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.

Answer:

***** : Expression
'hello' : Values
-87.8 : Values
- : Expression
/ : Expression
+ : Expression
6 : Values

2. What is the difference between string and variable?

Answer:

Aspect	**String**	**Variable**
Definition	A data type representing a sequence of characters.	A named reference to a value or data stored in memory.
Purpose	Used to store and manipulate text.	Used to store data of any type (including strings) and provide a way to access it.
Syntax	Defined by enclosing characters in quotes ('Hello', "World").	Defined by assigning a value to a name (name = 'Alice').
Mutability	Immutable (cannot be changed after creation).	Variables can be reassigned to new values.
Example	"Hello, World!"	Greeting = "Hello, World!"
Usage	Directly used in operations like concatenation, slicing, etc.	Acts as a reference to access and manipulate the stored value.

This table summarizes the core differences between a string and a variable in Python.

3. Describe three different data types.

Answer:

1. Integer (int)

- **Definition**: Represents whole numbers without a fractional component.
- **Example**: 5, -42, 1000
- **Usage**: Used for counting, arithmetic operations, and indexing.

2. List

- **Definition**: An ordered collection of items that can be of different data types. Lists are mutable.
- **Example**: [1, 2, 3], ['apple', 'banana', 'cherry']
- **Usage**: Used to store sequences of items, which can be accessed and modified.

3. Dictionary (dict)

- **Definition**: An unordered collection of key-value pairs, where each key is unique. Dictionaries are mutable.
- **Example**: {'name': 'Alice', 'age': 25}
- **Usage**: Used to store data values in key-value pairs for fast lookups and modifications.

---

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

Answer:

Components of an Expression

An expression in Python is made up of:

1. **Operands**: These are the values or variables involved in the operation.
    - Example: In the expression 3 + 5, 3 and 5 are operands.
2. **Operators**: These are symbols that perform operations on the operands.
    - Example: In the expression 3 + 5, + is an operator.

Types of Expressions

Expressions can be of various types, such as:

- **Arithmetic expressions**: Perform arithmetic operations.
    - Example: 3 + 5, a - b * c
- **Logical expressions**: Evaluate to True or False.
    - Example: a and b, x > y
- **String expressions**: Combine or manipulate strings.
    - Example: "Hello, " + "world!"

Purpose of Expressions

All expressions in Python:

1. **Evaluate to a value**: The primary purpose of an expression is to be evaluated and produce a result.
    - Example: 3 + 5 evaluates to 8.
2. **Produce side effects (in some cases)**: Some expressions, such as function calls, may have side effects (e.g., modifying a variable, printing output).
    - Example: print("Hello") prints "Hello" and evaluates to None.

In brief, expressions are combinations of values and operators that Python evaluates to produce another value. They are fundamental building blocks in Python programming, allowing for calculations, logical decision-making, and data manipulation.

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

Answer:

Spam = 0 : It is an Expression

Aspect	**Expression**	**Statement**
Definition	Combination of values, variables, and operators that produces a value.	An instruction that the Python interpreter can execute.
Purpose	To be evaluated and return a value.	To perform an action or control the flow of the program.
Example	3 + 5, a * b, len("hello")	if, for, while, def, assignment (x = 10)
Result	Always results in a value.	May or may not produce a value; mainly used for its effect.
Usage	Used within statements, assignments, function calls.	Controls program structure and execution flow.
Evaluation	Evaluates to a value.	Executes an action.
Code Example	result = 3 + 5	if x > 5: print("x is greater than 5")

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

Answer: 22, since value of Bacon is not updated by the code.

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

Answer:

‘spam’ + ‘spamspam’ : Output will be ‘spamspamspam’
‘spam’*3 : Output will be spam

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

Answer:

According to the rules for declaring a variable in python:
A variable name must start with a letter or the underscore character.
A variable name cannot start with a number.

Hence, Egg is a variable where 100 is invalid.

---

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

Answer:

int()
float()
string()

---

10. Why does this expression cause an error? How can you fix it?
‘I have eaten’ + 99 + ‘burritos’

Answer:

Since, int and string concatenation is not possible so we will get the error.
In order to fix it we can modify expression as mentioned below:
'I have eaten' + '99' + 'burritos'
