CSCI 127: Joy and Beauty of Data

Lecture 10: Files

Reese Pearsall

Summer 2021

https://reesep.github.io/classes/summer2021/127/main.html

Announcements

Lab 6 due **Thursday** @ 11:59 PM

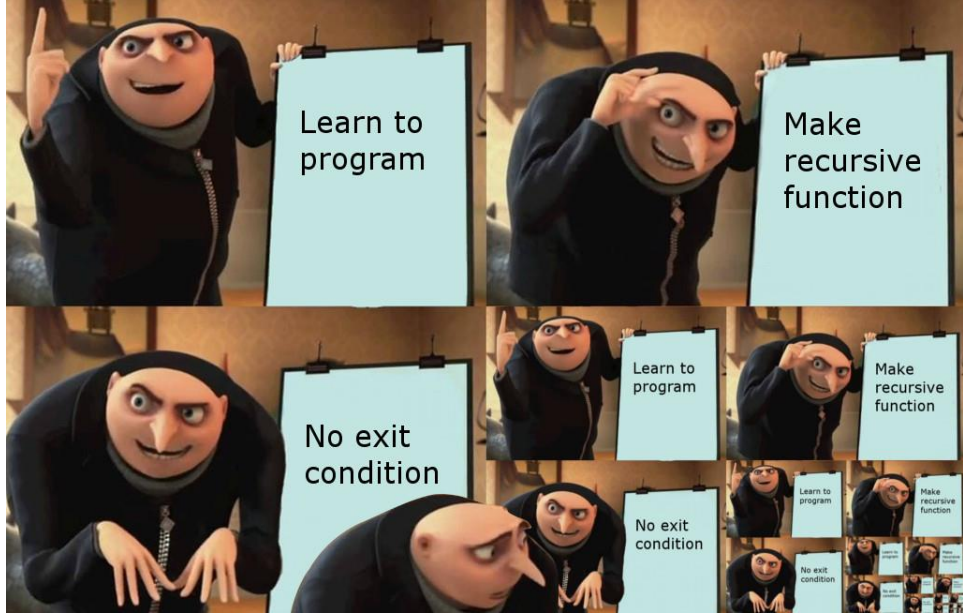
Program 3 due **Monday 6/7** @ 11:59 PM

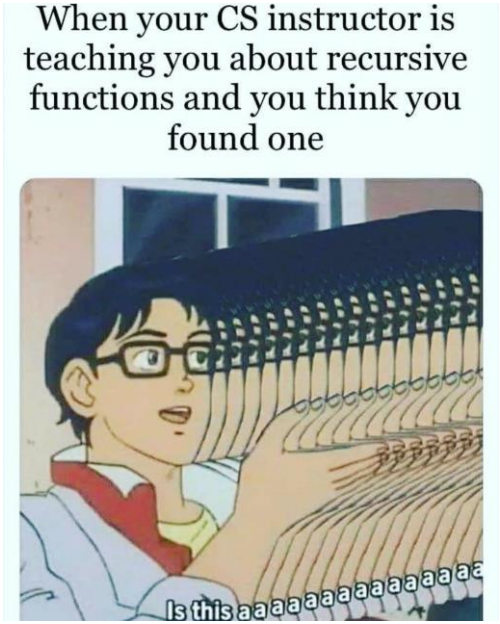
(After today, you will be able to finish both Lab 6 and Program 3)

