



## 6.3. Index Operator: Working with the Characters of a String

The **indexing operator** (Python uses square brackets to enclose the index) selects a single character from a string. The characters are accessed by their position or index value. For example, in the string shown below, the 14 characters are indexed left to right from position 0 to position 13.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
L	u	t	h	e	r	C	o	I	I	e	g	e	

-14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1

It is also the case that the positions are named from right to left using negative numbers where -1 is the rightmost index and so on. Note that the character at index 6 (or -8) is the blank character.

Save & Run    4/30/2021, 6:04:46 PM - 2 of 2    Show in CodeLens

```
1 school = "Luther College"
2 m = school[2]
3 print(m)
4
5 lastchar = school[-1]
6 print(lastchar)
7
```

t  
e

Activity: 1 -- ActiveCode (ac5\_3\_1)

The expression `school[2]` selects the character at index 2 from `school`, and creates a new string containing just this one character. The variable `m` refers to the result.

The letter at index zero of `"Luther College"` is `L`. So at position `[2]` we have the letter `t`.

If you want the zero-eth letter of a string, you just put 0, or any expression with the value 0, in the brackets. Give it a try.

The expression in brackets is called an **index**. An index specifies a member of an ordered collection. In this case the collection of characters in the string. The index *indicates* which character you want. It can be any integer expression so long as it evaluates to a valid index value.

Note that indexing returns a *string* — Python has no special type for a single character. It is just a string of length 1.

### 6.3.1. Index Operator: Accessing Elements of a List or Tuple

The syntax for accessing the elements of a list or tuple is the same as the syntax for accessing the characters of a string. We use the index operator ( `[]` – not to be confused with an empty list). The expression inside the brackets specifies the index. Remember that the indices start at 0. Any integer expression can be used as an index and as with strings, negative index values will locate items from the right instead of from the left.

When we say the first, third or nth character of a sequence, we generally mean counting the usual way, starting with 1. The nth character and the character AT INDEX n are different then: The nth character is at index  $n-1$ . Make sure you are clear on what you mean!

Try to predict what will be printed out by the following code, and then run it to check your prediction.  
(Actually, it's a good idea to always do that with the code examples. You will learn much more if you force yourself to make a prediction before you see the output.)

Save & Run    4/30/2021, 6:05:00 PM - 2 of 2    Show in CodeLens

```
1 numbers = [17, 123, 87, 34, 66, 8398, 44]
2 print(numbers[2])
3 print(numbers[9-8])
4 print(numbers[-2])
5
```

87  
123  
8398

## Activity: 2 -- ActiveCode (ac5\_3\_2)

**Save & Run**

4/30/2021, 6:05:20 PM - 2 of 2

**Show in CodeLens**

```
1 prices = (1.99, 2.00, 5.50, 20.95, 100.98)
2 print(prices[0])
3 print(prices[-1])
4 print(prices[3-5])
5
```

1.99  
100.98  
20.95

## Activity: 3 -- ActiveCode (ac5\_3\_3)

**Check your understanding**

sequences-3-1: What is printed by the following statements?

```
s = "python rocks"
print(s[3])
```

- A. t
- B. h
- C. c
- D. Error, you cannot use the [ ] operator with a string.

**Check me****Compare me**

✓ Yes, index locations start with 0.

## Activity: 4 -- Multiple Choice (question5\_3\_1)

sequences-3-2: What is printed by the following statements?

```
s = "python rocks"
print(s[2] + s[-4])
```

- A. tr
- B. to
- C. ps
- D. nn
- E. Error, you cannot use the [ ] operator with the + operator.

**Check me****Compare me**

✓ For -4 you count from right to left, starting with -1.

## Activity: 5 -- Multiple Choice (question5\_3\_2)

sequences-3-3: What is printed by the following statements?

```
alist = [3, 67, "cat", [56, 57, "dog"], [ ], 3.14, False]
print(alist[5])
```

- A. []
- B. 3.14
- C. False
- D. "dog"

**Check me****Compare me**

✓ Yes, 3.14 is at index 5 since we start counting at 0 and sublists count as one item.

## Activity: 6 -- Multiple Choice (question5\_3\_3)

Assign the value of the 34th element of `lst` to the variable `output`.

[Save & Run](#)

4/30/2021, 7:18:23 PM - 12 of 12

[Show in CodeLens](#)

```
1 lst = ["hi", "morning", "dog", "506", "caterpillar", "balloons", 106, "yo-yo", "pyt
2 output = lst[33]
3
```

## Activity: 7 -- ActiveCode (ac5\_3\_4)

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

You passed: 100.0% of the tests

[Save & Run](#)

5/13/2021, 6:12:57 PM - 25 of 25

[Show in CodeLens](#)

```
1 l = ("hi", "goodbye", "python", "106", "506", 91, ['all', 'Paul', 'Jackie', "UMSI",
2 checking = l[22]
3
4
```

## Activity: 8 -- ActiveCode (ac5\_3\_5)

Result	Actual Value	Expected Value	Notes
Pass	'leade... best'	'leade... best'	Testing that checking has the correct element assigned.

You passed: 100.0% of the tests

Assign the value of the last character of `lst` to the variable `output`. Do this so that the length of `lst` doesn't matter.

[Save & Run](#)

4/30/2021, 7:20:27 PM - 2 of 2

[Show in CodeLens](#)

```
1 lst = "Every chess or checkers game begins from the same position and has a finite
2 output = lst[-1]
3
```

## Activity: 9 -- ActiveCode (ac5\_3\_6)

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

You passed: 100.0% of the tests

[Note](#)

Why does counting start at 0 going from left to right, but at -1 going from right to left? Well, indexing starting at 0 has a long history in computer science having to do with some low-level implementation details that we won't go into. For indexing from right to left, it might seem natural to do the analogous thing and start at -0. Unfortunately, -0 is the same as 0, so  $s[-0]$  can't be the last item. Remember we said that programming languages are formal languages where details matter and everything is taken literally?

You have attempted 10 of 9 activities on this page

✓ Completed. Well Done!

6.2. Strings and Lists">

ngs and Lists">

6.4. Disambiguating []: creation vs indexing">



© Copyright 2017 bradleymiller. Created using Runestone 4.1.17.

| Back to top