Nick Greco
Oct 12, 2024
Foundations of Programming: Python
Assignment01

Basic Input / Output Python Script

Introduction

Python is a high-level, versatile programming language known for its readability and simplicity, making it popular for web development, data analysis, artificial intelligence, and more. In this assignment, I will explain the steps used to create a script that asks the user to enter a student's first and last name, and then prints that the student is registered for a course. This script demonstrates knowledge of input and output functions, data types, string concatenation, the use of variables and constants, and coding best practices.

Preparation

Python Installation in Windows

Python was installed on my Windows-based computer by performing the following steps:

1. Downloaded the latest version of the software from https://www.python.org/downloads/.
2. Ran the installer, ensuring to select "Add Python to PATH" at the bottom of the installer window.
3. Specified the installation location as: *C:\Python*.
4. Verified the installed version, as shown below in Figure 1.

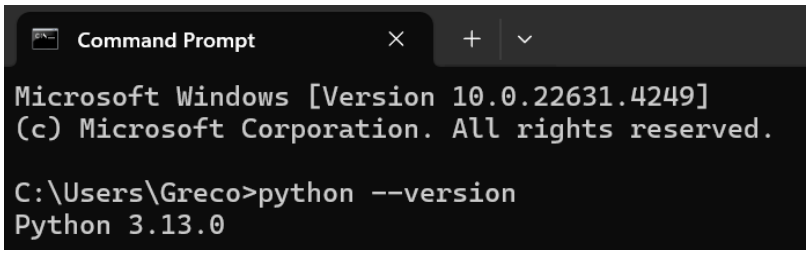

Figure 1: Python Version verified using Command Prompt

Header Template

I approached this assignment with the mindset of establishing templates and tools to reduce overhead on future assignments. After reading Python Enhancement Proposal 8 - Style Guide

for Python Code (Python Software Foundation. (n.d.). *PEP 8 – Style Guide for Python Code*. Retrieved October 13, 2024, from https://peps.python.org/pep-0008/) and the assigned course material, I created a text file header template, shown in Figure 2.

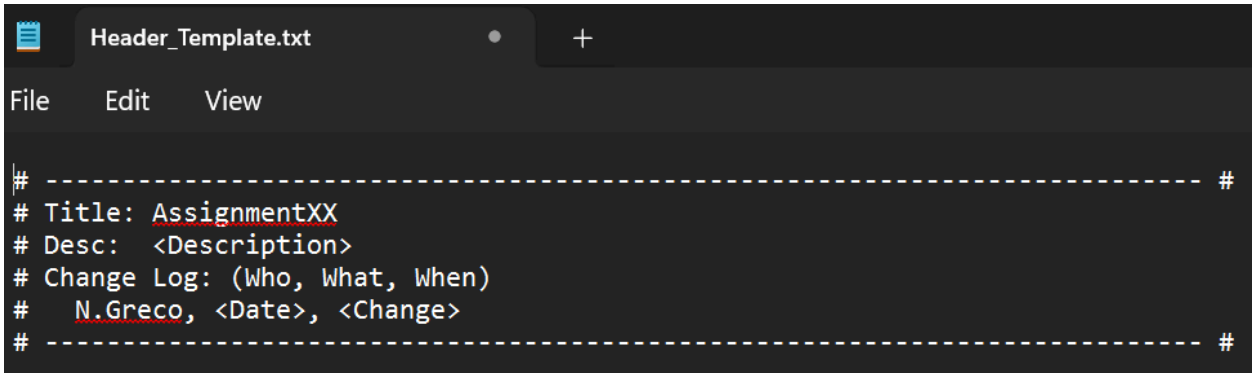

Figure 2: Script Header Template

The template limits line length to 79 characters and incorporates the following required content:

- **Title**: Name of the assignment.
- **Description**: Purpose of the assignment.
- **Change Log**: A record of who modified the code, when it was modified, and what was changed.

Creating the Script

Overview

The script was created using the header template above and is shown in Figure 3 below.

```
# ------------------------------------------------------------------------ #
# Title: Assignment01
# Desc:  This assignment demonstrates using constants, variables, and print()
# Change Log: (Who, What, When)
#   N.Greco, 10/12/2024, Created Script
# ------------------------------------------------------------------------ #

# Setup Code
# Define the data variables.
COURSE_NAME: str = "Python 100"
student_first_name: str = ""
student_last_name: str = ""

# Main Body
# Prompt the user to input the student's first and last names.
student_first_name = input("Please enter the student's first name: ")
student_last_name = input("Please enter the student's last name:  ")
# Display that student is registered for the course.
print(student_first_name + " " + student_last_name + " is registered for " + COURSE_NAME + ".")
print(student_first_name + " " + student_last_name + " is registered for " \
      + COURSE_NAME + ".")
```

Figure 3: "Assignment01" Python Script

This simple code accomplishes the following:

1. **Setup**: It defines a constant variable, “COURSE_NAME” with the value "Python 100", representing the name of a course.
2. **User Input**: It prompts the user to enter a student's first and last names, storing these inputs in the variables “student_first_name” and “student_last_name”.
3. **Output**: It prints a message to the console indicating that the student, whose name was entered, is registered for the specified course. The message is displayed in two formats: one using a simple string concatenation and the other using a line continuation character (\) for better readability.

Concepts Demonstrated

The code demonstrates several key programming concepts:

1. **Variable Declaration**: The use of variables to store data, such as “COURSE_NAME”, “student_first_name”, and “student_last_name”.
2. **Type Annotations**: The use of type hints to indicate the expected data type of the variables.
3. **Input Handling**: The input() function is used to collect user input from the console.
4. **String Concatenation**: Combining multiple strings using the “+” operator to create a complete message.
5. **Output**: The print() function displays messages to the console, allowing for user interaction and feedback.
6. **Line Continuation**: Using the backslash “\” for line continuation in the second print statement to improve code readability, per PEP 8 recommendations.

