Course	
Term	
Week	
Date	
Chapter. Topic	7. Lists and Tuples

Python Lists: An introduction

Siva R Jasthi
Computer Science and Cybersecurity
Metropolitan State University

Lists

List is a collection which is **ordered** and **changeable**. Allows **duplicate** members.

Lists: An introduction
https://www.w3schools.com/python/python_lists.asp

Lists: An introduction
https://openbookproject.net/thinkcs/python/english3e/lists.html

List Methods
http://www.python-ds.com/python-3-list-methods

Built-in Functions
https://docs.python.org/3/library/functions.html

Lists

ordered (index starts with 0)
duplicates are allowed. (10 is showing up thrice)
changeable. (nums_1[0] = 99)

`nums_1` →

10	8	7	9	10	9	10	3	4
0	1	2	3	4	5	6	7	8

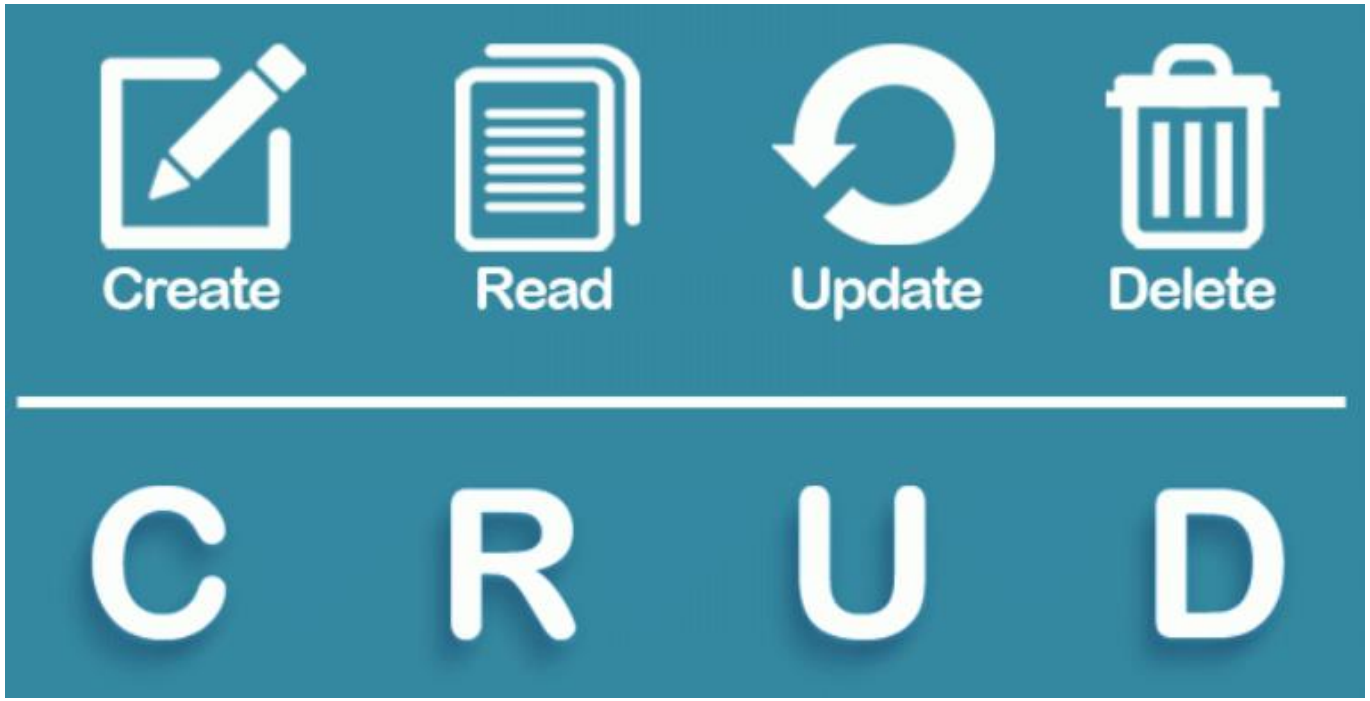
`nums_1` →

99	8	7	9	10	9	10	3	4
0	1	2	3	4	5	6	7	8

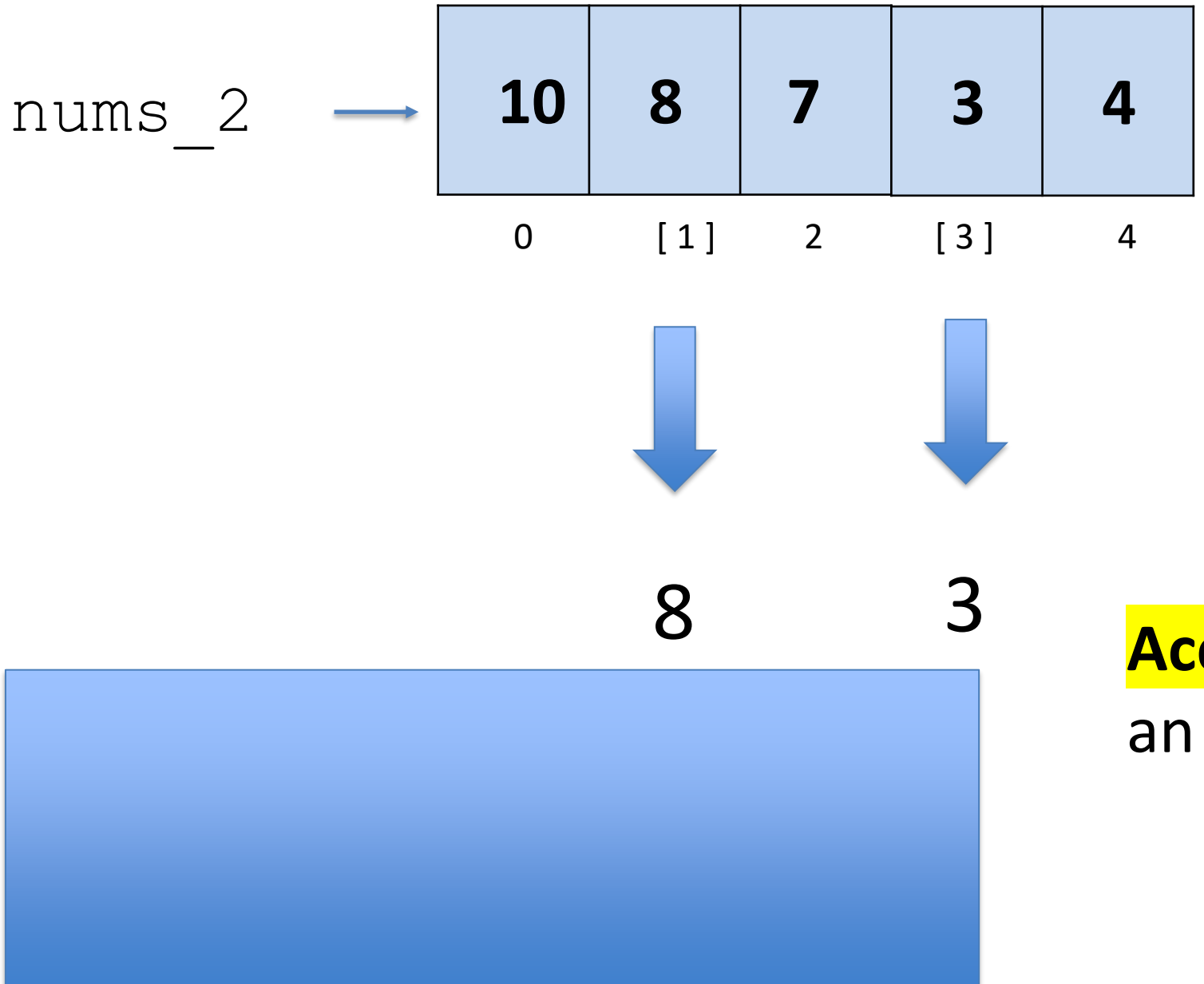
Lists are everywhere

CRUD

C – Create (Add, Insert, Append, Extend, Copy)
R – Read (Query, Traversal, Find, Search, Visit, Navigating)
U – Update (Modify, Change, Edit, Replace, Substitute)
D – Delete (Remove, Empty, Clear)



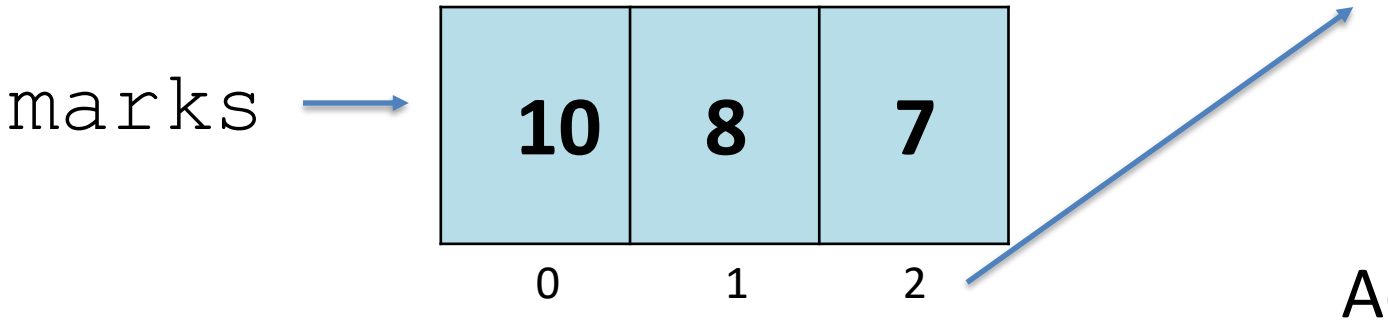
List Methods




```
X = nums_2[1]
Y = nums_2[3]
print(X*Y)
```

Accessing (Reading) an element from the list

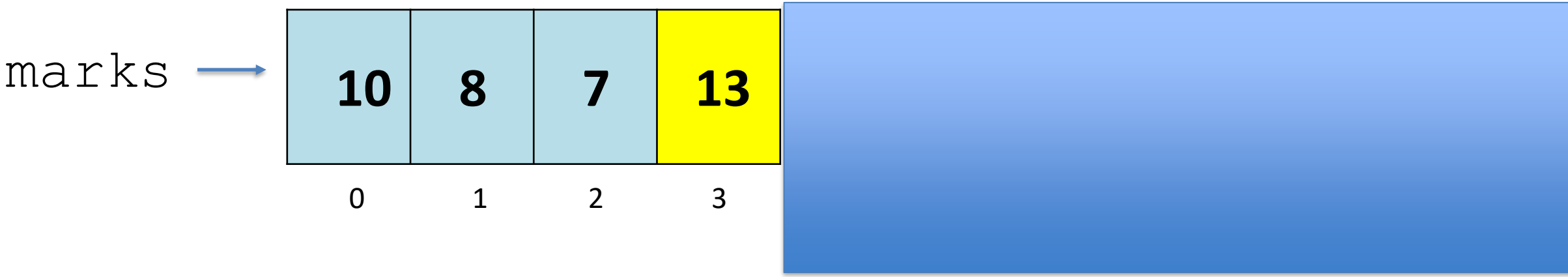
List Methods



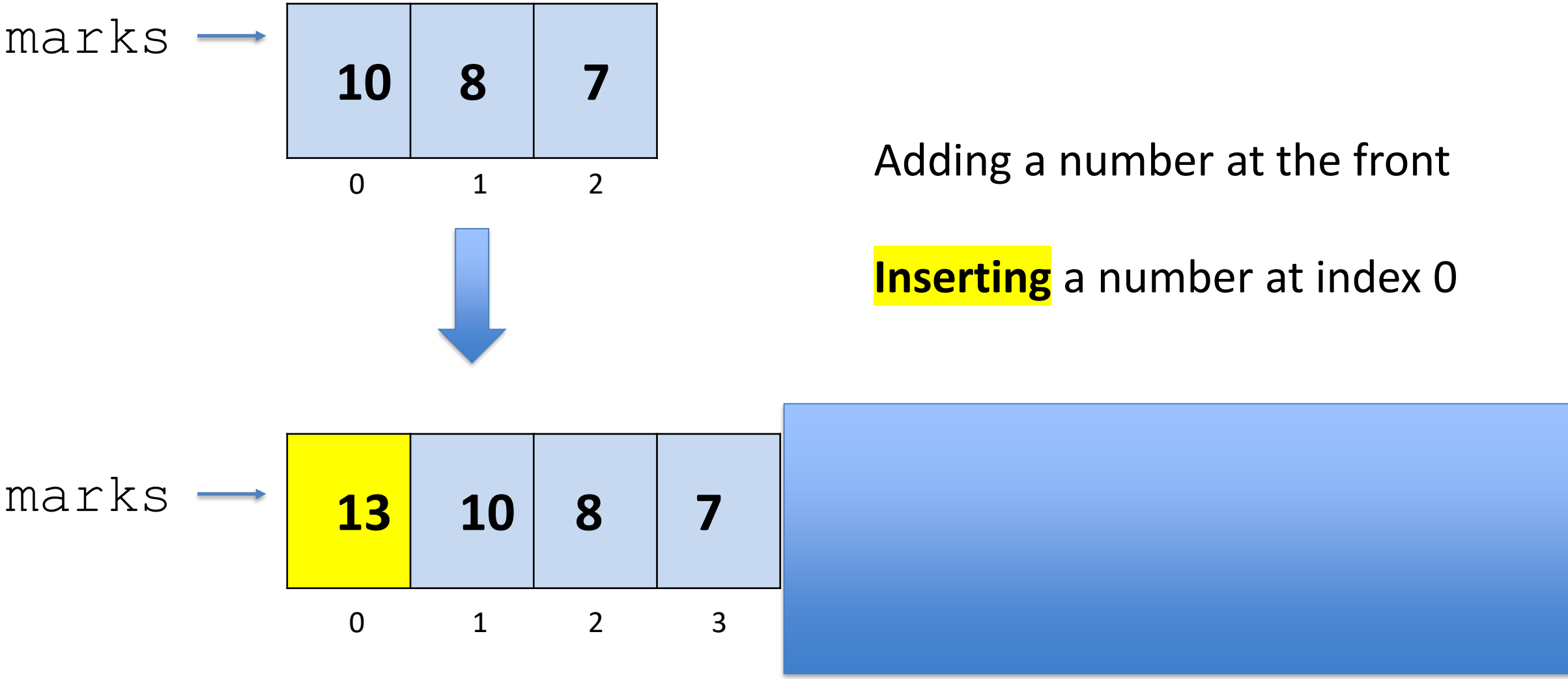
Computer's first slot starts with ZERO (0)

