

# Cursor Tips

## 10 Power-Tips for Super-Charging Cursor (and similar AI coding tools)

#	Tip	Why it helps	Quick How-to
1	<b>Bootstrap your project's "Rules" file in minutes.</b>	A curated ruleset gives the model instant domain knowledge and style guidance.	Clone <i>awesome-cursor-rules-mdc</i> and copy the relevant Markdown (.mdc) file(s) into <code>.cursor/rules/</code> or Rules > Import. The repo already includes 800+ ready-made rule bundles and even a CLI to auto-generate new ones for any library you name.
2	<b>Use the Rules CLI to generate missing docs for bespoke libs.</b>	Converts a library's README + web search into a structured Cursor rule so you don't have to hand-write it.	<code>uv run src/generate_mdc_files.py my-internal-lib</code> —then commit the output.
3	<b>Slurp an entire GitHub repo into context with one keystroke.</b>	Perfect when you need third-party code (or another micro-service) referenced in Agent mode.	In any GitHub URL, replace <code>github</code> with <code>gitingest</code> (or install the CLI). Cursor (or any LLM) can now <code>@</code> -reference the generated <code>digest.txt</code> .
4	<b>Trim the fat before ingesting.</b>	Fewer tokens → faster, cheaper, more relevant results.	<code>gitingest ... --max-size 200KB --exclude 'tests/**,*.png'</code> or use <code>.cursorignore</code> to keep noise out of both Cursor indexing <b>and</b> the digest.
5	<b>Park large reference chunks inside a Notepad, then</b>	Notepads let you attach files and rich Markdown once	Create a Notepad called <b>API-Glossary</b> , attach the OpenAPI spec, then type <code>@API-Glossary</code>

#	Tip	Why it helps	Quick How-to
	<b>@-mention them.</b>	and reuse them across chats or composer prompts without polluting your global Rules file.	whenever you need endpoints in context.
6	<b>Convert recurring “how-to” threads into dynamic templates.</b>	Speeds up boilerplate tasks (e.g., new React hook pattern, Terraform module).	In the same Notepad, add a <b>Template ▼</b> section; the Agent will expand it when you @-reference.
7	<b>Discover niche rule packs &amp; MCP servers on cursor.directory.</b>	Someone probably already wrote rules for your stack (NestJS, Supabase, Chakra, etc.). Use them instead of starting from scratch.	Search the <i>Rules</i> tab or click <i>Generate</i> to auto-create a starter pack; plug any listed MCP into Settings › Docs.
8	<b>Keep your rule packs fresh with the directory’s “Trending” feed.</b>	Ecosystems evolve—new ESLint, new security best practices.	Subscribe to the RSS or check weekly; merge diffed rules straight into <code>.cursor/rules/</code> .
9	<b>Chain gitingest → Notepad → @reference for external SDKs.</b>	Gives you full-text access to vendor SDKs without permanently indexing them.	curl https://gitingest.com/stripe/stripe-python > stripe.txt, attach stripe.txt to a Notepad, then @stripe.txt when needed.

#	Tip	Why it helps	Quick How-to
10	<b>Treat Rules as "always-on," Notepads as "sometimes," digests as "on-demand."</b>	Clear mental model avoids context bloat and token overruns in MAX mode.	<b>1</b> Rules = project conventions & invariants; <b>2</b> Notepads = reusable docs/templates; <b>3</b> Digests = transient deep dives (delete when done).

*Drop these into your workflow and Cursor's MAX mode will spend its extra tokens on **useful** context instead of random noise.*