

1. .NET Versions

Pre .NET Core

- .Net framework 1.0: initial release, windows forms, ASP.NET, CLR, ADO, NET. 2002
- .Net framework 2.0: Generics, Anonymous Methods, partial class. 2005
- .Net framework 3.0: added WPF, WCF, WF, CardSpace. 2006
- .Net framework 3.5: LINQ, Lambada expressions, ASP.NET. 2007
- .Net framework 4.0: Parallel computing, MEF, code contracts. 2010
- .Net framework 4.5-4.8: Roslyn compiler integration, TLS updates. 2012-2019

.Net Core Era

- .Net Core 1.0: Modular, cross-platform, CLI tools, NuGet-based. 2016
- .Net Core 2.0: .Net standard support, wider APIs. 2017
- .Net Core 3.0/3.1 LTS: Desktop support, C# 8, Blazor support. 2019

.Net 5 and beyond

- .Net 5: first unified version, dropped core name, c# 9. 2020
- .Net 6: MAUI preview, performance improvements, hot reload. 2021
- .Net 7: Native AOT, C# 11, minimal APIs improvements. 2022
- .Net 8: Unified Web, Cloud and mobile dev, MAUGA. 2021
- .Net 9: Improvements to blazor, MAUI, AI tooling 2024

2. Namespace

- A namespace in .NET is a container that organizes classes, interfaces, enums, structs, and other types into a logical hierarchy. It helps avoid naming conflicts and makes large codebases easier to manage.
- Avoid naming conflicts
- Logical organization of code
- Logical organization of code

3. .Net Core

- .NET Core is a cross-platform, open-source, modular version of the .NET framework, designed for building modern apps on Windows, macOS, and Linux.
- Cross-platform
- High performance
- Modular
- Side-by-side installations
- Open Source
- Modern tooling
- Support for APIs, Microservices, Web, Console, Desktop

4. Solution in .Net

- A Solution (.sln file) in .NET is a container that holds one or more Projects (like APIs, class libraries, etc.).

Benefits of a Solution:

- Organize multi-project systems (UI + API + Data + Tests)
- Maintain clean architecture (e.g., separation of concerns)
- Shared build/config settings
- Easy for IDEs (like Visual Studio) to manage