

## String Methods

---

### Textbook

---

## String Methods



[Methods](#) are keywords which a program can use to modify data within a variable. Methods are functions that are built into the Python language. We will learn to create our own functions later.

To use a method on a variable, you must type the name of the variable then type a period and the name of the method you wish to use. After the name of the method you must type opening and closing parentheses. This is shown below.

```
string1.methodname()
```

In the example above, "string1" is the name of the string variable. The ".methodname()" written after the variable name is a method called "methodname."

Programmers can write their own methods, but python comes with many methods already prewritten for the user.

Let's explore what this looks like.

## Lower and Upper Methods

The [lower and upper methods](#) are used to make all characters in a string change to be either lowercase or uppercase.

- These functions have no effect on numbers or special characters.

```
1 string1 = "I love Coding!"
2 print(string1.lower())
3 print(string1.upper())
```

Try it!

## Replace Method

The [replace method](#) changes a specified character or characters (letters, numbers, or symbols) in the string to something else. Unlike the other methods you have used thus far, you will need to put something inside the parentheses.

The replace method calls for two separate strings to be placed inside of the parentheses separated by a comma. The first string is what you want to change and the second is what you want to change it to.

```
1 string1 = "Coding is cool!"
2 print(string1.replace("cool", "fun"))
```

Try it!

Note: the characters inside the replace method must be **identical** to the characters in the string for the method to work correctly.

## isalpha() Method

The `isAlpha()` method checks to see if every character in a text is a letter.

```
1 greeting = "Hello"
2
3 print(greeting.isalpha())
```

Try it!

This will return `True` since every character is a letter.

Let's look at another example:

```
1 greeting = "4 wheel drive"
2
3 print(greeting.isalpha())
```

Try it!

This will return `False` since not every character is a letter.

## isnumeric() Method

Similar to the `isalpha()` method, the `isnumeric()` method checks to see if every value in a string is a number.

```
1 greeting = "10058"  
2  
3 print(greeting.isnumeric())
```

Try it!

This will return `True` because all the values in the string are numbers.

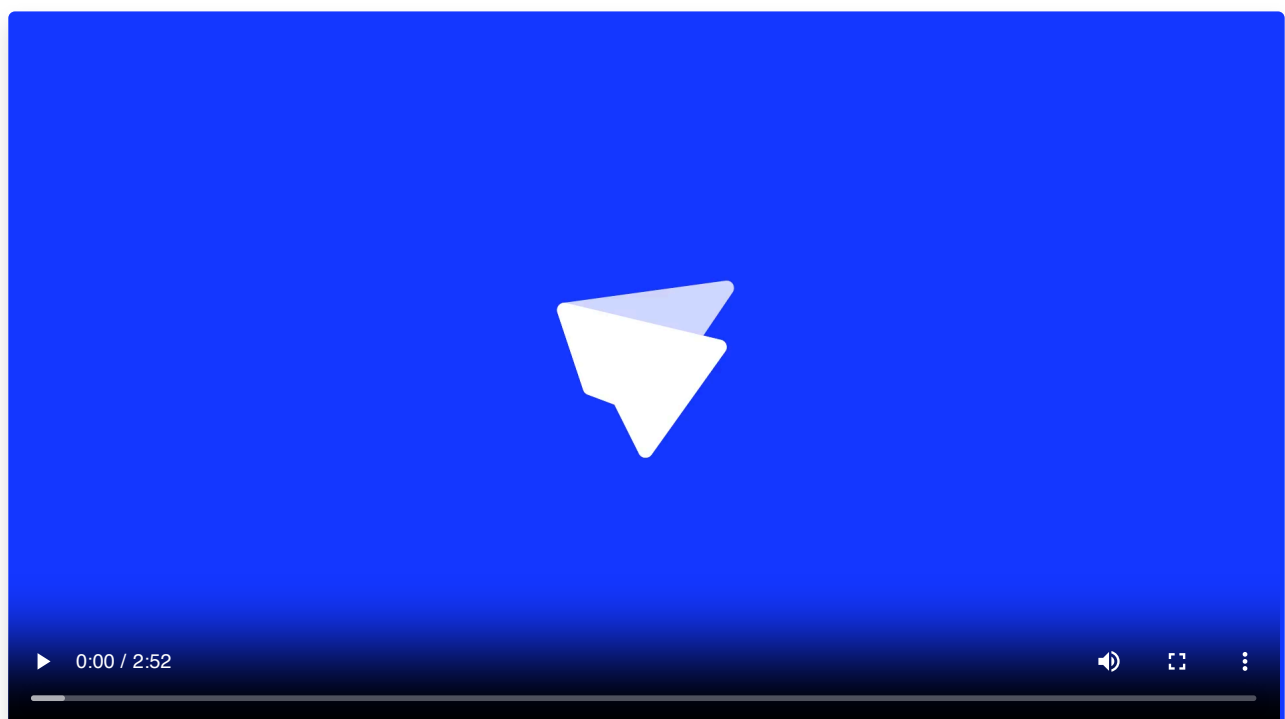
## Decomposing Large Tasks with Small Code Segments

