Python Basics 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

Ans: There are a total of 4 Operators and 3 Expressions, They are:
Operators: *,-,/,+
Expressions: 'hello', 87.8, 6

2. What is the difference between string and variable?

Ans: A Variable is used to store of information, and a String is a type of information you would store in a Variable. A variable is created the moment you first assign a value to it. Example: x = 5, y = "John". Here x and y are variables.

A String is a group of characters or a single character usually enclosed in Double quotes " " or single quotes ' '. Even triple quotes can be used in Python but generally used to represent multiline strings and docstrings. Example: my_string = 'Hello'

3. Describe three different Data Types ?

Ans: Three fundamental Data types in python are int, float, complex.

1. **int data type**: We can use int data type to represent whole numbers (integral values) Example: int_num=100
2. **float data type**: We can use float data type to represent floating point values (decimal values) Example: flo_num=1.3e3
3. **complex data type**: Complex number is represented by complex class. It is specified as (real part) + (imaginary part)j. Example: com_num=10+3.5j

```
# Example for int data type
int_num=100
print(int_num, type(int_num))
# Example for float data type
flo_num=1.3e3
print(flo_num, type(flo_num))
# Example for Complex data type
com_num=10+3.5j
print(com_num, type(com_num))
```

```
100 <class 'int'>
1300.0 <class 'float'>
(10+3.5j) <class 'complex'>
```

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

Ans: An expression is a combination of values, variables, operators, and calls to functions. Expressions need to be evaluated. If we ask Python to print an expression, the interpreter evaluates the expression and displays the result. An expression is evaluated as per the precedence of its operators. So that if there is more than one operator in an expression, their precedence decides which operation will be performed first.

```
5*4+50-50 # Is an Expression, The Python Interpreter Evaluates it to 20
```

```
20
```

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

Ans: An expression is a combination of values, variables, and operators.When we type an expression at the prompt, the interpreter evaluates it, which means that it finds the value of the expression.An expression is evaluated as per the precedence of its operators. So that if there is more than one operator in an expression, their precedence decides which operation will be performed first.

eg: 5*4+50-50 is an example of a expression

A statement is a unit of code that has an effect, like creating a variable or displaying a value.When we type a statement, the interpreter executes it, which means that it does whatever the statement says. In general, statements don't have values. A statement is an instruction that a Python interpreter can execute. There are mainly four types of statements in Python, Print statements, Assignment statements, Conditional statements and Looping statements.

eg: courseName = 'Yay!! I am going to be a data scientist', spam = 10

```
#Example:
5*4+50-50 # Is a Expression
courseName = 'Yay!! I am going to be a data scientist' # Is a Statement
print("iNeuron") # Is a Expression Statement
```

```
iNeuron
```

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

```
bacon = 22
bacon + 1
```

Ans: The variable bacon is set to 22 .The expression bacon + 1 does not reassign the value in bacon (that would the case if the expression is like bacon = bacon + 1 instead of bacon + 1)

```
# Example Case#1
bacon=22
bacon + 1
print(bacon)
```

```
22
```

```
#Example Case#2
bacon=22
```

```
bacon=bacon+1
print(bacon)
```

23

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

```
'spam'+'spamspam'
'spam'*3
```

Ans: Both expressions evaluate to the string **'spamspamspam'** Where as the first expression follows String Concatentation and the second expression follows String Multiplication

```
print('spam'+'spamspam') # string concatenation
print('spam'*3) # string multiplication
```

```
spamspamspam
spamspamspam
```

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

Ans: As per python,Variable names cannot begin with a number. The python rules for naming a variable are :-

1. Variable name must start with a letter or the underscore character.
2. Variable name cannot start with a number.
3. Variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, & _).
4. Variable names are case-sensitive (name, GAURAV and gaurav are three different variables).
5. The reserved words(keywords) cannot be used naming the variable.

```
egg='Mihir' # Valid variable Initilization
100='hello' # Invalid Variable Initilization
print(egg) #prints the value of egg ie Ineuron
print(100) # Raises a Syntax Error as 100 is not a valid variable name
```

```
  Input In [7]
    100='hello' # Invalid Variable Initilization
    ^
SyntaxError: cannot assign to literal
```

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

Ans: The int(),float(),and str() functions will evaluate to the integer,floating-point number,string version of the value passed to them.

```
# Examples:
print('int(10.0) -> ',int(10.0)) # int() function converts given input to int
print('float(10) -> ',float(10)) # float() function converts given input to float
print('str(10) -> ',str(10)) # str() function converts given input to string
```

```
int(10.0) ->  10
float(10) ->  10.0
str(10) ->  10
```

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

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

Ans: This cause of error is 99.because 99 is not a string. 99 must be typecasted to a string to fix this error. The correct way of representing is mentioned below:
Input: 'I have eaten ' + str(99) + 'burritos.'
Output: 'I have eaten 99 burritos.'

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

I have eaten 99 burritos