Codex CLI Cheatsheet (v0.130+)

Installation & CLI

Install & Auth

- `npm i -g @openai/codex`: Install (or `brew install --cask codex`)
- `codex login`: Browser login (ChatGPT plan)
- `codex login --with-api-key`: API key via stdin
- `codex doctor` / `codex update`: Diagnose / update

Core Commands

- `codex`: Interactive TUI in current repo
- `codex exec "prompt"`: Non-interactive run (`--json`, `-o out.md`)
- `codex resume --last`: Resume previous thread
- `codex review --uncommitted|--base main|--commit <sha>`: Code review
- `codex mcp`: Manage MCP servers (`add/list/logout/remove`)
- `codex mcp-server`: Run Codex as MCP server
- `codex cloud`: Browse/apply Codex Cloud tasks
- `codex app`: Launch desktop app
- `codex plugin ...`: Plugins & marketplaces
- `codex features list`: Feature flags
- `codex completion zsh`: Shell completions

Keyboard Shortcuts

While Agent Works

- `Esc`: Interrupt; press again to backtrack transcript
- `Tab`: Queue follow-up prompt
- `Ctrl + C`: Cancel (twice to quit)
- `Ctrl + D`: Exit session
- `Ctrl + O`: Copy latest output (= `/copy`)
- `Alt + R`: Raw scrollback for clean copy

Composing

- `@file`: Fuzzy file search & attach
- `!command`: Run local shell command inline
- `Shift+Enter` / `Ctrl+J`: Newline
- `Up/Down`: Draft history
- `Esc Esc`: Edit previous message
- `Ctrl + G`: Open prompt in external editor

Slash Commands

Session

- `/new`: Fresh conversation (best habit)
- `/clear`: Wipe screen only
- `/compact`: Summarize to free tokens
- `/resume`: Reopen past thread
- `/fork`: Clone conversation to new thread
- `/copy`: Copy last response
- `/diff`: Git diff incl. untracked files
- `/status`: Session config + token usage
- `/exit` (`/quit`): Exit

Config & Model

- `/model`: Model + reasoning effort
- `/permissions`: Approval policy
- `/theme`: Highlighting theme
- `/vim`: Vim composer mode
- `/personality`: friendly / pragmatic / none
- `/statusline` / `/title`: Footer / title fields
- `/keymap`: Remap shortcuts ([tui.keymap])
- `/mcp`: MCP servers panel
- `/hooks`: Inspect lifecycle hooks
- `/init`: Generate AGENTS.md
- `/apps`: ChatGPT app connectors (`$mentions`)
- `/plugins`: Plugin browser
- `/ps` / `/stop`: Background tasks list/stop

Configuration (config.toml)

Precedence

CLI flags → profile → project `.codex/config.toml` → user `~/.codex/config.toml` → system `/etc/codex/config.toml`

- `/debug-config`: Show which layer won

Common Keys

```
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"
```

- Profiles: `codex -p fast`

Hooks Example

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

File Structure

- `~/.codex/config.toml`: Global settings
- `~/.codex/auth.json`: Credentials
- `.codex/config.toml`: Project settings
- `AGENTS.md`: Project instructions (`/init`)
- `.agents/skills/`: Project skills; `~/.codex/skills/`: personal

Environment Variables

- `OPENAI_API_KEY`: API key auth
- `CODEX_HOME`: Config/session location override
- `CODEX_CA_CERTIFICATE`: Corporate proxy CA
- `HTTPS_PROXY`: Proxy support

Workflow Tips

- Sandbox: OS-native (Seatbelt/macOS, Bubblewrap/Linux)
- Default Auto mode edits without asking; network blocked until approved
- `--search` enables live web; cached search on by default
- `--oss` runs local models (Ollama/LM Studio)
- `-C <dir>` set root, `--add-dir` grant extra dirs