SIDEBAR AI

Project Prompts Cheat Sheet

Sidebar

AI · Project Prompts Cheat Sheet

Which prompt to type for which job in the learning-blog project, and which rule file Claude reads for it. Summarised from `CLAUDE.md` and the files in `rules/`. `<n>` is a number, `<m>` a chapter number.

COURSES

`art1 / art1-1`	Identical. No outline yet: writes the outline and stops. Outline exists: writes Chapter 1.
`art1-3`	Writes chapter 3, if the outline exists and it is the next chapter due.
`/course-chapter art1-3`	The skill form of the same prompt.
`flesh out <topic>`	Turns a topic idea into 1-4 course outlines. Writes no chapters.
`progress / /progress`	Summary of the session's work, shown in chat and saved to `progress-reports/`.

LANGUAGE LESSONS

`<language>:<topic>`	One topical lesson, e.g. `hungarian:food`. Verb-tense topics get the grammar format.
`/language-lesson <language>:<topic>`	Skill form of the above, with PDF.
`vdd:<language>:<verb>`	Verb deep dive: one verb, fully conjugated, with mini-dialogues.
`/verb-deep-dive <language>:<verb>`	Skill form of `vdd:`.

LANGUAGE COURSES

`<language>_basic<n>`	Outline of Basic Conversation course `n`. `german_basic2-1` is the same prompt.
`<language>_basic<n>-<m>`	Chapter `m` of that course.
`<language>_intermediate<n>`	Same pattern at the Intermediate level, e.g. `german_intermediate1-3`. `_advanced<n>` likewise.
`<language>_grammar<n>`	Grammar track course (outline, then `-<m>` chapters). Reference style, not dialogs.
`<language>_alphabet<n>`	That language's writing system course, e.g. `hungarian_alphabet1`.

SCRIPTS, KANJI, SONGS

`hiragana<n> / katakana<n>`	Script courses, chapters as `hiragana1-4`.
`kanji<n>-<m>`	Kanji course chapter: one kanji per chapter.
`/new-kanji <char> [number]`	Builds one kanji page, tile and World of Kanji badge. Number `0` means no badge.
`song:<title>:<artist>:<language>`	Line-by-line song lesson. Copyright-free lyrics only.
`songvocab:<title>:<artist>:<language>`	Vocabulary list only, no lyrics reproduced.

PROGRAMMING LESSONS

`python: for loops`	`<language>: <topic>`: a standalone lesson plus a companion exercise chapter.
`python: for loops no ex`	Lesson only (`no exercises` means the same).
`python: for loops ex only`	Exercise chapter only (`exercises only` means the same).
`<language>: <topic>`	A programming-language name before the colon means a programming lesson; a natural language means a language lesson.

SIDEBAR PAGES

`sidebar: <topic>`	A standalone deep-dive lesson in the Sidebar section.
`cheat sheet <topic>`	One compact, printable cheat sheet.
`cheat-sheet-<series>_1-4`	A batch from `rules/cheat_sheet_series.md`; `_5` alone builds one sheet.
`/links <topic> [update]`	A curated link collection for a topic, or additions to an existing one.
`/my-tools <name>`	Builds an interactive tool from `my-tools/<name>.yaml`.

NOTES, LOGS AND HOUSEKEEPING

`add this to q&a`	Copies the latest question and answer into `faqs/qa_with_claude.md`.
`korean history question <topic>`	A sourced note on how accurate a drama is; appended to `faqs/korean_history_notes.md` automatically.
`(any agent question)`	A question about Claude Code agents or subagents is appended to `faqs/agent_questions.md` automatically.
`conversation`	Saves the dialogue since the last time to a new file in `conversations/`.
`/restart`	Wrap-up: saves the progress report and the transcript before closing.
`/rewrite / /fix-names`	Work on files in the `html-rewrite` staging folder: bring them up to current rules, or rename chapter files.

READING THE PROMPT

`no space or underscore before the number`	`korean_basic3`, `react1-3`. Forms like `korean_basic_3` are retired and get a question back.
`colon: lesson`	Text with a colon is a lesson; text without one is a course prompt.
`which rules apply`	Course prompts: `prompt_rules.md`. Language: `language_*.md`. Python and other languages: `programming_lesson_rules.md`. Sidebar: `sidebar_rules.md`.
`unsure`	Claude reads the likely rule file, or asks, rather than guessing.

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.