

Mis à jour le 12/03/2026

S'inscrire

Formation Agent Harness : Dominez le code avec Claude

3 jours (21 heures)

Présentation

Notre formation Agent Harness vous apprend à piloter Claude comme un véritable coéquipier de développement : génération de code fiable, refactorings rapides et automatisation de tâches répétitives. Vous repartez avec des patterns réutilisables pour accélérer vos cycles build-test-review sur des cas concrets.

La formation vise à concevoir et exécuter des workflows agentiques avec Harness : définition d'objectifs, découpage en étapes, gestion du contexte, et contrôle qualité (tests, lint, sécurité). L'approche met l'accent sur la traçabilité des décisions et la réduction des hallucinations via des consignes, contraintes et validations systématiques.

Chaque séquence alterne démos et ateliers : création d'un agent "code reviewer", d'un agent "migration" et d'un agent "bug fixer". Livrables : prompts structurés, scripts Harness, checklists de validation, et un mini-projet intégrant génération, tests et revue automatisée.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Configurer un projet Harness et exécuter des runs reproductibles.
- Rédiger des instructions robustes (rôles, contraintes, critères d'acceptation).
- Orchestrer des tâches multi-étapes avec contexte et mémoire contrôlés.
- Automatiser refactorings, corrections et migrations avec validations.
- Mettre en place des garde-fous : tests, lint, revue et logs exploitables.

Public visé

- Développeurs back-end / front-end
- Tech leads et responsables qualité
- Ingénieurs DevOps / plateforme
- Consultants et formateurs techniques

Pré-requis

- Maîtrise d'un langage de programmation (JavaScript, Python, Java, etc.)
- Pratique de Git et des workflows de revue (PR/MR)
- Notions de tests automatisés et de CI
- Confort avec la ligne de commande

Pré-requis techniques

- Machine avec 8 Go RAM minimum (16 Go recommandé)
- macOS, Linux ou Windows (WSL2 recommandé)
- Un éditeur de code (VS Code, IntelliJ, etc.)
- Accès à Claude et installation/configuration de Harness
- Node.js ou Python selon les ateliers, plus un terminal (Bash/Zsh/PowerShell)

Programme de notre formation Agent Harness

[Jour 1 - Matin]

Fondamentaux de Claude Code et Architecture des Agents

- Introduction à Claude Code CLI : installation, authentification et configuration
- Philosophie de l'Agent Harness : pourquoi encapsuler l'IA ? (Sécurité, Contexte, Contrôle)
- Comprendre la boucle de raisonnement Agentic Workflow (Thought -> Action -> Observation)
- Gestion du budget de tokens et des limites de l'API Anthropic
- Panorama des capacités : édition de fichiers, exécution de commandes et recherche
- Atelier pratique : Installation de l'environnement et premier "Pair Programming" avec Claude Code sur un projet local.

[Jour 1 - Après-midi]

Maîtriser le Contexte et le Protocole MCP

- Le Model Context Protocol (MCP) : connecter Claude à votre écosystème (GitHub, Slack, DB)
- Configuration des serveurs MCP existants et création d'un serveur MCP custom
- Stratégies de réduction de contexte : éviter la saturation des fenêtres de prompt
- Gestion du fichier de configuration .claudecode.toml et règles de projet
- Utilisation des outils de lecture de structure (ls, cat, grep) par l'agent

- Atelier pratique : Créer un connecteur MCP pour permettre à Claude de lire une documentation technique privée.

[Jour 2 - Matin]

Construction d'un Agent Harness Sécurisé

- Isolation et exécution : Utilisation de Docker ou de sandbox type E2B
- Mise en place de Garde-fous (Guardrails) : filtrage des commandes dangereuses
- Validation humaine : configurer les modes "Architecte" vs "Agent"
- Gestion des permissions fines sur le système de fichiers et les processus
- Architecture d'un harnais minimaliste : entrées, sorties et boucles de contrôle
- Atelier pratique : Développer un "Harness" Python qui intercepte les commandes de Claude avant exécution.

[Jour 2 - Après-midi]

Automated Engineering : Refactoring et Tests

- Stratégies de Test-Driven Development (TDD) pilotées par l'agent
- Refactoring de dette technique : analyse de complexité assistée par l'IA
- Génération de tests unitaires et d'intégration (Playwright, Jest)
- Migration de code (ex: JS vers TS) et mise à jour de dépendances
- Utilisation de l'outil "edit_file" pour des modifications précises sans réécriture totale
- Atelier pratique : Migration d'un module legacy avec couverture de tests générée par Claude Code.

[Jour 3 - Matin]

Industrialisation et Intégration CI/CD

- Intégrer Claude Code dans les pipelines GitHub Actions ou GitLab CI
- Automated PR Review : configurer un agent pour commenter les Pull Requests
- Génération automatique de documentation technique (Swagger, README)
- Gestion des secrets et variables d'environnement dans un flux agentique
- Audit de sécurité automatisé par l'agent durant le build
- Atelier pratique : Création d'un bot de CI qui corrige automatiquement les erreurs de Lint.

[Jour 3 - Après-midi]

Observabilité et Monitoring des Agents

- Tracer les décisions de l'IA : logging des Chain of Thoughts
- Évaluation de la performance des agents : taux de succès et benchmarks
- Coût et ROI : monitorer la consommation de l'API Anthropic en production

- Vers l'autonomie complète : exploration des limites actuelles et éthique
- Checklist de mise en production d'un agent autonome
- Atelier pratique : Projet final - Résolution d'un ticket GitHub complexe de A à Z (Analyse -> Code -> Test -> PR).

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.