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 :- Values : - 87.8,6,’hello’

Expression :- \*, - , /, +

2. What is the difference between string and variable?

ANS :- A string consist of values within the “ ” or ‘ ‘ while variable is similar a container which is used to store a data of specific data types.

3. Describe three different data types.

ANS :- In python there are various types of data types among them numeric data types ( int, float, complex ) string data types (str) and sequence data type (list, tuple, range).

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

ANS :- In Python, an expression is a combination of values, variables, operators, and function calls that can be evaluated to produce a result. It's like a recipe that tells Python what to do and how to calculate or manipulate something.

Expressions can be as simple as a single value, such as `5` or `"Hello"`, or they can be more complex combinations of values and operators, like `2 + 3` or `x \* (y + 5)`.In simple words, expressions in Python are like instructions that tell Python what to do with the values and data you have, and they always produce a result when evaluated.

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

ANS :- An expression is a combination of values, variables, operators, and function calls that can be evaluated to produce a result. It's like a recipe that tells Python what calculation or manipulation to perform. When you run an expression, it returns a value. On the other hand, a statement is a complete instruction or action that Python can execute. It's like a command or a line of code that tells Python to do something. Statements can include assignments, loops, conditionals, function definitions, and more. The main difference between expressions and statements is that expressions always produce a value when evaluated, while statements don't necessarily have a value or return anything. Statements are used to perform actions, control the flow of a program, or change the state of variables.

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

bacon = 22

bacon + 1

ANS :- In the given code snippet, the line bacon = 22 assigns the value 22 to the variable bacon. However, the line bacon + 1 does not modify the value of bacon itself

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

'spam' + 'spamspam'

'spam' \* 3

ANS :- Both expressions ultimately evaluate to the string 'spamspamspam'.

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

ANS :- In Python, variable names need to follow certain rules and conventions. Here are the rules for naming variables in Python:

a. Variable names can contain letters (a-z, A-Z), digits (0-9), and underscores (\_).

b. Variable names must start with a letter or an underscore. They cannot start with a digit.

c. Variable names are case-sensitive. For example, `eggs` and `Eggs` are considered different variables.

d. Variable names should not be the same as Python keywords or built-in function names.

Based on these rules, `eggs` is a valid variable name because it starts with a letter and contains only letters and no reserved characters. It adheres to the naming conventions and is not a Python keyword.On the other hand, `100` is an invalid variable name because it starts with a digit (`1`). Variable names cannot start with digits; they must start with a letter or an underscore.

In summary, `eggs` is a valid variable name because it follows the rules for variable naming in Python, while `100` is invalid because it violates the rule of not starting with a letter or an underscore.

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

ANS :- int(): This function can be used to convert a value to an integer. It takes a number or a string representing a number as an argument and returns the corresponding integer.

float(): This function can be used to convert a value to a floating-point number. It takes a number or a string representing a number as an argument and returns the corresponding floating-point number.

str(): This function can be used to convert a value to a string. It takes any value as an argument and returns its string representation.

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

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

ANS :- The given expression `'I have eaten ' + 99 + ' burritos.'` causes an error because you are trying to concatenate a string (`'I have eaten '`) with an integer (`99`) directly using the `+` operator. In Python, you can only concatenate strings with other strings, not with integers.

To fix the error, you need to convert the integer `99` to a string before concatenating it with the other strings. You can do this by using the `str()` function to convert the integer to its string representation

Here's the corrected expression:

```python

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

```

In the corrected expression, `str(99)` converts the integer `99` to the string `'99'`, allowing it to be concatenated with the other strings.