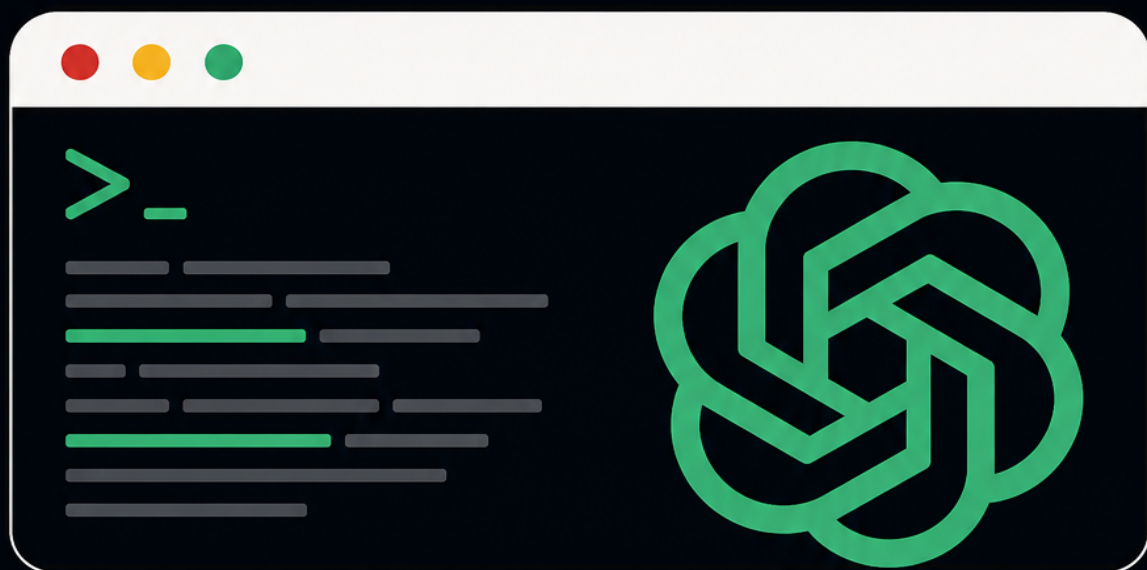


CODEX

CLI

THE DEFINITIVE GUIDE

FROM FIRST PROMPT TO **PRODUCTION WORKFLOWS**



CLI
MASTERY



SKILLS &
AUTOMATION



MCP
INTEGRATION



SUBAGENTS
& TEAMS



WORKFLOWS
& PIPELINES



SAFE, SECURE
& SCALABLE

GREG HARTMANN

Codex CLI: The Definitive Guide

From First Prompt to Production Workflows

Steve Publications

This book is available at <https://leanpub.com/codexclithedefinitiveguide>

This version was published on 2026-07-18



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Steve Publications

Contents

- From First Command to Production Workflows 1**
 - About This Book 1
- Introduction 2**
 - What This Book Covers 2
 - Who This Book Is For 3
 - How to Read This Book 3
 - A Note on Safety 3
- Chapter 1: What Is Codex CLI? 5**
 - From TypeScript to Rust: The Evolution of Codex 5
 - Architecture Overview: How Codex Works Under the Hood 6
 - The Five Surfaces: CLI, Desktop, Web, IDE Extension, and Cloud 7
 - When to Use Codex CLI (and When Not To) 8
 - Ecosystem Context: Codex vs. Claude Code, Cursor, Copilot 9
 - Acknowledging Codex CLI’s Shortcomings 10
- Chapter 2: Installation and First Run 12**
 - System Requirements and Prerequisites 12
 - Installing via npm, Homebrew, Install Script, or Binary 12
 - Windows Installation: Native and WSL2 14
 - Verifying Your Installation 15
 - Your First Interactive Session 16
- Chapter 3: Authentication and Credential Management 17**
 - ChatGPT OAuth Sign-In Flow 17
 - API Key Authentication 17
 - Enterprise Access Tokens 17
 - Credential Storage: File vs. Keyring 17
 - CI/CD Authentication Patterns 17
 - Troubleshooting Auth Failures 17

