Assignment 1

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.

* : expression

'hello' : value

-87.8 : value

- : expression

/ : expression

+ : expression

6 : value

2. What is the difference between string and variable?

String	Variable
A string is a data type that represents a sequence of characters, enclosed in either single quotes or double quotes	A variable in is a named container that holds a value. It is used to store and reference data. Variables can hold various types of data, including strings.
A string is a specific type of data used to represent textual information	A variable, on the other hand, is a named reference to a value stored in the computer's memory.

3. Describe three different data types.

1. Integer (int): The integer data type represents whole numbers without any fractional parts. It can be positive, negative, or zero.

For example:

My_int =10

2. String (str): The string data type is used to represent sequences of characters, such as text. Strings are enclosed in single quotes ('') or double quotes ("").

For example:

A = “I am Bhavesh

3. Boolean (bool): The boolean data type represents a binary value indicating either true or false. It is useful for making logical comparisons and controlling program flow. The two possible boolean values in Python are True and False.

For example:

My_b= True

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

an expression is a combination of operators and operands that is interpreted to produce some other value

- **Arithmetic expressions:** These are expressions that use arithmetic operators like +, - , *, /, %, //, and ** to perform calculations on numeric values or variables. For example, a * b - 5 is an arithmetic expression.
- **Relational expressions:** These are expressions that use relational operators like <, >, <=, >=, ==, and != to compare two values or variables and return a boolean value (True or False). For example, a == b is a relational expression.
- **Logical expressions:** These are expressions that use logical operators like and, or, and not to combine or negate boolean values or variables and return a boolean value. For example, (a > b) and (c < d) is a logical expression.
- **Bitwise expressions:** These are expressions that use bitwise operators like &, |, ^, ~, <<, and >> to perform operations on individual bits of integers or variables and return an integer value. For example, a & b is a bitwise expression.
- **Identity expressions:** These are expressions that use identity operators like is and is not to check if two values or variables refer to the same object in memory and return a boolean value. For example, a is b is an identity expression.

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

An expression is a combination of values, variables, operators, and function calls that can be evaluated to produce a single value12. An expression can be used as part of a statement, or as an argument to a function call. For example, spam + 5 is an expression that adds 5 to the value of spam and returns the result.

The main difference between an expression and a statement is that an expression always returns a value, while a statement may or may not return a value

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

bacon = 22

bacon + 1

- The variable bacon contains 23.
- The first line of code assigns the value 22 to the variable bacon.
- The second line of code adds 1 to the value of bacon, which results in the value 23.

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

'spam' + 'spamspam'

'spam' * 3

- The first term, 'spam' + 'spamspam', should have the value 'spamspamspam'.
- The second term, 'spam' * 3, should have the value 'spamspamspam'. In Python, strings are concatenated by using the + operator. So, when we add two strings together, the result is a new string that contains the characters of both strings. For example, if we add the strings 'spam' and 'spamspam' together, the result is the string 'spamspamspam'.
- The * operator can also be used with strings to repeat them a certain number of times. For example, if we multiply the string 'spam' by 3, the result is the string 'spamspamspam'.

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

- The variable name eggs is valid because it starts with a letter and contains only letters, numbers, and underscores. The variable name 100 is invalid because it starts with a number.
- In Python, variable names can start with a letter or an underscore. They can contain letters, numbers, and underscores, but they cannot start with a number. This is because numbers are used to represent values, and variable names are used to represent references to values. If a variable name started with a number, it would be difficult to distinguish between the variable name and the value that it represents.
- Additionally, variable names cannot contain spaces. This is because spaces are used to separate words in Python code. If a variable name contained a space, it would be difficult to read and understand the code.

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

The three functions that can be used to get the integer, floating-point number, or string version of a value are int(), float(), and str(). The int() function converts a value to an integer. The float() function converts a value to a floating-point number. The str() function converts a value to a string.

For example,

- the following code converts the value '100' to an integer:

Int('100') :This will return the value 100.

- The following code converts the value '100.0' to a floating-point number:

Float('100.0') : This code will return the value 100.0.

- The following code converts the value '100' to a string:

String("100") :This would be display 100 as a string

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

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

The expression 'I have eaten ' + 99 + ' burritos.' causes an error because the + operator cannot be used to concatenate a string with an integer. To fix the error, you can use the str() function to convert the integer to a string before concatenating it with the other string.

Correct will be : print("I have eaten",str(99),"burritos"