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:**

**Values:**

**'hello' (string value)**

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

**6 (integer value)**

**Expressions:**

**\* (multiplication operator)**

**- (subtraction operator)**

**/ (division operator)**

**+ (addition operator)**

2. What is the difference between string and variable?

**Answer:**

**String:**

**A string is a data type that represents a sequence of characters.**

**It is used to store and manipulate textual data such as words, sentences, or even empty spaces.**

**Strings are typically enclosed in quotation marks, either single (' ') or double (" ").**

**Examples of strings: "Hello, world!"**

**Variable:**

**A variable is a named storage location in a program that can hold a value.**

**It is used to store and refer to different types of data, including strings, numbers, and more complex data structures.**

**Variables allow you to assign values to them and refer to those values by their name throughout the program.**

**Examples of variables:**

**name = "John" (a variable named "name" storing the string "John").**

**age = 25 (a variable named "age" storing the integer 25).**

**pi = 3.14159 (a variable named "pi" storing the floating-point number 3.14159).**

3. Describe three different data types.

**Answer:**

1. **Integer (int):**

**The integer data type represents whole numbers without any fractional or decimal part.**

**Examples of integers: -5, 0, 42, 1000.**

**Integers can be positive or negative.**

1. **String (str):**

**The string data type represents sequences of characters, typically used for storing textual information.**

**Strings are enclosed in quotation marks (single or double) to differentiate them from other data types.**

**Examples of strings: "Hello, World!", "OpenAI", "42", "Python".**

1. **Floating-point Number (float):**

**The floating-point data type represents numbers that have both an integer and fractional part.**

**Examples of floating-point numbers: 3.14, -0.5, 2.71828.**

**Floating-point numbers can represent both positive and negative values, as well as numbers with decimal places.**

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

**Answer:**

**Constants: Fixed values such as numbers or strings. For example, "hello", 42, or 3.14.**

**Variables: Named containers that hold values. They can represent different types of data, including numbers, strings, or more complex objects. For example, x, name, or count.**

**Operators: Symbols that perform specific operations on the elements of an expression. Examples of operators include arithmetic operators (+, -, \*, /), comparison operators (>, <, ==), logical operators (and, or, not), and more.**

**Function calls: Invoking predefined functions or user-defined functions to perform specific operations or computations. For example, sin(x), len(string), or my\_function(arg1, arg2).**

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

**Answer:**

**The key difference between an expression and a statement lies in their functionality and purpose within a programming language:**

**Expression:**

**An expression is a combination of values, variables, operators, and function calls that can be evaluated to produce a single value.**

**It represents a computation that yields a result and can be used as part of a larger expression or as an argument to a function.**

**Examples of expressions:**

**2 + 3 (evaluates to 5)**

**x + 5 (evaluates to the value of variable 'x' plus 5)**

**len("hello") (evaluates to the length of the string "hello")**

**Statement:**

**A statement is a complete instruction or command in a programming language that performs an action or changes the program's state.**

**Unlike expressions, statements do not produce a value.**

**Examples of statements:**

**Assignment statement: spam = 10 (assigns the value 10 to the variable 'spam')**

**Conditional statement (if-else):**

**bash**

**Copy code**

**if x > 0:**

**print("Positive")**

**else:**

**print("Non-positive")**

**Loop statement (for loop):**

**scss**

**Copy code**

**for i in range(5):**

**print(i)**

**In summary, expressions are used to compute values, while statements are used to perform actions, control program flow, or change the program's state. An assignment statement, like spam = 10, is a specific type of statement that assigns the value 10 to the variable 'spam'. Statements do not produce a value, but they execute actions that can have side effects on the program's behavior or data. In contrast, expressions are used to calculate values that can be used in various ways within a program.**

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

bacon = 22

bacon + 1

**Answer:**

**After running the code:**

**makefile**

**Copy code**

**bacon = 22**

**bacon + 1**

**The variable bacon will still contain the value 22.**

**In the second line, bacon + 1 is an expression that adds 1 to the value of bacon. However, this expression does not modify the value of bacon itself or assign the result back to the variable. Therefore, the value of bacon remains unchanged at 22. If you want to update the value of bacon to the result of the expression, you would need to reassign it explicitly, like this:**

**python**

**Copy code**

**bacon = bacon + 1**

**Or using the shorthand assignment operator:**

**python**

**Copy code**

**bacon += 1**

**In either case, the value of bacon would become 23.**

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

'spam' + 'spamspam'

'spam' \* 3

**Answer:**

**'spam' + 'spamspam': The result would be the string 'spamspamspam', as the second string 'spamspam' is appended to the first string 'spam'**

**'spam' \* 3: The result would be the string 'spamspamspam', as the string 'spam' is repeated three times.**

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

**Answer:**

**Start with a letter or underscore:**

**Valid variable names must start with a letter (a-z, A-Z) or an underscore (\_).**

**In this case, 'eggs' starts with the letter 'e', which satisfies this requirement.**

**Can contain letters, numbers, or underscores:**

**Following the initial character, variable names can consist of letters, digits (0-9), or underscores (\_).**

**In this case, 'eggs' only contains letters, which is allowed.**

**Cannot start with a number:**

**Variable names cannot begin with a number.**

**In this case, '100' starts with the number '1', violating this rule.**

**Hence, while 'eggs' satisfies the rules and conventions for variable names, '100' is invalid as it starts with a number. It's worth noting that variable names should also be chosen to be descriptive, meaningful, and adhere to any specific naming conventions or style guidelines recommended by the programming language or community.**

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

**Answer:**

**In Python, you can use the following three functions to convert a value to its integer, floating-point number, or string representation:**

**int(): This function can be used to convert a value to an integer. It takes a numeric string or a floating-point number and returns the corresponding integer value by truncating the decimal part. For example:**

**python**

**Copy code**

**int("42") # Output: 42**

**int(3.14) # Output: 3**

**float(): This function is used to convert a value to a floating-point number. It takes a numeric string or an integer and returns the corresponding floating-point representation. For example:**

**python**

**Copy code**

**float("3.14") # Output: 3.14**

**float(42) # Output: 42.0**

**str(): This function converts a value to its string representation. It can be used to convert numbers, booleans, or other data types to a string. For example:**

**python**

**Copy code**

**str(42) # Output: '42'**

**str(3.14) # Output: '3.14'**

**str(True) # Output: 'True'**

**These functions allow you to convert values between different data types, enabling you to perform operations specific to those data types or represent them in a desired format.**

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 concatenating a string ('I have eaten ' and ' burritos.') with an integer (99) directly. In Python, concatenation with the + operator is only allowed between two strings or two values of the same type.**

**To fix the error and perform the desired concatenation, you can convert the integer 99 to a string using the str() function. Here's the corrected version of the expression:**

**python**

**Copy code**

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

**By wrapping 99 with str(), it is converted to the string '99'. Now all the elements in the expression are strings, allowing them to be concatenated successfully. The resulting string would be 'I have eaten 99 burritos.'.**