Chapter 4: Configuration Deep Dive 19

 Config File Locations and Precedence 19

 Core Settings: Model, Sandbox, Approval Policy 19

 Feature Flags and Experimental Capabilities 19

 Profiles for Different Workflows 19

 Custom Model Providers (Azure, Bedrock, Ollama, OpenRouter) 19

 Enterprise Managed Configuration 19

Chapter 5: The Interactive Terminal Experience 21

 Starting an Interactive Session 21

 Slash Commands Reference 21

 Keyboard Shortcuts and Navigation 21

 Image Input and Visual Context 21

 Session Management: Resume, Clear, and Compact 21

 TUI Customization and Themes 21

Chapter 6: Sandboxing, Permissions, and Security 23

 Sandbox Modes: read-only, workspace-write, danger-full-access . . . 23

 Approval Policies: untrusted, on-request, never, granular 23

 Auto-Review: Delegating Trust to an AI Reviewer 23

 Network Access Controls and Proxy Configuration 23

 OS-Level Sandboxing: Seatbelt, bwrap, Windows Sandbox 23

 Security Best Practices and Common Pitfalls 23

Chapter 7: Project Context with AGENTS.md 25

 What Is AGENTS.md and Where It Lives 25

 Writing Effective Project Instructions 25

 Layered Context: Global, Project, and Override Files 25

 Context Window Management and Token Budgeting 25

 The .codexignore File 25

 Cross-Tool Portability with AGENTS.md 25

Chapter 8: Skills – Reusable Task Expertise 27

 What Are Skills and How They Work 27

 The SKILL.md Format and Directory Structure 27

 Skill Discovery: Implicit vs. Explicit Invocation 27

 Creating Skills with \$skill-creator 27

 Installing and Distributing Skills 27

 Best Practices for Writing Effective Skills 27

Chapter 9: Model Context Protocol (MCP) Integration	29
What Is MCP and Why It Matters	29
Configuring MCP Servers in config.toml	29
Managing Servers via codex mcp Commands	29
STDIO vs. Streamable HTTP Servers	29
Tool Control: Enabling, Disabling, and Approval Modes	29
Real-World MCP Server Recipes and Walkthroughs	29
Chapter 10: Subagents and Multi-Agent Orchestration	31
How Subagents Work: Architecture and Benefits	31
Triggering Subagents: Prompts, AGENTS.md, and Skills	31
Custom Agent Definitions in TOML	31
Parallel Execution Patterns and Git Worktrees	31
CSV Batching for High-Volume Tasks	31
Multi-Agent Best Practices and Cost Management	31
Chapter 11: Non-Interactive Mode and Automation	33
codex exec: Headless Execution	33
JSON Output and Schema Validation	33
Session Resumption in Automated Workflows	33
GitHub Actions Integration (openai/codex-action)	33
Scripting Patterns and Bash Wrappers	33
Stdin Piping and Prompt Composition	33
Chapter 12: Code Review and Git Integration	35
The /review Command: Modes and Scope	35
Pre-Commit Review Workflows	35
GitHub Pull Request Reviews with @codex review	35
Branch and Commit Inspection	35
Git Checkpoint Strategies	35
Custom Review Guidelines	35
Chapter 13: Codex Cloud and Remote Execution	37
What Is Codex Cloud and When to Use It	37
codex cloud Commands: exec, list, apply	37
Cloud Environment Setup Scripts and Container Caching	37
Remote SSH Connections	37
Best-of-N Attempts and Retry Strategies	37
Cloud vs. Local: Decision Framework	37

- Chapter 14: Model Selection and Reasoning Configuration 39**
 - The GPT-5.6 Family: Sol, Terra, Luna 39
 - Legacy Models: GPT-5.4, GPT-5.3-Codex, Spark 39
 - Known Model Limitations 39
 - Reasoning Effort Levels: none to ultra 39
 - Model Verbosity and Summary Modes 39
 - Cost vs. Capability Trade-Offs 39
 - Per-Agent and Per-Task Model Selection 40
- Chapter 15: Performance Optimization 41**
 - Understanding Your Token Budget 41
 - Context Compaction: How It Works 41
 - Reducing MCP Server Overhead 41
 - File Exclusion Strategies with .codexignore 41
 - Session Length Management 41
 - Cost Estimation and Monitoring 41
- Chapter 16: Troubleshooting and Common Pitfalls 43**
 - Authentication and Billing Errors 43
 - Sandbox and Permission Failures 43
 - Context Window Overflow 43
 - MCP Server Timeouts and Connection Issues 43
 - Installation and Environment Problems 43
 - The Diagnostic Mindset: Systematic Debugging 43
- Chapter 17: Real-World Workflows and Case Studies 45**
 - Solo Developer: Full-Stack Feature Implementation 45
 - Team Workflow: Code Review Pipeline Integration 45
 - CI/CD: Automated Testing and Bug Fixing 45
 - Legacy Codebase Refactoring 45
 - Documentation Generation and Maintenance 45
 - Security Audit Workflows 45
- Chapter 18: Migration Strategies – From Other Tools to Codex CLI . . . 47**
 - Why Migrate? The Business Case 47
 - Migrating from Claude Code 47
 - Migrating from Cursor 47
 - Migrating from GitHub Copilot 47
 - Migrating from Aider 47
 - Team Migration: Onboarding Without Disruption 47