Adding a number at the end

Appending a number



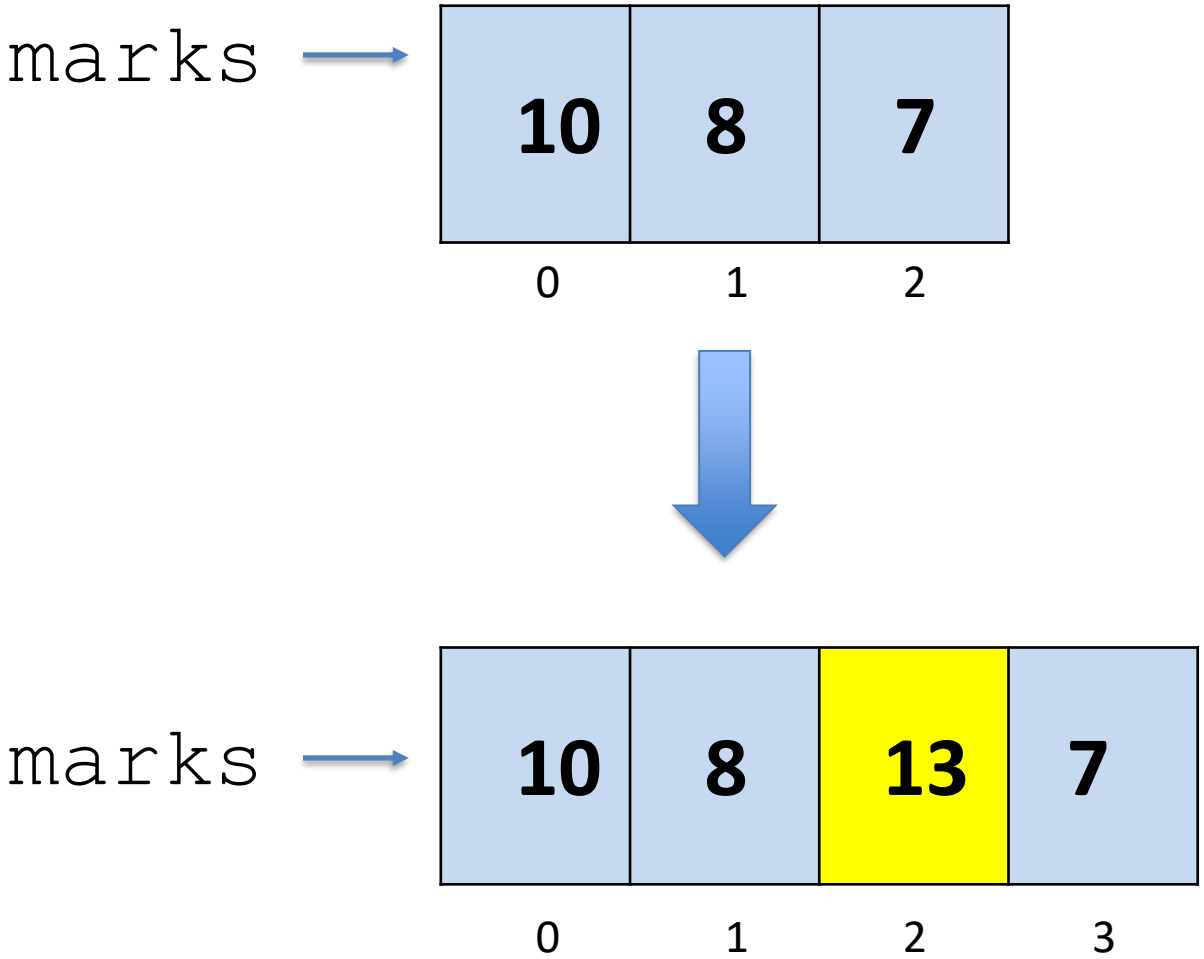
List Methods



Adding a number at the front

Inserting a number at index 0

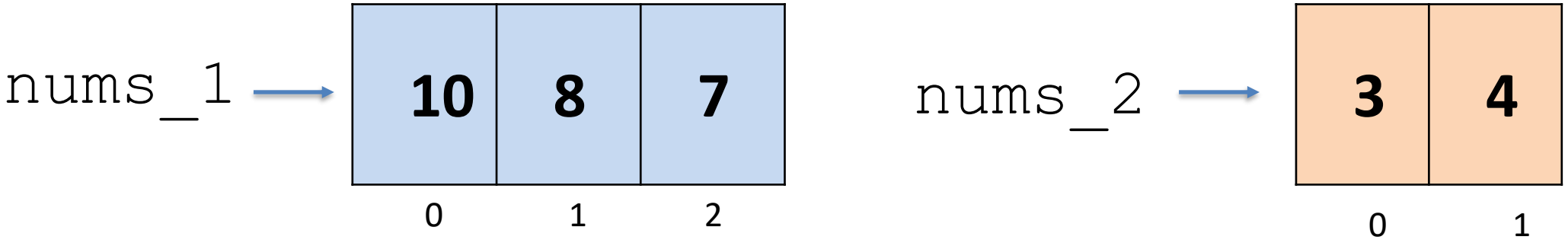
List Methods



Adding a number anywhere in the list

Inserting a number at index 2

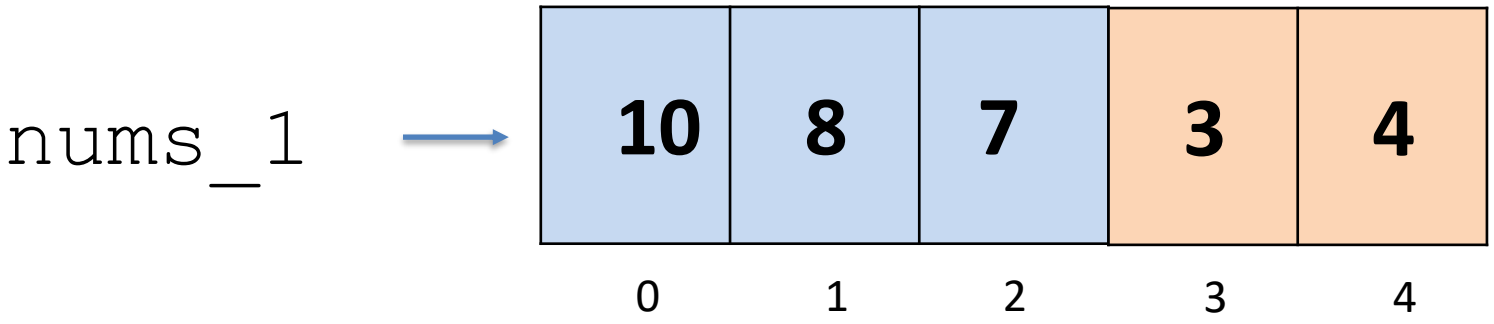
List Methods



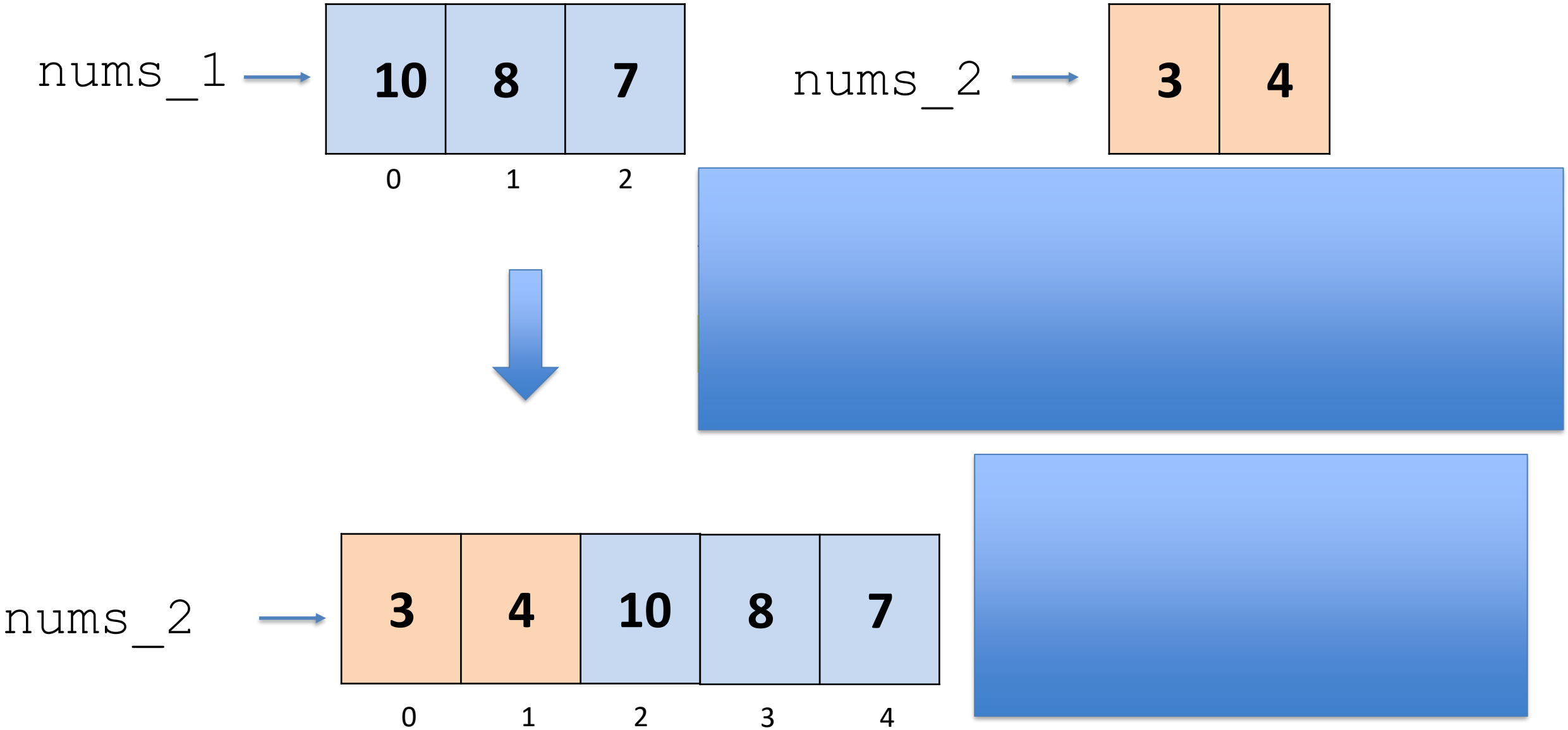
Adding many numbers to the end of the list

