

retengr

Claude Code pour Développeurs : Intégrer l'IA agentique dans son workflow de développement logiciel.

Durée : 3 jours (21 heures)



Méthode pédagogique

La formation est volontairement orientée pratique.

Les participants travaillent sur une application fil rouge permettant de mettre progressivement en œuvre les différentes fonctionnalités de Claude Code.

Les apports théoriques sont immédiatement suivis de démonstrations et de travaux pratiques.

60 % pratique / 40 % théorie.

Présentation

Claude Code est un outil d'IA agentique conçu pour accompagner les développeurs directement dans leur environnement de travail. Au-delà de la génération ponctuelle de code, il permet de comprendre une base de code, rechercher et modifier plusieurs fichiers, exécuter des commandes, corriger des erreurs, écrire des tests, refactorer une application et prendre en charge des tâches de développement en plusieurs étapes.

Cette formation permet aux développeurs d'apprendre à utiliser Claude Code comme un véritable partenaire de développement, depuis la compréhension d'un projet existant jusqu'à l'implémentation, au test et à la validation de nouvelles fonctionnalités.

Les participants apprennent à donner à Claude Code le bon contexte, formuler des instructions efficaces, organiser et piloter son travail agentique, contrôler les modifications proposées et intégrer l'outil dans leurs pratiques de développement et leurs workflows Git.

Une attention particulière est portée à la maîtrise du code généré : l'objectif n'est pas de déléguer aveuglément le développement à l'IA, mais de savoir déterminer ce qui peut lui être confié, contrôler ses actions, vérifier ses choix techniques et maintenir les exigences de qualité, de sécurité et de maintenabilité du projet.

La formation repose majoritairement sur la pratique. Les participants travaillent sur une base de code afin d'expérimenter Claude Code sur des



situations proches de leur quotidien : exploration d'un projet, implémentation d'une fonctionnalité, debugging, génération de tests, refactoring, revue de code et automatisation de tâches de développement.

À l'issue des trois jours, les participants disposent d'une méthodologie concrète pour intégrer Claude Code à leur workflow quotidien et accélérer leurs développements tout en conservant la maîtrise technique du code produit.



Objectifs

- Comprendre le fonctionnement et les principes de Claude Code
- Installer et configurer Claude Code dans leur environnement
- Utiliser efficacement Claude Code depuis le terminal et leur environnement de développement
- Fournir le contexte nécessaire à une tâche de développement
- Explorer et comprendre rapidement une base de code existante
- Formuler des instructions efficaces pour piloter un agent de développement
- Planifier et décomposer une tâche complexe avant son implémentation
- Générer et modifier du code sur plusieurs fichiers
- Utiliser Claude Code pour rechercher et corriger des bugs
- Générer, compléter et améliorer des tests
- Refactorer du code avec l'assistance de l'IA
- Utiliser Claude Code avec Git et dans un processus de revue de code
- Personnaliser son environnement et ses instructions projet
- Identifier les limites, risques et bonnes pratiques d'un développement assisté par IA
- Intégrer Claude Code de manière efficace dans son workflow quotidien

Audience

Développeurs, Lead developers, Software engineers, Tech leads, Architectes logiciels souhaitant expérimenter le développement agentique, Toute personne participant activement au développement et à la maintenance d'applications



Le formateur

Le formateur est un expert du domaine qui intervient sur le sujet depuis plusieurs années en formation mais aussi en conseil.

Doté d'une grande qualité d'écoute, sa pédagogie et sa compétence technique vous permettront d'acquérir les compétences visées.

Il saura alterner entre théorie, pratique, et retours d'expérience.

Pré-requis

Cette formation s'adresse à des développeurs disposant déjà d'une pratique du développement logiciel.

Maîtriser au moins un langage de programmation

Être à l'aise avec Git

Connaître les principes courants du développement logiciel et des tests

Être à l'aise avec l'utilisation d'un terminal

Disposer d'un environnement de développement fonctionnel

Disposer d'un accès à Claude Code adapté aux travaux pratiques

Aucune expérience préalable de Claude Code n'est nécessaire.

