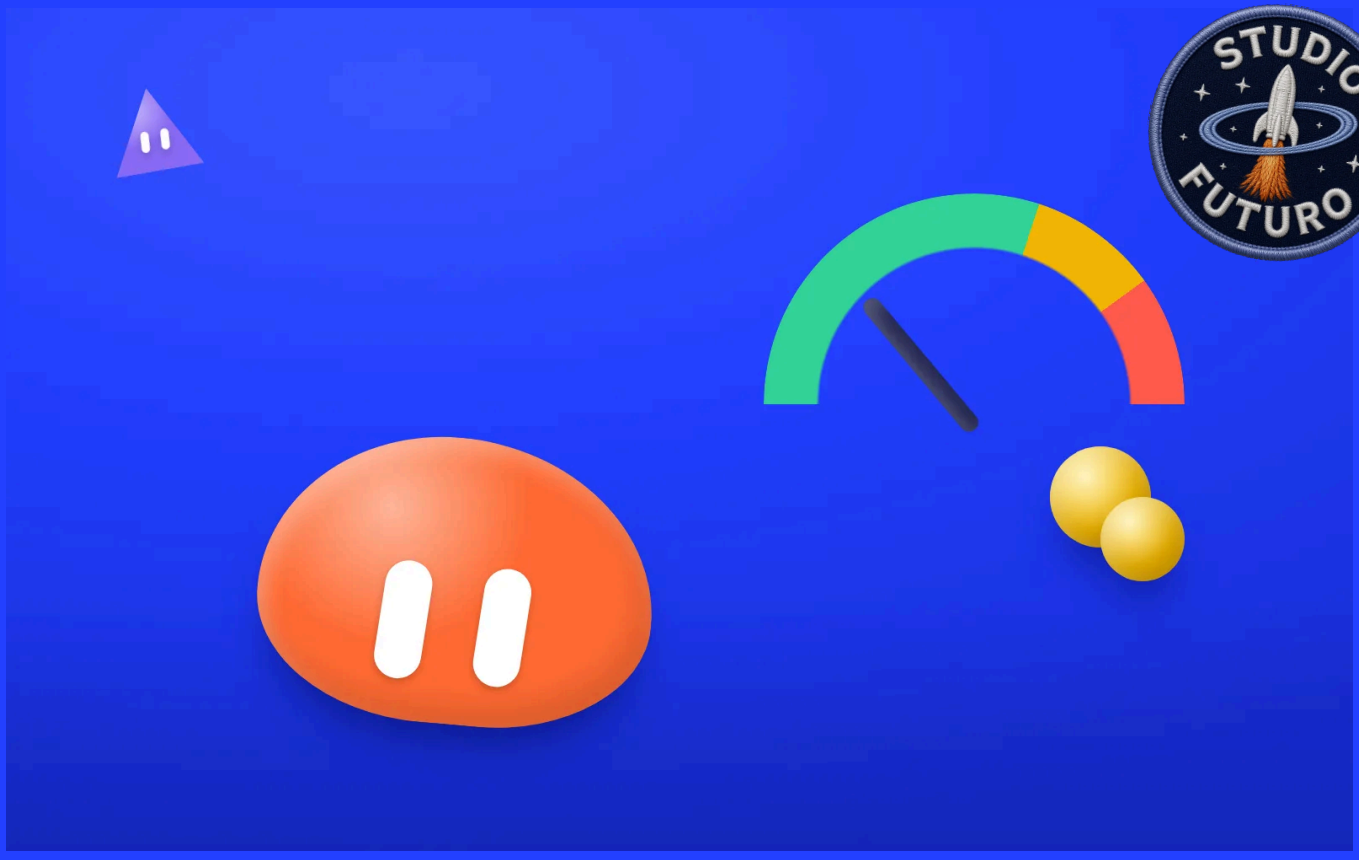


A PILL FOR CLAUDE CODE USERS, EVEN NEW ONES

Make Claude Code use less.

Seven habits to stop burning millions of tokens in a single chat, and Quack, the free plugin that reminds you of them on its own.

7 habits

To use today, by hand.

1 free plugin

Quack makes them automatic.

2 commands

To install it, on page 6.

BEFORE YOU START

5 words to know.



