02 The Basics of Strings

May 12, 2022

1 The Basics of Strings

1.1 What is a String?

A string is a sequence of characters. A handy metaphor is a friendship bracelet, where you string together letters to make a name.

1.2 Strings and String Literals

`firstFriend` and `secondFriend` are variables of strings. The line within `Console.WriteLine` is also a string. It's a **string literal**. A string literal is text what represents a constant string.

Try that out with the following code. Press play and see what comes out.

Next, try changing the variables to see different names.

[1]:
```
string firstFriend = "Maria";
string secondFriend = "Sophia";
Console.WriteLine($"My friends are {firstFriend} and {secondFriend}");
```

```
My friends are Maria and Sophia
```

1.3 String Properties

As you explore more with strings, you'll find that strings are more than a collection of letters. You can find the length of a string using `Length`. `Length` is a **property** of a string and it returns the number of characters in that string.

Try that out by seeing how long the names of the friends are:

[2]:
```
Console.WriteLine($"The name {firstFriend} has {firstFriend.Length} letters.");
Console.WriteLine($"The name {secondFriend} has {secondFriend.Length} letters.
  ↪");
```

```
The name Maria has 5 letters.
The name Sophia has 6 letters.
```

2 String Methods

2.1 Leading and Trailing Spaces

Suppose your strings have leading or trailing spaces (also called **white space**) that you don't want to display. You want to trim the spaces from the strings. The `Trim` method and related methods `TrimStart` and `TrimEnd` do that work. You can just use those methods to remove leading and trailing spaces.

> Play around with trimming in the following code. The brackets are there to help you see all the white space.

[3]:
```
string greeting = "      Hello World!       ";
Console.WriteLine($"[{greeting}]");

string trimmedGreeting = greeting.TrimStart();
Console.WriteLine($"[{trimmedGreeting}]");

trimmedGreeting = greeting.TrimEnd();
Console.WriteLine($"[{trimmedGreeting}]");

trimmedGreeting = greeting.Trim();
Console.WriteLine($"[{trimmedGreeting}]");
```

```
[      Hello World!       ]
[Hello World!       ]
[      Hello World!]
[Hello World!]
```

2.2 Replace

You can also replace substrings with other values. For example, in the code below, you can take "Hello World!" and replace "Hello" with "Greetings", to make "Greetings World!"

> Try it out. What else could you replace "Hello" with?

[4]:
```
string sayHello = "Hello World!";
Console.WriteLine(sayHello);
sayHello = sayHello.Replace("Hello", "Greetings");
Console.WriteLine(sayHello);
```

```
Hello World!
Greetings World!
```

2.3 Changing Case

Sometimes you need your strings to be all UPPERCASE or all lowercase. `ToUpper` and `ToLower` do just that. > The following example seems a bit mixed up. Can you fix it so "whisper" is all lowercase, and "shout" is all uppercase?

[5]:

```
Console.WriteLine("WhiSPer".ToUpper());
Console.WriteLine("sHoUt".ToLower());
```

```
WHISPER
shout
```

[ ]:

```

```