Assignment 1

1.

In [7]:
```
a = *
type(a)     # expressions will be mathematical operator
```

```
  File "C:\Users\DELL\AppData\Local\Temp/ipykernel_14872/908972201.py", line 1
    a = *
        ^
SyntaxError: invalid syntax
```

In [6]:
```
a = 'hello'
type(a)
```

Out[6]:
```
str
```

In [8]:
```
a = -87.8
type(a)
```

Out[8]:
```
float
```

In [9]:
```
a = -
# Expressions mathematical operator
```

```
  File "C:\Users\DELL\AppData\Local\Temp/ipykernel_14872/722493199.py", line 1
    a = -
        ^
SyntaxError: invalid syntax
```

In [ ]:
```
/
+
#mathematical operators
```

In [10]:
```
a = 6
type(a)
```

Out[10]:
```
int
```

2.

difference between string and variable

String:

A string is simply a series of characters. Anything inside quotes is considered a string in Python,and you can use single or double quotes around your strings like this:

In [4]:
```
"This is a string."
a string.'
```

Out[4]: 'This is also a string.'

variable:

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

3.

Three different data types are:

1. Numbers : When a number is assigned to a variable number class object is created.

- int : eg- a = 10

-float: eg a = 10.22

-complex eg a = 3 +4j

2. String: The string can be defined as the sequemce of characters represented in a single or double quotes.

e.g "Hello all"

3. Tuples: Tuples also store the collection of the elements of different data types. A tuple is the same as the list, but a tuple is immutable (non-editable or cannot modify the size and elements value). To create a tuple uses the () simple parenthesis; within this brackets, store all the elements separated with the comma (,).

4. List: List stores a collection of different types of elements. The list is mutable (editable). It is the same as arrays in C, but the list stores elements of different data types. To create a list uses the [] square brackets; within these brackets, stores all the elements separated with the comma (,).

4.

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

An expression is made up of combination of operators, constants and variables. An expression may consist of one or more operands, and zero or more operators to produce a value.

These types of expressions are used to show the relation between two entities. These entities may be an integer, floating-point number, or something else.

5.

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

6.

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

In [7]:
```
bacon = 22
bacon
```

Out[7]:
```
22
```

In [8]:
```
bacon + 1
```

Out[8]:
```
23
```

7.

What should the values of the following two terms be?

In [9]:
```
'spam' + 'spamspam'
```

Out[9]:
```
'spamspamspam'
```

In [10]:
```
'spam'* 3
```

Out[10]:
```
'spamspamspam'
```

Addition operator of string

Multiplication operator of string and integer.

8.

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

Because variable names cannot begin with a number.

9.

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

int() float() string()

10.

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

In [11]:
```
'I have eaten' + 99 + 'burritos'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_18332/2733022945.py in <module>
----> 1 'I have eaten' + 99 + 'burritos'
```

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

In [14]:
```
'I have eaten' + ' 99 ' + 'burritos'
```

Out[14]:
```
'I have eaten 99 burritos'
```

In [ ]:
```

```