

FREE RESOURCE

The AI Token-Saving Cheat Sheet

27 practical habits that stretch any ChatGPT, Claude, or Gemini plan further — no upgrade required.

futurefactors.ai · From the AI Tokenomics guide by Sana Mian

How to use this.

Twenty-seven habits is a lot. Don't try all of them at once. Skim the seven groups below, pick three that match how you actually use AI day to day, and run them for a week. Everything here applies to a flat-fee ChatGPT, Claude, or Gemini subscription — you don't need to touch a developer API to use any of it.

01 START HERE — THE FOUR FUNDAMENTALS

Start fresh, not forever

Continue a thread and every reply re-reads the whole conversation above it. Begin a new chat once a topic wraps up, instead of dragging it along for the rest of the day.

Reuse instead of re-uploading

Put a file, style guide, or reference doc you'll use more than once into a persistent space (Claude Projects, a custom GPT, Gemini Gems). Content stored there is available across chats without being re-read from scratch each time.

Match effort to the task

Extended or deep thinking modes are built for hard, multi-step problems. Leave them off for routine questions — a typo fix or a quick reformat doesn't need a model reasoning step by step first.

Batch related questions

Three small messages cost three full re-reads of your conversation. One message with three requests costs one.

Instead of three messages, try: "Three things — summarize this article, list the main points, and suggest a headline."

02 PROMPT AND CONTEXT HABITS

Write the instruction, not the essay

A long, meandering prompt feels thorough, but every extra sentence is extra tokens on every message in that thread, not just the first one. Specific beats rambling, on cost and on quality.

Put the stuff that never changes first

In a longer, reusable prompt (a template, a style guide, a set of rules), put the fixed part at the top and the part that

changes at the bottom. Several providers process a repeated, unchanged opening more cheaply than fresh text.

Ask for the length you actually want

Output tends to cost more per token than input. If you need one sentence, ask for one sentence — don't let a model generate three paragraphs you'll only skim.

Try adding: "Answer in one paragraph, no more than 80 words."

Summarize before you paste, not after

Pasting a 40-page document to ask about one paragraph sends the whole document through, every single follow-up. Paste just the section you need, or summarize first and work from the summary.

Reach for the lightest tool first

Before opening the heaviest, most capable model for a routine task, ask whether a lighter model or a quick template would do. Escalate only when the task actually needs the extra reasoning.

03 FIX IT, DON'T RESTACK IT

Edit your last message instead of replying

When an answer misses, the instinct is to type a correction underneath. That's expensive — the model now re-reads the bad answer AND your correction AND everything else, forever. Most chat tools let you edit and regenerate your original message instead, which replaces the exchange rather than stacking on top of it.

Ask for a fix to just the part that's wrong

When a long response is mostly right, don't ask for a full redo.

Try: "Keep everything else, just rewrite the second paragraph to be shorter."

Tell it to skip the preamble

By default, most AI chat tools pad answers with pleasantries, a recap of your question, and a closing summary. All of that costs output tokens for content you don't read.

Try adding to your instructions: "Lead with the answer. Skip the opening pleasantries and the wrap-up."

Ask for a second, more critical pass

A polished answer to the wrong question wastes more tokens than one blunt follow-up. Before accepting a first draft, ask what's weak about it — it often catches in one message what would otherwise take three or four rounds of "refine this a bit more."

Try: "What are the top three weaknesses of this? Be direct."

04 MANAGE THE CONVERSATION ITSELF

Keep one topic per chat

Ask about a work proposal, then a recipe, then a LinkedIn post in the same thread, and the model re-reads the proposal and the recipe every time it answers about the post. One topic, one chat.

Carry the work forward, don't abandon it

When a long session gets slow or forgetful, you don't have to choose between losing progress and paying to keep it

going. Copy your latest working version, open a new chat, and paste it in with a one-line brief of where things stand and what's next.

Let it ask you questions

A fifteen-word prompt that ends with a request for clarifying questions often costs less than a five-hundred-word prompt trying to cover every case up front — and it gets you a better first answer.

Try: "I want to [task], for [who it's for]. Ask me what you need to know before you start."

Watch for a conversation quietly going stale

A long chat doesn't announce that it's overloaded, it just gets worse while still costing full price. A rough early-warning trick some power users use: ask the model to start every reply with a specific word or your name. The moment it stops, the thread has drifted — that's the cue to start fresh.

05 SET IT UP ONCE

Save standing preferences in settings, not in every chat

Re-explaining your tone, format, and background in every new conversation is a small setup tax, paid repeatedly. Most tools have a persistent preferences or custom-instructions field — set your defaults there once.

Build a reusable prompt shape for requests you send often

If you find yourself writing a similar request from scratch every time, save the shape of it: what the task is, the background it needs, the goal, and the format you want back. Swap out only the specific detail each time.

06 FILES, SCREENSHOTS, AND DOCUMENTS

Convert before you upload

A single PDF page can run 1,500–3,000 tokens once it's read as both text and, in some tools, an image of the page. Extract just the text you need into plain text or a simple document instead of uploading the whole file.

Crop screenshots tight

A full screenshot can cost over a thousand tokens; a tight crop of just the relevant part can drop that by an order of magnitude. Crop before you paste.

Don't upload files "just in case"

Every file you attach is tokens spent, whether or not the model needed to read all of it for this specific question. Only attach what the task in front of you actually requires.

07 EVERYDAY TIMING AND TOOLS

Turn off what you're not using

Web search, connectors, and extended thinking modes each add tokens to a response, even when they weren't the

reason you needed help. Leave them off by default and switch them on per task.

Plan out loud before asking for the finished thing

Opening with “build me X” and then spending ten messages steering it toward what you meant costs far more than describing what you want up front and letting the model ask questions first.

Talk instead of type, if your prompts tend to be vague

Typed prompts tend toward short and underspecified (“make it better”), which invites guesswork and more back-and-forth. Speaking a request out loud, if your tool supports voice input, tends to naturally include more of the context a model needs in one shot.

Spread heavy sessions across the day

Usage on most plans resets on a rolling window, typically every few hours, not at midnight. Splitting a big workload into a morning and an afternoon session can make more of your plan available than burning it all in one sitting.

Know where to check your actual usage

Most tools have a usage page in account settings showing exactly how much of your current window and any weekly cap you've used. Check it occasionally so a limit is never a surprise.

Want the full breakdown?

Read the full guide, with an interactive token calculator and a visual breakdown of what a token actually is, at:

futurefactors.ai/ai-tokenomics-what-drives-your-ai-costs