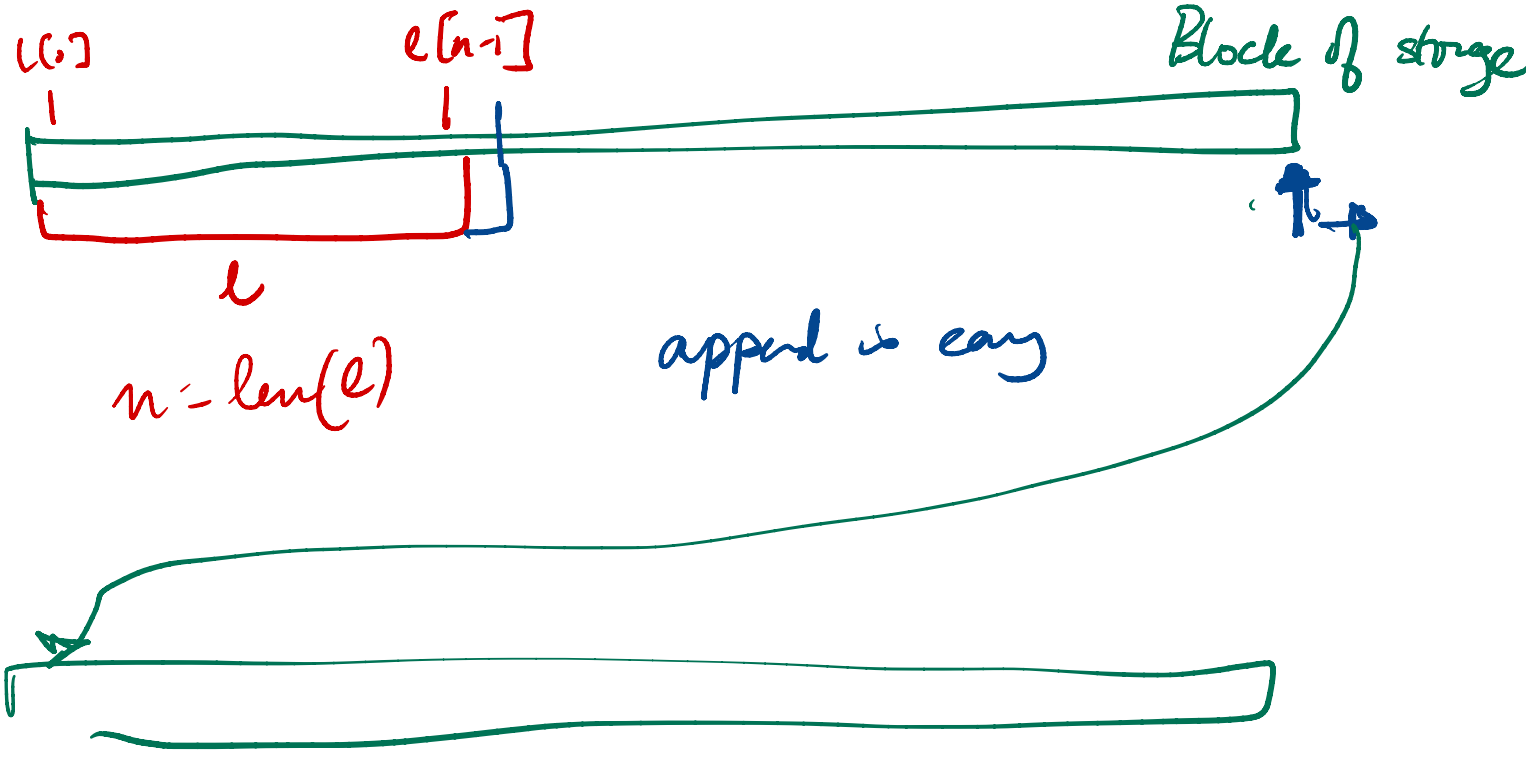
