

Sean's Setup — How I Use AI to Build Stuff

Hey! This is a walkthrough of every tool I use on my computer. I basically talk to an AI and it builds websites, writes documents, controls my browser, and does pretty much anything I ask. I'm going to explain each piece so you can set it up too. No coding experience needed — I'll explain everything as we go.

First, Some Quick Definitions

Before we start, here are a few words you'll see a lot:

- **Terminal:** That black screen with text you see in hacker movies. It's just a way to talk to your computer by typing commands instead of clicking buttons. On Mac, it's an app called "Terminal." Think of it like texting your computer.
 - **Install:** Downloading something and setting it up so it works on your computer.
 - **Command:** A line of text you type into the terminal to make something happen. Like typing `open` . opens a folder.
 - **Plugin:** An add-on that gives an app extra powers. Like a mod in a game.
 - **Repository (repo):** A project folder that's tracked by Git (explained below) so you can save versions of your work and back it up online.
 - **Git:** A tool that saves snapshots of your project so you can undo mistakes and keep a history of everything you've done. Think of it like Google Docs version history but for your whole project.
 - **GitHub:** A website where your Git projects live online. Like Google Drive but for code.
-

The Big Picture

Here's how my setup works in plain English:

*I **talk out loud** into my computer (using Wispr Flow). My words get converted into text. That text goes into **Claude Code**, which is an AI that lives in my terminal. Claude reads what I said, figures out what I want, and then **builds it** — writes code, edits files, searches the internet, even controls my web browser. When I'm done, I say "save" and everything gets backed up to the internet automatically.*

That's it. I talk, it builds. Now let me show you each piece.

1. Claude Code — The Brain

What it is in simple terms: Claude is an AI (made by a company called Anthropic) that lives in your terminal. You type or talk to it, and it can do almost anything on your computer — write files, build websites, search Google, run programs, and more. It's like having a really smart assistant that can actually touch and change things on your computer, not just chat.

How it's different from ChatGPT: ChatGPT just talks to you in a browser window. Claude Code actually lives ON your computer. It can open your files, edit them, run programs, create folders — it's hands-on, not just a chatbot.

How to get it:

1. You need a Mac (it works on other computers too, but Mac is easiest)
2. You need Node.js installed (I'll explain how below)
3. You need to pay for Claude — either an API key from Anthropic, or a Claude Pro subscription (\$20/month) or Claude Max (\$100-200/month for heavy use)

How to install it: Open your terminal and paste this:

```
npm install -g @anthropic-ai/claude-code
```

Then to start it, just type:

```
c laude
```

And you're talking to the AI. That's it.

2. Ghostty — The Terminal App

What it is in simple terms: Remember how I said the terminal is that text screen where you type commands? Well, your Mac comes with a basic one, but Ghostty is a better version. It's faster, looks cleaner, and works better with Claude Code.

Think of it like this: Your Mac comes with Safari for browsing the web, but a lot of people download Chrome because it's better. Ghostty is like the Chrome of terminals.

How to get it: Go to <https://ghostty.org> and download it. Open it, and you're good.

Do you NEED it? No — you can use the terminal that already comes on your Mac. But Ghostty is nicer and I'd recommend it.

3. Wispr Flow — Talk Instead of Type

What it is in simple terms: An app that lets you talk out loud and it types what you say. Anywhere on your computer. You hold a button, talk, let go, and your words appear as text wherever your cursor is.

Why this is a game-changer: Instead of typing out "Hey Claude, can you build me a website with a blue header and three sections" — I just SAY that out loud. Wispr Flow types it for me. This is how I work so fast. I literally just talk and things get built.

How to get it:

1. Go to <https://wisprflow.com> and download it
2. Install it on your Mac
3. It'll ask for permission to use your microphone and accessibility features — say yes to both
4. Set up a hotkey (a keyboard shortcut to start/stop talking) — pick whatever feels natural
5. Now hold that key, talk, and let go. Your words show up as text.

It works everywhere — in your terminal with Claude, in a Google Doc, in a text message, in Slack, anywhere you can type.

4. Compound Engineering — The Expansion Pack

What it is in simple terms: Remember how in video games you can buy DLC (downloadable content) that adds new weapons, maps, and abilities? Compound Engineering is like a massive DLC pack for Claude Code. It gives Claude 50+ new superpowers.

How to install it: Once you have Claude Code running, type this command:

```
/install-plugin compound-engineering@every-marketplace
```

What superpowers does it add?

The Main Workflows (the ones I use the most)

- **/brainstorm** — When you have an idea but aren't sure how to build it yet. You and Claude talk it out together. It asks you smart questions, suggests approaches, and helps you figure out what you actually want before you start building. Like planning a project with a really smart friend.
- **/plan** — Once you know what you want, this creates a step-by-step blueprint. It figures out what files need to be created, what order to do things in, and what decisions need to be made. Like getting an architect's plan before building a house.
- **/work** — This is the "go build it" command. Claude follows the plan and actually creates the thing — writes the code, creates the files, builds the website, whatever the plan says.
- **/review** — After something is built, this sends in multiple AI "reviewers" to check the work. They look for bugs, security issues, and things that could be better. Like having a team of editors review an essay.

Browser Control

- **agent-browser** — This lets Claude open a real Chrome browser and use it. It can go to websites, click buttons, fill out forms, take screenshots, and read what's on the page. I use this when I need Claude to look at something on the internet, or test a website I'm building.

Design Tools

- **frontend-design** — Claude becomes a web designer. It creates good-looking websites and interfaces, not just functional ones.
- **gemini-imagegen** — Claude can generate images using Google's AI. You describe what you want and it creates it.

