

◆ FREE GUIDE

AI ESSENTIALS · GUIDE NO. 11

# THE 5 AI **LOOPS** I RUN EVERY DAY

The five self-running agent workflows I actually use daily — with the exact setup for each — plus the full 70-loop library, organized so you can find yours fast.

~10 MIN READ · FOR EVERYONE WHO COMMENTED "LIBRARY" · SETUP FOR YOUR FIRST LOOP: ~2 MINUTES

**Hank Barker // AI Essentials // @hankhelpsai**

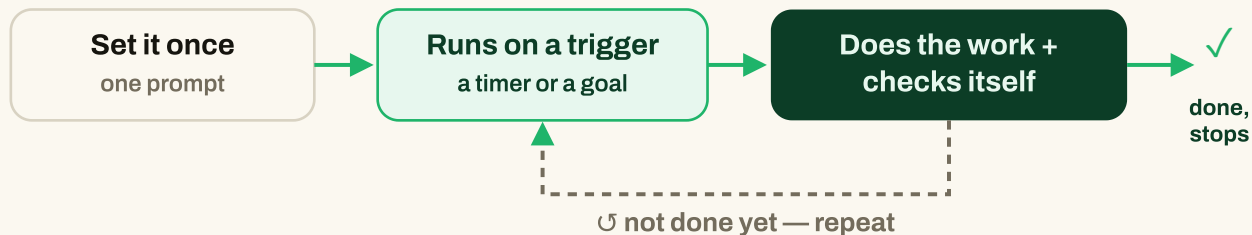
THE SHORT VERSION

- A **loop** is a task you hand an AI agent that it repeats on its own until a stop condition is met — you set it once, it runs.
- Three commands power almost every loop: `/loop` (timer), `/schedule` (cloud), `/goal` (until a target is hit).
- My daily five: **nightly log, feedback sweep, automation scout, pre-publish fact-check, format tester.**
- Start with the **nightly log** — easiest win, and you'll feel it the next morning.

## WHAT A LOOP ACTUALLY IS

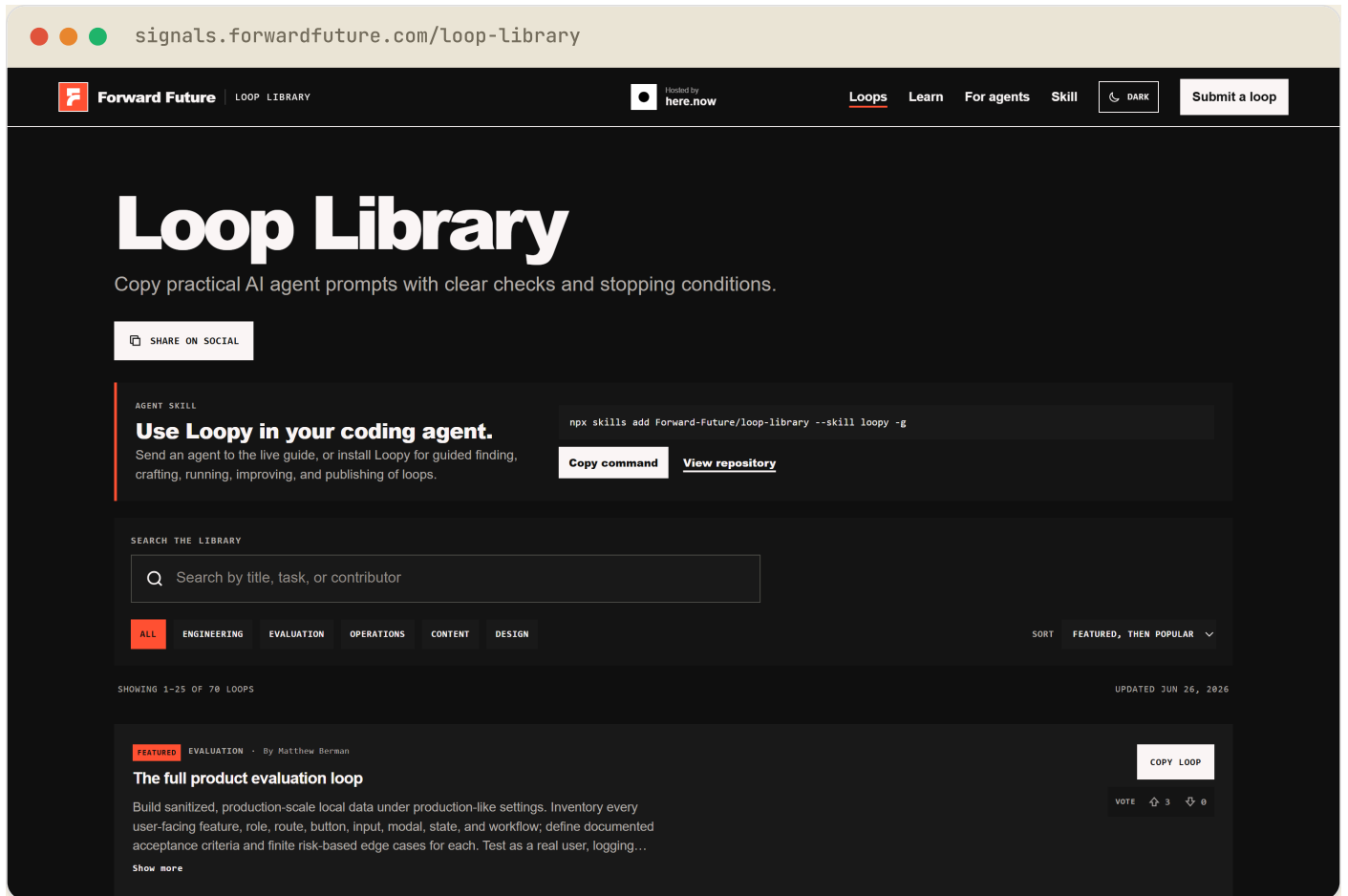
A loop is a task you hand to an AI agent that it **repeats on its own** until a stop condition is met. You're not re-prompting it every time — you set it once and it runs, checks its own work, and keeps going (or stops) based on a rule you gave it. That's the whole idea, and it's what turns AI from a thing you operate into a thing that operates *for* you.

