**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 - Multiplication Operator

'hello' : Value - String

-87.8 : Value - Integer

- : Expression - Subtraction Operator

/ : Expression - Division Operator

* : Expression - Sum Operator

6 : Value - Integer

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

A **string** is a datatype often a character value wrapped in either single or double quotes. Multiline string is represented with triple quotes.

E.g : “Welcome Harish, You’ve enrolled into iNeuron’s Full stack Data science course”

“98”, “True”

A string can be any of the other datatypes only that they should be wrapped in the quotes as shown in the above example.

A **variable** in other hand as the name suggests, a varying object defined in the programming tool as a placeholder to store values which could be keep varying as per the requirement.

Variable can hold any data type .

E.g : name = “Harish”, age = 26, data\_scientist = TRUE

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

There are several datatypes in python which can store different types of data

* Numerical data can be defined using class ‘int’ (integers) or ‘float’ (floating point numbers) and also Complex Numbers can be defined using class ‘complex’
* Boolean data in terms of TRUE or FALSE (logical conclusions)
* Character based data can be defined using class ‘string’

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

A piece of code that evaluates something is known as an expression.  It could be a simple operation like adding two variables.

Expressions are nothing but the representation of a value.

Expressions are usually made of values (operand) and an operator. The operators could be logical or numerical. An expression could simply be a constant number with no operations performed and no operator or variable. It is simply something in the code that is evaluated.

E.g:

a=2

b=5

c=a+b

print(c) — prints 7 on the screen.

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

An expression can be anything, as long as it is valid and understandable by Python. While, statement is the entire line in which some evaluation occurs and this evaluation is assigned to a variable. However, assignment is not required for a line of code to be a statement.

In the example given

spam=10

The value 10 is an expression in and of itself, but the entire line that initializes a variable with the name "spam" and assigns it the value 10 is a statement. An expression is any piece of code that is evaluated.

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

**bacon = 22**

**bacon + 1**

bacon still would be 22.

The first line, bacon =22, is an assignment statement that assigns the variable bacon the value of the integer expression 22. The second statement, "bacon+1," is simply an expression with no assignment; thus, the original variable is unaffected.

If, on the other hand, the statement had been: bacon+=1,

The variable's value is updated when the augmented operator is used. The new value is going to be 23.

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

'spam' + 'spamspam'

'spam' \* 3

Both expressions would be evaluated to produce the same result : ‘spamspamspam’

First case : String concatenation simply binds the multiple strings together

Second case : When an integer is multiplied by a string ('spam,' the string is replicated the number of times multiplied by the integer.

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

A variable is an identifier, a placeholder for which a value can be assigned. A Python variable is a memory location reserved for storing values. The variable can be given alphabetical or alphanumeric names. Variable names cannot be made up of numerical values.

Python interprets numerical values like 100 as expressions and thus cannot be used as variable names.

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

int(value) converts the value to integer

float(value) converts the value to a floating point number

str(value) converts the value to a string

However, characters cannot be converted to numerical data types using int() or float(). But, a number can be converted to a string of characters using str().

Example:

int(‘a’) is invalid

str(12) is valid, it evaluates as ‘12’

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

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

Concatenation is represented by the addition operator with strings. Strings can only be concatenated with other strings. Python evaluates 99 as a numerical value in the preceding code because quotes (‘ ’) is not available.

Solution 1: Add ‘’ to 99

'I have eaten ' + **‘99’** + ' burritos.'

Solution 2: Use str() on 99

'I have eaten ' + **str(99)** + ' burritos.’

Both will evaluate and generate the same: ‘I have eaten 99 burritos’