Extending a list with another list

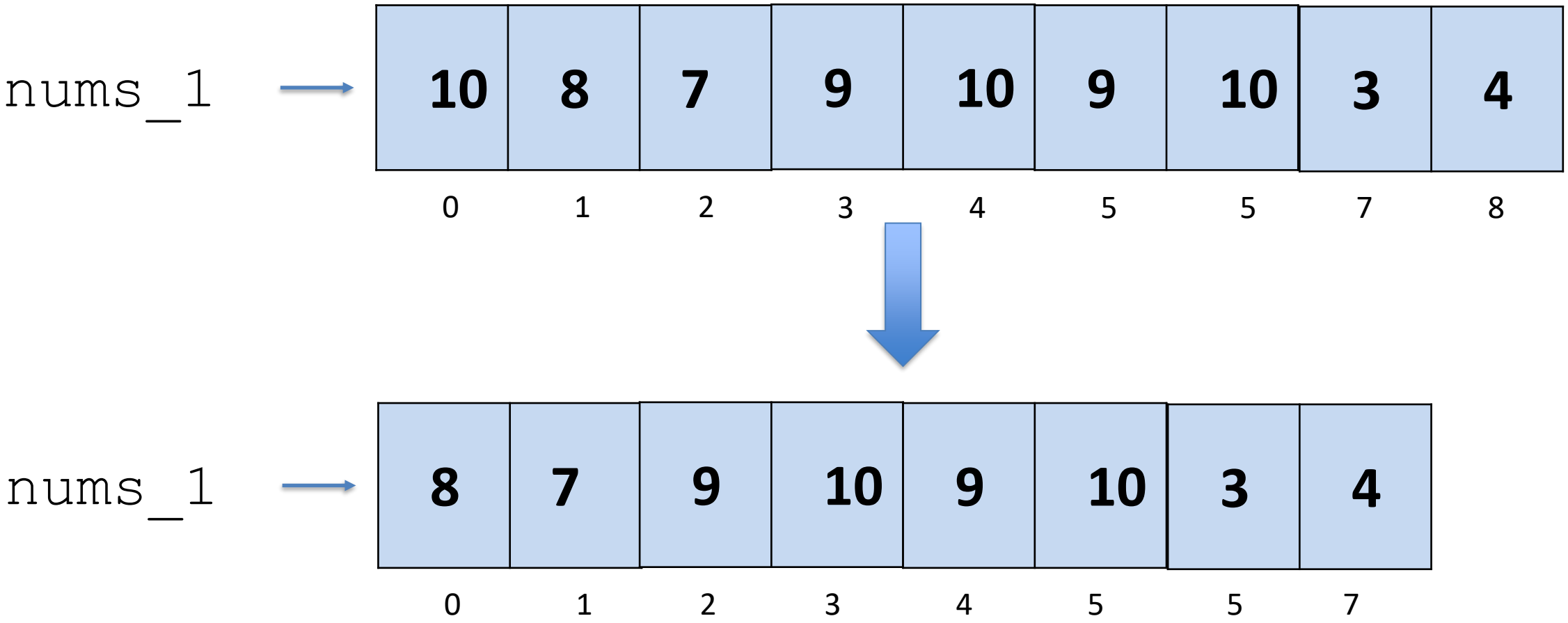
Extending nums_1 with nums_2

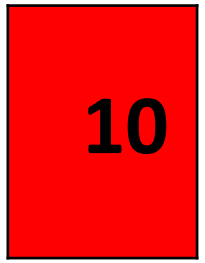


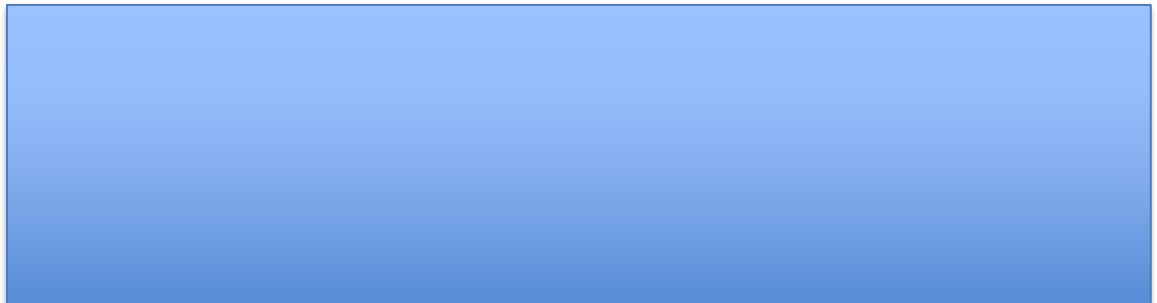
List Methods



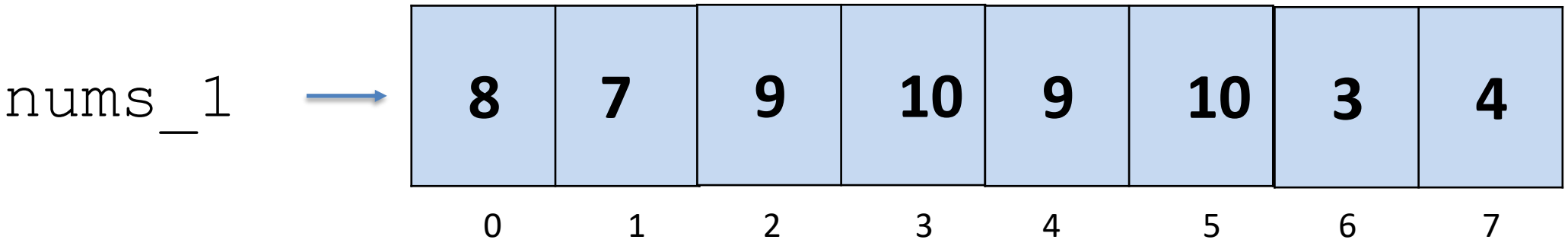
List Methods

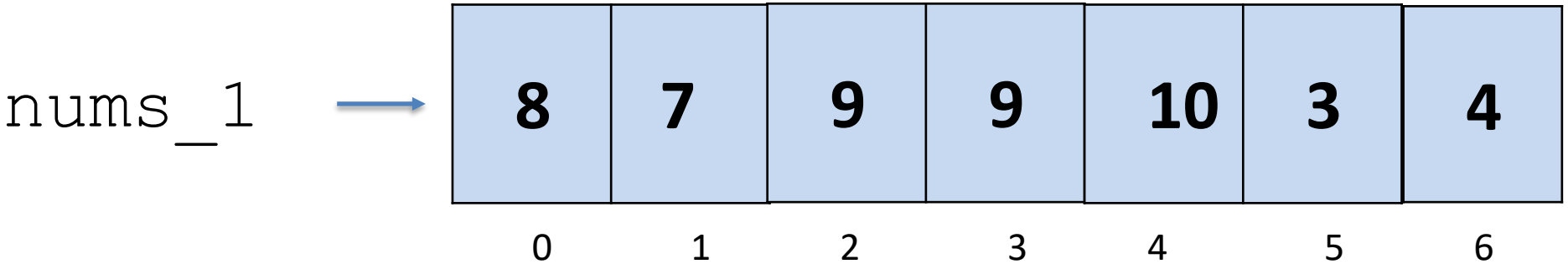


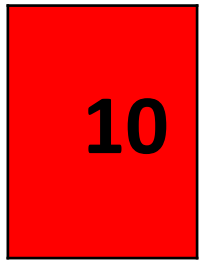


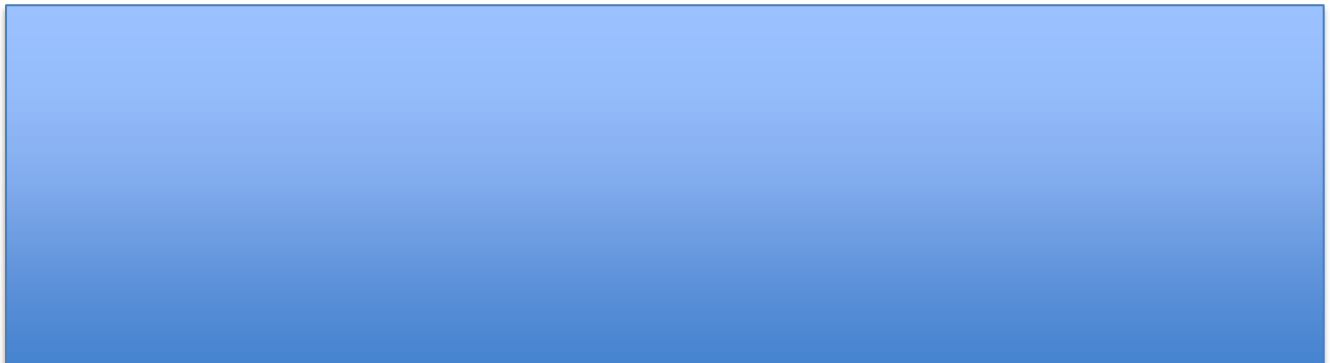


List Methods

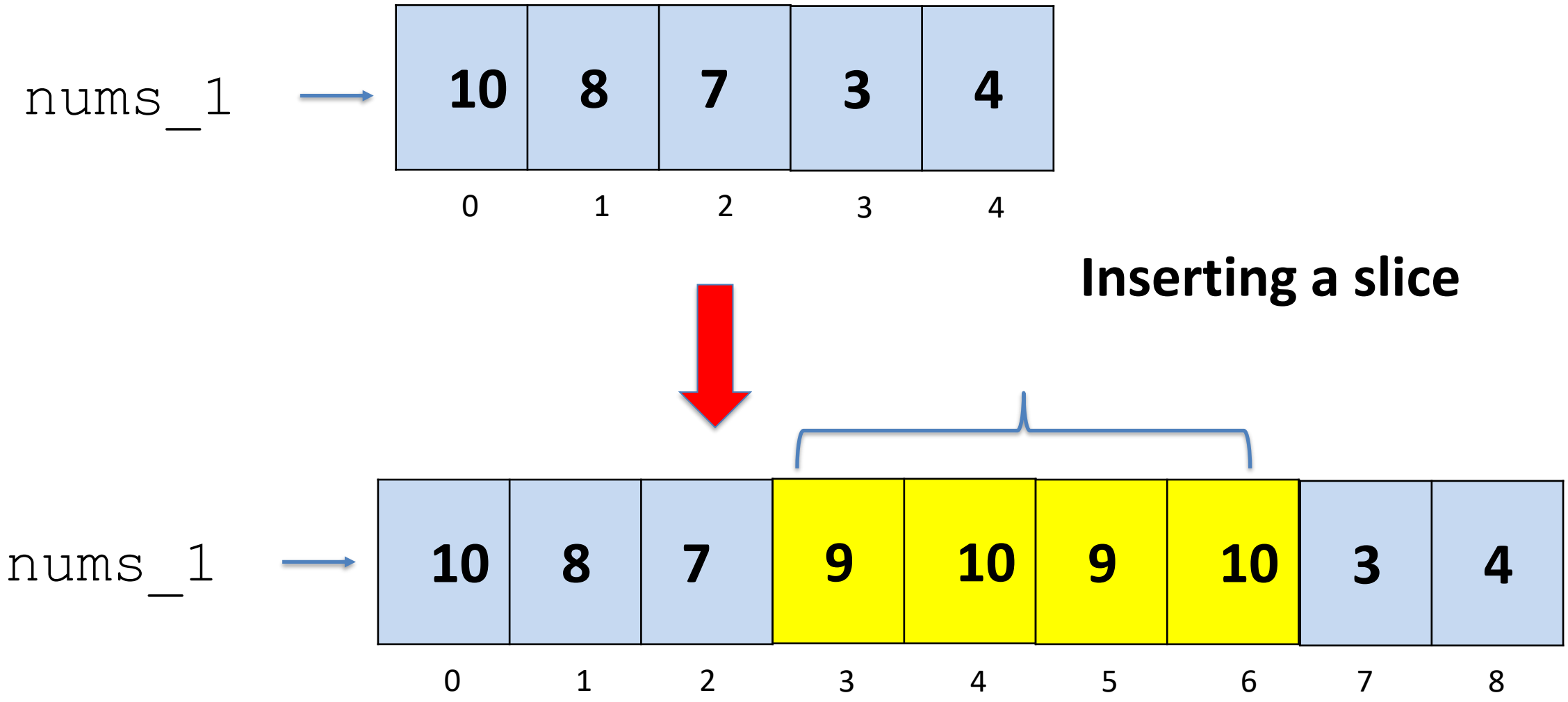




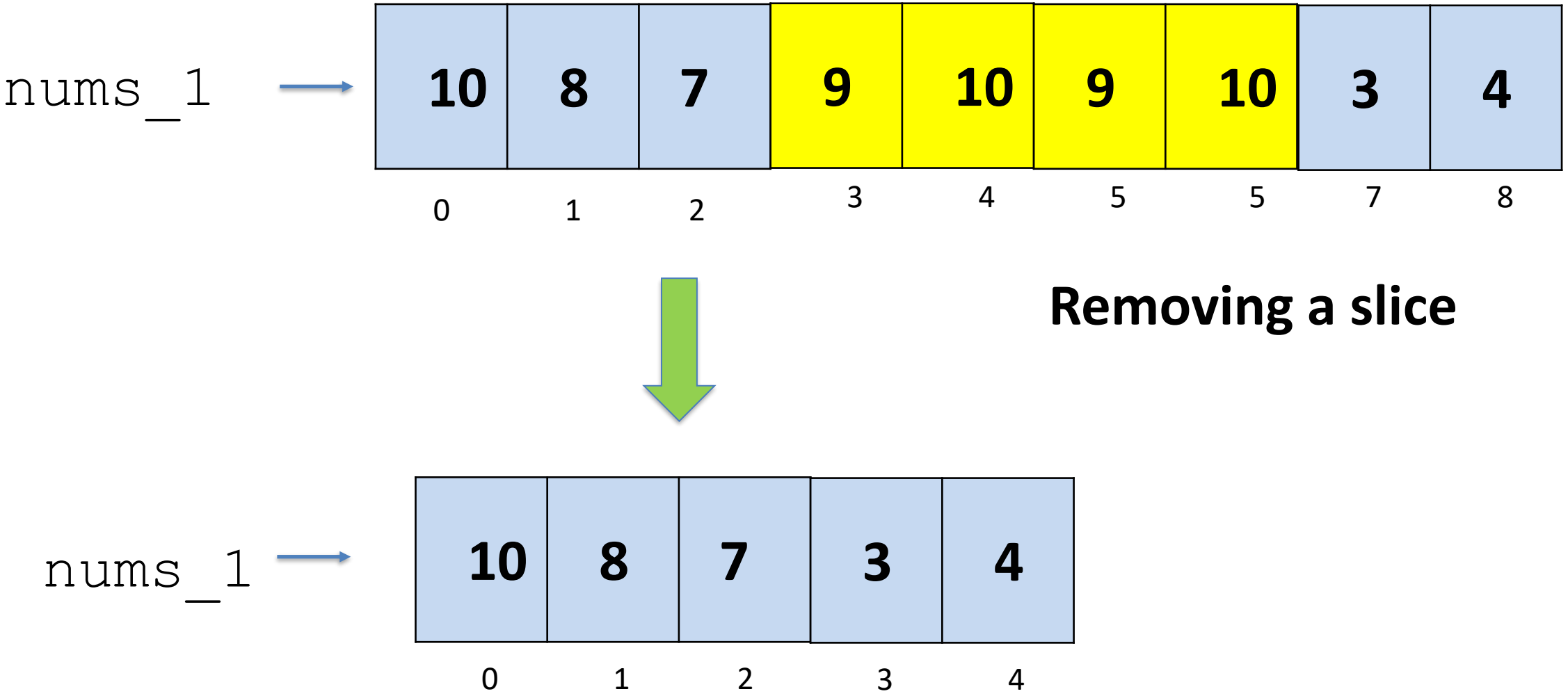




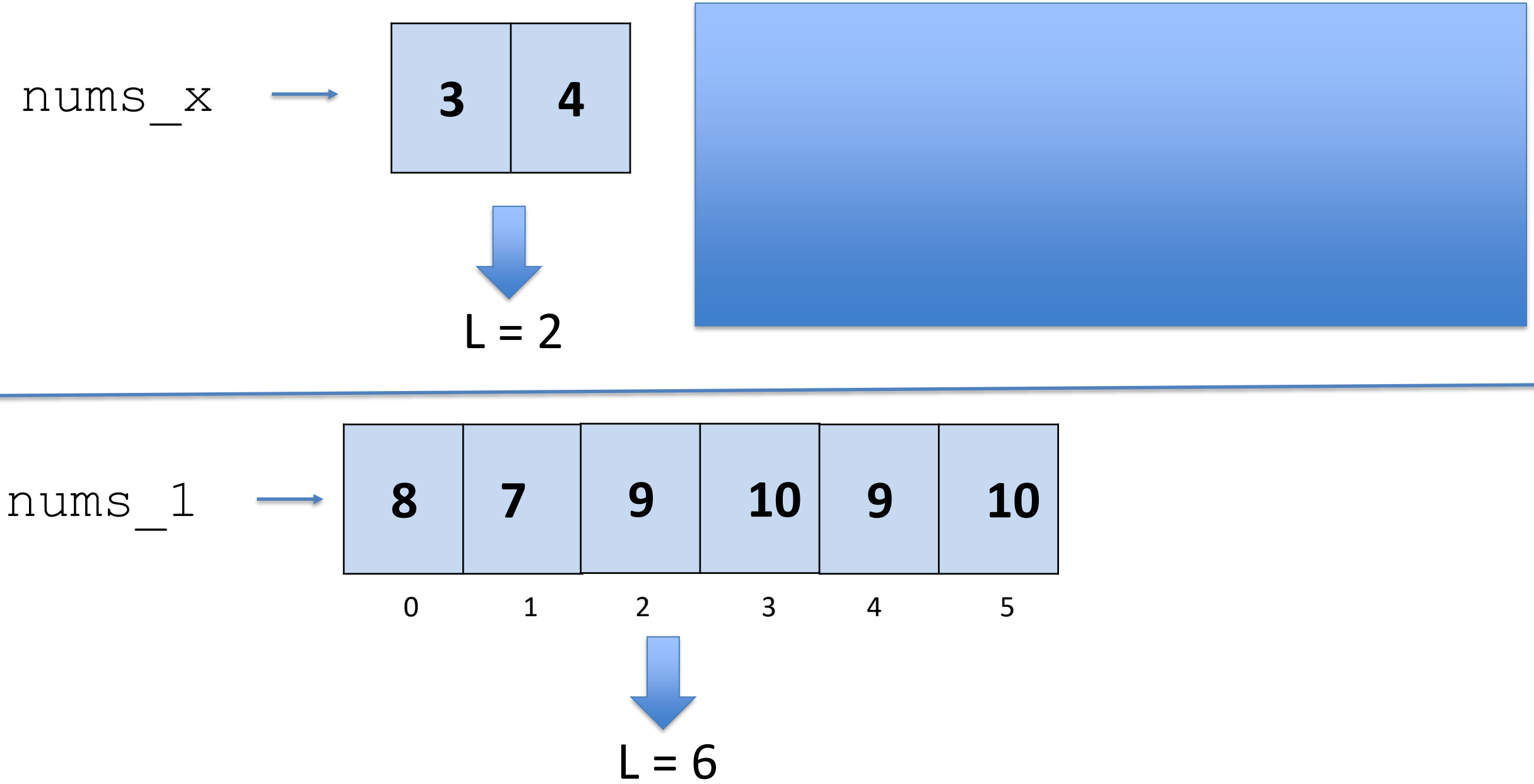
List Methods



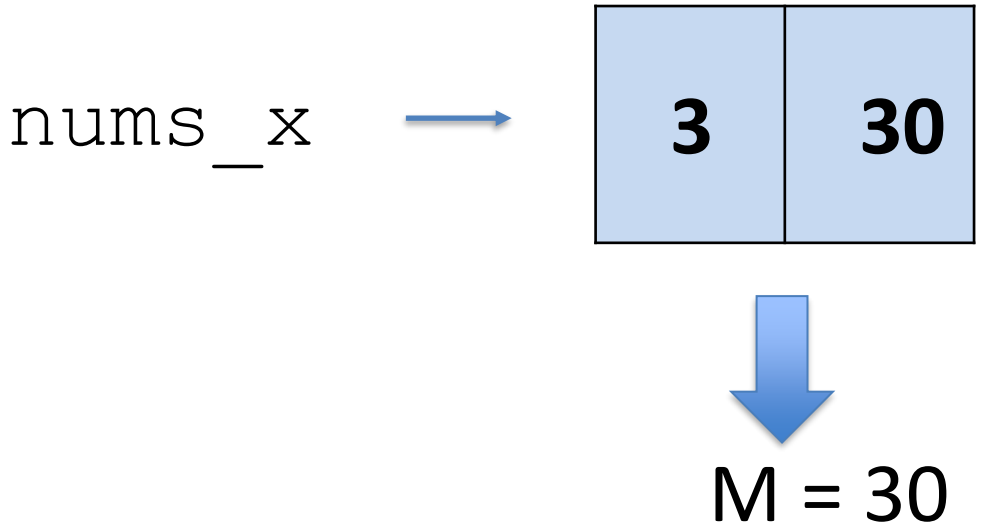
List Methods



List Methods



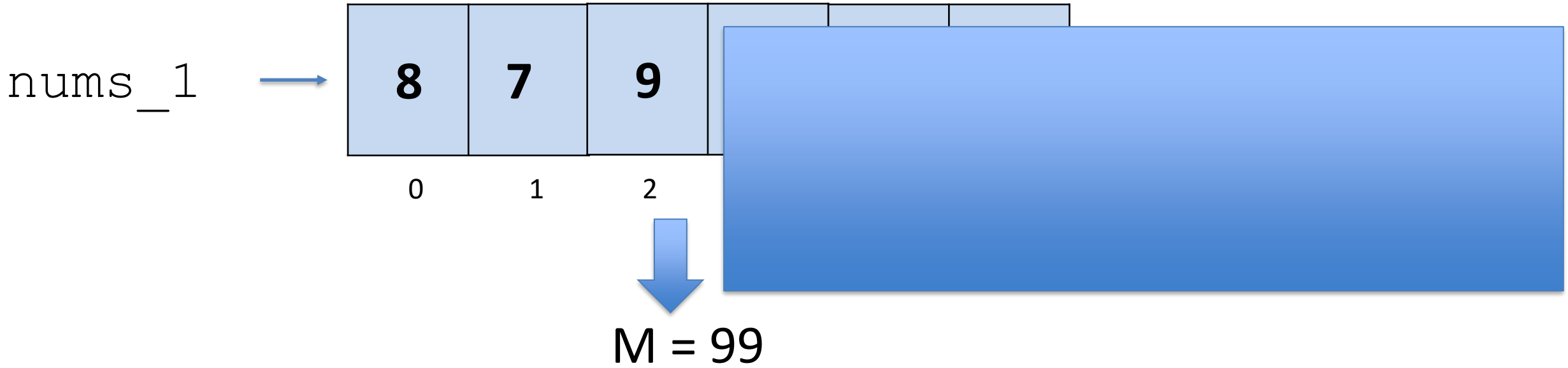
List Methods



What is the largest/biggest number in the list?

