

## How to replace a char in string with an Empty character in C#.NET

[Ask Question](#)

▲ I have a string like this:

61 `string val = "123-12-1234";`

▼ How can I replace the dashes using an empty string in C#.

★ I mean `val.Replace(char oldChar, newChar);`

5

What needs to go in `oldChar` and `newChar` .

c#

edited Oct 31 '14 at 13:52



[Matas Vaitkevicius](#)

35.1k 17 171 181

asked Jun 16 '11 at 14:12



[SaiBand](#)

1,656 9 40 62

3 `string val = "123-12-1234".Replace("-", String.Empty );` // Note: please at least attempt to solve a problem before posting such questions. – [Brandon Moretz](#) Jun 16 '11 at 14:15 ✎

9 Why the downvote? It may be a simple question to answer but that does not make it wrong surely? New users may become disheartened quite quickly... – [Peter Kelly](#) Jun 16 '11 at 14:17

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- 3 possible duplicate of [How to replace occurrences of "-" with an empty string?](#) – [cobbal](#) Jun 16 '11 at 15:15
- 5 Somebody mind explaining why this got closed as *too localized* ? – [Xeo](#) Jun 16 '11 at 17:27

## 8 Answers



You can use a different overload of `Replace()` that takes string.

94



```
val = val.Replace("-", string.Empty)
```



answered Jun 16 '11 at 14:14



[Bala R](#)

91.6k 21 164 191

- 1 +1 for mentioning that the alternative method is an "overload" – [Nick](#) Jun 16 '11 at 14:19

Thanks, that did the trick. – [SaiBand](#) Jun 16 '11 at 14:29

Try replacing the `\0` character using the above overload and it will not work! The `\0` character will not be replaced. This is especially true when working with string data received from Snmp devices – [Steve Johnson](#) May 14 '17 at 14:41

The reason for why char to char replacement overload will not work here - [Why is there no Char.Empty like String.Empty?](#) – [RBT](#) Jun 6 '18 at 2:55



Since the other answers here, even though correct, do not explicitly address your initial doubts, I'll do it

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



If you call `string.Replace(char oldChar, char newChar)` it will replace the occurrences of a character with another character. It is a one-for-one replacement. Because of this the length of the resulting string will be the same.

What you want is to *remove* the dashes, which, obviously, is not the same thing as replacing them with another character. You cannot replace it by "no character" because 1 character is always 1 character. That's why you need to use the overload that takes strings: strings can have different lengths. If you replace a string of length 1, with a string of length 0, the effect is that the dashes are gone, replaced by "nothing".

answered Jun 16 '11 at 14:42



[R. Martinho Fernandes](#)

165k 59 383 470

---

3 Thanks for giving a detailed explanation. – [SaiBand](#) Jun 16 '11 at 17:11

---

1 After all that, you didn't demonstrate. :P Oh well. +1 anyway for the explanation – [cHao](#) Jun 16 '11 at 17:42

---

1 You can find more information [here](#). – [Dinei](#) Mar 29 '16 at 12:48

---



This seems too simple, but:

5

```
val.Replace("-", "");
```



answered Jun 16 '11 at 14:14



[Richard Dalton](#)

31.3k 6 64 85

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



```
val.Replace("-", "");
```

5

[MSDN Source](#)

edited Jul 15 '14 at 11:47

[Soner Gönül](#)

82k 27 153 287

answered Jun 16 '11 at 14:14

[Coeffect](#)

7,494 2 20 40



```
string val = "123-12-1234";
```

4

```
val = val.Replace("-", ""); // result: 123121234
```



answered Jun 16 '11 at 14:14

[Tim S. Van Haren](#)

8,211 2 26 34

Yes, just checked the results. That worked for me – [SaiBand](#)  
Jun 16 '11 at 14:31



If you are in a loop, let's say that you loop through a list of punctuation characters that you want to remove, you can do something like this:

4

[Home](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

## Stack Overflow

Tags

Users

Jobs

## Teams

Q&amp;A for work

[Learn More](#)

```

private const string PunctuationChars = ".,!?${";
foreach (var word in words)
{
    var word_modified = word;

    var modified = false;

    foreach (var punctuationChar in Punctu
    {
        if (word.IndexOf(punctuationChar) :
        {
            modified = true;
            word_modified = word_modified.l

""");

        }
    }
}
//////////MORE CODE
}

```

The trick being the following:

```
word_modified.Replace("" + punctuationChar, "");
```

answered Apr 17 '16 at 13:45



**Razvan Dumitru**

4,905 3 19 42



1

If you want to replace a char in a string with an empty char that means you want to remove that char from a string, read the answer of R. Martinho Fernandes.



Here is an exemple of how to remove a char from a string (replace with an "Empty char"):

```

    if (indexOfChar >= 0)
    {
        input = input.Remove(indexOfChar, 1);
    }
    return input;
}

```

or this version that removes all recurrences of a char in a string :

```

public static string RemoveCharFromString(string input)
{
    int indexOfChar = input.IndexOf(charItem);
    if (indexOfChar < 0)
    {
        return input;
    }
    return RemoveCharFromString(input.Remove(indexOfCh.
}

```

edited May 2 '16 at 14:07

answered May 2 '16 at 11:05



Alexei Bondarev

573 5 9



0

I have a latin version of the code of Razvan Dumitru because us use a million indicator even. Offcourse I use a double replace :D



```

public static string CleanNumb(string numb)
{
    foreach (char c in ".,'")
        numb = numb.Replace(c, ' ');
}

```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

answered Sep 13 '18 at 14:31



[Jorge Alejandro Raigoza](#)

1 1

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).