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

The operators are +, -, \*, and /. The values are 'hello', -87.8, and 6

**2. What is the difference between string and variable?**

A Variable is a store of information, and a String is a type of information you would store in a Variable. A String is usually words, enclosed with ""

Eg String x ="Renuka thejeshwini" X is the Variable, and we declared it as a String, use the single = to assign the text to it.

**3. Describe three different data types.**

### Integer (int)

It is the most common numeric data type used to store numbers without a fractional component (-707, 0, 707).

### Boolean (bool)

It represents the values true and false. When working with the boolean data type, it is helpful to keep in mind that sometimes a boolean value is also represented as 0 (for false) and 1 (for true).

### Floating Point (float)

It is also a numeric data type used to store numbers that may have a fractional component like monetary values do (707.07, 0.7, 707.00).

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

A combination of operands and operators is called an **expression**. The expression in Python produces some value or result after being interpreted by the Python interpreter. An expression in Python is a combination of operators and operands.

An example of expression can be : x = x + 10*x*=*x*+10. In this expression, the first 1010 is added to the variable x. After the addition is performed, the result is assigned to the variable x.

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

In programming language terminology, an “expression” is a combination of values and functions that are combined and interpreted by the compiler to create a new value, as opposed to a “statement” which is just a standalone unit of execution and doesn’t return anything. One way to think of this is that the purpose of an expression is to create a value (with some possible side-effects), while the sole purpose of a statement is to have side-effects

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

**bacon = 22**

**bacon + 1**

the bacon variable is set to 22. The bacon +1 expression does not reassaign the value in bacon

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

**'spam' + 'spamspam'**

**'spam' \* 3**

Both expressions evaluate to the string

‘spamspamspam’

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

Variable names cannot begin with numbers

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

The int(), float(), and string() functions will evaluate to the integer floating\_point number, and string versions of the value passed to them.

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

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

The expression causes an error because 99 is an integer, and only strings can be concatenated to other strings with the +operator. The correct way is I haven eaten ' + 99 + ' burritos’.