Cours 1

Claude Code en Action

Anthropic Academy — Resume detaille

Yoann Boulch | 01/04/2026

1. Qu'est-ce que Claude Code ?

Claude Code est un outil CLI révolutionnaire développé par Anthropic qui intègre l'intelligence artificielle générative directement dans votre flux de travail de développement. Il combine la puissance de Claude avec votre terminal, permettant une programmation agentique où l'IA devient un véritable partenaire de codage.

Caractéristiques principales:

- CLI intégré : Fonctionne directement depuis votre terminal
- Programmation agentique : Claude peut prendre des décisions autonomes sur la base du contexte
- Intégration terminale native : Accès direct aux commandes bash et outils système
- Gestion de contexte avancée : Comprend vos fichiers et structure de projet

Architecture de Claude Code:

Flux de Claude Code

2. Installation et configuration

L'installation de Claude Code est simple et rapide. Vous aurez besoin de Node.js et d'une clé API Anthropic pour commencer.

Étapes d'installation:

```
$ npm install -g @anthropic-ai/claude-code
```

Configuration de l'authentification:

```
$ claude auth --setup
$ export ANTHROPIC_API_KEY='votre-clé-api'
```

Premier lancement:

```
$ claude --version
$ claude init mon-projet
```

3. Gestion du contexte

La gestion du contexte est au cœur de Claude Code. Les fichiers CLAUDE.md permettent de définir le contexte à différents niveaux : personnel, projet et répertoire.

Niveaux de contexte:

- Personnel (~/.claude/CLAUDE.md) : Préférences globales
- Projet (./CLAUDE.md) : Instructions spécifiques au projet
- Répertoire (.claude/CLAUDE.md) : Contexte local au répertoire

Chargement du contexte:

Claude Code charge les contextes dans l'ordre de spécificité croissante. Le contexte personnel est chargé en premier, suivi du contexte projet, puis du contexte répertoire.

4. Outils principaux

Claude Code fournit plusieurs outils pour interagir avec votre système et vos fichiers.

- Read : Lire le contenu de fichiers
- Write : Créer ou remplir entièrement des fichiers
- Edit : Modifier des parties spécifiques de fichiers
- Bash : Exécuter des commandes shell
- Glob : Trouver des fichiers par pattern
- Grep : Rechercher du contenu dans les fichiers

Comparaison des outils principaux:

Outil	Objectif	Cas d'usage
Read	Lire fichiers	Consulter code existant
Write	Créer fichiers	Nouveaux fichiers
Edit	Modifier	Changements précis
Bash	Exécuter	Commandes système
Glob	Trouver	Recherche par pattern
Grep	Chercher	Contenu spécifique

5. Commandes slash personnalisées

Claude Code permet de créer des commandes personnalisées qui encapsulent des workflows courants et améliorent la productivité.

Créer une commande personnalisée:

```
Les commandes sont définies dans .claude/commands.json
```

Exemple de configuration:

```
{ "commands": { "review": { "description": "Examine le code", "behavior": "analyze" } } }
```

6. Intégration MCP Server

Les serveurs MCP (Model Context Protocol) permettent d'étendre les capacités de Claude en connectant des outils externes.

Configuration MCP:

```
Les serveurs MCP sont configurés dans ~/.claude/settings.json
```

Serveurs MCP populaires:

- GitHub : Gestion des PRs et issues
- Slack : Intégration avec Slack
- Google Drive : Accès aux fichiers Drive
- Jira : Gestion de projets Jira

7. Intégration GitHub

Claude Code s'intègre profondément avec GitHub, permettant des revues de code, la gestion des issues et des workflows de commit automatisés.

Workflows GitHub:

- Revue de PRs : Analyse automatique des pull requests
- Gestion d'issues : Créer et résoudre des issues
- Workflows de commit : Commits structurés avec messages détaillés

Exemple : Créer une PR

```
claude create-pr --branch feature --title 'Nouvelle fonctionnalité'
```

8. Système de hooks

Les hooks permettent d'exécuter du code à des moments clés du processus d'IA, offrant un contrôle fin sur le comportement de Claude Code.

Types de hooks disponibles:

- PreToolUse : Avant l'exécution d'un outil
- PostToolUse : Après l'exécution d'un outil
- Notification : Notifications de progression
- Stop : Arrêt de l'exécution

Exemple de hook PreToolUse:

```
hooks: PreToolUse: 'Valider que le fichier existe'
```

9. Claude Code SDK (Agent SDK)

Le SDK permet une utilisation programmatique de Claude Code pour créer des agents personnalisés et des workflows automatisés.

Utilisation du SDK:

```
const { Agent } = require('@anthropic-ai/sdk'); const agent = new Agent({ model:
'claude-3-5-sonnet', tools: ['read', 'write', 'bash'] });
```

Sous-processus et délégation:

- Créer des agents enfants pour des tâches parallèles
- Isolation du contexte entre agents
- Communication inter-agents

Avantages du SDK:

■ Le SDK permet une automatisation complète et une intégration profonde de Claude dans vos applications.

10. Bonnes pratiques et récapitulatif

Points clés à retenir:

- Utilisez la gestion du contexte pour adapter Claude à vos besoins
- Exploitez les outils principaux (Read, Write, Edit, Bash, Glob, Grep)
- Automatisez avec les commandes slash et les hooks
- Intégrez GitHub pour un contrôle de version intelligent
- Utilisez le SDK pour des cas d'usage avancés