

The Claude Fable 5.1 System Prompt Study Guide

Clone the leaked 8,000-line Anthropic prompt, read it in Claude Code without blowing your context, and pull out the tool rules for your own skills.

Why I made this guide

On September 1, 2026 the full system prompt for Claude Fable 5.1 landed in a public GitHub repo. It is the complete instruction set Anthropic loads before your first message on claude.ai: the behaviour rules, the memory rules, the search rules, and every tool definition. The video quotes it at over 275,000 characters and around 8,000 lines. When I pulled the raw file on September 12, 2026 it measured about 413,000 characters across 7,810 lines, so the number moves with each capture.

The basic way to read it fails. Most people open the file on GitHub, scroll for two minutes, get lost somewhere in the Gmail tool definitions, and close the tab. The second most common move is pasting the whole thing into a chat and asking for a summary, which either gets rejected for size or comes back as a vague paragraph that tells you nothing you could reuse.

This guide gives you a way to read it that produces something you can keep: clone the repo, let Claude Code read it in slices, and extract the patterns (the tool-definition format, the ordered decision checklists, the search rules) into files you control. One evening, one folder, and you come out with a reusable template for how a frontier lab writes instructions.

01

How it actually works

1. You clone the repo so the file is on your disk. Claude Code can then read it with line offsets and grep instead of trying to swallow 413,000 characters in one go, which no context window enjoys.
 2. You ask Claude Code targeted questions per section. The file has named top-level headings (claude_behavior, memory_filesystem, search_instructions, Tools and so on), so every question can point at one heading and stay small.
 3. You have Claude write what it finds into your own files: a section map, a tool-definition template, and a checklist of rules. Those files go into a skill folder, and that is what you reuse. The leaked prompt stays where it is.
-

The one thing almost everyone gets wrong

People treat the file as something to read top to bottom, or paste into a chat, when it is really a reference to search. Roughly two thirds of it (lines 2,067 to 7,421 in the September 12 copy) is the Tools section, and most of that is JSON schemas for Gmail, Google Calendar and Google Drive connectors. The parts that teach you something about prompt writing are the first 2,000 lines and the shape of any three or four tool definitions. Read those on purpose; skim the rest with grep.

WATCH OUT

Do not paste the whole file into a Claude Code session or a chat. At around 413,000 characters it eats most of a context window before you have asked a single question, and Claude Code's file reader only returns about 2,000 lines per call anyway. Work from the cloned file with line ranges and grep, and keep each question pointed at one heading.

The file

There is no code to install in this one. The thing you are studying is a markdown file in the repo below, and the two blocks after it are lifted from that file so you can see the shape before you open it.

SOURCE CODE

The leaked Claude Fable 5.1 system prompt, in the asgeirtj/system_prompts_leaks repo:

https://github.com/asgeirtj/system_prompts_leaks/blob/main/Anthropic/claude-fable-5.1.md

Anthropic/claude-fable-5.1.md (as pulled September 12, 2026)

7,810 lines, about 413,000 characters, 97 tool definitions

```
# claude_behavior
  ## product_information
  ## refusal_handling
  ## legal_and_financial_advice
  ## tone_and_formatting
  ## reply_after_tool_calls
  ## user_wellbeing
  ## anthropic_reminders
  ## evenhandedness
  ## responding_to_mistakes_and_criticism
  ## knowledge_cutoff
# memory_filesystem
  ## What's already filed
  ## File format
  ## Where it goes
  ## When to write
  ## Calibration
  ## Read before writing
# end_conversation_tool_info
# persistent_storage_for_artifacts
# mcp_app_suggestions
# suggest_catalog_plugins_and_skills
# past_chats_tools
# preferences_info
# computer_use
  ## skills
  ## file_creation_advice
  ## file_handling_rules
  ## producing_outputs
  ## sharing_files
  ## artifact_usage_criteria
  ## package_management
# request_evaluation_checklist
# when_to_use_visualizer_for_inline_visuals
# search_instructions
  ## core_search_behaviors
  ## search_usage_guidelines
  ## CRITICAL_COPYRIGHT_COMPLIANCE
  ## search_examples
  ## harmful_content_safety
  ## critical_reminders
# Tools (97 definitions: bash_tool, create_file, str_replace, view,
      web_fetch, web_search, memory_*, present_files, chart_display_v0,
      Gmail:*, Google Calendar:*, Google Drive:*, visualize ...)
# anthropic_api_in_artifacts
# available_skills
# network_configuration
```

