7 Claude Code Tools Blowing Up Right Now

A Complete Setup and Installation Guide

From Zero to Fully Configured

CreativeAIMag.com
March 2026

What Is This Guide?

The Claude Code community has been building incredible tools that supercharge what Claude can do. This guide covers seven of the most popular ones, what they do, why they matter, and exactly how to install each one step by step.

Don't worry if you've never used a terminal before. We'll walk you through everything from the very beginning.

Prerequisites: Installing Claude Code

Before installing any of these tools, you need Claude Code itself. Claude Code is a command-line tool that lets you talk to Claude directly from your terminal. Here's how to get it set up.

What You Need

- **A paid Claude account** (Claude Pro, Team, or Enterprise). The free plan does not include Claude Code.
- **A computer running** Mac, Linux, or Windows (with WSL).
- **An internet connection.**

Step-by-Step: Install Claude Code

On Mac or Linux

1. Open your Terminal app. On Mac, you can find it by pressing Cmd + Space and typing "Terminal".
2. Copy and paste this command, then press Enter:

```
curl -fsSL https://claude.ai/install.sh | bash
```

3. Close your terminal and open a new one (this refreshes your system path).
4. Type `claude` and press Enter. Follow the login prompts to connect your Anthropic account.

On Windows

1. Open PowerShell as Administrator (right-click the Start button, select "Terminal (Admin)").
2. Copy and paste this command, then press Enter:

```
irm https://claude.ai/install.ps1 | iex
```

3. Close PowerShell and reopen it.

4. Type `claude` and press Enter. Log in with your Anthropic account.

> **Note:** Windows users may need WSL (Windows Subsystem for Linux) for some tools. If you run into issues, search "install WSL" in your Start menu and follow the setup.

Once Claude Code is running, you're ready to install the tools below!

How Plugin Installation Works

Installing a Claude Code plugin is simple. You open your terminal, type a command, and it downloads and configures itself. Here are the basic commands you'll use:

```
claude plugin add <plugin-name> Install a plugin

claude plugin list List installed plugins

claude plugin update --all Update all plugins

claude plugin remove <plugin-name> Remove a plugin
```

Now let's install each tool.

Tool 1: Obsidian Skills

What it does: Teaches Claude how to read, search, and write to your Obsidian vault. Turns your notes into an AI-accessible knowledge base.

Created by: kepano (Steph Ango, CEO of Obsidian)

Best for: Anyone who uses Obsidian for notes, journals, or research.

GitHub: github.com/kepano/obsidian-skills

Installation

1. Open your terminal.
2. Run the following command:

```
claude plugin add kepano/obsidian-skills
```

3. Navigate to your Obsidian vault folder:

```
cd /path/to/your/obsidian/vault
```

4. Start Claude Code with `claude` and ask it to search or summarize your notes!

Tool 2: n8n-MCP

What it does: Gives Claude deep knowledge of n8n's 1,236+ automation nodes so you can describe workflows in plain English and Claude builds them.

Best for: Building automations without manually wiring nodes.

GitHub: github.com/czlonkowski/n8n-mcp

Installation

1. Open your terminal.
2. Run the following command:

```
claude plugin add czlonkowski/n8n-mcp
```

3. You'll also need n8n running locally or on a server. Install n8n with:

```
npm install -g n8n
```

4. Start Claude Code and describe the automation you want to build. Claude will generate the n8n workflow for you.

Tool 3: UI/UX Pro Max

What it does: A design intelligence skill with 50+ UI styles, 97 color palettes, 57 font pairings, 99 UX guidelines, and a Design System Generator.

Created by: nextlevelbuilder

Best for: Frontend development and building polished UIs fast.

GitHub: github.com/nextlevelbuilder/ui-ux-pro-max-skill

Installation

1. Open your terminal.
2. Run the following command:

```
claude plugin add nextlevelbuilder/ui-ux-pro-max-skill
```

3. Navigate to your web project folder.
4. Start Claude Code and ask it to generate a design system or suggest a color palette for your project.

