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 1. \* , -, / are expressions

'hello', -87.8, 6 are values

2. What is the difference between string and variable?

Ans 2. A Variable is a store of information, and a String is a type of information you would store in a Variable. A String is usually words, enclosed with ""

3. Describe three different data types.

Ans 3. brief descriptions of three different data types in Python:

Integer (int): The integer data type represents whole numbers without any decimal points. It includes positive numbers, negative numbers, and zero. Integers are commonly used for counting, indexing, and performing mathematical operations.

Float: The float data type is used to represent decimal numbers. It is useful for handling real numbers and performing calculations involving fractions or decimal values. Floats are often used in scientific and mathematical computations.

String (str): The string data type is used to represent sequences of characters. It is commonly used to store and manipulate text in Python. Strings can contain letters, digits, symbols, and special characters. They are often enclosed in quotes (single, double, or triple) to denote their beginning and end.

These three data types have distinct properties and are used in various scenarios while programming in Python. Understanding their characteristics is essential for effectively working with different kinds of data in Python.

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

Ans 4. An expression in Python is a combination of values, variables, operators, and function calls that, when evaluated, produces a result. Expressions can involve various components and operators to perform computations and manipulate data.

Expressions are used to perform tasks such as mathematical calculations, string operations, logical evaluations, and more. They can combine different types of values and variables to produce a new value. For example, you can add two numbers together, concatenate two strings, or evaluate a logical condition.

Expressions can consist of literals, which are fixed values like numbers or text, variables that hold values, and operators that perform operations on the values. The operators can be arithmetic operators for mathematical computations, comparison operators to compare values, logical operators for boolean evaluations, and more.

Additionally, expressions can involve function calls, where you invoke predefined or custom functions to perform specific tasks. The functions can take arguments as inputs, and their return values can be part of an expression.

When an expression is evaluated, the Python interpreter processes the expression and calculates or resolves it to produce a value. This resulting value can be stored in a variable, used in further computations, or directly displayed or used in program logic.

In summary, expressions in Python are combinations of values, variables, operators, and function calls that are evaluated to produce a result. They allow for computations, data manipulation, and decision-making within a program.

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

Ans 5. In Python, expressions and statements are both fundamental components of the language, but they serve different purposes:

Expression: An expression is a combination of values, variables, operators, and function calls that, when evaluated, produces a value. It can be as simple as a single value or more complex with multiple components and operators. Expressions are used to perform computations and produce results. For example, 2 + 3 is an expression that evaluates to 5. Expressions can be part of a larger statement or used independently.

Statement: A statement is a unit of code that performs an action or carries out a specific task. It is a complete instruction that can include expressions and other constructs. Statements can modify variables, control program flow, define functions or classes, and more. They are the building blocks of a program's logic and structure. For example, an assignment statement like spam = 10 assigns the value 10 to the variable spam.

The key difference between expressions and statements is that expressions produce a value, whereas statements do not. Expressions are evaluated to compute a result, which can be used or stored for further use. On the other hand, statements do not produce a value directly but perform an action or modify the program's state.

In summary, expressions are used to calculate values, while statements are used to control the flow and behavior of a program by performing specific tasks or actions. Expressions can be part of statements, but not all statements contain expressions.

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

bacon = 22

bacon + 1

Ans 6. 22

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

'spam' + 'spamspam'

'spam' \* 3

Ans 7. 'spamspamspam'

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

Ans 8. In Python, variable names need to follow certain rules and conventions. Here are the reasons why 'eggs' is a valid variable name while 100 is invalid:

Starting character: Variable names in Python must start with a letter (a-z, A-Z) or an underscore (\_). In this case, 'eggs' starts with a letter 'e', which satisfies this rule.

Subsequent characters: After the first character, variable names can contain letters, digits, or underscores. However, they cannot start with a digit. 'eggs' only contains letters, which makes it a valid variable name.

Avoiding reserved words: Python has reserved words, also known as keywords, that have special meanings in the language. These keywords cannot be used as variable names. For example, if, for, while, and print are reserved words. 100 is a numeric literal and not a valid variable name because it is not allowed to use numeric literals directly as variable names.

Therefore, while 'eggs' satisfies the rules for variable names by starting with a letter and containing only letters, 100 is invalid as it starts with a digit and violates the naming rules.

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

Ans 9. Certainly! Here are three functions that can be used to obtain the integer, floating-point number, or string version of a value in Python:

int(): The int() function can be used to convert a value to an integer. It takes a numeric or string input and returns the corresponding integer representation. If the value cannot be converted to an integer, a ValueError is raised.

float(): The float() function is used to convert a value to a floating-point number. It accepts numeric or string inputs and returns the equivalent floating-point representation. If the value cannot be converted to a float, a ValueError is raised.

str(): The str() function is employed to convert a value to a string representation. It takes any input value, such as an integer, float, Boolean, or other data types, and returns the corresponding string representation.

These functions are essential for data type conversion and enable you to manipulate and process values in different formats based on your programming needs.

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

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

Ans 10. 'I have eaten ' + str(99) + ' burritos.'