Currently grading the midterm exam and program 2

Gradebook on D2L has been updated to reflect different grade categories

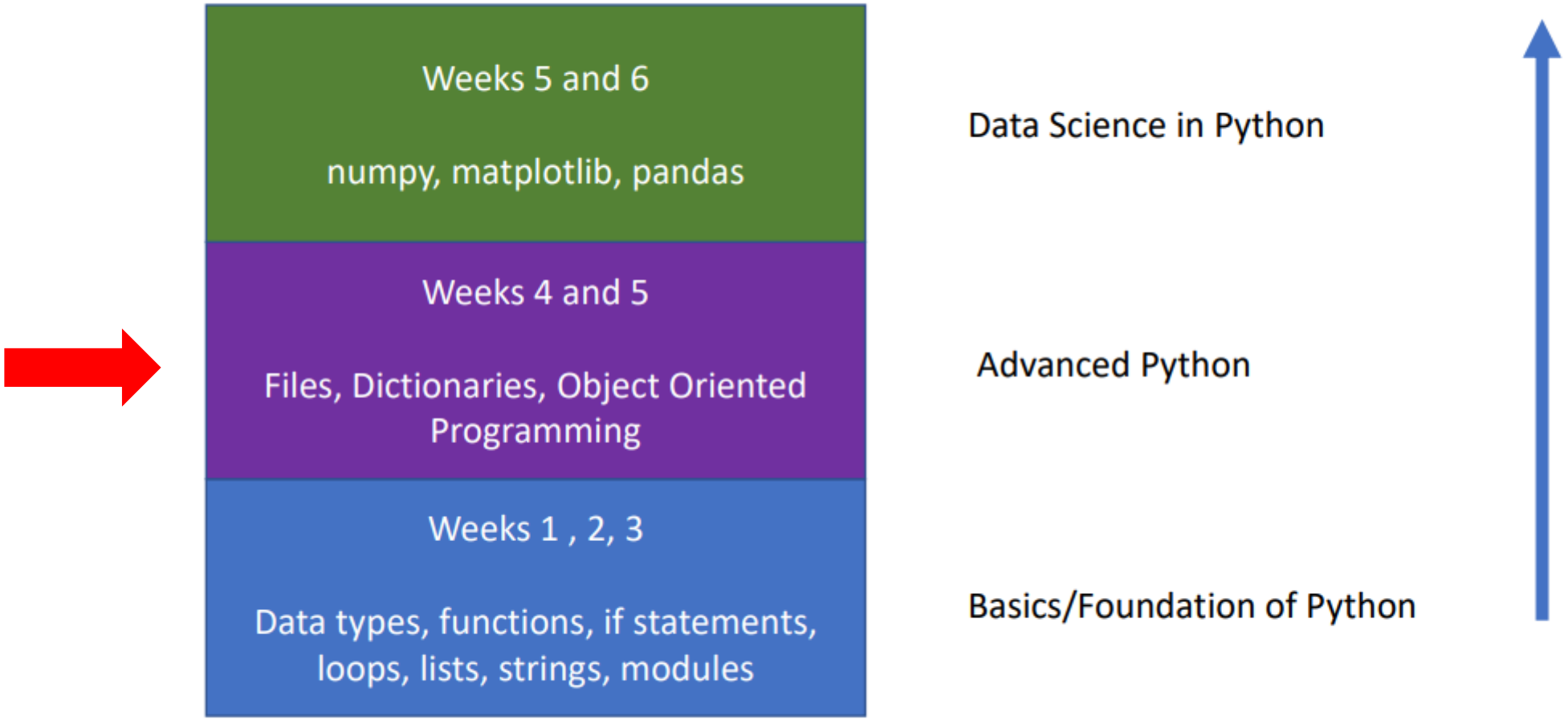
*Remember to sign up for a 1 on 1 time w/ Reese





P.S. Thank you for filling our mid-semester survey ☺


Don´t forget, you can:
• Start late
• Start over
• Be unsure
• Try and fail
And still succeed


Weeks 5 and 6
numpy, matplotlib, pandas
Weeks 4 and 5
Files, Dictionaries, Object Oriented Programming
Weeks 1 , 2, 3
Data types, functions, if statements, loops, lists, strings, modules
Data Science in Python
Advanced Python
Basics/Foundation of Python

Reading from files -- Motivation

So far, we've defined data *within* our program.

