CODEX SETUP | ONE PAGE | CODEX IN ACADEMIA

Set clear rules for Codex

What Codex may do, what it must ask about, what it must never do, and what else belongs in AGENTS.md. By Usman Khalid, UAE University. Codex facts checked in OpenAI's documentation on 9 November 2026.

1 AGENTS.md states the rules. Settings enforce the ones they can.

AGENTS.md holds your standing rules

Read before every task. It guides Codex but blocks nothing.

`~/.codex/AGENTS.md`	Rules for all projects
`<project>/AGENTS.md`	One project. The closer file wins
`/init`	Writes a first draft to edit

Settings enforce some of the rules

Codex applies them whatever the model decides.

`~/.codex/config.toml`	Folders, network, approvals
`~/.codex/rules/default.rules`	Commands Codex may not run
`~/.codex/hooks.json`	Checks, like a no-delete hook

2 Your rules for what Codex may do, must ask about, and must never do

Allowed, inside the project

- ✓ Read, create and edit project files
- ✓ Run Python, R and LaTeX, and Stata through PyStata
- ✓ Read raw data, but never change it
- ✓ Move retired files to `archive/`
- ✓ Update README, work log and CHANGELOG
- ✓ Check git status and diffs

Ask first

- ? Each git commit, with a one-line summary
- ? Renaming or moving folders
- ? Reading or writing outside the project
- ? Installing packages or going online
- ? Big rewrites, after a dated backup
- ? Changing a rule in AGENTS.md
- ? Paid jobs, such as API batch runs
- ? Sharing a file with an outside service

Never

- × Delete files. Archive them instead
- × `git push`, `reset --hard` or `--no-verify`
- × Open ID, bank, personal or password files
- × Type passwords, 2FA codes or API keys
- × Send email, submit forms, pass CAPTCHAs
- × Read student records or papers you referee
- × Create files named nul, con, aux or prn
- × State a number not read from real output

These are your rules, not Codex guarantees. Section 3 shows the ones that settings can enforce. The rest depend on AGENTS.md and on your own review.

3 What Codex settings can enforce

Stay in the project — A permission profile (beta) that denies reads outside the project, so your ID, bank and personal folders are unreadable. If a tool fails to start, add its folder as `"read"`. Do not also set `sandbox_mode`, or Codex ignores the profile.

Commits and internet — The sandbox keeps `.git` read-only and runs commands offline, so Codex has to ask. Web search is separate and on by default, so the sample turns it off.

Never push or delete — Rules with `decision = "forbidden"` (experimental) stop `git push` and `reset --hard`, which need to leave the sandbox. They may not see deletes inside the project, so add a `PreToolUse` hook, which sees commands and file edits. Keep the project in git, and test in a spare folder.

Keys — `ignore_default_excludes = false` drops variables whose names contain KEY, SECRET or TOKEN from the commands Codex runs. Files are not covered, so the sample also denies `.env` files. Keep keys in a password manager.

Windows — `prefer_mxc = true` picks MXC, the recommended Windows sandbox. The older unelevated sandbox cannot deny reads.

University data — Use an Edu or Business workspace (no training by default). On Plus or Pro, turn training off in Data controls.

config.toml and default.rules, untested sample

```
# ~/.codex/config.toml
approval_policy = "on-request"
default_permissions = "project-only"
web_search = "disabled"
[permissions.project-only]
extends = ":workspace"
[permissions.project-only.filesystem]
":root" = "deny"
":minimal" = "read"
# folders your tools need, for example
'C:\Program Files' = "read"
'C:\texlive' = "read"
[permissions.project-only.filesystem.":workspace_roots"]
"**/*.env" = "deny"
[shell_environment_policy]
ignore_default_excludes = false
[features]
prefer_mxc = true
# ~/.codex/rules/default.rules
prefix_rule(pattern = ["git", "push"],
            decision = "forbidden")
prefix_rule(pattern = ["git", "reset", "--hard"],
            decision = "forbidden")
prefix_rule(pattern = [["rm", "del", "Remove-Item"]],
            decision = "forbidden")
```

4 What else goes in AGENTS.md, and what stays out

Sections worth having in AGENTS.md

1. **Role.** Who Codex works for, and the standard, such as a field-journal referee.
2. **Evidence.** Numbers only from real output, tagged verified or inference.
3. **Design first.** No verdict on an estimate before the specification is run.
4. **Writing.** Your spelling and style rules, and the words to avoid.
5. **Software.** Which programs to use, and how, such as Stata through PyStata.
6. **Records.** Update the README, work log and change log after significant steps.
7. **Literature.** Your own notes first, then your reference manager, then outside search.
8. **Working style.** Plan big jobs, say what you will do, report the outcome first.
9. **Figures.** White background, PNG at 1,200 px for documents, PDF for LaTeX.
10. **House formats.** Your templates for decks, documents and other outputs.

Keep out of AGENTS.md

- × Passwords, API keys and tokens. Codex sends the file to OpenAI's models
- × Student names, IDs or confidential notes
- × Change logs. Keep them in the work log
- × Instructions meant for other AI tools
- × Long procedures. Make them skills in `.agents/skills`

Size. Codex reads at most 32 KiB (`project_doc_max_bytes`), so keep the file short and copy only the rules.

Memories. Off by default. Rules that must always apply go in AGENTS.md.

Shortcut. With `project_doc_fallback_filenames = ["CLAUDE.md"]` in config.toml, Codex reads a project's CLAUDE.md when it has no AGENTS.md. The CLI and IDE extension read config.toml. In the desktop app, Custom (config.toml) mode uses its permissions. Otherwise start in Ask for approval mode.

Untested. The sample has not been run as it stands, and its paths are examples. Try it in a spare folder first. Profiles are beta and rules are experimental.

Sources. OpenAI's Codex documentation at learn.chatgpt.com/docs and Help Center article 11369540, read 9 November 2026. Check them again before you rely on this page.