Durée :	2 jours soit 14 heures	Référence :	IF-IADEV1

Public visé :

Développeurs back-end, front-end, full-stack ou mobile, développeurs d'applications métier, intégrateurs et administrateurs qui écrivent du code ou des scripts. Ils travaillent en entreprise, en ESN, en administration ou en collectivité et n'ont pas ou peu utilisé d'assistant IA de code.

Pré-requis :

- Pratiquer régulièrement au moins un langage de programmation (Python, JavaScript/TypeScript, Java, C#…).
- Utiliser un IDE et les commandes Git de base (commit, branche).

Objectifs pédagogiques :

- Expliquer le fonctionnement d'un assistant de code et ses limites.
- Choisir le mode d'interaction adapté à la tâche : complétion, chat, édition ou agent.
- Rédiger des demandes précises et fournir le contexte utile pour obtenir un code exploitable.
- Utiliser l'assistant pour comprendre, tester, déboguer et documenter du code.
- Vérifier le code produit et appliquer les règles de confidentialité de base.

Compétences acquises à l'issue de la formation :

- Être capable de configurer un assistant IA dans son environnement de développement et d'y définir des instructions de projet.
- Savoir formuler une demande contextualisée et itérer jusqu'à obtenir un code conforme au besoin.
- Être en mesure d'utiliser l'IA pour analyser, tester, déboguer et documenter un code existant.
- Savoir évaluer de manière critique le code généré (exactitude, sécurité, licences, confidentialité) avant de l'intégrer.

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.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.

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
- 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

Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert…).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquise ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin

Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.

Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :
Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr
Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.

Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- AGEFIPH
- FIPHFP
- MDPH de votre département

Description / Contenu

Module 1 – Comprendre les assistants de code IA

- Du modèle de langage à l'assistant de code : tokens, fenêtre de contexte, non-déterminisme
- Origine des erreurs : API inventées, connaissances datées, contexte incomplet
- Panorama actuel : assistants intégrés à l'IDE, agents en ligne de commande, agents dans le cloud

- Modèles, offres individuelles et entreprise : ce qui change pour l'usage des données
- Démonstration : une même demande traitée en complétion, en chat et en mode agent

Module 2 – Prendre en main l'assistant dans l'IDE

- Installation et configuration de GitHub Copilot dans VS Code, choix du modèle
- Complétion, chat, édition multi-fichiers, mode agent : rôle de chaque mode
- Ajouter du contexte : fichiers, sélection, terminal, erreurs du compilateur
- Fichier d'instructions de projet (copilot-instructions.md, AGENTS.md)
- Atelier : configurer son environnement et rédiger un premier fichier d'instructions pour un projet fourni

Module 3 – Formuler des demandes efficaces pour le code

- Structure d'une demande : objectif, contexte, contraintes, format attendu, critères d'acceptation
- Utiliser exemples, signatures et tests existants comme contexte
- Découper une tâche, itérer, repartir d'une conversation vierge
- Erreurs fréquentes : demande vague, contexte surchargé, validation sans lecture
- TP : implémenter une fonctionnalité simple (validation de formulaire, endpoint d'API) en comparant une demande vague et une demande structurée

Module 4 – Comprendre, tester et déboguer avec l'IA

- Explorer un code inconnu : architecture, flux, dépendances
- Générer des tests unitaires et identifier les cas limites
- Diagnostiquer une erreur à partir d'une trace ou d'un test en échec
- Documenter : commentaires, README, messages de commit
- TP : sur une application sans tests, produire une suite de tests, corriger deux anomalies et documenter le module

Module 5 – Garder le contrôle : qualité, sécurité, confidentialité

- Grille de relecture du code généré : logique, cas limites, performance, lisibilité
- Failles fréquentes : injections, secrets en clair, dépendances inexistantes ou obsolètes
- Données à ne jamais transmettre, paramètres de confidentialité, exclusion de fichiers
- Licences et code proche de code public : filtres disponibles et responsabilité du développeur
- Mise en situation : auditer un extrait généré contenant des défauts volontaires, puis le corriger avec l'assistant

Module 6 – Cas de synthèse

- Réaliser de bout en bout une petite évolution sur l'application fil rouge : analyse, implémentation, tests, relecture, commit
- Restitution et justification des choix : ce qui a été délégué à l'IA, ce qui a été vérifié et comment