings

What would it print?

info.txt:

```
The quick brown fox
jumped over
the lazy
bear
sitting by the tree
```

```
info = open('info.txt', 'r')
line = info.readline()
print(line)
line = info.readline()
print(line)
```

What would it print?

info.txt:

```
The quick brown fox
jumped over
the lazy
bear
sitting by the tree
```

```
info = open('info.txt', 'r')
line = info.readline()
print(line)
line = info.readline()
line = info.readline()
print(line)
line = info.readline()
print(line)
```

What would it print?

info.txt:

```
The quick brown fox
jumped over
the lazy
bear
sitting by the tree
```

```
info = open('info.txt', 'r')
lines = info.readlines()
for i in lines:
    print(lines)
```

What would it print?

info.txt:

```
The quick brown fox
jumped over
the lazy
bear
sitting by the tree
```

```
info = open('info.txt', 'r')
line = info.readline()
lines = info.readlines()
for i in lines:
    print(i)
line = info.readline()
print(line)
```

Can also iterate over a file directly

```
info = open('info.txt', 'r')
for line in info:
    print(line)
```

This is because the **file** type is iterable!

Which one prints something different than the rest?

A

```
info = open('info.txt', 'r')
for line in info:
    print(line)
```

B

```
info = open('info.txt', 'r')
line = info.readline()
while line != '':
    print(line)
    line = info.readline()
```

C

```
info = open('info.txt', 'r')
lines = info.readlines()
for line in lines:
    print(line)
```

D

```
info = open('info.txt', 'r')
line = info.readlines()
print(line)
```

Removing characters

- You can use the **`strip`** function to remove characters at the beginning and end of a string
  - **`string.strip(chars)`**
    - will remove any of the characters in **`chars`** from the beginning or end of **`string`**
  - **`string.lstrip(chars)`**
    - will remove any of the characters in **`chars`** from the beginning of **`string`**
  - **`string.rstrip(chars)`**
    - will remove any of the characters in **`chars`** from the end of **`string`**

What would it print?

numbers.txt:

```
1,1,5
2,7,70
3,30,50
4,5,50
4,50,50
```

```
info = open('numbers.txt', 'r')
lines = info.readlines()
for line in lines:
    s = line.split(',')
    if int(s[1]) > 15:
        to_print = s[2].strip('\n')
        print(to_print)
```

Write the program

- Create a file named numbers.txt with the contents to the right in it
- Create a new program that
  - Reads a file named numbers.txt
  - Sums all of the numbers in the whole file
  - Prints out the final sum
- Given the input shown on this slide, the program should print: `15`

numbers.txt:

```
1
0
5
7
2
```

Write the program

- Create a file named numbers.txt with the contents to the right in it
- Create a new program that
  - Reads a file named numbers.txt
  - Sums all of the numbers in the whole file
  - Prints out the final sum
- Given the input shown on this slide, the program should print: **24**

numbers.txt:

```
1,2,1
0,4,1
5,5,5
```

```
def main():
    number_file = open('numbers.txt', 'r')
    value = 0
    for line in number_file:
        value += int(line)
    print(value)

main()
```

Write the program

- Read in the contents of dialogue.txt
- Print out every-other line

dialogue.txt:

```
Joker: Its simple, we kill the Batman.
Maroni: If its so simple, why haven't you done it yet?
Joker: If your good at something never do it for free.
Chechen: How much you want?
Joker: Half.
```

output:

```
Joker: Its simple, we kill the Batman.
Joker: If your good at something never do it for free.
Joker: Half.
```

Modes

- To read a file:
  - Use `'r'` for reading the contents of a file
- To write to a file: use
  - Use `'a'` to append to the existing file content
  - Use `'w'` to write to a file, and replace existing content

Writing to a file

- After you have opened the file in either the read or append mode

```
a_file = open(file_name, mode)
```

- Use the **`write`** function to write text content to the file

```
a_file.write('put this content in a file')
```

- When finished writing, **`close`** the file **`a_file.close()`**

What would the contents of words.txt be?

words.txt:

```
The lion
jumped over
the bear
```

```
words = open('words.txt', 'w')
words.write('The slow wolf')
words.write('jumped over')
words.write('the bear')
words.close()
```

What would the contents of words.txt be?

words.txt:

```
The lion
jumped over
the bear
```

```
words = open('words.txt', 'w')
words.write('The slow wolf\n')
words.write('jumped over\n')
words.write('the bear\n')
words.close()
```

What would the contents of words.txt be?

words.txt:

```
The lion
jumped over
the bear
```

```
words = open('words.txt', 'a')
words.write('The slow wolf\n')
words.write('jumped over\n')
words.write('the bear\n')
words.close()
```

What would it print?

dictionary.txt:

```
aardvark
abacus
acorn
advil
aerodynamic
```

```
word_file = open('dictionary.txt', 'a')
word_file.write('after\n')
word_file.write('affiliate\n')
word_file.write('aggregate\n')
word_file.close()
word_file = open('dictionary.txt', 'r')
lines = word_file.readlines()
i = 0
for line in lines:
    if i % 2 == 0:
        print(line)
    i += 1
```