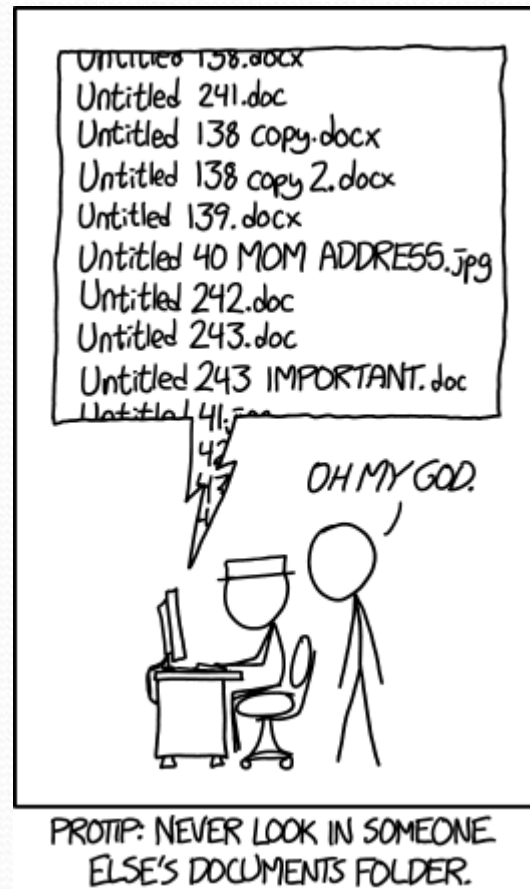


CS 115, Autumn 2021

Lecture 30: files



Thanks to Marty Stepp and Stuart Reges for parts of these slides

File Input/output (I/O)

- **name** = `open("filename")`
 - opens the given file for reading, and returns a file object
- **name**.`read()` - file's entire contents as a string

```
>>> f = open("weather.txt")
>>> f.read()
'42\n34\n35\n46\n45\n43\n43\n49'
```

File Input/output (I/O)

- **name** = `open("filename")`
 - opens the given file for reading, and returns a file object
- **name**.`readlines()`
 - file's entire contents as a list with each line as a string

```
>>> f = open("weather.txt")
>>> f.readlines()
['42', '34', '35', '46', '45', '43', '43', '49']
```

File paths

- **absolute path:** specifies a drive or a top "/" folder

```
C:/Documents/smith/hw6/input/data.csv
```

- Windows can also use backslashes to separate folders.

- **relative path:** does not specify any top-level folder

```
names.dat
```

```
input/kinglear.txt
```

- Assumed to be relative to the *current directory*:

```
file = open("data/readme.txt")
```

If our program is in H:/hw6 ,

open will look for H:/hw6/data/readme.txt

File input question

- We have a file `weather.txt`:

```
16.2  
23.5  
19.1  
7.4  
22.8  
18.5  
-1.8  
14.9
```

- Write a program that prints the change in temperature between each pair of neighboring days.

```
16.2 to 23.5, change = 7.3  
23.5 to 19.1, change = -4.4  
19.1 to 7.4, change = -11.7  
7.4 to 22.8, change = 15.4  
22.8 to 18.5, change = -4.3  
18.5 to -1.8, change = -20.3  
-1.8 to 14.9, change = 16.7
```

File input answer

```
# Displays changes in temperature from data in an input file.
```

```
def main():  
    input = open("weather.txt")  
    lines = input.readlines()  
    prev = float(lines[0])          # fencepost  
  
    for i in range(1, len(lines)):  
        next = float(lines[i])  
        print(prev, "to", next, ", change =", (next - prev))  
        prev = next
```

Gas prices question

- Write a program that reads a file `gasprices.txt`
 - Format: *Belgium \$/gal US \$/gal date ...*

```
8.20 3.81 3/21/11 8.08 3.84 3/28/11 ...
```

- The program should print the average gas price over all data in the file for both countries:

```
Belgium average: 8.3
```

```
USA average: 3.9
```

Multiple tokens on one line

You can use `read()` to read the whole file into a string and the `split` function to break a file apart

- `str.split()` – splits a string on blank space
- `str.split(other_str)` – splits a string on occurrences of the other string

```
>>> f = open("hours.txt")
>>> text = f.read()
'1 2\n45 6\n'

>>> f = text.split()
['1', '2', '45', '6']
```

Looping through a file

- The result of `split` can be used in a `for ... in` loop
- A template for reading files in Python:

```
file = open("filename")
text = file.read()
text = text.split()
for line in text:
    statements
```

Gas prices solution

```
def main():
    file = open("gasprices.txt")
    belgium = 0
    usa = 0
    count = 0
    lines = file.read().split()

    for i in range(0, len(lines), 3):
        belgium += float(lines[i])
        usa += float(lines[i + 1])

    print("Belgium average:", (belgium / count), "$/gal")
    print("USA average:", (usa / count), "$/gal")
```