

# OPENAI CODEX CLI QUICK REFERENCE

CLI usage, modes, configuration, key commands

## Getting Started

### Install

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

### Authentication

```
export OPENAI_API_KEY="sk-..."
```

Set in shell profile or .env file

### First Run

```
codex "explain this project"
codex "add input validation to app.py"
```

## Commands

### Interactive (Default)

```
codex # start interactive session
codex "fix the login bug" # with initial prompt
```

### codex exec

```
codex exec "write unit tests for utils.py"
codex exec "refactor db.py to use async"
```

Non-interactive — runs task to completion

### codex review

```
codex review # review staged changes
codex review --diff HEAD-3 # review last 3 commits
```

## Modes

### Approval Modes

**suggest** Show proposed changes, require approval for every file edit and command

**auto-edit** Apply file edits automatically, require approval for shell commands

**full-auto** Apply edits and run commands without approval

### Setting the Mode

```
codex --approval-mode suggest "add tests"
codex --approval-mode auto-edit "refactor"
codex --approval-mode full-auto "fix lint"
```

## Configuration

### Config File

```
# ~/.codex/config.yaml
model: o4-mini
approval_mode: suggest
providers:
  - name: openai
    api_key_env: OPENAI_API_KEY
```

Per-project overrides: .codex/config.yaml in project root

### Project Instructions

```
# AGENTS.md (in project root)
- Run tests with: uv run pytest
- Use ruff for linting
- Never modify migration files
```

### Useful Config Options

**model** Model to use (e.g., `o4-mini`, `o3`)

**approval\_mode** Default approval mode

**providers** API provider config (key, base URL)

**history** Save conversation history: `true` / `false`

## Sandboxing

### How It Works

Codex runs commands in a sandboxed environment to prevent unintended side effects. Network access is disabled by default. File writes are restricted to the project directory.

### Sandbox Options

**macOS** Apple Seatbelt (default, built-in)

**Linux** Docker-based sandbox

**--full-auto** Requires sandbox to be enabled

**--dangerously-auto-approve** Skip sandbox (not recommended)

## Tips

### Effective Prompts

**Be specific** "add retry logic to fetch\_data()" > "improve code"

**Reference files** "fix the bug in src/auth.py" narrows scope

**State constraints** "don't change the public API" sets boundaries

**Iterate** Follow up with refinements in the same session

### Workflow Patterns

```
# Explore → plan → execute
codex "explain the auth module"
codex "plan how to add OAuth support"
codex --approval-mode auto-edit "add OAuth"
```

```
# Review before commit
git add -A
codex review
```

### Common Flags

**--model, -m** Override model for this session

**--approval-mode** Set approval mode

**--quiet, -q** Minimal output

**--no-project-doc** Ignore AGENTS.md