Credential and Provider Transitions	48
When Not to Migrate	48
Chapter 19: Advanced Techniques and Future Directions	49
Codex as an MCP Server for External Orchestration	49
The OpenAI Agents SDK Integration	49
Record and Replay: Building Skills from Sessions	49
Plugins: Distributable Skill Bundles	49
Hook Systems and Event-Driven Automation	49
The Road Ahead: What to Watch For	49
Conclusion	51
References	52
Glossary	53
Index	54

From First Command to Production Workflows

About This Book

Codex CLI is OpenAI's terminal-native coding agent, capable of reading codebases, editing files, running commands, and automating entire development workflows from the command line. This book takes you from installation through mastery, covering every layer of the platform: configuration, sandboxing, skills, MCP integration, subagents, automation pipelines, and real-world production patterns. Whether you are a solo developer looking to accelerate your coding workflow or a team lead building CI/CD pipelines with AI agents, this book gives you the knowledge to use Codex CLI safely, effectively, and at scale.

Introduction

In April 2025, OpenAI released Codex CLI as a TypeScript-based Node.js process that called the OpenAI API from your terminal. It was unassuming in its debut: a single binary, an Apache 2.0 license, and a README that described itself as “an experimental project under active development.” Within two months, OpenAI had already begun rewriting the core in Rust. By the summer of 2026, the rewrite was complete, native sandboxing worked on every major operating system, and the platform had grown into a full-fledged agentic coding system with multi-agent orchestration, Model Context Protocol integration, a skills ecosystem, lifecycle hooks, and cloud execution environments. The repository on GitHub has accumulated approximately 99,200 stars and nearly 15,000 forks, making it one of the most widely used open-source developer tools in the AI coding ecosystem [1].

The speed of this evolution is worth pausing on. Most developer tools take years to mature from prototype to production platform. Codex CLI compressed that timeline into months, driven by a combination of factors: OpenAI’s internal investment in the project, the competitive pressure from Claude Code and other terminal-native agents, and a community of contributors who treated the codebase as their own. The result is a tool that is powerful but still evolving, capable but occasionally rough around the edges.

This book exists because Codex CLI is not just a chatbot that happens to live in your terminal. It is a layered platform, and its power comes from understanding those layers: configuration, sandboxing, skills, MCP integration, subagents, and automation pipelines. Most users scratch the surface with basic prompts and never unlock the deeper capabilities that make Codex CLI a genuine force multiplier for software development.

What This Book Covers

The chapters progress logically from beginner to advanced. You will start with installation and authentication, then move through configuration, the interactive terminal experience, sandboxing, and security. From there you will

learn about project context management with AGENTS.md, the skills system, MCP server integration, subagent orchestration, non-interactive automation, code review workflows, cloud execution, model selection, performance optimization, troubleshooting, and real-world production patterns.

Each chapter is designed to be readable on its own while building on the concepts introduced earlier. If you are already comfortable with Codex CLI basics, you can jump directly to the chapters that match your needs. If you are starting from scratch, reading sequentially will give you the most complete understanding.

Who This Book Is For

This book assumes you are familiar with the command line, basic Git operations, and have some experience writing code. You do not need prior experience with AI coding agents, though familiarity with tools like GitHub Copilot or ChatGPT will help. The content is relevant for:

- **Individual developers** who want to accelerate their coding workflow
- **AI engineers** building and customizing agent systems
- **DevOps practitioners** integrating AI agents into CI/CD pipelines
- **Technical leaders** evaluating and deploying Codex CLI across teams
- **Security-conscious developers** who need to understand the safety model

How to Read This Book

Code examples use the `codex` command prefix. Configuration snippets show TOML format, which is Codex CLI's native configuration language. Commands that should be run in a terminal are shown in code blocks. Configuration file paths use `~/.codex/` for user-level settings and `.codex/` for project-scoped settings, following the convention Codex CLI uses on disk.

Version numbers referenced throughout the book reflect the state of Codex CLI as of July 2026, with the latest stable release being v0.144.x. Where features were introduced in specific versions, this is noted inline. The book references official OpenAI documentation at developers.openai.com/codex as the authoritative source for all technical claims.

A Note on Safety

Codex CLI can edit files, run commands, and interact with external systems. Throughout this book, safety is treated as a first-class concern, not an afterthought. The sandboxing and approval system exists to protect you from unintended consequences, and understanding how to configure these protections appropriately for your workflow is one of the most important skills you will develop. Never disable sandboxing or approvals on a production machine without understanding the risks.

