

Claude Code training for developers

2 days (14 hours)

Overview

Our Claude Code training for developers will enable you to take full advantage of an AI assistant designed for code generation, understanding, and refactoring. With a focus on productivity, Claude Code integrates into teams' daily workflows (IDE, CI/CD, API) to accelerate development, enhance quality, and automatically document code bases.

This comprehensive program will provide you with an in-depth understanding of how Claude Code works, covering its advanced coding assistance capabilities, best practices for prompt engineering, and effective use for generating tests, documentation, and manipulating structured formats such as JSON and YAML.

You will also learn how to integrate Claude Code via API/SDK and into IDEs (VS Code/JetBrains), as well as how it works with orchestration tools (e.g., LangChain) to build intelligent workflows. Finally, the training will guide you through the creation and integration of custom tools (agents, scripts, pipelines) into your projects, while mastering best practices for security and API key management.

This training course will be conducted using an accessible version of the tool: [Claude Code](#), which has most of the advanced features useful for development.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Understand the advanced features of Claude Code and its positioning for developers

- Generate, refactor, and document code with Claude Code
- Interpret and manipulate structured formats (JSON, YAML) using Claude Code
- Effectively integrate Claude Code into modern applications via API, SDK, and IDE
- Orchestrate Claude Code with frameworks (e.g., LangChain) to create workflows and developer tools
- Implement generated tests, assisted reviews, and automated CI/CD pipelines
- Ensure secure API key management and apply security best practices specific to code-oriented LLMs

Target audience

- Developers
- AI engineers

Prerequisites

- Programming knowledge
- Familiarity with JSON formats and orchestration principles (e.g., LangChain)

Our Claude Code training program: Accelerate and secure development

[Day 1 - Morning]

Getting started with Claude Code and assisted generation

- Understanding Claude Code: positioning, use cases, and limitations in the development cycle
- Context engineering: prompt rules, context structuring, token management
- Plan Mode: building a reliable execution plan (planning, iterations, validation)
- Handling structured formats (JSON, YAML) and producing usable documentation
- Best practices: refactoring, test generation, assisted code review, verification strategy
- Hands-on workshop: Initialize a mini-project, define a context, generate/refactor, and produce an initial set of tests.

[Day 1 - Afternoon]

Installation, configuration, and interfaces (CLI, IDE, Web)

- Access to Claude Code: CLI, plugins, IDE extensions
- Configuring the environment: tree structures, permissions, access policies (deny/allow/ask)

- Essential flags: --model, --system-prompt, session options
- Claude Code on the Web: launching tasks in the background and monitoring execution
- Provider support: Azure AI Foundry
- Hands-on workshop: Install and configure the tool, secure permissions, run a CLI task and a web task.

Interactive chat, slash commands, and productivity

- Chat modes: iterations, constraints, style guidelines, file calls, and navigation
- Slash commands: init, add-dir, context, status, memory, review, todos, cost
- Using the shell to automate tasks
- New workflows: /export and /resume commands
- Hands-on workshop: Create a team slash command and apply it to a real feature.

[Day 2 - Morning]

Plugins, MCP, and tool integrations

- Plugin concepts: creation, distribution, and real-world examples
- MCP: concepts, adding, listing, use cases, limitations, and best practices
- Advanced MCP: remote servers (SSE/HTTP), OAuth support, MCP resources referenced by @mention
- MCP display improvements: titles, annotations, and tools that are easier to read in the /mcp view
- CI/CD: Claude Code GitHub Actions v1.0, migration from beta and automatic detection of execution mode
- Hands-on workshop: Connecting a tooling MCP and automating review and documentation generation in a CI pipeline.

[Day 2 - Afternoon]

Agent Skills, sub-agents, and Hooks

- Sub-agents: creation, orchestration, context isolation, and task delegation
- Agent Skills: creating a skill and choosing the right mechanism
- Automatic discovery of skills: personal, project, plugins, and monorepos
- Hot reload of Skills: immediate application of changes without restarting
- Hooks: automating validations and governing tool usage
- Hands-on workshop: Create a pull request review skill and a QA subagent, then add a permission control hook.

Security, governance, and production deployment of agent workflows

- Key and secret management: best practices, compartmentalization, rotation, and auditing
- Code-oriented LLM security: risks, sensitive data, access policies, and traceability
- Skill security: auditing third-party skills, risks of malicious scripts, and prompt injection prevention

- Costs and management: reading/usage, monitoring, optimization, and implementation of safeguards
- Agentic ecosystem: understanding the shift towards Cowork-type tools and their impact on workflows
- Hands-on workshop: Building a complete workflow, with security rules, permission control, and cost tracking and cost monitoring.

Companies concerned

This training is aimed at both individuals and companies, large or small, wishing to train their teams in new advanced IT technology or to acquire specific professional knowledge or modern methods.

Placement at the start of training

The placement test at the start of the training course complies with Qualiopi quality criteria. Once they have finalized their registration, learners receive a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training course, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and reflection sessions, and group work.

Validation

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training course.