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

**Answer - \* - operator (multiplication)**

**'hello' - value (string)**

**-87.8 - value (floating point number)**

**- operator (subtraction/negation)**

**/ - operator (division)**

**+ - operator (addition)**

**6 - value (integer)**

2. What is the difference between string and variable?

**Answer –**

**A string is a collection of letters, numbers, and symbols enclosed in quotes, like "hello" or "123". It represents a specific piece of information.**

**A variable, on the other hand, is a container that can hold different types of information, including strings. It has a name and can be assigned a value, like name = "John". This means that the variable named name now contains the string value "John".**

**In summary, a string is the information itself, while a variable is a way to store and access that information.**

3. Describe three different data types.

**Answer –**

**Integer: An integer is a whole number without a fractional part, such as -3, -2, -1, 0, 1, 2, 3. It is a data type that represents numerical values that can be used in mathematical operations like addition, subtraction, multiplication, and division.**

**String: A string is a sequence of characters, such as letters, numbers, and symbols, enclosed in quotes. Examples of strings are "Hello, World!", "12345", or "This is a string". Strings are often used to represent textual information.**

**Boolean: A boolean is a data type that represents logical values. It can have one of two values: True or False. Booleans are often used in programming to control the flow of code, as they allow the program to make decisions based on whether a condition is true or false. For example, a program might check if a certain condition is true before executing a certain block of code.**

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

**Answer –**

**An expression is like a math problem, it's made up of numbers, symbols (like plus or minus), and sometimes letters or words.**

**When you solve the math problem, you get a value, and the same thing happens when you evaluate an expression, you get a value as the result.**

**Expressions can be used to do things like add numbers together, compare values, or manipulate text, and they're an important part of programming.**

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

**Answer –**

**In programming, an expression is a piece of code that produces a value. For example, 5 + 3 is an expression that produces the value 8.**

**On the other hand, a statement is a line of code that performs some action, such as assigning a value to a variable or calling a function. For example, spam = 10 is a statement that assigns the value 10 to the variable named spam.**

**So, the main difference between an expression and a statement is that an expression produces a value, while a statement performs an action. An expression can be a part of a statement, but a statement cannot be a part of an expression. In the example spam = 10, 10 is an expression that produces the value 10, and the entire line spam = 10 is a statement that assigns that value to the variable spam.**

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

bacon = 22

bacon + 1

**Answer –**

**After running the code, the variable bacon would still contain the integer value 22.**

**This is because the expression bacon + 1 does not modify the value of bacon itself; it only calculates the result of adding 1 to bacon. If you wanted to modify the value of bacon to be 1 greater than its current value, you could reassign it like this:**

**bacon = 22**

**bacon = bacon + 1**

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

'spam' + 'spamspam'

'spam' \* 3

**Answer –**

**'spam' + 'spamspam' would result in the string 'spamspamspam'. This is because the + operator when applied to two strings concatenates them together, joining the first string with the second string.**

**'spam' \* 3 would result in the string 'spamspamspam'. This is because the \* operator when applied to a string and an integer repeats the string the specified number of times. In this case, 'spam' is repeated 3 times to give the same result as the previous expression.**

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

**Answer –**

**In Python and many other programming languages, variable names must follow certain rules in order to be considered valid.**

**One of the basic rules for naming variables in Python is that the name must begin with a letter (either uppercase or lowercase) or an underscore (\_). After the first character, the name can contain letters, numbers, and underscores, but cannot begin with a number. Based on this rule, eggs is a valid variable name because it begins with a letter (e) and contains only letters. On the other hand, 100 is not a valid variable name because it begins with a number (1) and does not follow the rule that the first character must be a letter or an underscore.**

**However, it's worth noting that 100 could be used as a value assigned to a variable with a valid name, such as x = 100. In this case, x is a valid variable name and 100 is a valid value assigned to that variable.**

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

**Answer –**

**In Python, the following three built-in functions can be used to get the integer, floating-point number, or string version of a value:**

**int() - This function can be used to convert a value to an integer. For example, int(3.14) would return the integer 3.**

**float() - This function can be used to convert a value to a floating-point number. For example, float(3) would return the float 3.0.**

**str() - This function can be used to convert a value to a string. For example, str(3.14) would return the string '3.14'.**

**It's worth noting that these functions may not work for all types of values, and may raise errors if they are applied to incompatible types. For example, calling int('hello') would raise a ValueError because 'hello' cannot be converted to an integer.**

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

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

**Answer –**

**This expression would cause a TypeError because you are trying to concatenate a string and an integer. Specifically, the expression 'I have eaten ' + 99 is trying to concatenate the string 'I have eaten ' with the integer 99, which is not allowed.**

**To fix this, you could convert the integer to a string using the str() function before concatenating it with the other string. For example: 'I have eaten ' + str(99) + ' burritos.'**

**This would convert the integer 99 to the string '99', which can then be concatenated with the other string using the + operator. The resulting expression would be a string: 'I have eaten 99 burritos.'**