



# C# 11 and .NET 7 - Modern Cross-Platform Development Fundamentals : Start Building Websites and Services with ASP.NET Core 7, Blazor, and EF Core 7, 7th Edition

Price, Mark J.

Monografia

**Publisher's Note:** Microsoft will stop supporting .NET 7 from May 2024. The newer 8th edition of the book is available that covers .NET 8 (end-of-life November 2026) with C# 12 and EF Core. Purchase of the print or Kindle book includes a free PDF eBook.

**Key Features** Explore the newest additions to C# 11, the .NET 7 class libraries, and Entity Framework Core 7. Create professional websites and services with ASP.NET Core 7 and Blazor. Build your confidence with step-by-step code examples and tips for best practices.

**Description** Extensively revised to accommodate the latest features that come with C# 11 and .NET 7, this latest edition of our guide will get you coding in C# with confidence. You'll learn object-oriented programming, writing, testing, and debugging functions, implementing interfaces, and inheriting classes. Next, you'll take on .NET APIs for performing tasks like managing and querying data, working with the filesystem, and serialization. As you progress, you'll also explore examples of cross-platform projects you can build and deploy, such as websites and services using ASP.NET Core. Instead of distracting you with unnecessary graphical user interface code, the first eleven chapters will teach you about C# language constructs and many of the .NET libraries through simple console applications. Having mastered the basics, you'll then start building websites, web services, and browser apps. By the end of this book, you'll be able to create rich web experiences and have a solid grasp of object-oriented programming that you can build upon.

**What you will learn** Build rich web experiences using Blazor, Razor Pages, the Model-View-Controller (MVC) pattern, and other features of ASP.NET Core. Write, test, and debug functions. Query and manipulate data using LINQ. Integrate and update databases in your apps using Entity Framework Core models. Build and consume powerful services using the latest technologies, including Web API and Minimal API.

**For .NET 6 developers:** C# 11 can be used with .NET 6, including features like raw string literals. EF Core 7 targets .NET 6, so you can benefit from its new features like `ExecuteUpdate` and `ExecuteDelete` for more efficient data modifications.

**Who this book is for** This book is primarily for beginners, but intermediate-level C# and .NET programmers who have worked with C# in the past and want to catch up with the changes made in the past few years will also find plenty of useful information in it. Prior exposure to C# or .NET is not a prerequisite, but you should have a general understanding of programming before you jump in. If you already have some C# and .NET skills and want to focus on developing apps, we recommend that you pick up Mark's other .NET book, *Apps and Services with .NET 7*, instead.

**Título:** C# 11 and .NET 7 - Modern Cross-Platform Development Fundamentals Start Building Websites and Services with ASP.NET Core 7, Blazor, and EF Core 7, 7th Edition

