

**Templates**

Default AGENTS.md

First run (recommended)

OpenClaw uses a dedicated workspace directory for the agent. Default:
~/openclaw/workspace (configurable via agents.defaults.workspace).

1. Create the workspace (if it doesn't already exist):

```
mkdir -p ~/openclaw/workspace
```

2. Copy the default workspace templates into the workspace:

```
cp docs/reference/templates/AGENTS.md ~/openclaw/workspace/AGENTS.md  
cp docs/reference/templates/SOUL.md ~/openclaw/workspace/SOUL.md  
cp docs/reference/templates/TOOLS.md ~/openclaw/workspace/TOOLS.md
```

3. Optional: if you want the personal assistant skill roster, replace AGENTS.md with this file:

```
cp docs/reference/AGENTS.default.md ~/openclaw/workspace/AGENTS.md
```

4. Optional: choose a different workspace by setting agents.defaults.workspace (supports ~):



```
agents: { defaults: { workspace: "~/.openclaw/workspace" } },  
}
```

>

Safety defaults

Don't dump directories or secrets into chat.

Don't run destructive commands unless explicitly asked.

Don't send partial/streaming replies to external messaging surfaces (only final replies).

Session start (required)

Read `SOUL.md`, `USER.md`, `memory.md`, and `today+yesterday` in `memory/`.

Do it before responding.

Soul (required)

`SOUL.md` defines identity, tone, and boundaries. Keep it current.

If you change `SOUL.md`, tell the user.

You are a fresh instance each session; continuity lives in these files.

Shared spaces (recommended)

You're not the user's voice; be careful in group chats or public channels.

Don't share private data, contact info, or internal notes.

Memory system (recommended)



Daily log: `memory/YYYY-MM-DD.md` (create `memory/` if needed).

Long-term memory: `memory.md` for durable facts, preferences, and decisions.

>

On session start, read today + yesterday + `memory.md` if present.

Capture: decisions, preferences, constraints, open loops.

Avoid secrets unless explicitly requested.

Tools & skills

Tools live in skills; follow each skill's `SKILL.md` when you need it.

Keep environment-specific notes in `TOOLS.md` (Notes for Skills).

Backup tip (recommended)

If you treat this workspace as Clawd's "memory", make it a git repo (ideally private) so `AGENTS.md` and your memory files are backed up.

```
cd ~/.openclaw/workspace
git init
git add AGENTS.md
git commit -m "Add Clawd workspace"
# Optional: add a private remote + push
```

What OpenClaw Does

Runs WhatsApp gateway + Pi coding agent so the assistant can read/write chats, fetch context, and run skills via the host Mac.

macOS app manages permissions (screen recording, notifications, microphone) and exposes the `openclaw` CLI via its bundled binary.



Direct chats collapse into the agent's main session by default; groups stay isolated as `agent:<agentId>:<channel>:group:<id>` (rooms/channels: `agent:<agentId>:<channel>:channel:<id>`); heartbeats keep background tasks alive.

Core Skills (enable in Settings → Skills)

mcporter – Tool server runtime/CLI for managing external skill backends.

Peekaboo – Fast macOS screenshots with optional AI vision analysis.

camsnap – Capture frames, clips, or motion alerts from RTSP/ONVIF security cams.

oracle – OpenAI-ready agent CLI with session replay and browser control.

eightctl – Control your sleep, from the terminal.

imsg – Send, read, stream iMessage & SMS.

wacli – WhatsApp CLI: sync, search, send.

discord – Discord actions: react, stickers, polls. Use `user:<id>` or `channel:<id>` targets (bare numeric ids are ambiguous).

gog – Google Suite CLI: Gmail, Calendar, Drive, Contacts.

spotify-player – Terminal Spotify client to search/queue/control playback.

sag – ElevenLabs speech with mac-style say UX; streams to speakers by default.

Sonos CLI – Control Sonos speakers (discover/status/playback/volume/grouping) from scripts.

blucli – Play, group, and automate BluOS players from scripts.

OpenHue CLI – Philips Hue lighting control for scenes and automations.



OpenAI Whisper – Local speech-to-text for quick dictation and voicemail transcripts.

Gemini CLI – Google Gemini models from the terminal for fast Q&A.

>

agent-tools – Utility toolkit for automations and helper scripts.

Usage Notes

Prefer the `openclaw` CLI for scripting; mac app handles permissions.

Run installs from the Skills tab; it hides the button if a binary is already present.

Keep heartbeats enabled so the assistant can schedule reminders, monitor inboxes, and trigger camera captures.

Canvas UI runs full-screen with native overlays. Avoid placing critical controls in the top-left/top-right/bottom edges; add explicit gutters in the layout and don't rely on safe-area insets.

For browser-driven verification, use `openclaw browser` (tabs/status/screenshot) with the OpenClaw-managed Chrome profile.

For DOM inspection, use `openclaw browser eval|query|dom|snapshot` (and `--json` / `--out` when you need machine output).

For interactions, use `openclaw browser click|type|hover|drag|select|upload|press|wait|navigate|back|evaluate|run` (click/type require snapshot refs; use `evaluate` for CSS selectors).

[Device Model Database](#)

[AGENTS.md Template](#) >

Powered by [mintlify](#)