AI Agent: Tools, prompts, and the decision-making process.

Basic LLM Chain is like asking someone a question and getting an answer. AI Agent is like hiring a research assistant – you describe the task, and they figure out the steps.

In Lesson 3, you built an email classification tool using Basic LLM Chain —a single prompt input, a single response output. But what about tasks where AI needs to decide what to do, not just answer a question? That's where AI Agents come in . And they change everything about what your workflow can do.

What makes an agent?

Basic LLM Chain is like asking someone a question and getting an answer. AI Agent is like hiring a research assistant – you describe the task, and they figure out the steps.

Here's what AI Agent nodes do that Basic LLM Chains can't:

1. Receive a task (from the system prompt + user input)
2. Reasoning for which tool to use (the "Re" in ReAct)
3. This is done by calling a tool (the word "Act" in ReAct).
4. Observe the tool's output.
5. Loop - deciding whether to call another tool or return the final answer.

This loop is called the ReAct (Reason + Act) pattern. The system can search the web, read the results, decide if more context is needed, search Wikipedia, combine the findings, and then write a summary—all from a single input.

From n8n version v1.82.0 onwards, all system types (OpenAI Functions, Plan-and-Execute, Conversational, etc.) are merged under the Tools Agent. You don't need to select a system type – just connect your tools, configure the system prompt, and the Tools Agent will handle the routing.

3 pillars of an n8n agent

Each AI Agent requires 3 elements to be configured:

1. **LLM Provider** (Brain): Attach a child node to an OpenAI, Anthropic, Google, or Groq conversational model. This is the model that performs inference and decision-making. For systems, use a capable model – gpt-4o or

claude-3.5-sonnet are good default choices. Smaller models often fail when using multi-step tools.

2. **Tools** (Hands): Tools are child nodes that provide the system with various capabilities. Without tools, an agent is just an expensive chat node. n8n includes:

Tools	Its function
SerpAPI	Web search (Google search results)
Wikipedia	Look up factual information on Wikipedia.
Code Tool	Write and run JavaScript or Python code.
HTTP Request Tool	Call any API
Calculator	Perform the calculation.
Workflow Tool	Calling another n8n workflow a tool
MCP Client Tool	Call any MCP server (Lesson 6)

3. **System Prompt** (Instructions): The system prompt tells the agent who it is, what tools it has, and how to use them. This is your most powerful control lever – a good system prompt will make the difference between a confused agent and a reliable one.

? **Quick test** : You connect 3 tools to an AI agent but don't write a system prompt. What will happen?

Answer : The agent will still operate, but it will use tools inconsistently. Without guidance, the agent will make its own decisions about when and how to use each tool – this often means it chooses the first tool that seems appropriate and ignores the others. System prompts allow you to control tool selection.

Build: Multi-tool research agent

You will build an agent to solve the research question, search for recent data on the web, check Wikipedia for context, and write a comprehensive summary.

Step 1: Create the foundation

1. New workflow ? Add **Chat Trigger** (this action creates a chat interface for testing)
2. **Add AI agent** node
3. Connect Chat Trigger with AI agent

Step 2: Attach LLM

Click on the AI ??agent node ? in the **Model** section , add **the OpenAI Chat Model** child node :

1. Verification information: Your OpenAI key
2. Model: gpt-4o (agents require strong reasoning capabilities - gpt-4o-mini may struggle with multi-tool tasks)

Step 3: Connect the tools

Still within the AI ??agent node, add 3 more tools:

Tool 1 : SerpAPI (web search) - Add authentication information: register at serpapi.com for a free plan (100 searches/month) - Now agents can search for information in real time on Google.

Tool 2 : Wikipedia - No authentication required - this tool queries Wikipedia's public API - Good for factual definitions, historical context, and background information.

Tool 3 : Code Tool - Language: JavaScript - No authentication required - this tool runs code in the n8n sandbox - the agent can write and execute code to calculate, process data, or format

Step 4: Write the system prompt.

This is the important part. In the AI ??agent configuration, find the system prompt field and write:

```
B?n là tr? lý nghiên c?u. Khi ???c giao m?t câu h?i: 1. LUÔN LUÔN tìm ki?
m trên web tr??c b?ng SerpAPI ?? tìm d? li?u g?n ?ây, c?p nh?t 2. S? d?
ng Wikipedia ?? tìm thông tin n?n t?ng, ??nh ngh?a và b?i c?nh l?ch s? 3. S
? d?ng Code Tool khi b?n c?n tính toán s? li?u, x? lý d? li?u ho?c ??nh d?
ng k?t qu? 4. T?ng h?p các phát hi?n t? nhi?u ngu?n thành m?t b?n tóm t?
t rõ ràng, có trích d?n Quy t?c: - Trích d?n ngu?n c?a b?n (URL web ho?
c "Wikipedia: Tên bài vi?t") - N?u k?t qu? tìm ki?
m trên web và Wikipedia mâu thu?n, hãy ?u tiên ngu?n g?n ?ây h?n - N?u b?
n không th? tìm th?y thông tin ?áng tin c?y, hãy nói rõ ?i?u ?ó - ??
ng bao gi? b?a ??t thông tin - Gi? b?n tóm t?t cu?i cùng c?a b?n d??i 300 t
?
```

Pay attention to the specificity of this. You are instructing the agent on when to use each tool, how to handle conflicts, and what output format to use. Vague instructions will create vague agents.

Step 5: Check

Click on "Test workflow" and use the chat interface. Try the following questions:

1. "What is NVIDIA's current market capitalization and how has it changed since 2023?"
2. "Compare the populations of Tokyo and New York, including their metropolitan areas."
3. "What is Retrieval Augmentation Generation (RAG) and when was this concept first introduced?"

