

Sessions garanties disponibles

Formation GitHub Copilot : accélérer et fiabiliser le développement avec l'IA

Réf. SF33963

Intégrez GitHub Copilot à votre workflow quotidien pour coder, tester et documenter plus vite — de la complétion intelligente aux modes agentiques.

2 j Durée	1 500 € HT Inter / participant	4 100 € HT Intra (≤12 pers.)	43 Sessions à venir
--------------	-----------------------------------	---------------------------------	------------------------

Objectifs pédagogiques

- ✓ Installer et configurer GitHub Copilot dans son environnement de développement
- ✓ Utiliser la complétion contextuelle et Copilot Chat pour générer, expliquer et transformer du code
- ✓ Concevoir des prompts et des instructions personnalisées adaptés au contexte et aux conventions du projet
- ✓ Automatiser la génération de tests unitaires et de documentation
- ✓ Refactorer et sécuriser du code existant à partir des recommandations de Copilot
- ✓ Mettre en œuvre les modes agentiques (Copilot Edits, agent mode, MCP, CLI) pour automatiser des tâches de développement de bout en bout
- ✓ Évaluer les limites du code généré et appliquer les bonnes pratiques de vérification, de qualité et de confidentialité

Public visé & prérequis

Public visé Chefs de projet, Développeurs

Prérequis- Pratique courante d'au moins un langage de programmation - Familiarité avec un IDE, idéalement VS Code
- Bases de Git et du workflow GitHub (commits, branches, pull requests) - Compte GitHub avec licence GitHub Copilot active (essai ou licence entreprise) - Poste de développement avec VS Code et droits d'installation d'extensions

Programme détaillé

Découverte de GitHub Copilot : fonctionnement et mise en route

- L'assistant IA dans le cycle de développement : positionnement de GitHub Copilot dans l'écosystème des assistants de code (IDE, terminal, plateforme GitHub)
- Fonctionnement de Copilot : LLM, fenêtre de contexte et signaux de l'IDE — comment naissent les suggestions
- Types d'assistance : complétion de ligne, génération de blocs, transformation de code, explication et documentation
- Limites de l'outil : hallucinations, code vulnérable, complexité mal maîtrisée — pourquoi la relecture humaine reste la règle
- Installation dans VS Code : extensions, authentification et gestion des licences
- Paramétrage : personnalisation des suggestions, réglages de confidentialité et de télémétrie

Activités pratiques :

- Installation et configuration complète de Copilot dans l'IDE
- Premières complétions guidées et tour des fonctionnalités de l'interface

Complétion de code et Copilot Chat au quotidien

- Complétion contextuelle : exploiter les inline completions et arbitrer entre plusieurs suggestions
- Interfaces de dialogue : Chat View, Inline Chat, Quick Chat et Smart Actions — choisir le bon canal selon la situation
- Compréhension de code : expliquer un module inconnu, analyser la structure d'un projet existant
- Transformation de code : corriger, compléter et convertir du code via le chat
- Documentation : génération de commentaires, d'exemples et de documentation inline

Activités pratiques :

- Prise en main d'un module existant : explication puis documentation avec Copilot Chat
- Génération d'une première fonctionnalité complète sur le projet fil rouge

Prompt engineering et context engineering

- Structure d'un prompt efficace : rôle, contexte, contraintes et itération (role prompting, chain of thought)
- Instructions personnalisées (copilot instructions) : encoder les conventions du projet — naming, architecture, patterns
- Participants de chat : ``@workspace``, ``@terminal``, ``@github`` — donner à Copilot la vision du projet, du terminal et de la plateforme
- Ciblage du contexte : références ``#``, sélection de code, fichiers et dossiers
- Commandes slash : tirer parti des commandes intégrées (`/docs`, `/tests`, `/explain...`)

Activités pratiques :

- Rédaction d'instructions personnalisées reproduisant les conventions internes d'une équipe
- Génération d'une application simple à partir d'une spécification rédigée en prompts itératifs

Copilot dans le cycle de vie du projet : tests, documentation, refactoring

- Tests unitaires : génération de jeux de tests, couverture des cas limites et des conditions d'erreur
- Documentation projet : README, documentation d'API et commentaires structurés générés puis relus
- Refactoring assisté : améliorer la lisibilité, la performance et la fiabilité d'un code existant
- Migration de code : transformer un code legacy ou changer de technologie avec l'appui de Copilot
- Sécurisation : audits simples et recommandations de sécurité via le chat

Activités pratiques :

- Production d'un jeu de tests unitaires complet sur le fil rouge, cas limites inclus
- Refactoring d'un code existant à partir des recommandations de Copilot

Modes agentiques et automatisation du workflow

- Copilot Edits : modifications multi-fichiers pilotées depuis le chat
- Custom commands et custom modes : industrialiser les tâches récurrentes (mode Plan, modes spécifiques à l'équipe)
- Agent mode : confier une tâche de développement de bout en bout à l'agent dans l'IDE
- MCP (Model Context Protocol) : connecter Copilot à des outils et données externes
- Background agents : déléguer du travail à distance, de l'issue GitHub à la pull request
- Copilot CLI : l'agent en ligne de commande pour travailler hors de l'IDE
- Automatisation GitHub : commits, issues, pull requests et review de PR assistée par Copilot

Activités pratiques :

- Création d'une custom command de revue de code
- Connexion d'un serveur MCP (ex. GitHub) et exploitation dans une session agent
- Workflow automatisé de bout en bout : de l'issue à la pull request reviewée par Copilot

Qualité, sécurité et bonnes pratiques en équipe

- Copilot Autofix : détection et correction des vulnérabilités typiques (OWASP)
- Confidentialité : protection des données sensibles et réglages adaptés au contexte professionnel
- Vérification humaine : relecture systématique et responsabilité sur le code livré
- Bonnes pratiques d'adoption : ce que Copilot accélère vraiment, ce qui doit rester sous contrôle du développeur

Activités pratiques :

- Détection puis correction guidée de vulnérabilités sur le code du fil rouge

Prochaines sessions

DATE	LIEU	STATUT
29/10/2026	à distance	Sur confirmation
29/10/2026	Paris	Sur confirmation

DATE	LIEU	STATUT
29/10/2026	Nantes	Sur confirmation
29/10/2026	Lille	Sur confirmation
26/11/2026	Lyon	Sur confirmation
26/11/2026	à distance	Sur confirmation

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

4,9/5 — 165 avis

Selim B. — Très bonne formation, très instructif même après quelques années de Python dans les bras. Une très petite équipe de stagiaires qui facilite les échanges. Je redoutais le mode...

Alexis C. — Anthony explique bien les concepts attendus et prend le temps de répondre aux questions et aux différents problèmes rencontrés. Le support présenté créé par lui-même est différent des...

Votre formateur

Evan B. — Formateur Copilot & GitHub

Evan intervient sur des formations portant sur l'IA appliquée au développement et les assistants de code. Parmi les formations qui lui sont confiées figurent notamment « GitHub Copilot (GH-300) »...

Ils nous font confiance

Cette année notre filière Développement et Ingénierie de Logiciels a accompagné plus de 459 entreprises dont :

SOPRA STERIA RENAULT SAS ORANGE CAMPUS TECH ORANGE INNOVATION SCHOOL STMICROELECTRONICS ARPEGE GRANT THORNTON FRANCE ORANGE BUSINESS RTE SII ILE DE FRANCE +449 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com