

Lecture 5, 25 August 2026

Madhavan Mukund

<https://www.cmi.ac.in/~madhavan>

Programming and Data Structures with Python

Lists

Indexing

Functions on lists

isprime(n)

↑
name

↑
arguments

range(2, 6, 3)

①. l.append(v)
name argument

$\approx$ append(l, v)

Ask l to append v to itself



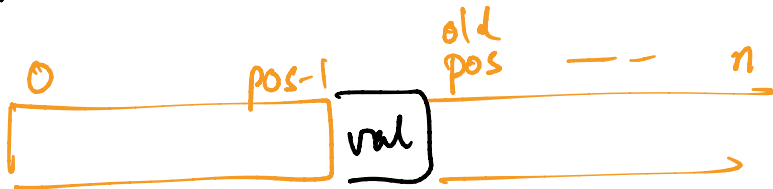
"Object Oriented"

$l + [v] \rightsquigarrow$ new list

$l[pos] = val$

— lose element $l[pos]$

$l.insert(pos, val)$



$l = [1, 5, 3]$

$newl = l$

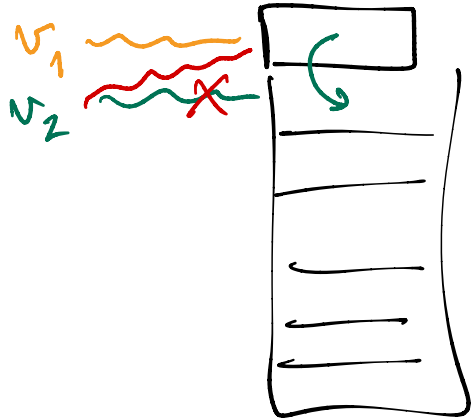
$newl.sort() \rightsquigarrow [1, 3, 5]$

name $\rightsquigarrow$ 

$$v2 = v1$$

No options

- ① Copy value of v_1 to a new space
- ② Make $v2$ point to same space as $v1$



$l = [7, 12, 3, 6, 5]$ $\leftarrow$ allow arbitrary "keys"

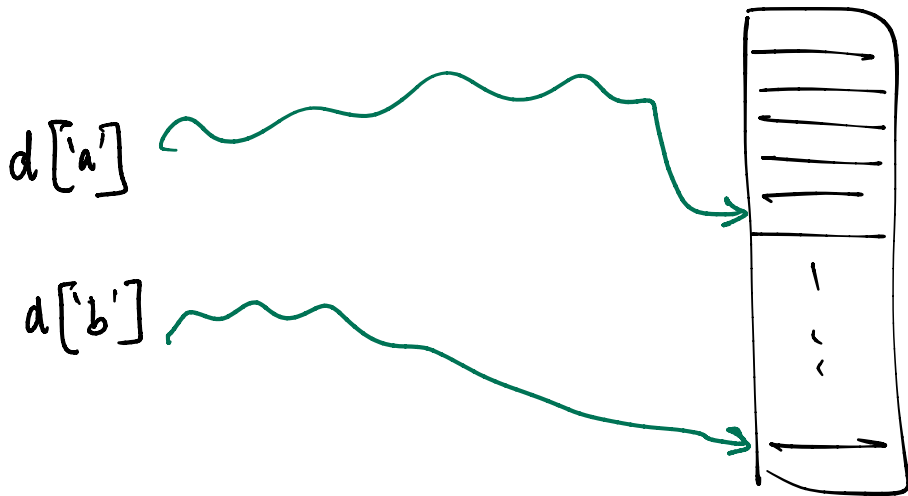
$0 \mapsto 7$

$1 \mapsto 12$

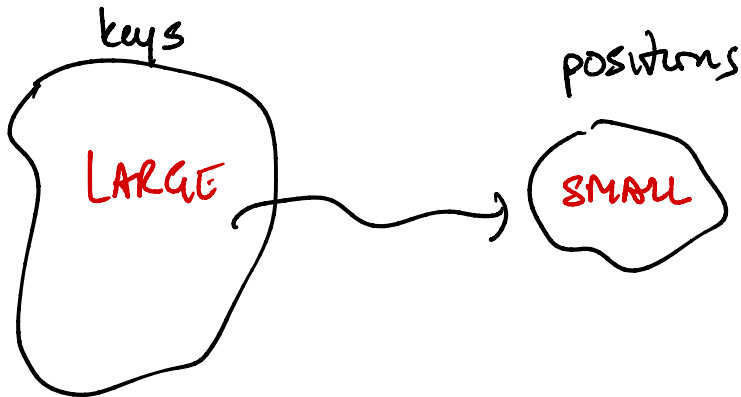
$\vdots$

$4 \mapsto 5$

How do dictionaries work?



Map keys to locations



hashing

$$h: X \rightarrow Y$$

large small

Ideally

$$x \neq y$$

$$\Rightarrow h(x) \neq h(y)$$



$$x \neq y, h(x) = h(y)$$

"collision"

Assume

Hashing is "perfect" in Python dictionaries

Ignore collisions