That's all you need to follow this guide. If you know them already, skip to the next page.

Token	Claude's "fuel". Every word it reads or writes uses a few tokens. A 500-line code file can be worth several thousand.
Context	Everything Claude has in front of it while it answers: the conversation, the files it has read, the project instructions. It re-reads all of it on every message.
Cache	A short-term memory that makes that re-reading cheaper. It expires after a pause: usually 5 minutes if you pay per use, one hour on a subscription.
/compact and /clear	Two commands you type in the chat. /compact summarizes the conversation and carries on from the summary. /clear empties it and starts fresh, while the project files stay.
Subagent and effort	A subagent is a helper that works in its own space, often on a cheaper model, and brings back only a summary. Effort is how much Claude thinks before answering: low, medium or high.

WHAT IT IS

Claude doesn’t work too much. It re-reads too much.

Claude doesn't remember a chat the way a person does. On every message it re-reads the whole context from the start. In a long chat, even "ok, thanks" costs as much as re-reading a book.

Picture a colleague who, before answering you, reads out loud every email of the week.

AN EXAMPLE WITH ROUND NUMBERS: 40 MESSAGES IN AN AFTERNOON

6 M

tokens read with one 150,000-token chat used for days

800,000

tokens read with one chat per task, around 20,000 tokens each

7x

less reading for the same work

READ THIS CAREFULLY

It's an example to show the scale, not a measurement. The cache lowers the price of re-reading but doesn't remove it, and after a pause it expires: the first message when you come back re-reads everything at full price.

It applies to everyone: if you pay per use the bill grows, on a subscription the limit runs out sooner.

THE HABITS, BY HAND

Light chats, the right models.

1 One chat per task

When you move to a different job, empty the chat. The new one starts light. Claude finds the project rules and files again on its own.

TYPE

`/clear`

2 Summarize at a pause

If a task is long, have it summarized halfway. Claude carries on from the summary. Do it at a resting point, not in the middle of a delicate fix.

TYPE

`/compact`

3 The right model and effort

For everyday work a mid-size model is enough. With opusplan the strongest model writes the plan and the mid-size one carries it out. For small edits, low effort. Don't switch model mid-task: the cache starts over.

TYPE

`/model sonnet`
`/model opusplan`
`/effort low`

4 Helpers for long searches

Finding a function, reading docs, searching the web fills the chat with text you won't need again. Ask Claude to hand it to a subagent: it works in its own space and brings back the summary.

ASK

`"Hand the search to a subagent"`

A TYPICAL DAY

NEW TASK	HALFWAY	LONG SEARCH	END OF DAY
`/clear`	`/compact`	`subagent`	`note where you are`

THE LAST THREE

Read little, watch the numbers.

5 A short CLAUDE.md

It's the project's instruction file, and Claude reads it on every message. The Claude Code docs suggest keeping it under 200 lines. The rest goes into documents Claude opens only when needed.

GOAL

`under 200 lines`

6 Files read in parts

To fix one function you don't need to read 2,000 lines. Ask Claude to find the function and read only those lines.

ASK

`"Find the function, read only those lines"`

7 Watch the numbers

/context shows what fills the chat now, /usage how much you've used. Check them once a day for a week: you learn fast where the tokens go.

TYPE

`/context`
`/usage`

SHORT ON TIME? START WITH TWO

Almost all the waste comes from the single chat. Start with **/clear between tasks** and **/model sonnet for routine work.** Add the other habits, or the plugin, later.



THE FREE PLUGIN

Quack does it automatically.

A Claude Code plugin that turns the habits into rules and reminders.
Free and open source, by Alek Dobrohotov and Studio Futuro.

HOW TO INSTALL: IN CLAUDE CODE, ONE AT A TIME, THEN RESTART

```
/plugin marketplace add AlekDob/quack-marketplace
/plugin install quack@quack-marketplace
```

Rules at start

Short answers, files read in parts, searches handed to helpers, one chat per task.

Reminders at the right time

Every 100,000 tokens: "New task? /clear. Same task? /compact." And when a pause expired the cache.

Big files

Before Claude reads a file of more than 400 lines in full, it reminds Claude to look for just the right part.

A code map

/quack:index builds an index of folders, files and functions with exact line numbers. It refreshes itself.

