Assignnt-1

1. The given elements are :

\*

'hello'

-87.8

-

/

+

6

In this above values we have,

->Values are:

‘hello’-It is a string

-87.8 -It is negative floating point number

6 -It is non negative Integer value

->The expressions are:

‘-’ - It is a subtraction operator

/ - It is a division operator

+ - It is a addition operator

\*. – It is a multiplication operator

2) Differences between variable and strings

|  |  |
| --- | --- |
| Variable | Strings |
| i)Variable is a name which is used to store a specific value in it. | i)String is a data type which is a sequence of characters |
| ii) It can be Integer, Floating point numbers ,Strings,….etc | ii) String is a sentence. Here we are storing the number of characters. Including special characters. |
| iii) Variable is used to manage the data in a program. | iii) String is used to manipulate the text based data. |
| iv) In variables the values are differ in execution of a program. | iv) Here we are storing the strings in a specific format. And the data is enclosed in braces in double or single quotes. |
| iv)Examples for Variable is  var=”Welcome to python course” | iv)Example for strings is  “Welcome to python course” |

1. Expression:

An expression is a combination of operators and operands.

Where,

Operators: These are the symbols or keywords which are used to perform the operation on the same type. Operators including + , -, \* , / …etc are called operators.

Ex: A+B\*C/D, P/Q\*R+Z…..etc

Operands : These are the keywords, constants, variables, ….etc are called as operands.

Ex: 1+2, a\*b/c, ….etc

* These combination of operators and operands form a expression. These expressions are fundamentals to programming. They can perform calculations, make decisions, and control the flow of a program. Additionally, expressions are used in a wide range of operations.
* By using the BODMAS rule computer make the result of the expression
* B- Bracket O-Orders D-Division M-Multiplication A-Addition S-Substraction.

Ex:

>>> A=12

>>>B=18

>>>C=7

>>>Result=A\*B+C/A

>>>print(Result)

216.58333333333334

5) An expression is a piece of code that produces a value. It can be as simple as a single constant or variable, or it can be more complex, involving operations, functions, and other elements. For example, `5 + 3` is an expression that evaluates to `8`.

A statement, on the other hand, is a complete line of code that performs some action. It doesn't necessarily produce a value. For example, an assignment statement like `spam = 10` assigns the value `10` to the variable `spam`. While this line of code does something (it assigns a value), it doesn't have a value itself.

Expression:

- Produces a value.

- Can be a combination of constants, variables, operators, and function calls.

- Can be part of a statement.

- Example: `5 + 3` evaluates to `8`.

Statement:

- Performs an action.

- Doesn't produce a value (or its value is usually not used in further computations).

- Example: `spam = 10` assigns the value `10` to the variable `spam`.

6) Given,

>>>bacon=22

>>>bacon+1

23

* Here we are adding a one to the bacon variable. Means we are calculating the value of bacon+1 value only. But we are not assigning to the any of the other values. So, the value of the bacon is not changed.
* To change the value of bacon the we use the above example type of expression

>>>bacon=bacon+a

>>>print(bacon)

23

* In this above example we are assigning the value to the bacon. So, the value can be changed.

7)

The values of the two terms are:

* `'spam' + 'spamspam'` will result in the string `'spamspamspam'`. This is because the `+` operator concatenates the two strings together.
* `'spam' \* 3` will result in the string `'spamspamspam'`. This is because the `\*` operator repeats the string `'spam'` three times, concatenating them together.

8)

Here's why `eggs` is a valid variable name while `100` is not:

1. Variable names must start with a letter (a-z, A-Z) or an underscore (\_). They cannot start with a number.

- `eggs` starts with a letter ('e'), so it's valid.

- `100` starts with a number ('1'), which is not allowed.

2. Variable names can only contain letters (a-z, A-Z), numbers (0-9), and underscores (\_). They cannot contain special characters or spaces.

- `eggs` contains only valid characters ('e', 'g', 's'), so it's valid.

- `100` contains only numbers, but as it starts with a number, it's invalid.

3. Variable names cannot be a keyword or reserved word in the programming language.

- `eggs` is not a reserved word, so it's valid.

- `100` is not a reserved word either, but it starts with a number, which is not allowed.

That’s why Using egg as a variable name is correct and using 100 as variable name is incorrect.

9)

In Python supports these three functions to convert values from one data type to another data type:

1. Integer Conversion:

- Function: `int()`

- Syntax: int(value)

- Ex:

>>>int(10.23)

10

2. Floating-Point Conversion:

- Function: `float()`

- Syntax: float(value)

- Ex:

>>>float(10)

10.0

3. String Conversion:

- Function: `str()`

- syntax: str(value)

- Ex:

>>>str(“10”)

10

These functions allow you to convert values between different data types, which can be useful in various situations, such as when you need to perform operations that require compatible data types.

10)

The expression ` 'I have eaten ' + 99 + ' burritos.'` causes an error because you're trying to concatenate a string with an integer directly. In Python, you can concatenate strings with other strings, but not with integers.

To fix this, you need to convert the integer `99` into a string before concatenating it. You can do this using the `str()` function. Here's the corrected expression:

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

'I have eaten 99 burritos.'