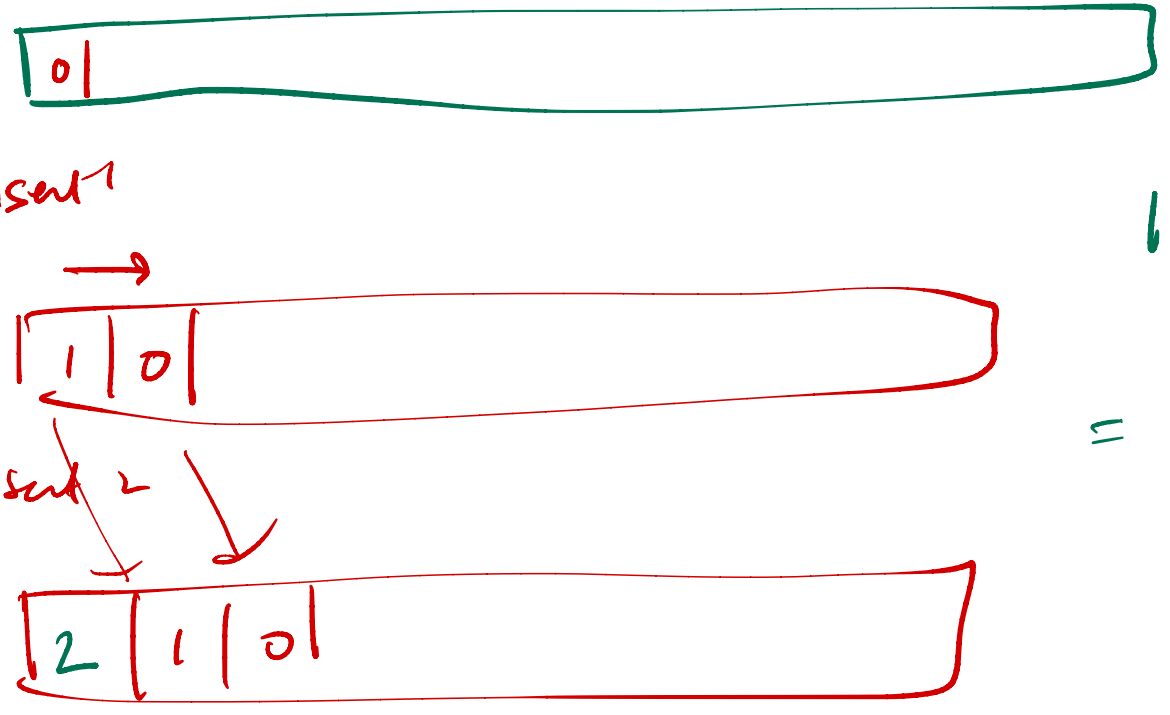
# Python list