The chapters that follow will give you everything you need to go from your first codex command to running sophisticated multi-agent workflows in production. Let us begin.

Chapter 1: What Is Codex CLI?

Codex CLI is OpenAI's terminal-native coding agent. It reads your codebase, edits files, runs commands, and automates development workflows directly from the command line. Unlike chat-based assistants that generate code snippets for you to copy and paste, Codex CLI operates as an autonomous agent within your local environment: it can inspect your repository, make changes, run tests, and iterate until a task is complete.

This chapter traces how Codex CLI evolved from a simple prototype into a sophisticated agentic platform, explains its internal architecture, and situates it within the broader developer tooling ecosystem.

From TypeScript to Rust: The Evolution of Codex

When OpenAI first released Codex CLI in April 2025, it was a TypeScript project running on Node.js. The original implementation was straightforward: a process that accepted natural language prompts, called the OpenAI API, and executed commands in your terminal. It worked, but it had limitations. The JavaScript runtime made sandboxing difficult because Node.js subprocess management does not provide the same level of isolation as compiled binaries. Performance was inconsistent with large codebases due to JavaScript's garbage collection pauses. And the architecture did not scale well to more complex agent behaviors like parallel subagent execution or efficient file system monitoring.

OpenAI responded by rewriting Codex CLI from scratch in Rust. The rewrite, which began in mid-2025 and was effectively complete by early 2026, was not cosmetic. Rust's memory safety guarantees eliminated an entire class of bugs related to buffer overflows and use-after-free errors. Compiled performance removed garbage collection pauses and reduced memory consumption. And Rust's mature ecosystem for system-level programming enabled capabilities that were impractical in the TypeScript version: native sandboxing on every major operating system, efficient file system monitoring using platform-native APIs, parallel subagent execution with proper resource

management, and a more robust event-driven architecture built on async Rust [4,14].

The decision to rewrite was also strategic. An open-source Rust codebase signals serious commitment to performance and transparency, distinguishing Codex CLI from cloud-only alternatives and making it easier for security-conscious organizations to audit the agent's behavior before deployment.

By mid-2026, the codebase is approximately 95 to 96 percent Rust, with the remainder consisting of Python tooling scripts and build configuration. The exact percentage varies slightly between versions as new components are added, but the Rust rewrite is effectively complete. The project is open source under the Apache 2.0 license and hosted on GitHub at github.com/openai/codex, where it has accumulated approximately 99,200 stars and nearly 15,000 forks as of July 2026 [1]. The release cadence is rapid: the project ships dozens of versions each month, with alpha releases available for early testing and stable releases vetted before general availability. As of this writing, the latest stable release is v0.144.6 (July 18, 2026), with active alpha development on the v0.145.x branch [1].

Architecture Overview: How Codex Works Under the Hood

Codex CLI's architecture can be understood as five layers that interact in a loop:

The Terminal UI Layer. The TUI (Text User Interface) presents a full-screen interactive session where you type prompts and see Codex's responses, file edits, command outputs, and approval requests. It supports keyboard shortcuts, slash commands, image input, and customizable themes. When running non-interactively via `codex exec`, this layer is replaced by JSONL event streaming to stdout and progress messages to stderr.

The Agent Core. This is the decision-making engine. Built in Rust, it manages the agentic loop: receiving your prompt, consulting context (AGENTS.md, skills, MCP tools), selecting actions, executing tools, and iterating until the task is complete. The agent core handles session state, context compaction, and reasoning effort configuration. A source-level analysis of the codebase (549K lines of Rust as of mid-2026) reveals a queue-pair protocol (Op-Event channels) for internal communication and a Guardian AI approval system that uses a separate LLM to judge tool calls [18].

The Tool Layer. Codex CLI has a set of built-in tools for file operations (read, write, edit, apply patches), shell command execution, Git operations, web search, and image analysis. These tools are sandboxed by default and subject to approval policies. Additional capabilities come from MCP servers, which expose third-party tools through the Model Context Protocol.

The Model Layer. Codex CLI connects to OpenAI's API (or custom providers) to run language models. The model selection is configurable per session, per agent, or via command-line flags. As of July 2026, the primary models available include the GPT-5.6 family (Sol, Terra, Luna), with GPT-5.4 and GPT-5.3-Codex still supported in API-key mode [5]. The model layer handles token management, reasoning effort levels, and response formatting.

The Sandbox Layer. This is the security boundary. On macOS, Codex CLI uses Seatbelt for sandbox enforcement. On Linux and WSL2, it uses bubblewrap (bwrap) with seccomp filters and Landlock. On Windows, it uses restricted tokens, ACLs, and firewall rules [3,6]. The sandbox layer enforces file system boundaries, network access restrictions, and command execution policies based on your configuration.

