

Mis à jour le 30/12/2024

S'inscrire

Formation C# 12 avec .NET 8 et Visual Studio 2022

UNE MISE À NIVEAU RAPIDE POUR TOUS LES DÉVELOPPEURS

5 jours (35 heures)

Présentation

C# est indéniablement l'un des langages de programmation les plus complets à la disposition des ingénieurs aujourd'hui, couplé à la puissance de .NET 8. Cette formation a été spécialement conçue pour mettre à niveau rapidement tous les développeurs sur les fondamentaux ainsi que les nouveautés du Framework .NET 8 avec Visual Studio 2022 et le langage C# en version 12.

Dans cette formation polyvalente conçue comme un véritable "couteau suisse" du .NET, vous aborderez les principaux concepts de C# afin de gagner en autonomie sur la plateforme Microsoft .NET 8. **Cette remise à niveau, destiné à tous les programmeurs, vous permettra d'acquérir toutes les bases solides afin de vous rendre autonome dans vos futurs développements.** Cette formation se veut à la fois ludique, pédagogique et interactive.

Dans un premier temps, nous passerons en revue tous les concepts de base de la POO au travers du langage moderne qu'est le C#, avec des exemples de la vie réelle. Nous aborderons ensuite toutes les subtilités de .NET ainsi que ses nouveautés sur les dernières années, de C# 8 à C#12, et plus encore.

Nous verrons le bon usage des types, des delegates, ainsi que des cas de figure portant sur la programmation concurrente et asynchrone, afin que vous puissiez répondre aux questions techniques les plus poussées. Enfin, nous aborderons le dernier-né des Frameworks Web Front : Blazor, en exposant nos propres Web API et en maîtrisant efficacement la persistance de données.

Comme toujours, nous vous enseignerons la dernière version de l'outil à savoir **C# 13** avec .NET 8 Core Version avec **Visual Studio 2022** sortie le 8 novembre 2021.

Objectifs

- Maîtriser les principaux concepts de programmation orientée objet en C#
- Découvrir les nouveautés ainsi que les dernières évolutions de C# et .NET au travers de Visual Studio 2022
- Acquérir une autonomie dans le développement, le débogage et la compréhension de la documentation en .NET
- Connaître les fondamentaux du nouveau Framework Web Blazor et des applications .Net 8
- Construire votre Web API via ASP.NET Core et accéder facilement à une base de données
- Implémenter des tests
- Initier une démarche clean code

Public visé

- Développeurs issus de n'importe quel langage de programmation
- Chefs de projets techniques

Pré-requis

- Connaissance pratique d'un langage de programmation récent
- Des connaissances basiques sur le Framework .NET est un plus
- [Tester Mes Connaissances](#)

PRÉ-REQUIS LOGICIELS

- [Visual Studio Community](#) installé (pour créer des applications modernes pour Android, iOS, Windows, ainsi que des applications web et des services cloud)
- Un compte Microsoft (pour accéder à Visual Studio)
- Installer à minima le SDK pour .Net 8

RECOMMANDATIONS DE LECTURE AVANT ET APRÈS LA FORMATION

- Je vous recommande les blogs suivant :
 - [Blog de Thomas Levesque's](#) pour des astuces et des bonnes pratiques en C#.
 - [Blog Ardalís de Steve Smith](#) qui propose des articles sur le développement logiciel, les pratiques de codage et les technologies Microsoft dont C#.
- [La documentation officielle de C#](#) : elle contient des informations détaillées sur le langage C# et ses fonctionnalités.
- Les vidéos de la [chaîne YouTube de Microsoft Developer](#) : elle propose des tutoriels vidéo pour les développeurs, y compris des [vidéos sur C#](#) et [Visual Studio](#).
- Les forums de développeurs C# : les forums de développeurs C# peuvent être utiles pour obtenir de l'aide et des conseils sur la programmation en C# comme celui de [Developpez.net](#).

Programme de notre formation C# 12 avec Visual Studio 2022

Introduction : Qu'est-ce que .NET en 2024 ?

- À qui s'adresse .NET ?
- Les principales évolutions du Framework depuis sa création
- Le futur de .NET
- Architecture
 - Managed Code
 - Le IL / CLR / Jit
- Langage Interprété VS Langage Compilé
- L'IDE Visual Studio Versus L'éditeur Visual Studio Code Versus Rider
- Ne pas confondre IDE vs Langage vs Framework .NET
- Évolutions & Principaux ajouts du langage C#
- Le rythme des versions LTS & Current et leurs supports respectifs par Microsoft
- .NET 8 :
 - .NET Aspire
 - Améliorations apportées
 - Prise en charge native d'AOA
 - Globalisation
 - Conteneurs
 - NuGet
 - Compilation AOT

Présentation rapide de l'écosystème .NET

- Core Versus Mainstream Versus Standard
- Tableau des versions du langage par version de framework
- Les différents Frameworks spécialisés :
 - WPF
 - Entity
- La documentation MSDN
- GitHub
- Git + courte introduction

