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?

String is a data type that represents a sequence of characters within quotation marks (either single or double quotes). Strings are immutable (i.e, cannot be changed once created).

whereas,

Variable is a name that represents a value stored in the memory. Variables can be assigned different values throughout the program.

String example:

name = "abc"

print(name)

Output : abc

Variable example:

age = 25

print(age)

Output : 25

3. Describe three different data types.

Three different data types are described below:

Integer: The integer data type represents whole numbers without any fraction or decimal parts. It can be positive or negative.

String: The string data type is used to represent a sequence of characters. Strings are enclosed within quotation marks. For example, "Hello World"

List: A list represents an ordered collection of items. Lists can contain elements of different data types such as integers, strings, etc. For example: [1, 2, 3], ["grapes", "banana", "apple"]

There are many more data types like float, boolean, tuple, dictionary, etc.

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

An expression is a combination of values, variables and operators that evaluates to a result. It can be a single value or a complex combination of multiple operations.

They are used to perform computations and manipulate data. They are evaluated by the Python interpreter, which processes the expression and produces a single resulting value.

For example:

x = 5

y = 2

z = x + y \* 3

print(z)

Output – 11

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

Expressions: They produce a value when evaluated. They are used to calculate values, generate results, or provide arguments to functions. For example, 2 + 3 evaluates to 5.

Statements: They do not produce a value. Instead, they perform an action or modify the program's state. Statements define the structure, behavior, and control flow of a program. For example: an assignment statement- spam = 10 assigns the value 10 to the variable spam.

6. After running the following code, what does the variable bacon contain? bacon = 22 bacon + 1

bacon will be the same value 22. Here, In second line, bacon + 1 is an expression that adds 1 to the value of bacon. However, the result of this expression is not assigned to any variable or printed. Therefore, it does not affect the value stored in the bacon variable.

7. What should the values of the following two terms be? 'spam' + 'spamspam' 'spam' \* 3

'spam' + 'spamspam' = spamspamspam

'spam' \* 3 = spamspamspam

Both the cases will produce the same result 'spamspamspam'

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

In Python, variable names must follow certain rules and conventions.

Variable names in Python must start with either a letter (a-z, A-Z) or an underscore (\_). They cannot start with a digit (0-9). Since 'eggs' starts with a letter ('e'), it is a valid variable name. 100 starts with a digit, so it an invalid variable name.

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

We can use the below three functions to get the integer, floating-point number, or string version of a value:

int(): This function converts a value to an integer. It takes a numeric value and returns the corresponding integer value.

eg: value = 10.5

integer\_value = int(value)

print(integer\_value) # Output: 10

float(): This function converts a value to a floating-point number. It takes a numeric value and returns the corresponding floating-point number.

eg: value = "3.14"

float\_value = float(value)

print(float\_value) # Output: 3.14

str(): This function converts a value to a string. It takes any value as an argument and returns its string representation.

eg: value = 42

string\_value = str(value)

print(string\_value) # Output: "42"

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

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

The given expression 'I have eaten ' + 99 + ' burritos.' causes an error because it involves concatenating different data types without converting them to a compatible type (strings and numeric value).

To fix this, we can make the data types to same type (i.e, by converting the integer to string type)

'I have eaten ' + str(99) + ' burritos.' - This will solve our purpose.

Output: I have eaten 99 burritos.