

M A V E R I C K A I

MAKE 100 VERSIONS OF CLAUDE FIGHT TO THE DEATH

THE ARENA SKILL: FULL SETUP GUIDE

When Claude keeps giving you answers that miss, asking again usually gets you another version of the same miss. The Arena skill hands one task to up to 100 Claude sub-agents, has each one attack it a different way, then makes them fight until one answer is left standing.

Install the free skill with one paste

Run a quick 16-agent arena before the full 100

Write a task brief every agent can actually use

Know the real cost of each bracket size

Eleven copyable prompts and commands

October 2026

Start Here: How the Arena Works

THE PLAN

1. Paste the GitHub link into Claude Code and install the skill.
2. Check that /arena shows up and the plan command runs.
3. Turn on accept-edits mode so you are not approving hundreds of files.
4. Run a quick 16-agent arena on a real task.
5. Save the full 100-agent run for the answer that really matters.

Before you start

Tool: Claude Code, in the terminal or the Claude desktop app. The skill needs Python 3.8 or newer. It is free and MIT licensed, with no API key or signup.

Usage: The skill costs nothing, but the tokens come out of your plan. A full 100-agent run is 595 sub-agent calls, so start with the 16-agent quick mode.

Time: About five minutes to install and check. A full run takes much longer than a normal answer because it runs in waves of 10.

If you came from the ARENA video, this is the full setup. Install it once, then reach for it whenever Claude's answer is not where you want it to be.

You type one command. Claude spins up sub-agents (100 by default) and gives every one of them the exact same task, word for word. Each one also gets its own strategy card: a way of reasoning, a workflow and a strategy. The card is what makes each agent come at the problem from a different angle.

Then they pair off. Each agent attacks its opponent's answer, defends its own, and rewrites it to fix whatever it conceded. A separate judge scores the match on a written rubric and the loser is out. 100 becomes 50, then 25, 13, 7, 4, 2 and finally 1.

I tested it by giving 100 of them one job: write the perfect hook for my next video. The hook that survived beat everything I had come up with in the previous 24 hours of writing.

Inside the Arena: Cards and Judging

Two things decide how a run plays out: the card each agent is dealt and the rubric the judge scores with.

What's on a strategy card

Part of the card	Examples
Reasoning mode (15 options)	First principles, inversion, analogy, contrarian, working backwards, expert panel
Workflow (12 options)	Draft, critique, rewrite. Outline first. Build, then break. Three drafts, pick one.
Strategy (12 options)	Simplest thing that works, maximal rigor, user empathy first, edge cases first

That adds up to 2,160 possible cards, and no two agents get the same one. All 100 agents are the same Claude model you are already using. The cards are the only thing that makes them different.

How the judge scores each match

Criterion	Points
Correctness	30
Completeness against the task	25
Robustness to the attacks raised	20
Specificity	15
Clarity	10

An answer with a verified fatal flaw cannot beat one without, whatever the totals say. The judge never sees the strategy cards, and a longer answer earns nothing for being longer.

TIP

The skill never edits your files. Every agent writes inside a folder called `.arena` in the folder where you run it. The winning answer comes back to you, and using it is your call. The skill is made by Jake Schincariol and is free under the MIT license. Find it at github.com/Jakeschincariol/arena-skill.

Copy This First: The One-Paste Install

Open Claude Code and paste this. It is the install message from the skill's own instructions.

PROMPT 1 | COPY & PASTE

```
https://github.com/Jakeschincariol/arena-skill  
Install this skill, then confirm /arena works.
```

Claude should download the skill and copy its folder into your personal skills folder, then check that it loads. Approve the steps it asks about. If /arena does not appear when you type a slash, start a fresh Claude Code session and check again.

Prefer to install it yourself?

Run these commands in your terminal. They download the repository and copy the arena folder into your personal skills folder, so the skill works in every project. The first line just makes sure that folder exists.

PROMPT 2 | COPY & PASTE

```
mkdir -p ~/.claude/skills  
git clone https://github.com/Jakeschincariol/arena-skill.git  
cp -r arena-skill/skills/arena ~/.claude/skills/
```

Or install it as a plugin

Run these two commands inside Claude Code.

PROMPT 3 | COPY & PASTE

```
/plugin marketplace add Jakeschincariol/arena-skill  
/plugin install arena-skill@arena-skill
```

As a plugin, the command shows up as /arena-skill:arena instead of /arena. Use the folder copy above if you want the short version.

TIP

Want it in one project only? Copy the same arena folder into that project's .claude/skills folder instead of your personal one.

Step 1: Check That It Works

Type `/` in Claude Code and look for `arena` in the list. Then ask Claude to print the plan for a run. The `plan` command only does the math. It does not start an arena or write any files.