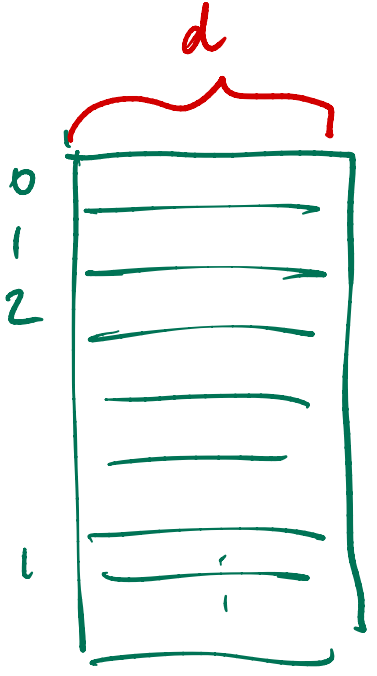

Insert n times

$1 + 2 + 3 + \cdots n-1$

$= \frac{n(n-1)}{2}$

Python can execute approx $10^7$ op/sec

Array

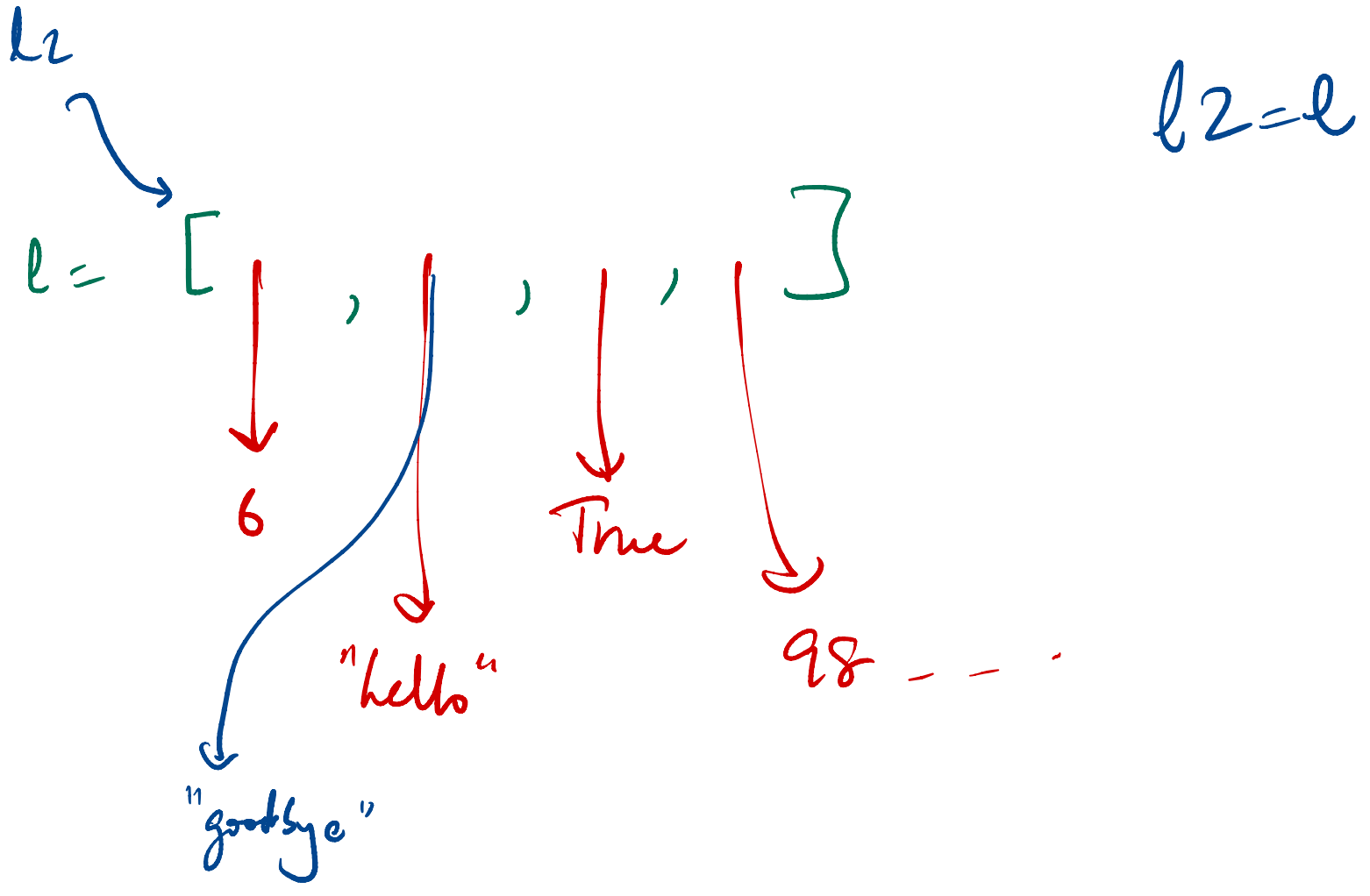



