

What is .NET?

Learn about the .NET platform that powers C# applications

If you're learning C#, you're going to encounter a lot of "dot net": .NET, .NET Framework, .NET Core, dotnet, ASP.NET, the list goes on. This article will demystify these concepts and give you some context before diving into C# programming.

What is .NET?

Formally, .NET is "an open source developer platform, created by Microsoft, for building many different types of applications. You can write .NET apps in C#, F#, Visual C++, or Visual Basic."

Informally, .NET is the tool that lets you build and run C# programs (we'll avoid F#, Visual C++, Visual Basic for now).

When you download .NET, you're really downloading a bunch of programs that:

- Translate your C# code into instructions that a computer can understand

- Provide utilities for building software, like tools for printing text to the screen and finding the current time

- Define a set of data types that make it easier for you to store information in your programs, like text, numbers, and dates

There are a few versions of .NET. They do the same job but they are meant for different operating systems:

.NET Framework is the original version of .NET that only runs on Windows computers.

.NET Core is the new, cross-platform version of .NET that runs on Windows, MacOS, and Linux computers. Since it's more flexible and [Microsoft is actively enhancing this version](#), we'll be using this one throughout the path.

Whenever we refer to ".NET" from now on, we really mean .NET Core.

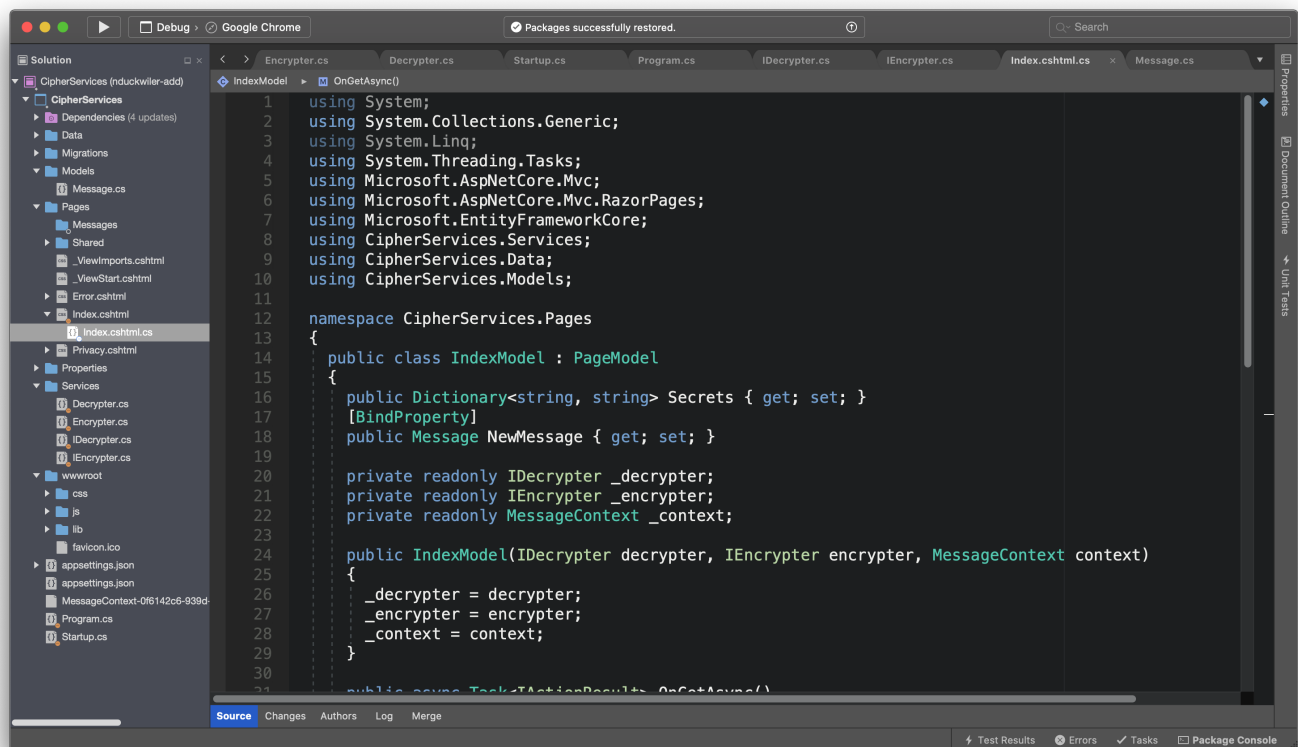
Conveniently, [Microsoft itself plans to rename .NET Core to .NET](#) towards the end of 2020.