Programme

Jour 1 - Prendre en main Claude Code et apprendre à travailler avec un agent - 1. Comprendre Claude Code et le développement agentique [1.5h]

- Évolution du développement assisté par IA
- Du copilote de code à l'agent de développement
- Positionnement et principes de Claude Code



- Comprendre ce que Claude Code peut faire dans un projet
- Accès au système de fichiers et à la base de code
- Recherche et navigation dans le projet
- Modification de fichiers
- Exécution de commandes et de tests
- Utilisation des outils à disposition de l'agent
- Comprendre la boucle agentique : analyser → agir → observer → corriger
- Ce que Claude Code sait bien faire
- Ses limites et les situations dans lesquelles le développeur doit reprendre la main
- Sécurité et maîtrise des actions
- Comprendre les permissions
- Contrôler les actions réalisées
- Éviter l'exécution d'actions indésirables
- Protéger secrets, credentials et données sensibles
- Conserver une validation humaine sur les opérations critiques

Jour 1 - 2. Installer et configurer son environnement [1h]

- Installation et prérequis
- Authentification et configuration
- Prise en main de l'interface en ligne de commande
- Organisation d'une session de travail
- Comprendre le contexte fourni à Claude
- Gérer efficacement la quantité de contexte
- Intégration avec l'environnement de développement
- Premières commandes et interactions

Jour 1 - 3. Comprendre une base de code avec Claude Code [2h]

- Faire explorer un repository existant
- Comprendre l'architecture générale d'une application
- Identifier les composants principaux
- Rechercher où une fonctionnalité est implémentée
- Comprendre les dépendances entre fichiers
- Suivre un flux fonctionnel dans le code
- Localiser une règle métier



- Interroger Claude sur une portion de code
- Faire expliquer du code complexe ou peu documenté
- Générer une cartographie ou une synthèse d'architecture
- Identifier les zones susceptibles d'être impactées par une modification

Jour 1 - 4. Bien piloter Claude Code [2.5h]

- Pourquoi une demande vague produit des modifications difficiles à contrôler
- Donner le contexte métier et technique
- Définir précisément le résultat attendu
- Indiquer les contraintes d'architecture
- Préciser les conventions du projet
- Fournir des exemples
- Définir les critères d'acceptation
- Demander à Claude d'explorer avant de modifier
- Faire produire un plan avant l'implémentation
- Décomposer une tâche complexe
- Travailler par étapes
- Challenger le plan proposé
- Corriger la trajectoire en cours de tâche
- Vérifier le résultat final

Jour 2 - Développer, tester et maintenir avec Claude Code - 5. Implémenter une fonctionnalité avec Claude Code [2h]

- Transformer une user story en plan d'implémentation
- Faire analyser les impacts
- Identifier les fichiers à modifier
- Faire proposer plusieurs approches
- Challenger les choix techniques
- Implémenter progressivement
- Modifier plusieurs fichiers de manière cohérente
- Respecter l'architecture et les conventions existantes
- Vérifier les modifications
- Faire corriger une implémentation



Jour 2 - 6. Debugger avec Claude Code [1.5h]

- Fournir efficacement un bug à l'agent
- Exploiter messages d'erreur, logs et stack traces
- Demander à Claude de rechercher la cause plutôt que de corriger immédiatement
- Générer des hypothèses
- Explorer le code concerné
- Reproduire un bug
- Identifier la cause racine
- Proposer une correction
- Vérifier qu'une correction n'introduit pas de régression

Jour 2 - 7. Générer et améliorer les tests [1.5h]

- Analyser les tests existants
- Identifier les scénarios manquants
- Générer des tests unitaires
- Générer des tests d'intégration lorsque le projet s'y prête
- Construire des cas limites
- Générer des données de test
- Exécuter les tests
- Analyser les échecs
- Corriger code et tests
- Éviter les tests superficiels générés uniquement pour « faire passer » le code
- Utiliser les tests comme garde-fou du développement agentique

Jour 2 - 8. Refactoring et amélioration de la qualité du code [2h]

- Identifier les zones de dette technique
- Demander une analyse avant refactoring
- Simplifier du code complexe
- Réduire les duplications
- Améliorer la lisibilité
- Extraire des composants ou fonctions
- Moderniser progressivement du code existant
- Maintenir les comportements existants



- Utiliser les tests pour sécuriser le refactoring
- Comparer le code avant/après
- Faire expliquer les choix réalisés par Claude

Jour 3 - Industrialiser son utilisation de Claude Code - 9. **Donner durablement le contexte du projet à Claude [1.5h]**

