OpenCode Cheatsheet

AI-Assisted Programming for PhD Researchers

OpenCode evolves fast, and this sheet matches the version pinned for the course (see Setup). If a command below doesn't behave as described, run `opencode --help` *and check the official docs.*

Install / update

Command	What it does
`curl -fsSL https://opencode.ai/install | bash`	Installs OpenCode (macOS/Linux)
`brew install anomalyco/tap/opencode`	Installs OpenCode via Homebrew (macOS)
`opencode upgrade`	Updates OpenCode to the latest version (verify with `opencode --help` if this changed)

Start

Command	What it does
`opencode`	Launches the terminal UI in the current project (run from the project root)
`opencode auth login`	Stores an API key for a model provider (e.g. Mistral); in some versions this is the in-session `/connect` command — check `opencode --help`

In-session

Key / command	What it does
`Tab`	Toggles between **Plan** mode (read-only, suggests changes) and **Build** mode (executes changes)
`/undo`	Reverts the last change OpenCode made
`/redo`	Re-applies a change you just undid

Key / command	What it does
`/compact`	Compacts the conversation history to free up context
`/init`	Scans the repo and generates/updates `AGENTS.md` with project context
`@file`	References a specific file in your message (e.g. `@src/main.py`)
`!command`	Runs a shell command directly from the chat

> **i** Note
>
> `/init`, `/undo`, `/redo`, and `/compact` are built-in commands but can be overridden by a same-named file in `.opencode/commands/`. Check `opencode --help` if a command listed here has changed.

Config (`opencode.json`)

Project-level configuration lives in `opencode.json` at the repository root. A minimal example selecting a model and adding Mistral as a provider:

```
{
  "$schema": "https://opencode.ai/config.json",
  "model": "mistral/mistral-medium-latest",
  "provider": {
    "mistral": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "Mistral",
      "options": {
        "baseURL": "https://api.mistral.ai/v1",
        "apiKey": "{env:MISTRAL_API_KEY}"
      },
      "models": {
        "mistral-medium-latest": { "name": "Mistral Medium 3.5" }
      }
    }
  }
}
```

> **Note**
>
> OpenCode may add a native Mistral provider (selectable via `/connect`) after this snapshot was taken; check `opencode --help` / the providers docs if a native block is available instead of the `openai-compatible` shim above.

A `permission` block controls which actions run automatically vs. need your approval (`"allow"` / `"ask"` / `"deny"`, last matching rule wins):

```
{
  "permission": {
    "bash": { "*": "ask", "git *": "allow", "rm *": "deny" },
    "edit": "ask"
  }
}
```

MCP (`opencode.json`)

MCP (Model Context Protocol) servers add external tools. A server is either `remote` (a hosted URL) or `local` (a command OpenCode launches). Lab 5 wires GitHub's hosted server:

```
{
  "mcp": {
    "github": {
      "type": "remote",
      "url": "https://api.githubcopilot.com/mcp/",
      "enabled": true
    }
  }
}
```

A **local** server takes a launch command instead: `"type": "local", "command": ["npx", "-y", "<package>"]`, with an optional `"environment": { "API_KEY": "{env:VAR}" }`. Authenticate a server that needs it with `opencode mcp auth <name>`.

Command	What it does
`opencode mcp list`	Lists configured MCP servers
`opencode mcp auth <server-name>`	Authenticates with an MCP server
`opencode mcp logout <server-name>`	Removes stored credentials for an MCP server

Skills / commands

OpenCode discovers **Claude Code-compatible skills** directly, no conversion needed. It looks for `SKILL.md` files at:

- `.claude/skills/<name>/SKILL.md` (project) or `~/.claude/skills/<name>/SKILL.md` (global)
- `.opencode/skills/<name>/SKILL.md` (project, native) or `~/.config/opencode/skills/<name>/SKILL.md` (global, native)

To install a skill, copy its folder into `.claude/skills/` in your project. The course ships writing skills under `skills/writing/`, used in Lab 5; copy that folder to `.claude/skills/` to make it available in OpenCode.

Custom slash commands work the same way: markdown files with YAML frontmatter under `.opencode/commands/` (filename becomes the command name, e.g. `test.md` → `/test`).

> ! Important
>
> Skill and command discovery paths are version-sensitive. Verify against the current OpenCode skills docs if this sheet feels out of date.

Bibliography