Python Assignment -1

1. Ans.

From the given list Values are – 'hello', -87.8 ,6

And expressions are - *, -, /, +

2. Ans.

1. Nature:

String: A string is a specific type of data that represents text or a sequence of characters.
Variable: A variable is a symbolic name or identifier that can represent various types of data, including strings, numbers, and more.

2. Content:

String: It contains actual textual data, such as "Hello, World!".
Variable: It doesn't contain data itself; it serves as a reference or a label for a data value. For example, a variable named `name` can hold the string "John".

3. Usage:

String: Used for representing and manipulating text and character data.
Variable: Used to store and manage data of different types, including strings, numbers, and more. They play a central role in program logic and data manipulation.

3. Ans.

Integer (int):

An integer is a fundamental data type that represents whole numbers, both positive and negative, without any fractional or decimal parts. Integers are typically used for counting and representing quantities that can be expressed as whole numbers, such as 1, -5, 100, or 0.

They are generally efficient in terms of memory usage and are used for tasks like indexing arrays, counting iterations in loops, and performing arithmetic operations.

String:

A string is a data type used to represent text or sequences of characters. It can contain letters, numbers, symbols, spaces, and special characters. Strings are essential for tasks involving textual data, such as storing names, messages, file contents, and more. They are often manipulated with various operations like concatenation, splitting, and searching for specific substrings.

Float (Floating-Point):

Float is a data type used to represent numbers with fractional or decimal parts. It's used for real numbers, including numbers like 3.14, -0.001, or 2.71828. Floating-point numbers are suitable for tasks where precision with decimal places is required, such as scientific calculations, financial calculations, and representing physical measurements.

4. Ans.

An expression in programming is made up of the following elements:

Values: These are the basic building blocks of an expression. Values can be literals or the result of other expressions. For example, 5 or "Hello, " are values.

Variables: Variables are symbols that represent data values. We can use variables in expressions to refer to the data they hold. For instance, if we have a variable x, we can use it in an expression like x + 3.

Operators: Operators are symbols or keywords that represent actions or operations to be performed on values or variables. Common operators include +, - , * , / .

Functions: Functions are reusable blocks of code that perform specific tasks. They take input values and return a value. we can use functions in expressions to perform complex operations.

What all expressions do is evaluate to a value. When we use an expression in a program, the programming language processes it, calculates the result according to the specified operations, and returns a value. This value can be used in various ways within the program, such as for assignments to variables, as part of control flow conditions, or for output to the user.

5. Ans.

The key difference between an expression and a statement in programming lies in their fundamental purpose and behaviour:

1. Expression:
An expression is a combination of values, variables, operators, and functions that can be evaluated to produce a single value.
Expressions typically return a result, and this result can be used within larger expressions or assigned to a variable.
Example: `5 + 3` is an expression that evaluates to `8`, which can be used as part of a larger expression or assigned to a variable like `result = 5 + 3`.

2. Statement:
   A statement is a line of code that performs an action. Statements do not produce a value like expressions do; instead, they execute some operation or control the flow of a program.
   Assignment statements, like `spam = 10`, are a type of statement that assigns a value to a variable.
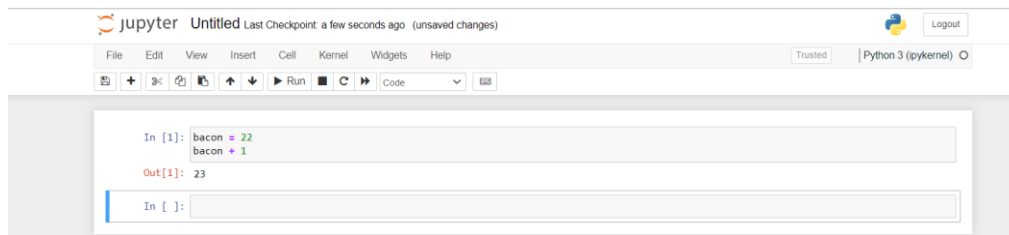
expressions are used to calculate and produce values, while statements are used to control program flow and perform actions. Assignment statements, like the one mentioned (`spam = 10`), are statements that assign a value to a variable but do not themselves produce a value.

6. Ans.

    bacon = 22
    bacon + 1

    After running this code bacon will contain 23.

7. Ans.

    'spam' + 'spamspam':

    This expression uses the + operator to concatenate two strings.
    The result of this expression is the string 'spamspamspam'. It combines
    the two strings, resulting in a longer string.

    'spam' * 3:

    This expression uses the * operator to repeat a string multiple time.
    The result of this expression is the string 'spamspamspam'. It repeats the
    string 'spam' three times to create a longer string.
    So, in both cases, the resulting value is 'spamspamspam'.

8. Ans.
   Variable names typically must start with a letter (a-z or A-Z) or an underscore (_). Some languages allow other characters, such as numbers, but they cannot be the first character.
   "eggs" starts with a letter 'e', making it a valid variable name.
   "100" starts with a number '1', which is why it's usually considered invalid as a variable name.

9. Ans.

   Integer Conversion:

   To convert a value to an integer version, we can typically use a function like int()named after the data type we want to convert to.

   Floating-Point Conversion:

   To convert a value to a floating-point number, we can use a function like float()in our programming language.

   String Conversion:

   To convert a value to a string, we can use a function like str()that converts the value to a string representation.

10. Ans.

   The expression 'I have eaten ' + 99 + ' burritos.' causes an error because we are attempting to concatenate a string with an integer directly, which is not allowed. When the code encounters the `+` operator between a string and an integer, it may not know how to perform the operation because it's ambiguous.

   To fix this, we should convert the integer `99` to a string before concatenating it with the other strings. we can do this using the `str()` .

```
In [13]: 'I have eaten ' + 99 + ' burritos.'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3640\3992538259.py in <module>
----> 1 'I have eaten ' + 99 + ' burritos.'

TypeError: can only concatenate str (not "int") to str
```

```
In [14]: 'I have eaten ' + str(99) + ' burritos.'
```

```
Out[14]: 'I have eaten 99 burritos.'
```

```
In [ ]:
```