Python Secret Sauce – Basics

Names refer to values: Think of it as a tag, an arrow that refers to the value (aka object)

Many names can refer to a value.

So if

x = 20

and

y = x,

y is NOT referring to x, Y is referring to 20.

In the case of lists

L = [‘a’, ‘b’, ‘c’]

M = L

M does NOT have a separate list in memory. It has a reference to the same list that L refers to.

(append method makes an in-place modification)

x = x + 1 : rebinds x to a new object

L.append(‘d’) : carries out an in-place modification

L + [‘e’] : this is NOT an in-place operation. It creates a copy of L and adds an element to it.

L += [‘f’, ‘g’]

Is the same as

L = L + [‘f’, ‘g’] or

L.append([‘f’, ‘g’])

Assignments also at play in loops

```
For x in L:
        do something(x)
```

Each element in L is assignment to x.
That is.   x = L[0], then do something. Then, x = L[1].

range(8), starts from 0 and terminates at 7!

def func(x) : is an assignment of x to a value