

A PILL FOR GROK BOT USERS, EVEN FROM ZERO

Make Grok Bot use less.

9 free tools that make the Bot read only the useful part of web pages, PDFs and spreadsheets. No coding needed: the Bot installs them, in half an hour.

9 tools

Free and open source.

30 minutes

The Bot installs them, once.

1 prompt

Ready to copy, 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	The Bot's "fuel". Every word the Bot reads or writes uses a few tokens. An ordinary web page can be worth tens of thousands of tokens, even if you only care about three lines.
Weekly usage	The tank of tokens your plan gives you each week. When it's empty, the Bot stops until the reset, unless you move to a higher plan.
Repository	Also called "repo": a project's public folder on GitHub, the site where developers share their work. The nine tools in this guide are free, open source repos.
The Bot's computer	Grok Bot doesn't work on your PC but on its own cloud computer. The tools get installed there: nothing changes on your machine.
Skill	A saved instruction the Bot follows every time, without you rewriting it. You open it by typing / in the chat.

WHAT IT IS

The Bot doesn't work too much.
It reads too much.

Everything you make Grok Bot read is paid for in tokens. A web page is not just the article: it has menus, ads and hidden code. A spreadsheet has thousands of rows when you need one number. These tools sit in the middle: they take the big file, keep the useful part and pass only that to the Bot.

Think of an assistant who, instead of reading you a whole binder out loud, brings you just the page with the answer.

WHY IT MATTERS: THE TEST NUMBERS

4.5M

tokens for 7 real jobs done the normal way

57,900

tokens for the same 7 jobs, through the tools

-98.7%

the total saving, almost all from one test (DuckDB)

READ IT CAREFULLY

The numbers come from Jordan Rafealov's video, where he ran each job twice. Without the spreadsheet test, the cut ranges from about a third to 99%, depending on the file.

And the Grok Bot docs warn: an hourly or short-interval routine can use a week of usage in a day. If the Bot reads whole files on every run, the bill multiplies.

BEFORE AND AFTER, IN THE VIDEO'S TEST

Each one trims one kind of file.

1 Trafilatura

Web pages without the clutter. Keeps only the article text, drops menus, ads and code. For news, product pages, competitor sites.

`"Summarize this page in 5 points: [link]"`

-93%



400,000 → 26,000 tokens

2 jq

Only the fields you need. From a long reply from a site it keeps the fields you ask for: prices, totals, names, emails.

`"From this list, give me only name, price and availability"`

-99%

18,000 → 174 tokens

3 ripgrep

The right lines, not the whole file. Searches many files for a word and returns only the matching lines.

`"In the quotes folder, find where Smith Ltd appears"`

-99%

30,000 → 300 tokens

4 DuckDB

Questions to a spreadsheet. Reads the file on the Bot's computer and passes it only the answer.

`"In sales-2026.csv, what were the 5 best-selling products in October?"`

-99.99%

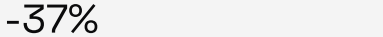
4 million → 89 tokens

5 Docling

PDFs into clean text. Readable headings and tables. For invoices, contracts, reports, quotes.

`"From this PDF contract, tell me the term, price and penalties"`

-37%



12,700 → 8,000 tokens

In blue: what to write to the Bot. It picks the right tool itself.

THE LAST FOUR

Fewer empty runs, less work to redo.

6

changedetection.io

Only what changed. Watches a page and reports the differences. For competitor prices, job listings, suppliers. Stays on all the time.

`"Watch [competitor]'s pricing page and tell me when it changes"`

-43%

35,000 → 20,000 tokens

7

RTK

Shorter command output. Useful if the Bot works on code, websites or servers. If you only use it for email and documents, you can ignore it.

-52%

5,000 → 2,399 tokens

8

Tenacity

Retry without starting over. Inside a script, when a call fails it waits and retries on its own. The Bot doesn't redo the whole job.

3 tries

worked on the third, in 2.2 seconds

9

Pydantic

Check data before using it. Finds missing or wrong fields: contacts without an email, orders without a price.

2 errors

found in a sample reply

SHORT ON TIME? START WITH THREE

For a small business the jobs that burn the most usage are nearly always the same: web pages, spreadsheets and PDFs. Start with **Trafilatura, DuckDB and Docling**. Add the others when you need them.



HOW TO DO IT

The Bot installs it. You work as before.



1 Open a conversation with the Bot

Your usual one is fine.

2 Paste the prompt below

The Bot installs the tools on its computer and tells you line by line whether it worked. If one fails, the others still work.

3 Let it save the skill

From then on it knows which tool to use for each kind of file.

4 Work as before, check in a week

See whether your weekly usage lasts the week. Rules you don't need can go.

READY TO COPY

```
On your computer, install these free tools if they aren't there yet: trafilatura, jq, ripgrep, duckdb,
docling, rtk. For each one, tell me in one line whether the install worked.
Then save these rules as a skill called "read less" and always follow them:
- Web page: extract the text with trafilatura, then read only that.
- Long JSON reply: keep only the fields you need with jq.
- Searching files: use ripgrep and read only the matching lines.
- Spreadsheet or CSV: ask the question with duckdb, read the result.
- PDF: convert it to text with docling, then read the text.
- Long command output: pipe it through rtk.
- Python scripts: tenacity to retry, pydantic to check the data.
- If a step fails, retry at most twice, then stop and tell me.
Before reading a whole file, ask yourself whether part of it is enough.
At the end, explain in plain words what you installed and how you'll use it.
```

SMALL TEST, 10 MINUTES



Same file, twice.

Pick a job you do often with a big file, such as a sales CSV or a long PDF. Ask the Bot your question and watch how much your weekly usage moves. Then paste the prompt and ask the same question on the same file. If the Bot says it used duckdb or docling and usage moves much less, it works.

FREQUENTLY ASKED QUESTIONS

Do I need to code?

No. You paste a prompt and the Bot does the rest. Tenacity and Pydantic only matter if the Bot writes programs for you.

Does it cost anything?

The tools are free. Installing them uses a bit of usage, once.

Can I break something?

The tools live on the Bot's computer, not yours. At worst one fails to install and the Bot works as before.

What if the Bot doesn’t use the tools?

Remind it: "use the read less skill". Or type / in the chat and pick the skill.

Does it work with other assistants?

Yes, with those that can use a terminal, such as Claude Code or Codex. Not with an assistant that only lives in the browser.

WATCH OUT FOR

Real numbers, read carefully.

- **The numbers are the video's.** In the "before" runs the Bot read the whole file: if you already work in a tidy way, you'll save less.
- **The 98.7% comes almost entirely from DuckDB.** On pages, PDFs and commands, expect a cut between a third and 95%.
- **An extract can miss a detail.** If an answer looks incomplete, have the whole file read that one time.
- **Limits and plans change.** Check the Grok Bot docs and your account.

SOURCES

Video "9 GitHub Repos That Cut Grok Bot Usage by 98.7%" by Jordan Rafealov. Grok Bot docs (docs.x.ai). Official GitHub repositories of the nine tools. No affiliation.



Want us to set it up?

We configure the tools and rules on your Bot, measure usage before and after on your real work, and leave you the skills ready to go.

hello@studiofuturo.ai
studiofuturo.ai/en/pillole