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.

*****

This is an expression it is used to perform multiply or exponential.

'hello'

This is a string and it is a value.

-87.8

This is a floating-point number and it is a value.

-

This is an expression used in subtraction.

/

This is an expression again it is used to perform division.

+

This is an expression used to perform addition or concatenation.

6

This is a value and it is an integer.

2. What is the difference between string and variable?

Ans: **String** is a data type in python. And string represents through the single (‘ ’) or double(“ ”) quotation marks. If we put this quotation mark at any value like either it is a numeric value or any kind of alphabet so that value converts into the string. Mostly it is used for any written work like if we want to write some sentences or words in python for that we have to use this quotation marks and the sentences, words or letters are converted into string and python understand this thing and execute according to it.

Ex: **‘Ineuron’** or **“Ineuron”**, **‘123’**, **“0.9863”**, **‘I’**, this all are strings.

For converting any value in the string, we have two methods first one is to put the quotation marks on over the value or we can use the **str() function** this function is also to convert the value in the string.

Variable: Variable is a kind of container in python it stores or contains the value either it is a string, integer, floating-point number, Boolean, or any kind of data type it will store all of the values. For creating any variable, we have to write its name with the assignment operator that’s it, now our variable is ready to store any value.

Ex: **Name =** , **name =** , **add =** , **Ineuron =** , this all are variables.

There are certain things we have to remember when we creating a variable.

1. We can’t give any space between two names of variable if we want to create a variable like Roll No = , so instead of this, we have to create a variable like **Roll_No =** or **RollNo =** .

2. We can’t start a variable name with any number, if we create a variable like **21Roll_No =** in this case python throw an error. We can use the number after some alphabetical term like **Roll_No21 =** or **Roll_No_21 =** this is a genuine variable.

3. If we can’t use the assignment operator **(= sign)** after our variable name so python can not understand it and give an error. When creating a variable initially assign some value and if you want to create a null variable so you can create a variable like this Roll_No = ‘ ’ or Roll_No = 0 (**technically it’s not null they consist some values first consist of a string and the second one having an integer**) in this case python will understand it. And with the help of some operations, you can fill your null variables too.

3. Describe three different data types.

Ans: There are several data types in python and now we will discuss the basic one's:

1. **String** = String is a data type in python this data type represents through quotation marks with the help of this data type we can write some sentences, words, or letters in python and python also understand it. Ex: 'Name', "Place", 'I', "Course", 'Data Science', 'ML','123', '1.23'

2. **Integer** = Integer also a data type in python this data type is used for numerical value (**without decimal**) when we use the numbers in python those numbers are integer data types. Ex: 1, 23, 98897, 543, 000, 0

3. **Float** = Float or floating-point number is also data type in python basically this data type is used to represent the floating-point numbers generally a decimal number. When we put a decimal point in any integer it will become a float data type. Ex: 1.0, 23.4, 456.78, 0.1234, 12.432.

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

Ans: Basically, Expressions are a combination of variables, values, or mathematical operators. It generates single values, which by itself is an expression like if we write 4 + 3 so this is an expression. Expressions always return a value; they can print the result value as well. Example of expressions: **"Hello" + "INeuron"** is an expression, **5.0/4+(12-7)** it is also an expression**.**

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

Ans: We known expressions are the combination of variables, values, or operators. And if we talk about the statement so statements are like syntax or a structure means it follows their path, statement can't return or print any value without using a **print() function.** Examples of statements are **loops**, **classes**, **conditional statements**, **def**, **assignment statement**, and many more so they all follow their path, and if we want to know what they will throw as an output so we have to use the print function over there.

Consider this given example **spam = 10** it is an assignment statement so in this we can see that there is a spam variable and it has some value which is 10 and if we want to see the output on the screen so we use the print function. In simple words, expressions are those who will give their output on the screen directly, and in the case of statements, you have to take some more steps to get the output on the screen.

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

```
bacon = 22
bacon + 1
```

Ans: After executing this program bacon contains the same value which is 22.

In this program, we assign 22 to the bacon variable and in the next line, we increment the bacon by 1, if we print the next line so we will get 23 but we do not initialize this in our main variable like **bacon = bacon + 1** that's why bacon will not change and have the same value as previous.

7. What should the values of the following two terms be?
'spam' + 'spamspam'
'spam' * 3

Ans: The answer is same for both the line. In the first line, we concatenate the string with **+** operator so it will throw **spamspamspam** output. And in the second line we just multiply our string **spam** by 3. So, it will also concatenate the string by 3 times and give the same output as **spampamspam.**

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

Ans: Because there is a rule in python, we cannot make a variable that starts from any numeric value and **100** is a numeric value that's why it is invalid and eggs is valid. We can use numeric value after some alphabetical terms like **eggs100** and this acceptable and valid in python.

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

Ans: **int()** this function is used to convert into integer form**.**
float() this function is used to convert in floating-point number**.**
str() this function is used to convert any value in string form**.**

There are certain things we should have to remember like we cannot convert string data type into integer or float if the string contains an alphabetical value. It's a basic logic like we cannot convert '**A' into integer** or '**data' into an integer or a floating-point number** and so on. And if the string contains any numerical value like **'123'** so it will be converted into an integer or a float. And we can convert an integer and float into the string. Like 123 can be converted into string format and 1.23 also able to convert into a string.

10. Why does this expression cause an error? How can you fix it?
'I have eaten ' + 99 + ' burritos.'

Ans: This expression will try to concatenate the string with integer and that's the reason the program throws an error because we cannot concatenate some other data type to string value. If we want to remove that error so we will convert this 99 (**integers**) into the string and for that we have to put a quotation mark on 99 or we will use the str() function.

Two possible ways to remove that error:

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

Or

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