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

Map, Filter and Reduce

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 are everywhere

Outline

1. Recap:
    1. What are lists?
    2. List methods vs Python built-in functions
    3. Python 101 Built-in Functions colab notebook
2. Map – Filter – Reduce
3. Traversal of the lists
    1. Get the Values
    2. Get the Index
    3. Get both the (Index, Value)
4. Lists and Strings
5. When to use?
    1. Built-in Function
    2. List Traversal

Python Built-in Functions

See python101_built_in_functions.ipynb

List doesn't own these functions.
Tuple doesn't own these functions
Set doesn't own these functions.
Dictionary doesn't own these functions.

No one owns these.

These methods will work on any item / data structure / dat type.

Built-in Functions

A	E	L	R
abs()	enumerate()	len()	range()
aiter()	eval()	list()	repr()
all()	exec()	locals()	reversed()
any()			round()
anext()	F	M	
ascii()	filter()	map()	S
	float()	max()	set()
B	format()	memoryview()	setattr()
bin()	frozenset()	min()	slice()
bool()			sorted()
breakpoint()	G	N	staticmethod()
bytearray()	getattr()	next()	str()
bytes()	globals()		sum()
		O	super()
C	H	object()	
callable()	hasattr()	oct()	T
chr()	hash()	open()	tuple()
classmethod()	help()	ord()	type()
compile()	hex()		
complex()		P	V
	I	pow()	vars()
D	id()	print()	
delattr()	input()	property()	Z
dict()	int()		zip()
dir()	isinstance()		
divmod()	issubclass()		_
	iter()		__import__()

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

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

Built-in Functions vs List Methods

```
# List Methods
nums = [1, 2]        # [1, 2]
nums.append(3)       # [1, 2, 3]
nums.insert(0, 10)   # [10, 1, 2, 3]
nums.insert(0, 9)    # [9, 10, 1, 2, 3]
nums.insert(1, 8)    # [9, 8, 10, 1, 2, 3]
nums.reverse()       # [3, 2, 1, 10, 8, 9]
nums.sort()          # [1, 2, 3, 8, 9, 10]
nums.pop(1)          # [1, 3, 8, 9, 10]
nums.pop()           # [1, 3, 8, 9]
nums.pop(1)          # [1, 8, 9]
nums.append(3)       # [1, 8, 9, 3]
nums.remove(1)       # [8, 9, 3]
nums.extend([3,4])   # [8, 9, 3, 3, 4]
print(nums)
nums.clear()
print(nums)
nums.append(1)
nums.append(1)
nums.append(2)
print(nums)   # [1, 1, 2]


# Built-in functions
a = min(nums)
b = max(nums)
c = len(nums)
d = sum(nums)
```

Python tutor link is here.

