THE BACKTEST MACHINE

Test any trading strategy with Claude + TradingView in ~20 minutes — then make the winner executable. Companion sheet to the video. Everything here is copy-paste ready.

PART 1 — BUILD YOUR OWN PINE SCRIPT WITH CLAUDE

The loop is always the same four steps: **articulate** → **code** → **run** → **read**. Works on any strategy — from a book, a YouTube video, or your own head.

STEP 1 — Articulate. A strategy must become objective rules before it can be tested. Describe it loosely; Claude formalizes it:

PROMPT 1 — ARTICULATE

```
Take [describe your strategy in one sentence] and turn it into a fully
objective rule set for [ASSET] on the [TIMEFRAME] timeframe. Give me:
exact entry rule, exact exit rule, stop logic, and position sizing.
No discretion — every rule must be computable.
```

STEP 2 — Code it. Same chat, one more message:

PROMPT 2 — CODE IT

```
Now write this as TradingView Pine Script v6 strategy code. Include
0.1% commission per side, fills on next bar open, $100,000 starting
capital. Ready to paste straight into the Pine Editor.
```

STEP 3 — Run it. TradingView (free tier is fine) → open the chart of YOUR asset on YOUR timeframe → Pine Editor (bottom panel) → paste → *Add to chart* → open the *Strategy Tester* tab. The arrows on the chart are every trade it would have taken; the tester is the report card.

STEP 4 — Read the verdict. Paste the tester's performance summary back into Claude:

PROMPT 3 — READ THE VERDICT

```
Here are the backtest results: [paste performance summary].
Explain what the win rate, profit factor and max drawdown are telling
me, and give an honest verdict: is this strategy worth trading, worth
fixing, or worth binning?
```

PROMPT 4 — IMPROVE (honestly)

```
Suggest the single most promising improvement to this strategy, then
test it and tell me honestly whether it actually improved the results.
```

PART 2 — THE WINNER: EMA 9/21 CROSS

The strategy that topped our 12-strategy test on Bitcoin. It owns BTC when momentum turns up (9 EMA crosses above 21 EMA) and sits in cash when it rolls over. Wrong on most trades — it wins by cutting losers instantly and riding trends for weeks.

```
//@version=6
strategy("EMA Cross 9/21", overlay=true,
     initial_capital=100000,
     default_qty_type=strategy.percent_of_equity, default_qty_value=100,
     commission_type=strategy.commission.percent, commission_value=0.1,
     process_orders_on_close=false)

fastEMA = ta.ema(close, 9)
slowEMA = ta.ema(close, 21)

goLong = ta.crossover(fastEMA, slowEMA)
goFlat = ta.crossunder(fastEMA, slowEMA)

if goLong
    strategy.entry("Long", strategy.long)
if goFlat
    strategy.close("Long")

plot(fastEMA, "EMA 9",  color=color.new(color.green,  0), linewidth=2)
plot(slowEMA, "EMA 21", color=color.new(color.orange, 0), linewidth=2)
```

Reproduce the video's test: chart **BTCUSD**, timeframe **1D (Daily)** — the timeframe is not optional, this strategy lives on the daily — tester range July 2023 → today. Signals evaluate on the daily close; fills happen on the next open. No leverage, long-only.

The three numbers that matter

Win rate — how often you're right. High is NOT required: our winner wins ~35% of trades.
Profit factor — dollars won per dollar lost. Above 2 is serious; the winner runs ~3.
Max drawdown — the worst peak-to-trough pain. This decides whether you'd actually survive the ride. The winner's edge over holding wasn't just return — it was **half the drawdown**.

PART 3 — THE CAVEATS (read before you trust any backtest)

1 · There is no one-size-fits-all strategy. A strategy is never good or bad — it is matched or mismatched to an asset, a timeframe, and a market regime. Proof from our own test: the same EMA 9/21 rules beat Bitcoin buy-and-hold, then LOST to buy-and-hold on the Nasdaq (+40% vs +75%) — because BTC moves ~1.7%/day and trends violently, while the Nasdaq grinds at ~0.9%/day. Trend-following pays a whipsaw tax; only violently trending assets cover the bill. Test YOUR asset on YOUR timeframe. Never borrow a backtest from a different market.

2 · Timeframe flips results. Supertrend lost money on the BTC daily chart and topped the table on the weekly. Same rules, same coin. Always re-test when you change timeframe — never assume.

