

String Methods Continued

Textbook

String Methods Continued

Let's learn about some more string [methods](#) and get some practice using them.



Strip method

Strings have many methods associated with them. One of those methods can be used to remove any excess white space at the beginning or end of a string. This is called the [strip method](#) and is done by typing ".strip()" after the name of the string variable.

```
string1 = " I love Coding! "  
  
print(string1.strip())
```

This will print out

```
I love Coding!
```

Notice how the string now does not have the spaces on each end.

Indexing

The [index number](#) is a number that says where a character is in a string. While indexing, we start counting at zero.

Let's say this is our string - "Coding is awesome!". Each character in this string has its own index number, **starting at zero**.

In this example, the first character (C) is at index 0. The indexes increase by one as you move through each character. Let's say you wanted to print the "g" in this string. "g" is at the 5th index, so our code would look like this:

```
1 string1 = "Coding is awesome!"
2 print(string1[5])
```

Try it!

This would print **g**.

When indexing, spaces also count as an index value. So for the below example, what value is at index 8?

```
1 string1 = "Coding is awesome!"
2 print(string1[8])
```

Try it!

This would print **s** because spaces count for an index number.

You can also start counting backwards in your string by using negative indexing.

```
1 string1 = "Coding is awesome!"
2 print(string1[-3])
```

Try it!

This would print **m**. Notice that when indexing from the end you start counting at one, not zero.

Slicing

Slicing is when we use the indexes to print specific "slices" of a string. To do this, we use colons in the brackets. The first number is the index to start at, and the second number is where to end.

```
1 string1 = "I love coding!"
2 print(string1[4:9])
```

Try it!

This would print out **ve co** because it starts at the 4th index "v" and prints the characters 4 until 9, but not including index 9.

There's one more addition you can make when slicing. The first number is where you start, the second number is where you end, and the third number is how many characters you skip. The default is 1, and that's what the computer will assume if nothing is there.

If you add in the third number, it could look like this:

```
1 string1 = "I love coding!"
2 print(string1[2:9:2])
```

Try it!

This code will print `lv o` because it starts at index 2, ends at 9, and skips over 2 each time.

If a section is left blank, it will go either to the beginning or the end of the string.

```
1 string1 = "I love coding!"
2 print(string1[:5])
```

Try it!

This will print out `I lov`

See what happens if you put a negative number in that third space! Play around with string slicing.

You can read more about this by using the links at the end of the lesson.

Reverse a String

To reverse a string, you start slicing at the end of the string and work backwards. Moving back through the string with a negative number in the third number slot will pull out each letter in turn from back to front.

```
1 string1 = "yodel"
2 reverse = string1[len(string1)::-1]
3 print(reverse)
```

Try it!

Split a String

See what happens with the `.split` method!

```
1 string1 = "I love spaghetti!"
2
3 print(string1.split())
```

Try it!

Length of a String

This returns how many characters the string is long. **For counting how many characters, we start counting at 1.**

```
1 string1 = "I love spaghetti!"
2
3 print(len(string1))
```

Try it!

This will return **17**.

Notice how the spaces are included in the total character count.

Find a Character in a String

This method returns the index value of a certain character. It starts looking at the beginning of the string and returns the index value of the first instance of the character. For example, if your string was `string1 = "Mississippi"` and you were looking for `string1.find("s")`, it would return `2` because that's the index value of the first s.

Remember that indexing starts counting at 0! When we index, we consider spaces as an index value also.

