02 Prepare: Preparation Material

Overview

In this lesson, you will build on the basic input/output ideas from your programs in the last lesson, and learn to store more information and display it using better formatting. While not all programs use the console input and output approach that we have taken thus far (where people type the answers and see the results in text), almost all programs make use of many string variables and combine and format them in one form or another.

Preparation Material

I. VARIABLES

One of the things you learned in the previous lesson was how to store data in a variable. A variable is like a name that we attach to that data, so that later we can refer to it when we need it.

In your programs this week, you are going to start to see many variables used simultaneously in the same program. This is very common, and doesn't cause any problem for the computer. As long as you are consistent and always use the same name for the same data, you won't have any problems.

Because we will have to keep track of more and more variables, it becomes increasingly important to choose good names for them. For example, while you may remember what `x` means now, in a few weeks, months, or years, you might forget. On the other hand a variable name like `color` or even `favorite_color` is much more descriptive and will help you and others better understand your code.

Hint from Instructor:

You can't use spaces in variables, so if you want a long variable name with multiple words, the Python style is to use underscore characters between the words, such as: `this_is_a_very_long_variable_name` .

Other style guides, especially for other languages may use different approaches, such as "camel case" where each subsequent word is capitalized, such as `thisIsAnotherLongVariable`, but in this class, you should stick with the underscore, or "snake case," approach.

II. COMMENTS

Comments are a way for you to include notes in your code. They don't affect the program in any way, but they make it easier for someone to understand the code when they look at it later. To add a comment to your code, include the `#` sign before the text that you want to be a comment.

Please watch the following videos for more information about comments:

» Comments (3 mins)

» Demo: Comments (3 mins)

III. COMBINING AND FORMATTING STRINGS

As you learned in the previous lesson, "Strings" are variables that are a sequence of characters (for example, letters, numbers, spaces, symbols, etc.). Please watch the following videos that demonstrate how to combine, format, and display strings in different ways:

» String Concepts (6 mins)

» Demo: Strings (4 mins)

» Formatting Strings (4 mins)

» Demo: Formatting Strings (4 mins)

As shown in these videos, some helpful string functions available in Python are:

Code	Result
`words = "the cat IN THE hat"`	`the cat IN THE hat`
`words.capitalize()`	`The cat in the hat`
`words.title()`	`The Cat In The Hat`
`words.upper()`	`THE CAT IN THE HAT`
`words.lower()`	`the cat in the hat`
`words.count("t")`	`3`
`words.lower().count("t")`	`4`

Notice that `words.count("t")` resulted in a 3, because it did not count the capital "T" in the sentence. On the other hand, `words.lower().count("t")` resulted in a 4, because it first converted everything to lowercase, and then counted them, so when it counted the t's, the capital T in that sentence was first converted to a lowercase t, and then it was counted.