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:**

Following can be considered as expressions:

'hello'

-87.8

6

2. What is the difference between string and variable?

**Ans:**

A string is python data type which can hold characters. It can be defined using single, double and triple quotation marks. Ex. ‘satish’, “python 3.11.4”, “””Hey there”””.

A variable is representative name and is a pointer towards an object. There are certain rules on how to define a name of variable.

Valid variable name: satish\_, satish1

Invalid variable names: 123satish, #satish

3. Describe three different data types.

**Ans:**

**Int**: Int data type can hold integer values. Function int() can be used to type cast the float. For type casting, a string containing full integers can be done as well. However, a string containing float values throws an error.

Ex. ‘3478’ 🡪 int(‘3478’) 🡪 outputs 3478

‘3.4’ 🡪 int(‘3.4’) 🡪 throws an error

Int Ex. 1, 44, 76, 7776.

**String**: String data type can hold characters. They must be defined using single, double and triple quotation marks. Strings are of immutable type meaning once created their values cannot be changed.

Ex. ‘satish’, ‘python 3.11.4’, ‘6’, “3.142”

**Tuple**: Tuple data type can hold multiple values and each value can be called its elements. They must be defined using parenthesis around the values separated by comma.

Elements of tuple can be of any data type. Tuples are of immutable nature meaning their values cannot be changed once it is created.

Ex. (1,), (1,2,5,67,4,4,3), (1, ‘fatigue’, 3.33, [1,4,6.6])

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

**Ans:**

Expressions in python are made of data types and operators. However, expression without operators is also valid. +, -, \*, /, \*\*, %, &, >, <, >=, <=, ~ are some of the operators. In the expression, parenthesis can be used to segregate some part of expression from other. Variable names can be used in expressions.

Ex. (1+3) / (5.5 + x\_new)

2

6.655

2 > 3

‘This is python programming syntax’ + str(5.5556) + f’ Value is: {x\_new}’

numpy.sum(numpy.average(x\_new, axis=0))

All expressions evaluate to a single data type. If operators are in the expression, these operators operate on objects in the sequence they present.

Ex. numpy.sum(numpy.average(x\_new, axis=0)) 🡺 evaluates to float

3/2.3 🡺 evaluates to float

4555 🡺 evaluates to int

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

**Ans:**

spam = 10 is expression as it evaluates to a single data type, here, 10 i.e., integer.

An expression is made of variables and operator which together after defined operations evaluates to a single data type. However, a statement is made for doing a task, ex. If condition used in a task where we want to execute certain code after it satisfies some condition. Another examples could be while loop, for loop, etc.

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

bacon = 22

bacon + 1

**Ans:**

22

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

'spam' + 'spamspam'

'spam' \* 3

**Ans:**

'spamspamspam'

‘'spamspamspam '

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

**Ans:**

Variable names must start from alphabet character (uppercase or lowercase) or single or double underscore (\_).

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

**Ans:**

int() 🡪 to get integer

float() 🡪 to get float

str() 🡪 to get string

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

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

**Ans:**

Above expression gives error because integer (99) cannot be added to a string. Hence, we must type cast it to the string as below,

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