MSTIMAJ
mstimaj.com

MSTIMAJ LEARNING GUIDE / 01

Skills in Codex

Find the skills you already have, call one by its name, and build your own when the thing you need is not in the list.

THE ONE THING TO REMEMBER

A skill is called by its own name. `$pdf`, `$excel`, `$github`. Only two begin with the word skill, and those two are the ones that build and install skills.

WHAT IS INSIDE

01 Where you are typing

02 The list, and the name

03 Make your own

04 Skills people actually use

05 Sources and further reading

BY THE END YOU CAN

- Open Codex in the folder you actually meant.
- See every skill available to you.
- Call one without looking up its name.
- Build a skill for a job you keep explaining.
- Know where your skills live on your computer.

For Windows, in Command Prompt. Written 10 November 2026 and checked against OpenAI's own documentation the same day. Commands and the skills you see change between versions, so what is on your screen always wins.

01 / WHERE YOU ARE TYPING

Two places, one window

Codex runs inside Command Prompt. You open Command Prompt, walk into the folder you want to work in, and start Codex there. From that moment the window belongs to Codex, and what you type goes to Codex rather than to Windows.

Getting in

1. Open **Command Prompt**.
2. Walk into your folder with `cd`. Most people skip this, and skipping it is why Codex ends up working on the wrong thing.
3. Type `codex`. It opens in that folder and nowhere else.

```
COMMAND PROMPT
C:\Users\You> cd Documents\Education Center
C:\Users\You\Documents\Education Center> codex
```

What you are looking at

Codex takes over that same window. The top line names it and shows the version. The arrow near the bottom is where everything you type goes. The blue line underneath tells you which model you are on and which folder you are in.

```
CODEX, NOW RUNNING IN THAT WINDOW
>_ OpenAI Codex (v0.167.0)
   ~

> Ask Codex to do anything
GPT-6-Astra low Fast · ~
```

THE MIX-UP WORTH AVOIDING

Do not type `/skills` at a plain `C:\>` prompt. Windows does not know what it means. Start Codex first, then type it at the arrow.

01 / WHERE YOU ARE TYPING

So what is a skill?

A folder with your instructions in it. At minimum a file called `SKILL.md` saying what the skill is for and how the work should be done. It can also carry scripts and reference files.

The point of the whole thing is that work you explain once becomes work you never explain again.

```
SKILL.MD
---
name: weekly-update
description: Summarise the weekly project notes I provide.
---

Produce three headings: Completed, Next steps, Blockers.
Use only the facts I give you. Keep it under two hundred words.
Ask for the reporting dates when I have not supplied them.
```

The part between the dashed lines is the label: what it is called and when to use it. Everything under that is the work.

Where skills live

FOLDER	WHO GETS IT
`.agents\skills`	Only this project. Keep it here when the skill is about one job.
`%USERPROFILE%\.agents\skills`	You, everywhere. Keep it here when you want it in every project.

> **IF A SKILL DOES NOT SHOW UP**
>
> Check it is in one of those two folders, and that you started Codex from the right place. A skill saved into one project will not appear in another.

02 / FINDING AND CALLING ONE

The list, and the name

Route one: open the list

Type `/skills` at the arrow. A list comes up from the bottom of the window with every skill available to you, each one with a line saying what it does. Choose one, then describe the job. Use this when you cannot remember the name.

Route two: say the name

If you already know it, skip the list. Type a dollar sign and the name of the skill itself, then your request on the same line.

```
AT THE CODEX ARROW
> $pdf Turn these notes into a two page checklist.
```

READ THIS TWICE

The name is **just the name**. `$pdf`, `$excel`, `$github`, `$gmail`, `$agents`. There is no `$skill-` in front of it. Only two skills begin with that word, and they are on the next page.

THIS WORKS	THIS DOES NOT
`$agents`	`$skill-agents`
`$excel`	`$skill-excel`
`$pdf`	`$skills` on its own

02 / FINDING AND CALLING ONE

Commands worth keeping nearby

The first three are typed at a Windows prompt, before Codex is running. The rest are typed at the Codex arrow, after it is.