A LOOP — SET IT ONCE, IT RUNS ITSELF



Set it once, give it a trigger and a finish line — it runs and checks itself until done.

The best public catalogue of these is the **Loop Library** from Forward Future — 70 practical agent loops, each with clear checks and a stopping condition. It's where the full list at the end of this guide comes from:



The Loop Library — 70 copy-paste agent loops with built-in checks and stop conditions. — **Source:** [Forward Future — Loop Library](#)

# THE 3 COMMANDS THAT POWER EVERY LOOP

In Claude Code, almost every loop is one of three commands pointed at the right job. That's the entire toolkit.

`/loop`

**On a timer**

...on your machine

`/schedule`

**In the cloud**

...even when your PC is off

`/goal`

**Until it's hit**

...a measurable target

Pick by where and how long it should run.

COMMAND	RUNS	BEST FOR
/loop [interval] [task]	On a timer, on your machine	Frequent, lightweight checks while you work
/schedule [when]: [task]	In the cloud — even when your computer is off	Daily/weekly jobs that must run regardless
/goal [condition]	Until a measurable target is hit	Open-ended work with a clear finish line

## THE 5 I RUN EVERY DAY



### 1. The nightly log

Wake up to a diary of progress



### 2. The feedback sweep

Never fix the same thing twice



### 3. The automation scout

Finds its own next job



### 4. The pre-publish fact-check

Catches the wrong stat before your audience does



### 5. The format tester

Consistency from a system, not inspiration

The daily drivers — each is just one of the three commands pointed at a job worth repeating.

## 1 The nightly log

**What it does:** every night it writes up what changed across my projects — what shipped, what moved, what's still open — into a running file. I wake up to a diary of progress instead of trying to remember where I left off. **Why it earns daily use:** context is the thing you lose overnight; this hands it back every morning for free.

```
/schedule daily at 9pm: review what changed across my projects today and append a dated entry to progress-log.md - what shipped, what moved, what's still open. Keep it to 10 bullet points.
```

**TIP**

Point it at a single `progress-log.md` so it builds one continuous history it can read back later.

## 2 The feedback sweep

**What it does:** takes every correction I gave that day — every "no, do it this way" — and turns the recurring ones into permanent rules, so I stop fixing the same thing twice. **Why it earns daily use:** most people re-explain the same preferences forever; this makes the agent learn instead.

```
/loop 4h go through my corrections from this session. Any preference I've stated more than once, add it as a rule to CLAUDE.md so you follow it automatically going forward.
```

**TIP**

Once a rule lands in `CLAUDE.md` it's applied on every future task automatically — no re-prompting.

## 3 The automation scout

**What it does:** watches the work I keep doing by hand and flags what's worth automating — the agent basically finds its own next assignment. **Why it earns daily use:** you're the bottleneck on things you don't even notice; this surfaces them.

```
/schedule weekly on Friday: review my last week of sessions and find repetitive work I keep doing manually. For each one, tell me what a loop or skill to automate it would look like. Rank by time saved.
```

**TIP**

This is the highest-leverage loop here — it's how you find the *next* four loops worth building.

