# Cheat sheet: the tools of your Brain

Your AI calls these tools by itself. You can also ask for one by name, for example: “Use smart_recall and look for the deploy.”

| Tool | What for | Example call |
|---|---|---|
| **session_start** | Briefing at the start of a session: last state, new lessons. | `session_start(focus="login bug")` |
| **smart_recall** | Find knowledge for a task (by meaning, not exact words). | `smart_recall(query="fix login redirect loop")` |
| **remember_context** | Remember one fact under a name. | `remember_context(key="auth_architecture", content="Sessions run through Keycloak")` |
| **recall_context** | Fetch a remembered fact by its name. | `recall_context(key="auth_architecture")` |
| **learn_from_attempts** | After a fix, record what worked and what did not. | `learn_from_attempts(topic="deploy:api", outcome="success", what_worked="...")` |
| **causal_trace** | On an error: look up known causes and fixes. | `causal_trace(problem="login breaks after restart")` |
| **recall_best_solution** | Get the best known solution for a topic. | `recall_best_solution(topic="deploy:api")` |
| **brain_predict** | Before a risky step: what went wrong there before. | `brain_predict(context="upgrading Keycloak from 21 to 24")` |
| **brain_from_git** | Fill the Brain from your git history. | `brain_from_git(repo_path=".", limit=100)` |
| **brain_doctor** | Checks key, Brain and connection. | `brain_doctor()` |
| **forget_context** | Delete remembered entries (names, globs with * work). | `forget_context(keys=["wip:*"])` |
| **session_end** | At the end of a session, sum up what was done. | `session_end(summary="Fixed the login bug", files_changed=["src/auth.ts"])` |

## Set up in one line

```
Claude Code:  /plugin marketplace add cachly-dev/cachly-mcp
              /plugin install cachly-brain@cachly
Others:       npx @cachly-dev/mcp-server@latest autopilot
Check:        npx @cachly-dev/mcp-server@latest doctor
```