Formation **Claude Code**

Conçue pour les développeurs, ingénieurs DevOps ou chefs de projet techniques, cette formation vous apprendra à tirer parti de Claude Code pour accélérer vos projets. Vous y apprendrez à installer et configurer l'outil, structurer des prompts efficaces pour générer, tester et refactorer du code, et intégrer Claude Code dans vos workflows Git/GitHub avec automatisation des tests et de la CI/CD. Le programme aborde également l'extension via le Model Context Protocol, la connexion à des services externes comme Playwright ou Figma, et les bonnes pratiques de sécurité et de collaboration humain-IA. En maîtrisant ces compétences, vous boosterez votre productivité et pourrez orchestrer vos équipes de développement autour d'assistants IA performants.

Durée

2 jours

Objectifs pédagogiques

- Installer et configurer Claude Code et son environnement sur un poste de travail pour lancer un projet de base
- Formuler et structurer des prompts efficaces pour générer, tester et refactorer du code dans un projet concret en appliquant des bonnes pratiques de prompt engineering
- Intégrer Claude Code dans un workflow professionnel utilisant Git et GitHub afin de créer des branches, générer des revues de pull requests et automatiser les tests CI/CD
- Étendre Claude Code via le Model Context Protocol (MCP) pour connecter des services externes (Playwright, Figma) et créer des commandes personnalisées afin d'automatiser des tâches récurrentes
- Concevoir, configurer et orchestrer des subagents Claude Code pour répartir des tâches de développement complexes (tests, revue de code, débogage, audit)
- Évaluer les risques de sécurité et de conformité (RGPD) liés à l'utilisation d'assistants IA et appliquer des stratégies de collaboration humain-IA pour assurer la qualité et la fiabilité du code

Public

Développeurs, ingénieurs DevOps, Chefs de projet techniques

Prérequis

Maîtrise d'au moins un langage de programmation, connaissance pratique de Git et GitHub, expérience avec la ligne de commande et les IDEs, notions de tests unitaires et de pipelines CI/CD.

Programme de formation

Phase d'inclusion

Module 1 — Prise en main et installation

- Présentation de Claude Code et des modèles Anthropic
- Installation du CLI et configuration des clés API (VS Code, JetBrains)
- Exploration de l'interface, des raccourcis et commandes de base
- Gestion des contextes (/init, fichiers CLAUDE.md)

Exemples d'activités pratiques : installer Claude Code et initialiser un dépôt Git, créer un fichier CLAUDE.md et configurer une session de travail

Module 2 — Prompt engineering et génération de code

- Principes de structuration des prompts
- Gestion des fichiers et refactorisation avec Claude Code
- Utilisation de Plan Mode et Thinking Mode
- Techniques Best-of-N et prompts comparatifs

Exemples d'activités pratiques : construire une API REST ou microservice via prompts, refactoriser un script existant avec Claude Code, générer et exécuter des tests unitaires créés par l'IA

Module 3 — Intégration Git/GitHub et automatisation CI/CD

- Workflows Git et bonnes pratiques de branches
- Intégration avec GitHub : pull requests, merges, sécurité
- Utilisation et structuration des fichiers CLAUDE.md
- Création de commandes réutilisables pour accélérer les workflows
- Mise en place de pipelines CI/CD avec Claude Code

Exemples d'activités pratiques : connecter un dépôt GitHub et automatiser une pull request, exécuter des tests dans un pipeline CI/CD via prompts Claude, développer une commande personnalisée pour automatiser un workflow

Module 4 — Extensibilité MCP, intégration d'outils externes et sécurité

- Introduction au Model Context Protocol (MCP)
- Connexion à des services externes (Playwright, Figma)
- Création de hooks et de commandes personnalisées
- Gestion des risques : RGPD, sécurité des tokens, limites des modèles
- Bonnes pratiques de collaboration humain-IA et gouvernance

Exemples d'activités pratiques : générer un composant UI à partir d'une maquette Figma, automatiser un test UI via Playwright MCP, rédiger des guidelines de sécurité et de conformité pour un projet utilisant Claude Code

Module 5 — Agents, Subagents et orchestration avancée

- Concept et architecture des subagents

- Création de subagents personnalisés
- Orchestration multi-agents (séquençage + parallélisme)
- Cas d'usage : code-review, test-automation, security-audit, debugging
- Bonnes pratiques : scopes étroits, moindre privilège, gestion contexte

Exemples d'activités pratiques : créer un subagent test-runner proactif ; Implémenter une feature avec une équipe de 4 subagents coordonnés ; Explorer un codebase en parallèle avec agents multiples ; Workflow TDD avec subagents de vérification indépendants.

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.