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.

*	product or multiplication / **EXPRESSION**
'hello'	string / **VALUE**
-87.8	float / **VALUE**
-	minus or difference / **EXPRESSION**
/	division or divided by / **EXPRESSION**
+	Sum or Addition / **EXPRESSION**
6	integer / **VALUE**

2. What is the difference between string and variable?

A Variable is used to store any kind of data or datatype like int, str, float,.. etc

A String is a specific data type.

3. Describe three different data types.

Float (float) This is a Number based datatype. In this kind of datatype, number is considered along with its decimal component.

Integer (int) This is a Number based datatype. In this kind of datatype, only the integer part of the number is taken into consideration. That means, decimal component is not taken into consideration.

String (str) This is a Text based datatype. In this kind of datatype, the data is considered as text only. That mean, even the numbers are considered as text.

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

An expression is made of values, variables, and operators.

Expression has the ability to do mathematical operations like Addition, Subtraction, Division, Multiplication etc.. Essentially, it does calculation or **evaluation.**

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

A **statement** will **creates** a value or **displays** a value.

```
In [1]:
1  spam = 10    #this is a statement because it CREATES A VALUE
2
3  print(spam)  #this is a statement because it DISPLAYS A VALUE

10
```

An **expression** is made of values, variables, and operators.

In Python, **statement** is **executed**, and **expression** is **evaluated**. That is statement will do whatever it says (like assigning, printing etc.). Whereas an **expression** will calculate the value of equation (like addition or subtraction or division etc)

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

bacon = 22

bacon + 1

The variable bacon contain **23** after running the code.

```
In [1]: 1 bacon = 22
        2 bacon + 1

Out[1]: 23
```

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

'spam' + 'spamspam'

'spam' * 3

`Its same output for both. 'spamspamspam'`

```
In [1]: 1 'spam' + 'spamspam'

Out[1]: 'spamspamspam'

In [2]: 1 'spam' * 3

Out[2]: 'spamspamspam'
```

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

Because in python, the variable name should start with a letter or _ (underscore). Python cannot accept a variable name that starts with a number.

```
In [1]: 1 eggs = 5

In [2]: 1 100 = 5
          File "<ipython-input-2-2c13fee6e2f2>", line 1
            100 = 5
            ^
        SyntaxError: cannot assign to literal

In [3]: 1 _100 = 5

In [4]: 1 100_eggs = 5
          File "<ipython-input-4-ee3414740901>", line 1
            100_eggs = 5
               ^
        SyntaxError: invalid decimal literal
```

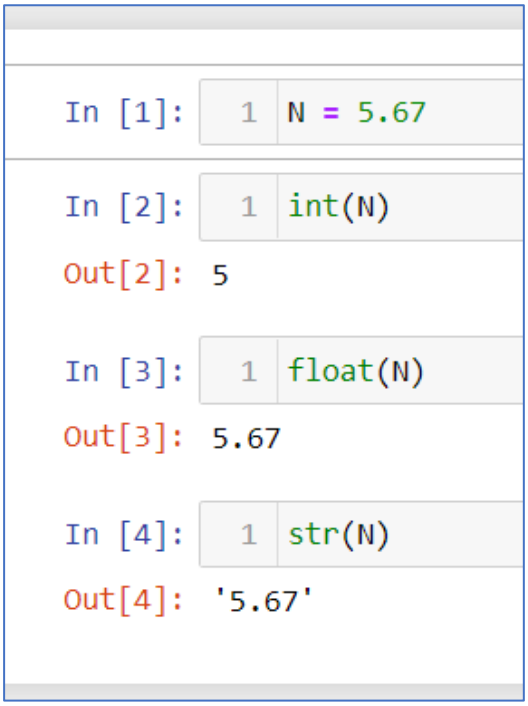
9. What three functions can be used to get the integer, floating-point number, or string version of a value?

N = 5.67 # Let us say

Integer → **int**(N) → 5

Floating-Point Number → **float**(N) → 5.67

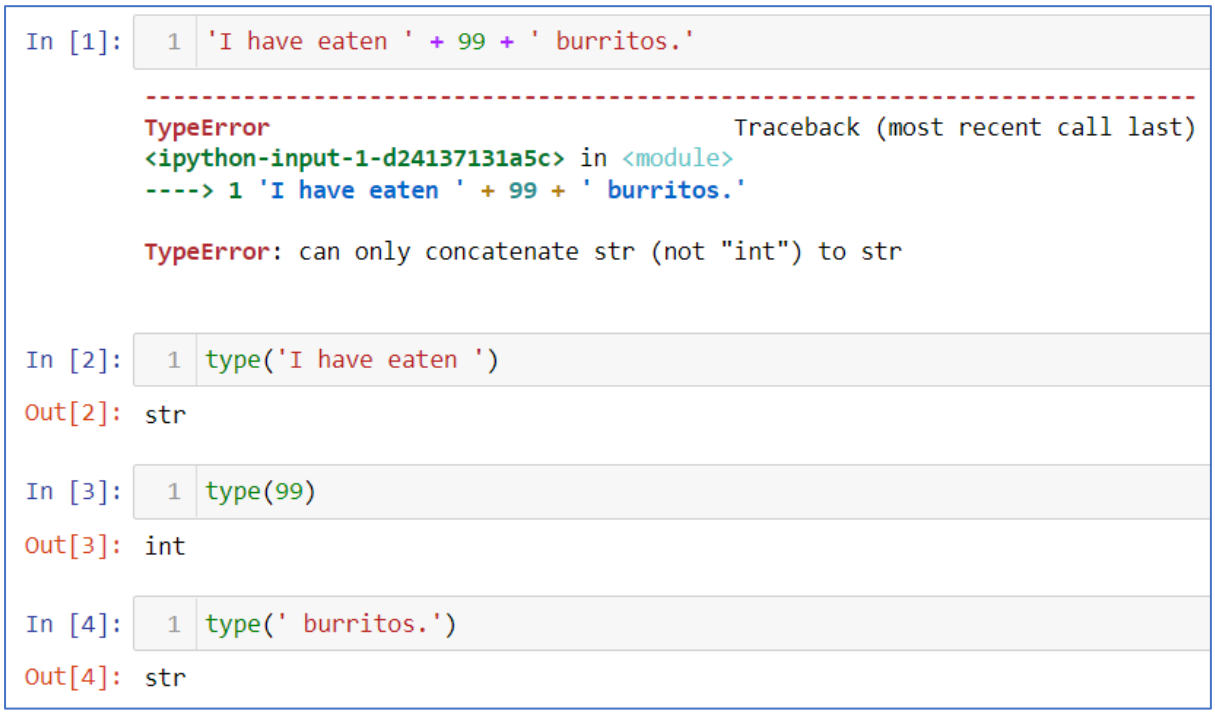
String → **str**(N) → 5.67



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

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

Here, 'I have eaten ' and ' burritos.' are string data types and 99 is an integer data type. Addition (concatenation) of string data type with integer data type is not possible.



To avoid the error, the same can be written as

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

Note that 99 is included within braces, converting 99 into a string.

```
In [5]: 1 'I have eaten ' + '99' + ' burritos.'
Out[5]: 'I have eaten 99 burritos.'

In [6]: 1 type('I have eaten ')
Out[6]: str

In [1]: 1 type('99')
Out[1]: str

In [8]: 1 type(' burritos.')
Out[8]: str
```