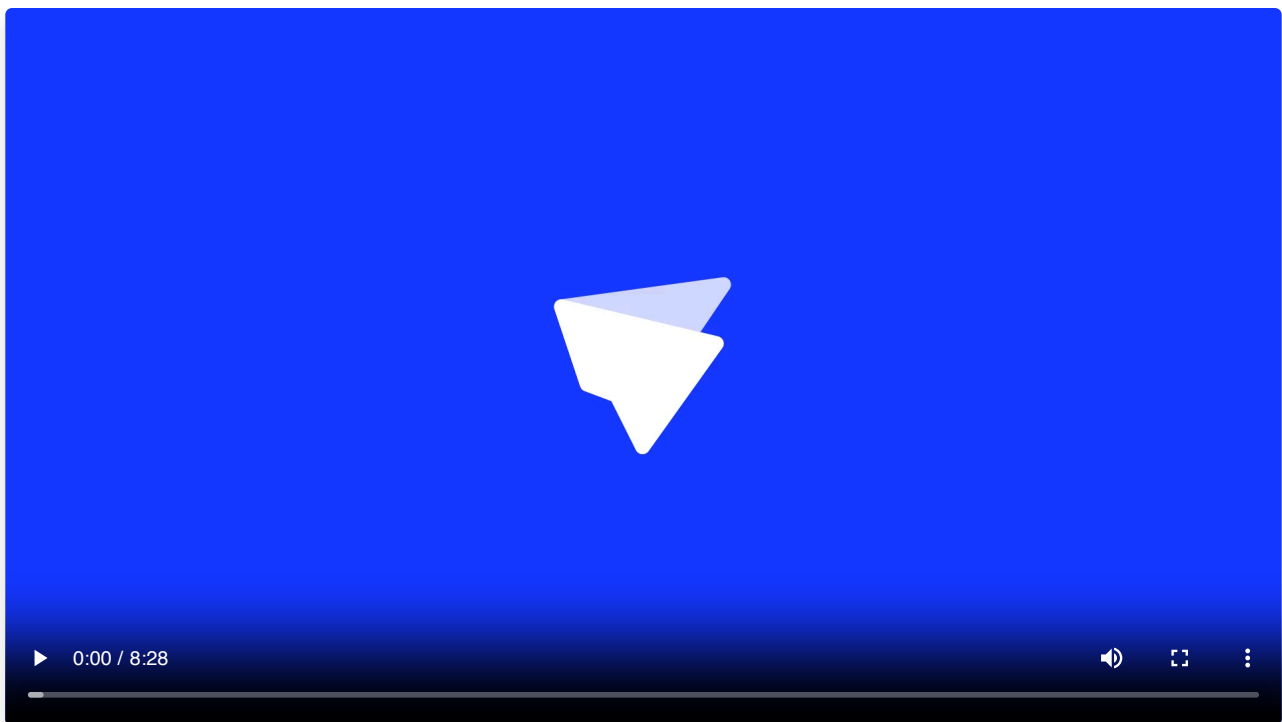
```
1 string1 = "I love Mississippi!"
2
3 print(string1.find("s"))
4
```

Try it!

You can also tell it to start from the end of the string and look for the first instance of a character from the end. This is done with `string1.rfind()`.

```
1 string1 = "I love Mississippi!"
2
3 print(string1.rfind("s"))
```

Try it!



Challenge Help

The challenges in this section are meant to **help you stretch and grow**. They can be more difficult with the variety of methods. We recommend you review **string** indexing to help you with the challenges.

Checkpoint

String Methods Continued

Create a variable named `string1` and assign it to a string.

Use the following methods on the string.

Practice using each of the following methods on the string named `string1`.

1. Use **slicing** on the string
2. Use **split** on the string
3. Find the **length** of the string
4. Use **find** to find the location of a letter in the string.
5. Make sure to print out the string after using each method

So you will have **4 print statements** in the end.

Requirements:

- Use the slicing method on a variable named `string1`
- Use the split method on a variable named `string1`
- Use the length method on a variable named `string1`
- Use the find method on a variable named `string1`

Questions (14)

1. Where do most of the common string methods come from?

MULTIPLE CHOICE

Choose the correct answer:

- A. You, the programmer, need to build the methods before using them.
- B. Python comes with many methods already built in.
- C. Methods are always custom built for each situation.
- D. Methods must first be imported.

