CITS1401 Computational Thinking with Python (2022S2)

Information

Flag question

Information text

We will often need to store values of calculations or input, and we can do this using variable. Think of variables as a storage box with a label; you can store one type of data in there (a number, a word, etc) and label the box. For example, I could store an age of '18' in the variable 'age'. The data in variables can also be changed or replaced. The names we give variables are important too as they need to be easily read by others, to have good Python programming style.

As in the previous section, type the line or lines given the left column, inspect the output, read the notes, and figure out what's happening!

Input Line	**Notes**
`temp - 212`	This is what we call an expression, not a statement (expressions will evaluate to some value, while statement is a complete line of code that performs some action). It will carry out the operation it is set assigned to do (which in this case, subtract 212 from the variable temp).
`temp - 20`	You just ensure that the variable used must be declared (i.e. assigned a value) before using. If a variable name appears in an expression, then the value of the variable is used in its place.
`temp`	Just calling the variable will display the value it currently holds in shell. It will only work in the shell
`temp212 = 212`	Python allows you to use digits in variable names...
`212temp = 212`	... but not at the start!
`degrees_f = 212`	You can also use the underscore character (type as Shift/-)...
`degrees_f = 212`	... but that is the only non-alphanumeric character allowed in Python for variable names.
`flubber11_flonk7 = 212`	For us to understand what the code is doing later, it is a good idea to give it a meaningful name. Computers don't mind whatever they are called as they have much better memory to remember what they hold, but for us humans it is trickier to remember what the variables are used for, especially in a group coding exercises and when you have to review your code later on.
`d_f = 212`	What do you think this variable stores? To have good Python style variables should be at least 3 characters long but are often much longer using sensible words. The example above of degrees_f is a great example, as it's both a valid Python variable and also easy to read.
`degrees - 32`	You can't use variables that don't exist, i.e., variables that you haven't already used on the left-hand side of an assignment statement (e.g. do "degree = 20" first before using it in an expression).
`Temp - 32`	Names are case sensitive! (Note the capital T at the start)
`freezing = 32`	This should be all good.

`factor = 5 / 9`	The right-hand side of an assignment statement (the bit to the right of the equal sign) is the one that gets calculated and then stored to the variable on the left-hand side.
`(degrees_f - freezing) * factor`	Building expressions from variables is at the heart of programming. Once they are assigned values, you can "call" them in any expressions to use their stored values.
`degrees_f = 32`	Variables are called variables because the value associated with them can change (vary) through re-assignment.
`(degrees_f - freezing) * factor`	[Use the up-arrow key to retrieve the earlier expression.] Now we convert 32 degrees Fahrenheit to Celsius.
`degrees_f = degrees_f + 1`	That equals sign doesn't mean "equality" in a mathematical sense! It means "evaluate the expression on the right-hand side, then associate the variable on the left-hand side with that value. So what will the value of degrees_f be now?
`degrees_f`	Were you right?

Information

Flag question

Information text

A summary about choosing variable names ("identifiers")

When choosing names for your variables there are three different aspects to keep in mind.

1. **Syntax.** The Python language, like all programming languages, has set of rules that define what statements are valid. The set of rules is called the *syntax* of the language. (You can call it grammer of Enlish language). In English, the statement "This man eats meat" is grammatically correct while the statement "Eats this man meat" is incorrect, because the word order breaks the grammer rules of English. [As a slight aside, the statement "Meat eats this man" is syntactically correct but *semantically* invalid, because it doesn't make sense.]

   The syntax rules of Python define what constitutes a valid identifier for a variable. Specifically a valid identifier is any letter or underscore followed by a sequence of zero or more letters, digits or underscores. As shown above, if you type statements like **`1x = 7`** into the Python shell, the Python interpreter will give an error message because `1x` is an invalid identifier according to the Python syntax.

2. **Style.** If you type the statement `q9 = 7` into the Python shell it will work fine but the identifier `q9` is (in most contexts) meaningless to the human reader. `q9` is a *stylistically poor* identifier. Defining precisely what is or isn't good style is generally impossible - it's just too subjective. However, in CITS1401 there are style rules that help you to write better code and these are usually enforced in your projects.

In general variable identifiers must satisfy the following two rules to be considered good in style

- They must be at least three characters long. [There are some exceptions to this, which will be introduce later.]
- The identifiers should be meaningful. Normally in Python, no upper-case characters are preferred, so `circle_area<` is preferred over `CircleArea`.
- **Readability.** A *good* identifier not only satisfies both the syntax and style checks but also must be readable by other humans (and by you when you try to read your own code!). This is somewhat subjective and unfortunately can't (yet) be checked by the computer. But in CITS1401 you are required to think about readability whenever you define yourself a new variable. Names like `age_in_years` and `reflection_coefficient` are obviously meaningful while names like `x123_yzq`, `thing` and `blah` are equally obviously not, even though they satisfy the variable naming rules.

Question 6

Not complete
Marked out of 1.00
Flag question

Question text

Select all the variable names that are syntactically correct from the list below.
Note: Here, the style of the variable names will not be checked (i.e. having a meaningful name), but only checking that Python will not give an error.

Select one or more:

☑ a1b2c3

☑ abc123

☐ Welcome Home

☑ WelcomeHome

☑ this_is_a_very_long_variable_name

☑ G_B_D

~~☑ 123abc~~

☑

a1

☑

some_variable

☑

hurrah

☐

this_is_a_very_long_variable_name!

Check

Question 7

Not complete
Marked out of 1.00
Flag question

Question text

Which of the following are syntactically correct Python assignment statements? Here, you can assume all the variables are already defined. Select ALL that are correct.

Select one or more:
☐

car_speed + 5 = current_speed

☑

cooking_time = 40 + (15 / 7)

☐

3.1415 = pi

☑

average = total / numbers

☑

cooking_time = total_time / 2

☑

cooking_time = cooking_time - 50

Check

Question 8

Not complete
Marked out of 1.00
Flag question

Question text

Look at the Python expressions below. What is the value of 'orange' when all those expressions are executed in that order?

You are welcome to use Python to check your answer, but try to solve it without using Python first.

```
apple = 15

pineapple = apple * 2 + 1

orange = apple + pineapple

orange = orange - (apple * 2)
```

Answer: 16
Check
Previous page Next page

Network connection lost. (Autosave failed).

Make a note of any responses entered on this page in the last few minutes, then try to re-connect.

Once connection has been re-established, your responses should be saved and this message will disappear.

◀ Lab 00. Introduction
Jump to... Jump to...
Lab 01. Python basics (for late submission or approved special considerations) ▶
Skip <span id="mod_quiz_navblock_title">Quiz navigation</span>

Quiz navigation

Information i Information i Question 1 Question 2 Question 3 Question 4 Question 5 Information i This page Information i This page Question 6 This page Question 7 This page Question 8 This page Question 9 Question 10 Question 11 Question 12 Question 13 Question 14
Finish attempt ...
Time left
You are logged in as Zehua Zhu (Log out)
CITS1401 2022S2