**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) Multiplication Operator

**'hello'** (Value) String

**-87.8** (Value) Negative Integer

**-** (Expression) Subtraction Operator

**/** (Expression) Division Operator

* (Expression) Addition Operator

**6** (Value) Positive Integer

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

**Sol. Variable:** A variable is basically name assigned to a location in primary memory where we can store values such as integers, strings etc. Ex: **a = 5** (Here **a** is a variable which is assigned integer value 5).

**String:** String in python is a data type which is basically a series of characters quoted in “ “. Ex: “abcdeggfh”. A string can be stored in a variable like **(b = “Hello”)**

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

**Sol.** Data types are the types of data in python that we can assigned to variables and store in memory. Ex: 1). **Numeric** (Integers, float, Complex numbers): **Integers** are +ve /-ve integer values like -5 or 4. **Float** are numbers like 5.89 or -6.789.

2). **Boolean:** True/False

3). **Sequence** (Strings, Lists, Tuples): **Strings**: “Amit”, “Hello World” etc. **Lists:** [5,”a”,[6,7]] (Lists can contain other data types within [] and they are mutable). **Tuples:** (5,”a”,[6,7]) (Tuples are similar to lists enclosed within () but they are immutable).

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

**Sol.** An expression is made up of operators **( +, -, /, \*, %)** and operands (values like 5,6, “abc”). Ex: 5 + 6, “a” + “b”, 10 – 5 etc. All expressions are interpreted to produce some other value. Ex: 5 + 6 will result in 11, “a” + “b” will result in ab.

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

**Sol.** The evaluation of an expression like **5\*6** will always returns a result value like in this case **5\*6** will result in **30.** On the other hand, a statement may or may not returns a result value and it will only do what the statement states like **spam = 10** will assign value **10** to the variable **spam.**

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

bacon = 22 (value 22 will be assigned to variable **bacon**)

bacon + 1

**Sol.** Variable **bacon** will contain value 22 after running the code.

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

'spam' + 'spamspam' **Sol.** spamspamspam

'spam' \* 3 **Sol.** spamspamspam

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

**Sol.** There are certain rules in python to define a variable name:

* The first character of the variable should be an alphabet or (\_) underscore.
* Special characters (@, #, %, ^, &, \*) should not be used in variable name.
* Variable names are case sensitive. For example - eggs and EGGS are two different variables.
* Reserve words like **break, class, for** cannot be declared as variables.

So, based on the above mentioned rules **eggs** is a valid variable name while **100** is invalid.

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

**Sol. int()** : This function is used to get the integer version of a value.

**float() :** This function is used to get the floating-point number version of a value.

**str() :** This function is used to get the string version of a value.

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

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

**Sol. 'I have eaten '** is a string and **' burritos.'** is a string but **99** is an integer. Strings cannot be concatenated with integers but only with strings. That’s why the above expression cause an error. We can fix the above expression by converting 99 into string with the help of str() function. Ex: **‘I have eaten’ + str(99) + ‘burritos’.**