



// SECTION 01 • CLAUDE SKILL

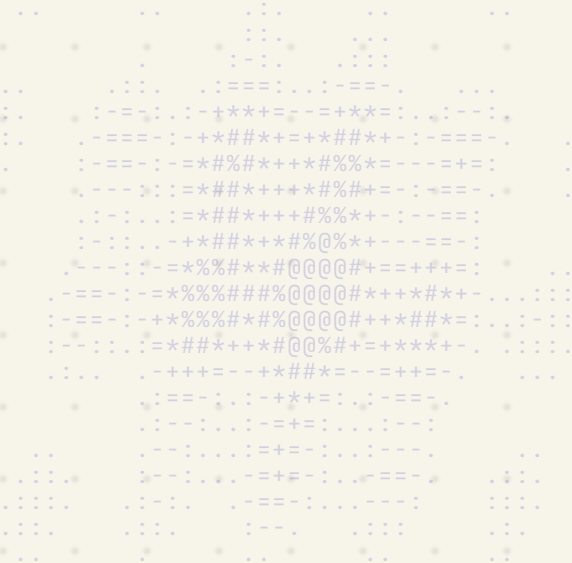
DAY TRADER SKILL



Eleven free Claude skills for day trading. Scan thousands of charts, track breaking news, time every entry and exit, and replay every outcome 10,000 times before you risk a dollar. Paper trading by default.

```
> boot opusjake_os
> resource: day-trader
> version: v1.0 • 2026
> status: ready_
```

Eleven skills, one trading desk. One scans thousands of charts for setups. One tracks the news on every stock you watch. One tells you the moment to get in, another sets your stop and target before you do. And the Time Machine replays the rest of the day 10,000 times so you see the honest odds first. Free, MIT, no signup, no API key. Nothing trades without your yes.



INSTALL

01

Copy the prompt below

The repo link plus one line

02



Paste it into Claude

Claude installs all eleven skills

03

/trader-scanner

Your first scan of the market

04

/trader-timemachine

The odds before you risk a dollar

Two lines. Paste this into Claude and it does the rest.

```
https://github.com/Jakeschincariol/day-trader-skill
```

```
Install these skills, then confirm /trader-scanner works.
```

// BY HAND

```
git clone https://github.com/Jakeschincariol/\
day-trader-skill.git
```

```
cp -r day-trader-skill/trader-* ~/.claude/skills/
```

// AS A PLUGIN

```
/plugin marketplace add Jakeschincariol/day-trader-skill
/plugin install day-trader-skill@day-trader-skill
```

As a plugin the commands show up namespaced, like `/day-trader-skill:trader-scanner`. Copy the folders instead if you want plain `/trader-scanner`. The tools are plain Python 3.8 or newer with nothing to pip install, and they read free public data: Yahoo Finance charts, Nasdaq Trader's symbol lists, SEC EDGAR and the House Clerk.

THE ELEVEN

Each one is a folder with a `SKILL.md` inside. Type the command, or just describe what you want and Claude picks.

COMMAND	THE JOB
<code>/trader-scanner</code>	Chart Scanner. Thousands of charts, one list of setups.
<code>/trader-news</code>	News Monitor. Breaking news on every stock you watch.
<code>/trader-entry</code>	Entry Timer. Watches a setup bar by bar, calls the moment.
<code>/trader-exit</code>	Exit Planner. Stop and target set before you enter.
<code>/trader-timemachine</code>	Time Machine. 10,000 replays plus every past time it fired.
<code>/trader-risk</code>	Risk Manager. Blocks anything over your limits.
<code>/trader-budget</code>	Budget Manager. Sizes every position to your max loss.
<code>/trader-broker</code>	Order Executor. Paper by default, never without your yes.
<code>/trader-alerts</code>	Price Alerts. Rings on the stop, the target, the bell.
<code>/trader-analyst</code>	Earnings Analyst. Reads the report, sums up what changed.
<code>/trader-pelosi</code>	Pelosi Tracker. Congress's trades the minute they are filed.

ONE TRADE, START TO FINISH



Everything lands in a `trader/` folder in your project, and each skill reads what the last one wrote. Here is the trade from the video: Tesla, Monday October 19 2026, real data.

1. **Scan.** At 10:15 AM ET, TSLA had closed above its 15-minute opening range: 354.47 to 362.18, the 10:10 candle closed at 364.18.
2. **News.** The morning's headlines, tagged earnings and deal. Tesla reports this week, but a day trade is out before the close.
3. **Entry.** At 10:12 it said waiting. At 10:15 it said triggered: the first 5-minute close above the range.
4. **Exit.** Stop at the range midpoint, 358.32, so 5.86 a share at risk. Target 370.04. Flat by 3:55 PM either way.
5. **Time Machine.** 10,000 replays of the rest of the day. The verdict: close to a coin flip.
6. **Budget.** A \$5,000 account with a \$50 max loss buys 8 shares. Stopped out, that is a \$46.88 loss.
7. **Risk.** Max loss, daily limit, trades per day, reward to risk, entry window. PASS or BLOCK, with the reason.
8. **Broker.** Claude shows you the ticket in plain words. You type yes for that exact order, and Alpaca's paper account takes it.

THE TIME MACHINE

The one that makes this different. Before you risk a dollar, it replays the rest of the session 10,000 times, each 5-minute candle drawn from the stock's own candles at that time of day over the last 59 sessions. Then it re-runs the same rule on every one of those real days.

ON THE TSLA TRADE	RESULT
Hit the target first	29.4%
Hit the stop first	25.5%
Still open at the close	45.1%
Times the setup fired in 59 real days	33
Real days it hit the target / the stop	13 / 15

Seed 19, so anyone can repeat it. A coin flip is not a reason to skip the trade or take it. It is a reason to know what you are buying before you buy it.

```
python3 trader-timemachine/timemachine.py run --write
```

THE FINE PRINT

Not financial advice. This is software that reads public data and does arithmetic. It cannot see the future, and most people who day trade lose money.

Nothing trades without your yes. Only `/trader-broker` can lead to an order, and only after `/trader-risk` passes that exact order and you type yes. Set your brokerage connector to "Needs approval" in Claude, never "Always allow". No options, margin or shorting unless you turn them on.

Paper first. `"paper": true` is the default. Keys live in environment variables, never in the chat.

The Time Machine is a simulation, not a prediction. It is built from the last 60 days of bars. Earnings, halts and headlines can do things those days never did.

A stop is a plan, not a guarantee. In a fast market or across a gap it can fill past its price.

The data is free and unofficial. Check anything that matters against your broker before you act on it.