

# Formation : Développer avec Claude Code

Maîtriser l'agent IA Claude Code pour automatiser, accélérer et sécuriser le développement logiciel

Formation pratique - 2j - 14h00 - Réf. DCL

NEW

Claude Code est un agent d'intelligence artificielle conçu pour assister les développeurs dans leurs activités quotidiennes de programmation. Il permet d'interagir directement avec une base de code afin d'analyser un projet. Cette formation va permettre aux participants d'intégrer Claude Code dans leur environnement de développement afin d'automatiser certaines tâches, accélérer la production de code et améliorer la qualité des projets logiciels.

## Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Comprendre le rôle et l'architecture de l'IA Claude Code dans le processus de développement logiciel
- ✓ Installer et configurer Claude Code dans son environnement de développement
- ✓ Maîtriser les commandes et modes d'interaction avec l'agent
- ✓ Rédiger des prompts efficaces pour des tâches de développement complexes
- ✓ Automatiser des workflows de développement itératif
- ✓ Appliquer les bonnes pratiques de sécurité et de gestion des coûts
- ✓ Appliquer les bonnes pratiques pour maximiser la productivité et la qualité du code avec Claude Code

## Public concerné

Développeurs, DevOps, architectes logiciels, tech leads.

## Prérequis

Maîtrise du terminal et de la ligne de commande. Connaissance d'au moins un langage de programmation. Notions de base en Git. Familiarité avec un environnement de développement intégré (IDE) comme Visual Studio Code.

### PARTICIPANTS

Développeurs, DevOps, architectes logiciels, tech leads.

### PRÉREQUIS

Maîtrise du terminal et de la ligne de commande. Connaissance d'au moins un langage de programmation.

Notions de base en Git. Familiarité avec un environnement de développement intégré (IDE) comme Visual Studio Code.

### COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

### MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Méthodes et moyens pédagogiques

Veuillez saisir la description pédagogique ici

### Méthodes pédagogiques

Travaux pratiques. Des exercices et travaux pratiques permettront de créer des applications réelles en utilisant exclusivement Claude Code.

## Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Programme de la formation

### 1 Introduction à Claude Code

- Qu'est-ce qu'un agent de développement ?
- Positionnement de Claude Code dans l'écosystème des outils IA pour développeurs.
- Architecture et fonctionnement : modèle Claude, contexte, outils disponibles.
- Cas d'usage : compréhension du code existant, refactoring, debugging, génération de code, documentation, tests.
- Limites, sensibilisation à la confidentialité du code et des données, calcul des coûts.

## Activités pédagogiques

### Travaux pratiques

Explorer les capacités de Claude Code sur un projet existant : analyse de codebase, identification de bugs, suggestions d'amélioration.

### 2 Installation et configuration

- Prérequis système et authentification.
- Installation via npm et configuration initiale.
- Gestion des clés API et des plans tarifaires.
- Configuration du fichier settings et des variables d'environnement.
- Personnalisation avec les fichiers CLAUDE.md (projet et utilisateur).
- Gestion des permissions et du mode d'approbation.

## Activités pédagogiques

### Travaux pratiques

Installer Claude Code, configurer son environnement et créer un fichier CLAUDE.md adapté à un projet de développement.

### MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

### MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

### ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr).

### 3 Commandes et modes d'interaction

- Le mode interactif REPL et ses commandes slash.
- Mode one-shot pour les tâches ponctuelles.
- Mode pipe pour l'intégration avec d'autres outils Unix.
- Navigation dans l'historique et gestion des conversations.
- Commandes essentielles : /init, /compact, /clear, /memory, /model.
- Lecture et manipulation de fichiers, exécution de commandes

## Activités pédagogiques

### Travaux pratiques

Réaliser une série de tâches de développement en utilisant les différents modes d'interaction : correction de bug, ajout de fonctionnalité, génération de tests.

### 4 L'art du prompting pour le développement

- Principes de rédaction de prompts efficaces pour le code.
- Fournir le contexte nécessaire : structure du projet, conventions, contraintes.
- Décomposer les tâches complexes en étapes.
- Utiliser les images et captures d'écran pour le contexte visuel.
- Itérer et affiner les résultats.
- Patterns de prompts réutilisables.

## Activités pédagogiques

### Travaux pratiques

Rédiger et tester différents prompts pour une même tâche de développement. Comparer les résultats et identifier les bonnes pratiques.

### 5 Workflows de développement avancés

- Gestion de projet avec Claude Code : de l'idéation au déploiement.
- Refactoring de code legacy à grande échelle.
- Génération et maintenance de documentation technique.
- Création et exécution de tests automatisés.
- Revue de code et analyse de pull requests.
- Débogage assisté et résolution de problèmes complexes.

## Activités pédagogiques

### Travaux pratiques

Prendre un projet legacy, effectuer un refactoring complet avec Claude Code : amélioration de la structure, ajout de tests, génération de documentation.

## 6 Automatisation et intégration CI/CD

- Utilisation de Claude Code en mode headless et non-interactif.
- Intégration avec GitHub Actions.
- Automatisation des revues de code.
- Création de hooks pre-commit personnalisés.
- Gestion des secrets et sécurité en environnement CI.

### Activités pédagogiques

#### Travaux pratiques

Configurer un workflow GitHub Actions utilisant Claude Code pour automatiser la revue de code et la génération de documentation sur chaque pull request.

## 7 Architecture Model Context Protocol (MCP)

- Comprendre le protocole MCP et son rôle.
- Serveurs MCP disponibles : filesystem, bases de données, APIs tierces.
- Configuration des serveurs MCP dans Claude Code.
- Étendre les capacités de Claude Code avec des outils personnalisés.
- Cas d'usage : intégration avec Postgres, Puppeteer, outils internes.

### Activités pédagogiques

#### Travaux pratiques

Configurer un serveur MCP pour connecter Claude Code à une base de données et effectuer des requêtes en langage naturel.

## 8 Bonnes pratiques et optimisation

- Gestion des coûts : suivi de la consommation, optimisation des tokens.
- Déterminer l'architecture de prompt la plus économique pour un cas d'usage donné.
- Sécurité : permissions, données sensibles, audit des actions.
- Organisation du travail : quand utiliser Claude Code vs coder manuellement.
- Travail en équipe : partage des configurations, conventions communes.
- Debugging de Claude Code : logs, verbose mode, résolution de problèmes.
- Veille et évolution de l'outil.

### Activités pédagogiques

- Auditer un projet existant : analyser la consommation, identifier les optimisations possibles, mettre en place des conventions d'équipe pour l'utilisation de Claude Code.

#### Travaux pratiques

Description des travaux pratiques