

Sessions garanties disponibles

Formation Claude Code : automatiser vos tâches de développement

Réf. SF34385

Apprenez les meilleures pratiques pour développer avec Claude Code et boostez vos projets grâce à l'IA !

2 j Durée	1 800 € HT Inter / participant	4 000 € HT Intra (≤12 pers.)	3 Sessions à venir
--------------	-----------------------------------	---------------------------------	-----------------------

Objectifs pédagogiques

- ✓ Installer et configurer Claude Code et son environnement sur un poste de travail pour lancer un projet de base
- ✓ Étendre Claude Code via le Model Context Protocol (MCP), les hooks et les plugins pour connecter des services externes et distribuer des configurations d'équipe
- ✓ 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
- ✓ Concevoir, configurer et orchestrer des subagents Claude Code pour répartir des tâches de développement complexes (tests, revue de code, débogage, audit)
- ✓ 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
- ✓ 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
- ✓ Créer, configurer et utiliser des skills personnalisés et des bundled skills (/simplify, /bach, /review, /loop) pour automatiser les tâches récurrentes et améliorer la qualité du code

Public visé & prérequis

- Public visé** Chefs de projet, Développeurs, Ingénieurs DevOps
- 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. À noter qu'un compte Pro ou Max est également nécessaire afin d'accéder à Claude Code.

Programme détaillé

Module 1 — Prise en main et installation

- Présentation de Claude Code et des modèles Anthropic (Opus, Sonnet, Haiku)
- Installation du CLI et configuration des clés API (VS Code, JetBrains)
- Exploration de l'interface, des raccourcis et commandes intégrées (/compact, /clear, /cost, /model)
- Gestion des contextes : /init, fichiers CLAUDE.md (projet, répertoire, utilisateur)
- Structure du répertoire .claude/: settings, permissions, skills, agents

Activités pratiques :

Installer Claude Code et initialiser un dépôt Git ; créer un fichier CLAUDE.md avec les conventions du projet et configurer une session de travail ; explorer les commandes intégrées et la gestion du contexte.

Module 2 — Prompt engineering et génération de code

- Principes de structuration des prompts pour Claude Code
- Gestion des fichiers et refactorisation avec Claude Code
- Utilisation de Plan Mode et Thinking Mode (extended thinking)
- Techniques Best-of-N et prompts comparatifs
- Sélection dynamique du modèle selon la tâche (/model haiku pour le lint, opus pour l'architecture)

Activités pratiques :

Construire une API REST ou un microservice via prompts ; refactoriser un script existant avec Claude Code ; générer et exécuter des tests unitaires créés par l'IA ; comparer les sorties de différents modèles sur une même tâche.

Module 3 — Skills : personnalisation et automatisation

- Comprendre le système d'extensibilité : skills vs. commandes intégrées vs. slash commands
- Anatomie d'une skill : structure SKILL.md, frontmatter YAML (name, description, model, allowed-tools, context)
- Niveaux de portée : projet (`./.claude/skills/`), utilisateur (`~/.claude/skills/`), entreprise (managed settings)
- Auto-découverte et invocation contextuelle des skills par Claude
- Bundled skills en détail :
- `/simplify` : revue parallèle post-implémentation (code reuse, qualité, efficacité)
- `/batch` : décomposition et parallélisation de changements à grande échelle via git worktrees
- `/review` : revue de code multi-agents avec scoring par confiance
- `/loop` : exécution récurrente d'un prompt sur un intervalle
- `/debug` : diagnostic de session via le log de debug
- `/claude-api` : chargement de la référence API Anthropic en contexte
- Création de skills personnalisées : templates, scripts embarqués, exemples
- Gestion du budget de contexte des skills (`/context`, `SLASH_COMMAND_TOOL_CHAR_BUDGET`)

Activités pratiques :

Créer une skill personnalisée pour appliquer les conventions de code d'un projet ; utiliser `/simplify` après un refactoring et analyser les suggestions ; lancer un `/batch` pour migrer un ensemble de fichiers en parallèle ; concevoir une skill avec scripts embarqués générant une visualisation HTML.

Module 4 — Intégration Git/GitHub et automatisation CI/CD

- Workflows Git et bonnes pratiques de branches avec Claude Code
- Intégration avec GitHub : pull requests, merges, sécurité
- Utilisation et structuration des fichiers CLAUDE.md partagés (synchronisation via git worktrees)
- Mise en place de pipelines CI/CD avec Claude Code
- Hooks déterministes : PreToolUse, PostToolUse, Stop, SessionStart
- Hooks shell et hooks HTTP
- Cas d'usage : formatage automatique (Prettier/Black), validation de sécurité, notifications

Activités pratiques :

Connecter un dépôt GitHub et automatiser une pull request ; exécuter des tests dans un pipeline CI/CD via prompts Claude ; configurer un hook PreToolUse pour valider les commandes dangereuses ; mettre en place un hook PostToolUse pour le formatage automatique.

Module 5 — MCP, plugins et intégration d'outils externes

- Introduction au Model Context Protocol (MCP) : architecture client/serveur
- Connexion à des services externes (Playwright, Figma, PostgreSQL)
- Configuration de serveurs MCP au niveau projet vs. utilisateur
- **Plugins** : le système de packaging de Claude Code
- Structure d'un plugin : plugin.json, skills/, agents/, hooks/, .mcp.json
- Installation et gestion : `/plugin install`, marketplaces, portée projet vs. utilisateur
- Plugins notables de l'écosystème : code-review, frontend-design, ralph-loop, Context7
- Distribution de configurations d'équipe via plugins

Activités pratiques :

Générer un composant UI à partir d'une maquette Figma via MCP ; automatiser un test UI via Playwright MCP ; créer un plugin d'équipe regroupant skills, agents et hooks ; installer et configurer un plugin depuis un marketplace.

Module 6 — Subagents, Agent Teams et orchestration avancée

- Concept et architecture des subagents : types (Explore, Plan, general-purpose, custom)
- Création de subagents personnalisés (`./.claude/agents/`)
- Orchestration multi-agents : séquençage et parallélisme
- **Agent Teams** : coordination inter-agents avec partage de liste de tâches et communication directe
- Cas d'usage avancés : code-review multi-agents, test-automation, security-audit, debugging
- Bonnes pratiques : scopes étroits, moindre privilège, gestion du contexte, limitation des tours (`max_turns`)
- Sécurité et conformité
- Gestion des risques : RGPD, sécurité des tokens, limites des modèles
- Bonnes pratiques de collaboration humain?IA et gouvernance

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 ; configurer un workflow TDD avec subagents de vérification indépendants ; rédiger des guidelines de sécurité et de conformité pour un projet.

Prochaines sessions

DATE	LIEU	STATUT
15/10/2026	à distance	Sur confirmation
17/11/2026	à distance	Session garantie
10/12/2026	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 — 65 avis

Rachelle Y. — Merci c'était super :)

Anthony G. — Formation très bien organisée, très claire Alison explique très clairement et est très à l'écoute, avec toujours une approche pratique essentielle. je n'avais jamais fait de Python et...

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 Solutions IA pour l'IT a accompagné plus de 11 entreprises dont :

LE PARISIEN MANAGEMENT ADEO SERVICES ADSN CHAPSVISION CMI FRANCE EASYMILE GIE APRIL COURTAGE OPENSEE SAS ORANGE BUSINESS VODAFONE AUTOMOTIVE TELEMATICS DEVELOPMENT SAS +1 autre

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