Python Basics 001

**1. In the below elements which of them are values or an expression? eg:- values can be integer or string and expressions will be mathematical operators.**

|  |  |
| --- | --- |
| \* | EXPRESSION |
| 'hello' | VALUE |
| -87.8 | VALUE |
| - | EXPRESSION |
| / | EXPRESSION |
|  | EXPRESSION |
| 6 | VALUE |

**2. What is the difference between string and variable?**

Variables are symbols that you can use to store data in a program. They are a placeholder or empty box within which you can fill with some data or value.

Strings are data, so we can use them to fill up a variable. Thus, string can be stored in the placeholder created by a variable.

Declaring strings as variables can make it easier for us to write code and manipulate strings throughout our Python programs.

**3. Describe three different data types.**

* Numbers
  + Integers, floating point numbers and complex numbers are all considered as Numbers data type in Python
  + They are defined as *int*, *float* and *complex* classes in Python respectively.
* Strings
  + String is an array or sequence of Unicode characters.
  + It is represented by *str* class.
  + Characters enclosed within single quotes or double quotes represents a string. Also, multi-line strings can be represented using triple quotes, ''' or """.
  + Strings, however, are immutable.
* Booleans
  + The Boolean data type has one of the two built-in values, True or False.
  + But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false.
    - Thus the numeric value ‘0’ results in False while all other numeric values results in True
    - Also any string will result in True while an empty string will result in False
  + It is represented by the *bool* class.

**4. What is an expression made up of? What do all expressions do?**

An expression is a combination of operators and operands (identifiers & literals) that is interpreted to produce some other value. In any programming language, an expression is evaluated as per the precedence of its operators.

The following are examples of expressions:

a = 1 + 2

b = 5

c = a \* b

The purpose of expressions is to represent value. They do not perform anything but simply represent value.

**5. This assignment statements, like spam = 10. What is the difference between an expression and a statement?**

An Expression always evaluates to a value. However a statement executes something, like creating a variable or displaying a value, it only does whatever the statement says.

Also, for any code, if it can assign a value to a variable then it is a expression. If not then it a statement. For example *x = print(‘abc’)* is an expression. While in Python 2, *Print ‘abc’* is a statement.

Every expression is a statement but every statement is not an expression.

**6. After running the following code, what does the variable bacon contain?**

bacon = 22

bacon + 1

Answer: The variable bacon contains 22. This is because bacon + 1 does not result in incrementing the value of bacon by 1 as there is no assignment.

Graphical user interface, text, application

Description automatically generated

**7. What should the values of the following two terms be?**

'spam' + 'spamspam'

'spam' \* 3

Answer: spamspamspam

Text

Description automatically generated

**8. Why is eggs a valid variable name while 100 is invalid?**

A variable name is not allowed to begin with an integer. Since 100 begins with an integer you will get an invalid syntax error. A variable name is allowed to begin with a alphabet or an underscore ‘\_’

9. What three functions can be used to get the integer, floating-point number, or string version of a value?

Integer: *int()*

Float: *float()*

String: *str()*

**10. Why does this expression cause an error? How can you fix it?**

'I have eaten ' + 99 + ' burritos.'

The expression results in a type error. While Python supports Strong Typing and UpCasting, it will fail to combine a string with an integer. We can fix the error by converting the integer 99 to a string as follows:

A picture containing graphical user interface

Description automatically generated