

Referência Rápida do OpenAI Codex CLI

Uso do CLI, modos, configuração, comandos principais

Início Rápido

Instalar

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

Autenticação

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

Defina no perfil do shell ou no arquivo .env

Primeiro Uso

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

Comandos

Interativo (Padrão)

```
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"
```

Não-interativo — executa a tarefa até o fim

codex review

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

Modos

Modos de Aprovação

suggest	Mostrar mudanças propostas, requerer aprovação para cada edição e comando
auto-edit	Aplicar edições automaticamente, requerer aprovação para comandos shell
full-auto	Aplicar edições e executar comandos sem aprovação

Definir o Modo

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

Configuração

Arquivo de Config

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

Overrides por projeto: .codex/config.yaml na raiz do projeto

Instruções do Projeto

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

Opções de Config Úteis

model	Modelo a usar (ex.: o4-mini , o3)
approval_mode	Modo de aprovação padrão
providers	Config do provedor API (chave, URL base)
history	Salvar histórico da conversa: true / false

Sandboxing

Como Funciona

O Codex executa comandos em ambiente sandbox para evitar efeitos colaterais indesejados. Acesso à rede desabilitado por padrão. Escritas de arquivo restritas ao diretório do projeto.

Opções de Sandbox

macOS	Apple Seatbelt (padrão, embutido)
Linux	Sandbox baseado em Docker
--full-auto	Exige sandbox habilitado
--dangerously-auto-approve	Ignorar sandbox (não recomendado)

Dicas

Prompts Eficazes

Seja específico	"add retry logic to fetch_data()" > "improve code"
Referencie arquivos	"fix the bug in src/auth.py" delimita o escopo
Declare restrições	"don't change the public API" define limites
Itere	Continue refinando na mesma sessão

Padrões de Workflow

```
# 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
```

Flags Comuns

--model, -m	Override do modelo para esta sessão
--approval-mode	Definir modo de aprovação
--quiet, -q	Saída mínima
--no-project-doc	Ignorar AGENTS.md