When you're faced with a big, complex problem in programming, trying to solve it all at once can feel overwhelming. A powerful strategy is to decompose the problem, which means breaking it down into smaller, more manageable pieces. Each of these smaller pieces can then be handled by its own dedicated code segment, making the overall problem much easier to tackle.

Imagine you need to build a program that handles customer orders for an online store. This is a big problem! You could decompose it into smaller code segments: one segment to take the customer's order, another to calculate the total cost, a third to process payment, and a final one to send a confirmation email. Each of these smaller tasks can be written and tested independently.

By defining these new code segments (which often become functions or methods, as you'll learn more about), you make your code more organized, readable, and easier to debug. If there's an issue with how the payment is processed, you know exactly which code segment to examine, rather than sifting through a massive block of code.

This approach also makes your code reusable; once you have a "calculate total cost" segment, you can use it in other parts of your program or even in entirely different projects. Decomposing problems into smaller, focused code segments is a fundamental skill for any programmer, allowing you to build complex and robust applications more effectively.



## Checkpoint

### String Methods

Declare **three** new string variables.

Use the following methods and print their output: `upper()` `lower()` `replace()`

Make sure you print each method's output!

#### Requirements:

- Use `upper()` on a string variable
- Use `lower()` on a string variable
- Use `replace()` on a string variable

### Questions (11)

#### 1. What happens when you use `string.lower()` or `string.upper()` on an integer data type?

MULTIPLE CHOICE

Choose the correct answer:

- A. Nothing
- B. It rounds down or rounds up.
- C. It drops the decimal.
- D. It rounds to the ones place or to the tens place.

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

DEBUG CODE

Code to Debug:

```
1 vacation.replace(beach, mountain)
```

#### 3. What would this code output?

MULTIPLE CHOICE

```
string = "I love dinosaurs so much!" print(string.replace("I", ""))
```

Choose the correct answer:

- A. love dinosaurs so much!
- B. I love otters so much!
- C. love dnosaurs so much!

#### 4. What would this code output?

MULTIPLE CHOICE

```
string = "I love otters so much!" print(string.replace("otters", "manatees"))
```

Choose the correct answer:

- A. I love otters so much!
- B. I love manatees so much!
- C. I love so much!
- D. I love otters and manatees so much!

#### 5. What will the following code print out?

MULTIPLE CHOICE

```
string1 = "My name is Simon." print(string1.lower())
```

Choose the correct answer:

- A. MY NAME IS SIMON.
- B. my name is simon.
- C. My name is simon.
- D. my name is Simon.

#### 6. What will the following code print out?

MULTIPLE CHOICE

```
string1 = "My name is Simon." print(string1.upper())
```

Choose the correct answer:

- A. MY NAME IS SIMON.
- B. my name is simon.
- C. My name is simon.
- D. my name is Simon.

