

## Formation **GitHub Copilot pour développeurs**

Améliorer la productivité et la qualité du code grâce à la programmation assistée par l'IA

Plongez dans l'avenir du développement logiciel avec notre formation GitHub Copilot. Conçue pour les développeurs désireux d'améliorer leur efficacité et leur créativité, cette formation complète allie des connaissances de pointe en matière d'IA à une expérience pratique et concrète. Grâce à des sessions de formation animées par des experts, vous maîtriserez GitHub Copilot, de la configuration de base à l'intégration avancée, transformant ainsi votre façon de coder.

### Durée

1 jour

### Objectifs pédagogiques

- ◆ Analyser les principales fonctionnalités et les avantages de GitHub Copilot pour optimiser les processus de développement
- ◆ Appliquer GitHub Copilot dans des scénarios de programmation réels
- ◆ Évaluer l'efficacité de GitHub Copilot dans divers contextes de codage, en assurant la qualité et la pertinence du code
- ◆ Créer des configurations personnalisées et utiliser les fonctionnalités avancées de GitHub Copilot pour adapter son assistance aux besoins spécifiques d'un projet

### Public

Développeurs, chefs de projet

### Prérequis

Maîtrise de base d'au moins un langage de programmation (par exemple, JavaScript, Python, Java).

Compréhension des processus de développement de logiciels et du contrôle de version, en particulier Git et GitHub.

Familiarité avec les environnements de développement intégrés (IDE) et les éditeurs de code.

## Programme de formation

### Introduction à GitHub Copilot et configuration

Présentation de GitHub Copilot.  
Comprendre les usages clés dans un contexte de développement logiciel.  
Installation et configuration dans différents environnements de développement.  
Démarrer avec GitHub Copilot : commandes de base et navigation dans l'interface.  
Exemples d'activités pratiques : installation de GitHub Copilot, génération de premiers bouts de code, exploration de l'interface utilisateur.

### Coder avec GitHub Copilot

Exploration des fonctionnalités de base : auto-complétion, génération de code et documentation inline.  
Gestions des suggestions : accepter, modifier et rejeter les recommandations.  
Explorer des documentations pour comprendre de nouvelles technologies.  
Exemples d'activités pratiques : création d'une application simple en utilisant l'auto-complétion de GitHub Copilot, exploration de la documentation d'une nouvelle bibliothèque.

### Améliorer la qualité du code avec GitHub Copilot

Refactoring avec Copilot : techniques pour améliorer un code existant.  
Générer des tests unitaires et d'intégration.  
Génération de code complexe et suggestions d'algorithmes.  
Exemples d'activités pratiques : refactorisation d'un morceau de code avec Copilot, écriture de tests pour des fonctions existantes.

### Intégrer GitHub Copilot dans le cycle de vie du développement logiciel

Trucs et astuces pour intégrer GitHub Copilot dans les flux de développement quotidiens.  
Planification, design, code, revue, debug... Dans quelles phases d'un projet utiliser Copilot ?  
Équilibrer le code généré par l'IA et la supervision humaine : assurer la qualité sans dépendance excessive.  
Comprendre les limites et considérations éthiques de l'utilisation de GitHub Copilot.  
Exemples d'activités pratiques : revoir et améliorer le code généré pour un projet, discuter des scénarios où Copilot est le plus bénéfique.

### Personnaliser GitHub Copilot en fonction de vos besoins

Modifier les paramètres de Copilot pour une assistance personnalisée.  
Entraîner Copilot sur votre codebase pour des suggestions plus pertinentes.  
Intégrer GitHub Copilot à d'autres outils de développement.  
Exemples d'activités pratiques : configurer Copilot pour un projet spécifique, intégrer Copilot à un pipeline CI/CD.

### Synthèse et conclusion de la formation Copilot pour développeurs

Examen d'études de cas réussies dans divers secteurs d'activité.  
Revue des apprentissages et perspectives d'application dans les différents contextes professionnels.  
Perspectives : l'IA dans le développement de logiciels et l'évolution de Copilot.  
Exemples d'activités pratiques : analyse d'études de cas, réflexions sur la manière dont Copilot peut résoudre des problèmes spécifiques dans les projets des participants.

## Moyens et méthodes pédagogiques

- ◆ La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques (hors formation de type séminaire).
- ◆ Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- ◆ Un support de cours numérique est fourni aux stagiaires

## Modalités d'évaluation

- ◆ **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- ◆ **En cours de formation**, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience, en cohérence avec les objectifs pédagogiques visés.
- ◆ **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.