# Introduction to ASP.Net Core

# Overview

What is ASP.Net Core?

ASP.Net and ASP.Net Core

Why should I choose and use ASP.NET Core?

Types of Applications can be Developed on top of ASP.Net Core

# What is ASP.Net Core?

ASP.Net (Active Server Pages .Net) core is a free and open-source web framework used to develop the next generation of ASP.NET applications.

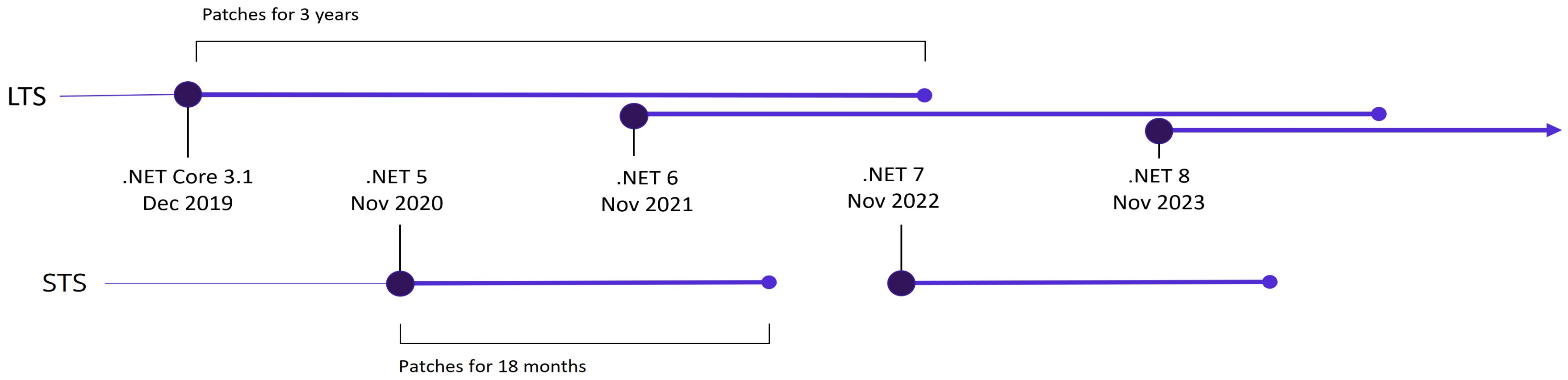
ASP.NET Core is developed and powered by Microsoft as well as the community user base.

It is a modular Framework that has the capability to run on legacy .NET Framework that runs on Windows as well as on cross platform systems.

Originally it was called asp.net next version and then later it was supposed to be called asp.net 5 when it is ready. However, Microsoft later changed the name to ASP.NET core during the first release.

# ASP.Net and ASP.Net Core

| ASP.NET Core | ASP.NET |
|---|---|
| Build for Windows, macOS, or Linux | Build for Windows |
| Razor Pages is the recommended approach to create a Web UI with ASP.NET Core 2.0. See also MVC and Web API | Use Web Forms, Signal R, MVC, Web API, or Web Pages |
| Multiple versions per machine | One version per machine |
| Develop with Visual Studio, Visual Studio for Mac, or Visual Studio Code using C# or F# | Develop with Visual Studio using C#, VB, C#,F# |
| Higher performance than ASP.NET | Good performance |
| Choose .NET Framework or .NET Core runtime | Use .NET Framework runtime |



- **LTS:** Long Term Support: Long-term support refers to the extended maintenance and updates provided for a software, system, or product, ensuring its stability, security, and functionality over an extended period of time

- **STS:** Standard Term Support: typically refers to the regular or default period during which a software, system, or product receives maintenance, updates, and technical assistance, which may be shorter than long-term support but still sufficient for most users' needs.

