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

**Solution:**

In the above example the elements ‘hello’, -87.8 and 6 are values whether the elements \*, -, /, + are expressions

2. What is the difference between string and variable?

**Solution:**

A **string is a value representing text. a variable is a name that can refer to any value**. quotes, double or single, (they mean the same) are used to create string literals, the quotes are there to indicate that the text that they enclose is not code, it is a value. A variable can contain any data type value

E.g. A = 34 or a = ‘string’ in this case A and a are called as variable and the value ‘string’ is known as string

3. Describe three different data types.

**Solution:**

i)Numeric:

The data types ‘int,’ ‘float,’ ‘complex’ are the numeric type data types the represents numeric values.

**e.**g. type (3) = int,

type (3.5) = float,

type(4+5j) = complex

ii) Sequential:

The data types ‘string’, ‘list’ and ‘tuple’ are known as sequential data types they store value inside in particular order so that anyone can access it by using index of position. String and list are mutable data types whereas tuple is immutable.

e.g. [3, ‘hello’, None] is list datatype and (3, ‘hello’, None) is tuple data type. They can contain any type data values and values are accessible by using their indices. we can modify list but we cannot modify tuple.

# iii) Boolean:

Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsie (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.

**Note**– True and False with capital ‘T’ and ‘F’ are valid Booleans otherwise python will throw an error.

e.g. type (‘True’), type(‘False), 98+2==101

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

**Solution:**

Expression is made of values and operators to give a datatype value as an output after evaluating the expression

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

Solution:

The difference is that in expression format it will assign value 10 or store the value 10 inside variable name ‘spam’ so that we can refer a value 10 as spam during coding and in case of statement format it just mean that we have information that spam is equal to 10.

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

bacon = 22

bacon + 1

**Solution:**

After running the above code the variable bacon will contain updated value bacon = 23

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

'spam' + 'spamspam'

'spam' \* 3

**Solution:**

In Both cases the output value should be same as spamspamspam, we are performing same operation using different operators

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

**Solution:**

In python while assigning a variable we should take care of that a variable name will not contain a python inbuilt keywords like list, int, def and also, we cannot initiate a variable name with int datatype. In above the 100 is representing int data type therefore we can not use it as variable name, instead we can assign variable name like \_100 or 100a

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

**Solution:**

In python we can do type casting of a value by using python inbuilt function.

e.g., we can get the 100 values in 3 diff. format by using int(100) , float(100) ,str(100)

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

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

**Solution:**

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

We are getting error because cannot concatenate string data type with int value. Instead we can perform str to str concatenation by using following code

'I have eaten ' + ‘99’+ ' burritos.'