2. What would this code output?

MULTIPLE CHOICE

```
string = "My favorite animal is the platypus!" print(string.rfind("p"))
```

Choose the correct answer:

- A. 31
- B. 26
- C. 1
- D. 17

3. What number do we start counting at when we are indexing from the beginning?

MULTIPLE CHOICE

Choose the correct answer:

- A. 0
- B. 1
- C. 2
- D. 5
- E. 10

4. True or False: You count spaces when you are indexing.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

5. What will the following code print out?

```
string1 = "Popcorn is so good at a movie." print(string1[4])
```

Choose the correct answer:

- A. o
- B. c
- C. good
- D. p

6. What will the following code print out?

```
string1 = "Bunjee jumping sounds scary." print(string1[6])
```

Choose the correct answer:

- A. e
- B. j
- C. a space
- D. u

7. What will the following code print out?

```
string1 = "Snorkeling is fun!" print(string1[-3])
```

Choose the correct answer:

- A. o
- B. r
- C. u
- D. f

8. What will the following code print out?

```
string1 = "The quick brown fox jumped over the lazy dog" print(string1[4:9])
```

Choose the correct answer:

- A. quick
- B. quic
- C. uick
- D. e quic

9. What will the following code print out?

```
string1 = "The quick brown fox jumped over the lazy dog" print(string1[8:])
```

Choose the correct answer:

- A. k brown fox jumped over the lazy dog
- B. brown fox jumped over the lazy dog
- C. b
- D. brown

10. What will the following code print out?

```
string1 = "I saw three ducks this morning" print(string1.split())
```

Choose the correct answer:

- A. ['I', 'saw', 'three', 'ducks', 'this', 'morning']
- B. ["I saw three", "ducks this morning"]
- C. "I saw three"
- D. "ducks this morning"

11. What will the following code print out?

```
string1 = "I saw three ducks." print(len(string1))
```

Choose the correct answer:

- A. 18
- B. 17
- C. 15
- D. 14

12. True or False: Spaces are not included in the total character count when finding the length of a string.**Choose the correct answer:**

- A. True
- B. False

13. What will the following code print out?

```
string1 = "Firetrucks are red" print(string1.find("t"))
```

Choose the correct answer:

- A. 4
- B. 5
- C. 1
- D. yes
- E. True

14. Edit the text box below to debug (fix) the code:**Code to Debug:**

```
1 string1 = "Firetrucks are red"
2
3 print(string1.find(r))
```

Challenges (7)

1. How many Words?

Write a program that takes a string as input. Print the number of words in the string.

For example:

Input: `Hello There World`

Output: `3`

Another example:

Input: `You are the best example to me!`

Output: `7`

Hint: This string method `string.split()` will take a string with many words in it and return a list of each word separated into its own string.

Hint: You can use this code `len()` to find the length of a list as well as the length of a string:

2. First and Last

Write a program that takes a string as input. Make sure the string has several instances of the letter "e". Print the index of the **first occurrence of "e"** and the **last occurrence of "e"**, separated by a **hyphen**.

For example:

Input: `I say we should test this sentence.`

Output: `7-33`

Another example:

Input: `We just caught a butterfly!`

Output: `1-21`

Hint: use `string.find()` and `string.rfind()`

Hint: the indexing number is considered an integer (so students need to turn it back into a string before concatenating)

3. String Character Selection

Write a program that takes one string input from the user.

In one print statement, print the **third character** of the string, the **third to last character** of the string, and the **fourth character**.

Note: Index counting starts at 0

Hint: You are looking to print the third character, NOT the character at index number 3.

For example:

Input: `ghostlike`

Output: `ois`

Another example:

Input: `hiking`

Output: `kii`

Hint: You will use concatenation in this challenge

4. Second Time

Write a program that takes in a string as input. The string needs to have at least two instances of the letter "g". Find and print the index of the second occurrence of the letter "g".