Every one of the 97 tools uses the same three-part pattern: a name, a plain sentence or two on when to use it (and often when not to), then a JSON schema with typed parameters and a required list. Here is the first one in the file, verbatim. Notice that the model is required to say why it is running the command as part of the call.

BASH_TOOL (VERBATIM FROM THE LEAKED FILE)

```
## bash_tool

Run a bash command in the container

```json
{
 "name": "bash_tool",
 "parameters": {
 "properties": {
 "command": {
 "description": "Bash command to run in container",
 "type": "string"
 },
 "description": {
 "description": "Why I'm running this command",
 "type": "string"
 }
 },
 "required": [
 "command",
 "description"
],
 "title": "BashInput",
 "type": "object"
 }
}
```

#### SITES SHOWN IN THE VIDEO

- The repo root, with the table of every captured prompt  
[https://github.com/asgeirtj/system\\_prompts\\_leaks](https://github.com/asgeirtj/system_prompts_leaks)
- Claude Code (Fable 5.1) prompt, added the same day  
[https://github.com/asgeirtj/system\\_prompts\\_leaks/blob/main/Anthropic/claude-code/claude-code-fable-5.1.md](https://github.com/asgeirtj/system_prompts_leaks/blob/main/Anthropic/claude-code/claude-code-fable-5.1.md)
- Claude Code headless (Fable 5.1) prompt, added September 5, 2026  
[https://github.com/asgeirtj/system\\_prompts\\_leaks/blob/main/Anthropic/claude-code/claude-code-headless-fable-5.1.md](https://github.com/asgeirtj/system_prompts_leaks/blob/main/Anthropic/claude-code/claude-code-headless-fable-5.1.md)

## Set it up in Claude Code or Codex

Nothing here needs an API key. You need git, a terminal, and either Claude Code or Codex installed. The whole setup is under five minutes; the reading is the part that takes an evening.

1. Open a terminal and check git is installed with `git --version`. On a Mac, if it is missing, the command itself offers to install the Xcode command line tools. Accept and wait.
2. Clone the repo somewhere you keep projects: `git clone https://github.com/asgeirtj/system_prompts_leaks.git` then `cd system_prompts_leaks`. The clone pulls every captured prompt in the repo (ChatGPT, Gemini, Grok, Perplexity, Codex and the Claude family), so expect a few hundred files.
3. Run the first test before opening any AI tool: `wc -l -c Anthropic/claude-fable-5.1.md`. A good result on the September 12, 2026 copy is 7810 lines and 414545 bytes. If the file is missing, run `git pull`; if it has moved, check the table in README.md, which lists the current path for every prompt.
4. Start your agent inside that folder. For Claude Code type `claude`; for Codex type `codex`. Starting inside the repo matters because both tools scope file reads to the folder you launched from.
5. Send the first prompt from the next section (the section map). A successful first run comes back with the list of top-level headings and their line numbers, and Claude should have used `grep` or a line-range read rather than reading the whole file. If it tries to read all 7,810 lines, stop it and tell it to `grep` for lines starting with a hash instead.
6. When you want to keep what you extract, make a skill folder. For Claude Code that is `~/.claude/skills/prompt-patterns/` with a `SKILL.md` inside, and the `name:` line in that file must match the folder name exactly (prompt-patterns). For Codex, put the same folder at `.codex/skills/prompt-patterns/` in your project and add a one-line pointer to it in `AGENTS.md`. Ask Claude to write the file; you only decide what goes in it.

#### QUICK TIPS

- Run `git pull` inside the folder every week or two. The repo is updated within days of a new model shipping, and the README table at the top shows what changed and when.
- Keep the leaked file out of your skill folder. Reference it by path or copy small excerpts. A 413,000-character file inside `~/.claude/skills/` gets read at the worst possible moments.

## 04

### The commands I actually run

These are the prompts I send to Claude Code once I am inside the cloned repo. Each one points at a heading or a line range so the read stays small. Replace anything in square brackets.