These layers interact in a continuous loop during an interactive session: you submit a prompt, the agent core consults context and selects actions, tools execute within the sandbox, results are returned to the model for analysis, and the cycle repeats until the task is complete or you intervene.

The Five Surfaces: CLI, Desktop, Web, IDE Extension, and Cloud

Codex CLI is one of five surfaces in the broader Codex platform. Understanding how they relate helps you choose the right tool for each task.

Codex CLI runs locally in your terminal. It gives you the most control over configuration, sandboxing, and automation. It is the primary surface for developers who want to work within their existing command-line workflow and for CI/CD integration.

The ChatGPT Desktop App provides a graphical interface with Codex mode. It shares configuration files with the CLI (both read from `~/.codex/config.toml`) and supports skills, MCP servers, and subagents. It is useful for developers who prefer a windowed application over a terminal but want the same underlying capabilities.

Codex Web runs at chatgpt.com/codex in your browser. This is the cloud-based agent surface where tasks run in remote sandboxed environments. It requires ChatGPT sign-in and is ideal for long-running tasks that should not execute on your local machine.

The IDE Extension integrates Codex into VS Code, Cursor, Windsurf, and other editors. It provides inline code generation, chat panels, and terminal integration within the editor environment. The extension shares MCP configuration with the CLI but has its own settings for editor-specific behavior.

Codex Cloud is the remote execution platform accessible via `codex cloud` commands from the CLI. It spins up containerized environments, runs tasks in parallel, and returns results as diffs or artifacts. Cloud environments support custom setup scripts, environment variables, secrets, and container caching.

Power users often combine these surfaces: using the CLI for deterministic repository operations, the desktop app for planning and coordination, cloud environments for tasks requiring internet access or heavy computation, and the IDE extension for tight coding loops within their editor.

When to Use Codex CLI (and When Not To)

Codex CLI excels at tasks that benefit from terminal-native interaction and automation. Strong use cases include:

- **Exploring unfamiliar codebases.** Ask Codex to explain architecture, find relevant files, and trace dependencies.
- **Implementing features end to end.** Describe what you want built and let Codex scaffold files, write code, run tests, and iterate.
- **Refactoring at scale.** Instruct Codex to rename symbols across a project, extract modules, or apply consistent patterns.
- **Code review.** Use the `/review` command to get prioritized findings on uncommitted changes before you commit.
- **CI/CD automation.** Run `codex exec` in pipelines for automated testing, bug fixing, and documentation generation.
- **Multi-agent workflows.** Spawn parallel subagents for independent tasks like security scanning, test analysis, and dependency review.

Codex CLI is less well suited for:

- **Tasks requiring a graphical interface.** UI design, visual debugging, and browser-based workflows are better handled by the desktop app or Codex Web.
- **Real-time collaborative editing.** Codex CLI is designed for single-user sessions; multi-person real-time collaboration is not its strength.
- **Simple one-off questions.** If you just need to look up a syntax detail, a quick web search or documentation lookup may be faster than starting a Codex session.

Ecosystem Context: Codex vs. Claude Code, Cursor, Copilot

The AI coding landscape in 2026 includes several major tools, each with different strengths and design philosophies. Understanding where Codex CLI fits helps you make informed choices about your toolchain. The competition is fierce, and the relative strengths of each tool shift as models improve and features ship. What follows reflects the state of play as of mid-2026.

Claude Code (by Anthropic) is the closest competitor to Codex CLI. Both are terminal-native agents that read codebases, edit files, and run commands. Claude Code uses Anthropic's Claude models and has its own skills system (also based on SKILL.md files). The agent skills open standard means many skills work across both tools [14]. Key differences: Codex CLI is open source (Apache 2.0), supports ChatGPT subscription billing in addition to API-key billing, and has deeper integration with the OpenAI ecosystem including cloud environments. In blind community tests involving 500+ Reddit developers, Claude Code achieved a 67 percent win rate on code quality, but Codex was rated more usable as a daily driver due to fewer usage limit frustrations [29].

Cursor is an AI-native code editor built on VS Code. It provides inline code generation, chat panels, and agent modes within a graphical IDE. Cursor does not have a standalone CLI mode but offers terminal integration within the editor. Choose Cursor if you prefer a full IDE experience; choose Codex CLI if you want terminal-first interaction and automation capabilities.

GitHub Copilot is Microsoft's AI coding assistant, available as a VS Code extension and through the GitHub Copilot Chat interface. It excels at inline code completion and contextual suggestions but does not have the autonomous

agent capabilities of Codex CLI (running commands, editing multiple files independently, multi-agent orchestration). Many developers use both: Copilot for inline assistance during coding and Codex CLI for larger-scale tasks.

