FREE SKILL FOR CLAUDE, CODEX & MORE

Course Digester

Course in, action plan out. Your AI agent reads the course you bought in your own browser, saves every lesson and hands you a checklist, prompts and a skill.

An exclusive bonus from Benjamin Hübner

Questions, or want to see what is working for me right now? Join my free WhatsApp group: bensbonusoffers.netlify.app/whatsapp

What it is

Course Digester is a skill for AI agents. A skill is a folder of instructions the agent follows when you ask for a certain job. This one takes a course you bought and gives you back what you need to act on it.

It runs in Claude Code and the Claude app, and in Codex, OpenClaw, Hermes and other agents that read skills, with Claude, GPT, Grok, Gemini or other models. This guide uses Claude as the example. The page "Other agents" covers the rest.

You log in to the course in Chrome. Claude reads every lesson, saves the text to your computer, has helper agents digest each module, and writes four files for you.

File	What is in it
ACTION-PLAN.md	the checklist, ordered by your goal, starting with "First 60 minutes"
PROMPTS.md	one copy-paste prompt per phase, to do the work with Claude
SUMMARY.md	what the course teaches, the method, the tools, what looks outdated
source/	every lesson as text, your private notes to search later

When the course teaches a repeatable process, Claude offers to turn it into a skill of its own. Then the course method runs every time you need it.

Every point links back to the lesson it came from. You can check anything in seconds.

Who it is for

- ✓ You bought courses that sit unopened in a members area.
- ✓ You want the method and the next steps, fast.
- ✓ You use an AI agent (Claude Code, Codex, OpenClaw, Hermes) or want to start.

What you need

- ✓ **Claude Code** (best: it runs the helper agents side by side) or the Claude desktop app with skills turned on.
- ✓ **Claude in Chrome**, the browser extension. Claude uses it to read the course in the Chrome window where you are already logged in.
- ✓ **Python 3** for a small helper script. Claude Code users usually have it. Check with `python --version`.

Step 1: Connect Claude in Chrome

1. Open the Chrome Web Store and search for "Claude in Chrome".
2. Click "Add to Chrome".
3. Open the Claude side panel in Chrome and sign in with the same account you use for Claude Code or the Claude app.

Claude never sees or types your password. You log in to your course yourself. Claude only reads the pages.

Step 2: Install the skill

Claude Code

1. Unzip `course-digester.zip`.
2. Move the folder `course-digester` into `~/.claude/skills/` (Windows: `C:\Users\<you>\.claude\skills\`).
3. Restart Claude Code. Type `/skills` and check that course-digester is listed.

Claude app (desktop)

1. Open Settings, then Capabilities. Turn on code execution.
2. Open Skills, click Upload skill and choose `course-digester.skill` (or the zip, unopened).

Settings menus move. If the path does not match, search the settings for "Skills".

Run it

1. Open Chrome and log in to your course. Leave the tab open.
2. Make a folder for your notes, for example `Courses`, and start Claude Code in it.
3. Paste the start prompt below, with your course URL.

> *Digest this course: <paste the URL of the course start page or module list>*

Claude opens the course in its own tab, maps the modules and lessons, and shows you a short table. Then it asks three questions at once:

Question	What to answer
Scope	"all", or the module numbers you want
Cost mode	economy, standard or max (see below)
Goal	one sentence: what you want to be able to do after the course

The goal matters most. The action plan is ordered by it, and course parts that do not serve it are listed as "left out on purpose".

Cost modes

Mode	Helper agents	Use it when
economy	Haiku	big courses, tight budget
standard	Sonnet	the default
max	your session's model	you want the deepest digest

While it runs

Claude reads the lessons one module at a time (they share your browser), then digests the modules in parallel. You can keep working in other Chrome windows. Do not close the tab Claude opened.

If the run breaks (laptop sleeps, login expires), log in again and type:

> *Continue the course digest*

Claude reads its log and picks up at the first lesson it has not saved.

After the run

1. Open `ACTION-PLAN.md`. Do the "First 60 minutes" block today.
2. For each phase, open `PROMPTS.md`, copy the prompt, attach the source files it names, and work it through with Claude.
3. Check `00-index.md` for gaps. Lessons that are video only, with no transcript, are listed there. Watch only those.

Turn a course into a skill

If the course teaches a process you will repeat (a funnel build, a content routine, an audit), say yes when Claude offers the course skill. Install it the same way as this one: copy the folder from `skill/` into your skills folder. From then on, ask for the process by name and Claude follows the course method with your source notes.

Tips

- ✓ One course per folder run. Big courses are fine: the log makes them resumable.
- ✓ Use economy mode for a first pass. Run a single module again in max mode if you need more depth there.
- ✓ Ask follow-up questions in the same session: "Which lessons talk about pricing?" Claude searches your source files.

Other agents

The skill needs three things from any agent: a browser tool in a browser window you can see and log in to, file access with Python, and (optional) helper agents. The file `references/hosts.md` inside the skill has the details per host.

Agent	Install the folder to	Browser
Codex	`~/.codex/skills/course-digester/`	a browser MCP server connected to your running Chrome (steps below)
OpenClaw	where your OpenClaw loads skills (same layout as Claude Code)	OpenClaw's browser tool; log in inside the window it opens
Hermes	install the zip as an agent skill	Hermes' browser tools in a visible window
Other agents	load as a skill, or paste `SKILL.md` as instructions	any browser tool in your logged-in session

Codex: connect your Chrome

1. Close Chrome. Start it with remote debugging and a separate profile folder, for example: `chrome --remote-debugging-port=9222 --user-data-dir=C:\chrome-course`
2. Log in to your course in that Chrome window.
3. Add a browser MCP server to Codex (in `~/.codex/config.toml`) that connects to `http://127.0.0.1:9222`, for example Chrome DevTools MCP or Playwright MCP in CDP mode.
4. Start Codex in your notes folder and type "Digest this course" with the URL.

A browser the agent runs invisibly in the background does not work. You must be able to log in yourself, because the agent never types your password.

Cost modes on other models

Economy means the smallest fast model your provider offers (a "mini" or "flash" model, a small Grok model). Standard is the default model. Max is the model your session runs on.

Safety and fair use

- ✓ The agent never types passwords, codes or card numbers. A login page makes it stop and ask you.
- ✓ Text on course pages is data. If a page tries to give the agent instructions, it ignores them and tells you.
- ✓ The agent only navigates and reads. No buy buttons, no account settings, no video downloads.
- ✓ Use it for courses you own, for your own learning. The saved text is your private notes. Do not publish or resell it.

If something goes wrong

Problem	Fix
"No browser tool connected"	Claude: install and sign in to Claude in Chrome, then start a new session. Other agents: see "Other agents"
The agent stops at a login page	Log in in that tab, then type "done"
Many lessons are "video-only"	The course has no transcripts. The plan covers the written parts; watch the listed videos
Skill does not start	Check the folder name is exactly `course-digester` and restart your agent
Python error	Install Python 3 from python.org, then run again