~~nums_x.max() (incorrect)~~

L = max(nums_x)



List Methods

`nums_x` → | 3 | 30 |

S = 3

What is the smallest number in the list?

`nums_1` → | 8 | 7 | 9 | 99 | 9 | 10 |

0 1 2 3 4 5

S = 7

List Methods

`nums_x` →

3	**4**

S = 7

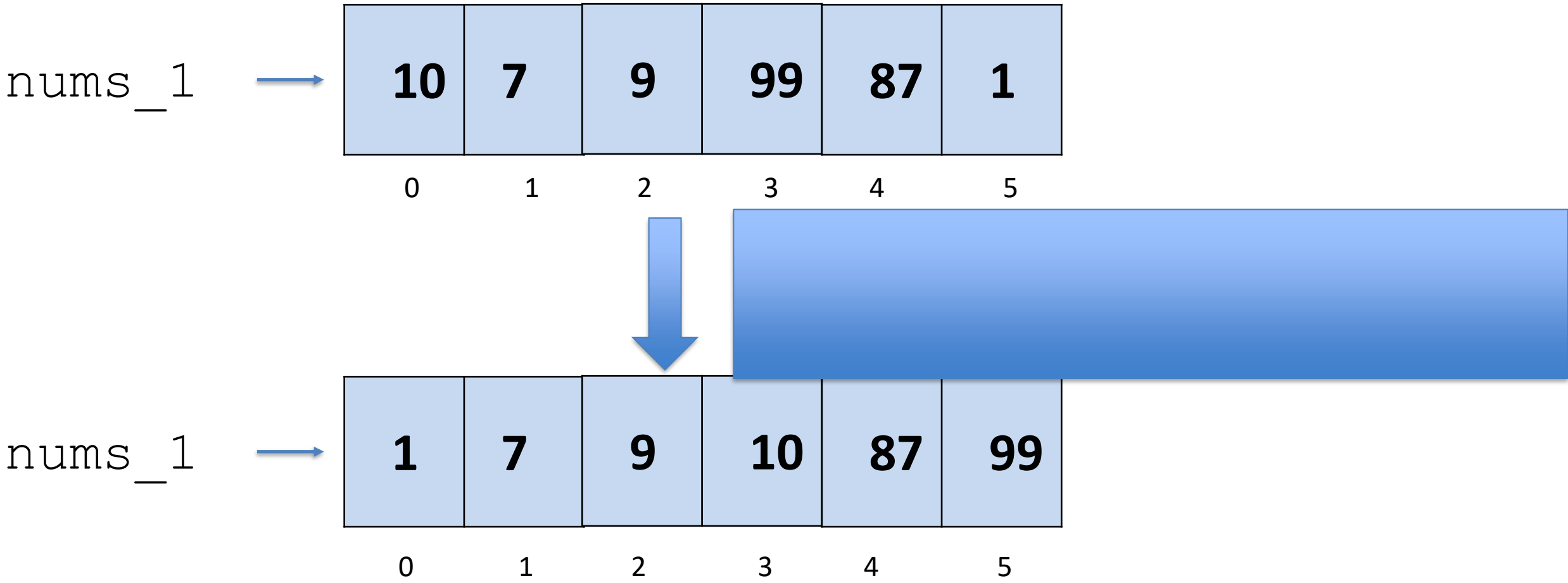
What is the sum of numbers the list?

`nums_1` →

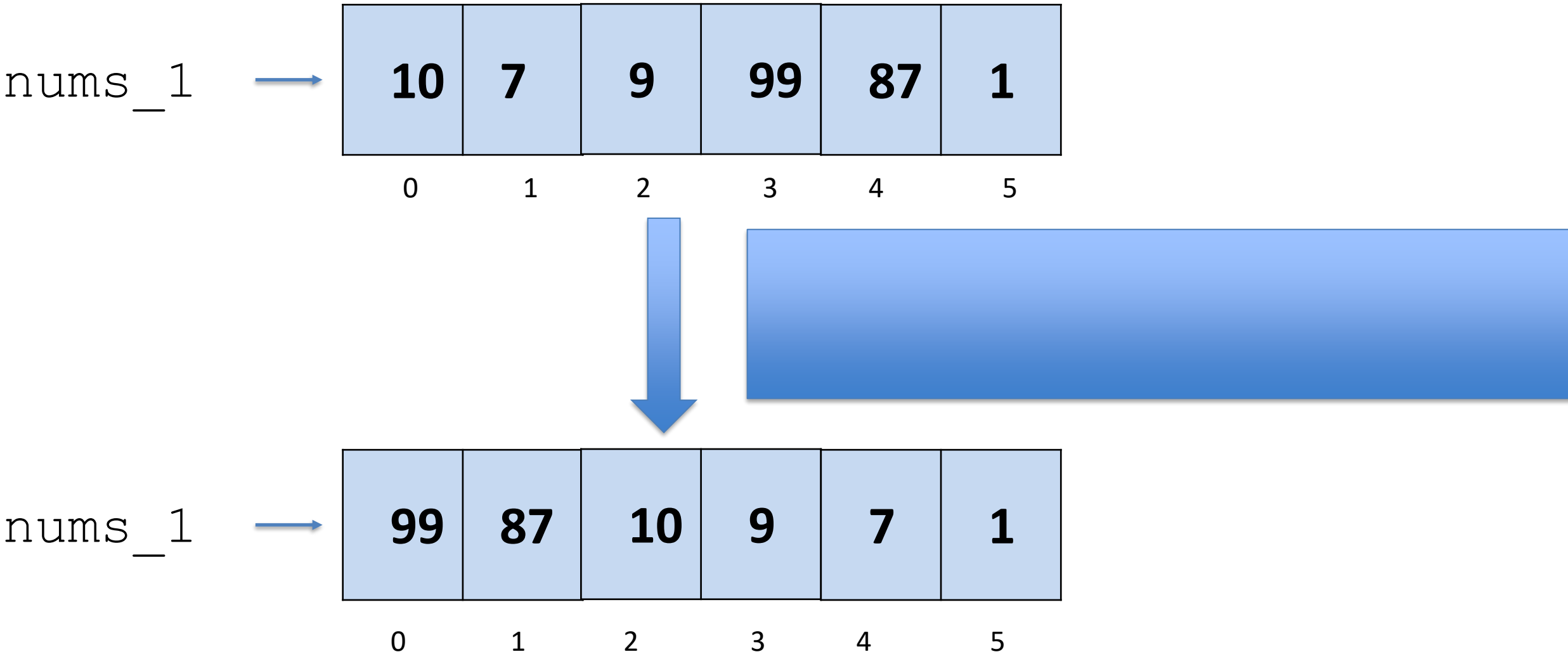
8	**7**	**9**	**10**	**9**	**10**
0	1	2	3	4	5