**Edición:** 7th ed

**Editorial:** Birmingham Packt Publishing, Limited 2022 2022

**Descripción física:** 1 online resource (819 pages)

**Contenido:** Cover -- Copyright -- Contributors -- Quick Chapter Reference -- Table of Contents -- Preface -- Chapter 1: Hello, C#! Welcome, .NET! -- Setting up your development environment -- Choosing the appropriate tool and application type for learning -- Pros and cons of the .NET Interactive Notebooks extension -- Using Visual Studio Code for cross-platform development -- Using GitHub Codespaces for development in the cloud -- Using Visual Studio for Mac for general development -- Using Visual Studio for Windows for general development -- What I used -- Deploying cross-platform -- Downloading and installing Visual Studio 2022 for Windows -- Microsoft Visual Studio for Windows keyboard shortcuts -- Downloading and installing Visual Studio Code -- Installing other extensions -- Managing Visual Studio Code extensions at the command line -- Understanding Microsoft Visual Studio Code versions -- Microsoft Visual Studio Code keyboard shortcuts -- Understanding .NET -- Understanding .NET Framework -- Understanding the Mono, Xamarin, and Unity projects -- Understanding .NET Core -- Understanding the journey to one .NET -- Understanding Blazor WebAssembly versioning -- Understanding .NET support -- Understanding .NET Runtime and .NET SDK versions -- Listing and removing versions of .NET -- What is different about modern .NET? -- Windows desktop development -- Web development -- Database development -- Understanding .NET Standard -- .NET platforms and tools used by the C# and .NET book editions -- Topics covered by Apps and Services with .NET 7 -- Understanding intermediate language -- Comparing .NET technologies -- Building console apps using Visual Studio 2022 -- Managing multiple projects using Visual Studio 2022 -- Writing code using Visual Studio 2022 -- Compiling and running code using Visual Studio -- Understanding the compiler-generated folders and files Understanding top-level programs -- Implicitly imported namespaces -- Revealing the hidden code by throwing an exception -- Adding a second project using Visual Studio 2022 -- Building console apps using Visual Studio Code -- Managing multiple projects using Visual Studio Code -- Writing code using Visual Studio Code -- Compiling and running code using the dotnet CLI -- Adding a second project using Visual Studio Code -- Exploring code using .NET Interactive Notebooks -- Using .NET Interactive Notebooks for the code in this book -- Reviewing the folders and files for projects -- Understanding the common folders and files -- Understanding the solution code on GitHub -- Making good use of the GitHub repository for this book -- Raising issues with the book -- Giving me feedback -- Downloading solution code from the GitHub repository -- Using Git with Visual Studio Code and the command line -- Cloning the book solution code repository -- Looking for help -- Reading Microsoft documentation -- Getting help for the dotnet tool -- Getting definitions of types and their members -- Looking for answers on Stack Overflow -- Searching for answers using Google -- Subscribing to the official .NET blog -- Watching Scott Hanselman's videos -- A companion book to continue your learning journey -- Practicing and exploring -- Exercise 1.1 - Test your knowledge -- Exercise 1.2 - Practice C# anywhere with a browser -- Exercise 1.3 - Explore topics -- Exercise 1.4 - Explore themes of modern .NET -- Summary -- Chapter 2: Speaking C# -- Introducing the C# language -- Understanding language versions and features -- Project COOL -- C# 1 -- C# 1.2 -- C# 2 -- C# 3 -- C# 4 -- C# 5 -- C# 6 -- C# 7.0 -- C# 7.1 -- C# 7.2 -- C# 7.3 -- C# 8 -- C# 9 -- C# 10 -- C# 11 -- Understanding C# standards -- Discovering your C# compiler versions -- How to output the SDK version Enabling a specific language version compiler -- Switching the C# compiler for .NET 6 -- Understanding C# grammar and vocabulary -- Showing the compiler version -- Understanding C# grammar -- Statements -- Comments -- Blocks -- Examples of statements and blocks -- Understanding C# vocabulary -- Comparing programming languages to human languages -- Changing the color scheme for C# syntax -- Help for writing correct code -- Importing namespaces -- Implicitly and globally importing namespaces -- Verbs are methods -- Nouns are types, variables, fields, and properties -- Revealing the extent of the C# vocabulary -- Working with variables -- Naming things and assigning values -- Literal values -- Storing text -- Verbatim strings -- Raw string literals -- Raw interpolated string literals -- Summarizing options for

