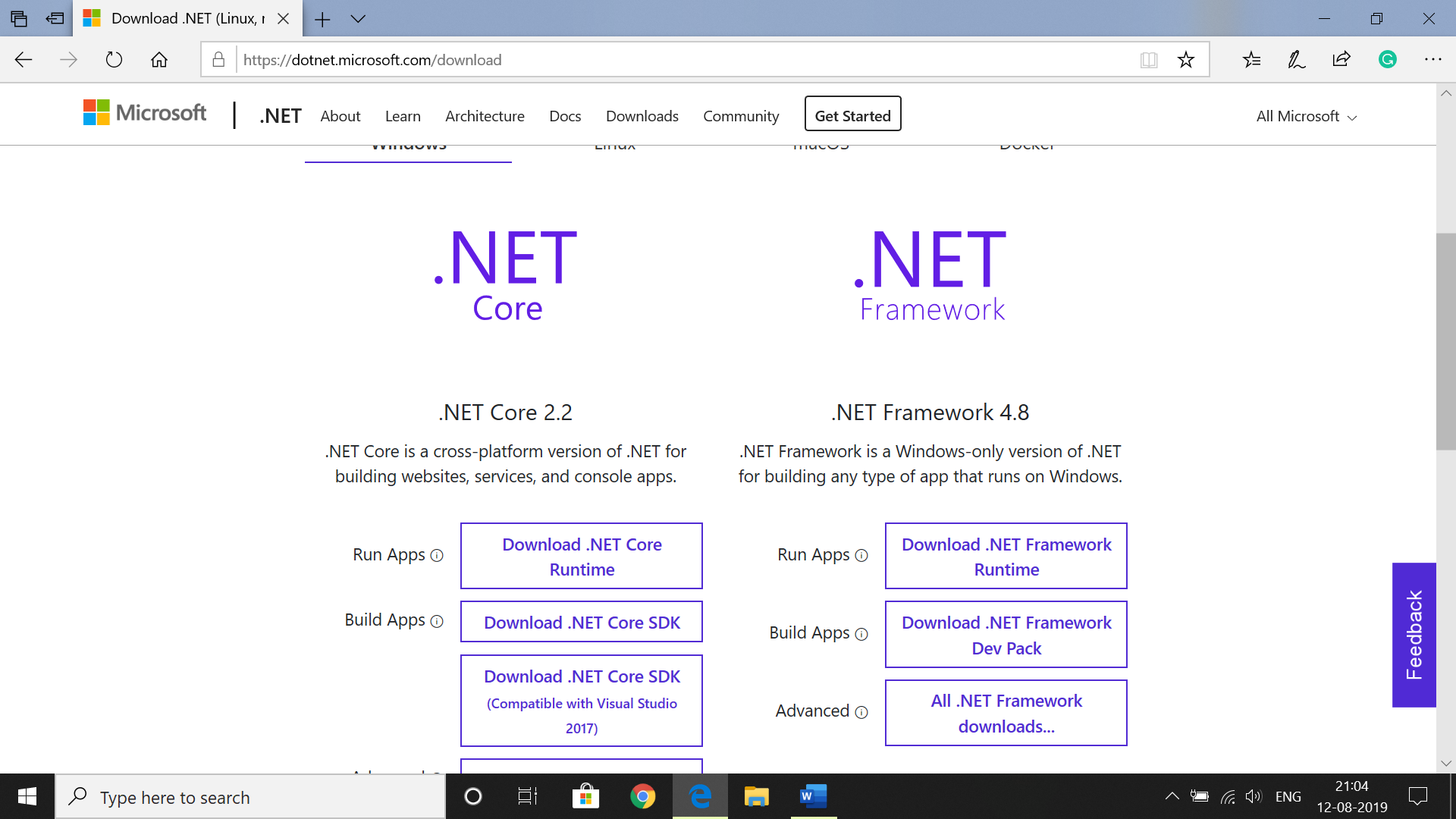
# Chapter 1 - .Net Core

Microsoft released .Net 1.0 in 2002. After that, there are many releases.

In 2016 Microsoft released a brand new .Net framework called .Net Core. Unlike the previous version .Net core runs on Windows, Mac, and Linux. .Net Core is lightweight, portable which can be embedded with the application, which makes it easier to deploy in a container.

## Installation

Head over to http://dot.net and download .Net Core SDK for your platform. Once installed run dotnet --help on command prompt. You will see a bunch of options. .Net core has a new command-line tooling system called .Net Core CLI.