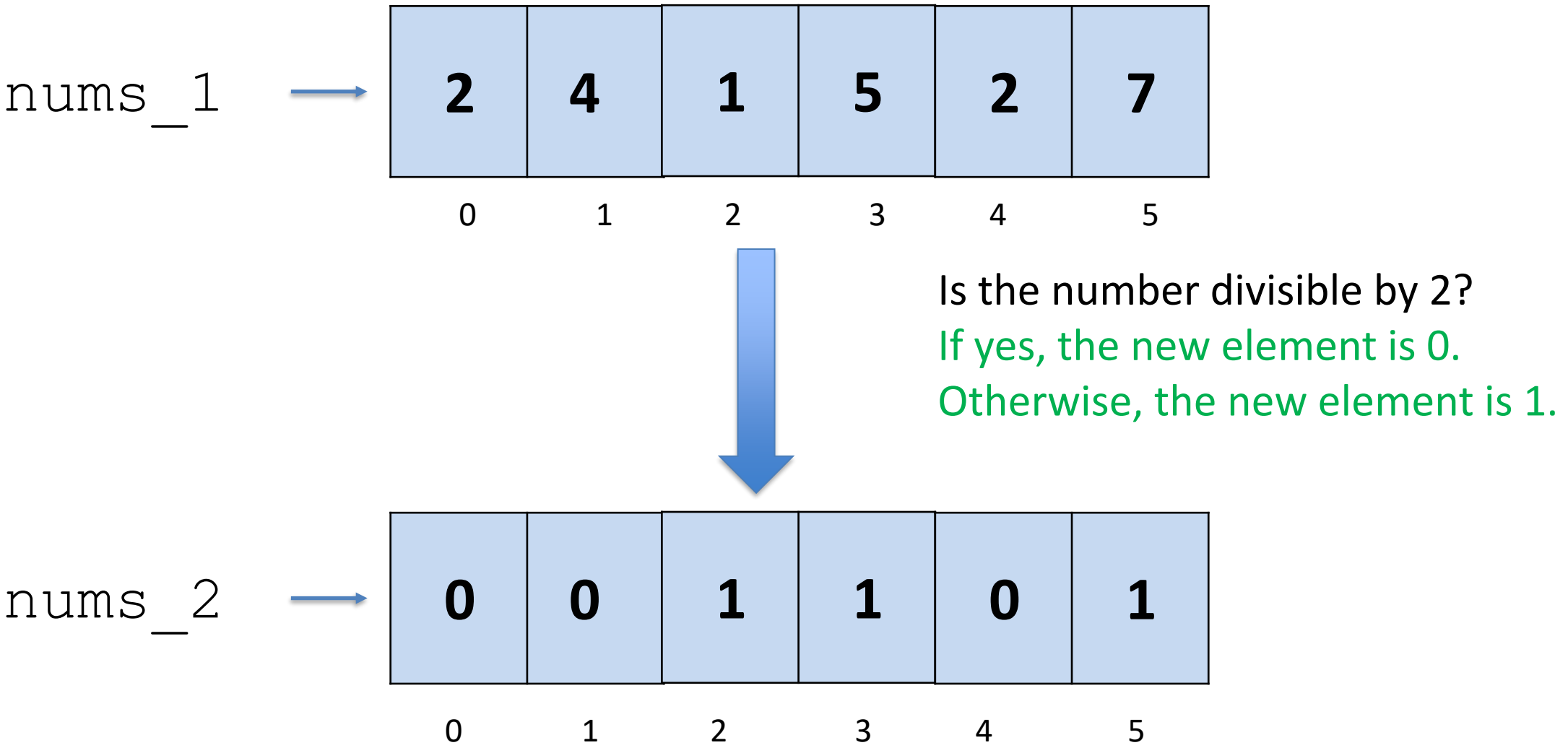
Map, Filter and Reduce

Map: Mapping the elements of a list to create a new list. The length of the lists must be the same.