If we have a lot of data, then our program can become very convoluted and messy

Sometimes, it is easier to have our data exist is some *external* file that we can read from

The files we will be working with will be *thousands* of lines long

Quick Review - .split()

`list_name = string_name.`**`split`***`(delimiter)`* is how we can divide up a string (`string_name`) based off a character (`delimiter`) and convert it into a list (`list_name`)

Quick Review - .split()

`list_name = string_name.`**`split`***`(delimiter)`* is how we can divide up a string (`string_name`) based off a character (`delimiter`) and convert it into a list (`list_name`)

greeting = “Hello. My name is Reese. Welcome to CSCI 127. I hope you enjoy.”

my_list = greeting.split(“.”)

Quick Review - .split()

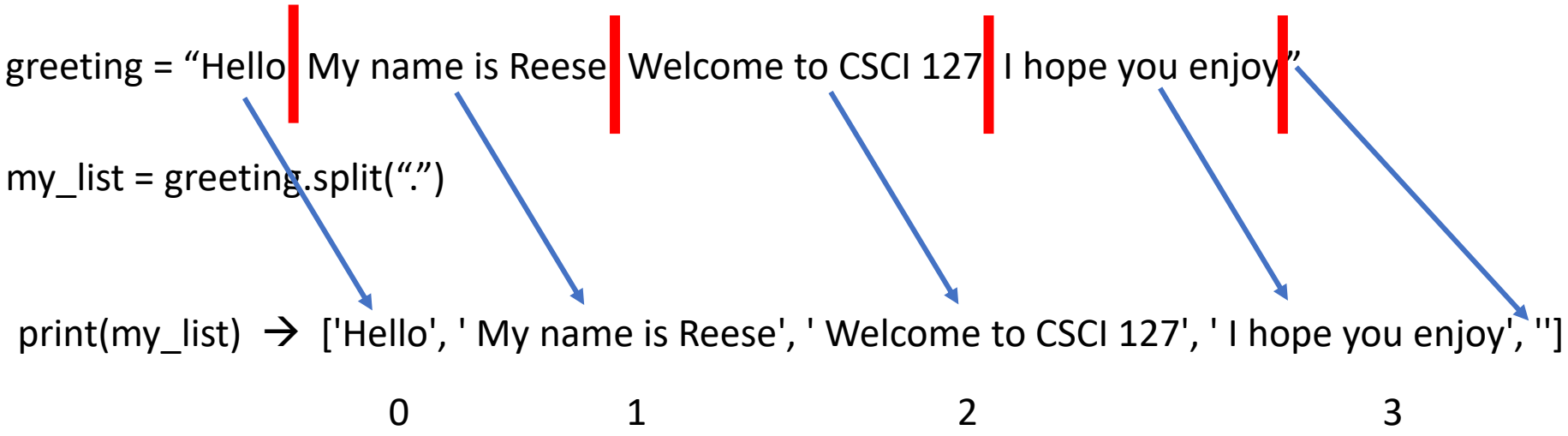
`list_name = string_name.`**`split`**`(`*`delimiter`*`)` is how we can divide up a string (`string_name`) based off a character (`delimiter`) and convert it into a list (`list_name`)

greeting = “Hello | My name is Reese | Welcome to CSCI 127 | I hope you enjoy”

my_list = greeting.split(“.”)

Quick Review - .split()

