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

Ans -1

: expression (multiplication operator)

hello : value (string)

-87.8 : value (float)

: expression (subtraction operator)

/ : expression (division operator)

: expression (addition operator)

6 : value (integer)

2. What is the difference between string and variable?

Ans 2

A string is a sequence of characters enclosed in quotes either single or double. It is used to represent text or messages in a program code.

A variable is a container that holds a value, which can be a string, number, or other data types. It has a name that can be used to refer to the value it holds. Variables are used to store and manipulate data in a program codes.

3. Describe three different data types.

Ans 3

Integer (int): An integer is a whole number with no decimal places. Integers can be positive or negative, and they can be used for counting or indexing.

String (str): A string is a sequence of characters enclosed in quotes either single or double. Strings are used to represent text or messages in a program, and they can be manipulated using various string functions.

Float (float): A float is a whole number with decimal places. Float can be positive or negative, and they can be used for counting or indexing.

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

Ans 4

An expression is made up of one or more operands (values or variables) and one or more operators that perform some kind of operation on those operands. Expressions can also include function calls, method calls, or other elements that return a value.

Expressions are used to perform calculations, manipulate data, make decisions, and create more complex logic in a program. Eg, 2 + 3 is an expression that uses the addition operator (+) to add the operands 2 and 3 together, resulting in the value 5. Another example of an expression is a function call like len("hello"), which calls the built-in len() function to return the length of the string "hello".

All expressions in programming evaluate to a value, which can be of any data type. The value that an expression evaluates to can then be assigned to a variable, printed to the console, or used in other expressions. Eg, the expression x = 2 + 3 assigns the value 5 to the variable x, while the expression print(2 + 3) prints the value 5 to the console.

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

Ans 5

A statement is a piece of code that performs some action, but does not necessarily evaluate to a value. Examples of statements include variable assignments (spam = 10), function calls (print("hello")), and control flow statements like if, while, and for loops.

The main difference between expressions and statements is that expressions always evaluate to a value, while statements do not necessarily do so. An assignment statement like spam = 10 sets the value of the variable spam to 10, but it does not evaluate to a value itself. In contrast, an expression like 2 + 3 evaluates to the value 5, which can then be used in other expressions or assigned to a variable using a statement.

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

bacon = 22

bacon + 1

Ans 6

After running code the variable bacon still contains the value 22 not 23. expression bacon + 1 evaluates to 23, but the value is not assigned to the variable bacon. The expression itself does not change the value of the variable bacon.

To update the value of bacon to 23 to use an assignment statement like bacon = bacon + 1 or the shorthand bacon += 1. This would increment the value of bacon by 1 and update its value to 23.

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

spam +’spamspam’

‘spam’ \* 3

Ans 7

The values of the two terms will depend on the value of the variable spam.

Assuming that spam is a string, the expression spam + 'spamspam' concatenates the value of spam with the string 'spamspam', resulting in a new string that contains spam followed by 'spamspam'. The values of the two terms will depend on the value of the variable spam.

Assuming that spam is a string, the expression spam + 'spamspam' concatenates the value of spam with the string 'spamspam', resulting in a new string that contains spam followed by 'spamspam'. the multiplication operator \* can only be used with a string and an integer, and not with two strings.

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

Ans 8

Variable names can consist of letters, numbers, and underscores, but they cannot start with a number. This is why eggs is a valid variable name, as it consists only of letters.

100 starts with a number, which makes it an invalid variable name. Variable names cannot start with a number because it can cause confusion with numeric literals, which are used to represent numeric values in Python code. If variable names could start with a number, it would be difficult to tell whether 100 is a variable name or a numeric literal representing the value 100.

9. What three functions can be used to get the integer, floating-point number, or string

version of a value?

Ans 9

To convert a value to a specific data type using the following three functions:

int(): This function is used to convert a value to an integer. If the value is a floating-point number, it will be turned towards zero. If the value is a string, it must consist of digits and an optional sign and can be in decimal, binary, octal, or hexadecimal format.

float(): This function is used to convert a value to a floating-point number. If the value is an integer, it will be converted to a float. If the value is a string it must be a valid floating-point number in decimal format.

str(): This function is used to convert a value to a string. It returns a string representation of the value, regardless of its original data type. If the value is a number, it will be converted to a string of digits.

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

‘I have eaten’ + 99 + ‘burritos.’

Ans 10

‘I have eaten’ + 99 + ‘burritos.’ causes a TypeError because you cannot concatenate a string with an integer. In this case, the integer 99 is not automatically converted to a string. the integer to a string using the str() function, and then concatenate the strings using the + operator.

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

print(message) # Output: "I have eaten 99 burritos."