6.5. Length

The `len` function, when applied to a string, returns the number of characters in a string.

Save & Run 4/30/2021, 6:19:36 PM - 2 of 2 Show in CodeLens

```
fruit = "Banana"
print(len(fruit))

```

```
6
```

Activity: 1 -- ActiveCode (ac5_5_1)

To get the last letter of a string, you might be tempted to try something like this:

Save & Run 4/30/2021, 6:19:40 PM - 2 of 2 Show in CodeLens

```
fruit = "Banana"
sz = len(fruit)
last = fruit[sz]       # ERROR!
print(last)

```

Activity: 2 -- ActiveCode (ac5_5_2)

Error

```
IndexError: string index out of range on line 3
```

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

That won't work. It causes the runtime error `IndexError: string index out of range`. The reason is that there is no letter at index position 6 in `"Banana"`. Since we started counting at zero, the six indexes are numbered 0 to 5. To get the last character, we have to subtract 1 from the length. Give it a try in the example above.

Save & Run 4/30/2021, 6:19:48 PM - 2 of 2 Show in CodeLens

```
fruit = "Banana"
sz = len(fruit)
lastch = fruit[sz-1]
print(lastch)

```

```
a
```

Activity: 3 -- ActiveCode (ac5_5_3)

Typically, a Python programmer would combine lines 2 and 3 from the above example into a single line:

```
lastch = fruit[len(fruit)-1]
```

Though, from what you just learned about using negative indices, using `fruit[-1]` would be a more appropriate way to access the last index in a list.

You can still use the `len` function to access other predictable indices, like the middle character of a string.

```
fruit = "grape"
midchar = fruit[len(fruit)//2]
# the value of midchar is "a"
```

As with strings, the function `len` returns the length of a list (the number of items in the list). However, since lists can have items which are themselves sequences (e.g., strings), it important to note that `len` only returns the top-most length.

Save & Run 4/30/2021, 6:19:52 PM - 2 of 2 Show in CodeLens

```
1 alist =  ["hello", 2.0, 5]
2 print(len(alist))
3 print(len(alist[0]))
4
```

```
3
5
```

Activity: 4 -- ActiveCode (ac5_5_4)

Note that `alist[0]` is the string `"hello"`, which has length 5.

Check your understanding

sequences-5-1: What is printed by the following statements?

```
s = "python rocks"
print(len(s))
```

- A. 11
- B. 12

Check me Compare me

✔ Yes, there are 12 characters in the string.

Activity: 5 -- Multiple Choice (question5_5_1)

sequences-5-2: What is printed by the following statements?

```
alist = [3, 67, "cat", 3.14, False]
print(len(alist))
```

- A. 4
- B. 5

Check me Compare me

✔ Yes, there are 5 items in this list.

Activity: 6 -- Multiple Choice (question5_5_2)

Assign the number of elements in `lst` to the variable `output`.

Save & Run 5/13/2021, 6:19:40 PM - 5 of 5 Show in CodeLens

```
lst = ["hi", "morning", "dog", "506", "caterpillar", "balloons", 106, "yo-yo", "pyt
output = len(lst)

```

Activity: 7 -- ActiveCode (ac5_5_5)

Result	Actual Value	Expected Value	Notes
Pass	52	52	Testing that output value is assigned to correct value.

You passed: 100.0% of the tests

You have attempted 8 of 7 activities on this page

6.4. Disabmiguating []: creation vs indexing">

abmiguating[]: creation vs indexing">

6.6. The Slice Operator">

✔ Completed. Well Done!

| Back to top