

Python Basic- Ass- 01

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 (Multiplication)

'hello' -> Values (String)

-87.8 -> Values (Integer)

- -> Expression (Subtraction)

/ -> Expression (Division)

+ -> Expression (Addition)

6 - Values (Integer)

2. What is the difference between string and variable?

Answer:

String is a data type which is text type data on the other hand variable is a store of information which can be any data type such as string, integer, float etc.

Example- "Mahmud" is a string. if we want to store it in any variable such as A then we should write as below

A = "Mahmud"

Now "Mahmud" has been stored in A variable.

3. Describe three different data types.

Answer:

We have a few numbers of data types among them we are going to discuss only three data types. which are string, integer and float.

String: String is a data type which is basically text type data. Numbers can be strings which can not perform any mathematical operation until converted to integer or float data type.

In python we use string data type as “str”. To write strings we have to use single or double quotes.

Integer: An integer is any whole number without fractions that can be zero, positive or negative. It is possible to do mathematical operations with integer type data. To write an integer, we just write it directly without using any sign or symbol.

In python we use integer data type as “int”.

Example- 0, -10, 20

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

Answer:

An expression in python is a combination of operators and operands. It is a combination of values, operators, and other python elements that the language interprets and acts upon. It is a combination of constants, variables, operators, and functions that the python interpreter can evaluate. Expressions can be simple or complex and can contain any combination of data types, including strings, numbers, tuples, and lists.

Expressions in python are representations of a value, object, or set of operations. They are composed of values (literals), variables, operators, and function calls. It can be used to perform calculations, manipulate data, and evaluate conditions. It can also be used to generate values and assign them to variables. Expressions are evaluated in the context of a program, taking into account the values of variables, functions, and operators.

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

Answer:

An expression in python is a combination of operators and operands whereas a statement is a piece of code that represents a command or action. Here “spam = 10” is both an expression and a statement. Because “spam = 10” contains both operator (‘=’ assignment operator) and operands (‘spam’ is *variable* & ‘10’ is *constant*). On the other hand, ‘spam = 10’ is an assignment statement. It assigns the value ‘10’ to the variable ‘spam’.

But if we write ‘print(“name”)’ this is a statement but not an expression. Because in this code there is no operator or operand so we can not call it as an expression but it can perform an action by printing the string inside the print function. In short, we can tell every expression is a statement but every statement is not an expression.

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

***bacon* = 22**

***bacon* + 1**

If we run the code *bacon* contains 22 as 22 is assigned to *bacon*. But the result will be 23 because we are adding 1 with this *bacon*. But this 1 will not be assigned to *bacon*.

So variable *bacon* contains 22 values as integers.

7. What should the values of the following two terms be?

'spam' + 'spamspam'

'spam' * 3

Answer:

Result of 'spam' + 'spamspam' will be 'spamspamspam'. Both values will be added by the plus operator.

Result of 'spam' * 3 will be 'spamspamspam'. spam will be printed 3 times because of the multiplication operator.

So both expressions will give the same value that is 'spamspamspam'.

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

Answer:

In Python, a variable name can contain letters, digits, and underscores, but it must begin with a letter or an underscore. This means that 'eggs' is a valid variable name, because it begins with a letter and only contains letters.

On the other hand, 100 is not a valid variable name in python because it begins with a digit. In Python, variables must begin with a letter or an underscore, not a digit.

It's important to note that in python, variable names are case-sensitive, so eggs, Eggs, and EGGS are all considered to be different variables.

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

Answer:

The `int()`, `float()`, and `str()` functions can be used to convert a value to its integer, floating-point number, or string representation, respectively.

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

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

Answer:

This expression causes an error because the `+` operator cannot be used to concatenate a string and an integer. In Python, you can only use the `+` operator to add two numbers or to concatenate two strings.

To fix this error, you can either convert the integer to a string using the `str()` function, or use a different string formatting method to insert the integer into the string.

Here is an example of how you could fix the expression using the `str()` function:

'I have eaten ' + str(99) + ' burritos.' or **'I have eaten {} burritos.'.format(99)**

Both of these revised expressions will produce the string *'I have eaten 99 burritos.'*