- L'objectif est d'éviter de répéter les mêmes instructions à chaque session.
- Définir les règles et conventions du projet
- Utiliser les mécanismes de mémoire et d'instructions de Claude Code
- Structurer un fichier CLAUDE.md
- Documenter architecture et conventions
- Définir les commandes de build et de test
- Fournir les règles de qualité
- Définir les pratiques à respecter
- Organiser les instructions selon les besoins du projet
- Maintenir ces instructions dans le temps

Jour 3 - 10. Git et revue de code assistée par IA [1.5h]

- Utiliser Claude Code dans un repository Git
- Comprendre les modifications apportées
- Analyser un diff
- Préparer des modifications cohérentes
- Générer des messages de commit pertinents
- Préparer une Pull/Merge Request
- Résumer les changements
- Analyser les impacts
- Utiliser Claude comme assistant de revue de code
- Rechercher bugs, incohérences et risques de régression
- Ne pas confondre revue IA et validation humaine

Jour 3 - 11. Personnaliser et étendre Claude Code [1.5h]

- Personnaliser son environnement de travail
- Créer des commandes et workflows réutilisables
- Comprendre les possibilités offertes par les hooks



- Automatiser certaines actions autour du développement
- Comprendre le principe des sous-agents spécialisés
- Répartir certaines tâches selon leur nature
- Comprendre le rôle du MCP
- Connecter Claude à des outils ou sources de contexte supplémentaires
- Identifier les usages pertinents en entreprise
- L'objectif ici n'est pas de tout maîtriser, mais de comprendre comment passer d'une utilisation individuelle à un environnement Claude Code adapté aux pratiques de l'équipe.

Jour 3 - 12. Utiliser Claude Code dans un workflow d'équipe [1h]

- Définir les usages adaptés au projet
- Partager les instructions et conventions
- Standardiser certaines tâches
- Documenter les pratiques d'utilisation
- Intégrer l'IA au cycle de développement
- Définir ce qui peut être délégué
- Définir ce qui doit obligatoirement être revu
- Sécurité du code généré
- Confidentialité et propriété intellectuelle
- Gestion des secrets
- Qualité et maintenabilité
- Maîtriser le coût et l'utilisation du contexte

Jour 3 - 13. Atelier final - Développer une fonctionnalité de bout en bout [1.5h]

- 1. Explorer
 - Comprendre le besoin
 - Explorer la base de code
 - Identifier les impacts
- 2. Planifier
 - Construire le plan d'implémentation
 - Définir les critères d'acceptation
 - Challenger la proposition
- 3. Développer



- Modifier le code
- Respecter les conventions du projet
- Contrôler chaque étape
- 4. Tester
- Générer ou compléter les tests
- Exécuter les tests
- Corriger les erreurs
- 5. Vérifier
- Relire le diff
- Rechercher les effets de bord
- Contrôler la qualité et la sécurité
- 6. Préparer la livraison
- Documenter les changements
- Préparer le commit et la revue de code

Modalités et délais d'accès à la formation

Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation, en interentreprises, dans la limite des places disponibles.