$a[i]$ is at $a[0] + i * d$

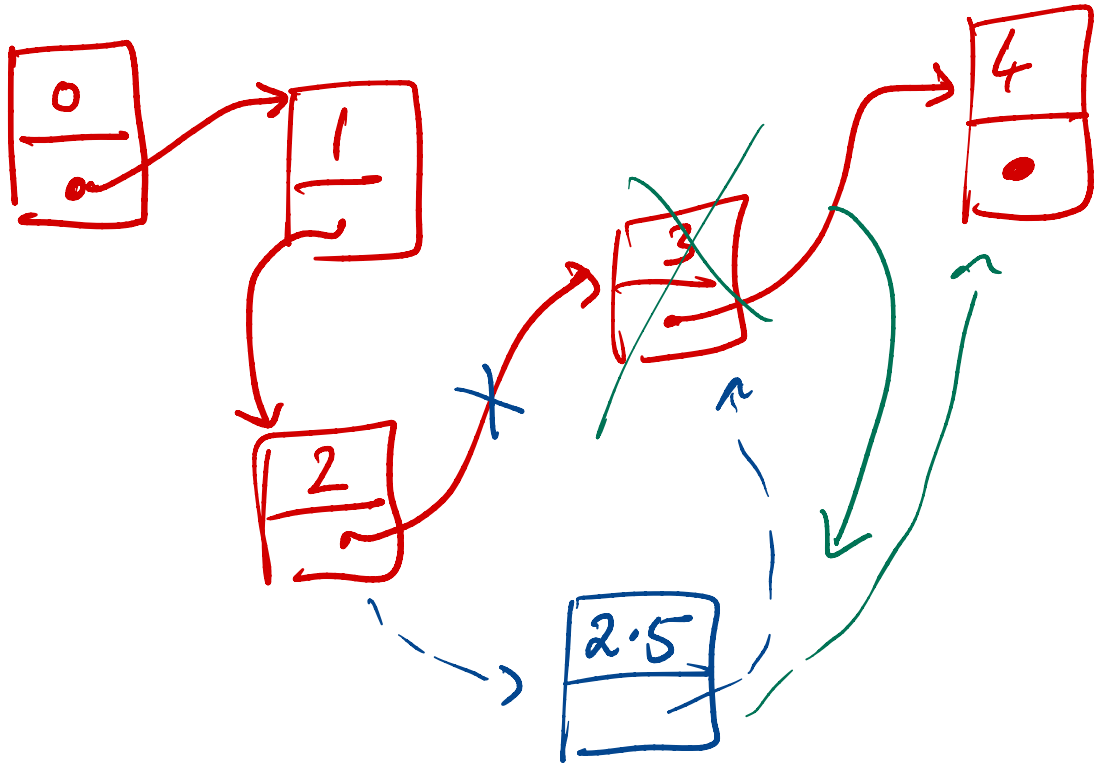
Python list


Not of fixed size

How does array implementation work?


l2
l2=l
l=
[
,
,
,
]
6
"hello"
True
98 - - -
"goodbye"

