



AKADEMIA
FORMATION

PROGRAMME DE FORMATION

Claude Code — Développement Assisté par IA

DURÉE

2 jours

14 heures

FORMAT

Inter · Intra

Présentiel ou distanciel

PUBLIC

Tout collaborateur

CERTIFICATION

Incluse

À PROPOS DE LA FORMATION

Claude Code — Développement Assisté par IA.

DURÉE

2 jours

14 heures

FORMAT

Inter · Intra

Présentiel ou distanciel

PÉDAGOGIE

Active

Petits groupes

CERTIFICATION

Incluse

Attestation délivrée

Objectifs pédagogiques

- Installer et configurer Claude Code sur les différentes surfaces (CLI, VS Code, JetBrains, Desktop, Web)
- Maîtriser la boucle agentique et le système d'outils intégrés (Read, Edit, Bash, Grep, Glob)
- Appliquer le workflow Explore-Plan-Code-Commit pour des modifications multi-fichiers
- Configurer le système de permissions et les modes d'approbation pour sécuriser les opérations
- Optimiser la fenêtre de contexte avec /compact, /clear, CLAUDE.md et l'auto-mémoire
- Créer des sous-agents personnalisés pour déléguer des tâches spécialisées
- Intégrer Claude Code dans un workflow Git complet (branches, commits, pull requests, code review)

PROGRAMME

Huit modules progressifs pour monter en compétences.

MODULE

01.

1H45H

Introduction et installation de Claude Code

CONTENU PÉDAGOGIQUE

- Comprendre le positionnement de Claude Code : outil agentique vs assistants inline (Copilot, Cursor)
- Les 5 surfaces d'utilisation : CLI, VS Code, JetBrains, Desktop App, Web (claude.ai/code)
- Installation pas à pas : curl natif, Homebrew, WinGet, extension VS Code, plugin JetBrains
- Première session : authentification, connexion au compte Claude Pro/Max, navigation dans l'interface
- Mobilité entre surfaces : Remote Control, /desktop, --teleport

MODULE

02.

1H45H

La boucle agentique et les outils intégrés

CONTENU PÉDAGOGIQUE

- Architecture de la boucle : Rassembler le contexte → Agir → Vérifier → Répéter
- Les modèles disponibles : Sonnet (rapidité) vs Opus (raisonnement profond), changement avec /model
- Extended Thinking et raisonnement adaptatif : niveaux d'effort (low, medium, high, max), ultrathink
- Les 5 catégories d'outils : fichiers (Read, Edit, Write), recherche (Glob, Grep), exécution (Bash), web (WebSearch, WebFetch), intelligence de code
- Atelier pratique : explorer une codebase existante avec Read, Grep et Glob

Le système de permissions

CONTENU PÉDAGOGIQUE

- Les 6 modes de permission : default, acceptEdits, plan, auto, dontAsk, bypassPermissions
- Cycler entre les modes avec Shift+Tab pendant une session
- Règles de permission : syntaxe allow/deny, wildcards Bash, patterns Read/Edit
- Hiérarchie des settings : managed > CLI > local > shared > user
- Atelier pratique : configurer des permissions adaptées à son workflow (npm, git, tests)

Workflow Explore, Plan, Code, Commit

CONTENU PÉDAGOGIQUE

- Explorer : questions larges puis ciblées, mentions @ pour référencer des fichiers, vocabulaire métier
- Planifier : Plan Mode (Shift+Tab ou --permission-mode plan), ouverture du plan dans l'éditeur (Ctrl+G)
- Coder : rédiger des prompts efficaces (spécifiques, vérifiables), interrompre et réorienter Claude
- Committer : staging, messages de commit, création de branches, ouverture de PR via gh
- Atelier complet : corriger un bug end-to-end (exploration, plan, fix, tests, commit, PR)

Gestion du contexte et mémoire projet

CONTENU PÉDAGOGIQUE

- Anatomie de la fenêtre de contexte : system prompt, auto memory, environment info, MCP tools, CLAUDE.md, rules, git status
- Commandes de gestion : /compact (compaction ciblée), /clear (session fraîche), /context (visualisation)
- CLAUDE.md : portées (managed, projet, utilisateur, local), écriture d'instructions efficaces (< 200 lignes)
- Organisation avancée : imports @path, .claude/rules/ avec règles conditionnelles par chemin (paths)
- Auto-mémoire : stockage dans ~/.claude/projects/, MEMORY.md, fichiers de sujets, commande /memory
- Génération automatique avec /init et le flux interactif multi-phases

Sous-agents et délégation de tâches

CONTENU PÉDAGOGIQUE

- Concept : fenêtre de contexte isolée, préservation du contexte principal, contrôle des coûts
- Sous-agents intégrés : Explore (3 niveaux de minutie), Plan, general-purpose, Claude Code Guide
- Créer un sous-agent custom : fichier .md avec frontmatter YAML, commande /agents, flag --agents
- Champs de configuration : name, description, tools, model, permissionMode, maxTurns, isolation
- Foreground vs Background : Ctrl+B pour basculer, permissions en arrière-plan
- Atelier pratique : créer un sous-agent code-reviewer et un sous-agent security-checker

CLI avancé et productivité

CONTENU PÉDAGOGIQUE

- Commandes CLI essentielles : `claude`, `claude -p`, `claude -c`, `claude -r`, `claude update`, `claude mcp`
- Flags avancés : `--model`, `--worktree`, `--output-format`, `--max-turns`, `--max-budget-usd`, `--effort`
- Sessions et continuité : sauvegarde JSONL, `--continue`, `--resume`, `--from-pr`, `--fork-session`
- Commandes slash : `/compact`, `/context`, `/memory`, `/init`, `/model`, `/effort`, `/resume`, `/permissions`, `/agents`
- Raccourcis clavier : `Ctrl+C`, `Ctrl+B`, `Shift+Tab`, `Option+P`, `Option+T`, `Esc+Esc`, `@`, `!`
- Utilisation comme utilitaire Unix : pipes, scripts, composition de commandes

Intégration Git et bonnes pratiques

CONTENU PÉDAGOGIQUE

- Git Worktrees : `--worktree` pour sessions parallèles, isolation des modifications, `.worktreeinclude`
- Code Review automatisée : revue de sécurité, linter Claude dans les scripts de build
- Introduction au MCP : connexion à des outils externes (Google Drive, Jira, Slack)
- Introduction aux hooks : `PreToolUse`, `PostToolUse`, `Notification`, cas d'usage (auto-formatage, blocage)
- Les 10 bonnes pratiques Claude Code : explorer d'abord, être spécifique, donner des critères de vérification, déléguer, itérer, nommer les sessions, gérer le `CLAUDE.md`, utiliser les sous-agents, surveiller le contexte, configurer les permissions
- Quiz de validation des acquis et retours d'expérience

PASSONS À L'ACTION

Construisons ensemble votre session sur-mesure.

Dites-nous vos contraintes (format, lieu, dates, nombre de participants) et recevez une proposition personnalisée sous 24 heures ouvrées.

Akademia Formation

SERVICE ADMINISTRATION DES
VENTES

adv@akademiaformation.com

www.akademiaformation.com

Devis personnalisé

RÉPONSE SOUS 24 H OUVRÉES

Format inter · intra

Présentiel ou distanciel

— FIN DU PROGRAMME —