Assignment_1(**Ineuron**)- by Pronab kurmi.

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.

* = expression

'hello' =value

-87.8 =value

- =expression

/ =expression

+ =expression

6 =value

2.What is the difference between string and variable?

Ans: - string- strings in python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes.

- ➔ ‘hello’ is same as “hello”
- ➔ For eg
  Print(“hello”)
  Output
  hello

variable-variables are nothing but reserved memory locations to store values.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.

- ➔ For eg
  A=100 #an integer assignment
  Print(A)

Output
100

3.Describe three different data types.

Ans: -Three data types are-

➔ Int or integer
➔ Bool or Boolean
➔ List

Int- integer is a whole number, positive or negative, python has no restriction on the length of an integer. Its value belongs to **int.**

➔ For eg
Print (2345676+2)
Output
2345678

Bool- Boolean type provides two built-in values, True and False. These values are used to determine the given statement true or false. It denotes by the class bool.

➔ type(true)
output
true
➔ type(false)
output
false

List- list is the collection which is orderd and changeable and allows duplicate members.

To use a list we use square brackets and separate values with commas.

➔ For eg

List1[=[1,2,3'a']

Print(List1)

Output

[1,2,3'a']

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

Ans:-An expression is the combination of values, variables, and operators.an expression evaluated using assignment operator.

➔ For eg
X=10
Z=x+20
Print(z)

Output

30

Python also defines expressions only contain identifiers, literals and operators.

Identifiers-any name that is used to define a class, function, variable module or object is an identifier.

Literals-in python there are the string literals, byte literals, integer literals, floating point literals and imaginary literals.

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

Ans:-

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

bacon = 22 bacon + 1

```
In [1]: bacon = 22
        bacon + 1
Out[1]: 23
```

solution is 23

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

'spam' + 'spamspam'

'spam' * 3

```
In [3]: 'spam' + 'spamspam'
Out[3]: 'spamspamspam'
```

string concatination

```
In [4]: 'spam' * 3
Out[4]: 'spamspamspam'
```

string replication

---

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

Ans: - eggs a valid variable name while 100 is invalid because we can't start giving variable an integer name. if we, we should begin with, a string-like alphabet name then integer. e100 or eggs100 is valid.

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

Ans: - Integer- int(),

floating-point number- float(),

string-str(),

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

'I have eaten' + 99 + 'burritos'

Ans: - because 99 is an integer it cannot be concatenated with strings, if we have to concatenate it we need to do typecasting. Or put the 99 with the cots.

```
In [16]: a='I have eaten' + "99" + 'burritos'

         print(type(a))
         print(a)

         <class 'str'>
         I have eaten99burritos
```

```
In [8]: 'I have eaten' + 99 + 'burritos'

        ---------------------------------------------------------------------------
        TypeError                                 Traceback (most recent call last)
        <ipython-input-8-54adfbe3d9ea> in <module>
        ----> 1 'I have eaten' + 99 + 'burritos'

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