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

DEBUG CODE

Code to Debug:

```
1 string1 = "Let's go swimming."  
2 print(string1.replace(swimming, hiking))
```

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

DEBUG CODE

Code to Debug:

```
1 string1 = "Let's go swimming."  
2 print(string1.replace("swimming", "hiking"))
```

9. Which string method can be used to make all the letters upper case?

MULTIPLE CHOICE

Choose the correct answer:

- A. string1.lower( )
- B. string1.upper( )
- C. string1.caps( )
- D. string1.raise( )

10. Which method can be used to replace parts of a string?

MULTIPLE CHOICE

Choose the correct answer:

- A. string1.replace( )
- B. string1.change( )
- C. string1.switch( )
- D. string1.swap( )

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

DEBUG CODE

Code to Debug:

```
1 string1-upper()
```

## Challenges (4)

### 1. 1 to one

Write a program that receives a string as input. This string will include at least one character "1". Replace every occurrence of "1" with "one".

For example:

Input: `Free food for every 1`

Output: `Free food for every one`

Another example:

Input: `All for 1 and 1 for all`

Output: `All for one and one for all`

**Reminder to match the output exactly! Check spelling, punctuation, capitalization, and spacing**

### 2. Shouting Sentences

Have you ever received a message that was all in uppercase letters? It often seems like messages like this are shouting.

Create an input that takes in a message and turns it into a shout.

For example:

Input: `Welcome to Biology Class`

Output: `WELCOME TO BIOLOGY CLASS`

Another example:

Input: `It's good to see you Blake`

Output: `IT'S GOOD TO SEE YOU BLAKE`

Make sure the input has both lower and uppercase letters.

**The output should print out the message all in uppercase letters.**

### 3. Group Thank You Message

You are part of a school club that just got a generous donation of money to help run activities! You are composing a thank you message to send to the donor for their kind donation.

When you showed the message to the Vice President of the club, they suggested that you **change all the instances of the word "I" to "We"**, so that the thank you note will be from the whole club. For example, instead of "I just wanted to thank you..." change it to "We just wanted to thank you..."

Create an input that asks for a thank you message with several instances of the word "I" in it.

Change the instances of the word "I" to the word "we."

For example:

Input: `I just wanted to thank you! I loved how much you helped. I will need to pay it forward!`

Output: `We just wanted to thank you! We loved how much you helped. We will need to pay it forward!`

Another example:

Input: `I just wanted to thank you for your donation. I think its awesome and I will use it well.`

Output: `We just wanted to thank you for your donation. We think its awesome and We will use it well.`

**Hint:** For this challenge, give all of the instances of the word "We" a capital "W". If it appears in the center of a sentence, it will just be capitalized and it's okay for this challenge.

### 4. Where is the @?

Write a program that takes a string as input. Make sure the string has several @ symbols.

Delete all of the "@" characters from this string, and **print the result**.

For example:

Input: `test@test.com`

Output: `testtest.com`

Another example:

Input: `@m@zon`

Output: `mzon`

This challenge can be a little tricky. See what happens when you replace a character with a string with nothing in it!

```
sentence.replace("item", "")
```



## Answer Keys & Solutions

### Checkpoint Solutions

#### String Methods

```
1 string_1 = "this"
2 string_2 = "IS"
3 string_3 = "a test"
4 print(string_1.upper())
5 print(string_2.lower())
6 print(string_3.replace("a", "the"))
```

### Questions

1. What happens when you use `string.lower()` or `string.upper()` on an integer data type?

MULTIPLE CHOICE

Correct Answer:

- |  |             |
|--|-------------|
| A. Nothing   | ✓ Correct   |
| B. It rounds down or rounds up.                      | ✗ Incorrect |
| C. It drops the decimal.                             | ✗ Incorrect |
| D. It rounds to the ones place or to the tens place. | ✗ Incorrect |

#### Explanation:

`string.lower()` or `string.upper()` are meant for strings

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

DEBUG CODE

Incorrect Code:

```
1 vacation.replace(beach, mountain)
```

Correct Solution:

```
1 vacation.replace("beach", "mountain")
```

#### Explanation:

We need quotation marks around the strings

### 3. What would this code output?

MULTIPLE CHOICE

Correct Answer:

- A. love dinosaurs so much! ✓ Correct
- B. I love otters so much! ✗ Incorrect
- C. love dnosaurs so much! ✗ Incorrect

#### Explanation:

This code replaces the string "I" with an empty string "".

