

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.

\* : expression

'hello' : value

-87.8 :value

- : expression

/ : expression

+ : expression

6 : value

2. What is the difference between string and variable?

String is a data type used to represent text whereas a variable is a named storage location in the memory or a container to hold value which could be text or number.

3. Describe three different data types.

i) Integer (int): They are used to represent whole numbers. It could be positive or negative, but shouldn't have any decimal points. Ex: 0, -5, 67

ii) Floating Point (float): They are used to represent numbers with decimal points. It also could be positive or negative. Ex: 4.5, -3.78

iii) String(str): They are used to represent text character including symbols. Ex: "Hello", "789", "Good Morning!!"

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

An expression is usually made up of mathematical operands and operators. Operand is the value on which the operation is being performed by the operator. Operators could be arithmetical like addition(+), subtraction(-) etc. or comparative like AND(&), NOT(~) etc. or assigning like equal to(=), +=. Expressions can also contain function calls, constants, and variables.

All expressions in programming languages evaluate to a single value. The value produced by an expression depends on the operators, operands, and any functions or variables involved.

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

An expression is a combination of values, variables, operators, and function calls that evaluates to a single value. Expressions are often used to represent computations or operations that produce a result. Expressions always evaluate to a single value. In many programming languages, expressions can appear within statements

A statement is a complete instruction that performs some action. It can include expressions but is not limited to them. Statements are typically complete lines of code that instruct the computer to do something, such as declaring a variable, assigning a value, controlling the flow of execution, or defining a function. Unlike expressions, statements do not necessarily evaluate to a value. Statements are the building blocks of a program's logic and control flow.

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

```
bacon = 22
```

```
bacon + 1
```

`bacon` will still contain 22 since there is no assignment back to the `bacon` variable after `bacon + 1`.

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

`'spam' + 'spamspam'`: The result would be `'spamspamspam'` since string concatenation is performed with `+`.

`'spam' * 3`: The result would be same as earlier, `'spamspamspam'`, but gotten through string repetition performed with `*`

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

Because in python variable names can't start with a digit.

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

If value is stored in a variable, a then

`int(a)` will give the integer version of the value stored in a

`float(a)` will give the float version of the value stored in a

`str(a)` version of the value stored in a

Or we can directly apply all the above function to a value like `"789"`

```
int("789") ==> 789
```

`float("789") ==> 789.0`

`str("789") ==> "789"`

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

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

Because the expression attempts to concatenate strings with an integer 99 without converting it to a string first. The corrected expression is:

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