1. In the below elements which of them are values or an expression? eg:- values can be integers or strings and expressions will be mathematical operators.
*
'hello'
-87.8
-
/
+
6

Answer
* : Operator (Multiplication)
'Hello' : Value (String)
-87.8 : Value (Float)
- : Operator (Subtraction)
/ : Operator (Division)
+ : Operator (Addition)
6 : Value (Integer)

2. What is the difference between string and variable?

Answer
string - Data type used to represent text. It is a sequence of characters enclosed in single, double, or triple quotes.
Example: **`"Hello, World!"`**

variable - Symbolic name or identifier associated with a value or data. It is a storage location in a computer program where you can store and manipulate data.
Example: **`name = "John"`**, here name is variable storing the string value **`"John"`**

In summary, a string is a specific type of data (text), while a variable is a placeholder or container for storing different data types, including strings.
In the example, the variable name can store various string values at different points in the program.

3. Describe three different data types.

Answer

Integer (int):
Data type that represents whole numbers without any decimal points. Integers can be positive or negative and include zero.
Examples: -5, 0, 42

Float (float):
Short for floating-point numbers is a data type that represents numbers with decimal points or in exponential form. Floats allow for the representation of a wide range of real numbers.
Examples: 3.14, -0.001, 2.5e2 (which is equivalent to 250.0)

String (str):
Data type that represents a sequence of characters typically used for text. Strings are enclosed in single, double, or triple quotes.
Examples: "Hello, World!", 'Python', "123"

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

Answer

An expression is made up of a combination of operators, variables, literals, and function calls that result in a value. In simpler terms, it's a set of instructions that, when executed, produces a result. Expressions can be as simple as a single variable or as complex as a combination of multiple operators and operands

All expressions, when evaluated, result in a value. This value can be of various types, such as numbers, strings, booleans, or objects. Expressions are the building blocks of algorithms and logic in programming. They allow developers to manipulate data, make decisions, and perform various computations within a program

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

Answer
An expression is a combination of literals, variables, operators, and/or function calls that evaluates to a value.
Examples: 5 + 3, x * 2, len("hello").

A statement is a complete line of code that performs an action. It may include expressions.
Examples: Assignment statement - spam = 10, Conditional statement - if x > 0

Difference:
Result: An expression always evaluates to a value. For example, 5 + 3 evaluates to 8. A statement may or may not produce a value. For instance, an assignment statement (spam = 10) does not produce a value that can be used elsewhere.
Purpose: Expressions are mainly used to represent computations or calculations. Statements are used to perform actions, control flow, or define structures in a program.
Usage: Expressions are often used within statements. For example, an expression can be part of an assignment statement (spam = 10), a conditional statement (if x > 0:)

6. After running the following code, what does the variable bacon contain?
bacon = 22
bacon + 1

Answer
After running the given code, the variable bacon will still contain the original value assigned to it, which is **22**.
The expression bacon + 1 calculates the sum of the current value of bacon and 1, but this result is not stored back into the variable bacon

7. What should the values of the following two terms be?
'spam' + 'spamspam'
'spam' * 3

Answer
'spam' + 'spamspam': Concatenation of strings. The result is **'spamspamspam'**
'spam' * 3: Repetition of string 3 times. The result is **'spamspamspam'**

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

Answer
In Python, variable names follow specific rules to ensure clarity and consistency in code. A valid variable name must commence with a letter (either uppercase or lowercase) or an underscore. Subsequent characters can include letters, digits, or underscores, but the name must not initiate with a digit.
For instance, the variable name eggs is deemed valid as it starts with the letter 'e' and consists of subsequent letters. On the contrary, the name 100 is invalid because it initiates with a digit, contravening the stipulation that variables cannot commence with a numerical value. Therefore, while eggs conform to the established conventions, 100 runs afoul of the prescribed rules for variable naming in Python

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

Answer
Integer Conversion: The int() function can be used to convert a value to an integer.
Example: **`int(3.14)`** will return 3.

Floating-Point Conversion: The float() function can be used to convert a value to a floating-point number.
Example: **`float(5)`** will return 5.0.

String Conversion: The str() function can be used to convert a value to a string.
Example: **`str(42)`** will return the string '42'.

10. Why does this expression cause an error? How can you fix it?
'I have eaten ' + 99 + ' burritos.'

Answer
The expression 'I have eaten ' + 99 + ' burritos.' causes an error because it involves attempting to concatenate a string with an integer without explicitly converting the integer to a string. In Python, the + operator is overloaded for string concatenation, but all operands must be of the string type.

To fix this, you need to convert the integer 99 to a string before concatenating
`'I have eaten ' + str(99) + ' burritos.'`