### 4. What would this code output?

MULTIPLE CHOICE

Correct Answer:

- A. I love otters so much! ✗ Incorrect
- B. I love manatees so much! ✓ Correct
- C. I love so much! ✗ Incorrect
- D. I love otters and manatees so much! ✗ Incorrect

#### Explanation:

This code replaces the string "otters" with the string "manatees".

### 5. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- A. MY NAME IS SIMON. ✗ Incorrect
- B. my name is simon. ✓ Correct
- C. My name is simon. ✗ Incorrect
- D. my name is Simon. ✗ Incorrect

**Explanation:**

.lower( ) makes the string lower case.

**6. What will the following code print out?**

MULTIPLE CHOICE

**Correct Answer:**

- |                      |             |
|----------------------|-------------|
| A. MY NAME IS SIMON. | ✓ Correct   |
| B. my name is simon. | ✗ Incorrect |
| C. My name is simon. | ✗ Incorrect |
| D. my name is Simon. | ✗ Incorrect |

**Explanation:**

.upper( ) makes the string upper case.

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

DEBUG CODE

**Incorrect Code:**

```
1 string1 = "Let's go swimming."  
2 print(string1.replace(swimming, hiking))
```

**Correct Solution:**

```
1 string1 = "Let's go swimming."  
2 print(string1.replace("swimming", "hiking"))
```

**Explanation:**

The strings all need quotation marks around them.

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

DEBUG CODE

**Incorrect Code:**

```
1 string1 = "Let's go swimming."  
2 print(string1.replace("swimming", "hiking"))
```

**Correct Solution:**

```
1 string1 = "Let's go swimming."
```

```
2 print(string1.replace("swimming", "hiking"))
```

**Explanation:**

`string1.replace()`

**9. Which string method can be used to make all the letters upper case?**

MULTIPLE CHOICE

**Correct Answer:**

- |                                 |             |
|---------------------------------|-------------|
| A. <code>string1.lower()</code> | ✗ Incorrect |
| B. <code>string1.upper()</code> | ✓ Correct   |
| C. <code>string1.caps()</code>  | ✗ Incorrect |
| D. <code>string1.raise()</code> | ✗ Incorrect |

**Explanation:**

`upper` is short for uppercase.

**10. Which method can be used to replace parts of a string?**

MULTIPLE CHOICE

**Correct Answer:**

- |                                   |             |
|-----------------------------------|-------------|
| A. <code>string1.replace()</code> | ✓ Correct   |
| B. <code>string1.change()</code>  | ✗ Incorrect |
| C. <code>string1.switch()</code>  | ✗ Incorrect |
| D. <code>string1.swap()</code>    | ✗ Incorrect |

**Explanation:**

`.replace` will swap out parts in the string.

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

DEBUG CODE

**Incorrect Code:**

```
1 string1-upper()
```

#### Correct Solution:

```
1 string1.upper()
```

#### Explanation:

The hyphen needs to be replaced with something.

## Challenges

### 1. 1 to one

#### Solution:

```
1 cheer = input("Create a string that includes the number 1")
2
3
4 print(cheer.replace("1", "one"))
```

### 2. Shouting Sentences

#### Solution:

```
1 shout = input("Create a sentence of something you might shout. Make sure it includes
lower and uppercase letters.")
2
3 print(shout.upper())
```

### 3. Group Thank You Message

#### Solution:

```
1 message = input("Create a thank you message with the word I in it")
2
3
4
5 print(message.replace("I", "We"))
```

### 4. Where is the @?

#### Solution:

```
1 sentence = input("Create a string that includes several @ symbols")
2
```

```
3  
4 print(sentence.replace("@", ""))
```