Andrei Arevalo

January 22nd, 2021

Foundations of Programming, Python

Assignment02

The 2-Number Calculator

# **Introduction**

In this assignment we are tasked with asking for user input, storing that input as a specific data type, performing mathematical operations on that input, and displaying all back to the user. This assignment is meant to build on the “input” and “print” functions used last week with mathematical operators and data types. Concatenation was also to be used to combine input with pre-defined text to create a messages back to the user. This program is useful in creating small, purposeful calculators that take in user input and process it.

# **Key Questions and Definitions**

## **Q: How does a computer use data?**

Computers can use data in mainly one of two ways; either stored on a drive or loaded into memory. The main difference between the two types of memory is that when data is stored on a drive, it can be recalled after the program ends or the computer turns off. An example of this would be a PowerPoint file that is saved to the desktop and then reopened. Data stored in memory is only temporarily, and the data “evaporates” after the data is used for its purpose. An example of this is the calculator app on the computer; once the calculation is performed, the data is lost.

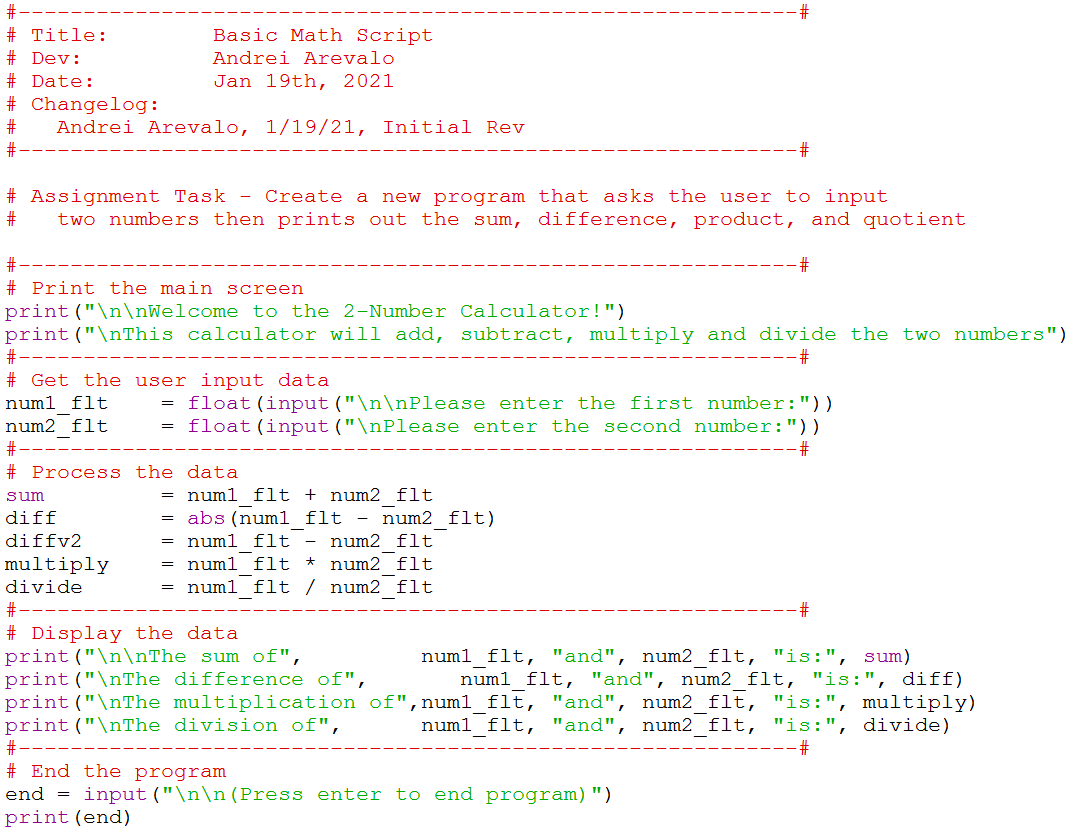
## **Important Sequences and Definitions**

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Description** | **Example** |
| String | Sequence of Unicode letters | Example = “VariableX” |
| Int | 32-bit signed integral type | Value = 2 |
| Float | Single-precision floating point | Value = 2.22 |
| bool | Boolean type (either true or false) | Value1 = True; Value2 = False |

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Addition | 7 + 3 = 10 |
| - | Subtraction | 7 – 3 = 4 |
| \* | Multiplication | 7 \* 3 = 21 |
| / | Division (True) | 7 / 3 = 2.3333 |
| // | Division (Integer) | 7 // 3 = 2 |
| % | Modulus | 7 % 3 = 1 |

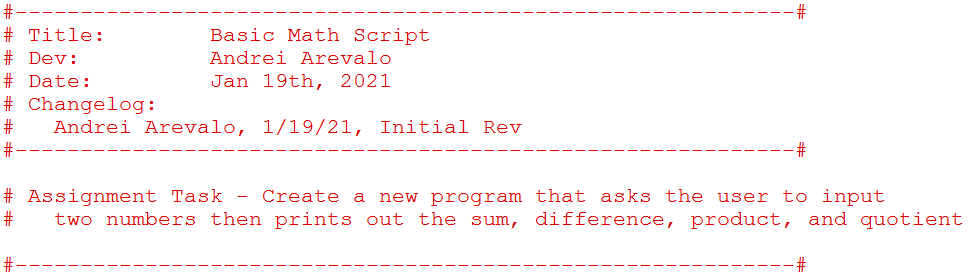
# **Creating the Script**

1. The main script is composed of 4 major components; printing the main screen, getting the user input data, processing the user input data, and displaying that data back to the user. (figure 1)