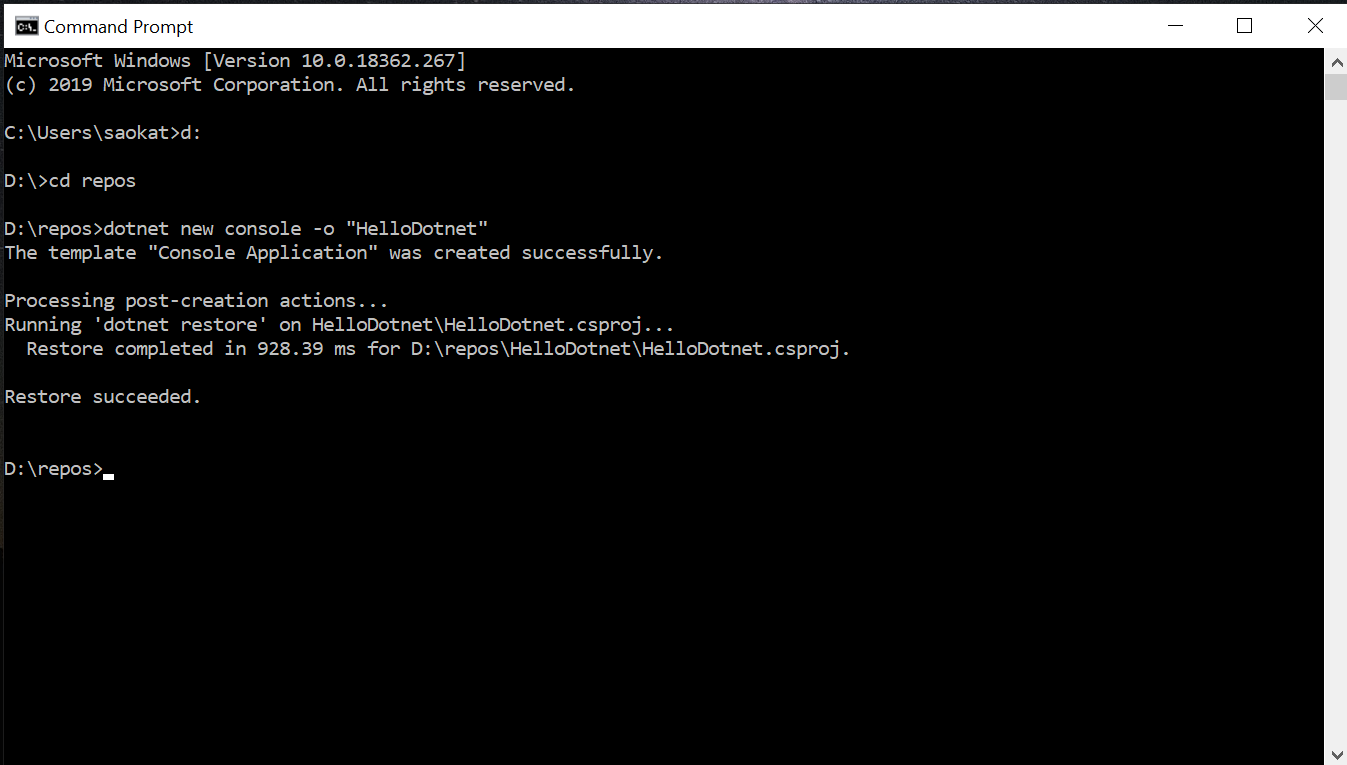


**Tools**

You can use .Net core CLI and an editor of your choice to develop .Net core App. VS Code is the most popular editor which runs on all platform. If you are on Windows, check the free Visual Studio Community Studio. In this book, we are going to use the VS Code and .Net core CLI.

**.Net core CLI**

.Net core CLI is a new Command-line Interface to build .Net core App. Run the following command to create a new Console App.



The -o flag specifies the output directory. We can create other types of App Lib, Web, API, etc.

**.Net Standard**

With the multiple .Net Implementations (.Net Framework, .Net Core, Mono, etc.), we need a standard API specification for code portability. .Net Standard is an API specification, which has multiple versions. Each version supports a set of APIs. For example, .Net Standard 2.0 has greater APIs than 1.6. Each .Net Core 2+ supports .Net Standard 2.0.

**CLR**

Common Language Runtime is the execution environment for all .net languages. When we compile a C# code (or other .net languages), the compiler generates an intermediate bytecode called IL (Intermediate Language). The CLR has JIT compiler which complies IL to platform-specific machine code. The CLR also provides automatic memory allocation and deallocation (garbage collector), threading, exception handling, code access security, etc.