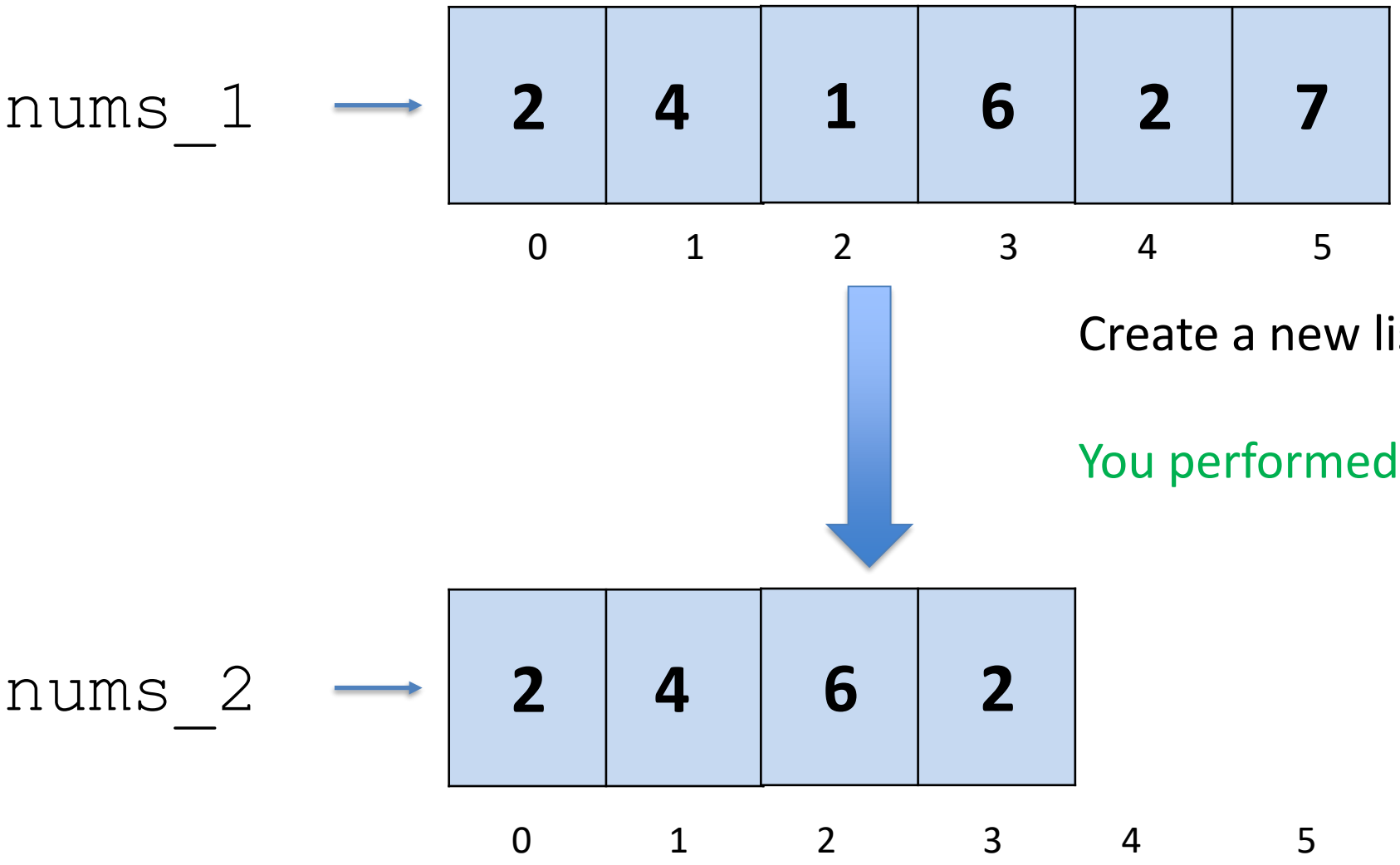
Filter: Creating a new list (which is shorter than the original list). Eg: Given a list, create a new list with even numbers.