PROMPT 4 | COPY & PASTE

```
Find the Arena skill's bracket.py and run its plan command twice: once with --quick and once with --agents 100. Show me both tables. Do not start an arena.
```

You should see two tables. The quick one totals 91 sub-agent calls over 4 rounds. The full one totals 595 calls over 7 rounds.

If you copied the folder yourself, you can run the plan straight from your terminal:

PROMPT 5 | COPY & PASTE

```
python3 ~/.claude/skills/arena/bracket.py plan --agents 100
```

If that says Python is missing or too old, install a current version of Python 3 first. There is nothing else to install.

Step 2: Turn On Accept-Edits Mode

Every agent saves its work as a file inside `.arena`. In Claude Code's default permission mode, that means one approval per file, which adds up to hundreds of prompts on a big run.

Before you start, press Shift+Tab until Claude Code shows that it will accept edits. The skill reminds you about this, but it will not change your settings for you.

If the task involves code, run the arena from that project so the agents can read your files. They still write only inside `.arena`, and code changes come back to you as a diff or full files for you to approve.

TIP

For writing tasks that do not need project files, run arenas from a dedicated folder, such as an `arena-runs` folder in your home directory. That keeps every `.arena` record out of your real projects.

Step 3: Run Your First Arena

There are two ways to start one.

After a bad answer. Just type `/arena`. It takes your last request as the task and the answer you did not like as the one to beat.

With a fresh task. Type `/arena`, any options, then the task. Start with `--quick`, which runs 16 agents over 4 rounds.

Try it on a hook

PROMPT 6 | COPY & PASTE

```
/arena --quick Write one opening hook for my next short-form video. The video is about [topic]. My audience is [who they are and what they care about]. The hook has to work spoken out loud in about [number] words and make a specific promise the video pays off. Give the final hook plus two backup versions.
```

Claude prints the size before it starts, such as "16 agents, 4 rounds, 91 sub-agent calls." After each round you get one line, such as "Round 2 done: 4 of 16 left." The attacks and verdicts stay out of your chat.

Claude may also offer an arena when you say something like "that's a bad answer, make them compete." When it starts itself that way, it asks first and gives you three choices: the full arena, the quick arena or an ordinary retry.

TIP

If the conversation gets compacted halfway through, nothing is lost. The whole tournament lives in one file, and the skill picks up where it stopped.

Step 4: Write a Task Every Agent Can Use

This step decides the result. The agents cannot see your chat. The skill writes a standalone task file at `.arena/<run>/task.md`, and that file is all they ever know. If a requirement never made it into the file, all 100 agents miss it.

Put everything in the task: who it is for, the format, the length, the tone and what "done" looks like. If you rejected an earlier answer, say what you disliked about it in your own words.

PROMPT 7 | COPY & PASTE

```
/arena --quick [What you want, in one sentence.]  
  
Who it is for: [audience, and what they already know]  
Format and length: [for example, one paragraph under 120 words, or a 5-row table]  
Must include: [facts, names, numbers, links or files it has to use]  
Must avoid: [words, claims, styles or approaches you do not want]  
Tone: [for example, direct and warm, like talking to a friend]  
Done looks like: [how you will judge whether it worked]  
What I disliked about the last answer: [in your own words, or delete this line]
```

Check the brief before a big run

Before you spend a full 100-agent run, have Claude write the brief in a normal message first. You can read it, fix it, then start the arena.

PROMPT 8 | COPY & PASTE

```
I want to run the Arena skill on this, but do not start it yet: [describe what you want].  
  
Write a standalone task brief that a stranger could follow without seeing this conversation. Include every requirement and constraint I have given you anywhere in this chat: audience, format, length, tone, what must not change and what done looks like. Add any context they would need, such as file paths or pasted data. Do not add requirements I never gave, and do not include your own view of the right answer. Show me the brief and wait.
```

Once it looks right, type `/arena --agents 100` and paste the approved brief after it. Leaving your own view of the right answer out of the brief matters. Otherwise you push all 100 agents the same way, which defeats the point.

Step 5: Read the Result

When the bracket finishes, you get:

1. The winning answer, in full.
2. Why it won: the attacks it survived, plus its strategy card on one line.
3. The round count, such as "7 rounds, 100 agents in, 1 left."
4. If you started from a rejected answer, a blind final judge's scores for the winner and your old answer. If the old one scored higher, the skill tells you plainly.
5. Where the full record lives.

If the winner would change files in your project, the skill does not apply it. It asks whether you want to apply it or change it.

