

Codex as a Disciplined Teammate

Lesson 2 · Context, boundaries, verification

Today's outcome

- Write an executable AI task brief
- Investigate an unfamiliar codebase safely
- Review a proposed diff and its tests
- Preserve useful context as files and playbooks

Treat Codex like a capable new teammate

Give it:

- desired outcome
- relevant files and sources
- scope and safety boundaries
- completion bar

Then review its work like you would any teammate's.

A strong task brief

Find why staff cannot send an inquiry update. Read these paths. Do not change files. Return: request path, cited evidence, likely hypotheses, open questions, and the smallest verification command.

Specific task. Bounded authority. Reviewable result.

Modern Codex workflow

Explore → plan → review → apply → verify

For unfamiliar or consequential work, ask for a plan first. Use repository instructions and current source files as durable context.

Demo

1. Read `AGENTS.md`
2. Trace a feature through UI, server, and test
3. Ask for an investigation plan
4. Verify every cited path
5. Run an approved check

What Codex should say

- “Here is the evidence I found.”
- “This is an inference.”
- “This requirement is ambiguous.”
- “I need approval before changing...”

It should not pretend to know missing business rules.

Lab: Codebase Detective

Mission: investigate, do not rush a fix.

Deliver: task brief, verified repository map, tool result, and open question.

Diff review questions

- Does the change match the acceptance criterion?
- Is its scope small and reversible?
- Would the test fail for the actual regression?
- Did it leak data, broaden permissions, or hide a failure?

Durable context beats chat memory

Keep:

- repository instructions
- test commands
- architectural notes
- approved playbooks
- evaluation cases

What takes an agent further?

Context: current requirements, files and examples.

Tools: actions the client can execute.

Reusable workflow: instructions and scripts for repeated jobs.

Feedback: tests, observed state and corrections.

Tools, MCP, and reusable skills

MCP connects an agent to external tools and context.

A skill packages a repeatable workflow with instructions and optional scripts/resources.

Account access and permissions determine what can run.

Let it execute an authorized task

Ask Codex to inspect, implement, run checks, diagnose failures, and iterate within the learner copy.

Give a clear stopping condition and review the result.

Use extra reasoning or parallel work when it improves measured outcomes.

Exit ticket

Write a task brief asking Codex to **find**, not fix, a suspected regression.