Full stack web development using python

List

Part-2

Saurabh Shukla (MySirG)

Agenda

1. Packing and Unpacking
2. built-in methods
3. list() method
4. comparison operator on list
5. concatenation operator
6. Repetition Operator
7. Slicing Operator

Packing and Unpacking

```
l1 = [20, 50, 30]

a, b, c = l1          # unpacking
```

number of variables in the left hand side must be same as the number of elements in the list.

```
a = 5
b = 6
c = 10

l2 = [a, b, c]        # packing
```

Built in methods

len() → returns length of specified iterable

min() → returns min value element

max() → returns max value element

sum() → returns sum of elements

sorted() → returns a sorted list of elements.

```
l = [24, 13, 5]
l1 = sorted(l)
```

[5, 13, 24]

list() method

```
l1 = list()          # l1 = []
l1 = list(10)        # l1 = [10]  Error
l1 = list(10,20,30)
l1 = list("MySirG")
l1 = list(range(5))
l1 = list([10,20])
```

list() method can take at most one argument.

one argument ⟶ iterable

Comparison Operator on list

```
l1 = [1, 2, 3]
l2 = [2, 3, 1]
l3 = [1, 2, 3, 4, 5]
l4 = [1, 2, 3]
```

`l1 == l2` False

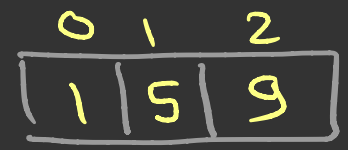
`l1 == l3` False

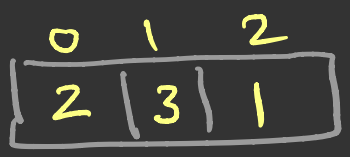
`l1 == l4` True

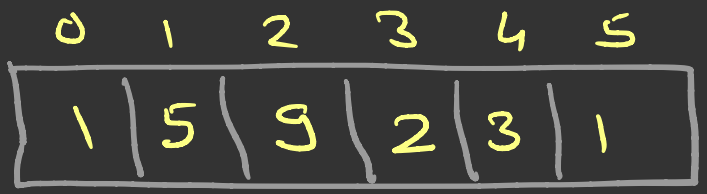
`l1 > l2` False

Concatenation Operator

```
l1 = [1, 5, 9]
l2 = [2, 3, 1]
l3 = l1 + l2
```



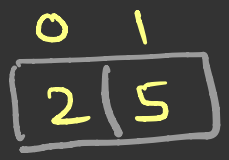




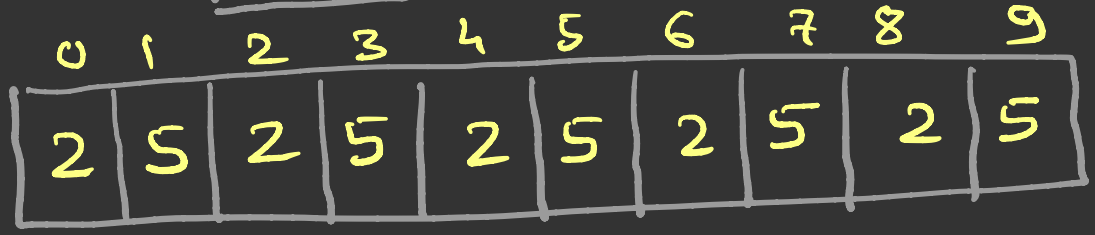

```
l1 += l2  ⟶  l1 = l1 + l2
```

Repetition Operator

l1 = [2,5]



l1 * 5



Slicing Operator

listObject [beg : end : step]

l1 = [20 , 40 , 10 , 30 , 60 , 50]

By default step = 1

By default beg , end = extreme end

l1 [2 : 6 : 2]

l1 [4 : 0 : -1]