Dot net versions

.NET Framework

Key Features:

1. Windows-Centric: Designed specifically for Windows-based development.
2. Rich Libraries: Includes APIs for Windows Forms, WPF, and ASP.NET.
3. Version Highlights:
    - **4.5**: Introduced asynchronous programming with async and await.
    - **4.6**: Enhanced debugging and 64-bit JIT compiler.
    - **4.8**: Last major update with improved high DPI support and modern cryptography algorithms.

Usage: Suitable for legacy applications; lacks cross-platform support.

.NET Core

Key Features:

1. Cross-Platform: Runs on Windows, macOS, and Linux.
2. Modular Design: Developers can include only required libraries.
3. Version Highlights:
    - **1.x**: First iteration focusing on lightweight applications.
    - **2.x**: Added API compatibility with .NET Framework and Entity Framework Core.
    - **3.x**: Introduced desktop application support (Windows Forms, WPF) and Blazor.

Usage: Ideal for lightweight, modern, and cross-platform applications.

.NET 5

Key Features:

1. Unified Platform: Combines features of .NET Framework and .NET Core.

2. Performance Boost: Improved garbage collection and JSON serialization.
3. Web and Cloud: Enhanced support for microservices and containerization.

Usage: A significant shift toward unification and modern development.

.NET 6

Key Features:

1. Long-Term Support (LTS): Supported for extended stability.
2. Hot Reload: Allows live code changes without restarting applications.
3. Multi-Platform Development: Full release of .NET MAUI for cross-platform UI.

Usage: Recommended for new projects due to its LTS and modern capabilities.

.NET 7

Key Features:

1. Performance Enhancements: Up to 10% faster execution for some workloads.
2. Improved Developer Experience:
    - Updates to LINQ and pattern matching.
    - Enhanced Docker container support.
3. Continued MAUI Improvements: Faster rendering and better control layouts.

Usage: Best for performance-critical and evolving modern applications.

.NET 8

Key Features:

1. Performance Enhancements:
    - Optimized AOT (Ahead-of-Time) compilation for smaller and faster applications.

    - Significant speed improvements for regular expressions and JSON handling.
2. Native AOT Support: Broader support for Native AOT in console, ASP.NET Core, and class libraries, reducing app size and startup times.
3. Blazor Enhancements:
    - Unified Blazor model for server and WebAssembly applications.
    - Simplified full-stack web app development with better debugging and project templates.
4. Cloud-Native Focus:
    - New tools and APIs for building scalable, containerized cloud applications.
    - Enhanced support for Kubernetes and container orchestration.
5. Cross-Platform Updates:
    - .NET MAUI improvements for building mobile and desktop apps.
    - Improved support for ARM64 devices and macOS.
6. C# 12: Introduced new features such as primary constructors for records and structs, and collection expressions for better LINQ queries.

.NET 9

Key Features:

1. Ahead-of-Time (AOT) Compilation: Significant runtime performance improvements.
2. Modernized Libraries: Updated controls in .NET MAUI and Fluent UI in WPF.
3. C# 13 and F# 9: Latest language features for better productivity and safety.

Usage: Cutting-edge projects seeking the latest capabilities in AI, cloud, and modern app development.