Research Tools

- **Context7** — This is built into the plugin. When Claude needs to look up how a coding library works, it can pull the latest official documentation in real time. This means it always has current info, not outdated stuff from its training.

And a lot more...

There are 50+ skills in here covering code review, security checking, file uploading, Ruby/Python/TypeScript expertise, and more. You don't need to memorize them — just know they're there, and Claude will use them when it needs to.

5. Agent-Browser — Claude Controls Chrome

What it is in simple terms: A tool that lets Claude open and control a real web browser. It's like giving Claude hands to use the internet.

Why this is cool: Normally, AI can only "think" and "talk." With agent-browser, Claude can actually GO to websites, read them, click things, fill out forms, and take screenshots. So if I say "go look at this YouTube video and tell me what it's about," Claude literally opens Chrome, goes to the video, and looks at it.

How to install it:

```
npm install -g agent-browser
```

You don't really need to learn the commands — Claude knows how to use it. You just tell Claude what you want ("go to this website and screenshot it") and it handles the rest.

6. CLAUDE.md — Claude's Memory

What it is in simple terms: A text file you put in your project folder that Claude reads every time you start it up. It's like leaving yourself a note that says "here's everything you need to know about this project."

Why it matters: Without this file, Claude starts every conversation with zero knowledge of your project. With it, Claude immediately knows:

- What your project is about
- What you've already built
- Your preferences and rules
- How to deploy/save your work
- What not to do

Think of it like this: Imagine hiring a new assistant every day who has amnesia. Every morning you'd have to re-explain everything. The CLAUDE.md file is like a detailed instruction manual you hand them so they're up to speed immediately.

How to make one: Create a file called `CLAUDE.md` in the main folder of your project. Write whatever context you want Claude to know. There's no special format — just write in plain English.

7. Git Worktrees — Run Multiple Claudes at Once

What it is in simple terms: A way to have Claude working on multiple things at the same time in different windows.

Think of it like this: Imagine you could clone yourself 4 times. One of you works on homework, one cleans your room, one texts your friends, and one watches YouTube — all at the same time. That's what worktrees let me do with Claude. I have 4 terminal windows open, each with its own Claude, each working on something different.

Do you need this right away? No. This is an advanced setup. Start with just one Claude in one terminal window. Once you're comfortable, you can explore worktrees later.

8. My Custom Shortcuts

/save

I made a custom command so I can just type `/save` and all my work gets:

1. Saved to my project
2. Merged with my main version
3. Backed up to GitHub (the internet)

I don't have to think about any git commands. One word and it's done.

Self-Improvement Hook

I set up a little script that makes Claude give itself a tip on how to work more efficiently every time it does a lot of work. So Claude is literally training itself to be better at helping me, in real time.

9. Other Tools Worth Having

These are extra tools I have installed. You don't need all of them right away, but they're useful:

Tool	What It Does (Simple)	How to Get It
Homebrew	A tool that makes installing other tools easy. Like an app store for your terminal.	Paste this in terminal: <code>/bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"</code>
Node.js	Required for Claude Code to work. It's what runs JavaScript programs on your computer.	<code>brew install node</code>
GitHub CLI (gh)	Lets you interact with GitHub from your terminal instead of the website.	<code>brew install gh</code>
yt-dlp	Downloads videos from YouTube and other sites.	<code>brew install yt-dlp</code>
ffmpeg	Edits and converts video/audio files. Claude uses this when I ask it to work with media.	<code>brew install ffmpeg</code>

Python	A programming language. Lots of tools need it to run.	<code>brew install python</code>
---------------	---	----------------------------------

Step-by-Step: Set Everything Up From Zero

If you want to copy my setup, here's the order to do it. Open your terminal and follow along:

Step 1 — Install Homebrew (the tool installer)

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Follow the instructions it gives you. This takes a few minutes.

Step 2 — Install Node.js (needed for Claude)

```
brew install node
```

Step 3 — Install Claude Code

```
npm install -g @anthropic-ai/claude-code
```

Step 4 — Start Claude for the first time

```
claude
```

It'll ask you to log in or enter an API key. Follow the prompts.

Step 5 — Install the Compound Engineering plugin Inside Claude, type:

```
/install-plugin compound-engineering@every-marketplace
```

Step 6 — Install agent-browser

```
npm install -g agent-browser
```

Step 7 — Download Ghostty Go to <https://ghostty.org> and download it.

Step 8 — Download Wispr Flow Go to <https://wisprflow.com> and download it. Set up your hotkey.

Step 9 — You're done. Open Ghostty, type `claude`, and start talking to it. Ask it to build you something. Try: "Build me a simple website about my favorite hobby." Watch what happens.

Tips for Getting Started

1. **You don't need to know how to code.** Claude writes the code for you. Just describe what you want in plain English.

2. **Be specific.** Instead of "make me a website," try "make me a website about basketball with a dark theme, a section about my favorite players, and a contact form."
3. **Ask Claude for help.** If you're confused about ANYTHING, just ask Claude. Type "what does this mean?" or "explain what just happened" and it'll break it down.
4. **Start small.** Don't try to build a whole app on day one. Start with something simple — a personal website, a to-do list, a fun page about something you like.
5. **Use Wispr Flow early.** The sooner you get comfortable talking instead of typing, the faster everything gets. It feels weird at first but after a day it's natural.
6. **Don't be afraid to mess up.** Claude can undo pretty much anything. If something breaks, just tell Claude "that broke, undo it" and it will.

The Bottom Line

This whole setup costs about \$20/month (for Claude Pro) and takes maybe 30 minutes to install. After that, you can build websites, write documents, automate tasks, control your browser, generate images, and do things that would normally take a professional developer hours — just by talking.

Welcome to the future. Have fun with it.

— Sean

Last updated: February 2026