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

1. **'hello'**: This is a string value.
2. **-87.8**: This is a floating-point value.
3. **\***: This is an operator symbol. It is not a value or an expression on its own, but it can be used in expressions.
4. **-**: This is an operator symbol. It is not a value or an expression on its own, but it can be used in expressions.
5. **/**: This is an operator symbol. It is not a value or an expression on its own, but it can be used in expressions.
6. **+**: This is an operator symbol. It is not a value or an expression on its own, but it can be used in expressions.
7. **6**: This is an integer value.

I hope this helps clarify which elements in the list are values or expressions. Let me know if you have any more questions.

Top of Form

Regenerate

Bottom of Form

2. What is the difference between string and variable?

A string is a sequence of characters, such as a word or a phrase, that is enclosed in quotation marks. For example, "Hello, World!" is a string. Strings are used to store and manipulate text data in programs.

A variable, on the other hand, is a named location in a program's memory where a value can be stored and accessed. A variable can hold any type of data, including numbers, strings, and objects.

3. Describe three different data types.

1. **Integers**: Integers are whole numbers that can be positive, negative, or zero. They are used to represent whole quantities or counts. In Python, integers are represented by the **int** data type. For example:42, -1,0

**Floating-point numbers**: Floating-point numbers, also known as floats, are numbers with decimal points. They are used to represent continuous or fractional quantities. floats are represented by the **float** data type. Example ,25.26.

**Strings**: Strings are sequences of characters, such as words or phrases, that are used to store and manipulate text data. In Python, strings are represented by the **str** data type and are enclosed in quotation marks, EXAMPLE “HELLO WORD “

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

An expression is a combination of values, variables, and operators that produces a new value. Expressions are a fundamental building block of programming languages, as they allow you to manipulate and process data in your programs.

Expressions are made up of values, variables, and operators. Values can be literal values, such as numbers or strings, or they can be stored in variables. Operators are special symbols that perform operations on values and variables, such as arithmetic operations (e.g. **+**, **-**, **\***, **/**) or comparison operations (e.g. **==**, **!=**, **<**, **>**).

All expressions produce a result, which can be used in other expressions or assigned to a variable. For example, the expression **2 + 3** produces the result **5**, which can be stored in a variable or used in another expression:

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

A statement is a complete unit of execution in a program. It can be an assignment statement, a control statement (such as an **if** statement), or some other type of statement that performs a specific action or task.

This is an assignment statement. The left side of the **=** operator (**spam**) is the name of a variable, and the right side of the **=** operator (**10**) is an expression that evaluates to the value **10**. The assignment statement assigns the value of the expression to the variable, so that the variable **spam** now holds the value **10**.

So, to summarize the difference between expressions and statements:

* An expression is a combination of values, variables, and operators that produces a single value when evaluated.
* A statement is a complete unit of execution in a program that performs a specific action or task.

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

bacon = 22

bacon + 1

bacon = 22 bacon + 1

The variable **bacon** will contain the value **22**. The second line of code **bacon + 1** is an expression that calculates the value of **bacon** plus **1**, but this result is not assigned to any variable. Therefore, the value of **bacon** is not changed.

To update the value of **bacon** with the result of the expression, you would need to use an assignment statement:

Copy code

bacon = bacon + 1

This would reassign the value of **bacon** to the result of the expression **bacon + 1**, which is **23**.

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

'spam' + 'spamspam'

'spam' \* 3

, the **+** operator is used to concatenate (join) two strings. The **\*** operator is used to repeat a string a given number of times.

Therefore, the values of the following two expressions would be:

* **'spam' + 'spamspam'**: **'spamspamspam'**
* **'spam' \* 3**: **'spamspamspam'**

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

Therefore, **eggs** is a valid variable name because it starts with a letter and is followed by letters and/or digits (e.g., **eggs**, **eggs123**, **eggs\_and\_bacon**).

On the other hand, **100** is not a valid variable name because it starts with a digit. In Python, variable names cannot start with a digit.

Here are some examples of valid and invalid variable names in Python:

Valid:

* **eggs**
* **eggs123**
* **eggs\_and\_bacon**
* **\_private**

Invalid:

* **100** (starts with a digit)
* **spam!** (Contains an invalid character)
* **True** (Python reserves this name for the boolean value **True**)

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

there are several built-in functions that can be used to convert a value to a different data type. Here are three examples:

1. **int ()**: This function converts a value to an integer data type. If the value is a floating-point number, it will be truncated (not rounded) to an integer. If the value is a string, it must contain a numeric value that can be converted to an integer. For example: int (3.14) =3
2. **float ()**: This function converts a value to a floating-point number data type. If the value is an integer, it will be converted to a float. If the value is a string, it must contain a numeric value that can be converted to a float, float(2)=2.0
3. **str ()**: This function converts a value to a string data type. For example:

str(45)=”45”

Top of Form

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

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

causes an error because the **+** operator cannot be used to concatenate a string and an integer. In Python, the **+** operator is used to add two numbers or concatenate two strings, but it cannot be used to mix operands of different types.

To fix this error, you can either convert the integer **99** to a string using the **str ()** function, or use a different operator to perform the operation.

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