How do I get .NET?

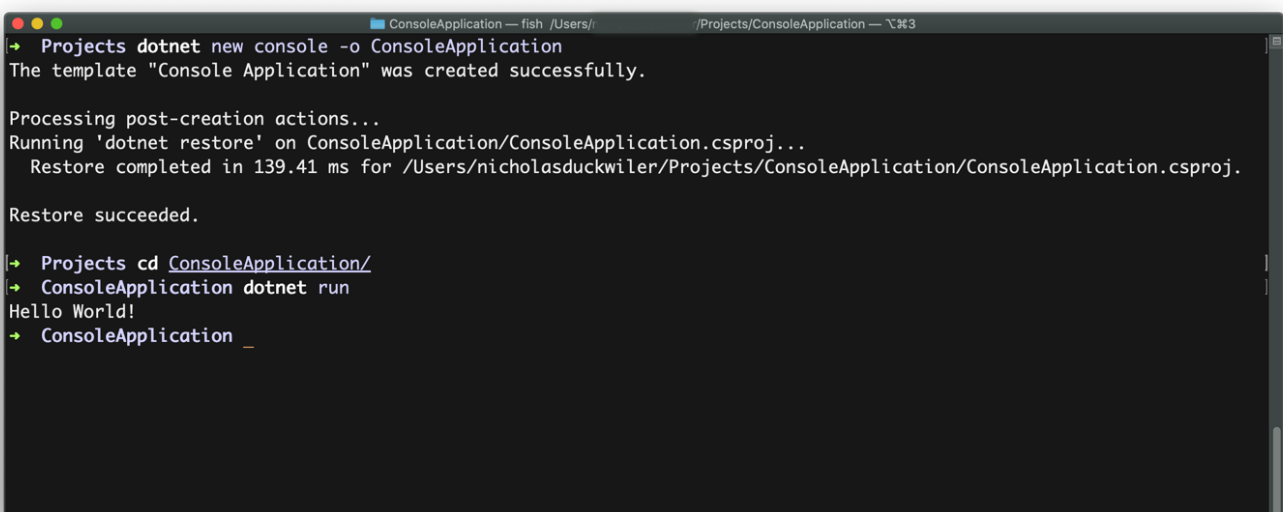
So we know we want to work with C#, and we know we need .NET. How do we get the .NET platform on our computers?

The two main ways are:

Download [Visual Studio](#), an integrated development environment (IDE) for .NET applications. It comes as an app, like the web browser you're using to view this article. It comes with the .NET platform, a code editor, and additional tools to help you write code.



Download the [.NET SDK](#) (software development kit). It also comes with the .NET platform, but it has no code editing tools. Instead of an app on your computer, the SDK is accessed via a command-line interface (CLI) — to use an SDK, you'll open up a terminal on your computer and type commands instead of clicking buttons. In this example, you can see a terminal in which the user ran commands like `dotnet new` and `dotnet run` to build a new application, then ran it (the app just prints out "Hello World!").



You'll see both of these throughout the skill path.

What about ASP.NET?

If we want to build programs specifically for the web, like websites, you'll need to add more tools on top of .NET. One of the most popular is ASP.NET. To keep them separate, developers call .NET a *platform* and ASP.NET a *framework*. This is all you need to know about ASP.NET to get started with C#: we'll cover the details in a separate article.

Summary

Hopefully, this helps you see the big picture before diving into the details of C#:

- C# is a programming language

- To build programs with C#, you need the .NET platform

- .NET comes in two major flavors, .NET Framework for Windows and .NET Core for Windows/macOS/Linux

- To get .NET Core on your computer, download Visual Studio or the .NET SDK

- To build programs like web apps and web services, you need the ASP.NET framework on top of .NET

Now let's get coding!