#### MAP THE FILE BEFORE READING IT

Do not read Anthropic/claude-fable-5.1.md in full. Run `grep` for lines starting with '#' or '##' with line numbers, then give me a table of every top-level section, its starting line, and its length in lines. Flag which sections are behaviour rules and which are tool definitions.

### EXTRACT THE TOOL-DEFINITION TEMPLATE

Read the Tools section of Anthropic/claude-fable-5.1.md starting at the '# Tools' heading, but only the first [4] tool definitions. Describe the exact pattern each one follows (description text, JSON schema, required fields, any ordering instructions in the parameter titles). Then write a blank template in the same format to tools-template.md that I can fill in for my own [SKILL OR AGENT NAME].

### PULL ONE BEHAVIOUR SECTION INTO A CHECKLIST

Read only the '## [tone\_and\_formatting]' section of Anthropic/claude-fable-5.1.md (use grep to find its line range first). Turn every rule in it into a one-line checklist item, quoting the original sentence under each item. Save it to checklists/[tone\_and\_formatting].md. Do not paraphrase away any numbers or named words.

### COMPARE TWO VERSIONS OF THE PROMPT

Compare Anthropic/claude-fable-5.md with Anthropic/claude-fable-5.1.md using diff on the section headings first, then on the '[search\_instructions]' section only. List what was added, removed, or reworded, with line numbers from each file. Keep it to the differences; I do not need a summary of the shared parts.

### TURN THE FINDINGS INTO A SKILL

Using tools-template.md and the files in checklists/, write ~/.claude/skills/[prompt-patterns]/SKILL.md. The name field must be '[prompt-patterns]' to match the folder. Include: the three-part tool definition pattern, the ordered 'stop at first match' checklist style from request\_evaluation\_checklist, and the search query rules (1 to 6 words, one search for simple facts). Cite the source file path and the date [September 12, 2026] at the top.

---

## When it breaks

Every one of these has happened to me or to someone I walked through it. The fixes are short.

ERROR	FIX
<b>git: command not found</b>	Install git first. On a Mac run <code>xcode-select --install</code> and accept the prompt; on Windows install Git for Windows. Then open a new terminal and retry the clone.
<b>GitHub says the file is too large to display, or the page hangs while rendering</b>	Use the Raw button, or skip the browser and read from your clone. The file is over 400,000 characters and the web view struggles with it.
<b>Claude Code tries to read all 7,810 lines and stalls or truncates</b>	Its file reader returns about 2,000 lines per call. Tell it to grep for headings first, then read a line range (for example lines 96 to 136 for <code>tone_and_formatting</code> ).
<b>Claude summarises instead of quoting the exact rule</b>	Ask for the original sentence with its line number under each checklist item, and say 'do not paraphrase'. Quoting from a local file is fine; it is the vague summary that wastes the read.
<b>wc shows a different line or byte count than this guide</b>	The repo updates the capture. Run <code>git log -1 -- Anthropic/claude-fable-5.1.md</code> to see when it last changed, and re-run the section map prompt so your line numbers match your copy.
<b>Path not found: Anthropic/claude-fable-5.1.md</b>	Files get renamed when a new version lands. Open README.md in the repo and use the path from the Claude.ai system prompts table.
<b>The skill does not load in Claude Code</b>	The folder name and the name: field in SKILL.md must match exactly, and the folder must sit directly under <code>~/.claude/skills/</code> . Restart Claude Code after creating it.
<b>Codex ignores the skill folder</b>	Codex looks for <code>.codex/skills/</code> inside the project and reads AGENTS.md. Add a line in AGENTS.md pointing at the skill and start Codex from the project root.

## Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

**Join the Build →**

<https://hub.digicuratoragency.com/join>