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:

- ‘hello’(String Value)
- -87.8(Floating Value)
- 6 (Integer Value)

Expressions:

- *(Multiplication Operator)
- - (Subtractor Operator)
- / (Division Operator)
- + (Addition Operator)

2. What is the difference between string and variable?

Answer:

String in Python:

- In Python, a string is a sequence of characters enclosed in either single quotes ('') or double quotes (""). It is a data type used to represent and manipulate text. For example, "hello" and ''world' are both strings in Python.
- Strings in Python are immutable.

Variable in Python:

- A variable in Python is a name given to a memory location that stores a value. It acts as a placeholder to store data of any type, including strings. Unlike strings, variables can hold various types of data, including numbers, lists, dictionaries, and more.
- Variables in Python are mutable.

3. Describe three different data types.

Answer:

Python offers a wide range of data types to handle different kinds of data. Here are three commonly used data types in Python:

Integer(int):

The integer data type represents whole numbers without fractional parts. It can be positive, negative, or zero. For example, 0, -5, and 10 are all integers. Python's integer type has no maximum limit, allows to work with arbitrarily large numbers. Integer can perform mathematical operations like addition, subtraction, multiplication, and division on integers

String(str):

The string data type represents sequences of characters enclosed in quotes. It is used to store and manipulate textual data. Strings can be created using single quotes (''), double quotes ("") or triple quotes (""" or '''). We can perform various operations on strings, such as concatenation (joining strings), slicing (extracting parts of a string), and accessing individual characters.

List(list):

A list is an ordered collection of items, which can be of different data types. Lists are created by enclosing comma-separated values within square brackets []. They are mutable, allowing you to add, remove, or modify elements. Lists can contain duplicate items and can be indexed and sliced to access specific elements or subsets.

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

Answer:

In Python, an expression is made up of one or more operators and operands. An expression represents a computation or a value in Python.

Components of an expression:

Operators: Operators are used to perform specific operations on the operands. Python supports various types of operators, such as arithmetic operators (+, -, *, /), comparison

operators (==, !=, <, >), logical operators (and, or, not), assignment operators (=, +=, -=), and more.

Operands: Operands are the values or variables on which the operators act. They can be literals (constant values) or variables that hold values of different data types like integers, strings, lists, etc.

Expressions do in Python:

Evaluate: Expressions are evaluated to produce a value. When Python encounters an expression, it calculates the result based on the operators and operands involved. The result can be a numeric value, a boolean value, a string, or any other data type depending on the expression.

Assign: Expressions can be used to assign a value to a variable. By assigning the result of an expression to a variable, you can store and manipulate the computed value for future use.

Control Flow: Expressions play a crucial role in controlling the flow of a program. They are used in conditions (if statements, while loops, etc.) to determine the execution path based on the evaluation of the expression. The result of an expression helps determine whether a certain block of code is executed or skipped.

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

Expression: An expression is a combination of literals, variables, operators, and function calls that produces a value when evaluated. It can be as simple as a single constant or variable, or it can be more complex involving multiple components and operators.

Expressions can be used within larger expressions or as part of statements. They always yield a value and can be used in various contexts, such as calculations, comparisons, or as arguments to functions.

Statement: A statement, on the other hand, is a complete instruction or action that performs a specific operation. It represents a complete unit of code that Python executes. Statements are used to control the flow of execution, define behavior, and perform actions. They don't necessarily produce a value when executed. Expressions produce values when evaluated, while statements perform actions.

In the above example, spam = 10 is an assignment statement that assigns the value 10 to the variable spam. It doesn't produce a value but changes the state of the program.

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

bacon = 22

bacon + 1

OUTPUT:

```
In [1]: bacon = 22
        bacon + 1

Out[1]: 23
```

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

'spam' + 'spamspam'

'spam' * 3

OUTPUT:

```
In [2]: 'spam' + 'spamspam'
        'spam' * 3

Out[2]: 'spamspamspam'
```

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

Answer:
In Python, variable names must follow certain rules and conventions. Here's an explanation of why eggs is a valid variable name while 100 is invalid.

Variable Naming Rules:

- Start with a letter or underscore: Variable names in Python must begin with a letter (a-z, A-Z) or an underscore (_). They cannot start with a number.
- Can contain letters, numbers, and underscores: After the first character, variable names can contain letters, numbers, and underscores. They are case-sensitive, meaning eggs and Eggs would be considered different variables.
- Avoid using reserved words: You cannot use reserved words, which have special meanings in Python, as variable names. Examples of reserved words include if, while, for, print, and so on.

EXPLANATION:

- The variable name eggs is valid because it starts with a letter (e) and contains only letters. It adheres to the naming rules, making it a valid variable name.
- On the other hand, 100 is not a valid variable name because it starts with a number (1). According to the rules, variable names cannot start with numbers. However, We can include numbers in variable names as long as they come after the first character.

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

- Integer Conversion: int() The int() function is used to convert a value to an integer.
- Floating-Point Conversion: float() The float() function is used to convert a value to a floating-point number.
- String Conversion: str() The str() function is used to convert a value to its string representation.

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

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

Answer:

In, This expression it causes an error because concatenate is missing and full stop is present in expression. we can fix this expression by adding concatenate in between the integer value of 99 and then we need to remove full stop at the end of burritos And Then, After we can run the expression so we can get the output correctly.

Output:

```
In [4]: 'I have eaten ' + "99" + ' burritos'
Out[4]: 'I have eaten 99 burritos'
```