Other contenders include Gemini CLI (Google), Aider, OpenCoder (now at 172K+ GitHub stars), and various open-source alternatives. The common trend across all these tools is the adoption of shared standards: AGENTS.md for project instructions, SKILL.md for reusable skills, and MCP for tool integration. This standardization means skills and configurations you build for Codex CLI often work with other agents as well.

The choice between tools often comes down to model preference, billing model, ecosystem integration, and workflow style. Codex CLI's strengths lie in its open-source nature, robust sandboxing, ChatGPT subscription support, cloud execution environments, and rapid release cadence driven by a large community of contributors.

Acknowledging Codex CLI's Shortcomings

A fair assessment requires honest discussion of where Codex CLI falls short. The community has been vocal about several persistent issues:

Usage limits and quota management. Perhaps the most widespread complaint concerns rate limits and usage caps. Since May 2026, many users reported that weekly quotas drain faster than expected, with some Plus-tier users exhausting their limits in two to three days of moderate use [3,19]. OpenAI introduced a “rate-limit reset banking” feature in June 2026, granting one free reset per cycle for Plus and Pro users, but the fundamental tension between generous access and sustainable capacity remains unresolved [3]. For heavy daily users, API-key billing with pay-as-you-go pricing may be more predictable than subscription-based access.

Performance regressions. Some users have reported performance degradation with newer GPT-5 models compared to earlier versions, particularly in code generation quality for certain task types. While OpenAI's benchmarks show improvement on standardized tests, anecdotal reports from the community suggest mixed results on real-world projects [2]. The rapid release cadence means features can be introduced before they are fully polished.

Web latency and stability. Starting in May 2026, a cluster of degradation records appeared on OpenAI's status page related to Codex Web response

latency, with parallel complaints about slowness, reconnect loops, and stream disconnects [9]. The CLI is less affected since it uses direct API connections, but cloud-based workflows can experience intermittent instability.

Experimental status. OpenAI's own README describes Codex CLI as “an experimental project under active development” that is “not yet stable” and “may contain bugs.” This is not marketing humility; the project genuinely ships breaking changes, deprecated features, and configuration shifts with regular frequency. Users who build automation around Codex CLI should expect to adapt their workflows as the platform evolves.

Missing features compared to competitors. Claude Code currently offers more sophisticated hooks (13 lifecycle events versus Codex's 5), a more mature skills marketplace, and in blind community tests, some developers report higher code quality from Claude's models [6]. These gaps may close over time, but they are real trade-offs to consider when choosing your primary tool.

Understanding these limitations does not diminish Codex CLI's value. It simply provides a realistic baseline from which to make informed decisions about where and how to deploy it.

Chapter 2: Installation and First Run

Before you can use Codex CLI, you need to install it and complete the authentication flow. This chapter covers every installation method on all supported platforms, from the recommended one-liner to manual binary installation for constrained environments. By the end of this chapter, you will have a working Codex installation and have completed your first interactive session.

System Requirements and Prerequisites

Codex CLI runs on macOS, Linux, and Windows. The minimum system requirements are:

- **macOS:** Version 12.0 (Monterey) or later
- **Linux:** Any modern distribution with glibc 2.28+ (Ubuntu 20.04+, Debian 10+, Fedora 31+, etc.)
- **Windows:** Windows 10 version 2004 or later, or WSL2 (Windows Subsystem for Linux)
- **Memory:** At least 4 GB RAM recommended (Codex itself is lightweight, but the models it calls may require significant memory for context processing)
- **Disk space:** Approximately 50 MB for the binary and dependencies
- **Network:** Internet connection required for API access and model inference

Codex CLI does not require Node.js or Python to run, since the production binary is compiled Rust. However, if you install via npm, Node.js and npm must be available. Git is not strictly required but is strongly recommended, as Codex CLI defaults to operating within Git repositories and many features (code review, checkpoints, session management) rely on Git.

Installing via npm, Homebrew, Install Script, or Binary

There are four primary installation methods. Choose the one that best fits your environment.

Method 1: npm (Recommended for most users)

The npm package is the most straightforward installation method if you already have Node.js installed:

```
1 npm install -g @openai/codex
```

This installs the latest stable version globally. To upgrade to a newer version:

```
1 npm install -g @openai/codex@latest
```

To check your installed version:

```
1 codex --version
```

The npm method works on macOS, Linux, and Windows. On Windows, you may need to run the command in an elevated PowerShell or Command Prompt if you encounter permission errors.

