

QUICK INSTALL

Windows

```
node -v
npm -v
npm install -g @openai/codex
codex --version
codex
```

Install Node.js 18+ from nodejs.org first if `node -v` fails.

Mac

```
brew install node
node -v
npm -v
npm install -g @openai/codex
codex --version
codex
```

Already have Node? Skip straight to the `npm install` line.

REQUIREMENTS CHECKLIST

- Windows 10/11 or recent macOS
- Latest stable VS Code
- Node.js 18+ and npm
- ChatGPT account with Codex access
- Homebrew (recommended, Mac)
- Enterprise: workspace admin approval may be required

Codex spans Free, Go, Plus, Pro, Business, Edu & Enterprise plans — usage limits vary by plan.

FIRST PROMPTS — BAD VS. GOOD

AVOID

Build me an app.

USE INSTEAD, IN ORDER

Review this codebase and explain the architecture before making any changes.

Create a step-by-step implementation plan. Do not edit files yet.

Implement Step 1 only. Keep the change scoped and explain each modified file.

Review the diff and identify any risks, missing checks, or follow-up improvements.

5-STEP SETUP

1. **Install Node.js** — verify with `node -v`
2. **Install Codex CLI** — `npm install -g @openai/codex`
3. **Open the project folder** in VS Code — not a single file. `cd path/to/project && code .`
4. **Install/open the Codex extension** — or fall back to Codex CLI in the integrated terminal
5. **Sign in with ChatGPT** — browser auth, then confirm Codex is connected in VS Code

SAFE WORKFLOW LOOP

- 1 **Analyze** — understand the codebase first
- 2 **Plan** — implementation plan, no edits yet
- 3 **Approve** — review the plan before it touches files
- 4 **Implement** — one scoped step at a time
- 5 **Review** — read the Git diff, file by file
- 6 **Test** — lint, type-check, build, run tests
- 7 **Commit** — only after review passes

RECOMMENDED VS CODE LAYOUT

With extension: Explorer | Editor | Codex Panel — terminal docked at the bottom.

CLI fallback: Explorer | Editor, with a terminal running Codex.

Keep code, Git diff, terminal output, and Codex's responses visible at the same time — that's what makes review fast.

Common Mistakes

- Starting with a vague, open-ended prompt
- Letting Codex touch too many files in one pass
- Skipping the Git diff review
- Accepting new dependencies without a second look
- Working without project instructions (AGENTS.md)
- Skipping lint/build checks before commit
- Running Codex directly on a production branch

RELATED AIBX READS

[Claude Code for VS Code](#) · [Best AI Coding Tools](#) · [Claude vs ChatGPT for Coding](#) · [AI Agents Explained](#) · [Cursor vs Codex](#) · [Enterprise AI Development](#)

PROMPT TEMPLATES

Architecture Review

Review this project and explain the architecture. Identify routes, components, data sources, and deployment assumptions. Do not edit files.

Debugging

Find the source of this error and propose the smallest safe fix. Explain the cause first.

Refactoring

Refactor this component for readability without changing behavior. Keep props and styling the same.

Feature

Add a [feature] using existing project patterns. Create a plan first, then wait for approval.

Code Review

Review this PR for maintainability, accessibility, performance, and security. Don't rewrite unless asked.

Testing

Identify missing tests for this feature and propose a test plan before writing any code.

Documentation

Document this feature: setup, usage, assumptions, and known limitations.

TROUBLESHOOTING

Issue	Fix
codex not found	Restart terminal & VS Code; check the npm global path; reinstall if needed
Node not found	Reinstall Node; restart terminal; verify <code>node -v / npm -v</code>
Sign-in fails	Confirm ChatGPT account & plan access; try another browser; check enterprise restrictions
Extension unavailable	Use Codex CLI in the VS Code integrated terminal instead
Permission/approval issues	Review approval settings & workspace file permissions
Enterprise restrictions	Workspace admin must enable access / connect GitHub for cloud workflows

AGENTS.MD STARTER

```
# Project Standards

## Stack
Next.js App Router, TypeScript,
React, Tailwind CSS, Vercel

## Development Rules
- Preserve existing architecture
- No new dependencies without approval
- Prefer reusable components
- Follow existing naming conventions
- Preserve accessibility & semantic HTML
- No unrelated changes

## Workflow Rules
- Analyze before editing
- Plan before changing files
- Approval before multi-file edits
- One scoped step at a time
- Explain each modified file

## Verification
TypeScript check, lint, build,
tests, Git diff review
```

TOOL COMPARISON

Tool	Best For	Style
Codex	Repo-aware tasks, planning, refactoring, review	Agentic workflow
ChatGPT	Strategy, explanation, architecture discussion	Conversational
GitHub Copilot	Inline suggestions, autocomplete	Continuous assistant
Claude Code	Agentic repo analysis & implementation	Terminal / agent

These tools evolve quickly — verify current features, pricing, and admin controls before standardizing on one.

ENTERPRISE GOVERNANCE

LOWER-RISK, OK WITH REVIEW

- Documentation & internal tooling
- Test scaffolding
- Refactoring, bug investigation
- Migration planning

HIGHER-RISK, EXTRA SCRUTINY

- Authentication & payments
- Production infrastructure
- Database migrations
- Security, permissions, compliance code

Required Controls

- Branch-based workflow + PR review
- CI validation, dependency approval
- Project instructions, scoped changes
- Human approval before merge