

warning "Backend update (October 2026)" GitHub Models inference was retired on July 30, 2026. Any references below are retained only for historical context while this lab is migrated. Use Lab 013 ? Free LLM backends to configure Foundry Local or Ollama before continuing.

Lab 027: Agent Memory Patterns

Level: L200 Path: Pro Code / SK Time: ~35 min ? Cost: Free (local)

What You'll Learn

The 4 types of agent memory and when to use each In-context memory (conversation history management) External memory (vector store + SQLite for structured data) Episodic memory (what happened in past sessions) Building a memory-aware agent that remembers user preferences

Introduction

Without memory, every agent conversation starts from zero. A customer service agent that forgets the customer's name mid-conversation, or an assistant that can't recall what you decided last week, creates a frustrating experience.

The four memory types:

Type	Storage	Lifetime	Example	In-context	LLM context window	Single request
Conversation history	External semantic	Vector DB	Persistent	Past support tickets		
External structured	SQL DB	Persistent	User preferences, order history	Episodic		
Key-value / file	Session or persistent	"Last session we discussed X"				

Prerequisites

Python 3.11+ pip install openai chromadb GITHUB_TOKEN set

tip "Sample data included" This lab uses orders.csv and products.csv from the repo.

[diagram or code sample]

Lab Exercise

Step 1: In-context memory (conversation history)

The simplest memory ? just keep the message history. The challenge: context windows have limits.

[diagram or code sample]

Step 2: Structured memory with SQLite

For facts that need precision (preferences, orders, settings), use a database, not the context window.

[diagram or code sample]

Step 3: Semantic episodic memory with ChromaDB

ChromaDB is a free, local vector database ? no Docker needed.

[diagram or code sample]

Step 4: Memory-aware agent

Combine all three memory types into one agent:

[diagram or code sample]

Memory Pattern Decision Guide

[diagram or code sample]

Next Steps

Agentic RAG uses retrieval memory: ? Lab 026 ? Agentic RAG SK has built-in memory abstractions: ? Lab 023 ? SK Plugins & Memory

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

Download generated PDF Download generated ZIP

Supporting file: CHEATSHEET.md

Supporting files: Lab 027: Agent Memory Patterns

Start with the documented local or free path. Record assumptions and validation evidence.

Keep hosted services optional unless the lab marks them required.

Supporting file: README.md

Lab 027: Agent Memory Patterns

Use these files with the lab guide and generated kit.