

Durée : 5 jours soit 35 heures

Référence : IF-C#FON

Public visé :

- Développeurs
- Ingénieurs de développement

Pré-requis :

- Pratique d'un langage informatique
- Connaissances générales en programmation

Objectifs pédagogiques :

- Comprendre un programme C#
- Concevoir, implémenter et organiser des objets
- Choisir les types et collections adaptées à un problème
- Interroger des collections en LinQ
- Respecter les conventions et bonnes pratiques du langage

Compétences acquises à l'issue de la formation :

- Comprendre un programme C#
- Concevoir, implémenter et organiser des objets
- Choisir les types et collections adaptées à un problème
- Interroger des collections en LinQ
- Respecter les conventions et bonnes pratiques du langage

Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure.

La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz.

L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration.

En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Moyens et supports pédagogiques :

**Cadre présentiel**

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection sur tableau blanc
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique

**Cadre téléprésentiel**

Session dispensée via notre solution iClassroom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

Informations sur l'accessibilité :



## Description / Contenu

### Module 1 : Prise en main de C#

- Relation de C# et .Net
- Intégration dans Microsoft Visual Studio
- Première application C#
- Exécution et débogage
- Mode natif : Native AOT

### Module 2 : Types et instructions

- Variables : Déclaration/initiaisation
- Opérateurs, conversions simples
- Tests et boucles
- Tableaux uni. et multidimensionnels
- Tuples et tuples nommés
- Types références et types valeurs
- Gestion du null : ?, ??, ?.

### Module 3 : Structuration d'un programme

- Enumérations : enum
- Attributs / annotations
- Méthodes : Définition, retour, appel, paramètres
- Paramètres avancés : out, ref, readonly ref, params
- Conversions complexes
- Exceptions : Déclenchement, traitement
- Documentation intégrée
- Architectures MVC, MVVM

### Module 4 : Encapsulation

- Objets : class, struct, record
- Champs, méthodes et propriétés
- Modificateurs : private, public, static, init, ...
- Gestion du null : required, !, attributs spécifiques

### Module 5 : Gestion des dépendances

- Espace de nom
- Bibliothèque de classe, références
- Gestionnaire NuGet
- Découplage par interface : définition, implémentation
- "Garbage Collection" et IDisposable

### Module 6 : Collections génériques

- Collections usuelles : List, Dictionary
- Interfaces : IEnumerable, ICollection, IList
- Performances : StringBuilder, yield return, stackalloc, Span
- Conception d'un type générique : covariance/contravariance

### Module 7 : LinQ

- Opérations élémentaires
- Expressions lambda
- Filtres et agrégations

### Module 8 : Syntaxes avancées

- Notation asynchrones : async, await
- Délégués et événements
- Redéfinition d'opérateurs, indexeurs
- Classes abstraites
- Programmation dynamique par réflexion

### Modules 9 : Tests

- Projet de test unitaire
- Cas de tests usuels et extrêmes
- Assertions usuelles
- Vérification des exceptions