AI TOOLS WIKI · MÉMO

MemPalace *mémo*

Les commandes à connaître pour **MemPalace**, la mémoire locale pour agents IA. Elle stocke tes conversations et documents **mot pour mot** (sans résumé) et les retrouve par recherche sémantique. Le contenu est rangé en « ailes » (personnes, projets), « pièces » (sujets) et « tiroirs » (contenu original).

Attention aux faux sites : les seules sources officielles sont le dépôt GitHub, le paquet PyPI `mempalace` et mempalaceofficial.com. D'autres domaines ont diffusé des logiciels malveillants.

Famille Mémoire et contexte · **Éditeur** MemPalace · **Licence** MIT
Note du wiki 8/15 · **Commandes vérifiées le** 25 septembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/memoire-et-contexte/mempalace

Installation

`npx skills add MemPalace/mempalace`	Recommandé : installer les skills, puis demander à l'agent de configurer MemPalace
Ou directement, en environnement isolé	
`uv tool install mempalace`	ou : pipx install mempalace
`mempalace init ~/projects/monapp`	
`mempalace mcp`	affiche la configuration MCP à ajouter à ton client

Trois skills : `mempalace` (installation et opérations), `mempalace-recall` (chercher dans la mémoire avant de répondre), `mempalace-task` (délégation entre agents).

Commandes principales

`mempalace init <dossier>`	Analyse un projet (personnes, projets, sujets) et crée le « palais ».

Commandes principales (suite)

`mempalace mine <dossier>`	Charge des fichiers de projet dans la mémoire.
`mempalace mine ~/.claude/projects/ --mode convos --wing monapp`	Importe les sessions Claude Code d'un projet.
`mempalace search "pourquoi est-on passé à GraphQL" --wing monapp`	Recherche sémantique, limitable à une aile (`--wing`) ou une pièce (`--room`).
`mempalace wake-up`	Contexte d'orientation compact (~600 à 900 tokens) pour démarrer une session.
`mempalace sweep <dossier-transcripts>`	Stocke chaque message séparément pour une recherche plus fine (reprend là où il s'était arrêté).
`mempalace status`	Ce qui a été rangé : nombre de tiroirs, répartition par aile et pièce.
`mempalace compress --dry-run`	Prévisualise la compression des tiroirs.
`mempalace repair`	Reconstruit l'index vectoriel après une corruption ou un changement de modèle d'embedding.
`mempalace hook run --hook SESSION --harness claude`	Hooks de sauvegarde automatique (aussi `STOP`, `PRECOMPACT`, et `--harness codex`).

Outils MCP utilisés par l'agent

Outil	Quand l'agent l'utilise
`mempalace_status`	Au démarrage, pour charger le palais.

Outils MCP utilisés par l'agent (suite)

Outil	Quand l'agent l'utilise
`mempalace_search`	Avant de répondre à une question sur le travail passé.
`mempalace_add_drawer`, `mempalace_kg_add`	Quand tu lui demandes de retenir quelque chose.
`mempalace_diary_write`	En fin de session importante, pour écrire son journal.
`mempalace_list_agents`	Pour découvrir les agents spécialisés et leur aile.

Conseils d'utilisation

- **Brancher les hooks de sauvegarde automatique en premier** : Claude Code supprime les transcriptions de sessions après 30 jours par défaut. Sans hook, cet historique est perdu.
- **Récupérer l'historique existant tout de suite** : sauvegarder les fichiers JSONL de `~/.claude/projects/`, puis lancer `mempalace mine ~/.claude/projects/ --mode convos`.
- **Une aile par projet** (`--wing`) : les recherches restent ciblées et plus pertinentes.
- **Choisir le modèle d'embedding** à l'installation : `embeddinggemma-300m` pour le multilingue (dont le français), `all-MiniLM` pour l'anglais seul, plus léger.
- **Tout reste en local** par défaut (ChromaDB). Rien ne sort de la machine sans ton accord.
- **Premier appel lent** : le modèle d'embedding (80 à 300 Mo) est téléchargé une fois.

Sources (documentation officielle)

- README MemPalace
- Référence CLI
- Configuration de la rétention Claude Code
- Discussion #1388 – sessions qui expirent après 30 jours