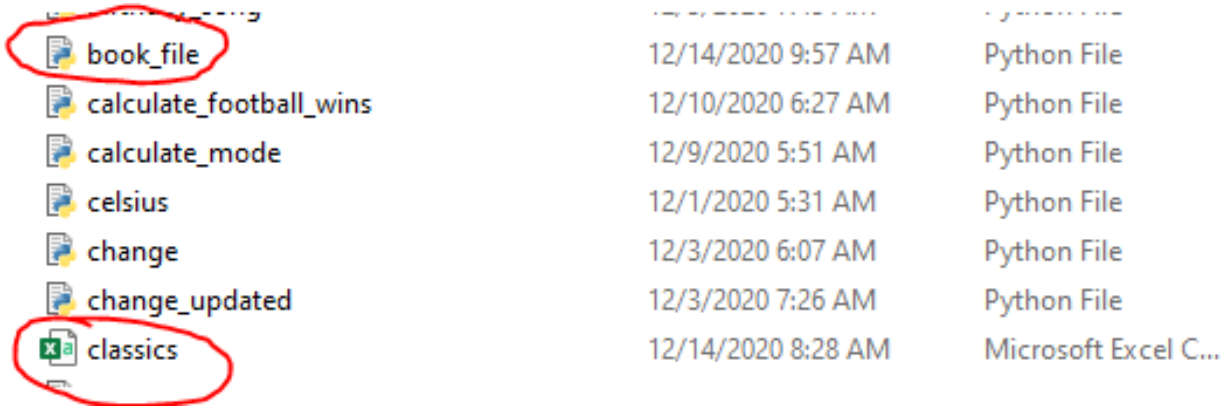
`list_name = string_name.`**`split`**`(delimiter)` is how we can divide up a string (`string_name`) based off a character (`delimiter`) and convert it into a list (`list_name`)



Reading from files

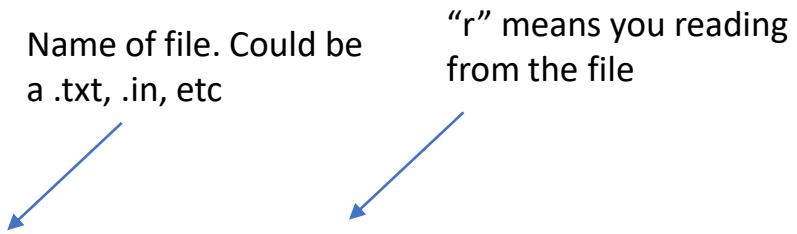
Steps for reading a file

STEP 0: Make sure your .py file and file you are reading from are in the same folder/directory !!



Reading from files

Steps for reading a file




```
file = open("my_file.csv", "r")

file.readline()
for each_line in file:
    row = each_line.split(",")

    ## do stuff here with row


file.close()
```

1. Open the file

2. Discard the column headers (if necessary)

3. Do a for loop through each line in the file

4. Convert each line of the file (string) to a list using **.split()**. If you reading from a .csv file, you will always split on a comma (","), otherwise you will need to figure out what to split on.

Close the file

Using classics.csv (https://corgis-edu.github.io/corgis/csv/)

Write a function that will:

Calculate the book with most sentences

Given an author name, calculate the number of books published by them

Given a year, calculate number of books published that year

Announcements (Wednesday):

Lab 6 due **Thursday** @ 11:59 PM

Program 3 due **Monday 6/7** @ 11:59 PM

Modified Virtual help session/Office Hours today:

1:00 p.m. - 3:45 p.m

Today: More on reading from files, writing to files

PyCharm

A robust, flexible, (much better) text editor used to write python programs

- Includes plenty of quality-of-life improvement and helpful tools

(Professional Python developers don't use IDLE… they use something like PyCharm)

https://www.jetbrains.com/pycharm/

You will have to do a free trial (which will get you through the end of this class)
I believe you can get the pro-version by being a student (not 100% sure though…)

For the sake of consistency, I will continue to use IDLE ☺

COVID Data

Using the covid.csv file, lets write a function (or two) and calculate some cool information!

Writing to Files

Name of file that will be created

"w" for write

```
file = open("out.txt", "w")

for i in range(10):
    file.write("hello x" +str(i))

file.close()
```

1. Open the file

2. Write to file using **.write()**
Note: You cannot use commas like you can with print()
Note2: Everything you write must be a string

3. Close the file
IMPORTANT

Example: Arcade High Score Table

Write a Python Program that will read in a series of 10 arcade game scores along with a player name. These scores are sorted from greatest to least

When the program is run, it should prompt the user for a new score and name. If the score the user inputs is the highest score so far, it should get placed at the very top of the scores. Otherwise, it should get placed in the appropriate spot in the top 10 (if it made it within the top 10)

The program should then write out to the same file it read from with the updated high score list. The high score list should ONLY have the top 10 scores and player names