storing text -- Storing numbers -- Storing whole numbers -- Exploring whole numbers -- Storing real numbers -- Writing code to explore number sizes -- Comparing double and decimal types -- Storing Booleans -- Storing any type of object -- Storing dynamic types -- Declaring local variables -- Specifying the type of a local variable -- Inferring the type of a local variable -- Using target-typed new to instantiate objects -- Getting and setting the default values for types -- Exploring more about console apps -- Displaying output to the user -- Formatting using numbered positional arguments -- Formatting using interpolated strings -- Understanding format strings -- Getting text input from the user -- Simplifying the usage of the console -- Importing a static type for a single file -- Importing a static type for all code files in a project -- Getting key input from the user -- Passing arguments to a console app -- Setting options with arguments -- Handling platforms that do not support an API -- Understanding async and await Improving responsiveness for console apps -- Practicing and exploring -- Exercise 2.1 - Test your knowledge -- Exercise 2.2 - Test your knowledge of number types -- Exercise 2.3 - Practice number sizes and ranges -- Exercise 2.4 - Explore topics -- Summary -- Chapter 3: Controlling Flow, Converting Types, and Handling Exceptions -- Operating on variables -- Exploring unary operators -- Exploring binary arithmetic operators -- Assignment operators -- Exploring logical operators -- Exploring conditional logical operators -- Exploring bitwise and binary shift operators -- Miscellaneous operators -- Understanding selection statements -- Branching with the if statement -- Why you should always use braces with if statements -- Pattern matching with the if statement -- Branching with the switch statement -- Pattern matching with the switch statement -- Simplifying switch statements with switch expressions -- Understanding iteration statements -- Looping with the while statement -- Looping with the do statement -- Looping with the for statement -- Looping with the foreach statement -- Understanding how foreach works internally -- Storing multiple values in an array -- Working with single-dimensional arrays -- Working with multi-dimensional arrays -- Working with jagged arrays -- List pattern matching with arrays -- Summarizing arrays -- Casting and converting between types -- Casting numbers implicitly and explicitly -- Converting with the System.Convert type -- Rounding numbers -- Understanding the default rounding rules -- Taking control of rounding rules -- Converting from any type to a string -- Converting from a binary object to a string -- Parsing from strings to numbers or dates and times -- Errors using Parse -- Avoiding exceptions using the TryParse method -- Handling exceptions -- Wrapping error-prone code in a try block -- Catching all exceptions Catching specific exceptions -- Catching with filters -- Checking for overflow -- Throwing overflow exceptions with the checked statement -- Disabling compiler overflow checks with the unchecked statement -- Practicing and exploring -- Exercise 3.1 - Test your knowledge -- Exercise 3.2 - Explore loops and overflow -- Exercise 3.3 - Practice loops and operators -- Exercise 3.4 - Practice exception handling -- Exercise 3.5 - Test your knowledge of operators -- Exercise 3.6 - Explore topics -- Summary -- Chapter 4: Writing, Debugging, and Testing Functions -- Writing functions -- Understanding top-level programs and functions -- Times table example -- Writing a times table function -- A brief aside about arguments and parameters -- Writing a function that returns a value -- Converting numbers from cardinal to ordinal -- Calculating factorials with recursion -- Documenting functions with XML comments -- Using lambdas in function implementations -- Debugging during development -- Using the Visual Studio Code integrated terminal during debugging -- Creating code with a deliberate bug -- Setting a breakpoint and starting debugging -- Using Visual Studio 2022 -- Navigating with the debugging toolbar -- Using Visual Studio Code -- Debugging windows -- Stepping through code -- Customizing breakpoints -- Hot reloading during development -- Hot reloading using Visual Studio 2022 -- Hot reloading using Visual Studio Code and the command line -- Logging during development and runtime -- Understanding logging options -- Instrumenting with Debug and Trace -- Writing to the default trace listener -- Configuring trace listeners -- Switching trace levels -- Adding packages to a project in Visual Studio 2022 -- Adding packages to a project in Visual Studio Code -- Reviewing project packages -- Logging information about your source code -- Unit testing Understanding types of testing

**ISBN:** 9781803248950 1803248955

**Materia:** Application software-Development Web sites-Authoring programs

**Enlace a formato físico adicional:** Print version Price, Mark J. C# 11 and . NET 7 - Modern Cross-Platform Development Fundamentals Birmingham : Packt Publishing, Limited,c2022

- Gran Vía, 59 28013 Madrid
- (+34) 91 456 03 60
- [informa@baratz.es](mailto:informa@baratz.es)