WHERE	TYPE THIS	WHAT IT DOES
Command Prompt	`codex`	Start Codex in this folder.
Command Prompt	`codex login`	Sign in.
Command Prompt	`codex resume`	Go back to a previous session.
Inside Codex	`/skills`	Open the list of skills.
Inside Codex	`/model`	Choose the model and how hard it thinks.
Inside Codex	`/new`	Start a fresh conversation.
Inside Codex	`/quit`	Leave Codex and go back to Windows.

Type `/` inside Codex to see every command your version offers. The list changes as Codex is updated.

The same thing, in ChatGPT on the web

The ideas carry over and one character changes. In ChatGPT the dollar sign becomes an `@`, so the creator is `@skill-creator`. Everything else holds: the list is still the list, and the name is still just the name.

> **THE MIX-UP WORTH AVOIDING**
>
> Do not type `/skills` at a plain `C:\>` prompt. Windows does not know what it means. Start Codex first, then type it at the arrow.

03 / THE TWO THAT BUILD

Make your own

You go through the list and the thing you need is not in it. That is what these two are for, and they are the only two whose names begin with the word skill.

TYPE THIS	WHAT YOU ARE ASKING FOR
`$skill-creator`	Build a new skill, or improve one you already have.
`$skill-installer`	Bring in a skill that somebody else already made.

A prompt you can copy

AT THE CODEX ARROW

```
> $skill-creator Create a skill named weekly-update.
  Use it when I ask for a weekly project summary. Read the notes
  I give you and produce three headings: Completed, Next steps,
  Blockers. Keep it under two hundred words. Do not invent
  progress. Ask me for the dates if I have not given them.
```

Skill creator asks what the skill does, when it should kick in, and whether plain instructions are enough or it needs scripts as well. Answer those and it writes the folder for you.

Five minutes, start to finish

1. Open Command Prompt, `cd` into a practice folder, type `codex`.
2. Type `/skills` and read what you have.
3. Call `$skill-creator` with the prompt above.
4. When it is done, type `/skills` again. Yours is in the list.

If the result misses something, tell `$skill-creator` what was wrong and have it revise the skill. You are not fixing one answer; you are fixing the instructions behind every answer it gives from now on.

04 / WHAT YOU MIGHT SEE

Skills people actually use

Examples, not a ranking, and your list will not match this one. Two come with Codex. The rest arrive with whatever packages and tools your setup has. Open `/skills` and read your own.

`$skill-creator` BUILT IN
Builds a new skill, or improves one you have.

`$skill-installer` BUILT IN
Brings in a skill somebody else already made.

`$pdf`
Makes, reads and checks PDF files, page layout included.

`$excel`
Builds and reads spreadsheets, with formulas and charts.

`$design`
Turns notes into a formatted document from a template.

`$agents`
Builds things with the Agents API.

`$github`
Works with your repositories.

`$gmail`
Reads and drafts mail in your account.

ONE MORE TIME

`/skills` opens the list. `$name` calls one. `$skill-creator` makes a new one. In ChatGPT on the web the dollar sign becomes an `@`, and everything else holds.

What to carry forward

The habit worth building is not memorising skill names. It is noticing when you are explaining the same thing for the third time, and turning that explanation into a skill while you still remember what you wanted.

05 / KEEP GOING

Sources and further reading

Everything in this guide was checked against these on 10 November 2026. If something on your screen disagrees with something here, your screen is newer.

Build skills, OpenAI

learn.chatgpt.com/docs/build-skills

The official page. Where Codex looks for skills, how to write SKILL.md, and what the built-in creator and installer do.

Codex skills, OpenAI developers

developers.openai.com/codex/skills

The developer-facing version of the same material.

Codex CLI, OpenAI developers

developers.openai.com/codex/cli

Everything the command line itself can do, beyond skills.

Getting started with the Codex CLI, OpenAI Help

help.openai.com/en/articles/11096431-openai-codex-cli-getting-started

Installing it and signing in, written for people who have not used a terminal before.

openai/skills on GitHub

github.com/openai/skills

Skills you can install with `$skill-installer` instead of writing your own.

Keep learning with me

The Practice Room has the film this guide came from, with the window on screen while it is explained, plus every other lesson. Everything there is free.

THE LESSON FOR THIS GUIDE

mstimaj.com/practice-room/how-to-use-skills-in-codex/