Practicum 4 - Repetition & Lists

DS2001 - Computer Science Practicum
Fall 2021
September 29/30, 2021
Deadline: October 1st, 2021 at 12:00pm noon Boston time

What we're practicing today:

- while loops (repetition)
- reading data from files (file i/o)
- lists (indexing into them)

Handouts:

- While loops (from DS 2000)
- Lists (from DS 2000)

What to do if you miss practicum this week:

- Fill out the "I'm missing class" form ASAP
- Follow up in office hours with Felix (find office hours links on the course website)
- Come to practicum next week if you are well again. **Do not come to class if you are not feeling well.**

Everyone should create one file for the whole practicum called **p4.py**, and you'll add to it as you go. You'll define **one** `main()` function and add the content from subsequent tasks to it!

Alert! This week we will start making small deductions for submissions that 1) have errors when they run through the autograder and 2) lack answers to group questions (include these as comments in your file!).

Task 0: Write your file comment, your `main()`, and `import` the `random` and `matplotlib` modules

First, write down your name in a comment at the top of your file and a small description of what the file is, like we did last time. This is a good habit to get into for all your programs!

Write down your group members' names in comments below this.

Then, write down a skeleton for your main() function. All the code that we write today will go inside this function.

```
import matplotlib.pyplot as plt # so that we can graph things
import random # so that we can generate random numbers

def main():
    # we'll start off with a print so that we can run our code
    print("Practicum 4!")

# don't forget to call your main function!
main()
```

Task 1: Read the data into a list

Today, we're going to take a look at enrollment trends here at Northeastern. We've gotten this data from NEU's Common Data Set. We've provided you with a few different files with enrollment data here at Northeastern. Each one of these files has one number per line, corresponding to the enrollment levels for the academic years beginning in 2010 - 2020.

Read in the contents of the `enrollment.txt` file to a list of integers, then print out the length of the list and its contents to ensure that you have done this successfully.

Example:

```
Data points: 11
Enrollments: [15905, 16385, 16685, 17107, 17400, 17990, 17894, 18230, 18403, 18313,
18979]
```

Hint: use the `len()` function to find the number of elements in a list.

Group Question: if I gave you an enrollment file that was formatted the same, except it contained the data from 1898 to 2020, what would you need to change about your code as it is currently written?

Task 2: Graphing the enrollments

Now, we'll graph these values over time. Using either a while loop or a for loop, iterate through the contents of your list from Task 1 and plot each point on your graph. Make sure to set the x-values appropriately. The first x-value should be 2010 and the last value should be 2020. The enrollment numbers will be your y-values.

Note: there are many different ways to accomplish this! We would like you to use a loop in your solution though!

When you've finished iterating through your list, make sure to call `plt.show()` to see your graph!

Task 3: Read in the first time first year

Now, read in the data from the file `firstyear_firsttime.txt` to a list of integers, then print out the length of the list and its contents to ensure that you have done this successfully. This data represents the enrollments of first time first year (FTFY) students here at NEU.

Example:

```
Data points: 11
First Time First Year Enrollments: [2836, 3082, 2664, 2861, 2944, 2797, 2676, 3108, 2746,
2995, 3128]
```

When you've finished doing this, modify your graphing code so that you *also* plot the data points for first time first year enrollment on your graph.

Hint: make sure that your ordering is correct here—you may find that it's useful to wait to graph both your enrollment data and your FTFY data until after you've read in both files!

Group Question: what's one thing that you might do to improve the legibility of the graph that you've produced? Do you know how to make this improvement already?

Task 4: Bet on whether or not a random student is a First Year First Time student

We all know that the best way to tell whether or not a student is a FTFY student is to ask them, however, our program doesn't know how to do that! Seems like a great opportunity for us to learn about probability and place some bets!
What you'll do now is ask the user what year they are interested in, then present them with the odds that any one student is a FTFY student, then ask them whether or not they think that the student is a FTFY student, then tell them whether or not they were correct!

Calculate the odds by dividing FTFY enrollment for the target year by total enrollment for that year. This will tell you the probability that a randomly selected student will be a FTFY student.

Example (losing) (user input in **bold underlined**):

```
Year? 2020
Enrollment: 18979
First Time First Year Enrollment: 3128
Odds: 16%
I've chosen a student!
Do you think that the student I've selected is a first time first year student (y/n)? y
Too bad! The student was not a first time first year student
I chose student number: 11599
```

Example (winning) (user input in **bold underlined**):

```
Year? 2018
Enrollment: 18403
First Time First Year Enrollment: 2746
Odds: 15%
I've chosen a student!
Do you think that the student I've selected is a CS student (y/n)? n
You won!
I chose student number: 1282
```

Hint: to know whether or not a student is a FTFY student, imagine that all students have a certain number: 1, 2, 3, …
Next, imagine that we numbered them in order so that all FTFY students received the numbers between [1, number of FTFY students that year] and all other students received bigger numbers. Now, if we randomly select one of these student numbers, we'll know if they were a FTFY student!

Done? Make the following improvements

- Make Task 4 repeat until the user guesses correctly.
- Tasks 2 & 3: Make it so that your graphing code only calls plt.plot() twice (so that it is not in a loop). Use your loop to construct the list of years (x-values), then once it's done, call plt.plot(x_values_list, y_values_list) for enrollments and first time first year enrollments.
- Make the user input case insensitive for Task 4.
- If the user enters anything other than "y" or "n", re-prompt them in Task 4.
- Calculate the average enrollment for the most recent 4 years here at NEU
- Make a graph of the rolling 4-year average enrollment here at NEU. Your first plotted point will be the average from 2010 - 2013, next point from 2011 - 2014, etc.
- Give your graphs titles and labels.

Remember to submit whatever you have working at the end of the allotted time!