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' => String

-87.8 => Float

- => Expression

/ => Expression

* => Expression

6 => Integer

2. What is the difference between string and variable?

Ans:

In Python, a string is a data type used to represent text, which is a sequence of characters enclosed within single quotes (' '), double quotes (" "), or triple quotes (''' ''' or """ """). Strings can contain letters, numbers, symbols, and spaces.

A variable, on the other hand, is a name that refers to a value stored in the computer's memory. Variables can store different types of data, including strings, numbers, lists, dictionaries, and more. In Python, variables are created by assigning a value to a name using the assignment operator (=).

3. Describe three different data types.

Ans:

Integer (int):

Integers are whole numbers, positive or negative, without any decimal point. They can range from negative infinity to positive infinity. In Python, integers are represented using the int data type. For example:

x = 10

y = -5

String (str):

Strings are sequences of characters, enclosed within single quotes (' '), double quotes (" "), or triple quotes (''' ''' or """ """). They are used to represent text data. Strings can contain letters, numbers, symbols, and spaces. In Python, strings are represented using the str data type. For example:

name = "Alice"

message = 'Hello, World!'

List:

Lists are ordered collections of items, which can be of different data types. They are mutable, meaning you can modify the elements after the list is created. Lists are created by placing comma-separated values between square brackets [ ]. In Python, lists can contain any combination of data types. For example:

my\_list = [1, 2, 'three', 4.5, True]

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

Ans:

An expression in Python is made up of one or more operands and operators, combined in a manner that evaluates to a value.

Operands: These are the values or variables that the operators act upon. For example, in the expression 2 + 3, the operands are 2 and 3.

Operators: These are symbols that represent computations like addition, subtraction, multiplication, division, etc. For example, in the expression 2 + 3, the operator is +.

Expressions can also include function calls, method calls, and other valid Python syntax that evaluates to a value.

All expressions in Python, when evaluated, produce a value. This value can be of various data types, such as integers, strings, booleans, or more complex data types like lists or dictionaries. Expressions can be as simple as a single variable or value, or they can be complex, involving multiple operations and functions.

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

Ans:

The primary difference between an expression and a statement in Python lies in their functionality and purpose:

Expression:

An expression is a combination of values, variables, operators, and function calls that evaluates to a single value.

It can be as simple as a single value or variable, or it can be complex, involving multiple operations.

Expressions can be used anywhere a value is expected, such as in assignments, function arguments, or within other expressions.

Example: 2 + 3, x \* 5, len(my\_list), etc.

Statement:

A statement is a complete line of code that performs some action. It may or may not produce a value.

Statements are typically instructions that change the state of the program, control its execution flow, or define its structure.

Assignment statements, like spam = 10, are a type of statement where a value is assigned to a variable.

Other examples of statements include conditional statements (if, elif, else), loop statements (for, while), import statements, function definitions, etc.

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

bacon = 22

bacon + 1

Ans:

The variable bacon still contains the value 22.

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

'spam' + 'spamspam'

'spam' \* 3

Ans:

The values of the two terms are:

'spam' + 'spamspam':

This is concatenation of two strings. It combines the string 'spam' with the string 'spamspam', resulting in 'spamspamspam'.

'spam' \* 3:

This is string multiplication. It repeats the string 'spam' three times, resulting in 'spamspamspam'.

So, both terms would have the value 'spamspamspam'.

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

Ans:

Variable names must start with a letter (a-z, A-Z) or an underscore (\_).

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

Ans:

In Python, we can use the following three functions to convert values between different data types:

int(): This function can be used to convert a value to an integer. It takes a numeric value or a string representing a numeric value as its argument and returns an integer. If the value is a floating-point number, it truncates the decimal part.

Example:

x = int(3.14) # x will be 3

y = int("10") # y will be 10

float(): This function can be used to convert a value to a floating-point number. It takes a numeric value or a string representing a numeric value as its argument and returns a floating-point number.

Example:

x = float(10) # x will be 10.0

y = float("3.14") # y will be 3.14

str(): This function can be used to convert a value to a string. It takes any value as its argument and returns a string representation of that value.

Example:

x = str(10) # x will be "10"

y = str(3.14) # y will be "3.14"

These functions are useful for converting values between integers, floating-point numbers, and strings as needed in your Python code.

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

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

Ans:

The expression 'I have eaten ' + 99 + ' burritos.' causes an error because you are trying to concatenate a string ('I have eaten ') with an integer (99) directly, which is not allowed in Python. The + operator for strings expects both operands to be strings.

To fix this error, you need to convert the integer 99 to a string before concatenating it with the other strings. You can do this using the str() function. Here's the corrected expression:

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

This will result in the string 'I have eaten 99 burritos.', where 99 is now treated as a string and properly concatenated with the other strings.