

Riferimento Rapido OpenAI Codex CLI

Utilizzo CLI, modalità, configurazione, comandi principali

Per Iniziare

Installazione

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

Autenticazione

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

Imposta nel profilo shell o nel file .env

Primo Avvio

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

Comandi

Interattivo (Predefinito)

```
codex # avvia sessione interattiva  
codex "fix the login bug" # con prompt iniziale
```

codex exec

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

Non interattivo — esegue il task fino al completamento

codex review

```
codex review # revisiona le modifiche staged  
codex review --diff HEAD~3 # revisiona gli ultimi 3 commit
```

Modalità

Modalità di Approvazione

suggest Mostra le modifiche proposte, richiede approvazione per ogni modifica file e comando

auto-edit Applica le modifiche ai file automaticamente, richiede approvazione per i comandi shell

full-auto Applica modifiche ed esegue comandi senza approvazione

Impostare la Modalità

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

Configurazione

File di Configurazione

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

Override per progetto: .codex/config.yaml nella root del progetto

Istruzioni di Progetto

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

Opzioni di Configurazione Utili

model	Modello da usare (es. o4-mini , o3)
approval_mode	Modalità di approvazione predefinita
providers	Configurazione provider API (chiave, base URL)
history	Salva cronologia conversazione: true / false

Sandboxing

Come Funziona

Codex esegue i comandi in un ambiente sandbox per prevenire effetti collaterali indesiderati. L'accesso alla rete è disabilitato per impostazione predefinita. Le scritture su file sono limitate alla directory del progetto.

Opzioni Sandbox

macOS	Apple Seatbelt (predefinito, integrato)
Linux	Sandbox basata su Docker
--full-auto	Richiede che la sandbox sia abilitata
--dangerously-auto-approve	Salta la sandbox (non consigliato)

Consigli

Prompt Efficaci

Sii specifico	"aggiungi logica di retry a fetch_data()" > "migliora il codice"
Cita i file	"correggi il bug in src/auth.py" restringe lo scope
Specifica i vincoli	"non cambiare l'API pubblica" definisce i limiti
Itera	Fai follow-up con raffinamenti nella stessa sessione

Pattern di Workflow

```
# Esplora -> pianifica -> esegui  
codex "explain the auth module"  
codex "plan how to add OAuth support"  
codex --approval-mode auto-edit "add OAuth"  
  
# Revisiona prima del commit  
git add -A  
codex review
```

Flag Comuni

--model, -m	Override modello per questa sessione
--approval-mode	Imposta la modalità di approvazione
--quiet, -q	Output minimale
--no-project-doc	Ignora AGENTS.md