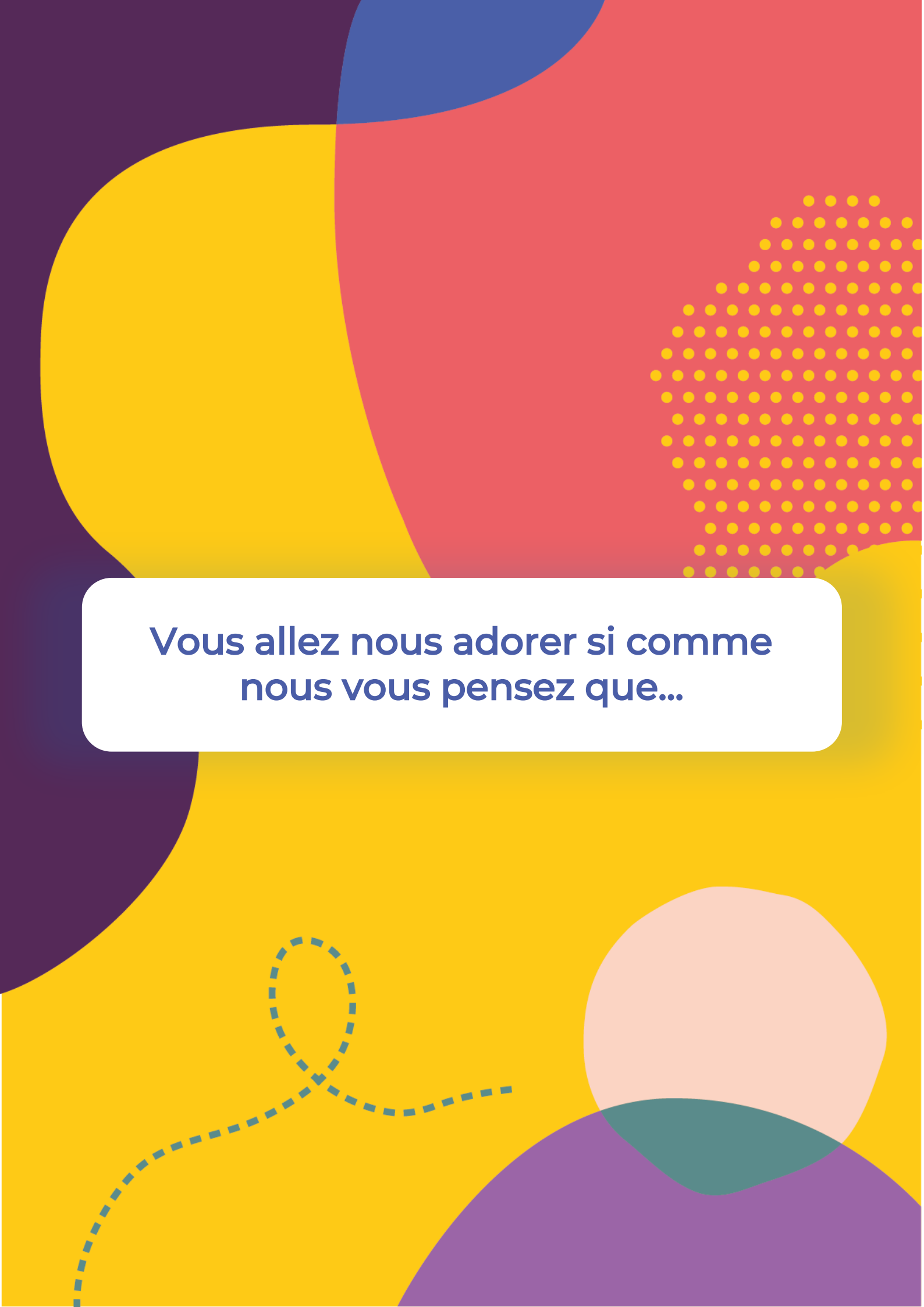
Pour les formations organisées en intra entreprise, la liste des participants peut être modifiée jusqu'à 24h ouvrés avant le début de la formation.

Accessibilité

RETENGR facilite l'accessibilité de ses formations.

Cette formation est accessible aux personnes en situation de handicap.

Si vous avez un besoin d'accès spécifique, contactez Céline BOURREIL (celine.bourreil@retengr.com) qui étudiera avec Handifiel's (notre référent handicap) votre demande et vous proposera les meilleures solutions.

The background is a vibrant collage of abstract shapes and colors. It includes a large yellow circle on the left, a pinkish-red area on the top right with a yellow dotted pattern, a blue shape at the top center, a purple shape on the bottom right, and a light pink circle. A dashed teal line winds through the lower left. The text is centered in a white rounded rectangle.

**Vous allez nous adorer si comme
nous vous pensez que...**

Une formation doit être au service de la performance du collaborateur et de l'entreprise

Ceci nécessite une quête constante d'excellence de la part de l'organisme formateur avec une adaptation systématique aux enjeux de l'entreprise, la mise à jour régulière des supports de cours et une veille technologique indispensables pour toujours être à la pointe du domaine.



L'expertise technique est aussi importante que les qualités pédagogiques



Nos formateurs sont tous des experts de leur domaine. Mais qu'ont-ils de plus que les autres ? Nous les sélectionnons en plus pour leurs qualités de pédagogue et leurs méthodes d'enseignements. Nous plaçons les qualités pédagogiques au même niveau que l'expertise afin que nos stagiaires tirent le meilleur de leurs formations.

L'excellence naît de l'excellence

Beaucoup de nos clients se classent parmi les leaders de leurs industries respectives ou parmi les start-ups les plus prometteuses. Nous savons que former les collaborateurs de telles entreprises nécessite de prêter attention à chaque détail en prodiguant un accompagnement à la hauteur de l'ambition de nos stagiaires. C'est pourquoi nous savons faire des leaders d'aujourd'hui les champions de demain !





retengr

Faire du leader d'aujourd'hui, le
champion de demain