**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.**

\* : Multiplication Operator, Expression

'hello' : String ‘hello’, Value

-87.8 : Integer, Value

- : Subtraction Operator, Expression

/ : Division Operator, Expression

* : Sum Operator, Expression

6 : Integer, Value

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

The character sequences (single or multiple characters) like ‘Hello’, ‘a’ or even numbers represented within ‘’ like ‘234’ are known as strings. Python sees the strings as a sequence of characters. String is a datatype.

On the other hand, a variable is the identifier, an allocation to which a certain value can be assigned. A Python variable is a reserved memory location to store values. The variable can be assigned names which are alphabetical or alphanumeric. Numerical values cannot be used as variable names.

Examples of a variable assignment, here a, b and c are the variables. These are assigned the values of 10, ‘Hello’ and b\*a respectively. ‘Hello’ is a string.

a=10

b=’Hello’

c=b\*a

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

There are many datatypes available in python which can store different types of data

* Character based data can be stored using ‘string’ datatypes
* Numerical data can be stored using ‘int’ (integers) or ‘float’ (floating point numbers)
  + Complex Numbers can be stored using ‘complex’ datatype
* Boolean data in terms of TRUE or FALSE (logical conclusions)

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

An expression is a piece of code that evaluates something. It could be a mathematical operation like performing a sum of two variables or constants or just printing a simple output.

Expressions are made of values (operands) and operators. The operators could be logical or numerical. An expression could be just a constant number with no operations performed without any operator or variable. It is just something which is available in the code that gets evalauated.

Example of an expression in a statement to evaluate sum of two numbers

a=10

b=3

c=a+b

print(f‘Sum of c is {c}’)

In the code snippet above, the values 10, 3, sum equation and print command are all expressions; each expression is assigned to certain variables. Each line with a combination of variable, expression and the corresponding assignment is called a statement.

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

An expression could be anything, just anything valid and understandable by Python. While, statement is the complete line where some evaluation happens, and this evaluation gets assigned to some variable. However, assignment is not a mandatory requirement for a line of code to be statement

In the example provided

spam=10

The value 10 by itself is an expression, but the complete line which initialises a variable with name ‘spam’ and assigns it the value of 10 is a statement. Any chunk of code that gets evaluated is called an expression.

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

**bacon = 22**

**bacon + 1**

bacon still contains 22.

**Explanation**: The first line, bacon =22 is an assignment statement which assigns the value of the integer expression 22 to the variable bacon. The second statement “bacon+1” is just an expression without any assignment; hence the original variable remains unaffected.

*However, if the statement had been: bacon+=1*

*Usage of augmented operator updates the value in the variable. The new value would be 23.*

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

'spam' + 'spamspam'

'spam' \* 3

Both expressions would be evaluated to generate the same : ‘spamspamspam’

When an integer is multiplied to a string (‘spam’), the string gets replicated those many times

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

A variable is the identifier, an allocation to which a certain value can be assigned. A Python variable is a reserved memory location to store values. The variable can be assigned names which are alphabetical or alphanumeric. Numerical values cannot be used as variable names.

Numerical values like 100 is interpreted by python as an expression and hence cannot be used a variable name.

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

int(value) converts the value to integer

float(value) converts the value to a floating point number

str(value) converts the value to a string

However, characters cannot be converted to numerical data types using int() or float(). However, a number can be converted to a string of characters using str().

Example:

int(‘a’) is invalid

str(234) is valid, it evaluates as ‘234’

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

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

Addition operator with strings represents concatenation. Strings can be concatenated with strings only. In the code above, 99 is evaluated by python as a numerical value as ‘’ is unavailable.

Solution 1: Add ‘’ to 99

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

Solution 1: Use str() on 99

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

Both will evaluate and generate the same: ‘I have eaten 99 burritos’