Member-only story

5 MCP Servers Every Python Dev Should Install in 2026 (Claude Desktop Ready)

5 min read · Sep 6, 2025

Tarun Singh Follow

Listen Share More

The top 5 MCP servers to supercharge Python agents in 2026 — terminal, web scraping, knowledge graph memory, GitHub, and a safe Python executor.

5 MCP Servers Every Python Dev Should Install in 2026

Claude Desktop Ready

- Bright Data Web
- Desktop Commander
- Graphiti
- GitIngest
- Code Executor

The top 5 MCP servers to supercharge Python agents in 2026 — terminal, web scraping, knowledge graph memory, GitHub, and a safe Python executor.

TL;DR

If you're building Python-first AI agents, these **MCP servers** (Model Context Protocol) unlock safe access to your filesystem, terminal, the public web, GitHub repos, and executable Python sandboxes — directly from clients like **Claude Desktop** or **Cursor**. MCP is the open standard that gives your AI tools and context securely. The VergeModel Context Protocol

. . .

Why MCP matters in 2026

MCP standardizes how LLM apps speak to tools and data. Instead of bespoke plugins, you add servers once and every MCP-capable client can call them — ideal for **agentic workflows** where Python services, repos, or web data are in play. The ecosystem has exploded since late-2024, with official examples and a growing gallery of community servers. The VergeGitHubModel Context Protocol

. . .

Quick start (Claude Desktop config)

Drop servers into your Claude Desktop config so your agent can call them:

```
{
  "mcpServers": {
    "desktop-commander": {
      "command": "npx",
      "args": ["desktop-commander-mcp", "start"]
    },
    "brightdata-web": {
      "command": "npx",
      "args": ["@brightdata/mcp", "start"]
    },
    "graphiti": {
      "command": "python",
      "args": ["-m", "graphiti.mcp_server"]
    },
    "gitingest": {
      "command": "npx",
      "args": ["gitingest-mcp", "start"]
    },
    "code-executor": {
      "command": "python",
      "args": ["-m", "mcp_code_executor", "--conda-env", "py310"]
    }
  }
}
```

> Tip: keep each server in its own virtual env or TS project for easy upgrades.

. . .

1) Desktop Commander MCP — “Your terminal + files, on tap”

When you want your AI to **search files**, **edit code**, and **run commands** without wrestling with shell history, Desktop Commander is the reliable pick. It builds on the Filesystem server and adds long-running command support (great for migrations, tests, and scripts). GitHubDesktop Commander

Sample prompts

```
> Find all Python files mentioning "OpenAI" under /manon
> Run: pytest -q
> Replace "TODO:" with "NOTE:" in *.py under /scripts
```

Best for: project bootstraps, bulk refactors, CI-like runs locally.

. . .

2) Bright Data Web MCP — “Block-aware web scraping for agents”

Need pricing pages, product listings, or SERP-like data inside your flow? Bright Data’s MCP server exposes a suite of **web access tools** (crawl, extract, rotate, bypass anti-bot controls) behind an MCP interface, with a free tier to test. Perfect for market checks and dataset bootstraps. GitHubdocs.brightdata.com

Prompt sketch

```
> Scrape /search?q=site:docs.python.org "pydantic" and return title, url
> Crawl https://example.com/shoes and extract name, price, in_stock
```

Best for: competitive intel, dataset generation for fine-tuning, price trackers.

. . .

3) Graphiti MCP — "Persistent memory via a knowledge graph"

Most agents forget. Graphiti gives you **entities + relationships** (who likes what, when did it happen) with semantic & hybrid search over a real knowledge graph — so your assistant can recall user preferences or project facts later. GitHubZep Documentation

Prompt sketch

```
> Add: "Alex" -[LIKES]-> "Dark Mode" (source: settings.json)
> What does Alex prefer for theme?
> Show recent relations about "billing pipeline"
```

Best for: user preference memory, incident timelines, project knowledge.

. . .

4) GitIngest MCP — "Skim any GitHub repo without cloning"

Point your agent at a repo and pull a **directory tree**, **key files**, and quick **summaries**. It's ideal for "is this the right lib?" checks or fast audits of unfamiliar codebases. Multiple implementations exist; the popular ones expose `directory_structure` and `read_important_files`. GitHub+1Glama – MCP Hosting Platform

Prompt sketch

```
> Summarize https://github.com/owner/project (focus: API client code)
> Show: docs/architecture.md and src/config.py
```

Best for: code discovery, due diligence, dependency reviews.

· · ·

5) Code Executor MCP — "Run real Python in a Conda sandbox"

Sometimes you need to **execute** the code the model proposes. This server runs Python in your chosen **Conda environment**, so your agent can import NumPy/Pandas, test snippets, and return outputs — without leaving chat. Great for data wrangling, quick metrics, or verifying algorithm changes. GitHubMCP MarketPlaybooks

Example task

```
# Prompt: "Run this quickly in 'py310':"
import numpy as np, pandas as pd
arr = np.array([5, 10, 15])
df = pd.DataFrame({"x": arr, "x2": arr**2})
print(df.describe().to_string())
```

Best for: prototyping transforms, validating PR diffs, micro-benchmarks.

· · ·

How these five fit together (a dev workflow)

1. **GitIngest** → scan a repo's layout & important files. GitHub
2. **Desktop Commander** → create a branch, run tests, and edit code locally. GitHub
3. **Code Executor** → verify the patch with small Python experiments. GitHub
4. **Bright Data Web** → fetch fresh web data for the feature. GitHub
5. **Graphiti** → persist decisions & preferences as a graph for future runs. GitHub

. . .

