

## Assignment : 01

**Question (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.

\*

&#39;hello&#39;

-87.8

-

/

+

6

**Solution :**

**Values or expression :**

&#39;hello&#39;

-87.8

6

**Mathematical Operators :**

\*

-

/

+

**Question (2).** What is the difference between string and variable?

**Solution :**

**String :** A string is a data type used to represent sequences of characters. It is used to store textual data, such as words, sentences, or any other form of text. In programming languages, strings are typically enclosed in quotation marks (single or double) to differentiate them from other data types.

Example: "Hello, World!"

**Variable :** A variable is a named storage location in a computer's memory where data can be stored and manipulated. It serves as a placeholder for values of different types, including strings. Variables

allow you to store and manipulate data in a way that makes your code more flexible and reusable.  
Example: name = "John"

**Question (3).** Describe three different data types.

**Solution :**

**Integer (int):** An integer data type represents whole numbers without any fractional or decimal parts. Integers can be positive, negative, or zero. They are used for counting, indexing, and performing arithmetic operations that involve whole numbers.

Example: age = 25

**Floating-Point (float):** The floating-point data type represents numbers with both integer and fractional parts. Floating-point numbers are used to represent real numbers that can have decimal points. They are suitable for representing quantities that can have a range of values, including those with decimal precision.

Example: pi = 3.14159

**String (str):** A string data type represents sequences of characters, typically used to represent textual information. Strings are enclosed in single or double quotation marks and can contain letters, numbers, symbols, and spaces. They are widely used for working with textual data, such as names, messages, and more.

Example: message = "Hello, World!"

**Question (4).** What is an expression made up of? What do all expressions do?

**Solution :** All expressions have a common purpose that is to produce a value. This value could be of various types, such as numbers, strings, booleans, or even more complex data structures. Expressions can be used in assignments, conditionals, loops, function calls, and many other programming constructs. Here's a breakdown of the components that make up an expression:

**Values:** These are the basic components, such as numbers (5, 3.14), strings ("Hello"), and booleans (True, False).

**Variables:** These are names that refer to specific values stored in memory. Variables allow you to store and manipulate data throughout your program.

**Operators:** These are symbols that represent specific operations, such as addition (+), subtraction (-), multiplication (\*), division (/), and comparison (==, !=, <, >), among others.

**Function Calls:** These involve invoking functions (predefined or user-defined) to perform specific tasks. Functions take arguments as inputs and return a value

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

**Solution :** In this example, `spam = 10` is a statement that assigns the value 10 to the variable `spam`. The right-hand side of the assignment (`10`) is an expression that evaluates to the value 10.

#### **Expression:**

An expression is a combination of values, variables, operators, and function calls that produces a single value. It can be thought of as something that computes or evaluates to a result. Expressions are used to perform calculations, create new values, and determine conditions. Expressions always have a value associated with them.

Examples of expressions:

- 1). `5 + 3` (computes the value 8)
- 2). `"Hello" + "World"` (computes the value "HelloWorld")
- 3). `len("Python")` (computes the value 6)

**Statement:** A statement is a complete line of code that performs a specific action or operation. Statements can include assignments, control flow structures (like conditionals and loops), and function definitions. Unlike expressions, statements do not necessarily produce a value.

Examples of statements:

- 1). Assignment statement: `spam = 10` (assigns the value 10 to the variable `spam`)
- 2). Conditional statement: `if x > 0: print("Positive")` (prints "Positive" if the condition is met)

**Question (6).** After running the following code, what does the variable bacon contain?

```
bacon = 22
```

```
bacon + 1
```

**Solution :** Now bacon contains value 23.

**Question (7).** What should the values of the following two terms be?

```
'spam' + 'spam'
```

```
'spam' * 3
```

**Solution :** spamspamspam

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

**Solution :** Variable names in Python cannot start with a digit. They must begin with a letter (either uppercase or lowercase) or an underscore (\_). Since 'eggs' starts with a letter, it's a valid variable name and 100 is invalid name.

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

**Solution :** In Python, we can use int(), float(), and str() functions to convert values into integer, floating-point, or string representations, respectively.

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

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

**Solution :** The expression `'I have eaten ' + 99 + ' burritos.'` causes an error because here we're trying to concatenate a string ('I have eaten ') with an integer (99). In Python, only strings with strings concatenation is allowed. Python does not allow any other data types directly.

To fix this problem, we can convert the integer 99 to a string using the `str()` function before concatenating.

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