SIDEBAR AI

Project Course Prompts Cheat Sheet

Sidebar

AI · Project Course Prompts Cheat Sheet

How course prompts of the form `topic<course>-<chapter>` behave, from `rules/prompt_rules.md` (agreed 2026-11-03), rules P5, P7, P9 and P11 and `CLAUDE.md`. Applies to ordinary and language courses.

PROMPT FORMS

`art1`	Same prompt as `art1-1`: course name, then course number, then optional `-chapter`.
`react1-3`	Course 1 of `react`, chapter 3.
`no separator`	Never an underscore or space between the name and number.
`german_basic2-5`	Language courses use the same shape: `<language>_basic<course>-<chapter>`.

CHAPTER 1 PROMPTS (ART1 / ART1-1)

`no outline yet`	Writes the outline (`_outline.md`), lists anything unclear, then stops. The streak is unchanged.
`outline exists, no Ch 1`	Says the outline exists, then writes Chapter 1. Streak +1.
`Chapter 1 exists`	Error. Claude says so and the streak resets.

ANY OTHER CHAPTER (ART1-2, ART1-3...)

`outline missing`	Incorrect prompt. Streak resets and Claude asks for `art1` or `art1-1`. No outline is written.
`chapter already written`	Error. Streak resets.
`next expected chapter`	Writes it. Streak +1.
`not the next one`	Claude asks: continue and keep the streak, or reset it. If you say continue, it writes the chapter.

STREAK

`+1`	Whenever a chapter is written.
`unchanged`	Outline-only replies, and clarifying questions and answers.
`reset`	The error cases above.
`where it lives`	A tally in Claude's memory, updated after every course prompt.

FILES AND NAMES

`_outline.md`	The course outline, in the course folder; read only when generating that course's chapters.
`<course_name_snake>_<course#>_<chapter#>.html`	Chapter file, from the full course name, e.g. `frontend_testing_1_1.html`. Courses before 2026-07-08 keep the older `<prefix><course>-<chapter>.html`.
`solutions/<prefix><course>-<chapter>_exercise<N>.txt`	One solution file per coding challenge, in the course's `solutions/` folder.
`<subject>/<course>/`	Course folder under `content/`; follow how sibling courses are nested; ask if no subject fits.

CHAPTER CONTENTS

`banner comment`	First in the file: Course, Chapter, File, Topic, Date Created / Date Updated (rule P1).
`styling`	Two stylesheet links and a `<genre>-lesson <course-slug>` wrapper. No per-chapter `<style>`.
`3 coding challenges`	Each in a `.challenge-block` linking, in a new tab, to its solution `.txt`.
`tables and code`	Comparison tables against known technologies where useful; code blocks per rule P12.

FLESH OUT <TOPIC> (P9)

`how many courses`	1-4, from the topic's real scope, following similar finished courses.
`outline per course`	Full chapter list written to `<course>/_outline.md`.
`tracking rows`	A row in `course_folder_mapping.md` and a line in `course_prompts.md`.
`never`	Generates chapter content. The outline is the deliverable.

WHEN THE LAST CHAPTER LANDS (P4, P7)

`combined PDF`	`py scripts/python/build_course_pdf.py <folder>` writes `<folder>/pdfs/<Title>_Course.pdf`.
`course_prompts.md`	Status becomes `Complete`; the row is trimmed to prompt, title, `Complete` and moved to Completed Courses.
`course_folder_mapping.md`	Row added when the outline is approved and updated when the course completes.
`no PDF built`	Listed in `rules/pending_pdfs.md` rather than skipped silently.
`later chapter prompts`	Any chapter prompt for a finished course is an error.

CROSS-REFERENCES (P11)

`in prose`	Write the full course name plus chapter number, e.g. `SEO Fundamentals 9`, not `seo1-9`, and not in code styling.
`finding the name`	Grep the prefix in `course_prompts.md`, then `course_name_index.md`, then `course_folder_mapping.md`.
`not found`	Ask rather than guess.
`old chapters`	Existing `seo1-9`-style references stay unless a rewrite is requested.

SNAPSHOT

Snapshot of the rule files as at 2026-11-08, summarised by reading them. Nothing was run to test it. If this sheet and a rule file ever disagree, the rule file wins, and the sheet needs regenerating. Rarely used prompt types are not covered.