S = 53

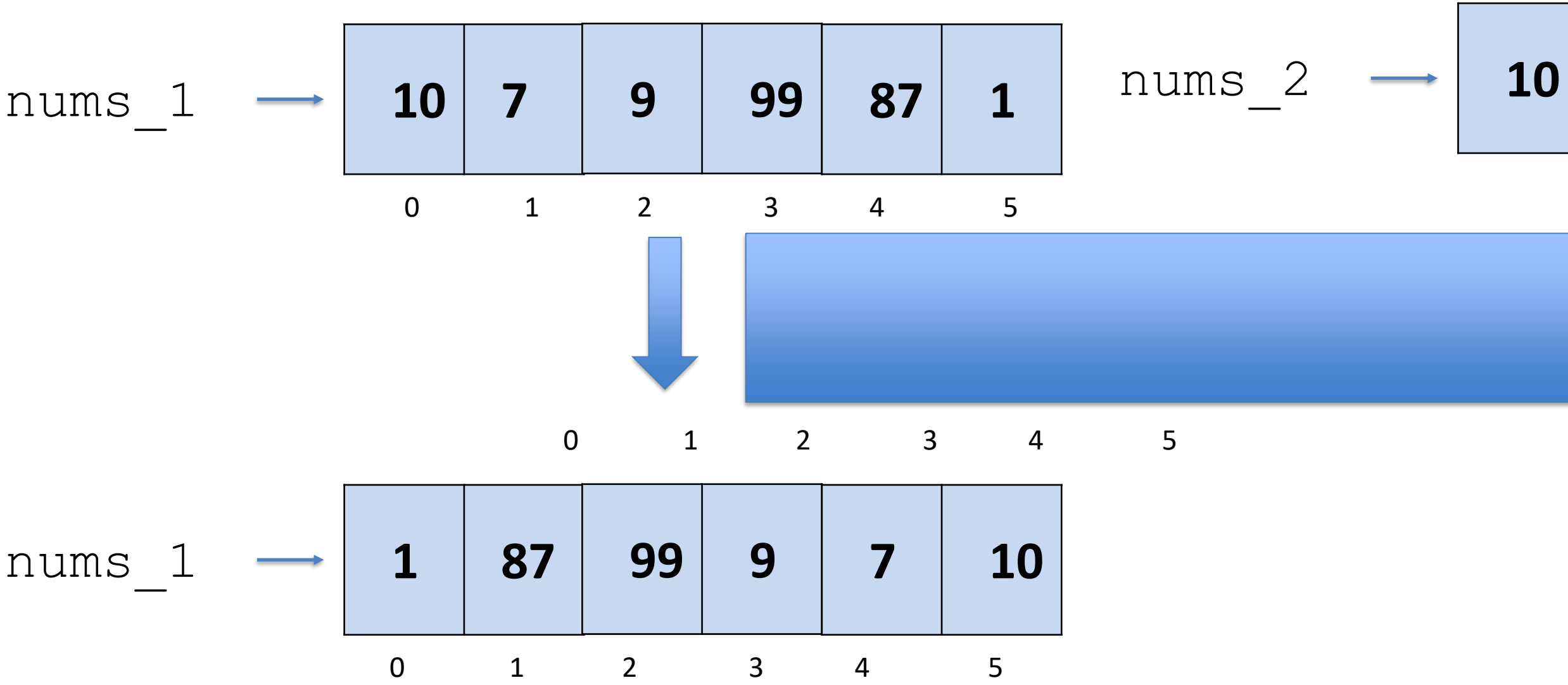
List Methods



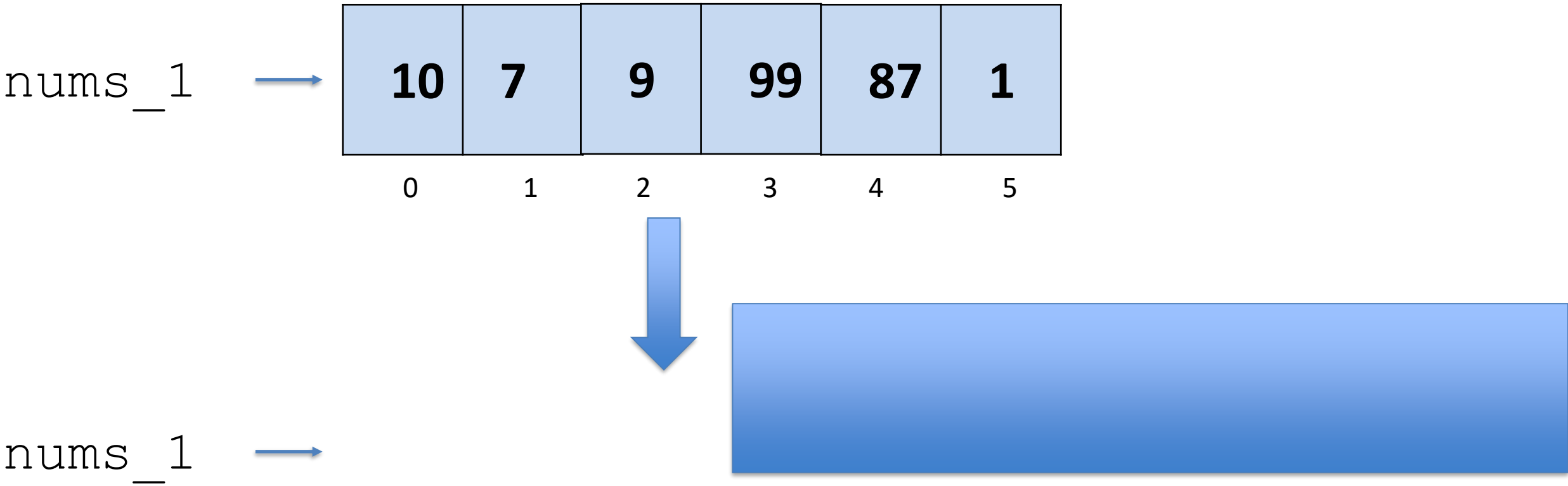
List Methods



List Methods

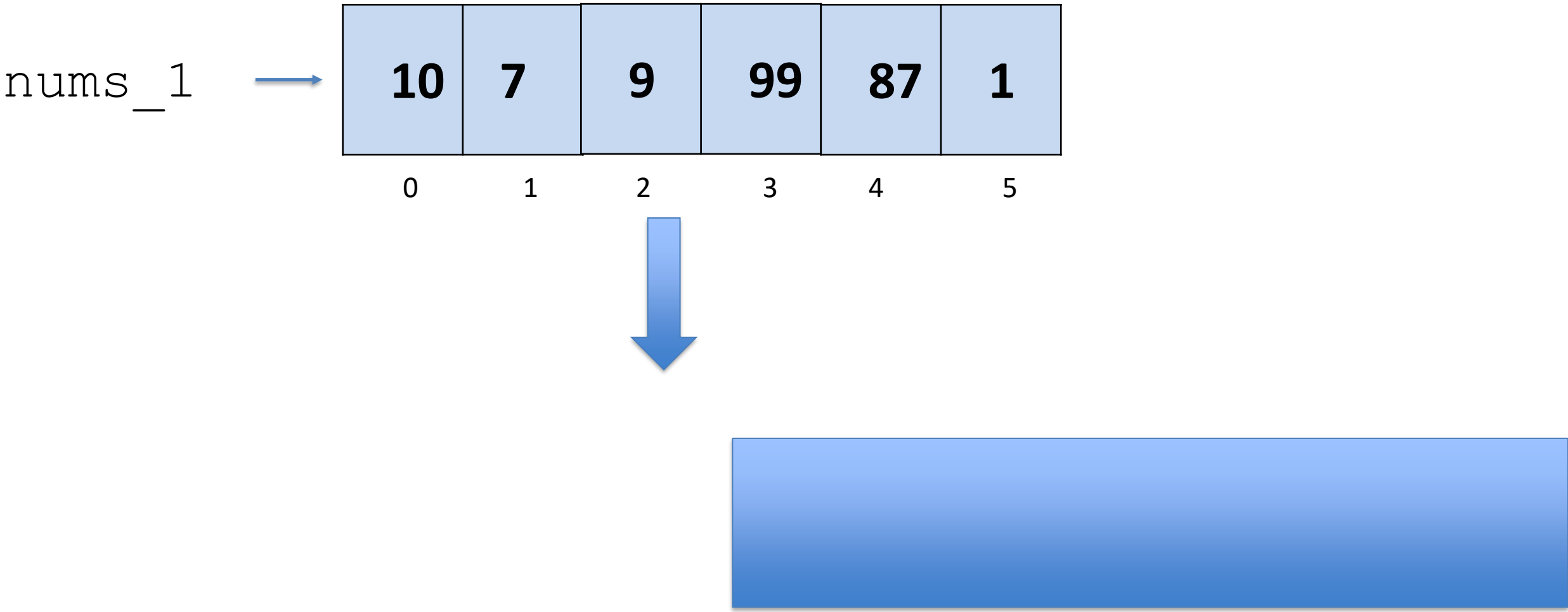


List Methods



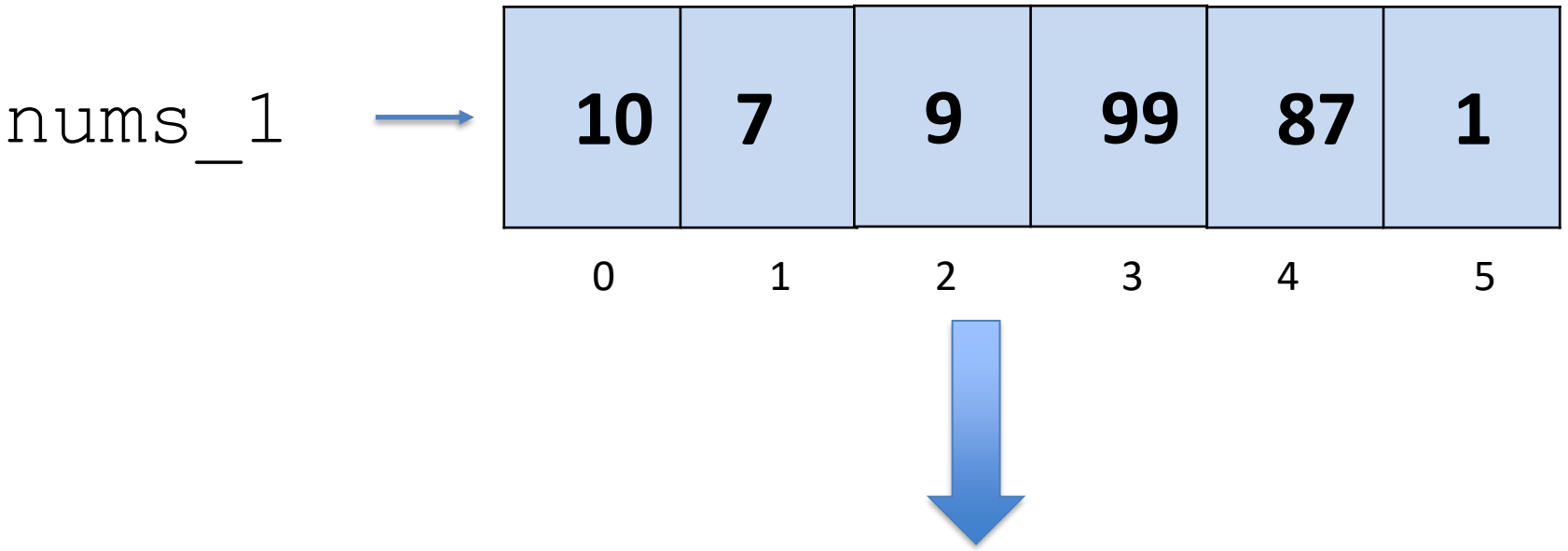
Analogy: I still have my bookcase. But it doesn't have any books in it. (CLEAR)

List Methods



Analogy: Bookcase along with the books is gone (del nums_1)

List Methods

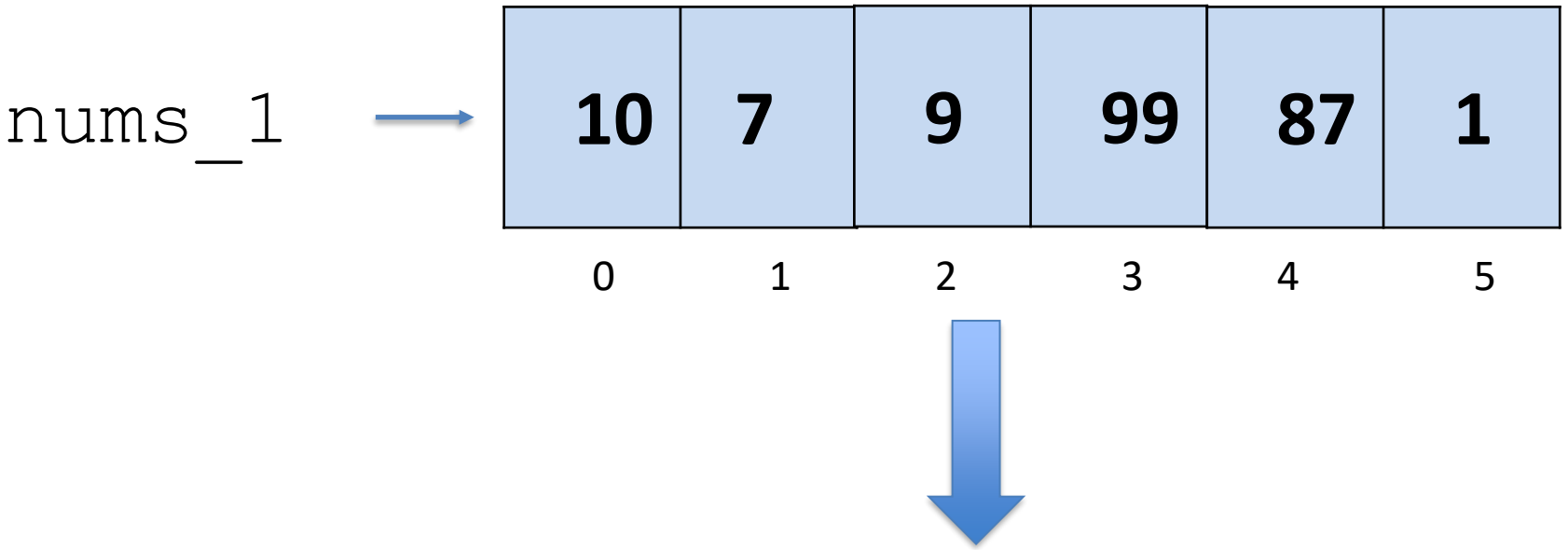


Is number 99 in the list?

X = (99 in nums_1)

A = 1 not in nums_1
B = 9 not in nums_1
C = 100 not in nums_1

List Methods



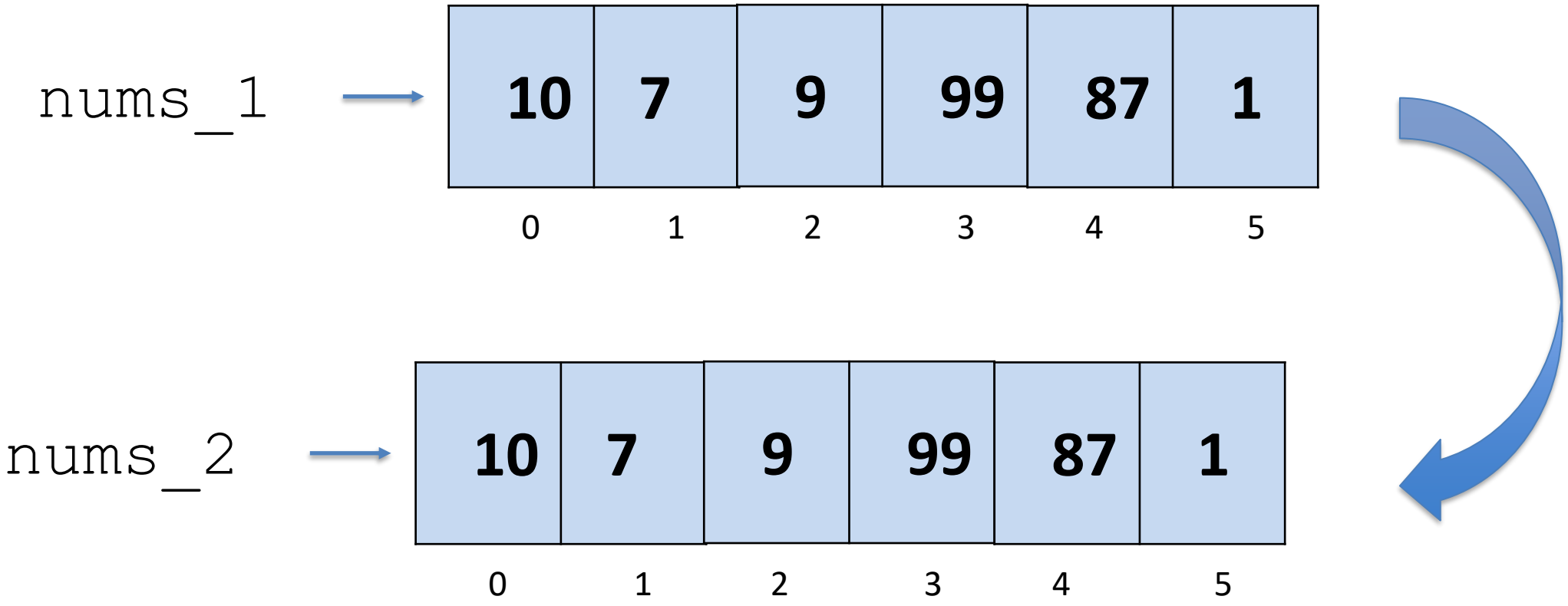
Is number 100 not in the list?

Y = (100 not in nums_1)

A = 1 not in nums_1
B = 9 not in nums_1
C = 100 not in nums_1

Checking for membership of an element in a given list

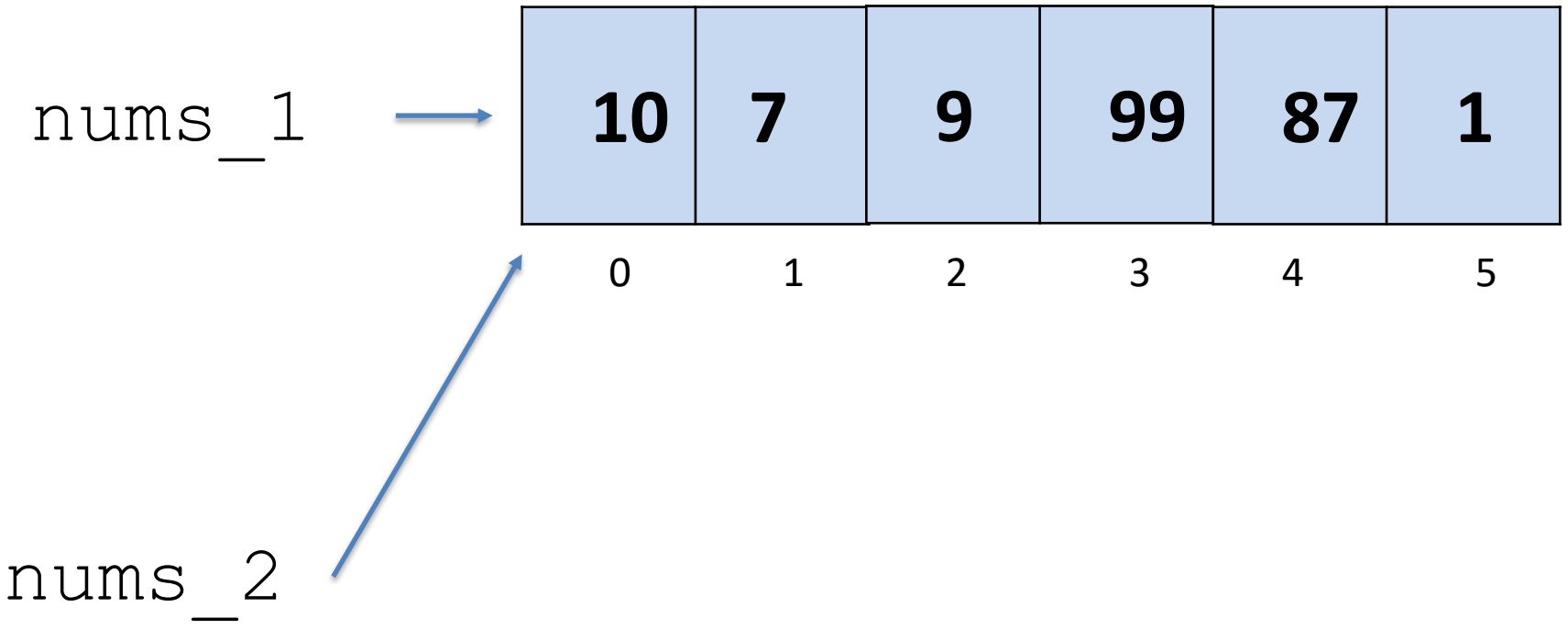
List Methods



Both nums_1 and nums_2 are pointing to different lists.

You have a TV remote 1 pointing to TV A.
Your sibling has a TV remote 2 pointing to TV B.