These concepts form a foundation for understanding basic programming and user interaction in Python.

Running the Script

Using Python IDLE

Python IDLE (Integrated Development and Learning Environment) is a simple IDE that is installed with Python. It provides a user-friendly interface for writing, testing, and debugging code. Figure 4 shows the script run in this environment.

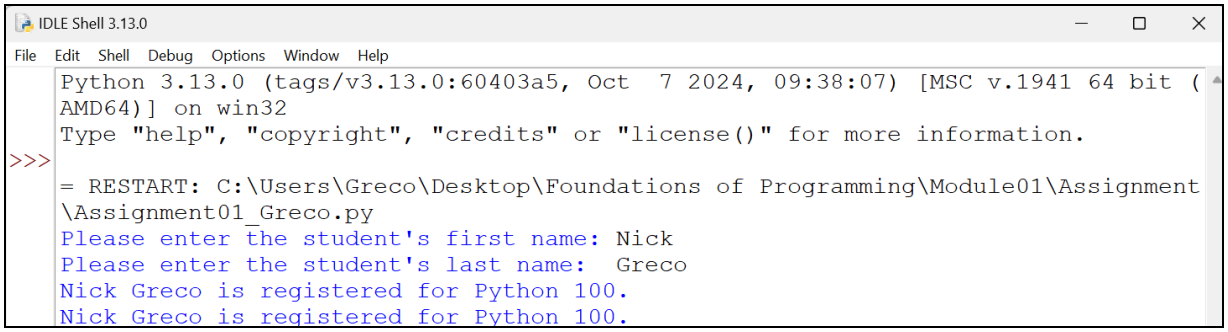

Figure 4: Script Run using IDLE

Using Windows Command Prompt

The Command Prompt is a command-line interpreter available in Windows operating systems. It allows users to execute commands, run scripts, and perform various system tasks without a graphical user interface. Figure 5 shows the process of navigating-to and running the script in this environment.

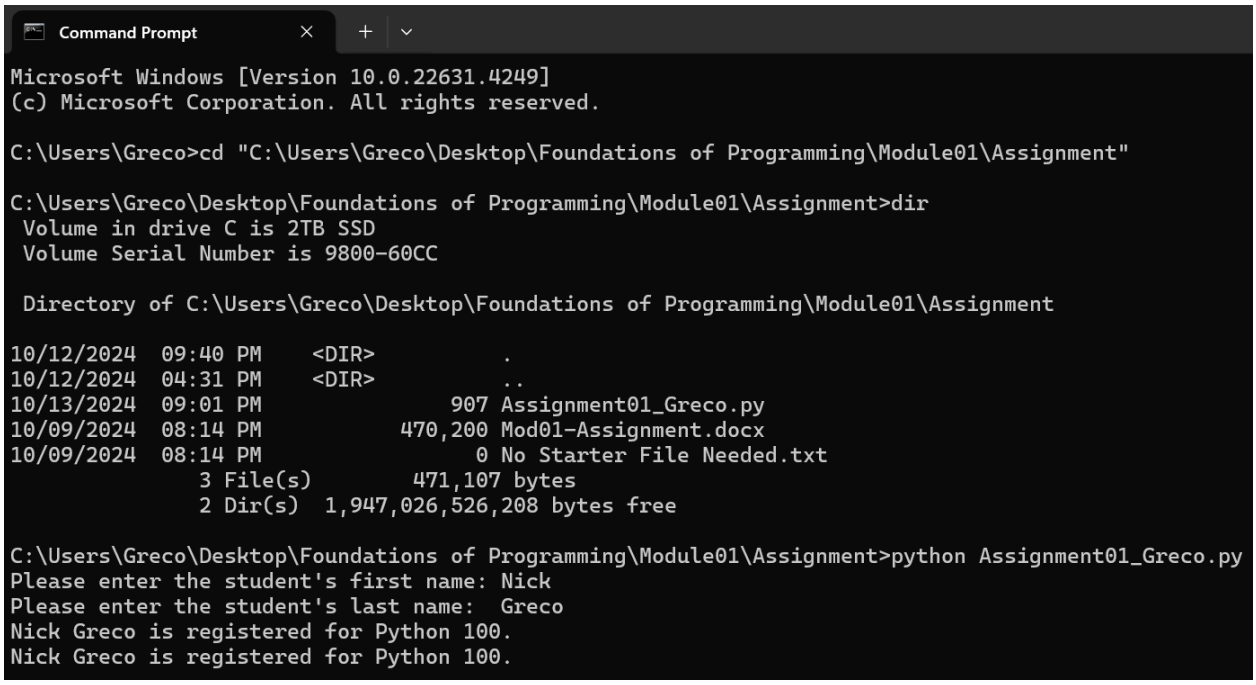

Figure 5: Script Run using Windows Command Prompt

Summary

In this assignment, I have provided a comprehensive overview of creating a simple Python script while highlighting the installation process, coding standards, and execution methods. Overall, this assignment not only reinforced my understanding of basic Python programming concepts but also underscored the significance of following coding standards and practices. The

knowledge gained will be instrumental in completing future assignments and projects, enabling me to write cleaner, more maintainable code.

Citations

OpenAI ChatGPT. (October 2024). https://chatgpt.com/: Aspects of this assignment were informed and created by queries I submitted to ChatGPT.