The duck team

Five helpers, each on the right model: they plan, code, search, research and take notes. Page 7.

Memory between chats

The project diary tells a new chat where you got to: /clear doesn't mean starting over.

Setup check

/quack:checkup measures what weighs on every message and suggests fixes. It applies only the ones you approve.

What it can’t do

Type /clear or /compact for you, or change the main chat's model. It warns you, and you type the command.

It adds about 750 tokens per session, less than a single whole-file read it saves you. It runs on your computer. It sends one anonymous ping when you install or update it, and you can turn it off. Everything at **quack.build**.

THE DUCK TEAM

Quack is the duck. It brings a team.

Programmers call it "rubber duck debugging": explain the problem to a rubber duck and the answer comes. Quack is that duck, and it brings five helpers. Each works on its own model and brings back only a summary.



Pia

PLANNER · OPUS · HIGH

Plans the big or unclear tasks. Never touches the code.



Kit

CODER · OPUS · MEDIUM

Builds Pia's plan or a long job in his own space. Reports only what changed.



Marco

SCOUT · HAIKU · LOW

Finds where things are. Paths and line ranges, never whole files.



Iris

RESEARCHER · SONNET · MEDIUM

Looks things up on the web and answers with sources.



Dante

SCRIBE · HAIKU · LOW

Diary, gotchas, decisions: the memory for the next chat.

WHO GETS THE WORK

A small or short task	**Your chat**
Chat on Sonnet, big or unclear task	**Pia plans, Kit builds**
Chat on Opus, long job	**Kit, the chat stays small**
Searching, researching, taking notes	**Marco, Iris, Dante**

WHAT WE MEASURED: SAME PROJECT, SAME TASKS, TWO RUNS EACH

TASK	WITHOUT QUACK	WITH QUACK
Explain a code flow	$0.40	$0.41
Rate limit on 3 files and a test	$0.56	$0.54
Same task, chat on Sonnet	$0.25	$0.25
Same task, Pia and Kit (Opus), chat on Opus	–	$0.92
Same task, Pia and Kit (Opus), chat on Sonnet	–	$0.63

On single tasks Quack moves the bill by a few points. The biggest lever is the model: on Sonnet the same job costs less than half. Pia and Kit buy quality on hard tasks, not discounts. Quack's savings are in long chats: check /usage after a week. Every run at github.com/AlekDob/quack-marketplace.

AFTER INSTALLING

One prompt and you’re set.



READY TO COPY, INSIDE YOUR PROJECT

```
Use the quack:checkup skill on this project.
Show me what weighs on every message in a simple table: item, how much it weighs, what to do about it.
Suggest the fixes one at a time and apply only the ones I approve.
Then use the quack:index skill to build the code map.
At the end, explain in plain words what changed and how best to work from tomorrow.
```

SMALL TEST, ONE WEEK

Before installing, open the chat you've used for days and type /context: see how many tokens it carries. Then install Quack, type /clear and take on your next task in a new chat. After a week, compare /usage with the week before.

FREQUENTLY ASKED QUESTIONS

Do I need to code?

You need to know how to use Claude Code. The rest is two commands to copy.

Subscription or API?

Both. On a subscription your limit lasts longer; on the API you spend less.

Do my files go anywhere?

No. Quack runs on your computer. When you install or update it, it sends one anonymous ping: the version and a random id. QUACK_NO_PING=1 turns it off.

Does it work with Cursor or Codex?

No, it's a Claude Code plugin. The seven habits work with any assistant.

WATCH OUT FOR

The plugin warns. You decide.

- **The 6 million is an example.** How much you save depends on how you work today.
- **/clear and /compact stay with you.** You know when a task is done.
- **A summary can drop a detail.** Before /compact, have Claude note where you got to.
- **Limits and prices change.** Check /usage and the Claude Code docs.

SOURCES

Claude Code cost guide (code.claude.com/docs/en/costs). PageIndex by VectifyAI for the tree-map idea. caveman for short answers. RTK for shorter command output.



Quack is free.

Open source, by Alek Dobrohotov and Studio Futuro. Want us to set it up for your team? We measure usage before and after, on your real projects.

quack.build
hello@studiofuturo.ai