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.

a. \*

b. 'hello'

c. -87.8

d. –

e. /

1. +
2. 6

Ans.

1. expression
2. value
3. value
4. expression
5. expression
6. expression
7. value

2. What is the difference between string and variable?

Ans. Strings are sequence of characters, such as letters, numbers, symbols, or spaces. They are represented in enclosed quotation marks; you can create a string by enclosing characters in either single quotes (' ') or double quotes (" "). But the quotation marks cannot be mismatched for example ‘Hello”. The quotation marks must be of same type. For example, "Hello, World!" and '12345' are strings.

Variables are like containers of a program. They are storage locations in the computer’s memory, and they are used to store and represent different types of data such as strings, numbers, Boolean values or objects. Variables are defined with a name and may be assigned a value. For example, in Python, you can create a variable called "message" and assign it the value "Hello, World!" by using the following code: message = "Hello, World!".

3. Describe three different data types.

Ans. Integers: An integer represents whole numbers without any fractional or decimal parts. Integers can be positive or negative. Integers have a fixed range of values they can represent, depending on the number of bits allocated for their storage. Common operations performed with integers include addition, subtraction, multiplication, and division. Examples of integers are -3, 0, 42, and 1000.

Float: Floats are decimal values or fractional numbers like 0.23. Floats can be negative or positive. They can store small as well as large numbers. Examples of floats are 3.14, -0.5, 2.71828, and 1.0e-5.

String: A string represents a sequence of characters, such as letters, numbers, symbols, or spaces. Strings are used to store and manipulate textual data. They are typically enclosed in quotation marks (single or double) in most programming languages. Strings can be concatenated, sliced, and subjected to various operations like finding substrings, changing case, and more. Example, “Hello World”.

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

Ans. An expression is a combination of values, variables, operators, and function calls that represents a computation and produces a value. It comprises values, variables, operators, and function calls. Values can be constants or variables that hold data. Variables are named storage locations used to store and manipulate values during program execution. Operators are symbols or keywords that perform operations on values or variables, including arithmetic, comparisons, logic, and assignments. Functions are blocks of code that can be called multiple times in a program using function calls.

Expressions are versatile tools in programming that allow us to perform various operations such as calculations, comparisons, assignments, and function calls. Here are some key points about expressions:

* Calculation: Expressions involving arithmetic operators can produce numeric results. For instance, the expression "5 + 6" calculates the sum and yields the value 11.
* Comparison: Expressions can compare values or variables using comparison operators. The outcome of a comparison expression is a Boolean value (True or False), indicating the truth or falsity of the comparison. For instance, the expression "x > 10" evaluates to either True or False based on the value of the variable "x".
* Assignment: Expressions can be used in assignment statements to assign values to variables. For example, the expression "x = 5 + 3" evaluates to 8 and assigns that value to the variable "x".
* Function Calls: Expressions can invoke functions and provide arguments to them. Functions can perform complex operations and return results that can be used within expressions. For instance, the expression "len('Hello')" 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?

Ans.

|  |  |
| --- | --- |
| **Expression** | **Statement** |
| An expression is a combination of values, variables, operators, and function calls that produces a value when evaluated.  Expressions can be used to do mathematical calculations or string concatenation or call a function like max(). | A statement is a complete unit of code that performs an action or carries out a specific task. Spam= 10 is an assignment statement, here spam is assigned a value of 10.  Statements include control flow statements (if-else, loops), function declarations, import statements, and more. |

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

bacon = 22

bacon + 1

Ans. After running the code bacon will contain 22 as we have not stored the value of bacon + 1 in bacon itself, hence the value of bacon remains unchanged.

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

'spam' + 'spamspam'

'spam' \* 3

Ans. ‘spam’ + ‘spamspam’ = ‘spamspamspam’ (There will be string concatenation)

‘spam’ \* 3 = ‘spamspamspam’ (Here the string is multiplied 3 times)

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

Ans. eggs is a valid variable name because variables in python must start with a letter or an underscore it cannot start with a number. Hence 100 is invalid.

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

Ans. To get integer version we will use int(), for floating-point number we use float() and for string we use str().

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

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

Ans. This expression will cause an error because ‘I have eaten’ and ‘bacon’ are strings. However, 99 is an integer. We cannot concatenate different data types directly in Python. To fix the error we can first convert 99 into a string. We can convert the integer using str(99). The expression will be:

‘I have eaten’ + str(99) + ‘burritos.’