# Linked lists in Python



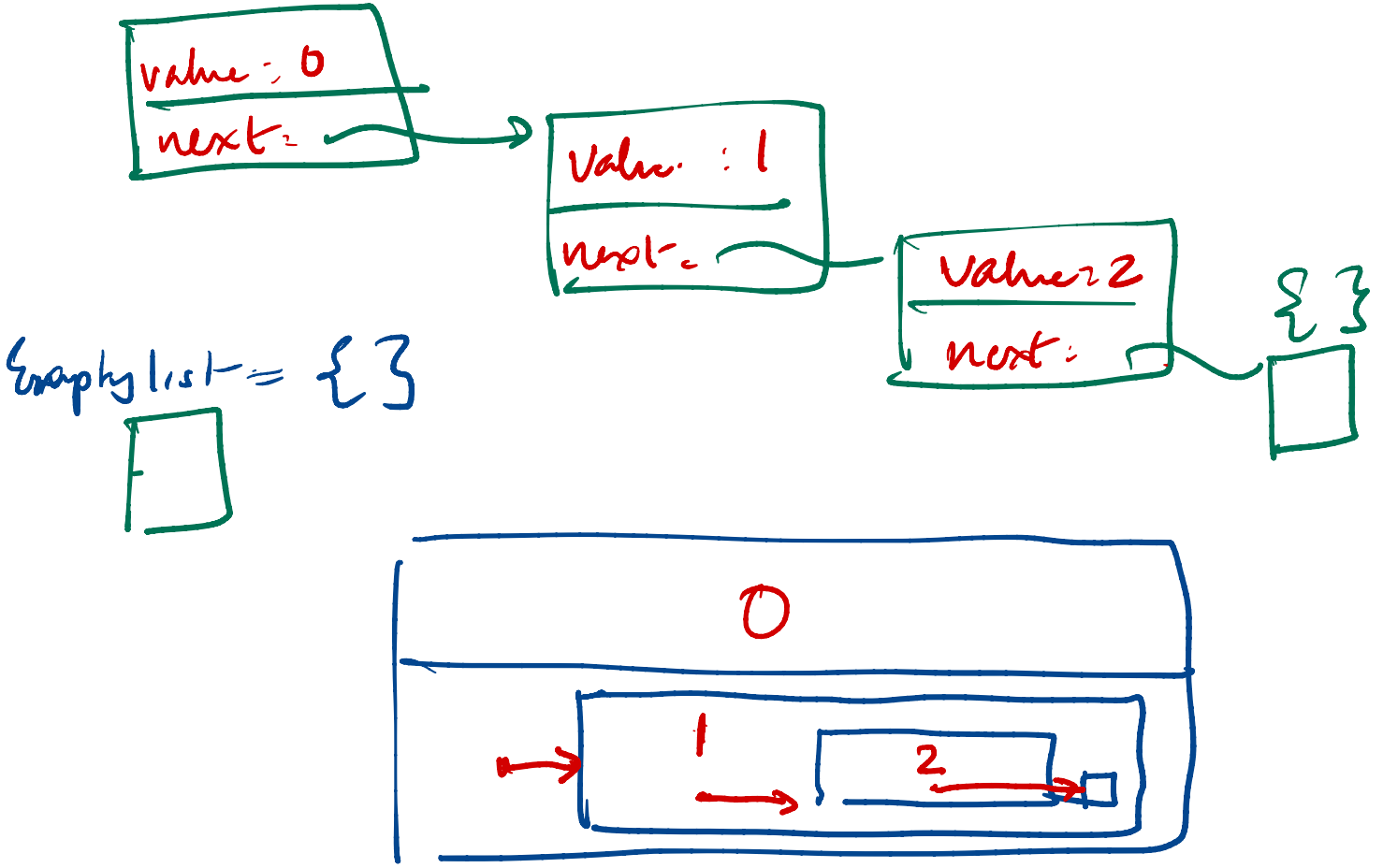
$l_0 = \{$"value": 0,

"next": $l_1 = \{$"value": 1

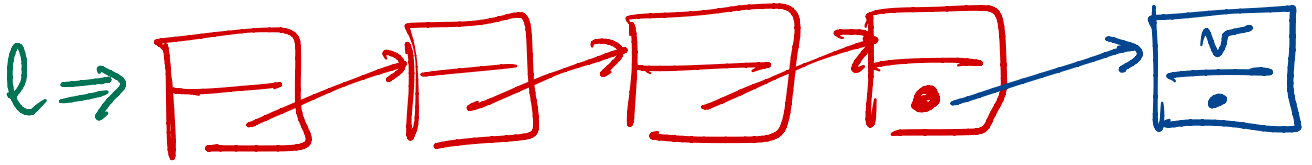
"next": $l_2 = \{$

$l_0$["next"] ⇝

$l_0$["next"]["value"]

