1. Based on the elements provided:

Values:

'hello' (string value)

-87.8 (floating-point number value)

6 (integer value)

Expression:

\*

-

/

+

1. **Difference between string and variable :**

A string is a data type that represents a sequence of characters. It is typically used to store and manipulate textual data. In many programming languages, strings are enclosed in quotation marks (single or double) to differentiate them from other types of data. For example, 'hello' and "world" are examples of strings.

A variable, on the other hand, is a named storage location that can hold a value. It is used to store and manipulate data in a program. Variables can hold various types of data, including strings. They are used to store and retrieve values during program execution. For example, you can declare a variable called "name" and assign it a string value like "John" or "Alice".

In summary, a string is a specific type of data that represents textual information, while a variable is a named container that can hold different types of data, including strings. Variables provide a way to store and manipulate data dynamically in a program.

1. **Here are descriptions of three different commonly used data types:**

i) Integer (int): An integer data type represents whole numbers without any fractional or decimal parts. It can be either positive or negative, including zero. Integers are used for a variety of purposes, such as counting, indexing, and performing arithmetic operations. Examples of integers are -5, 0, and 10.

ii) String (str): A string data type represents a sequence of characters. It is used to store textual data, such as words, sentences, or even individual characters. Strings are typically enclosed in quotation marks (either single or double) in programming languages. Examples of strings are "hello", "world", and "OpenAI".

iii) Boolean (bool): A boolean data type represents a logical value that can be either true or false. Booleans are used to make decisions or perform conditional operations in programming. They are often the result of a comparison or a logical operation. Examples of boolean values are True and False.

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

An expression is made up of one or more operands and operators.

Operands are the values or variables involved in the expression. They can be constants (fixed values) or variables (representing varying values). For example, in the expression "2 + 3", the operands are the numbers 2 and 3.

Operators are symbols or keywords that perform specific operations on the operands. They dictate how the operands are combined or manipulated. Examples of operators include arithmetic operators (+, -, \*, /), comparison operators (>, <, ==), logical operators (and, or, not), and assignment operators (=, +=, -=), among others.

Expressions in programming are used to perform calculations, make comparisons, evaluate conditions, or assign values. They can involve simple or complex operations and can include multiple levels of nesting. Expressions can be used in various contexts, such as mathematical calculations, conditional statements, looping constructs, or function calls.

When an expression is evaluated, it produces a resulting value based on the operations performed on the operands. The resulting value can be used further in the program for various purposes, such as storing in variables, displaying output, making decisions, or as input for other expressions.

In summary, expressions are constructed using operands and operators and are used to perform operations, make evaluations, and generate values in a programming language. They are fundamental in carrying out computations and implementing logic in programs.

1. **This assignment statements, like spam = 10. What is the difference between an**

**expression and a statement?**

The main difference between an expression and a statement lies in their purpose and behavior within a program.

An expression is a combination of operands and operators that can be evaluated to produce a value. It can be as simple as a single constant or variable, or it can involve complex operations. Expressions can be used within larger expressions or as parts of statements. They are used to perform computations, make comparisons, or retrieve values. For example, in the expression "2 + 3", the addition operator (+) combines the operands 2 and 3 to produce the value 5.

A statement, on the other hand, is a complete instruction that performs an action or controls the flow of a program. It is a unit of code that performs a specific task or operation. Unlike expressions, statements do not necessarily produce a value. They are used to carry out tasks such as assigning values to variables, looping, branching, or calling functions. For example, the statement "spam = 10" assigns the value 10 to the variable "spam".

In summary, expressions are evaluated to produce a value, while statements are executed to perform actions or control the program's flow. Expressions are often part of statements, and statements typically include expressions as part of their execution.

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

**bacon = 22**

**bacon + 1**

The variable bacon will still contain the value 22.

The second line, bacon + 1, is an expression that evaluates to 23, but it is not assigned to any variable. Therefore, the value of bacon remains unchanged. If you want to update the value of bacon to reflect the result of the addition, you would need to assign the expression back to the bacon variable, like this:

bacon = bacon + 1

This would update the value of bacon to 23.

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

**'spam' + 'spamspam'**

**'spam' \* 3**

The values of the following two terms would be:

'spam' + 'spamspam': The result would be the concatenation of the two strings 'spam' and 'spamspam', resulting in 'spamspamspam'. The plus operator (+) is used for string concatenation in this case.

'spam' \* 3: The result would be the repetition of the string 'spam' three times. The asterisk operator (\*) is used for string repetition or replication. Therefore, the result would be 'spamspamspam'.

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

In most programming languages, including Python, variable names need to follow certain rules and conventions. Here are some common guidelines:

Valid characters: Variable names can consist of letters (both uppercase and lowercase), digits (0-9), and underscores (\_). However, they cannot start with a digit.

Starting with a letter: Variable names should generally start with a letter (a-z or A-Z) or an underscore (\_). It's not recommended to start a variable name with a digit.

Based on these guidelines:

'eggs' is a valid variable name because it starts with a letter and consists of only valid characters (letters).

100 is an invalid variable name because it starts with a digit. It violates the rule of starting with a letter or an underscore.

However, it's worth noting that some programming languages may have specific rules or conventions that differ slightly. It's always best to consult the documentation or guidelines of the specific programming language you are using to understand the exact rules and conventions for variable naming.

1. **What three functions can be used to get the integer, floating-point number, or string**

**version of a value?**

int() ; float(); str()

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

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

The expression 'I have eaten ' + 99 + ' burritos.' causes an error because it involves concatenating a string with an integer value directly. In Python, you can only concatenate strings with other strings, not with integers or other data types.

To fix this, convert the integer value to a string before concatenating it with the other strings. Here's the corrected version:

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

By using the str() function, we convert the integer 99 to a string, allowing it to be concatenated with the other strings. The corrected expression will result in a valid string concatenation.