



Claude Rules Workflow

A One-Chapter Mini-Course

Covers:

Adding rules · Recovering rules after memory loss
Rule numbering (P1, R1, C1...) · Rule categories · Future improvements

Format: A4 · Single chapter, reference document

CHAPTER 1 OF 1

How the Permanent Rules System Works

RULES WORKFLOW · CH 1

How the Permanent Rules System Works

Writing rules, recovering them after memory loss, numbering, categories, and where this could go next

Claude's built-in memory is convenient but not guaranteed to survive forever — it can be cleared, lost between environments, or simply not exist yet in a brand-new session. This mini-course documents a parallel system: plain text/markdown rule files stored on disk, in a folder Claude can be pointed back to at any time. The rules themselves are just instructions written in plain English — there's nothing magic about the format, only the discipline of keeping them in one place and numbering them for easy reference.

1. Adding a New Rule

To add a rule, simply ask Claude directly, dictating the rule in plain English. Claude formats it consistently and assigns it the next number in sequence.

```
Please add a rule to the permanent_rules.md file as follows:
```

```
All files you generate should be stored in a subdirectory of the
folder C:\Users\emuba\OneDrive\My Learning\WebSites\debserver\claude-storage
(which I will call the claude root directory in prompts). The
subdirectory should be selected based on the following rules:
```

```
Files with extension .html should be stored in the html subdirectory.
Files with extension .txt should be stored in the 'text files' subdirectory.
Files with extension .sql should be stored in the sql directory.
Everything else should be stored in the claude root directory.
```

Claude rewrites this for clarity if needed, assigns it an ID (e.g. `P4`), and renumbers the file if anything was previously out of order — this became Permanent Rule P4 in practice.

2. Recovering Rules After Memory Loss

If a new session starts with no memory of any rules at all, the minimum viable recovery prompt is simply asking Claude to read the rules file and commit it to memory for the session:

Please read the following rules file and commit the rules to memory:

C:\Users\emuba\OneDrive\My Learning\WebSites\debserver\claude-generated-code\rules\permanent_rules.md

To make this even more reliable, a dedicated **recovery rule** exists specifically for this purpose — **R1**, stored in its own file, **recovery_rules.md**, separate from the permanent rules themselves:

****R1 – Session Rule Recovery****

At the start of every session, please read the permanent_rules.md file stored in the folder

C:\Users\emuba\OneDrive\My Learning\WebSites\debserver\claude-generated-code\rules and treat every rule inside it (P1, P2, P3, P4, ...) as active for the remainder of the session.

WHY A SEPARATE RECOVERY_RULES.MD FILE?

Keeping R1 in its own file means it can be pasted in on its own, even if the user can't remember exactly where permanent_rules.md lives or what's in it — R1 itself contains the full path, so it's self-sufficient as a recovery prompt.

3. The Rule Numbering System

Each rules file gets its own letter prefix, and rules inside it are numbered sequentially: **P1**, **P2**, **P3** ... for permanent rules, **R1** for recovery rules. This lets a rule be referenced directly in a prompt instead of re-explaining it every time.

Please generate a Go course using rule P1.

Claude reads **permanent_rules.md**, finds the rule numbered P1 (explanation style), and applies it without needing the full rule text repeated in the prompt.

4. Rule Categories

Nothing stops a single **permanent_rules.md** file from holding every rule, but separate prefixes per category keep numbering meaningful as the rule set grows — particularly useful for rules that need an example to follow, like styling consistency across a group of pages.

PREFIX	CATEGORY	EXAMPLE
P	Permanent / always-active rules	P1 – P4 in permanent_rules.md
R	Recovery instructions	R1 in recovery_rules.md

PREFIX	CATEGORY	EXAMPLE
C	CSS / styling consistency rules	C1 — match an example file's styling

A CSS consistency rule might read like this, referencing a saved example file as the visual reference:

****C1 — Pi Lesson Styling****

Use the same styling as the example file:

C:\Users\emuba\OneDrive\My Learning\WebSites\debserver\claude-generated-code\rules\examples\css\pi_lesson_01.html

Purpose: keeps a group of related lesson pages visually consistent without re-describing the CSS in every prompt.

Used in a prompt: `Please generate the next Raspberry Pi lesson using rule C1.` Claude opens the referenced example file, matches its styling, and applies it to the new page.

5. Could This Be Done Better?

This file-based system works well precisely because it doesn't depend on memory persisting at all — the files on disk are the source of truth, and memory is just a convenience layer on top. A few possible refinements worth considering as this evolves:

- **One combined index file** listing every rule across every category (P, R, C, ...) with a one-line summary, so it's quicker to scan what exists without opening each file.
- **A "search rules" habit** — explicitly asking Claude to check the rules folder before starting unfamiliar work, rather than relying on it to remember unprompted.
- **Periodic consolidation** — occasionally reviewing the rules files for duplicates or rules that have become outdated, the same way Claude's own memory system gets reviewed.

THIS WORKFLOW IS GENUINELY A FIRST ATTEMPT

As experience with Claude grows, a cleaner or more automated approach may turn out to work better — this mini-course documents what works *now*, not a final answer. Revisiting it periodically is expected, not a sign anything here was wrong.

CHAPTER 1 QUICK REFERENCE

- **Add a rule:** dictate it in plain English, Claude formats and numbers it
- **Recover rules:** ask Claude to read permanent_rules.md, or use R1 directly

- **Reference a rule:** "...using rule P1" / "...using rule C1"
- **Categories so far:** P (permanent), R (recovery), C (CSS/styling)
- **Source of truth is the files on disk**, not Claude's memory — memory is just a convenience layer
- **This system is a first pass** — expect it to be refined over time