

200 XP

# Exercise - Use the Remove() and Replace() methods

8 minutes

In this exercise, you'll remove characters from a string using the `Remove()` method and replace characters using the `Replace()` method.

Sometimes, you'll need to modify the contents of the string, removing or replacing characters. While you could replace characters with the tools you already know, it requires a bit of temporarily storing and stitching strings back together. Fortunately, the `string` data type has other built-in methods, `Remove()` and `Replace()`, for these specialized scenarios.

## Use the `Remove()` method

You would typically use `Remove()` when there's a standard and consistent position of the characters you want to remove from the string.

This exercise has data stored in older files that is fixed length, and character positions are allocated for certain fields of information. The first five digits represent a customer identification number. The next 20 digits contain a customer's name. The next six positions represent the customer's latest invoice amount, and the last three positions represent the number of items ordered on that invoice.

In the following steps you will need to remove the customer's name to format the data so that it can be sent to a separate process. Since you know the exact position and length of the user's name, you can easily remove it using the `Remove()` method.

## Remove characters in specific locations from a string

1. Delete or use the line comment operator `//` to comment out all of the code from the previous exercises.
2. Update your code in the Visual Studio Code Editor as follows:

C#

```
string data = "12345John Smith          5000 3 ";  
string updatedData = data.Remove(5, 20);  
Console.WriteLine(updatedData);
```

3. On the Visual Studio Code **File** menu, select **Save**.

The Program.cs file must be saved before building or running the code.

4. In the EXPLORER panel, to open a Terminal at your TestProject folder location, right-click **TestProject**, and then select **Open in Integrated Terminal**.

A Terminal panel should open, and should include a command prompt showing that the Terminal is open to your TestProject folder location.

5. At the Terminal command prompt, to run your code, type **dotnet run** and then press Enter.

You should see the following output:

Output

123455000 3

The `Remove()` method works similarly to the `Substring()` method. You supply a starting position and the length to remove those characters from the string.

## Use the `Replace()` method

You would use the `Replace()` method when you must replace one or more characters with a different character (or no character). The `Replace()` method is different from the other methods you've used so far, it will replace every instance of the given characters, not just the first or last instance.

## Remove characters no matter where they appear in a string

1. Update your code in the Visual Studio Code Editor as follows:

C#

```
string message = "This--is--ex-amp-le--da-ta";  
message = message.Replace("--", " ");  
message = message.Replace("-", "");  
Console.WriteLine(message);
```

2. Save your code file, and then use Visual Studio Code to run your code.

You should see the following output:

Output

This is example data

Here you used the `Replace()` method twice. The first time you replaced the string `--` with an empty space. The second time you replaced the string `-` with an empty string, which completely removes the character from the string.

## Recap

Here's the most important things to remember from this unit:

- The `Remove()` method works like the `Substring()` method, except that it deletes the specified characters in the string.
- The `Replace()` method swaps all instances of a string with a new string.

## Check your knowledge

1. If the code `message = message.Replace("B", "D");` is run, where `string message = "Big Dog";`, what is the new value of `message`? \*

- ☐ "Big Dog"
- ☐ "Big Bog"
- ☐ "Dig Dog"

Check your answers

---