Look at the final match

PROMPT 9 | COPY & PASTE

```
Open the latest run folder in .arena. Show me the final match: both finalists' revised answers and the judge's scores and reason. Do not start a new arena or change any files.
```

TIP

Treat the winner as the strongest answer this tournament found, not proof that it is right. The judges are Claude too. Read the attacks it survived and check any facts before you publish.

Pick the Right Size

Agents	Rounds	Sub-agent calls	Waves of 10
100 (default)	7	595	70
64	6	379	49
32	5	187	28
16 (--quick)	4	91	16
8	3	43	10

Every call reads the task and one or two answers, so a long task costs more per call. A rejected answer to beat adds one more call at the end. The skill prints these numbers before it starts.

The options

Option	What it does
--quick	16 competitors. The everyday setting.
--agents N	Any number of competitors. The default is 100.
--seed S	The same seed gives the same cards and the same bracket. The answers still vary.
--wave W	Agents per wave. The default is 10. Only raise it if you raised Claude Code's limit.

The waves exist because Claude Code runs at most 10 tool calls at once by default, set by `CLAUDE_CODE_MAX_TOOL_USE_CONCURRENCY`. That is why a 100-agent arena runs as 70 waves and takes a while.

TIP

Use --quick for everyday jobs. Save the full 100 for an answer you are betting on, like a launch plan or the hook for a big video.

What to Throw Into the Arena

The arena works best on a task with a clear goal and a way to judge the result. These are good starting points.

Use it for	Example command
Hooks and headlines	<code>/arena --quick write the headline for our pricing page</code>
Plans	<code>/arena --seed 7 plan my launch week, I have 6 hours a day</code>
Code fixes	<code>/arena --agents 32 fix the flaky test in tests/test_api.py</code>
Subject lines	<code>/arena --quick write a subject line under 45 characters for [email topic]</code>

Make a hard decision

PROMPT 10 | COPY & PASTE

```
/arena --agents 32 Recommend which of these options I should choose: [option A],  
[option B], [option C]. My goal is [goal]. My constraints are [budget, time, skills].  
Give one recommendation, the main trade-off it accepts, and the first actions to take  
this week.
```

When to skip it

Skip the arena for quick facts, simple questions and anything one normal retry with better instructions would fix. It also cannot rescue a vague task. Vague task in, 100 flavors of vague out.

Common Mistakes and Easy Fixes

If this happens	Try this
Hundreds of approval prompts	Press Shift+Tab to accept edits before the run starts.
/arena does not show up	Start a new Claude Code session. Check that ~/.claude/skills/arena contains SKILL.md.
You used the plugin install	The command is /arena-skill:arena.
The winner ignored a requirement	It was probably missing from the task file. Check .arena/<run>/task.md and run again with it written out.
Every answer feels generic	Add the audience, format, length and what done looks like.
A run used more of your plan than expected	Drop to --quick or --agents 8 and keep the task short.
The run stopped halfway	Ask Claude to resume the arena. The skill continues from its saved state.
Your old answer scored higher	Trust the score. Keep the old answer or rewrite the task and run again.
You raised Claude Code's tool limit	Pass --wave with the same number.
python3 is not found	Install Python 3.8 or newer.

Ask Claude to diagnose it

PROMPT 11 | COPY & PASTE

My Arena skill run is not working. Here is what happened: [describe it or paste the error].

Check where the skill is installed, whether python3 is available and at least version 3.8, and whether an .arena folder exists here. If a run was interrupted, run the skill's status command and tell me where it stopped. Explain the problem in plain words and give me the fix. Do not start a new arena or delete anything without asking me.

Quick Reference: The Arena Skill

Command	What it does
/arena	Turns your last request into a 100-agent arena, with the answer you rejected as the one to beat.
/arena --quick [task]	16 agents, 4 rounds, 91 sub-agent calls.
/arena --agents 32 [task]	32 agents, 5 rounds, 187 sub-agent calls.
/arena [task]	The full 100 agents, 7 rounds, 595 sub-agent calls.
/arena-skill:arena	The command name if you installed the plugin version.

Setup in four lines

Install: Paste the GitHub link into Claude Code with "Install this skill, then confirm /arena works."

Check: Type / and look for arena, then run the plan command.

Permissions: Press Shift+Tab to accept edits before a run.

Record: Everything lands in .arena inside the folder you ran it from.

DO THIS TODAY

Install the skill, then find one answer from this week you were not happy with. Rewrite the task so a stranger could follow it, run /arena --quick on it, and compare the winner with what you had.

Want More AI Tips Like This?

I post daily AI content, tool breakdowns, and prompts that actually help you get more done. Follow me for the latest.

@MavGPT

TikTok | Instagram | Facebook

Give it one task. Let the best answer win.