Prompting

Prompts

You interact with Codex by sending prompts (user messages) that describe what you want it to do.

Example prompts:

```
Explain how the transform module works and how other modules use it.
```

```
Add a new command-line option `--json` that outputs JSON.
```

When you submit a prompt, Codex works in a loop: it calls the model and then performs any actions (file reads, file edits, tool calls, and so on) indicated by the model output. This process ends when the task is complete or you cancel it.

As with ChatGPT, Codex is only as effective as the instructions you give it. Here are some tips we find helpful when prompting Codex:

- Codex produces higher-quality outputs when it can verify its work. Include steps to reproduce an issue, validate a feature, and run linting and pre-commit checks.
- Codex handles complex work better when you break it into smaller, focused steps. Smaller tasks are easier for Codex to test and for you to review. If you’re not sure how to split a task up, ask Codex to propose a plan.

For more ideas about prompting Codex, refer to workflows.

Threads

A thread is a single session: your prompt plus the model outputs and tool calls that follow. A thread can include multiple prompts. For example, your first prompt might ask Codex to implement a feature, and a follow-up prompt might ask it to add tests.

A thread is said to be “running” when Codex is actively working on it. You can run multiple threads at once, but avoid having two threads modify the same files. You can also resume a thread later by continuing it with another prompt.

Threads can run either locally or in the cloud:

- **Local threads** run on your machine. Codex can read and edit your files and run commands, so you can see what changes and use your existing tools. To reduce the risk of unwanted changes outside your workspace, local threads run in a sandbox.
- **Cloud threads** run in an isolated environment. Codex clones your repository and checks out the branch it's working on. Cloud threads are useful when you want to run work in parallel or delegate tasks from another device. To use cloud threads with your repo, push your code to GitHub first. You can also delegate tasks from your local machine, which includes your current working state.

Context

When you submit a prompt, include context that Codex can use, such as references to relevant files and images. The Codex IDE extension automatically includes the list of open files and the selected text range as context.

As the agent works, it also gathers context from file contents, tool output, and an ongoing record of what it has done and what it still needs to do.

All information in a thread must fit within the model's **context window**, which varies by model. Codex monitors and reports the remaining space. For longer tasks, Codex may automatically **compact** the context by summarizing relevant information and discarding less relevant details. With repeated compaction, Codex can continue working on complex tasks over many steps.