Install notes & safety rails

- Prefer **least-privilege**: point filesystem/terminal access to project-scoped folders.
- Gate any network scraping behind an allowlist and respect robots/ToS.
- Keep server versions pinned in `requirements.txt` or `package.json`.
- Track the evolving ecosystem via the official server directory & examples. GitHubModel Context Protocol

. . .

One-screen "Try it now" checklist

- Install **Claude Desktop** (or your MCP client of choice).
- Add the five servers in your config (snippet above).
- Test prompts from each section and commit the config to your dotfiles.
- Log wins: build time saved, bugs avoided, data captured — then scale.

. . .

References & further exploration

- Bright Data MCP server & overview pages (web data access, free tier). GitHubdocs.brightdata.com
- Graphiti MCP server (knowledge graph memory). GitHubZep Documentation
- GitIngest MCP (directory structure + important files). GitHub+1
- Desktop Commander MCP (terminal + files with long-running commands). GitHubDesktop Commander
- Code Executor MCP (Conda-based Python execution). GitHubMCP Market

- MCP ecosystem lists & official example servers. GitHub+1Model Context Protocol
- Background on MCP's launch & adoption. The Verge

Let's Connect + Fuel the Next Deep Dive ☕

I build at the intersection of **AI, code, and automation** — and I share hands-on guides from LLMs to real-world apps. If this helped, stay in touch (and consider tipping a coffee to keep these coming):

- 📧 **Email:** krtarunsingh@gmail.com
- 💼 **LinkedIn:** Tarun Singh
- 💻 **GitHub:** github.com/krtarunsingh
- ☕ **Buy Me a Coffee:** https://buymeacoffee.com/krtarunsingh

👉 If you found value here, **clap, share, and leave a comment** — it helps more devs discover practical guides like this.

A message from our Founder

Hey, Sunil here. I wanted to take a moment to thank you for reading until the end and for being a part of this community.

Did you know that our team run these publications as a volunteer effort to over 3.5m monthly readers? **We don't receive any funding, we do this to support the community.** ❤️

If you want to show some love, please take a moment to **follow me on LinkedIn, TikTok, Instagram**. You can also subscribe to our **weekly newsletter**.

And before you go, don't forget to **clap** and **follow** the writer!

AI Mcps Mcp Server Coding Software Development

Follow

Published in Artificial Intelligence in Plain English

33K followers · Last published 8 hours ago

New AI, ML and Data Science articles every day. Follow to join our 3.5M+ monthly readers.

Follow

Written by Tarun Singh

1.3K followers · 6 following

AI & ML Engineer | LLMs, RAG, Automation, OpenAI — sharing systems for money, focus & growth while building production-ready AI in Python

No responses yet

Bgerby

What are your thoughts?

More from Tarun Singh and Artificial Intelligence in Plain English

In Towards AI by Tarun Singh

15 RAG Chunking Techniques Every AI Engineer Should Know

Level Up Your RAG Apps: 15 Easy Chunking Strategies (with Examples!)

✦ Jul 17 307 6

In Artificial Intelligence in Plain English by Sandra Bats

OpenAI is coming for your org*sms... and we should be worried

I thought it was a joke. I really did. But it’s official. ChatGPT is going to get a ‘pervert mode’. CEO Sam Altman announced yesterday that...

Oct 23 1.5K 92

In Artificial Intelligence in Plain English by Alpha Iterations

Build Agentic RAG using LangGraph

A practical guide to build Agentic RAG with complete project code

Oct 19 277 1

Tarun Singh

The Next AI Boom: What Comes After AI Agents and Agentic AI?

Artificial Intelligence is no longer science fiction. It’s a living, breathing force that’s already transforming how we work, live, and...

May 24 243 18

See all from Tarun Singh

See all from Artificial Intelligence in Plain English

Recommended from Medium

In Level Up Coding by Swathi Ganesh

13 Rare Python Packages That Make AI Feel Like Magic

Rare yet the best python packages to discover

Oct 30 200 2

In AI Software Engineer by Joe Njenga

Anthropic Just Solved AI Agent Bloat — 150K Tokens Down to 2K (Code Execution With MCP)

Anthropic just released smartest way to build scalable AI agents, cutting token use by 98%, shift from tool calling to MCP code execution

✦ 2d ago 👏 285 💬 18

The Atomic Architect

You Haven’t Seen AI Until You Try Claude Sonnet 4.5’s New Feature — It Redefines Insane

I built a working expense tracker in eight minutes while my coffee was still hot, and it remembered every receipt when I closed my laptop…

Oct 26 325 19

In Coding Nexus by Code Coup

Claude Desktop Might Be the Most Useful Free Tool You’ll Install This Year

I didn’t expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.

Oct 23 953 37

In Python in Plain English by Mahad Nadeem

10 Python Tools That Made My Code Look Like It Was Written by a Senior Engineer

The best-kept secrets behind clean, automated, production-ready Python from someone who learned it the hard way.

✦ Nov 2 👏 105 💬 3

In Towards AI by Teja Kusireddy

We Spent $47,000 Running AI Agents in Production. Here’s What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic’s Model Context Protocol (MCP) are revolutionary. But...

Oct 16 👏 3.3K 💬 111

See more recommendations