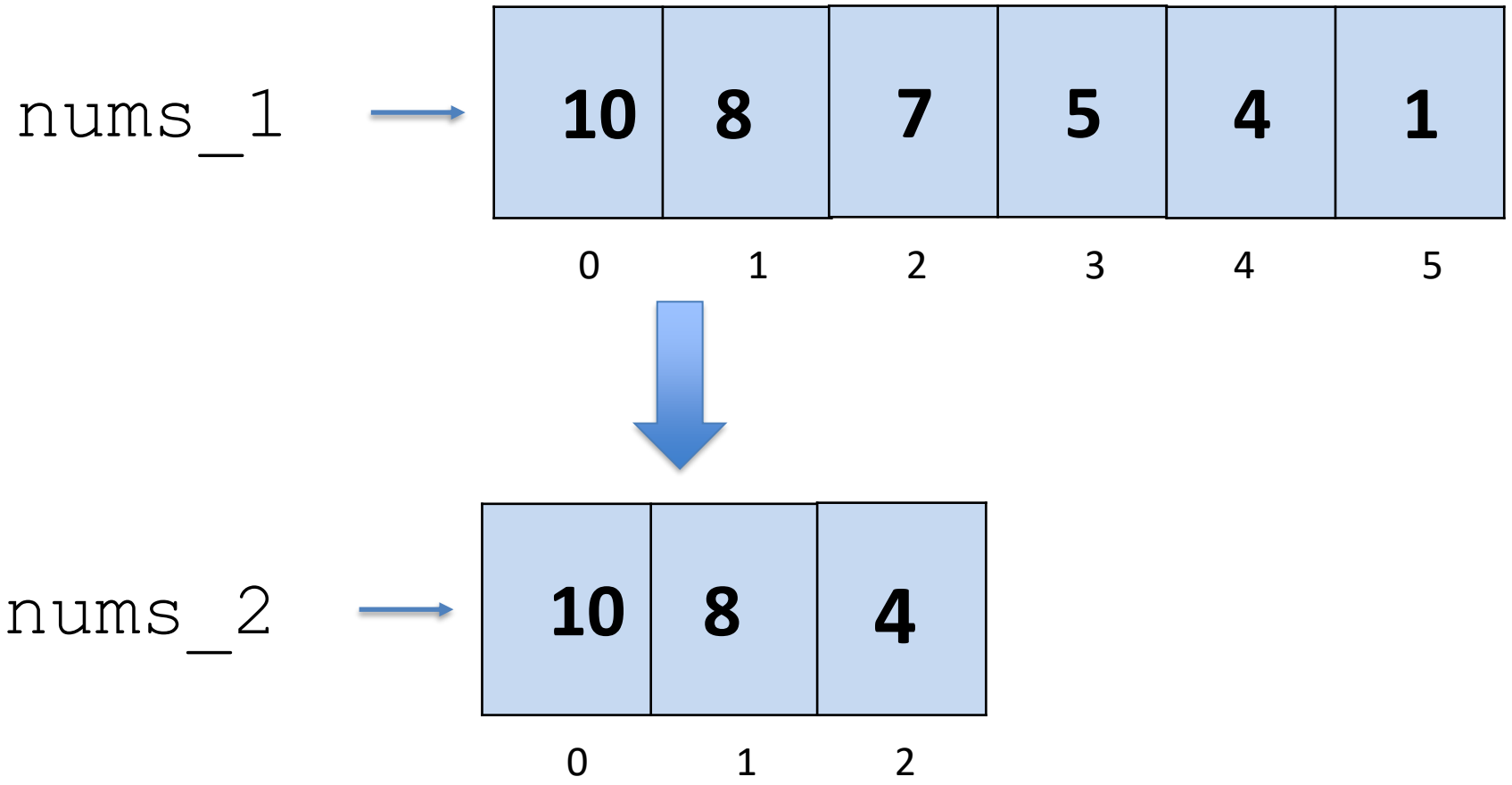
List Methods



Both nums_1 and nums_2 are pointing to the same list

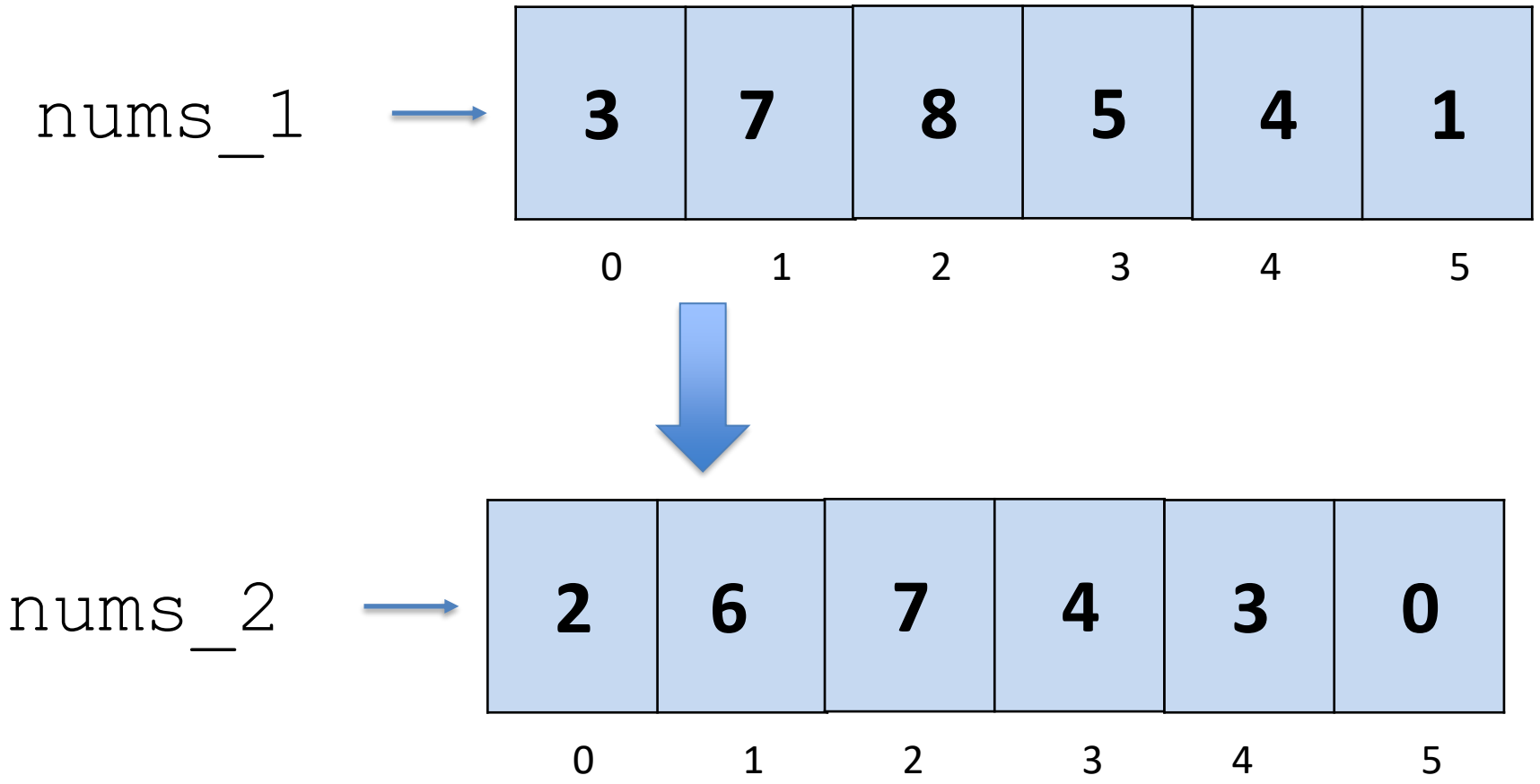
You have a TV remote 1 pointing to TV X.
Your sibling has a TV remote 2 pointing to the same TV X.

List Methods



todo

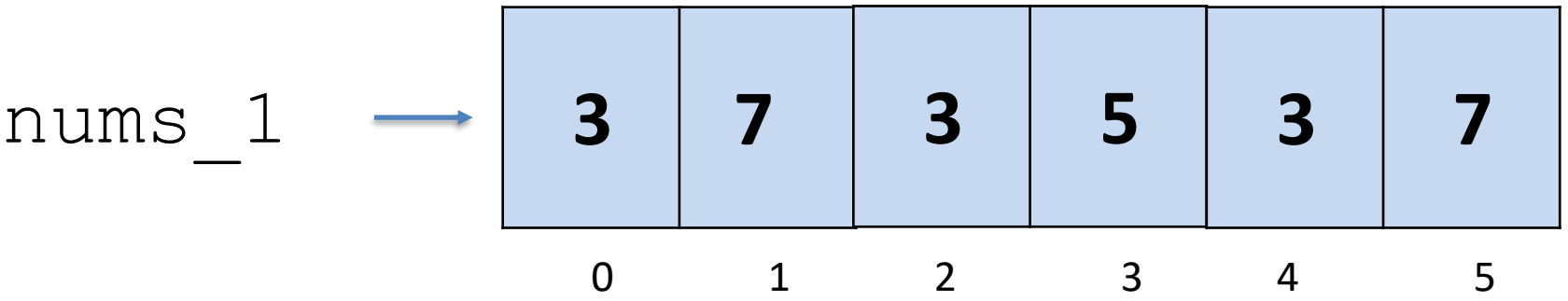
List Methods



Mapping an element to anther elements

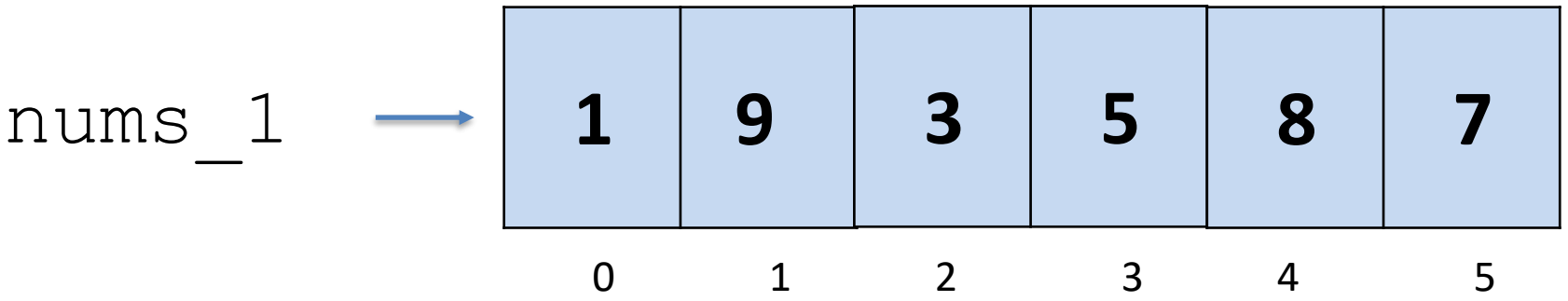
X2 = X1 - 1

List Methods



Count of 3 = 3
Count of 7 = 2
Count of 99 = 0

List Methods



What is the index of 5? = 3
What is the index of 7? = 5

What is the index of 99? = ERROR

Python’s Built-in Functions

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

Some functions are valid for lists.

I highlighted some.

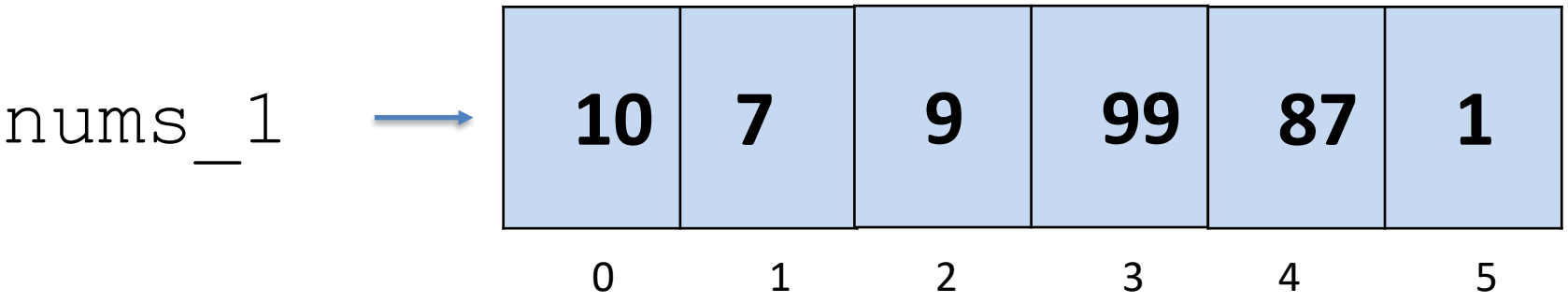
Can you find other functions that are valid on lists?

https://docs.python.org/3/library/functions.html

List Methods list.method_name(params)

Method	Purpose
append(x)	Add x to the end of the list
extend(list_x)	Add all items from list_x at the end of the list
insert(i,x)	Inserts an item at a given position. The first argument is the index of the element before which to insert. For example, a.insert(0, x) inserts at the front of the list.
remove(x)	Removes the first item x (note: there can be multiple items x in the list)
pop()	Removes the last item and returns the item
clear( )	Removed all elements in the list. Empties the list.
index(x)	Returns the index of the first item x.
count(x)	Counts the number of times x is appearing in the list
sort( )	Sorts the elements in ascending order. sort(reverse=True) sorts the elements in descending order
reverse( )	Reverses a list
copy( )	Returns a copy of the list. You can also use "list" built-in function for the same purpose.

Is this a built-in function? Or list method?



1. x = len(num_1)
2. y = num_1.pop( )
3. fun1(nums_1)
4. nums_1.fun2( )
5. fun2(nums_1)
6. nums_1.fun3( )

1. If you are calling a function using a list as a handle,
2. Then that is a list method.
3. Otherwise, it is a built-in function

Built-in Functions vs Methods for Lists

Built-in Functions: You pass in a “list” as an argument to the function.

Method: You call the method on a “list”.

For example, consider the list

```
>>> us_states = ["MN", "TX", "OH", "CA", "MA"]
```

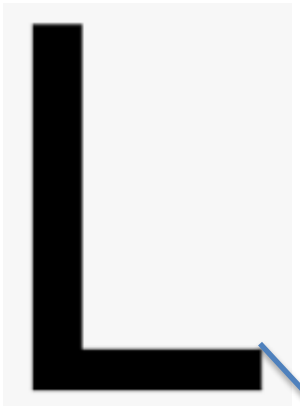
```
# Built-In Functions
>>> max(us_states)
'TX'
>>> min(us_states)
'CA'
>>> len(us_states)
5
>>> min(us_states)
'CA'
```

```
>>> us_states.reverse()
>>> print(us_states)
['MA', 'CA', 'OH', 'TX', 'MN']
>>> us_states.sort()
>>> print(us_states)
['CA', 'MA', 'MN', 'OH', 'TX’]
>>> us_states.append("FL")
>>> print(us_states)
['CA', 'MA', 'MN', 'OH', 'TX', 'FL']
```

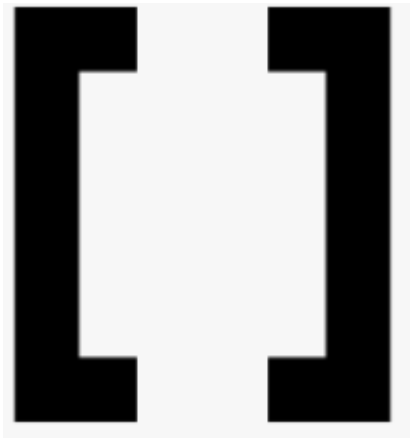
http://www.python-ds.com/python-3-list-methods

An example of a List

We use SQAURE Brackets to indicate a List.