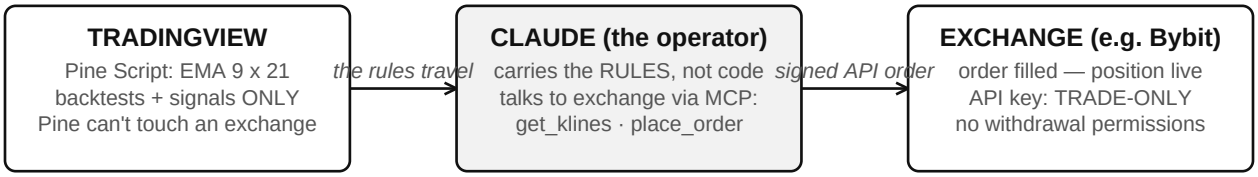
3 · Beware the lucky winner (selection bias). The best strategy out of 12 is always partly luck. Trade count is your first defense: 20+ trades is evidence; 1–3 trades is an anecdote. The only real cure is **forward testing** — run it on live, unseen data (paper first) before real money.

4 · Overfitting check. If a strategy only works with one magic parameter (9/21 works but 8/20 and 10/22 fail), it's curve-fit to the past. Ask Claude to test the neighbouring settings — a real edge is a plateau, not a spike.

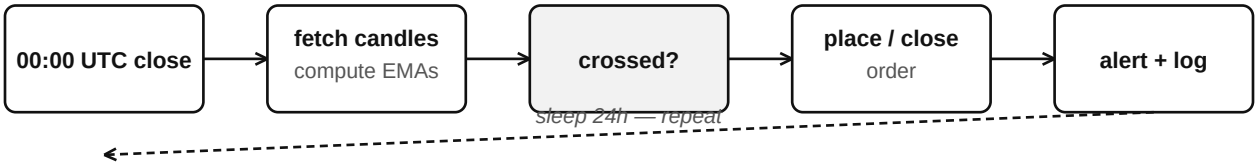
5 · A backtest is a verdict on the past — not a promise about the future. Fees, slippage and data feeds differ between platforms; expect the shape to hold, not the exact dollar figure. And drawdown decides survival before return decides profit: pick the strategy you can psychologically hold through its worst stretch.

PART 4 — FROM BACKTEST TO EXECUTION

Pine Script can simulate, but it can never place an order — TradingView is a lab, not a trading desk. To execute, the strategy's RULES (not the code) move to an operator that can touch an exchange:



THE DAILY LOOP — once per day, at candle close (matches the backtest)



"no cross" days: the bot does NOTHING. That discipline IS the strategy.

Step-by-step: making a strategy executable

1. Validate first. Only automate a strategy that survived Parts 1–3: real trade count, plateau not spike, matched asset + timeframe.

2. Open Claude Code and paste the build prompt below. It builds the bot around your exact rules.

3. Verify the logic: run the bot's `--backfill` command and check its trade list matches your TradingView backtest. If they disagree, stop and fix — never run logic you haven't verified.

4. Run in **paper mode** (the default) for at least several weeks. This is your forward test — the cure for selection bias.

5. Only then consider testnet → live, with a trade-only API key and a hard capital cap. Never skip the ladder: paper → testnet → small live.

THE CLAUDE CODE BUILD PROMPT (paste as-is, edit the rules block)

```
Build me a trading bot in Python. Repo: my-strategy-bot.

THE STRATEGY (do not modify the logic):
- Asset: BTC/USDT · daily candles only
- Entry: 9 EMA closes above 21 EMA -> long, 100% of allocated capital
- Exit: 9 EMA closes below 21 EMA -> flat. Long-only, no leverage.
- Evaluate signals ONCE per day on the daily close. Never intra-candle.

ARCHITECTURE: clean modules (data / signals / execution / state /
alerts). Daily OHLCV from the exchange public API. SQLite state so
restarts never double-enter. Telegram alert on every signal + daily
heartbeat. A --backfill command that replays the last 3 years and
prints the trade list so I can verify against TradingView.

MODES in config: paper (default) / testnet / live. Live mode must
REFUSE to start unless the API key is trade-only (no withdrawals)
and MAX_CAPITAL is set. One position max. Any unexpected API
response: log, alert me, halt. Start with a module plan, then build.
```

Non-negotiable safety rails

· Exchange API keys: trading permission ONLY — withdrawals disabled, IP-whitelisted.
· The bot sizes from a capital cap you set — never from your account balance.
· If any tutorial ever asks you to deposit funds INTO a contract or bot address to “activate” it — it's a scam. A real bot trades inside your own exchange account with your own keys.
· Start smaller than feels necessary. The forward test is the real test.

This sheet is education, not financial advice. Backtested performance is simulated and does not guarantee future results. Trade only what you can afford to lose.