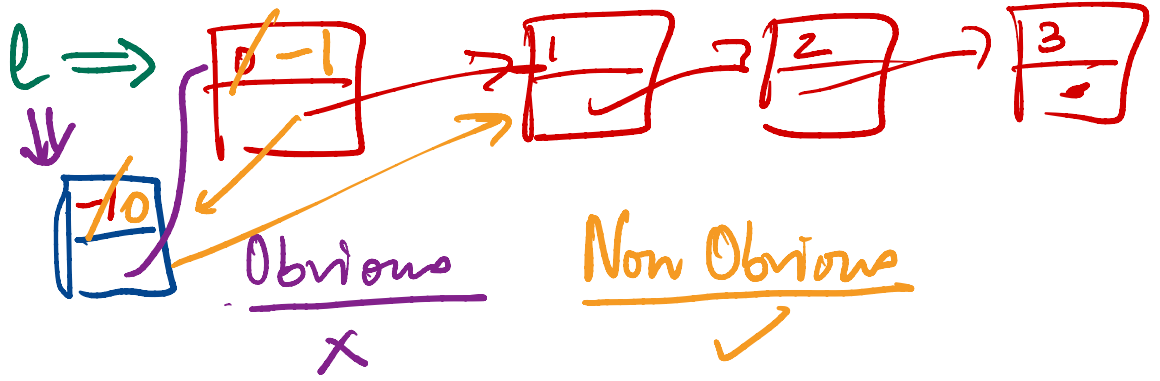

value : 0
next :
Value : 1
next :
Value : 2
next :
{}
Empty list = {}
0
1
2

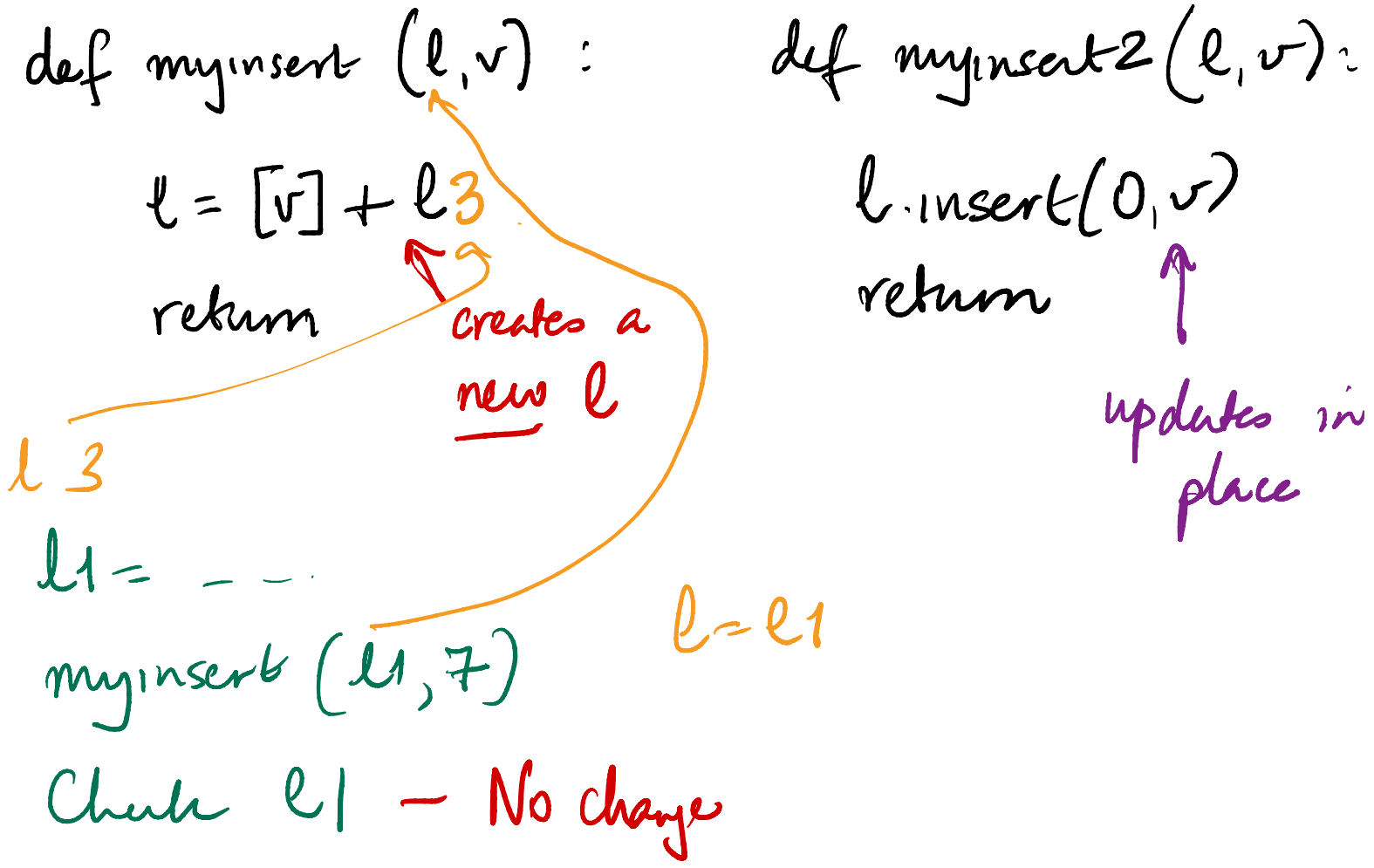
## Append




While next exists →

## Insert




def myinsert (l,v) :
l = [v] + l3
return
creates a new l
l 3
l1 = - -
myinsert (l1,7)
l=l1
Check l1 — No change
def myinsert2 (l,v):
l.insert(0,v)
return
updates in place