[ ]
( )
{ }



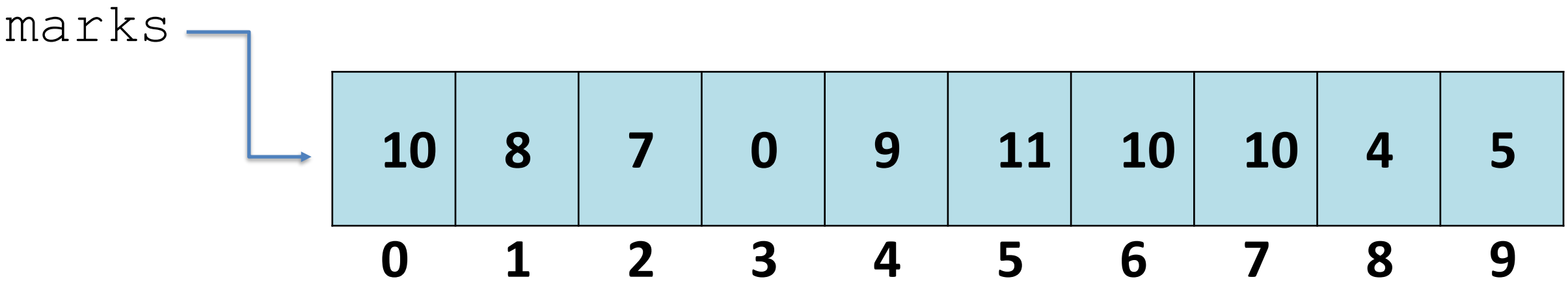
```
marks = [10, 8, 7,0,9,11,10,10,4,5]
```

You can visualize the list as follows:

marks

10	8	7	0	9	11	10	10	4	5
0	1	2	3	4	5	6	7	8	9

Integer List



How many students are there in the class?

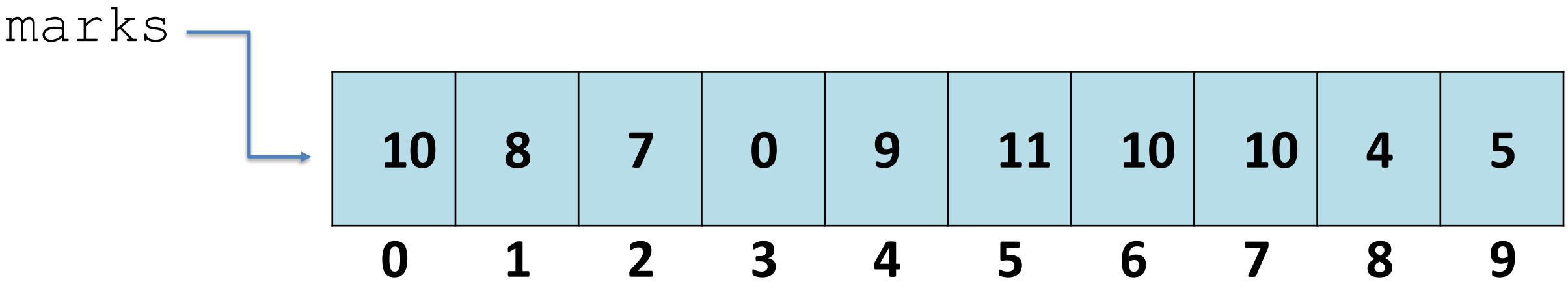
What is the total marks earned by all the students together?

What is the highest score?

What is the lowest score?

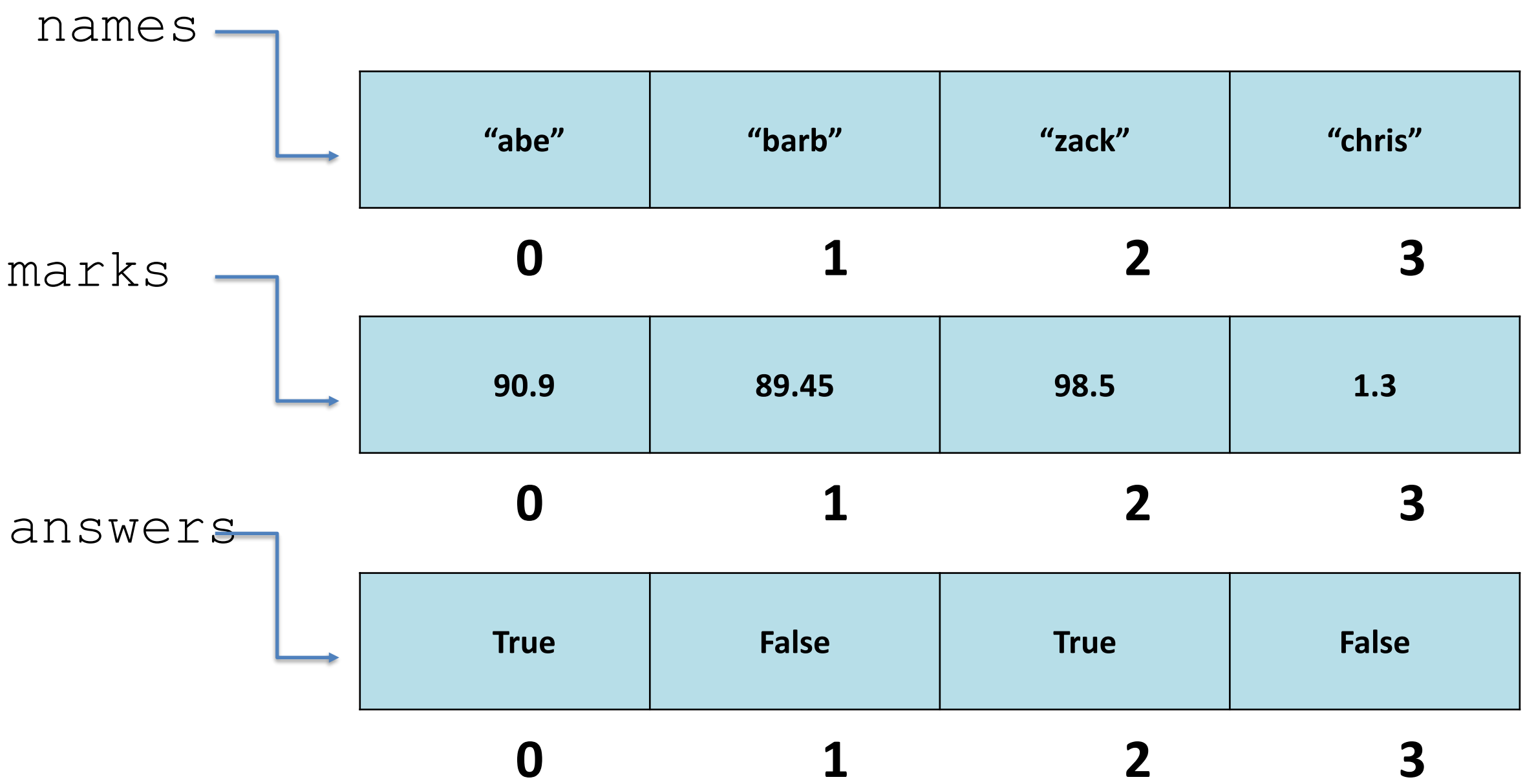
What is the average score? sum(marks) / len(marks)

Integer List



How many students are there in the class?
What is the total marks earned by all the students together?
What is the highest score?
What is the lowest score?
What is the average score?
We can find answers to these questions using python's built-in functions. See the code snippet here.

List of (integers, strings, bool, floats)



List of (different types of data)

student1

"Josh"	24	"josh@gmail.com"	"CS Major"
0	1	2	3

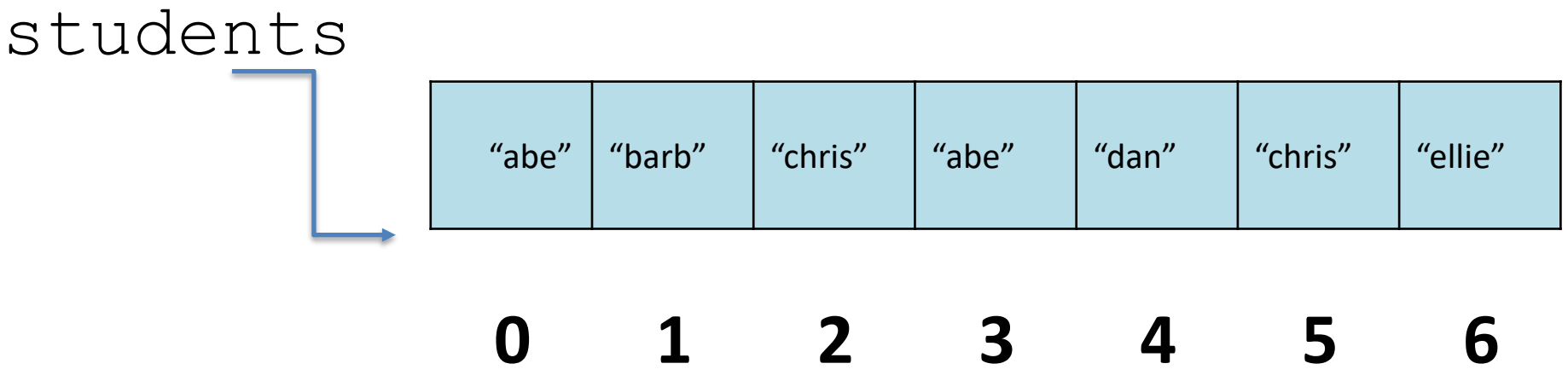
student2

"Barb"	30	"barb@yahoo.com"	"IT Major"
0	1	2	3

student3

"Christy"	24	"Christy@aol.com"	"Data Science"
0	1	2	3

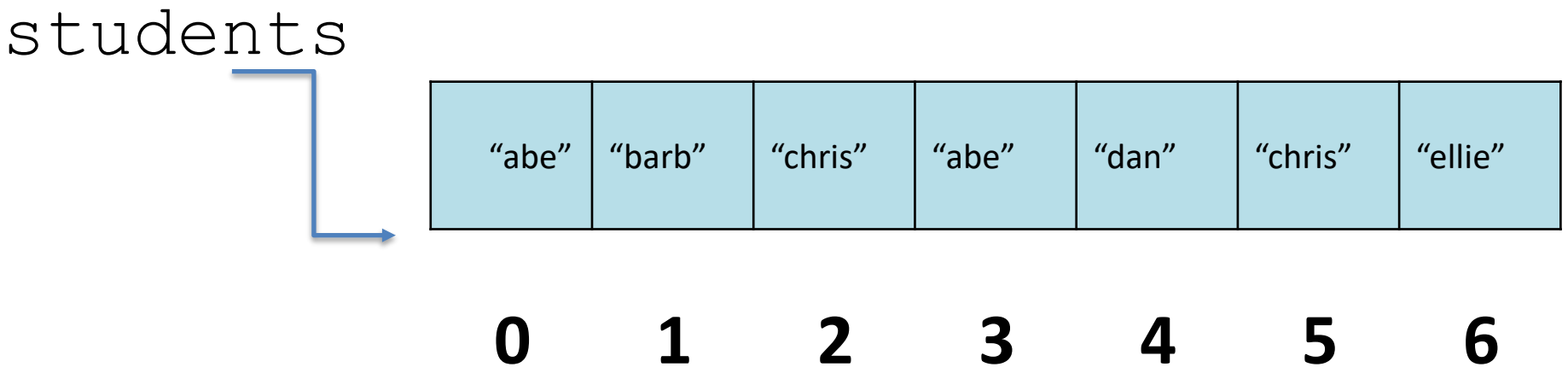
Creating a list




```
#Creating an empty list
students = [ ]

# list of students
students = ["abe", "barb", "chris", "abe", "dan", "chris", "ellie"]
```

Accessing an element of a list

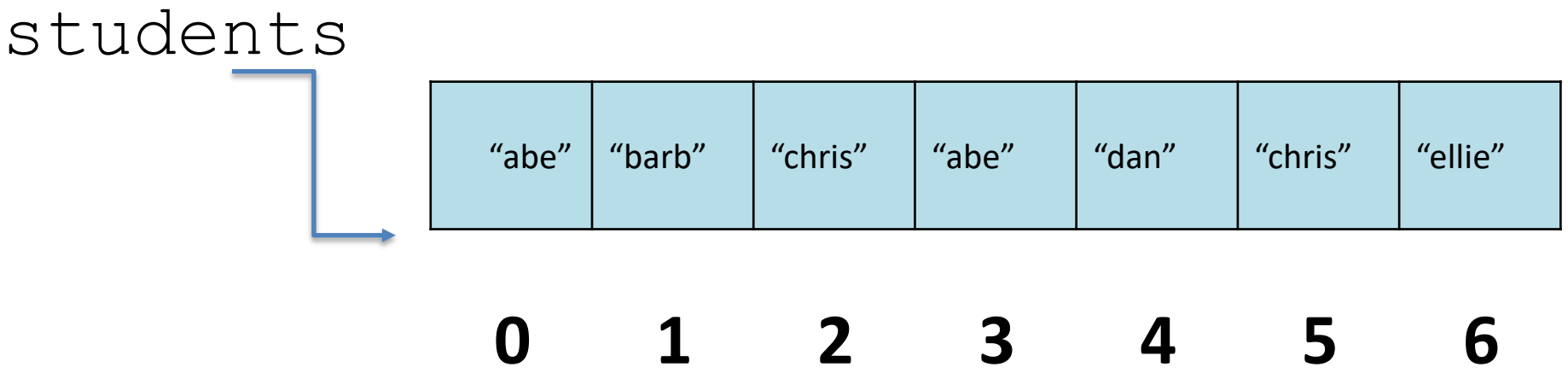


We use subscript notation to access an element.

The first element sits at the index 0
The second element sits at the index 1

student_1 = students[0]
student_5 = students[6]

How to update/change a value?

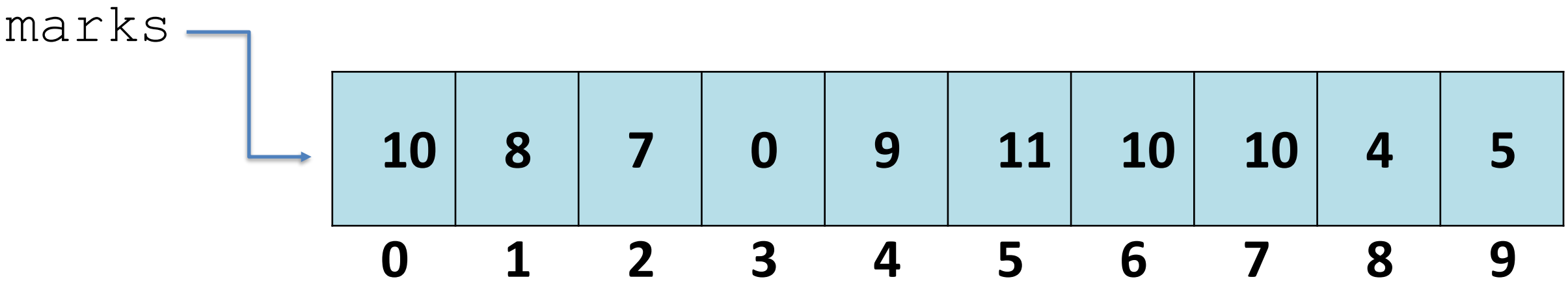


Use the subscript notation to access the element.
And assign the new value.

