User Input

28 November 2024

20:25

You can use the "input" function for a user to interact with the program:



Result: (User types 7)

A screen shot of a computer

Description automatically generated

To utilize this we can put it in a variable

A screen shot of a computer program

Description automatically generated

\*note the the user's input is always treated as a string

To resolve an issue (for example can use ">" between an int and str) we can make it a float:

A screen shot of a computer program

Description automatically generated

Int() would also work, but for this example it is bad because the user could input a decimal and get an error

String Formatting

30 November 2024

13:59

The message variable below shows string formatting using %s and % operators:

A screen shot of a computer

Description automatically generated

This is also an option for Python 3.6 +

A screen shot of a computer code

Description automatically generated

String Formatting with Multiple Variables

30 November 2024

14:06

To use string formatting with multiple variables, you must call both variables and double the formatting operators:

A screenshot of a computer screen

Description automatically generated

Result:

A computer screen with text

Description automatically generated

The same works for the newer method (just call both variables)

A screen shot of a computer code

Description automatically generated

**More String Formatting**

30 November 2024

14:12

There is also another way to format strings using the *"{}".format(variable)* form. Here is an example:

* 1. name = "John"
  2. surname = "Smith"
  4. message = "Your name is {}. Your surname is {}".format(name, surname)
  5. print(message)

Output: *Your name is John. Your surname is Smith*