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

**Answer –**

Values:

1. 'hello' (This is a string value)
2. -87.8 (This is a floating-point number value)
3. 6 (This is an integer value)

Expressions (Mathematical Operators):

1. \* (Multiplication operator)
2. - (Subtraction operator)
3. / (Division operator)
4. + (Addition operator)

2. What is the difference between string and variable?

A string is a specific type of data representing text, while a variable is a value that can change, depending on conditions or on information passed to the program.

It’s a kind of a container that can hold various types of data, including strings. You can assign a string to a variable to work with and manipulate the string data within your program.

Strings and variables are different concepts in programming.

1. String:
   * A string is a data type used to represent a sequence of characters. These characters can be letters, numbers, symbols, or even empty spaces.
   * Strings are typically enclosed in single quotes (' '), double quotes (" "), or triple quotes (''' ''' or """ """). For example, 'hello', "12345", and "This is a string!" are all strings.
   * Strings are used to store and manipulate textual data, such as names, messages, or any sequence of characters.
   * Strings are immutable in many programming languages, which means they cannot be changed once created. Operations on strings typically result in new strings.
2. Variable:
   * A variable is a container or a storage location in a program that holds a value. This value can be of any data type, including strings.
   * Variables are used to store and manage data that can change during the execution of a program.
   * Variable names are used to reference and access the data stored within them. These names are chosen by the programmer and follow specific rules (e.g., no spaces, no special characters except underscore or digits as allowed in many programming languages).
   * You can assign a string (or any other type of data) to a variable. For example, in Python, you can do something like my\_string = "Hello, World!" where my\_string is a variable holding a string value.

3. Describe three different data types.

a). Integer (int):

* The integer data type is used to represent whole numbers without any fractional or decimal parts. For eg: -3, 0, 42, and 1001.
* Integers can be both positive and negative, as well as zero.
* Arithmetic operations such as addition, subtraction, multiplication, and division can be performed on integers.

b). Floating-Point (float):

* The floating-point data type is used to represent real numbers that can have both integer and fractional parts, including numbers with a decimal point. For eg: 3.14, 0.5, 2.71828 etc
* Floating-point numbers are used for precise representation of values that may have fractional or decimal components.
* However, due to the limitations of binary representation, floating-point numbers can sometimes have rounding errors in certain calculations.

c). Boolean (bool):

The boolean data type has only two possible values: True and False. It is used to represent binary or logical values.

Booleans are commonly used in conditional statements and decision-making in programming.

They help determine the truth or falsity of expressions, and they are essential for controlling the flow of a program. For example, a boolean can be used to determine whether a condition is met or not, and based on this, the program can take different actions.

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

An expression in programming is made up of one or more of the following elements:

1. Values: Expressions can include constants or data values, such as numbers (integers or floating-point), strings, or boolean values.
2. Operators: Expressions often include operators, which are symbols or keywords that represent operations to be performed on values. Common operators include arithmetic operators (+, -, \*, /), comparison operators (>, <, ==), logical operators (and, or, not), and assignment operators (=).
3. Variables: Variables are used to store and reference values. In an expression, a variable can be used to represent the value it holds. For example, if you have a variable **x**, you can use it in an expression like **x + 5**.
4. Function calls: Functions are blocks of code that perform specific tasks. You can call functions within expressions, passing arguments to the function and using the result of the function in the expression.

Expressions, as a general concept, are used to compute or evaluate a result based on the values, operators, and variables included in them. The primary purpose of expressions is to produce a value.

What expressions do can vary depending on the context:

1. **Arithmetic Expressions:** These involve mathematical operations and calculate numeric results. For example, **5 + 3** is an expression that calculates the sum of 5 and 3.
2. **Boolean Expressions:** These involve logical operations and evaluate to a boolean value (True or False). For example, **x > 10** is a boolean expression that checks if the value of **x** is greater than 10 and results in either True or False.
3. **String Expressions:** These involve string operations, such as concatenation, and result in a new string. For example, **"Hello, " + "world!"** is an expression that concatenates two strings to form a new string.
4. **Assignment Expressions:** These assign a value to a variable. For example, **x = 42** is an assignment expression that stores the value 42 in the variable **x**.
5. **Function Call Expressions:** These call a function, which can have various effects, including returning a value or modifying data. For example, **len("hello")** is an expression that calls the **len** function to determine the length of the string "hello."

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

Some key differences between expressions and statements:

* Expressions produce values, whereas statements perform actions.
* Expressions can be embedded within statements to provide data for those statements.
* Statements are typically composed of one or more expressions and keywords that determine program behavior (e.g., control flow, loops, or function definitions).
* Expressions can be part of more complex statements, such as conditional statements, where they are used to make decisions.

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

bacon = 22

bacon + 1

**Answer** - The variable bacon will still contain the value 22

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

'spam' + 'spamspam'

'spam' \* 3

Answer:

'spam' + 'spamspam' will result into **‘spamspamspam’**

'spam' \* 3 will result into **‘spamspamspam’**

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

Because 100 doesn’t follow the general principle which variable names should.

i.e., It should start with a letter or an underscore.

Variable names cannot start with a digit.

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

Answer:

Three common functions are as follows:

int(): It takes a value (e.g., a string or a floating-point number) as its argument and returns an integer representation of that value. If the conversion is not possible, it may raise an error.

float(): It takes a value (e.g., an integer or a string) as its argument and returns a floating-point representation of that value.

str(): It takes a value of any data type as its argument and returns a string representation of that value.

These are essential for type conversion when we need to work with values of different data types and ensure that they are in the desired format.

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

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

Answer:

Here we are attempting to concatenate a string with an integer directly so it is causing an error.

Fix:

We can convert the integer into a string before concatenating it with the other strings.

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