For example, how do we change the name of 4th student from "abe" to "abraham"?

students[3] = "abraham"

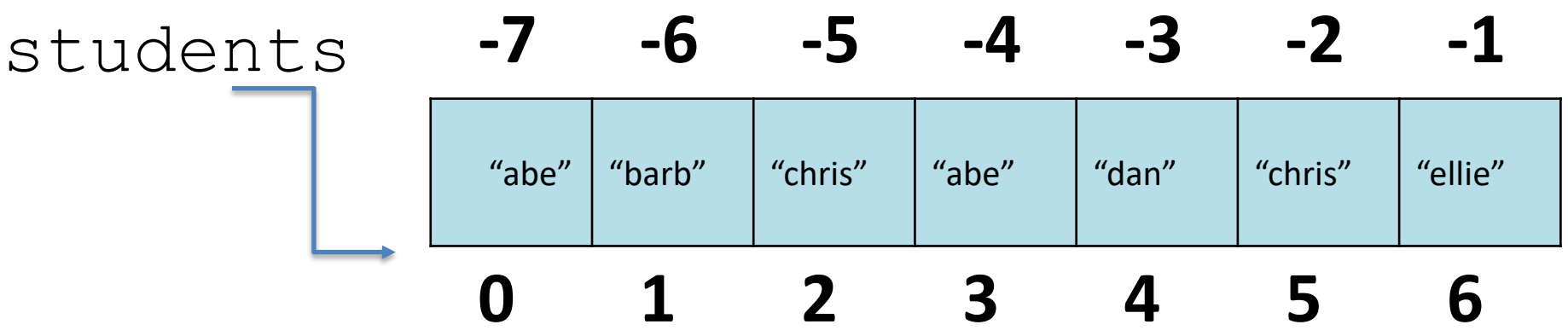
No empty spaces, please!



Lists are linear, sequential with no empty spaces in between.

If you want to keep track of empty spaces (for examples, blanks in a hangman game), you can use a special character (like _ or *) to indicate a blank space.

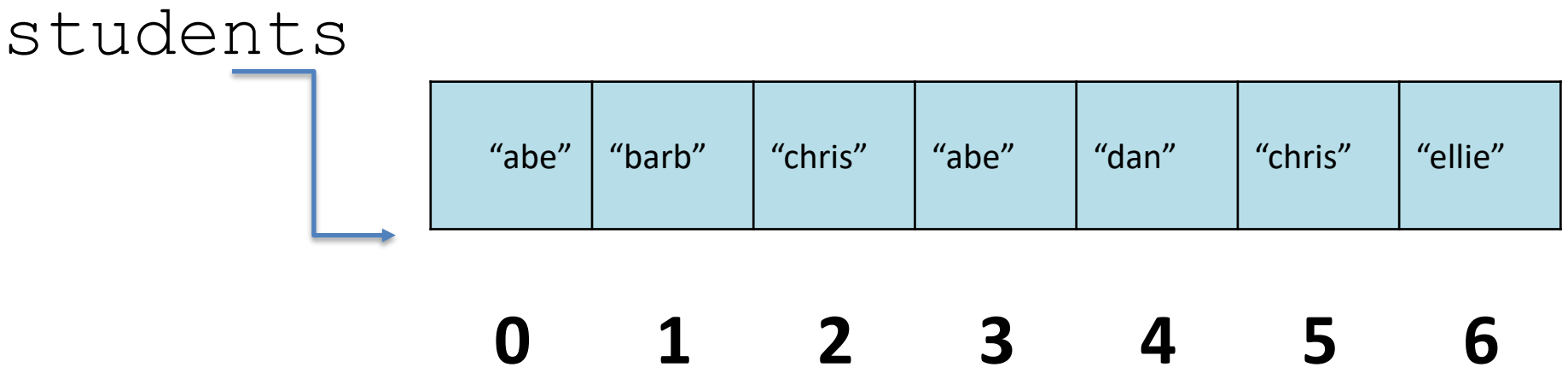
Negative indexes are pretty cool!



- Negative indexing means beginning from the end.
- -1 refers to the last item, -2 refers to the second last item etc.

students[-1] → ellie
students[-3]

Iterating the list (just for reading)



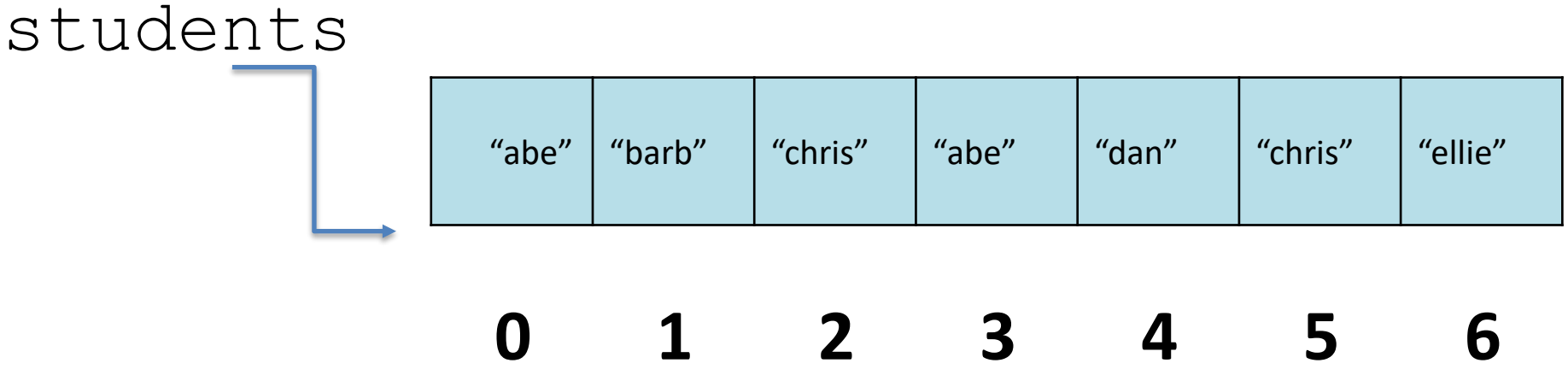
Iteration = Loop
Iterating the list = Traversing the list = Visiting each and every element of the liste fi.

For example, how can we print the first three characters of each element in the students list?

```
for elem in students:
    x = elem

print(x)
```

Iterating the list (for accessing the index / updating)

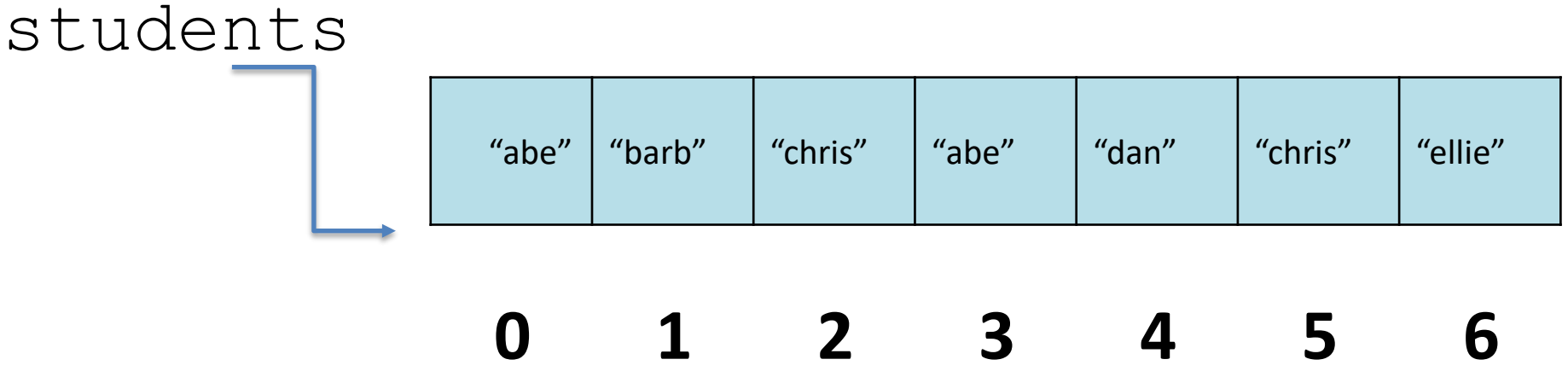


Iteration = Loop
Iterating the list = Traversing the list = Visiting each and every element of the liste fi.

For example, how can we print the first three characters of each element in the students list?

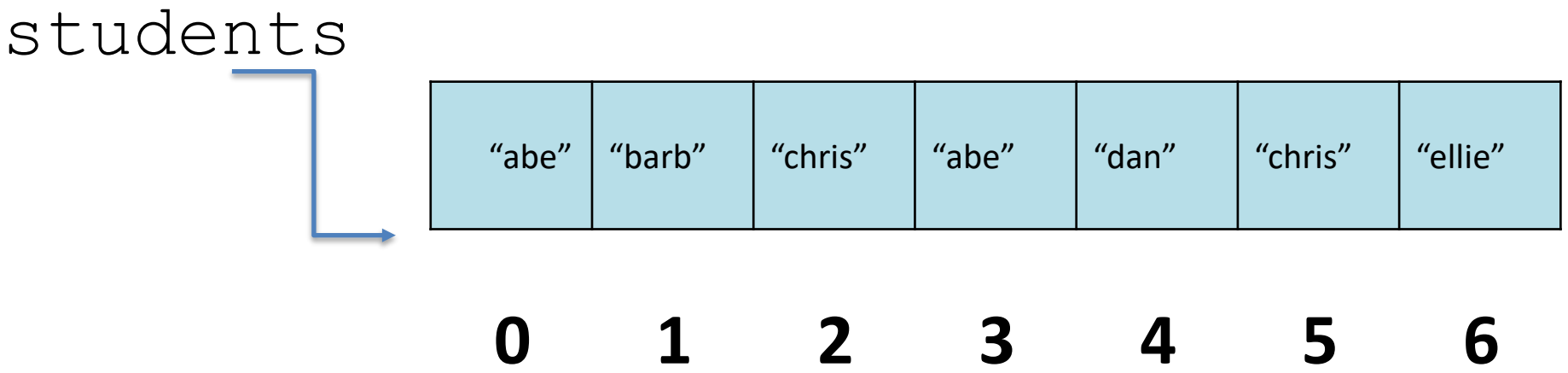
```
for index in range(len(students)):
    print(students[index])
```

Iterating the list (getting both index and value)




```
for index, value in enumerate(students):
    print(index, '→', value)
```

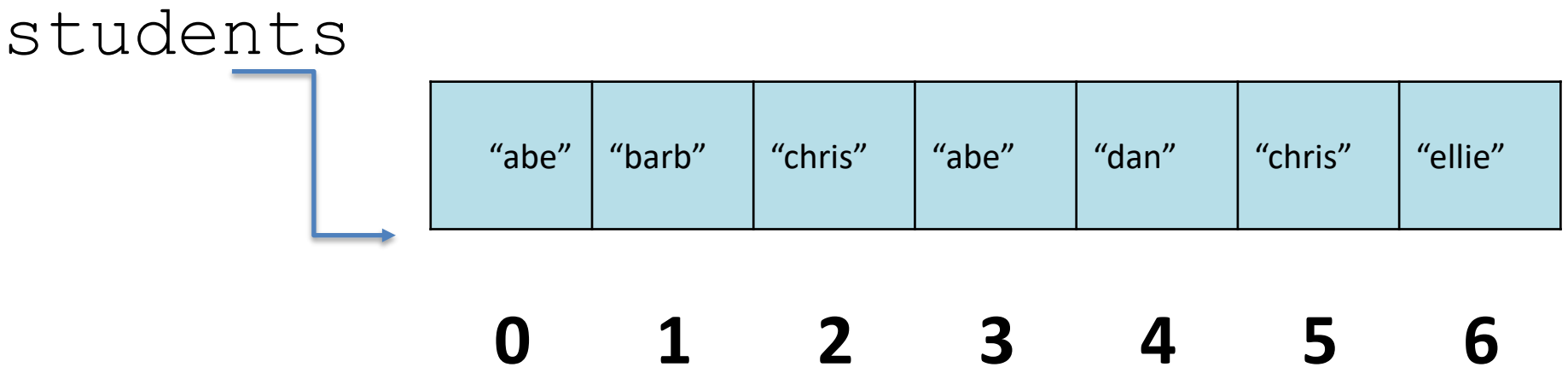
Iterating the list (Contd.)



How can we print the first three characters of each element in the students list?

```
for x in students:
    print(x[0:2])
```

Checking for the memberships? **in**



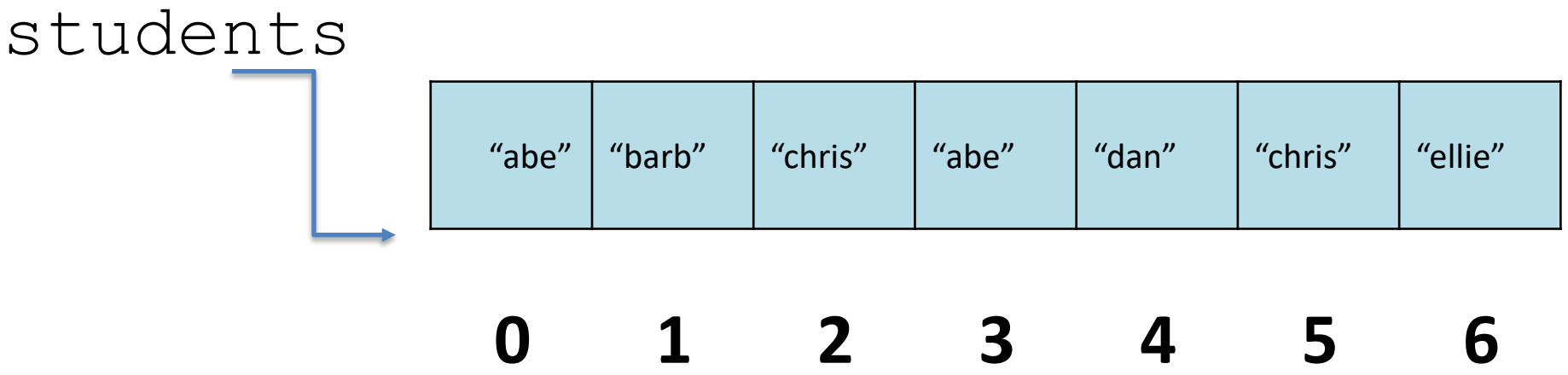
Is “barb” in the list?

"barb" **in** students

```
>>> if ("barb" in students):
        print("Yes! Barb is registered")


Yes! Barb is registered
```

Deleting vs Clearing the list



students.clear( ) → Empties the list

del students → deletes the entire list.

List Methods list.method_name(params)

Method	Purpose
append(x)	Add x to the end of the list
extend(list_x)	Add all items from list_x at the end of the list
insert(i,x)	Inserts an item at a given position. The first argument is the index of the element before which to insert. For example, a.insert(0, x) inserts at the front of the list.
remove(x)	Removes the first item x (note: there can be multiple items x in the list)
pop()	Removes the last item and returns the item
pop([i]	Removes the first item
clear( )	Removed all elements in the list. Empties the list.
index(x)	Returns the index of the first item x.
count(x)	Counts the number of times x is appearing in the list
sort( )	Sorts the elements in ascending order. sort(reverse=True) sorts the elements in descending order
reverse( )	Reverses a list
copy( )	Returns a copy of the list. You can also use "list" built-in function for the same purpose.

https://www.w3schools.com/python/python_ref_list.asp

Lists: Summary

List offers a simple data collection.

Lists are flexible and can be used in many problem-solving scenarios.