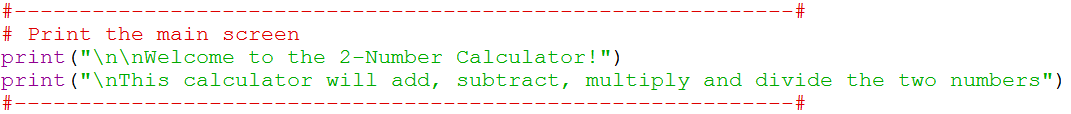
***Figure 1: The main script***

1. At the top of the script is the title block, which shows information about the script and keeps track of revision history in case the file needs to be revised in the future. The author and date are also included in this title block. In addition below the title block is a summary of the assignment with the task outlined. (Figure 2)



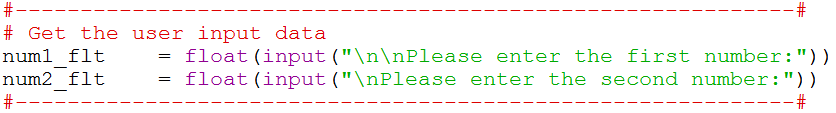
***Figure 2: Title block of the script***

1. The next part of the script uses the “print” function to show the user what program they are using. It also lets the user know how the data is going to be processed. (Figure 3)

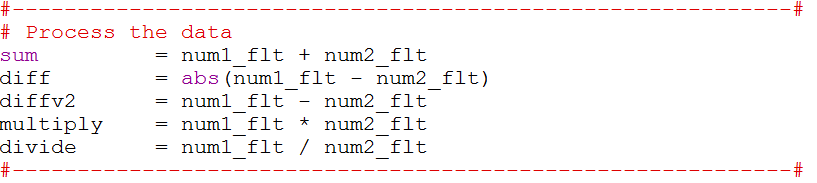


***Figure 3: Main screen of program***

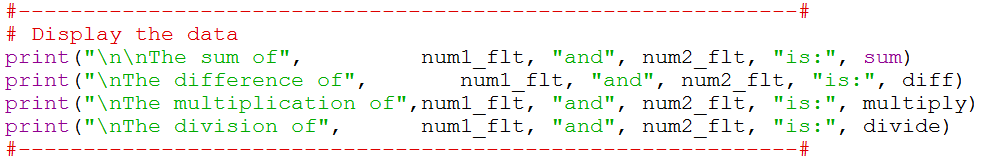
1. The next part of the script uses the “input” function to ask the user to input the two numbers. This is nested within the “float” function, which turns the input of the user into a floating point number. A reasonable substitute would be to use “int” which would turn the input into an integer. (Figure 4)

***Figure 4: Collecting data from the user***

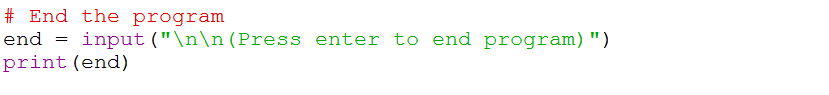
1. The next part of the script several operators to perform math calculations. For example, the addition operator is used to add the two numbers together. In addition, there are two different ways to show the difference between the first and second number. In one way you can subtract the second number from the first number; you can also use the absolute value of the subtraction to get the magnitude of the difference. Both are written into the code. (Figure 5)

***Figure 5: Processing the data***

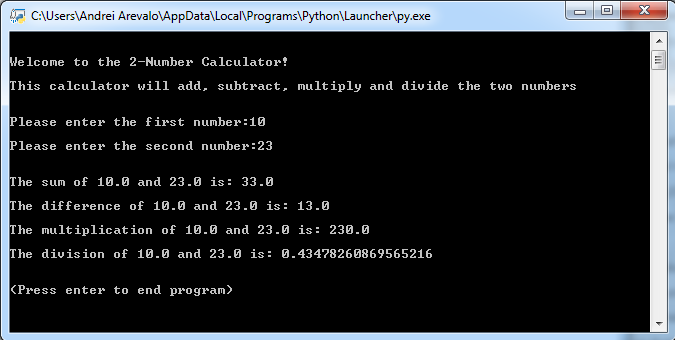
1. The next part of the script uses the “print” function to show the results of the processing. Concatenation is used to link text to the output numbers, and this is done through the use of commas nested within the print function. (Figure 6)

***Figure 6: Printing the data***

1. The last part of the script uses “input” to ask the user to end the program. (Figure 7)

 ***Figure 7: File Path for Python Class***

1. Running the program from the Python Terminal shows the program running outside of the IDLE. (Figure 8)



***Figure 8: Python terminal output of program***

# **Summary**

In this assignment we are tasked with asking for user input, storing that input as a specific data type, performing mathematical operations on that input, and displaying all back to the user. This program built on the “input” and “print” function showcased last week with mathematical operators and data types. Concatenation was also used to combine input with pre-defined text to create a message back to the user. This assignment was a good example of taking and processing user data.