Tool 4: Claude Mem

What it does: Gives Claude persistent memory across sessions. Automatically captures, compresses, and reinjects relevant context so Claude remembers your projects.

Created by: Alex Newman (@thedotmack)

GitHub Stars: 20,000+

Best for: Long-running projects where you need Claude to remember previous work.

GitHub: github.com/thedotmack/claude-mem

Installation

1. Open your terminal.
2. Run the following command:

```
claude plugin add thedotmack/claude-mem
```

3. That's it! Claude Mem works automatically in the background. Every session is captured, compressed, and stored locally on your machine.
4. Start a new Claude Code session and it will automatically pull in relevant memories from previous sessions.

Tool 5: GSD (Get Shit Done)

What it does: A spec-driven development system that interviews you, creates a plan, then executes it with parallel AI agents, each getting a fresh context window.

Created by: TACHES

GitHub Stars: 23,000+

Best for: Building complex features with structured planning. Trusted by engineers at Amazon, Google, Shopify, and Webflow.

GitHub: github.com/gsd-build/get-shit-done

Installation

1. Open your terminal.
2. Run the following command:

```
claude plugin add gsd-build/get-shit-done
```

3. Navigate to your project folder.
4. Start Claude Code and type `/gsd` to begin.
5. GSD will interview you about what you want to build, create a spec, and then execute the plan step by step.

Tool 6: Awesome Claude Code

What it is: The master directory of every Claude Code skill, plugin, hook, MCP server, and agent orchestrator the community has built. Not a plugin itself, but your go-to discovery hub.

Maintained by: hesreallyhim

Bookmark this: github.com/hesreallyhim/awesome-claude-code

How to Use It

No installation needed! Just visit the GitHub page above and browse the categories. It's organized into sections like Skills, Plugins, MCP Servers, Hooks, and Applications. When you find something interesting, follow the installation instructions in that specific tool's repository.

Some hidden gems worth exploring: **Book Factory** (writes entire nonfiction books), **Codex Skill** (lets Claude use OpenAI's Codex), and **AgentSys** (multi-agent code review).

Tool 7: Superpowers

What it does: Auto-enforces best practices like TDD, structured debugging, brainstorming gates, and code review. Production code written before a failing test gets deleted.

Created by: Jesse Vincent (obra)

GitHub Stars: 53,000+ (the most popular Claude Code plugin)

Best for: Disciplined, high-quality software development.

GitHub: github.com/obra/superpowers

Installation

1. Open your terminal.
2. Run the following command:

```
claude plugin add obra/superpowers
```

3. That's it! Superpowers activates automatically. Once installed, its skills trigger on their own during your workflow.
4. Start a coding session and you'll notice it enforces TDD, planning gates, and structured debugging without you having to do anything extra.

Quick Reference: All Install Commands

Here are all seven tools and their install commands in one place. Just copy and paste into your terminal.

```
# Obsidian Skills
claude plugin add kepano/obsidian-skills

# n8n-MCP
claude plugin add czlonkowski/n8n-mcp

# UI/UX Pro Max
claude plugin add nextlevelbuilder/ui-ux-pro-max-skill

# Claude Mem
claude plugin add thedotmack/claude-mem

# GSD (Get Shit Done)
claude plugin add gsd-build/get-shit-done

# Superpowers
claude plugin add obra/superpowers
```

Troubleshooting

"claude: command not found"

Close your terminal completely and open a new one. The installation adds Claude Code to your system path, but it only takes effect in new terminal windows.

"Permission denied" errors

On Mac/Linux, try running the command with `sudo` in front (e.g., `sudo claude plugin add ...`). On Windows, make sure you opened PowerShell as Administrator.

Plugin not working after install

Try restarting Claude Code by typing `exit` and then `claude` again. Some plugins need a fresh session to activate.

Need more help?

Each tool's GitHub page has a README with detailed documentation and an Issues section where you can ask questions. The Awesome Claude Code directory is also a great resource for finding help and discovering new tools.