9.13. Don’t Mutate A List That You Are Iterating Through

So far we’ve shown you how to iterate through a list:

Save & Run Original - 1 of 1 Show in CodeLens

```
colors = ["Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet"]

for color in colors:
    print(color)

```

```
Red
Orange
Yellow
Green
Blue
Indigo
Violet
```

Activity: 1 -- ActiveCode (ac8_12_1)

As well as accumulate a list by appending or deleting items!

Save & Run Original - 1 of 1 Show in CodeLens

```
colors = ["Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet"]
initials = []

for color in colors:
    initials.append(color[0])

print(initials)

```

```
['R', 'O', 'Y', 'G', 'B', 'I', 'V']
```

Activity: 2 -- ActiveCode (ac8_12_2)

You may be tempted now to iterate through a list and accumulate some data into it or delete data from it, however that often becomes very confusing. In the following code we will filter out all words that begin with P, B, or T.

Save & Run Original - 1 of 1 Show in CodeLens

```
colors = ["Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet", "Purple",

for position in range(len(colors)):
    color = colors[position]
    print(color)
    if color[0] in ["P", "B", "T"]:
        del colors[position]

print(colors)

```

```
Red
Orange
Yellow
Green
Blue
```

```
Violet
Purple
Brown
Turquois
Beige
```

Activity: 3 -- ActiveCode (ac8_12_3)

Error

```
IndexError: list index out of range on line 4
```

Description

This message means that you are trying to index past the end of a string or a list. For example if your list has 3 things in it and you try to access the item at position 3 or more.

To Fix

Remember that the first item in a list or string is at index position 0, quite often this message comes about because you are off by one. Remember in a list of length 3 the last legal index is 2

In the code above, we iterated through `range(len(colors))` because it made it easier to locate the position of the item in the list and delete it. However, we run into a problem because as we delete content from the list, the list becomes shorter. Not only do we have an issue indexing on line 4 after a certain point, but we also skip over some strings because they've been moved around. To see this more easily, try walking through this code in codelens. Note that each time we iterate through the list python does not reevaluate the iterator variable.

We can also try to accumulate a list that we're iterating through as well. What do you think will happen here?

Save & Run   Original - 1 of 1   Show in CodeLens

```
colors = ["Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet"]

for color in colors:
    if color[0] in ["A", "E", "I", "O", "U"]:
        colors.append(color)

print(colors)

```

```
['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet', 'Orange', 'Indigo']
```

Activity: 4 -- ActiveCode (ac8_12_4)

Though there is not an error, the behavior may not be expected. When we come across a color that begins with a vowel, that color is added to the end of the list. Again, because Python does not reevaluate the iterator variable we are not stuck adding colors that start with vowels for an infinite number of times. That's good in this case! Ultimately though, it can be confusing to write code like this. We recommend not iterating over a list that you will be mutating within the for loop.

You have attempted 5 of 4 activities on this page

9.12. Accumulator Pattern Strategies">

9.14. Summary">

9.12. Accumulator Pattern Strategies">

✔ Completed. Well Done!

9.14. Summary">Next Section - 9.14. Summary

| Back to top