**修改字符串内容（C# 编程指南）**

由于字符串是不可变的，因此一个字符串对象一旦创建，值就不能再更改（在不使用不安全代码的情况下）。不过，修改字符串的值然后将结果存储到新的字符串对象中有很多种方法。 [System.String](https://msdn.microsoft.com/zh-cn/library/system.string.aspx) 类提供作用于输入字符串并返回新字符串对象的方法。在很多情况下，可以将这个新对象赋给保存原始字符串的变量。 [System.Text.RegularExpressions.Regex](https://msdn.microsoft.com/zh-cn/library/system.text.regularexpressions.regex.aspx) 类提供其他一些以类似方式工作的方法。 [System.Text.StringBuilder](https://msdn.microsoft.com/zh-cn/library/system.text.stringbuilder.aspx) 类提供一个可“就地”修改的字符缓冲区。调用 [StringBuilder.ToString](https://msdn.microsoft.com/zh-cn/library/4b1063zt.aspx) 方法可新建包含此缓冲区的当前内容的字符串对象。

下面的示例演示替换或移除指定字符串中的子字符串的各种方法。

class ReplaceSubstrings

{

string searchFor;

string replaceWith;

static void Main(string[] args)

{

ReplaceSubstrings app = new ReplaceSubstrings();

string s = "The mountains are behind the clouds today.";

// Replace one substring with another with String.Replace.

// Only exact matches are supported.

s = s.Replace("mountains", "peaks");

Console.WriteLine(s);

// Output: The peaks are behind the clouds today.

// Use Regex.Replace for more flexibility.

// Replace "the" or "The" with "many" or "Many".

// using System.Text.RegularExpressions

app.searchFor = "the"; // A very simple regular expression.

app.replaceWith = "many";

s = Regex.Replace(s, app.searchFor, app.ReplaceMatchCase, RegexOptions.IgnoreCase);

Console.WriteLine(s);

// Output: Many peaks are behind many clouds today.

// Replace all occurrences of one char with another.

s = s.Replace(' ', '\_');

Console.WriteLine(s);

// Output: Many\_peaks\_are\_behind\_many\_clouds\_today.

// Remove a substring from the middle of the string.

string temp = "many\_";

int i = s.IndexOf(temp);

if (i >= 0)

{

s = s.Remove(i, temp.Length);

}

Console.WriteLine(s);

// Output: Many\_peaks\_are\_behind\_clouds\_today.

// Remove trailing and leading whitespace.

// See also the TrimStart and TrimEnd methods.

string s2 = " I'm wider than I need to be. ";

// Store the results in a new string variable.

temp = s2.Trim();

Console.WriteLine(temp);

// Output: I'm wider than I need to be.

// Keep the console window open in debug mode.

Console.WriteLine("Press any key to exit");

Console.ReadKey();

}

// Custom match method called by Regex.Replace

// using System.Text.RegularExpressions

string ReplaceMatchCase(Match m)

{

// Test whether the match is capitalized

if (Char.IsUpper(m.Value[0]) == true)

{

// Capitalize the replacement string

// using System.Text;

StringBuilder sb = new StringBuilder(replaceWith);

sb[0] = (Char.ToUpper(sb[0]));

return sb.ToString();

}

else

{

return replaceWith;

}

}

}

若要使用数组表示法访问字符串中的各个字符，可以使用 [StringBuilder](https://msdn.microsoft.com/zh-cn/library/system.text.stringbuilder.aspx) 对象，该对象重载 **[]** 运算符以提供对其内部字符缓冲区的访问。也可以使用 [ToCharArray](https://msdn.microsoft.com/zh-cn/library/ezftk57x.aspx) 方法将该字符串转换为一个字符数组。下面的示例使用 **ToCharArray** 创建该数组。然后修改该数组中的某些元素。之后再调用采用一个字符数组作为输入参数的字符串构造函数来创建一个新字符串。

class ModifyStrings

{

static void Main()

{

string str = "The quick brown fox jumped over the fence";

System.Console.WriteLine(str);

char[] chars = str.ToCharArray();

int animalIndex = str.IndexOf("fox");

if (animalIndex != -1)

{

chars[animalIndex++] = 'c';

chars[animalIndex++] = 'a';

chars[animalIndex] = 't';

}

string str2 = new string(chars);

System.Console.WriteLine(str2);

// Keep the console window open in debug mode

System.Console.WriteLine("Press any key to exit.");

System.Console.ReadKey();

}

}

/\* Output:

The quick brown fox jumped over the fence

The quick brown cat jumped over the fence

\*/

下面的示例针对的是一种非常罕见的情况，即您可能希望使用不安全代码以类似于 C 样式字符数组的方式就地修改字符串。此示例演示如何使用 fixed 关键字“就地”访问各个字符。此外还演示对字符串进行不安全操作可能产生的一个副作用，此副作用是由于 C# 编译器在内部存储（暂存）字符串的方式而导致的。通常，除非绝对必要，否则不应该使用这种方法。

class UnsafeString

{

unsafe static void Main(string[] args)

{

// Compiler will store (intern)

// these strings in same location.

string s1 = "Hello";

string s2 = "Hello";

// Change one string using unsafe code.

fixed (char\* p = s1)

{

p[0] = 'C';

}

// Both strings have changed.

Console.WriteLine(s1);

Console.WriteLine(s2);

// Keep console window open in debug mode.

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

}