Observe the agent's reasoning process in the results table. You'll see it decide which tool to call, process the results, and decide whether or not to make another call.

? **Quick check** : Your agent searched the web for the phrase "NVIDIA market capitalization," but the results are outdated. How can you improve this?

Answer : Add a date constraint to your system prompt: "When searching for financial data, include the current year in your search query." You can also add "Always include 2026 in search queries for time-sensitive data" to the system prompt. Then the agent will search for "NVIDIA market capitalization in 2026" instead of the regular query.

Techniques for creating system prompts for agents.

Writing system prompts for agents is a different skill than writing prompts for basic LLM sequences. With sequences, you control the exact input. With agents, you control the strategy—the agent decides the specific details.

3 effective models:

Template 1: Tool Selection Rules

Clearly inform the agent when to use each tool:

```
S? d?ng SerpAPI cho: s? ki?n hi?n t?i, giá c?, th?ng kê, tin t?c g?n ?ây S?
 d?ng Wikipedia cho: ??nh ngh?a, l?ch s?, khái ni?m khoa h?c, d? li?u ti?
u s? S? d?ng Code cho: tính toán, ??nh d?ng d? li?u, chuy?n ??i ??n v?
```

Template 2: Step-by-step strategy

Provide employees with a clear workflow:

```
??i v?i m?i câu h?i: 1. Tìm ki?m d? li?u hi?n t?i trên m?ng 2. Ki?
m tra Wikipedia ?? hi?u ng? c?nh 3. ??i chi?u c? hai ngu?n 4. Vi?t tóm t?
t có trích d?n
```

Template 3: Output Format Specification

Determine the exact format of the feedback you want to receive:

```
Hãy ??nh d?ng câu tr? l?i c?a b?n nh? sau: ## Tóm t?t [T?ng quan 2-3 ?o?
n] ## Các ?i?m d? li?u chính - [các g?ch ??u dòng v?i s? li?u c? th?
] ## Ngu?n tham kh?o - [danh sách các URL và bài vi?t Wikipedia ?ã s? d?ng]
```

Agents that provide clear instructions on what to do, when to use each tool, and how to format the output are far more reliable than agents that only offer vague prompts like "please help."

4 agent architecture models

As workflows become more complex, you will encounter the following four patterns:

Sample	How it works	When should it be used?
Serial request	LLM calls are sequential — the output of one instruction becomes the input of the next instruction.	Multi-step processing procedure (classification ? extraction ? summarization)
Single Agent	An agent + tool + inference loop	Most tasks (research, questioning, data processing)
Gatekeeper + Expert	A coordinating agent will delegate tasks to specialized agents.	Complex tasks with separate subtasks.

Sample	How it works	When should it be used?
Multi-Agent Team	Multiple agents collaborate in a network.	Advanced automation process for coordinating and managing multiple tasks (Lesson 8)

This course will cover working with individual agents (Lessons 4-6) and will mention the gatekeeper pattern in the final lesson (Lesson 8). Let's start simple – most practical workflows only require a single well-configured agent.

Key points to remember

1. AI agents use the ReAct loop – inferring the task, calling a tool, observing the results, and deciding on the next step.
2. Each agent needs three things: an LLM provider (the brain), tools (the hands), and system prompts (instructions).
3. The system prompt is your primary control lever – be specific about when to use each tool and how to format the output.
4. From version 1.82.0 onwards, all agent types are unified under Tools Agent - there is no need to choose between framework agents.
5. Use models that are capable of inferring agents (gpt-4o, claude-3.5-sonnet) - smaller models often fail in multi-step inference.

1. Question 1:

   An AI agent with 6 connected tools makes 12 API calls to complete a single task. What are the overheads involved in n8n?

   1. A. 12 n8n executions - one for each API call
   2. B. One n8n execution - but the LLM provider charges a token fee for each inference step and tool call.
   3. C. 6 executions - one for each tool

   EXPLAIN:

   n8n considers the entire workflow process as a single execution regardless of how many tool calls occur. However, your LLM provider (OpenAI, Anthropic, etc.) charges tokens per inference step. An agent that iterates 12 times will use 12 times as many tokens as a single prompt. Keep an eye on token usage in complex agent workflows – it can accumulate faster than you think.

2. Question 2:

   Your research agent consistently ignores Wikipedia and relies solely on web search. How would you address this?

   1. A. Remove the web search engine and force it to use Wikipedia.

2. B. Update the system prompt to provide clear instructions on when to use each tool – for example: 'Use Wikipedia for factual definitions and historical context. Use web search for recent events and current data.'
3. C. Increase the model's temperature to make it more creative.

EXPLAIN:

The system prompt is your primary control mechanism. Agents decide which tools to call based on instructions in the system prompt. If you don't specify when to use each tool, the agent will default to whichever tool is active first. Clearly describe the purpose of each tool and when the agent should prioritize using it.

3. Question 3:

What are the main differences between Basic LLM Chain and AI Agent in n8n?

1. A. AI Agent uses a different LLM model than Basic LLM Chain.
2. B. AI Agents can decide which tools to call, in what order, and repeat the process to infer the result – whereas Basic LLM Chains only process a request once.
3. C. AI Agents are faster because they process requests in parallel.

EXPLAIN:

AI agents are autonomous. When given a task, they reason about which tools to use, call them, evaluate the results, and decide whether to call additional tools or return the final answer. This reasoning loop (called ReAct - Reason + Act) is what makes it an agent, not a chain. Basic LLM chains process your request once and return a response—no loops, no tool calls.

Submit your work

Training results

You have completed **0** questions.

-- / --

Review the lesson