## 4 The pre-publish fact-check

**What it does:** before anything I write goes out, it checks every factual claim against my real sources — my notes, my docs, the actual data — and flags anything it can't back up. **Why it earns daily use:** one wrong stat kills your credibility; this catches it before the audience does.

```
Use a workflow to go through this draft and verify every factual claim against my sources. For each claim: mark it supported, unsupported, or needs-a-source, and quote the evidence.
```

**TIP**

Run it as a workflow so a *separate* agent checks the claims — fresh eyes catch what the writer misses.

## 5 The format tester

**What it does:** tests content formats — hooks, structures, angles — and keeps score until it finds one that repeatably works, instead of guessing every week. **Why it earns daily use:** consistency comes from a system, not inspiration; this builds the system.

```
/goal find a repeatable post format that beats my current baseline. Draft 3 variations per round, I'll tag the winners, keep only what wins across two rounds.
```

**TIP**

Give it a real baseline to beat ("my median post gets X") so "better" is measurable.

## THE FULL 70-LOOP LIBRARY, ORGANIZED

The 5 above are my daily drivers. Here's the rest of the library, grouped so you can find the ones that fit your work — every one comes with its own checks and stop condition:

The screenshot shows the 'Loop Library' on the website 'signals.forwardfuture.com'. The page features a dark theme and a navigation bar with 'Forward Future', 'LOOP LIBRARY', 'Hosted by here.now', and buttons for 'Loops', 'Learn', 'For agents', 'Skill', 'DARK', and 'Submit a loop'. The main content area displays a list of loops, each with a title, author, description, and a 'COPY LOOP' button. The visible loops are:

- The dependency management loop** (Engineering) - Review every Dependabot pull request currently open in [repository].
- The architecture satisfaction loop** (Engineering) - Refactor until you are happy with the architecture.
- The React Doctor 100/100 loop** (Engineering) - Bring every production React app in [repository] to a freshly verified React Doctor score of 100/100.
- The restartable handoff loop** (Operations) - Before ending [session or work period], create a restartable handoff.
- The React Doctor repair loop** (Engineering) - Run 'pnpm exec react-doctor --verbose --yes --offline --fail-on none' to record the baseline.

Each loop is a copy-paste prompt with a built-in finish line — filter by Engineering, Evaluation, Operations, Content, or Design. — **Source:** [Forward Future — Loop Library](#)

- **Engineering & code** — docs sweep, architecture refactor, sub-50ms page-load, production error sweep, 100% test coverage, repo cleanup, ticket-to-PR, dead-code housekeeper, dependency-CVE burndown, Dependabot triage, React health repair.
- **Design & frontend** — pixel-safe CSS trim, UI reconstruction from screenshots, thumbnail generator, UI/UX scorer, first-time onboarding fixer, accessibility (WCAG) repair.
- **Evaluation & quality** — quality-streak tester, full product evaluation, self-improving prompt champion, devil's-advocate critic, multi-LLM convergence check, promise-to-proof auditor, flaky-test stabilizer, loop-auditor (KEEP/PIVOT/RETIRE/KILL).
- **Operations & maintenance** — batch release, production data cleanup, post-release baseline, 5-minute repo maintainer, recent-feedback sweep, nightly changelog, SEO/GEO visibility fixer.
- **Research & content** — research-to-artifact, product-update podcast, pre-publish source-check, talk-to-five-buyers copy loop, one-post-a-week format test, living-story daily narrative.
- **Specialized** — adversarial-review, builder-reviewer handoff, completion-contract, versioned-experiment, goal-forge planner, groundtruth auditor, recovery-proof backup test, the Loop Hiring Manager.

**SHORTCUT — INSTALL "LOOPY"**

The Loop Library ships an agent skill that finds, crafts, runs, and improves loops for you. Add it once and just ask:

```
npx skills add Forward-Future/loop-library --skill loopy -g
```

## BUILD YOUR OWN LOOP (3 QUESTIONS)

Any task becomes a loop if you can answer these three. If you can't answer #3 yet, you don't have a loop — you have a wish.

### ANY TASK BECOMES A LOOP IF YOU CAN ANSWER 3 THINGS

**1 Trigger?**

A timer, or a goal.

**2 "Done"?**

The more measurable, the better

**3 Proof?**

A test, a file, a verified claim.

Trigger, finish line, proof — get those three and the command writes itself.

**WHAT TO DO NEXT**

Set up the **nightly log** right now — it's the easiest win and you'll feel it tomorrow morning. Then let the **automation scout** find your next four.

**SOURCES & WHERE TO LEARN MORE**

- [Forward Future — Loop Library \(all 70 loops, updated weekly\)](#)
- [Loopy — the Loop Library agent skill \(find, craft, run, improve loops\)](#)
- [New to skills? Comment "SKILLS" for the Claude Skills Playbook](#)

Built by Hank Barker // AI Essentials // @hankhelpsai. Full library sourced from Forward Future's Loop Library — last reviewed July 2026.