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.

\* Multiplication

'hello' String

-87.8 Float

- subtraction

/ division operator

* Addition

6 Integer

2. What is the difference between string and variable?

- A string is a data type used to represent text. It is a sequence of characters enclosed in quotes, either single or double quotes. For example, "hello" and 'world' are strings.

A variable, on the other hand, is a named storage location in a program that can hold a value of a certain data type. It is used to store and manipulate data in a program. Variables can hold different types of values, including strings, numbers, and Boolean values.

3. Describe three different data types.

- Integer: An integer is a whole number that can be positive, negative, or zero. In most programming languages, integers are represented using the "int" data type. Examples of integers include 0, 1, -5, and 10.

String:  A string is a sequence of characters that represents text. In most programming languages, strings are represented using the "str" data type. Strings are typically enclosed in quotes, either single or double quotes. Examples of strings include "hello", "world", and "123".

Boolean: A Boolean is a data type that represents a binary value, typically true or false. In most programming languages, Booleans are represented using the "bool" data type. Booleans are often used in conditional statements and loops to control the flow of a program. Examples of Boolean values include true, false, 1, and 0.

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

- In programming, an expression is a combination of values, variables, operators, and function calls that are evaluated to produce a result. Expressions can be as simple as a single value, or as complex as a multi-level computation involving multiple variables, functions, and operators.

Some examples of expressions are:

* 3 + 5
* "hello" + "world"
* (5 \* 2) / 3
* my\_variable + 1

All expressions evaluate to a value, which may be a number, a string, a boolean value, or another data type, depending on the types of values and operators used in the expression. The value of an expression may be used in various ways, such as being assigned to a variable, used as an argument to a function, or printed to the console.

Expressions are used extensively in programming to perform calculations, manipulate data, and control the flow of a program. They are a fundamental building block of programming, and understanding how to construct and evaluate expressions is essential for writing effective code.

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

- In contrast, a statement is a complete unit of code that performs a specific action or operation. A statement can include expressions as part of its syntax, but it also contains other components that define its behavior. Examples of statements include:

* Assignment statements, like spam = 10
* Conditional statements, like if x < 0:
* Loop statements, like for i in range (10):
* Function and method definitions, like def my \_function ():

So, to answer the question directly, an assignment statement like spam = 10 is a specific type of statement that assigns the value of 10 to the variable spam. It is not an expression, although it includes an expression (10) as part of its syntax. In other words, an assignment statement is a complete unit of code that performs a specific action, while an expression is a component of a statement or can stand alone as a statement.

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

bacon = 22

bacon + 1

- bacon=23

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

'spam' + 'spamspam'

'spam' \* 3

- 'spamspamspam'.

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

- In Python, variable names must follow certain rules to be valid. The rules are:

1. Variable names must start with a letter (a-z or A-Z) or an underscore (\_).
2. The rest of the variable name can contain letters, numbers (0-9), or underscores (\_).
3. Variable names are case-sensitive, so 'spam' and 'SPAM' are different variables.

Based on these rules, eggs is a valid variable name because it starts with a letter and contains only letters and/or underscores. It does not conflict with any Python keywords or built-in functions.

On the other hand, 100 is not a valid variable name because it starts with a number, which is not allowed according to the first rule. Variable names cannot start with a number in Python.

However, it is still possible to use 100 as part of a variable name, such as spam100 or variable\_100, as long as it follows the other rules for valid variable names.

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

- int (3.5) # returns 3, int ("42") # returns 42

float (3) # returns 3.0, float ("3.5") # returns 3.5.

str (42) # returns "42", str (3.14159) # returns "3.14159"

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

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

- This expression causes an error because it is trying to concatenate a string ('I have eaten ') with, an integer (99) without converting the integer to a string first. 'I have eaten ' + str(99) + ' burritos.'

## "I have eaten 99 burritos."