PYTHON ASSIGNMENT- 1

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:-

In the provided list:

'*' - This is an expression representing the mathematical operator for multiplication.
'hello' - This is a value, specifically a string.
-87.8 - This is a value, specifically a floating-point number.
'-' - This is an expression representing the mathematical operator for subtraction.
'/' - This is an expression representing the mathematical operator for division.
'+' - This is an expression representing the mathematical operator for addition.
6 - This is a value, specifically an integer.

2. What is the difference between string and variable?

Answer:-

A string and a variable are both fundamental concepts in programming, but they serve different purposes:

String:

-A string is a data type used to represent a sequence of characters. It can contain letters, numbers, symbols, and whitespace.
-Strings are typically used to store textual data, such as names, sentences, or any other type of textual information.
-In most programming languages, strings are usually enclosed within quotation marks (either single or double quotes).

Example - my_string = "Hello, world!"

Variable:

-A variable is a placeholder or container for storing data in a program. It has a name (identifier) and a value associated with it.

-Variables are used to store and manipulate data during the execution of a program. The data stored in a variable can be of various types, including strings, numbers, booleans, etc.

-Unlike strings, variables can hold different types of data, and their values can change during the execution of a program.

```
x = 5
name = "John"
```

3. Describe three different data types.

Answer:-

1-Integer (int):

-Integers are a fundamental data type used to represent whole numbers without any fractional or decimal part.

-They can be positive, negative, or zero.

-Integers are often used for counting or representing numerical values in many programming scenarios.

Examples of integers: -10, 0, 42, 100, etc.

2-String:

-Strings are used to represent sequences of characters, such as letters, digits, symbols, and whitespace.

-They are typically enclosed within single quotes (' ') or double quotes (" ").

-Strings are versatile and are commonly used for handling textual data, like names, sentences, file paths, etc.

Examples of strings: "Hello, world!", 'Python', "12345", etc.

3-Boolean (bool):

-Booleans represent logical values indicating either true or false.

-They are often used in programming to control the flow of execution through conditional statements (e.g., if-else statements) or to evaluate expressions.

-Booleans are essential for decision-making in algorithms and for creating conditions under which certain parts of a program should execute.

Examples of boolean values: True, False.

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

Answer:-

An expression in programming is made up of one or more operands and operators, and sometimes function calls, which are evaluated to produce a single value. Let's break down its components:

1-Operands:

These are the values or variables that the expression operates on. For example, in the expression 5 + 3, the operands are 5 and 3.

2-Operators:

Operators define how the operands are manipulated or combined to produce a result. Examples of operators include arithmetic operators (+, -, *, /), comparison operators (==, !=, <, >), logical operators (and, or, not), assignment operators (=, +=, -=), etc.

3-Functions (sometimes):

In some cases, expressions may involve function calls, where a function is invoked with certain arguments, and its return value becomes part of the expression. For instance, sqrt(16) where sqrt() is a function that calculates the square root of a number.
Expressions are evaluated by the programming language's interpreter or compiler according to a set of rules known as the precedence and associativity of operators. The evaluation process involves substituting the values of the operands into the expression and applying the operators to produce a single value.

Expressions are evaluated by the programming language's interpreter or compiler according to a set of rules known as the precedence and associativity of operators. The evaluation process involves substituting the values of the operands into the expression and applying the operators to produce a single value.

All expressions, when evaluated, produce a result, which can be of various types depending on the operators and operands involved. This result can be assigned to variables, used in further expressions, or used as a condition for control flow statements like if-else or loops.

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

Answer:-

The difference between an expression and a statement lies in their fundamental purpose and behavior within a programming language:

1-Expression:

An expression is a combination of values, variables, operators, and function calls that evaluates to a single value.
Expressions are designed to produce a value, and they can be as simple as a single variable or as complex as a multi-operation mathematical computation.
Examples of expressions: 5 + 3, x * y, sqrt(16), 3 == 4.

2-Statement:

A statement is a complete instruction that performs some action. It may or may not produce a value.
Statements are the building blocks of a program's logic and control flow. They control the execution of code by specifying tasks to be performed or actions to be taken.
Examples of statements: assignment statements (spam = 10), conditional statements (if-else), loop statements (for, while), function definitions, import statements, etc.

-An expression evaluates to a value and can be part of a statement.
-A statement performs an action and does not necessarily result in a value, although it may contain expressions.

In the example spam = 10:

10 is an expression, as it evaluates to the value 10.
spam = 10 is a statement, as it performs the action of assigning the value 10 to the variable spam.

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

Answer:- The variable bacon still contains the value 22. This is because while the expression bacon + 1 evaluates to 23, it is not assigned to any variable or used in any way. It's just an expression that calculates the value bacon + 1, but the result is not stored or printed.

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

Answer:-

1-'spam' + 'spamspam':

This expression concatenates the string 'spam' with the string 'spamspam'.
Result: 'spamspamspam'
2-'spam' * 3:

This expression multiplies the string 'spam' by 3. It effectively repeats the string three times.
Result: 'spamspamspam'

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

Answer:-

'eggs': This is a valid variable name because it starts with a letter ('e') and contains only letters. Additionally, it's meaningful and descriptive, making it a suitable choice for a variable name.

100: This is invalid as a variable name because it starts with a digit ('1'). According to the rules, variable names cannot begin with a digit in most programming languages.

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

Answer:-

1-Integer Conversion:

-To convert a value to an integer, you can use the int() function.
-Example:

```
        value = 10.5
        integer_value = int(value)
```

2-Floating-Point Conversion:

-To convert a value to a floating-point number, you can use the float() function.
-Example:

```
value = "3.14"
float_value = float(value)
```

3-String Conversion:

-To convert a value to a string, you can use the str() function.
-Example:

```
value = 42
string_value = str(value)
```

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 attempts to concatenate a string with an integer without explicitly converting the integer to a string.

To fix this error, you need to ensure that all parts of the expression are treated as strings. we can achieve this by converting the integer 99 to a string using the str() function before concatenating it with the other strings.

Here how we can fix this error:

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

By using str(99), you convert the integer 99 to a string, allowing it to be concatenated with the other strings without causing an error.

- (16) **where** sqrt() **is a function that calculates the square root of a number.**