

설치 및 CLI

설치 및 인증

- `npm i -g @openai/codex`: 설치 (또는 `brew install --cask codex`)
- `codex login`: 브라우저 로그인 (ChatGPT 플랜)
- `codex login --with-api-key`: stdin으로 API 키 로그인
- `codex doctor` / `codex update`: 진단 / 업데이트

핵심 명령어

- `codex`: 현재 저장소에서 대화형 TUI
- `codex exec "프롬프트"`: 비대화형 실행 (`--json`, `-o out.md`)
- `codex resume --last`: 이전 스레드 재개
- `codex review --uncommitted|--base main|--commit <sha>`: 코드 리뷰
- `codex mcp`: MCP 서버 관리 (`add/list/logout/remove`)
- `codex mcp-server`: Codex를 MCP 서버로 실행
- `codex cloud`: Codex Cloud 작업 탐색/적용
- `codex app`: 데스크톱 앱 실행
- `codex plugin ...`: 플러그인 및 마켓플레이스
- `codex features list`: 기능 플러그 목록
- `codex completion zsh`: 쉘 자동완성 생성

키보드 단축키

에이전트 작업 중

- `Esc`: 중단; 다시 누르면 대화 기록 되돌아가 편집
- `Tab`: 작업 중 후속 프롬프트 대기열 추가
- `Ctrl + C`: 취소 (2회 시 종료)
- `Ctrl + D`: 세션 종료
- `Ctrl + O`: 최근 출력 복사 (= `/copy`)
- `Alt + R`: 깔끔한 복사를 위한 raw 스크롤백

프롬프트 작성

- `@파일`: 파일 퍼지 검색 및 첨부
- `!명령어`: 로컬 셸 명령 인라인 실행
- `Shift+Enter` / `Ctrl+J`: 줄바꿈
- `Up/Down`: 임시 보관 프롬프트 히스토리
- `Esc Esc`: 이전 메시지 편집
- `Ctrl + G`: 외부 편집기에서 프롬프트 열기

슬래시 명령어 (Slash Commands)

세션

- `/new`: 새 대화 시작 (최고의 습관)
- `/clear`: 화면만 지우기
- `/compact`: 요약으로 토큰 확보
- `/resume`: 과거 스레드 다시 열기
- `/fork`: 대화를 새 스레드로 복제
- `/copy`: 마지막 응답 복사
- `/diff`: untracked 포함 git diff 표시
- `/status`: 세션 설정 + 토큰 사용량
- `/exit` (`/quit`): 종료

설정 및 모델

- `/model`: 모델 + 추론 강도 선택
- `/permissions`: 승인 정책 설정
- `/theme`: 구문 강조 테마
- `/vim`: Vim 컴포저 모드
- `/personality`: friendly / pragmatic / none
- `/statusline` / `/title`: 하단 바 / 제목 필드
- `/keymap`: 단축키 재매핑 (`(tui.keymap)`)
- `/mcp`: MCP 서버 패널
- `/hooks`: 라이프사이클 훅 확인
- `/init`: AGENTS.md 생성
- `/apps`: ChatGPT 앱 커넥터 (`$mentions`)
- `/plugins`: 플러그인 브라우저
- `/ps` / `/stop`: 백그라운드 작업 목록/중지

설정 (config.toml)

우선순위

CLI 플러그 → 프로파일 → 프로젝트 `.codex/config.toml` → 사용자 `~/.codex/config.toml` → 시스템 `/etc/codex/config.toml`

- `/debug-config`: 어떤 계층이 적용됐는지 표시

주요 키

```
model = "gpt-5.5"
model_reasoning_effort = "medium" #
minimal..xhigh
approval_policy = "on-request" #
untrusted|on-request|never
sandbox_mode = "workspace-write"
web_search = "cached"
[sandbox_workspace_write]
network_access = false
```

```
[profiles.fast]
model = "gpt-5.4-mini"
• 프로파일 사용: codex -p fast
```

훅 예시

```
[[hooks.PreToolUse]]
matcher = { tool_name = "shell" }
```

파일 구조

- `~/.codex/config.toml`: 전역 설정
- `~/.codex/auth.json`: 자격증명
- `.codex/config.toml`: 프로젝트 설정
- `AGENTS.md`: 프로젝트 지침 (`/init`)
- `.agents/skills/`: 프로젝트 스킬; `~/.codex/skills/`: 개인 스킬

환경 변수

- `OPENAI_API_KEY`: API 키 인증
- `CODEX_HOME`: 설정/세션 위치 변경
- `CODEX_CA_CERTIFICATE`: 기업 프록시용 CA
- `HTTPS_PROXY`: 프록시 지원

워크플로 팁

- 샌드박스: OS 네이티브 (macOS Seatbelt, Linux Bubblewrap)
- 기본 Auto 모드는 묻지 않고 편집; 네트워크는 승인 전까지 차단
- `--search`로 실시간 웹 검색; 캐시 검색은 기본 활성화
- `--oss`로 로컬 모델 실행 (Ollama/LM Studio)
- `-C <dir>` 루트 지정, `--add-dir` 추가 디렉터리 허용