



AKADEMIA
FORMATION

PROGRAMME DE FORMATION

Claude Code Avancé — Subagents, Skills et Teams

DURÉE

2 jours

14 heures

FORMAT

Inter · Intra

Présentiel ou distanciel

PUBLIC

Tout collaborateur

CERTIFICATION

Incluse

À PROPOS DE LA FORMATION

Claude Code Avancé — Subagents, Skills et Teams.

DURÉE

2 jours

14 heures

FORMAT

Inter · Intra

Présentiel ou distanciel

PÉDAGOGIE

Active

Petits groupes

CERTIFICATION

Include

Attestation délivrée

Objectifs pédagogiques

- Concevoir et déployer des sous-agents personnalisés avec frontmatter YAML avancé (isolation, mémoire, hooks)
- Créer des Skills réutilisables avec variables, injection de contexte dynamique et contrôle d'invocation
- Orchestrer des équipes multi-agents avec TeamCreate (tâches, mailbox, coordination)
- Implémenter des hooks à chaque étape du cycle de vie (PreToolUse, PostToolUse, SessionStart, Notification)
- Maîtriser le Claude Agent SDK en Python et TypeScript pour construire des agents de production
- Automatiser des pipelines CI/CD et des tâches planifiées avec Claude Code en mode headless
- Optimiser la gestion mémoire, le contexte et les performances pour des projets d'entreprise

PROGRAMME

Huit modules progressifs pour monter en compétences.

MODULE

01.

1H35H

Architecture des sous-agents avancés

CONTENU PÉDAGOGIQUE

- Rappel : concept de sous-agent, fenêtre de contexte isolée, préservation du contexte principal
- Sous-agents intégrés en détail : Explore (3 niveaux de minutie), Plan, general-purpose, Claude Code Guide
- Création avancée : fichier .md avec frontmatter YAML complet (name, description, tools, model, permissionMode)
- Champs avancés : maxTurns, skills, mcpServers, hooks, memory, background, effort, color, initialPrompt
- Isolation via Git Worktrees : isolation: worktree pour travail parallèle sans conflits

MODULE

02.

1H55H

Mémoire persistante et portée des sous-agents

CONTENU PÉDAGOGIQUE

- Les 3 portées mémoire : user (~/.claude/agent-memory/), project (.claude/agent-memory/), local
- Chargement automatique des 200 premières lignes de MEMORY.md par sous-agent
- Portée et priorité : managed settings > CLI --agents > .claude/agents/ > ~/.claude/agents/ > plugin
- Invocation : langage naturel, @-mention garantie, --agent pour session complète
- Foreground vs Background : permissions, Ctrl+B, auto-deny en arrière-plan
- Atelier pratique : créer un sous-agent avec mémoire persistante et isolation worktree

Création de Skills personnalisés

CONTENU PÉDAGOGIQUE

- Architecture d'une Skill : SKILL.md, templates, exemples, scripts exécutables
- Emplacements et portées : entreprise > personnel > projet > plugin
- Frontmatter YAML complet : name, description, when_to_use, argument-hint, allowed-tools, model, effort
- Variables de substitution : \$ARGUMENTS, \$ARGUMENTS[N], \${CLAUDE_SESSION_ID}, \${CLAUDE_SKILL_DIR}
- Injection de contexte dynamique : syntaxe !`command` pour exécuter des commandes avant l'envoi
- Contrôle d'invocation : disable-model-invocation, user-invocable, cycle de vie du contenu

Skills avancés et exécution isolée

CONTENU PÉDAGOGIQUE

- Exécution dans un sous-agent : context: fork, choix de l'agent (Explore, Plan)
- Auto-compaction des skills : budget 25 000 tokens, 5 000 par skill, priorité à la plus récente
- Skills intégrées : /simplify, /batch, /debug, /loop, /claude-api — analyse et personnalisation
- Hooks scopés aux skills : PreToolUse, PostToolUse dans le frontmatter
- Patterns de paths : activation conditionnelle avec glob patterns
- Atelier complet : créer une skill /review-pr avec injection dynamique (gh pr diff, gh pr view)

Agent Teams (TeamCreate) : orchestration multi-agents

CONTENU PÉDAGOGIQUE

- Concept : team lead, teammates, task list partagée, système de mailbox inter-agents
- Teams vs Subagents : communication directe, auto-coordination, coût en tokens
- Démarrage d'une équipe : prompt en langage naturel, création automatique des coéquipiers
- Modes d'affichage : in-process (Shift+Down, Ctrl+T) et split panes (tmux/iTerm2)
- Tâches et coordination : pending/in progress/completed, dépendances, file locking, self-claim
- Approbation de plan obligatoire : lecture seule jusqu'à validation par le lead
- Bonnes pratiques : 3-5 coéquipiers, 5-6 tâches par agent, éviter les conflits de fichiers

Hooks : automatiser le cycle de vie

CONTENU PÉDAGOGIQUE

- Les 4 types de handlers : command (shell), http (endpoint), prompt (LLM), agent (sous-agent)
- Événements du cycle de vie : SessionStart/End, PreToolUse, PostToolUse, Notification, SubagentStart/Stop
- Événements équipe : TeammateIdle, TaskCreated, TaskCompleted
- Structure de configuration : Hook Event > Matcher Group > Hook Handler, patterns de matchers
- Codes de sortie : 0 (succès), 2 (bloquant), autre (non-bloquant)
- Cas d'usage : auto-formatage PostToolUse, blocage de commandes dangereuses, notifications desktop
- Atelier pratique : implémenter un pipeline hooks complet (validation, formatage, notification)

Claude Agent SDK : agents programmatiques

CONTENU PÉDAGOGIQUE

- Installation Python et TypeScript : `claude-agent-sdk`, `@anthropic-ai/claude-agent-sdk`
- Authentification : `ANTHROPIC_API_KEY`, Amazon Bedrock, Google Vertex AI, Microsoft Azure
- Exemple de base : `query()` asynchrone, `ClaudeAgentOptions`, streaming des messages
- Outils intégrés du SDK : `Read`, `Write`, `Edit`, `Bash`, `Monitor`, `Glob`, `Grep`, `WebSearch`, `WebFetch`, `AskUserQuestion`
- Hooks dans le SDK : fonctions callback Python/TypeScript (pas des commandes shell)
- Subagents dans le SDK : `AgentDefinition`, délégation programmatique
- Sessions et reprise : capture du `session_id`, resume avec contexte complet
- Intégration MCP dans le SDK : serveurs externes (Playwright, outils custom)

Pipelines CI/CD, automatisation et bonnes pratiques équipes

CONTENU PÉDAGOGIQUE

- Mode headless (-p) : intégration GitHub Actions, pipelines automatisés, output structuré JSON
- Flags d'automatisation : `--max-turns`, `--max-budget-usd`, `--json-schema`, `--output-format stream-json`
- Tâches planifiées : `cloud scheduled tasks`, Desktop App, GitHub Actions, `/loop`, `/schedule`
- Boucles autonomes avec `/loop` : polling récurrent, monitoring continu
- `CLAUDE.md` d'équipe : organisation `.claude/rules/`, imports `@path`, cohérence multi-développeurs
- Gestion avancée du contexte : auto-compaction (`CLAUDE_AUTOCOMPACT_PCT_OVERRIDE`), Extended Thinking adaptatif
- Quiz de validation des acquis, atelier final multi-agents 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 —