Rappel sur les bases de Visual Studio : les fondamentaux

- Solution
- Projet / Shared Project
- Assembly
- DLL
- Nuget Package
- Ajout d'un projet à une solution existante
- Référence à un projet à partir d'un autre projet
- Référencement des bibliothèques externes
- Espaces de nommage
- Raccourci Clavier

Rappel sur les bases du C# : les fondamentaux

- Paradigme & Typage
- Les types Primitifs
- Les membres d'une classe
- Encapsulation et niveau de visibilité

- Structures vs Classes vs Record
- Les tuples
- Les usings
- Les constructeurs et les destructeurs
- Les types génériques
- Opérateurs et expressions
- Faire la différence entre les 2 univers : statique et instance
- Faire la différence entre les 2 univers : managé et non managé

Debug & Breakpoint

- Débogage d'une application en cours de développement
 - Réglage d'un point d'arrêt
 - Mettre une condition à un point d'arrêt
- Modification d'une variable en temps réel
- Enregistrement pendant le développement et l'exécution
 - Instrumentation avec débogage et traçage
 - Écrire dans l'écouteur de trace par défaut
 - Configuration des auditeurs de trace

Introduction à la programmation objet : rappel des fondamentaux de la POO en C#

- La programmation orientée objet : c'est simple au fond !
- Procédural vs POO
- Les grands principes fondateurs
 - Encapsulation
 - Héritage
 - Polymorphisme

Introduction aux bonnes pratiques du développeur

- Convention de nommage et règles de programmation
- Responsabilité
- Factorisation
- Les Principes de conception SOLID
- Dry/Yagni/Kiss/Wet/....
- Intention / Nommage
- Cycle de validation

Codage de base en C#

- Variables locales
- Static / const
- Déclarations et expressions
- Déclarations
- Expressions
- Booléens

- Commentaires
- Directives de prétraitement
- Symboles de compilation
- Type value et reference

Structures de données et écriture de code optimisé en C#

- LIFO
- FIFO
- Choisir la bonne structure de données pour optimiser les performances
 - Tableaux
 - Listes
 - Piles
 - Queue d'attente
 - Dictionnaires, tables de hachage et hashsets
 - Listes génériques
- Meilleures pratiques dans l'écriture de code optimisé en C#
 - Box et unboxing
 - Concaténation de chaînes de caractère
 - Traitement des exceptions
 - Instructions while, for & foreach
 - Délégués

Les classiques du C#

- Utiliser et implémenter des classes abstraites / virtual / Partial
- Utilisation et implémentation d'interfaces
- Différences entre une classe abstraite et une interface : dans quels cas utiliser l'une plutôt que l'autre ?
- Mutable vs Immutable
- Génériques
- Type nullable
- Type dynamique
- Surcharge d'opérateur : savoir définir l'égalité
- Les delegates
- Try / Catch / Finally
- Serialisation

Introduction aux design pattern

- Les différents types de Design pattern (patron de conception) / GoF
 - Création
 - Structuraux
 - Comportementaux
- Implémentation du Factory
- Implémentation du Decorator
- Implémentation de l'Observer

Introduction aux tests

- Différents types de test
 - Pyramide
 - Quadrant
- Comment tester et pourquoi ?
- Mock...
- TDD / BDD

Fonctionnalités de C# 8 à C# 11

- Les plus utiles par version

LE FUTUR : LES NOUVELLES FONCTIONNALITÉS DE C# 12

Introduction à la programmation concurrente et asynchrone

- Concurrence et Threading
- Cas d'utilisation du multi-threading
- Thread et Task
- Synchronisation et communication
- ThreadPool et efficacité
- Deadlocks et autres écueils à éviter

Introduction à la programmation asynchrone

- Différence entre appels synchrones et asynchrones
- Async opérations
- async et await
- Gérer la Progression et l'Abandon
- CancellationToken

Réflexion et Attributes

- Introspection des assemblies et classes
- Les types d'Attributes
- Paramètres des Attributes
- Génération de code : Emitters

PROJET .NET 7/8

- Utilisation Startup / program
- IoC & Co
- Middleware
- Environnement Variable + section
- Secrets

Les bases du Web

- Fonctionnement du Web
- HTTP/2 et HTTP/3
- Les cookies
- Le modèle MVC
- Validation de model
- Différentes architectures
 - N-Tiers
 - Onion Aka Clean Archi Aka Hexagonal Aka Port/adapter
 - Vertical Slice

Le Web avec Blazor (WebAssembly)

- Blazor Server vs Blazor client (WASM)
- NavBar HTML CSS
- Model
- Page
- HotReload
- CRUD

Web API / Minimal API

- Concevoir sa première API d'accès
- Controller
- Consommer l'API depuis le client

Persistence

- Utiliser le DbContext
- Génération automatique de la base de données via le Code-First
- CRUD
- Introduction aux requêtes LINQ
- Migration

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant

d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.