For example:

Input: got to go!

Output: 7

Another example:

Input: She gave me a gopher

Output: 14

Hint: remember that indexing numbers starts at "0". Think about how to use slicing `stringname[:]` and `stringname.find()` for this challenge.

Hint: try adding 1 to the index value

5. Cut the String

Write a program that takes in one string as input, and divides the string into two equal parts. **Then, it puts the second half in the front.**

If there are an odd number of characters, include the middle character in the second part that gets moved to the front.

Your program should swap the order of the two parts and print the result.

For example:

Input: testing

Output: tingtes

Input: swimming

Output: mingswim

Hint: For this challenge you will use string slicing `string1[:]`

Hint: For this challenge you will use `len()` to find out how many characters are in a string, then you can divide that by 2 to get the center index value.

6. Remove String Section

Write a program that receives a string as input. This string will have the letter "j" in it at least twice. Remove the first and last occurrence of the letter "j", along with all the characters between them.

For example:

Input: `Enjoy the juice!`

Output: `Enuice!`

Another example:

Input: `rejoice with jumps`

Output: `reumps`

Hint: Use `stringname[:]` for this challenge

Hint: Use `string.find()` and `string.rfind()` for this challenge. You might consider using concatenation as well.

7. Reverse String Section

Write a program that takes one string as input. This string will have at least two occurrences of the letter "j". Reverse the characters between the first and last instance of "j" **including the js**, and print the whole string.

Include both j's in your string answer.

For example:

Input: `Enjoy the jumping beans!`

Output: `j eht yoj`

Another example:

Input: `rejoice for the jerky!`

Output: `j eht rof ecioj`

Hint: For this challenge you might consider `using string1.find()` and `string1.rfind()`

Hint: For this challenge, experiment with the third value in `string1[: : -1]` (You can also put variable values between the brackets [])

Answer Keys & Solutions

Checkpoint Solutions

String Methods Continued

```
1 string1 = "Learning to code is challenging and fun!"
2
3 print(string1[2:9])
4
5 print(string1.split())
6
7 print(len(string1))
8
9 print(string1.find("s"))
```

Questions

1. Where do most of the common string methods come from?

MULTIPLE CHOICE

Correct Answer:

- A. You, the programmer, need to build the methods before using them. ✗ Incorrect
- B. Python comes with many methods already built in. ✓ Correct
- C. Methods are always custom built for each situation. ✗ Incorrect
- D. Methods must first be imported. ✗ Incorrect

Explanation:

Python has many useful methods built into the language.

2. What would this code output?

MULTIPLE CHOICE

Correct Answer:

- A. 31 ✓ Correct
- B. 26 ✗ Incorrect
- C. 1 ✗ Incorrect

D. 17

✗ Incorrect

Explanation:

This will be the index value of the last instance of the letter p. Don't forget to count spaces and start counting at 0.

3. What number do we start counting at when we are indexing from the beginning?

MULTIPLE CHOICE

Correct Answer:

A. 0

✓ Correct

B. 1

✗ Incorrect

C. 2

✗ Incorrect

D. 5

✗ Incorrect

E. 10

✗ Incorrect

Explanation:

Indexing does not start counting at 1 like we are used to.

4. True or False: You count spaces when you are indexing.

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

Spaces count as a character when indexing.

5. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. o

✓ Correct

B. c

✗ Incorrect

C. good

✗ Incorrect

D. p

✗ Incorrect

Explanation:

Remember, when indexing, start counting at 0.

6. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. e

✗ Incorrect

B. j

✗ Incorrect

C. a space

✓ Correct

D. u

✗ Incorrect

Explanation:

Remember, when indexing, start counting at 0. Don't forget to count the spaces.

7. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. o

✗ Incorrect

B. r

✗ Incorrect

C. u

✓ Correct

D. f

✗ Incorrect

Explanation:

When counting negative index numbers, start counting at 1 from the end.

