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

**1. Answer:**

**Values**:

* 'hello' (string)
* -87.8 (floating-point number)
* 6 (integer)

**Expression:**

* \* (Multiplication operator)
* - (Subtraction)
* / (Division)
* + (Addition)

2. What is the difference between string and variable?

2. Answer:

|  |  |
| --- | --- |
| String | Variable |
| * A string is a data type used to represent a sequence of characters. * It is typically used to store and manipulate textual data. * In most programming languages, strings are enclosed in quotation marks (either single or double quotes). * For example, "Hello, world!" is a string literal. * It represents a specific type of data (text) and is used for storing and manipulating textual information. | * A variable is a named container or memory location that holds a value. * It is used to store and retrieve data during the execution of a program. * Variables have a name and a data type associated with them. * The value of a variable can change throughout the program execution. * Variables can hold different types of data, including strings, numbers, Booleans, and more. * It is a named storage location that can hold different types of data, including strings. Variables provide a way to store and access values dynamically during program execution. |

3. Describe three different data types.

**3. Answer:**

The following are the three built-in data types in Python.

* Integer (int): The integer data type represents whole numbers without decimal points. It can be positive or negative. For example, 3, -5, and 0 are integers.
* String (str): The string data type represents a sequence of characters enclosed in single ('') or double ("") quotes. It is used to store and manipulate textual data. For example, 'Hello', "Python", and '123' are strings.
* Boolean (representing True or False), It often used in conditional statements and logical operations to control the flow of a program.

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

**4. Answer:**

* Expressions in Python are combinations of variables, values, and operators that produce a result. They perform calculations or comparisons and give back a value. For example, adding two numbers like 2 + 3, or comparing if a number is greater than another like 5 > 3, are expressions.
* Expressions in Python serve the purpose of producing a value. When an expression is evaluated or executed, it performs calculations, comparisons, or other operations and returns a result.

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

**5. Answer:**

* Expression:

An expression produces a value when evaluated.

It combines variables, values, and operators to perform calculations or comparisons.

Examples: 2 + 3, len("hello"), x > y.

* Statement:

A statement performs an action or controls the flow of a program.

It is a complete line of code.

Examples: spam = 10, if x > y:, for item in list:.

In summary, expressions produce values, while statements perform actions or control program flow. Expressions are used within statements to compute values.

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

bacon = 22

bacon + 1

**6. Answer:**

After running the given code, the variable bacon will still contain the value 22.

the bacon = 22 assigns the value 22 to the variable bacon. However, the bacon + 1 performs the addition operation but does not assign the result back to the bacon variable or store it anywhere. Therefore, the value of bacon remains unchanged at 22.

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

'spam' + 'spamspam'

'spam' \* 3

**7. Answer:**

The values of the two terms will be as follows:

'spam' + 'spamspam': The result of this expression will be 'spamspamspam'. The + operator concatenates (joins) two strings together, resulting in a new string that contains the combined characters of the original strings.

'spam' \* 3: The result of this expression will be 'spamspamspam' as well. The \* operator, when used with a string and an integer, repeats the string a specified number of times. In this case, 'spam' is repeated three times, resulting in the string 'spamspamspam'.

In both cases, the resulting value is the same: 'spamspamspam'.

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

8. Answer:

In Python, variable names need to follow certain rules to be considered valid. Here's why 'eggs' is a valid variable name, while 100 is considered invalid:

* Variable names can contain letters (both uppercase and lowercase), numbers, and underscores.
* The first character of a variable name cannot be a number.

On the other hand, 100 is invalid, here 100 starts with a number, violating the rule that a variable name cannot begin with a digit. Therefore, 100 is not considered a valid variable name in Python.

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

**9. Answer:**

In Python, you can use the following three functions to obtain the integer, floating-point number, or string version of a value:

1. int(): This function can be used to convert a value into an integer. It takes a value as an argument and returns the integer representation of that value.
2. float(): This function is used to convert a value into a floating-point number. It takes a value as input and returns the floating-point representation of that value.
3. str(): This function converts a value into a string representation. It takes a value and returns a string version of that value.

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

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

**10. Answer:**

The given expression 'I have eaten ' + 99 + ' burritos.' causes an error because it tries to concatenate a string with an integer value directly. In Python, concatenation using the + operator is only allowed between two strings.