## **About Strings**

The key thing to remember about C# strings is that they are immutable objects representing text as a sequence of Unicode characters (letters, digits, punctuation, etc.). Double quotes are used to define a string instance:

string fruit = "Apple";

Manipulating a string can be done by calling one of its **[methods](https://docs.microsoft.com/en-us/dotnet/api/system.string" \t "https://exercism.org/tracks/csharp/concepts/_blank)** or **[properties](https://docs.microsoft.com/en-us/dotnet/api/system.string" \t "https://exercism.org/tracks/csharp/concepts/_blank)**. As string values can never change after having been defined, all string manipulation methods will return a new string.

A string is delimited by double quote (") characters. Some special characters need **[escaping](https://devblogs.microsoft.com/csharpfaq/what-character-escape-sequences-are-available/" \t "https://exercism.org/tracks/csharp/concepts/_blank)** using the backslash (\) character. Strings can also be prefixed with the at (@) symbol, which makes it a **[verbatim string](https://csharp.net-tutorials.com/data-types/strings/" \l "aelm5298" \t "https://exercism.org/tracks/csharp/concepts/_blank)** that will ignore any escaped characters.

string escaped = "c:\\test.txt";string verbatim = @"c:\test.txt";

escaped == verbatim;*// => true*

If you only need a part of a string, you can use the **[Substring() method](https://docs.microsoft.com/en-us/dotnet/api/system.string.substring" \t "https://exercism.org/tracks/csharp/concepts/_blank)** to extract just that part:

string sentence = "Frank chases the bus.";string name = sentence.Substring(0, 5);*// => "Frank"*

The **[IndexOf() method](https://docs.microsoft.com/en-us/dotnet/api/system.string.indexof" \t "https://exercism.org/tracks/csharp/concepts/_blank)** can be used to find the index of the first occurence of a string within a string, returning -1 if the specified value could not be found:

"continuous-integration".IndexOf("integration")*// => 11*

"continuous-integration".IndexOf("deployment")*// => -1*

Finally, there are **[many ways to concatenate a string](https://docs.microsoft.com/en-us/dotnet/csharp/how-to/concatenate-multiple-strings" \t "https://exercism.org/tracks/csharp/concepts/_blank)**. The simplest one is by using the **[+ operator](https://csharp.net-tutorials.com/data-types/strings/" \l "aelm5211" \t "https://exercism.org/tracks/csharp/concepts/_blank)**.

string name = "Jane";"Hello " + name + "!";*// => "Hello Jane!"*

For any string formatting more complex than simple concatenation, **[string interpolation](https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/string-interpolation" \t "https://exercism.org/tracks/csharp/concepts/_blank)** is preferred. To enable interpolation in a string, prefix it with the dollar ($) symbol.

string name = "Jane";$"Hello {name}!";*// => "Hello Jane!"*