**.NET Core** is a free open source, a general-purpose development platform for developing modern cloud-based software applications on Windows, Linux, and macOS operating systems. It operates across several platforms and has been revamped to make .NET fast, scalable, and modern. .NET Core is one of Microsoft’s big contributions and released under the MIT License. It offers the following features:

* Cross-Platform
* Open Source
* High Performance
* Multiple environments and development mode etc.

**.NET** is a software **framework** that is designed and developed by Microsoft. The first version of the .Net framework was 1.0 which came in the year 2002. In easy words, it is a virtual machine for compiling and executing programs written in different languages like C#, VB.Net, etc. It is used to develop form-based applications, web-based applications, and web services. There is a variety of programming languages available on the .Net platform, VB.Net, and C# being the most common ones. It is used to build applications for Windows, mobile, web, etc. It provides a lot of functionalities and also supports industry standards.

.NET Framework supports more than 60 programming languages in which 11 programming languages are designed and developed by Microsoft. The remaining Non-Microsoft languages which are supported by .NET Framework but not designed and developed by Microsoft.

**Below is a table of differences between .Net Core and .Net Framework:**

| **BASED ON** | **.NET Core** | **.NET Framework** |
| --- | --- | --- |
| **Open Source** | .Net Core is an open source. | Certain components of the .Net Framework are open source. |
| **Cross-Platform** | Works on the principle of “build once, run anywhere”. It is compatible with various operating systems — Windows, Linux, and Mac OS as it is cross-platform. | .NET Framework is compatible with the windows operating system. Although, it was developed to support software and applications on all operating systems. |
| **Application Models** | .Net Core does not support desktop application development and it rather focuses on the web, windows mobile, and windows store. | .Net Framework is used for the development of both desktop and web applications as well as it supports windows forms and WPF applications. |
| **Installation** | .NET Core is packaged and installed independently of the underlying operating system as it is cross-platform. | .NET Framework is installed as a single package for Windows operating system. |
| **Support for Micro-Services and REST Services** | .Net Core supports the development and implementation of micro-services and the user has to create a REST API for its implementation. | .Net Framework does not support the development and implementation of microservices but it supports the REST API services. |
| **Performance and Scalability** | .NET Core offers high performance and scalability. | .Net Framework is less effective in comparison to .Net Core in terms of performance and scalability of applications. |
| **Compatibility** | .NET Core is compatible with various operating systems — Windows, Linux, and Mac OS. | .NET Framework is compatible only with the Windows operating system. |
| **Android Development** | .NET Core is compatible with open-source mobile application platforms, i.e. Xamarin, through the .NET Standard Library. Developers use Xamarin’s tools to configure the mobile app for specific mobile devices such as iOS, Android, and Windows phones. | .NET Framework does not support any framework for mobile application development. |
| **Packaging and Shipping** | .Net Core is shipped as a collection of Nugget packages. | All the libraries of .Net Framework are packaged and shipped together. |
| **Deployment Model** | Whenever the updated version of .NET Core gets initiated; it is updated instantly on one machine at a time, thereby getting updated in new directories/folders in the existing application without affecting it. Thus, .NET Core has a good and flexible deployment model. | In the case of .Net Framework, when the updated version is released it is first deployed on the Internet Information Server only. |
| **Support** | It has support for microservices. | It does not support creation and microservices. |
| **WCF Services** | It has no support for WCF services. | It has excellent support for WCF services. |
| **Rest APIs** | Supports Rest APIs | It also supports REST Services. |
| **CLI Tools** | .NET Core provides light-weight editors and command-line tools for all supported platforms. | .Net Framework is heavy for Command Line Interface and developers prefer to work on the lightweight Command Line Interface. |
| **Security** | .NET Core does not have features like Code Access Security. | Code access security feature is present in .NET Framework. |

**What to use .NET Framework or .NET Core?**

.NET Core is to be used for the server application when –

1. There is cross-platform need.
2. Targeting microservices or using Docker containers.
3. Need of high performance and scalable systems.
4. Side by side need of .NET versions per application.

.NET Framework is to be used for the server application when –

1. Application is to be built to run only on Windows.
2. If application uses .NET framework technologies not available for .NET Core.
3. Applications that are already running on .NET Framework.

**Conclusion-**

If the application has to be built from scratch, choose .Net Core but, if it is already existing good in the .Net Framework and you are not planning to migrate it, then it is a better one.