Reduce: Given a list, calculate something. Find a single value. Derive a scalar value. Examples: len(), min(), max(), sum(), sum()/len(), count of a value, index / position of a value

Map Methods


nums_1
2
4
1
5
2
7
0
1
2
3
4
5
Is the number divisible by 2?
If yes, the new element is 0.
Otherwise, the new element is 1.
nums_2
0
0
1
1
0
1
0
1
2
3
4
5

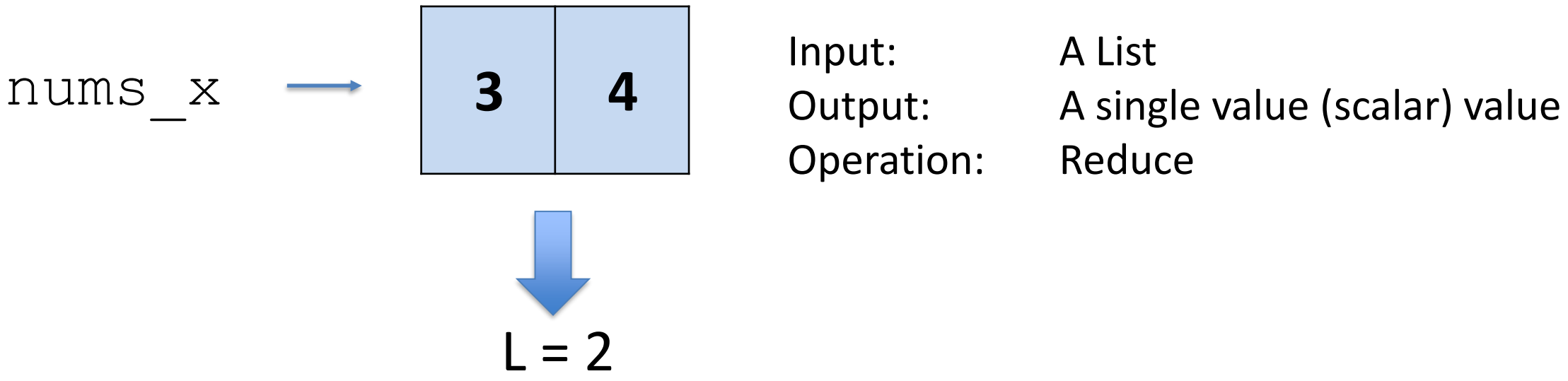
Filter Methods



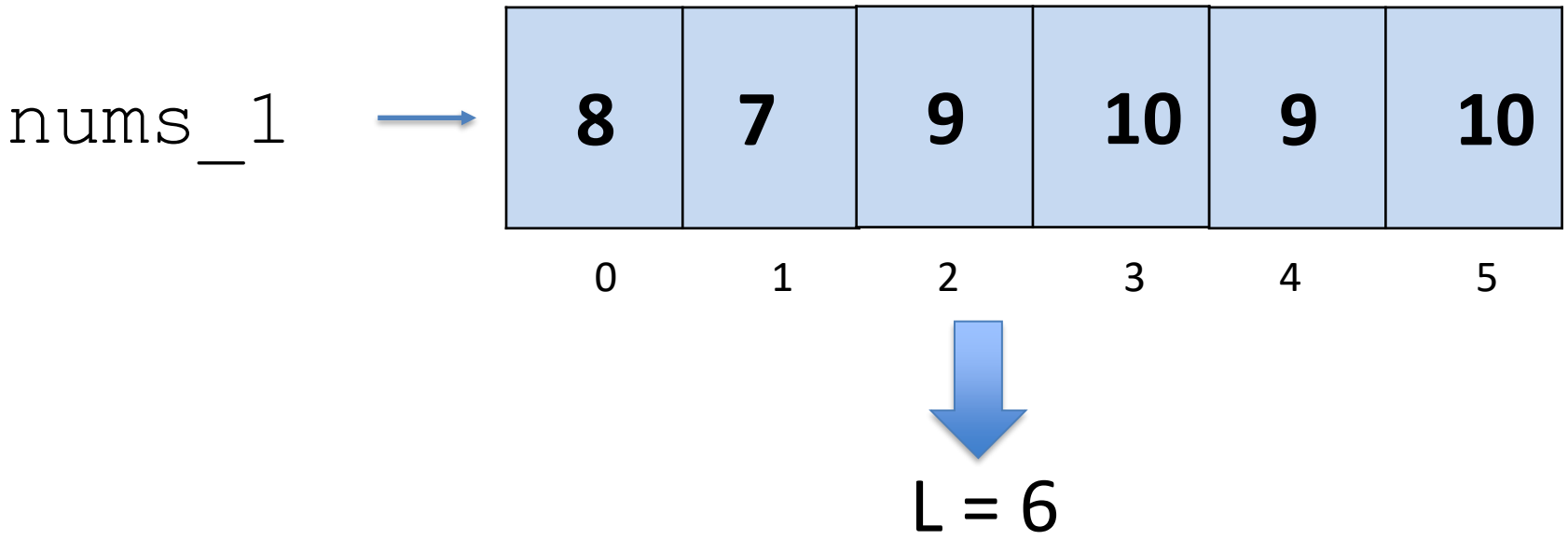
Create a new list with only EVEN numbers.

You performed “Filtering” operation.

“Reduce” operation on List



Input: A List
Output: A single value (scalar) value
Operation: Reduce



List Traversal – 1. Get the Value

```
#(Method 1) print every element in the list
# I don't care about index
# I care about only values
for elem in nums:
    print(elem)

print(elem)
```

```
nums = [1, 2, 3]

# Crating a Map: Creating a new list
new_nums = []
for elem in nums:
    new_nums.append(elem+10)

print("original: ", nums)
print("New: ", new_nums)
```

```
#(Method 1) print every element in the list
# I don't care about index
# I care about only values
nums = [4, 5, 1, 2]

total = 0
for elem in nums:
    total = total + elem

print("nums list: ", nums)
print("total : ", total)


# Best Practice: Use the built-in function
print("sum (build-in): ", sum(nums))
```

List Traversal – 2. Get the index (through range)

```
1  # Loop through range. Update the values (mapping)
2  nums = [3, 6, 2]
3
4  print(nums)
5
6  for index in range(len(nums)):
7      nums[index] = nums[index] + 10
8
9  # after the for loop
10 print(nums)
```

List Traversal – 3. Get the index, and value

```
15  # method 3:
16  print("printing both index and value through enumerate")
17  for index, value in enumerate(nums):
18      print(index, " --", value)
```

List Traversals – Summary (pythontutor link)

```
1  nums = [3, 5, 4, 9]
2
3  # method 1:
4  print("Printing only vlaues")
5  for elem in nums:
6      print(elem)
7
8  # method 2:
9  print("printing both index and value")
10 for index in range(len(nums)):
11     print(index, " --", nums[index])
12
13 # method 3:
14 print("printing both index and value through enumerate")
15 for index, value in enumerate(nums):
16     print(index, " --", value)
```

Lists and Strings: pythontutor link

```python
course = "python"
# Imagine this as a list
# ['p', 'y', 't', 'h', 'o', 'n']
for elem in course:
    print(elem)


print("word length:", len(course))
print("min of word:", min(course))
print("max of word:", max(course))
```

```python
# another example
names = ['zack', 'peter', 'anna', 'bindu', 'christy', 'diana']
print("list length:", len(course))
print("min of list:", min(course))
print("max of list:", max(course))
```

Lists: Map, Filter, Reduce

List is an ordered, indexed collection that can have duplicates.

All the list operations we perform can be visualized as follows.

Map: Create a new list based on a given list. Both the lists will have the same length

Filter: Create a new list by filtering out some elements of a given list. New list will be smaller than the given list.

Reduce: Derive one value (eg: max, min, len, sum, average, etc.) from a given list.