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.

Ans> \*, -, /, +, ‘hello’ are expression and ‘hello’, -87.8, 6 are values.

What is the difference between string and variable?

Ans> Strings are arrays of bytes representing Unicode characters, where are variable is a symbolic name given to a pointer to an object.

Describe 3 data types?

Ans>

1. Strings
   1. Strings are array of bytes representing Unicode characters.
   2. ‘welcome’ or ‘hello world’ are examples of strings.
   3. Strings can be accessed using square brackets.
   4. Strings support slicing to access a range of characters in the string.
   5. Strings are immutable i.e., updation or deletion of characters from a string is not allowed.
   6. To use a special character as a value string we need to use \ so that the special character next to it will be treated as a normal character.
   7. We can format the string using the format() method. We use braces as place holders which can hold arguments according to position to specify the order.
2. List
   1. List are used to store multiple items in a single variable.
   2. It is one of the 4 built-in data types in python used to store data.
   3. Example: listnew = [‘this’, ‘is’, ‘list’]
   4. In list duplication is allowed.
   5. List in mutable so values can be changed.
   6. The are ordered in nature due to which they can be accessed using index.
   7. Can contain heterogeneous data.
3. Dictionaries
   1. These are used to store data in kay value pairs.
   2. It an ordered collection of data.
   3. Dictionaries are mutable.
   4. They do not allow duplicates.
   5. Example- dictnew = {“key1”: “value1”, “key2”:”value2”}
   6. This also supports heterogeneous values.

What is an expression made up of? What do all expression do?

Ans> An expression is made up of combination of operators and operands which are used to produce some value.

What expressions do is they represent a value which is usually the result produced by the interpretation of the expression. However, a string is also an expression since it represents the value of the string as well.

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

Ans> Statements are instructions that a python interpreter can execute whereas the expression is used for representation of value. For example spam = 10 given above is an assignment statement not a expression where as a = a\*b is a statement with a\*b as an expression.

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

Bacon = 22

Bacon + 1

Ans> In bacon + 1 no assignment operator is used therefore the value will remain the same as previously declared i.e., 22

What should the values of the following two terms be?

‘spam’ + ‘spamspam’

‘spam’ \* 3

Ans> first value will be ‘spamspamspam’ as we are adding 2 string ‘spam’ and ‘spamspam’, and we know addition of string means concatenation.

The second value will also be ‘spamspamspam’ but the way it is carried out is different than simple concatenation, here it like we need a value that is 3 time the supplied value for a string it will mean a result with 3 time the string value.

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

Ans> in python the variable name should start with a letter or an underscore due to which eggs is a valid variable name where as 100 is not because it starts with a integer which is neither a underscore nor a letter.

What 3 functions can be used to get the integer, floating-point number, or string version of a value?

Ans> we can use the type casting functions to performed the operation.

1. To obtain the integer version put the values inside int() function
2. To obtain the floating-point number put the values inside float() function.
3. To obtain the string number put the values inside str() function.

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

‘I have eaten ‘ + 99 + ‘ burritos.’

Ans> This produces an error as the values involved are not compatible meaning the different datatypes in the expression cannot be implicitly converted to higher datatype for performing the addition operation due to which it throws an error. The fix this all we need to do is make 99 by explicit type casting thus making 99 string and hence the whole expression is now composed of strings.

‘I have eaten ‘ + str(99) + ‘ burritos.’