8. What will the following code print out?

Correct Answer:

- | | |
|-----------|-------------|
| A. quick | ✓ Correct |
| B. quic | ✗ Incorrect |
| C. uick | ✗ Incorrect |
| D. e quic | ✗ Incorrect |

Explanation:

The second number is the index value that will not be included--it will cut the string to the left of that value.

9. What will the following code print out?

Correct Answer:

- | | |
|---|-------------|
| A. k brown fox jumped over the lazy dog | ✓ Correct |
| B. brown fox jumped over the lazy dog | ✗ Incorrect |
| C. b | ✗ Incorrect |
| D. brown | ✗ Incorrect |

Explanation:

If there is no value in the slice--it will go all the way to the end of the string.

10. What will the following code print out?

Correct Answer:

- | | |
|--|-------------|
| A. ['I', 'saw', 'three', 'ducks', 'this', 'morning'] | ✓ Correct |
| B. ["I saw three", "ducks this morning"] | ✗ Incorrect |
| C. "I saw three" | ✗ Incorrect |
| D. "ducks this morning" | ✗ Incorrect |

Explanation:

.split() separates a string at the spaces.

11. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|-------|-------------|
| A. 18 | ✓ Correct |
| B. 17 | ✗ Incorrect |
| C. 15 | ✗ Incorrect |
| D. 14 | ✗ Incorrect |

Explanation:

When counting the length, start counting at 1.

12. True or False: Spaces are not included in the total character count when finding the length of a string.

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------|-------------|
| A. True | ✗ Incorrect |
| B. False | ✓ Correct |

Explanation:

Spaces are included in counting.

13. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--------|-------------|
| A. 4 | ✓ Correct |
| B. 5 | ✗ Incorrect |
| C. 1 | ✗ Incorrect |
| D. yes | ✗ Incorrect |

E. True

✖ Incorrect

Explanation:

`.find()` returns the index value of the first instance of the value.

14. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 string1 = "Firetrucks are red"
2
3 print(string1.find(r))
```

Correct Solution:

```
1 string1 = "Firetrucks are red"
2
3 print(string1.find("r"))
```

Explanation:

The value inside `find()` needs quotation marks.

Challenges

1. How many Words?

Solution:

```
1 string1 = input("Enter a string")
2
3 new_string1 = string1.split()
4
5 print(len(new_string1))
```

2. First and Last

Solution:

```
1 string = input()
2 print(str(string.find("e")) + "-" + str(string.rfind("e")))
```

3. String Character Selection

Solution:

```
1 string1 = input("Enter a string")
2
3 third = string1[2]
4
5 third_to_last = string1[-3]
6
7 fourth = string1[3]
8
9 print(third + third_to_last + fourth)
```

4. Second Time

Solution:

```
1 string = input('Enter a string')
2 first = string.find('g')
3
4 new_string = string[(first + 1):]
5
6 second=new_string.find('g') + 1
7
8 print(second + first)
```

5. Cut the String

Solution:

```
1 string1 = input("Enter a string")
2
3 string1_length = len(string1)
4
5 half = string1_length / 2
6
7 first_half = string1[0:int(half)]
8 second_half = string1[int(half):]
9
10 print(second_half + first_half)
```

6. Remove String Section

Solution:

```
1 string1 = input("Enter a string with the letter j in it twice")
2
3 firstj = string1.find("j")
4
5 secondj = string1.rfind("j")
6
7 new_string = string1[:firstj] + string1[secondj+1:]
8
9 print(new_string)
```

7. Reverse String Section

Solution:

```
1 string1 = input("Enter a string with the letter j in it twice")
2
3 firstj = string1.find("j")
4
5 secondj = string1.rfind("j")
6
7 new_string = string1[firstj:secondj+1]
8
9 reverse = new_string[len(new_string):-1]
10
11 print(reverse)
```