Method 2: Homebrew (macOS and Linux)

If you use Homebrew, install Codex CLI as a cask on macOS or a formula on Linux:

```
1 # macOS
2 brew install --cask codex
3
4 # Linux
5 brew install codex
```

Homebrew keeps the installation isolated from your system and makes upgrades simple:

```
1 brew upgrade codex
```

Method 3: Install Script (macOS and Linux)

OpenAI provides a one-line install script that downloads and installs the latest release:

```
1 curl -fsSL https://chatgpt.com/codex/install.sh | sh
```

This script detects your platform, downloads the appropriate binary, and places it in a directory on your PATH. It is designed for interactive use and includes safety checks to verify the download. For production or CI environments, prefer downloading the binary directly from GitHub Releases so you can pin a specific version.

Method 4: Pre-built Binary (All platforms)

For maximum control, download the pre-built binary from the GitHub Releases page. Available builds include:

- **macOS:** `codex-aarch64-apple-darwin.tar.gz` (Apple Silicon) or `codex-x86_64-apple-darwin.tar.gz` (Intel)
- **Linux:** `codex-x86_64-unknown-linux-gnu.tar.gz`
- **Windows:** `codex-x86_64-pc-windows-msvc.zip`

Example for Linux x86_64:

```
1 wget https://github.com/openai/codex/releases/download/v0.144.6/codex-x86_64-
  ↳ unknown-linux-gnu.tar.gz
2 tar -xzf codex-x86_64-unknown-linux-gnu.tar.gz
3 sudo mv codex /usr/local/bin/
4 sudo chmod +x /usr/local/bin/codex
5 codex --version
```

Binary installation is useful for CI/CD environments, systems where npm is not available, or when you need to pin a specific version for reproducibility.

Windows Installation: Native and WSL2

Windows users have two options: native Windows support and WSL2.

Native Windows. Starting with the Rust rewrite, Codex CLI runs natively on Windows without requiring WSL. Install via npm, the install script (PowerShell), or by downloading the Windows binary. The native Windows build supports all core features including sandboxing through Windows Sandbox integration.

To install via PowerShell:

```
1 # Using winget
2 winget install OpenAI.Codex
3
4 # Or using npm
5 npm install -g @openai/codex
```

WSL2. If you are already using WSL2 for development, running Codex CLI inside the Linux distribution gives you access to the full Linux toolchain and native sandboxing via bubblewrap. Install using any of the Linux methods described above, running the commands within your WSL2 terminal.

WSL2 is recommended if your development workflow already uses Linux tools extensively. Native Windows is preferable if you want to work directly with Windows file paths and tools.

Verifying Your Installation

After installation, verify that Codex CLI is working correctly:

```
1 codex --version
```

You should see output similar to `codex-cli 0.144.6`. If you see “command not found,” check that the installation directory is on your PATH. For npm installations, run `npm config get prefix` to find where global packages are installed, then add that bin directory to your PATH if it is not already included.

Generate shell completions for your shell:

```
1 # Bash
2 codex completions bash >> ~/.bashrc
3
4 # Zsh
5 codex completions zsh >> ~/.zshrc
6
7 # Fish
8 codex completions fish > ~/.config/fish/completions/codex.fish
```

These completions provide tab-completion for codex subcommands, flags, and options, making the CLI more pleasant to use.

Your First Interactive Session

Navigate to a project directory and run:

```
1 cd ~/your-project
2 codex
```

On first launch, Codex will prompt you to authenticate. Choose “Sign in with ChatGPT” if you have a ChatGPT Plus, Pro, Business, or Enterprise subscription, or choose API key authentication if you prefer usage-based billing. After authentication, the TUI appears with a welcome screen showing the current model and working directory.

Try a simple task:

```
1 Explain the architecture of this project. What are the main modules and how do
  ↳ they interact?
```

Codex will read your repository structure, analyze key files, and provide an explanation. You can follow up with more specific questions, ask for code changes, or use slash commands like `/status` to see your current configuration.

When you are done, exit the session by typing `Ctrl+C` or entering `/exit`. Your session is saved automatically, and you can resume it later with `codex resume`.

Before moving on to the next chapter, take a moment to explore the TUI. Try the `/model` command to see available models, `/permissions` to check your sandbox settings, and `/help` to see all available slash commands. Familiarity with these basics will make the deeper topics in later chapters easier to follow.

