Nathaniel Smith

April 27, 2020

Foundations of Programming: Python (IT FDN 100 A Sp 20)

Assignment02

Python Input and Output

# Introduction

In this paper I show how to use Python’s built-in functions and operators. I wrote a script that takes user input, processes that input, and then outputs the processed input to the user in a very readable format.

# Built-in Functions

Built-in functions are those functions that are part of Python and do not need to be defined by the programmer. Some examples of Python’s built-in functions are input(), print(), and float().

The input() function is used to get input from a user and could be as simple as requesting the user to press the enter key. The print() function is used to display text on the computer screen. And the float() function is used to change from one data type, such as a string or an integer, to the float data type. Other functions that change the data type are str() and int().

# Operators

Operators are another type of function, but they do not use the same syntax. Instead, they use symbols. Operators can operate on one to three operands.

Figure 1 below, taken from the course textbook, shows some of Python’s mathematical operators (Michael Dawson, Python Programming for the Absolute Beginner, Third Edition, page 29).

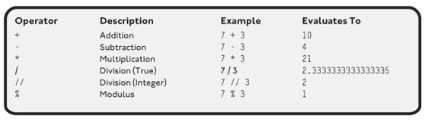


Figure 1. Mathematical Operators

The addition operator, +, is an overloaded operator because as shown in Figure 1 it can add two numbers, and it can join two strings by the operation known as concatenation. Another very useful operator is the assignment operator, =, which assigns values to variables.

# My Basic Math Python Script

I wrote a script that performs basic math on input provided by the user. I used the input() function with the assignment operator, =, to prompt and record the input from the user. The script is expecting the user to enter a number, and then enter another number. Since I used the input() function to get the data from the user, Python stored that data as a string in the two variables I created, strNum1 and strNum2.

My next step was to convert the variables from the string data type to the float data type using the float() function. Once the string variables were converted to float data type I used the assignment operator in conjunction with a mathematical operator to assign a sum with variable name fltSum, difference with variable name fltDiff, product with variable name fltProd, and quotient with variable name fltQuot using the +, -, \*, and / operators respectively.

In the final section of code I used the print() function four times to display the basic math results to the user in a readable format. The print() function is capable of printing the text I wrote in single quotes and to print the value of the variable written as the variable name without quotes.



Figure 2. BasicMath.py Script as Shown from PyCharm

# Run BasicMath.py from Windows Command Prompt

I launched the command prompt by typing “cmd” into the Windows search. I copied the path to the BasicMath.py file and pasted the path into the command prompt and pressed enter. Because the file was saved with the extension .py, and the .py filetype is associated to Python in Window’s system settings, the file was ran by Python automatically just by typing the path for the file into the command prompt. My BasicMath.py Python script is shown running from the command prompt in Figure 3.

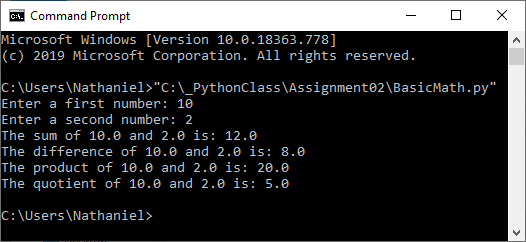


Figure 3. Running BasicMath.py in Windows Command Prompt

# Summary

Using Python’s built-in functions and operators I was able to write a script that takes input from a user, converts it from string data type to float data type, process the data using operators, and display the results to the user. In summary, the script is a calculator that prints the sum, difference, product and quotient of two numbers provided by a user.