

1. In the below elements which of them are values or an expression?
e.g.: - values can be integer or string and expressions will be mathematical operators.

*

'hello'

-87.8

-

/

+

6

Ans=

'hello' = Value (String)

-87.8 = Value (Floating-point number)

/ = Expression (Division operator)

6 = Value (Integer)

2. What is the difference between string and variable?

A string is a sequence of characters enclosed within quotes (either single quotes or double quotes) and represents textual data in programming. For example, "Hello Love" is a string. A variable, on the other hand, is a named memory location that stores a value. It can hold different types of values such as numbers, strings, and Boolean values. A variable can be assigned a value, updated or re-assigned with a new value during the program execution.

The difference between a string and a variable is that a string is a type of data while a variable is a container that can hold different types of data, including strings. A string is a specific value or

data type, whereas a variable can be assigned a string value or any other value, depending on its data type.

3. Describe three different data types.

1. **Integer:** An integer is a whole number without a decimal point, such as 1, 10, -5, etc. In many programming languages, integers have a specific size limit and can be signed or unsigned (able to represent negative or non-negative numbers).
2. **Floating-Point:** A floating-point number, also known as a decimal number, is a number that has a decimal point, such as 1.5, -0.25, etc. Floating-point numbers can represent a wider range of values than integers, but they can also be less precise due to the way they are stored in the computer's memory.
3. **Boolean:** A Boolean is a data type that represents a binary value, which is either true or false. Booleans are often used in conditional statements to make decisions in a program's logic flow. For example, if a user is logged in, the Boolean value would be true, whereas if they are not logged in, it would be false.

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

In programming, an expression is a combination of one or more values, variables, operators, and function calls that are evaluated to produce a resulting value. An expression can be as simple as a single value or variable, or it can be complex and include multiple operators and function calls.

All expressions have the primary purpose of producing a value. Depending on the operators and operands used, expressions can produce a wide variety of values, such as numeric values, Boolean values, and string values.

In addition to operators, expressions can also include function calls that perform specific operations on the values passed as arguments. The functions can be built-in functions provided by the programming language, or they can be user-defined functions created by the programmer.

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

In programming, an expression is a combination of one or more values, variables, operators, and function calls that produce a resulting value. For example, $3 + 4$ is an expression that evaluates to 7.

On the other hand, a statement is a line of code that performs an action or assigns a value. An assignment statement such as `spam = 10` is an example of a statement that assigns the value 10 to the variable `spam`.

In other words, an expression is a piece of code that produces a value, while a statement is a line of code that performs an action or causes a change in the program's state.

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

```
bacon = 22
```

```
bacon + 1
```

Ans- After running the following code, the variable `"bacon"` contains the value 22, as the expression `"bacon + 1"` does not modify the original value of `"bacon"`, but instead produces a new value of 23 which is not assigned to any variable.

So, if we want to update the value of `bacon` variable to 23, we need to assign the result of `"bacon + 1"` back to the `"bacon"` variable like this:

```
bacon = 22
```

```
bacon = bacon + 1
```

```
print(bacon)
```

```
>>>>> Output: 23
```

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

```
'spam' + 'spamspam'
```

```
'spam' * 3
```

Ans=== The values of the following two terms are:

- `'spam' + 'spamspam'`: This expression concatenates two strings `'spam'` and `'spamspam'` together, resulting in the string `'spamspamspam'`. The `+` operator when used with strings concatenates them together.
- `'spam' * 3`: This expression multiplies the string `'spam'` by 3, resulting in the string `'spamspamspam'`. The `*` operator when used with a string, repeats the string a given number of times.

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

Variable names can consist of letters (a to z or A to Z), digits (0 to 9), and underscore (`_`), but they cannot start with a digit. Therefore, `"eggs"` is a valid variable name because it starts with a letter, and can contain letters, digits, and underscores.

`"100"` is an invalid variable name because it starts with a digit. In Python, variable names cannot start with a digit, as this may cause confusion with numeric literals. For example, if we had a variable named `"100"`, it would be unclear whether it is a variable containing a number or a literal value of 100.

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

In Python, the following 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, as well as strings that represent integers.
- **`'float' ()`**: This function is used to convert a value to a floating-point number. It can be used with numeric values, as well as strings that represent floating-point numbers.
- **`'str' ()`**: This function is used to convert a value to a string. It can be used with any value, including numbers, Boolean values, and objects.

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

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

Ans= This expression causes an error because it is attempting to concatenate a string ('I have eaten ') with an integer value (99), which is not allowed in Python.

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