Chapter 3: Authentication and Credential Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

ChatGPT OAuth Sign-In Flow

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

API Key Authentication

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Enterprise Access Tokens

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Credential Storage: File vs. Keyring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

CI/CD Authentication Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Troubleshooting Auth Failures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 4: Configuration Deep Dive

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Config File Locations and Precedence

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Core Settings: Model, Sandbox, Approval Policy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Feature Flags and Experimental Capabilities

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Profiles for Different Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Custom Model Providers (Azure, Bedrock, Ollama, OpenRouter)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Enterprise Managed Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 5: The Interactive Terminal Experience

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Starting an Interactive Session

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Slash Commands Reference

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Keyboard Shortcuts and Navigation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Image Input and Visual Context

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Session Management: Resume, Clear, and Compact

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

TUI Customization and Themes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 6: Sandboxing, Permissions, and Security

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Sandbox Modes: read-only, workspace-write, danger-full-access

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Approval Policies: untrusted, on-request, never, granular

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Auto-Review: Delegating Trust to an AI Reviewer

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Network Access Controls and Proxy Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

OS-Level Sandboxing: Seatbelt, bwrap, Windows Sandbox

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Security Best Practices and Common Pitfalls

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 7: Project Context with AGENTS.md

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

What Is AGENTS.md and Where It Lives

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Writing Effective Project Instructions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Layered Context: Global, Project, and Override Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Context Window Management and Token Budgeting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The .codexignore File

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Cross-Tool Portability with AGENTS.md

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 8: Skills – Reusable Task Expertise

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

What Are Skills and How They Work

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The SKILL.md Format and Directory Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Skill Discovery: Implicit vs. Explicit Invocation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Creating Skills with \$skill-creator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Installing and Distributing Skills

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Best Practices for Writing Effective Skills

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 9: Model Context Protocol (MCP) Integration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

What Is MCP and Why It Matters

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Configuring MCP Servers in config.toml

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Managing Servers via codex mcp Commands

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

STDIO vs. Streamable HTTP Servers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Tool Control: Enabling, Disabling, and Approval Modes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Real-World MCP Server Recipes and Walkthroughs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 10: Subagents and Multi-Agent Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

How Subagents Work: Architecture and Benefits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Triggering Subagents: Prompts, AGENTS.md, and Skills

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Custom Agent Definitions in TOML

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Parallel Execution Patterns and Git Worktrees

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

CSV Batching for High-Volume Tasks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Multi-Agent Best Practices and Cost Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 11: Non-Interactive Mode and Automation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

codex exec: Headless Execution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

JSON Output and Schema Validation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Session Resumption in Automated Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

GitHub Actions Integration (openai/codex-action)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Scripting Patterns and Bash Wrappers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Stdin Piping and Prompt Composition

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 12: Code Review and Git Integration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The /review Command: Modes and Scope

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Pre-Commit Review Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

GitHub Pull Request Reviews with @codex review

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Branch and Commit Inspection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Git Checkpoint Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Custom Review Guidelines

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 13: Codex Cloud and Remote Execution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

What Is Codex Cloud and When to Use It

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

codex cloud Commands: exec, list, apply

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Cloud Environment Setup Scripts and Container Caching

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Remote SSH Connections

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Best-of-N Attempts and Retry Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Cloud vs. Local: Decision Framework

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 14: Model Selection and Reasoning Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The GPT-5.6 Family: Sol, Terra, Luna

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Legacy Models: GPT-5.4, GPT-5.3-Codex, Spark

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Known Model Limitations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Reasoning Effort Levels: none to ultra

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Model Verbosity and Summary Modes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Cost vs. Capability Trade-Offs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Per-Agent and Per-Task Model Selection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 15: Performance Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Understanding Your Token Budget

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Context Compaction: How It Works

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Reducing MCP Server Overhead

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

File Exclusion Strategies with .codexignore

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Session Length Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Cost Estimation and Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 16: Troubleshooting and Common Pitfalls

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Authentication and Billing Errors

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Sandbox and Permission Failures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Context Window Overflow

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

MCP Server Timeouts and Connection Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Installation and Environment Problems

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The Diagnostic Mindset: Systematic Debugging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 17: Real-World Workflows and Case Studies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Solo Developer: Full-Stack Feature Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Team Workflow: Code Review Pipeline Integration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

CI/CD: Automated Testing and Bug Fixing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Legacy Codebase Refactoring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Documentation Generation and Maintenance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Security Audit Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 18: Migration Strategies – From Other Tools to Codex CLI

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Why Migrate? The Business Case

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Migrating from Claude Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Migrating from Cursor

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Migrating from GitHub Copilot

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Migrating from Aider

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Team Migration: Onboarding Without Disruption

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Credential and Provider Transitions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

When Not to Migrate

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Chapter 19: Advanced Techniques and Future Directions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Codex as an MCP Server for External Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The OpenAI Agents SDK Integration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Record and Replay: Building Skills from Sessions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Plugins: Distributable Skill Bundles

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Hook Systems and Event-Driven Automation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

The Road Ahead: What to Watch For

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

References

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Glossary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.

Index

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/codexclithedefinitiveguide>.