RÉFÉRENCE RAPIDE OPENAI CODEX CLI

Utilisation CLI, modes, configuration, commandes clés

Démarrage

Installation

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

Authentification

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

Définir dans le profil shell ou le fichier .env

Première utilisation

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

Commandes

Interactif (par défaut)

```
codex                    # démarrer une session interactive
codex "fix the login bug"   # avec un prompt initial
```

codex exec

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

Non-interactif — exécute la tâche jusqu'à la fin

codex review

```
codex review                # revoir les changements indexés
codex review --diff HEAD~3  # revoir les 3 derniers commits
```

Modes

Modes d'approbation

`suggest`	Afficher les changements proposés, demander une approbation pour chaque modification et commande
`auto-edit`	Appliquer les modifications automatiquement, demander approbation pour les commandes shell
`full-auto`	Appliquer les modifications et exécuter les commandes sans approbation

Définir le mode

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

Configuration

Fichier de config

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

Remplacements par projet : .codex/config.yaml à la racine du projet

Instructions de projet

```
# AGENTS.md (à la racine du projet)
- Run tests with: uv run pytest
- Use ruff for linting
- Never modify migration files
```

Options de config utiles

`model`	Modèle à utiliser (ex. `o4-mini`, `o3`)
`approval_mode`	Mode d'approbation par défaut
`providers`	Config du fournisseur API (clé, URL de base)
`history`	Sauvegarder l'historique : `true` / `false`

Bac à sable

Fonctionnement

Codex exécute les commandes dans un environnement isolé pour éviter les effets secondaires non intentionnels. L'accès réseau est désactivé par défaut. Les écritures de fichiers sont limitées au répertoire du projet.

Options du bac à sable

`macOS`	Apple Seatbelt (par défaut, intégré)
`Linux`	Bac à sable basé sur Docker
`--full-auto`	Nécessite que le bac à sable soit activé
`--dangerously-auto-approve`	Ignorer le bac à sable (non recommandé)

Conseils

Prompts efficaces

`Être précis`	"ajouter une logique de réessai à fetch_data()" > "améliorer le code"
`Référencer les fichiers`	"corriger le bug dans src/auth.py" restreint la portée
`Énoncer les contraintes`	"ne pas modifier l'API publique" fixe des limites
`Itérer`	Affiner avec des suivis dans la même session

Patterns de workflow

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

# Revoir avant de commit
git add -A
codex review
```

Options courantes

`--model, -m`	Remplacer le modèle pour cette session
`--approval-mode`	Définir le mode d'approbation
`--quiet, -q`	Sortie minimale
`--no-project-doc`	Ignorer AGENTS.md