CS 106
Introduction to Computer Science I

07 / 17 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions / comments?
- File input and output
- Splitting strings into lists of strings by delimeters

Files

- Python’s file input and output
  - open function
    - Returns a <class '_io.TextIOWrapper'>
    - Parameters: file name and an optional mode (defaults to ‘r’ for read)
    - Modes: ‘r’, ‘w’, ‘a’, ‘r+’, ‘w+’, ‘a+’
    - ‘r’ is for opening files for reading
    - ‘w’ is for opening file for writing
    - ‘a’ is for opening file for appending
    - The + ones allow reading and writing

Files

- Methods you can call on the returned variable from open:
  - read (reads the entire file into a string that is returned)
  - readline (reads up to a newline and returns a string)
  - write (writes the parameter to the file)
  - seek (move the file pointer to a certain byte)
  - close (closes the file after reading/writing)

Let’s write some code that reads and writes files

- For loop to read each line of a file

```
for ___ in ___:


# a variable that will be assigned to a line of the
# file goes before in
# the file object goes after the in
# file must be opened prior to the loop
```

Using with

- The with keyword allows usage of files without requiring close to be done.

```
with open(‘data.txt’, ‘r’) as infile:
        # here use infile to read lines as you want
        # once we are out of the with, the file is closed
        # automatically
```

split and re.split

- split method in str
  - mystr = “Hello, World!\nWhat a nice day.”
  - e.g.
  - listofstrs =mystr.split() # splits on whitespace
  - # and discards empty strings
  - list2 = mystr.split(“,”) # splits on comma
  - Return a list of strings
- import re
- re.split function
  - Takes in a pattern (regular expression) and a string to be split
  - Regular expressions are beyond the scope of what I teach in CS106, but I will show you one useful one.

More about dictionaries

- [] vs. get

- Looping through a dictionary e.g.:

```
for word in worddict:
    print(word, worddict[word])
```

More example code

- Let’s start from that projected homerun program and instead of reading the data from the user, let’s get the data from a file.
- Let’s also write code to read an entire file and count up how many works are in it.