You were Last Reading: Transforming Sequences > Don't Mutate A List That You Are Iterating Through Continue Reading

6.8. Count and Index

As you create more complex programs, you will find that some tasks are commonly done. Python has some built-in functions and methods to help you with these tasks. This page will cover two helpful methods for both strings and lists: count and index.

You've learned about methods before when drawing with the turtle module. There, you used `.forward(50)` and `.color("purple")` to complete actions. We refer to forward and color as methods of the turtle class. Objects like strings and lists also have methods that we can use.

6.8.1. Count

The first method we'll talk about is called `count`. It requires that you provide one argument, which is what you would like to count. The method then returns the number of times that the argument occured in the string/list the method was used on. There are some differences between count for strings and count for lists. When you use count on a string, the argument can only be a string. You can't count how many times the integer 2 appears in a string, though you can count how many times the string "2" appears in a string. For lists, the argument is not restricted to just strings.

Save & Run 4/30/2021, 7:47:19 PM - 2 of 2 Show in CodeLens

```python
a = "I have had an apple on my desk before!"
print(a.count("e"))
print(a.count("ha"))

```

```
5
2
```

Activity: 1 -- ActiveCode (ac5_8_1)

The activecode window above demonstrates the use of count on a string. Just like with the turtle module when we had to specify which turtle was changing color or moving, we have to specify which string we are using count on.

Save & Run 4/30/2021, 7:47:22 PM - 2 of 2 Show in CodeLens

```python
z = ['atoms', 4, 'neutron', 6, 'proton', 4, 'electron', 4, 'electron', 'atoms']
print(z.count("4"))
print(z.count(4))
print(z.count("a"))
print(z.count("electron"))

```

```
0
3
0
2
```

Activity: 2 -- ActiveCode (ac5_8_2)

When you run the activecode window above, you'll see how count with a list works. Notice how "4" has a count of zero but 4 has a count of three? This is because the list `z` only contains the integer 4. There are never any strings that are 4. Additionally, when we check the count of "a", we see that the program returns zero. Though some of the words in the list contain the letter "a", the program is looking for items in the list that are *just* the letter "a".

6.8.2. Index

The other method that can be helpful for both strings and lists is the `index` method. The `index` method requires one argument, and, like the `count` method, it takes only strings when index is used on strings, and any type when it is used on lists. For both strings and lists, `index` returns the leftmost index where the argument is found. If it is unable to find the argument in the string or list, then an error will occur.

Save & Run 4/30/2021, 7:47:24 PM - 2 of 2 Show in CodeLens

```
music = "Pull out your music and dancing can begin"
bio = ["Metatarsal", "Metatarsal", "Fibula", [], "Tibia", "Tibia", 43, "Femur", "Oc

print(music.index("m"))
print(music.index("your"))

print(bio.index("Metatarsal"))
print(bio.index([]))
print(bio.index(43))

```

```
14
9
0
3
6
```

Activity: 3 -- ActiveCode (ac5_8_3)

All of the above examples work, but were you surprised by any of the return values? Remember that `index` will return the left most index of the argument. Even though “Metatarsal” occurs many times in `bio`, the method will only return the location of one of them.

Here’s another example.

Save & Run 4/30/2021, 7:47:26 PM - 2 of 2 Show in CodeLens

```
seasons = ["winter", "spring", "summer", "fall"]

print(seasons.index("autumn"))  #Error!

```

Activity: 4 -- ActiveCode (ac5_8_4)

Error

```
ValueError: list.index(x): x not in list on line 3
```

Description

A ValueError most often occurs when you pass a parameter to a function and the function is expecting one type and you pass another.

To Fix

The error message gives you a pretty good hint about the name of the function as well as the value that is incorrect. Look at the error message closely and then trace back to the variable containing the problematic value.

In the activecode window above, we’re trying to see where “autumn” is in the list seasons. However, there is no string called autumn (though there is string called “fall” which is likely what the program is looking for). Remember that an error occurs if the argument is not in the string or list.

Check your understanding

sequences-8-1: What will be stored in the variable ty below?

```
qu = "wow, welcome week!"
ty = qu.index("we")
```

● A. 5

○ B. 6

○ C. 13

○ D. 14

○ E. There is an error.

Check me Compare me

✓ Yes, when we get the index of a string that is longer than one character, we get the index for the first character in the string.

Activity: 5 -- Multiple Choice (question5_8_1)

sequences-8-2: What will be stored in the variable ty below?

```
qu = "wow, welcome week! Were you wanting to go?"
ty = qu.count("we")
```

○ A. 0

● B. 2

○ C. 3

○ D. There is an error.

Check me Compare me

✓ Yes, there is a difference between "we" and "We" which means there are only two in the string.

Activity: 6 -- Multiple Choice (question5_8_2)

sequences-8-3: What will be stored in the variable ht below?

```
rooms = ['bathroom', 'kitchen', 'living room', 'bedroom', 'closet', "foyer"]
ht = rooms.index("garden")
```

○ A. 0

○ B. -1

● C. There is an error.

Check me Compare me

✓ Yes, there is no 'garden' in the list, so we get back an error.

Activity: 7 -- Multiple Choice (question5_8_3)

6.7. Concatenation and Repetition">

ncatenation and Repetition">

6.9. Splitting and Joining Strings">

›

✓ Completed. Well Done!

| Back to top