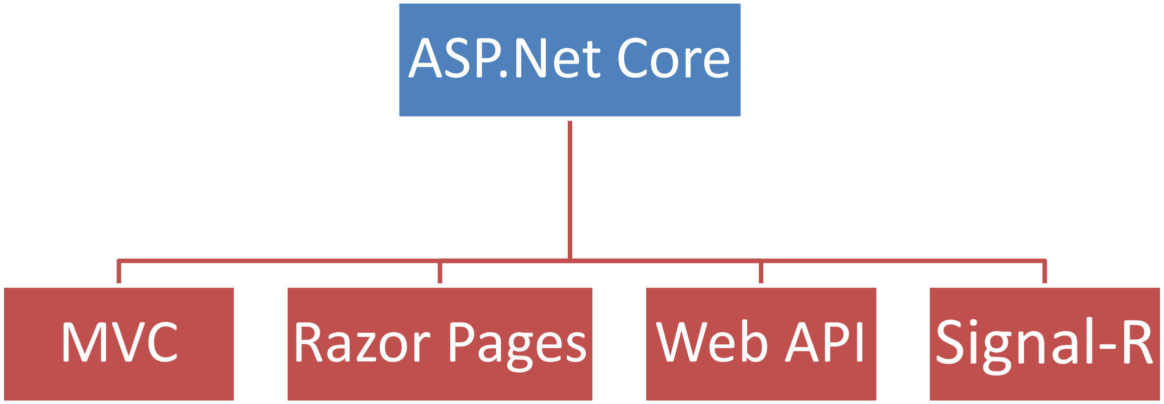
# Why should I Choose and use ASP.Net Core?

- **Cross-platform:** ASP.NET Core is designed to be cross-platform, meaning you can run it on Windows, macOS, and Linux, allowing for greater flexibility and deployment options.
- **High performance:** ASP.NET Core is optimized for performance, offering faster response times and better scalability, making it suitable for high-traffic and resource-intensive applications.
- **Open-source:** ASP.NET Core is open-source, which means you have access to the source code, can contribute to the framework, and benefit from a large community of developers and contributors. https://github.com/dotnet/aspnetcore
- **Modern architecture:** ASP.NET Core follows a modular and lightweight architecture, enabling you to choose and use only the components you need, reducing overhead and making development more efficient.

- **Cloud-ready:** ASP.NET Core is well-integrated with cloud platforms like Microsoft Azure, making it easier to deploy and scale applications in cloud environments.
- **Support for modern web standards:** ASP.NET Core includes support for the latest web standards and protocols, ensuring better security, performance, and compatibility with modern browsers.
- **Built-in dependency injection:** ASP.NET Core has built-in support for dependency injection, promoting cleaner and more maintainable code and facilitating unit testing.
- **Unified MVC and Web API frameworks:** ASP.NET Core provides a unified framework for building both web applications and web APIs, streamlining development and reducing the learning curve.

- **Active community and documentation:** The ASP.NET Core community is vibrant, and there are extensive official and community-driven documentation and resources available to assist with development.
- **Long-term support:** Microsoft typically provides long-term support for major releases of ASP.NET Core, ensuring you receive updates and security fixes for an extended period.
- **Hosting independence:** IIS, Apache, Docker, Self-Hosted in your own proc.
- **Support for JSON based configuration.**
- **Unit testing with xunit.net.**
- **Nuget based support.**

# Types of Applications can be Developed on top of ASP.Net Core



- **MVC (Model-View-Controller) Applications:** ASP.NET Core supports the MVC architectural pattern, allowing developers to build interactive web applications where the logic is separated into models, views, and controllers. This is a common approach for building dynamic and data-driven web applications.

- **Razor Pages Applications:** Razor Pages is a feature of ASP.NET Core that allows developers to build web applications with a simpler page-focused model. It's suitable for scenarios where you want to build web pages with minimal ceremony.
- **Web API Applications:** ASP.NET Core can be used to create RESTful APIs that serve data and functionality to various clients, including web, mobile, and desktop applications.
- **Signal-R (Real-Time Applications):** ASP.NET Core with SignalR can be used to create real-time applications that enable instant communication and updates between clients and the server. Examples include chat applications, live collaboration tools, and real-time dashboards.