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.

\*

'hello'

-87.8

-

/

6

**Solution 1.:**

**Values:**

'hello' = string value

-87.8 = floating-point value

6 = integer value

**Expressions:**

‘\*’ = multiplication operator

‘-‘ = subtraction/negation operator

‘ / ’ = division operator

‘+’ = addition operator

2. What is the difference between string and variable?

**Solution 2.**

**String:**

A string is a data type used to represent text or a sequence of characters. It is typically enclosed in quotation marks, either single ('') or double (""). Strings can contain letters, numbers, symbols, and whitespace. For example

name = "John" # Here, 'name' is a variable of type string, holding the value "John"

**Variable:**

A variable is a named container that holds a value. It is used to store and manipulate data in a program. Unlike a string, a variable can hold different types of data, including strings, numbers, booleans, and more. Variables are typically defined with a name and can be assigned a value that corresponds to a specific data type. For example:

age = 25 # 'age' is a variable of type integer, holding the value 25.

3. Describe three different data types.

**Solution 3.**

The three different data types as follows:

**Integer (int):**

The integer data type represents whole numbers without any fractional or decimal part. Integers can be positive or negative. Examples of integers are -5, 0, and 10. In most programming languages, integer data types have a specific range they can represent, such as -2,147,483,648 to 2,147,483,647 for a 32-bit signed integer.

**String (str):**

The string data type represents sequences of characters, typically used to represent text. Strings are enclosed in quotation marks (single or double) to differentiate them from other data types. Examples of strings are "Hello, World!", "Good Morning", and "12345". Strings can contain letters, numbers, symbols, and whitespace.

**Boolean (bool):**

The boolean data type represents a binary value that can be either true or false. Booleans are useful for making decisions and controlling program flow based on conditions. They are often the result of a comparison or logical operation. For example, a boolean can be used to check if a condition is true or false. Examples of boolean values are True and False.

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

**Solution 4.**

An expression is made up of one or more operands and operators. It represents a computation or evaluation that produces a value. The operands can be literals (specific values) or variables, and the operators perform operations on the operands.

Expressions can involve mathematical operations, logical operations, comparisons, function calls, and more, depending on the programming language and the context in which they are used. Some common operators used in expressions include arithmetic operators (+, -, \*, /), comparison operators (>, <, ==), logical operators (and, or, not), and assignment operators (=).

All expressions in programming have a purpose of producing a value. The value generated by an expression can be stored in a variable, used as a condition for control flow statements (e.g., if statements, loops), passed as an argument to a function, or used in any other way required by the program.

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

**Solution 5.**

**Expression:**

An expression is a combination of operands and operators that evaluates to a value. It can be as simple as a single literal value or a complex combination of variables, functions, and operators. Expressions are typically used to perform computations, produce results, or generate values. For example, 2 + 3 is an expression that evaluates to the value 5. Expressions can be used within statements.

**Statement:**

A statement, on the other hand, is a complete instruction or action that performs a specific task. It is a standalone unit of code that carries out an operation or controls the flow of the program. Statements can include expressions, but they also include other programming constructs such as loops, conditionals, function definitions, and assignments. Examples of statements include variable assignments (spam = 10), control flow statements (if, for, while), and function calls.

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

bacon = 22

bacon + 1

**Solution 6.**

The variable bacon will still contain the value 22.The line bacon + 1 is an expression that adds 1 to the value of bacon, but it is not assigned to any variable or stored anywhere. It is simply an expression that computes the result of bacon + 1, which in this case would be 23. However, since the result is not assigned to bacon or any other variable, the value of bacon remains unchanged as 22.

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

'spam' + 'spamspam'

'spam' \* 3

**Solution 7.**

1. **'spam' + 'spamspam':**

The expression 'spam' + 'spamspam' performs string concatenation, which means it combines the two strings together. The result of this expression would be the string 'spamspamspam'.

2.**'spam' \* 3:**

The expression 'spam' \* 3 performs string repetition. It repeats the string 'spam' three times. The result of this expression would be the string 'spamspamspam'.Therefore, both expressions would yield the same result of 'spamspamspam'

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

**Solution 8.**

In most programming languages, including Python, variable names need to follow certain rules and conventions. Here is the reason why 'eggs' is a valid variable name while '100' is invalid:

**Starting character:**

Variable names typically cannot start with a number. They should start with a letter (a-z, A-Z) or an underscore (\_). In this case, 'eggs' starts with a letter, making it a valid variable name.

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

int(): This function is used to convert a value to an integer. It can be used with numeric values or strings representing integers. If the value is a floating-point number, it will be truncated towards zero. Example:

value = int(10.5)

print(value) # Output: 10

float(): This function is used to convert a value to a floating-point number. It can be used with numeric values or strings representing numbers. Example:

value = float("3.14")

print(value) # Output: 3.14

str(): This function is used to convert a value to its string representation. It can be used with various data types, including integers, floating-point numbers, booleans, and more. Example:

value = str(42)

print(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 attempts to concatenate a string ('I have eaten ') with an integer (99). In Python, the + operator is used for both addition and string concatenation. However, it requires the operands on both sides to be of the same type.

To fix this error and perform the desired string concatenation, you need to convert the integer 99 to a string before concatenating it. One way to fix the expression is to use the str() function to convert the integer to a string:

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