



9.10. The Accumulator Pattern with Lists

We can accumulate values into a list rather than accumulating a single numeric value. Consider, for example, the following program which transforms a list into a new list by squaring each of the values.

[Save & Run](#)

Original - 1 of 1

[Show in CodeLens](#)

```
1 nums = [3, 5, 8]
2 accum = []
3 for w in nums:
4     x = w**2
5     accum.append(x)
6 print(accum)
7
```

```
[9, 25, 64]
```

Activity: 1 -- ActiveCode (ac8_9_1)

Here, we **initialize** the accumulator variable to be an empty list, on line 2.

We **iterate** through the sequence (line 3). On each iteration we transform the item by squaring it (line 4).

The **update** step appends the new item to the list which is stored in the accumulator variable (line 5). The update happens using the `.append()`, which mutates the list rather than using a reassignment. Instead, we could have written `accum = accum + [x]`, or `accum += [x]`. In either case, we'd need to concatenate a list containing `x`, not just `x` itself.

At the end, we have accumulated a new list of the same length as the original, but with each item transformed into a new item. This is called a mapping operation, and we will revisit it in a later chapter.

Note how this differs from mutating the original list, as you saw in a previous section.

Check your understanding

seqmut-9-1: What is printed by the following statements?

```
alist = [4,2,8,6,5]
blist = [ ]
for item in alist:
    blist.append(item+5)
print(blist)
```

- ☐ A. [4,2,8,6,5]
- ☐ B. [4,2,8,6,5,5]
- ☒ C. [9,7,13,11,10]
- ☐ D. Error, you cannot concatenate inside an append.

[Check me](#)[Compare me](#)

✔ Yes, the for loop processes each item of the list. 5 is added before it is appended to blist.

Activity: 2 -- Multiple Choice (question8_9_1)

seqmut-9-2: What is printed by the following statements?

```
lst= [3,0,9,4,1,7]
new_list=[]
for i in range(len(lst)):
    new_list.append(lst[i]+5)
print(new_list)
```

- ☐ A. [8,5,14,9,6]
- ☒ B. [8,5,14,9,6,12]
- ☐ C. [3,0,9,4,1,7,5]
- ☐ D. Error, you cannot concatenate inside an append.

[Check me](#)[Compare me](#)

✔ Yes, the for loop processes each item in lst. 5 is added before lst[i] is appended to blist.

Activity: 3 -- Multiple Choice (question8_9_2)

2. For each word in the list `verbs`, add an -ing ending. Save this new list in a new list, `ing`.

[Save & Run](#)

5/13/2021, 4:25:13 PM - 2 of 2

[Show in CodeLens](#)

```

1 verbs = ["kayak", "cry", "walk", "eat", "drink", "fly"]
2 ing = []
3 for item in verbs:
4     ing.append(item + "ing")
5 print(ing)
6

```

```
['kayaking', 'crying', 'walking', 'eating', 'drinking', 'flying']
```

Activity: 4 -- ActiveCode (ac8_9_2)

Result	Actual Value	Expected Value	Notes
Pass	['kay...ing']	['kay...ing']	Testing that the variable ing has the correct value.

Expand Differences

You passed: 100.0% of the tests

Given the list of numbers, `nums`, create a new list of those same numbers increased by 5. Save this new list to the variable `newlist`.

Save & Run

5/13/2021, 4:27:32 PM - 2 of 2

Show in CodeLens

```

1 nums = [5, 10, 15, 20, 25]
2 newlist = []
3 for i in range(len(nums)):
4     newlist.append(nums[i]+5)
5 print(newlist)
6

```

```
[10, 15, 20, 25, 30]
```

Activity: 5 -- ActiveCode (ac8_9_3)

Result	Actual Value	Expected Value	Notes
Pass	[10, ..., 30]	[10, ..., 30]	Testing that the newlist value contains the correct elements.

Expand Differences

You passed: 100.0% of the tests

Challenge Now do the same as in the previous problem, but do not create a new list. Overwrite the list `nums` so that each of the original numbers are increased by 5.

Save & Run

5/13/2021, 4:30:31 PM - 4 of 4

Show in CodeLens

```

1 nums = [5, 10, 15, 20, 25]
2 for i in range(len(nums)):
3     nums[i]=nums[i]+5
4 print(nums)

```

```
[10, 15, 20, 25, 30]
```

Activity: 6 -- ActiveCode (ac8_9_4)

Result	Actual Value	Expected Value	Notes
Pass	[10, ..., 30]	[10, ..., 30]	Testing that nums is assigned to correct values.

Expand Differences

You passed: 100.0% of the tests

For each number in `lst_nums` , multiply that number by 2 and append it to a new list called `larger_nums` .

Save & Run

5/13/2021, 4:32:49 PM - 2 of 2

Show in CodeLens

```
1 lst_nums = [4, 29, 5.3, 10, 2, 1817, 1967, 9, 31.32]
2 larger_nums = []
3 for i in range(len(lst_nums)):
4     larger_nums.append(lst_nums[i]*2)
5 print (larger_nums)
6
```

```
[8, 58, 10.6, 20, 4, 3634, 3934, 18, 62.64]
```

Activity: 7 -- ActiveCode (ac8_9_5)

Result	Actual Value	Expected Value	Notes
Pass	[8, 5...2.64]	[8, 5...2.64]	Testing that larger_nums has been created correctly.

Expand Differences

You passed: 100.0% of the tests

You have attempted 8 of 7 activities on this page

9.9. Non-mutating Methods on Strings">

9.11. The Accumulator Pattern with Strings">

9.9. Non-mutating Methods on Strings">

9.11. The Accumulator Pattern with Strings">Next Section - 9.11. The Accumulator Pattern with Strings

© Copyright 2017 bradleymiller. Created using Runestone 4.1.17.

[Back to top](#)