Soren Nielsen
10/21/2024
IT FDN 110 B Au 24: Foundations of Programming: Python
Assignment 02

Assignment 02:

Introduction:

This document outlines a python assignment to address the concepts of: using constants, variables, operators, string formatting, and file creation.

The Code:

Assignment02.py

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment02
# Desc: This assignment demonstrates using constants, variables,
#         operators, formatting, and files
# and calculations
# Change Log: (Who, When, What)
#   RRoot,10/21/2024,Created Script
#   Soren Nielsen,10/21/2024, Created Script
# ------------------------------------------------------------------------------------------ #

# Define the Data Constants
COURSE_NAME: str = "Python 100"
COURSE_PRICE: float = 999.98
STATE_TAX: float = .09
##note that round was added to make sure we didnt display percentages of cents
TOTAL_PRICE: float = round(COURSE_PRICE + COURSE_PRICE * STATE_TAX, 2)
FILE_NAME: str = "Enrollments.csv"

# Define the Data Variables
student_first_name: str
student_last_name: str
course_name: str
csv_data: str
file_obj = None

# Get data from the user

student_first_name = input("What is your first name?")
student_last_name = input("What is your Last Name?")

# Present the data to the user
csv_data = f"{student_first_name},{student_last_name},\
{COURSE_NAME},{COURSE_PRICE},{TOTAL_PRICE}"

print(csv_data)

# Process the data to a file

file_obj = open(FILE_NAME, "w")

file_obj.write(csv_data)

#close file
file_obj.close()
print("data recorded in: " + FILE_NAME)
```

Ln: 46 Col: 0

Fig. 1 The Body of the Code in Question

Expected Outcomes:

The expectation is that the code will print a string formatted with commas and an accompanying .csv file named “Enrollments.csv” with the same data written to it.

The Constants

All constant variables are declared at the beginning of the script along with the relevant data type.

```
# Define the Data Constants
COURSE_NAME: str = "Python 100"
COURSE_PRICE: float = 999.98
STATE_TAX: float = .09
##note that round was added to make sure we didnt display percentages of cents
TOTAL_PRICE: float = round(COURSE_PRICE + COURSE_PRICE * STATE_TAX, 2)
FILE_NAME: str = "Enrollments.csv"
```

Fig. 2 Declaration of Constants

Rounding

Since the multiplication in the total price calculation produced a number featuring fractions of cents the round() function was used to make sure TOTAL_PRICE accurately represented currency format.

Data Variables

In the next section the data variables are declared but not set to any values. Declaring variables like this makes the code more readable, though it isn’t strictly necessary.

```
# Define the Data Variables
student_first_name: str
student_last_name: str
course_name: str
csv_data: str
file_obj = None
```

Fig. 3 Declaration of Variables

Data Collection

Using the input() function the user enters data to be set to the appropriate string variables.

```
# Get data from the user

student_first_name = input("What is your first name?")
student_last_name = input("What is your Last Name?")
```

Fig. 4 Collecting User Input

Data Formatting and Presentation

The variables and constants are then formatted together into one csv_data string using brackets and f string.

```
# Present the data to the user
csv_data = f"{student_first_name},{student_last_name},\
{COURSE_NAME},{COURSE_PRICE},{TOTAL_PRICE}"
```

Fig. 5 Formatting the csv_data string

Print Function and the file argument

While the assignment brief was quite specific that the .write method was to be used to affix the string to .csv the first iteration of this assignment used the file argument in the print method which seemed to add the data as well.

Writing the Data to .CSV

A csv files was opened in write mode, the csv_data was written to the file and then the file was closed with a confirmation message.

```
# Process the data to a file

file_obj = open(FILE_NAME, "w")

file_obj.write(csv_data)

#close file
file_obj.close()
print("data recorded in: " + FILE_NAME)
```

Fig. 6 Writing file Data

Testing

The Code Was Confirmed to function in both IDLE and Terminal.

```
Mac-Pro:Week2 lukenielsen$ ls
Assignment02.py _Module02
Mac-Pro:Week2 lukenielsen$ Python3 Assignment02.py
What is your first name?Soren
What is your Last Name?Nielsen
Soren,Nielsen,Python 100,999.98,1089.98
data recorded in: Enrollments.csv
Mac-Pro:Week2 lukenielsen$
```

Fig. 7 Code Running in Terminal

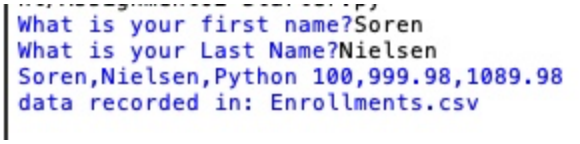

Fig. 8 Code Running in IDLE

Feedback from Assignment 01

In response to feedback from assignment 1 backslashes were used as line breaks to keep the code within the column width established in the header template. Additionally this knowledge document was updated with nested headers to (hopefully) make the document more readable.

Summary

The idea of formatting a single string to have commas and then write to a csv was initially counter intuitive to me. If the commas are data separators in the csv it seems like a strange move to make them part of the variable of the data we want to output rather than say writing each piece of data separately with a comma to separate. But the assignment brief was detailed enough that I was able to create the code in the way it was (hopefully) intended.