Think Tool node documentation

# Think Tool node#

The Think Tool node is a tool that invites an agent to do some thinking before answering. Essentially, it allows the agent to reflect on the question by talking to itself before providing an answer. In particular, this is useful for complex queries that require a thoughtful response.

Parameter resolution in sub-nodes

Sub-nodes behave differently to other nodes when processing multiple items using an expression.

Most nodes, including root nodes, take any number of items as input, process these items, and output the results. You can use expressions to refer to input items, and the node resolves the expression for each item in turn. For example, given an input of five name values, the expression {{ $json.name }} resolves to each name in turn.

name

{{ $json.name }}

In sub-nodes, the expression always resolves to the first item. For example, given an input of five name values, the expression {{ $json.name }} always resolves to the first name.

name

{{ $json.name }}

## Templates and examples#

Browse Think Tool integration templates, or search all templates

### Prompting example#

As an example, say you ask the agent:

Jack is looking at Anne. Anne is looking at George. Jack is married, George is not, and we don't know if Anne is married. Is a married person looking at an unmarried person?

The agent will do its thinking with the tool:

We have the following information: 1. Jack is looking at Anne. 2. Anne is looking at George. 3. Jack is married. 4. George is not married. 5. We don't know if Anne is married or not.

We need to determine if a married person is looking at an unmarried person. Let's consider both cases for Anne:

Case 1: Anne is married.

• If Anne is married, then she (married) is looking at George (unmarried). In this case, a married person (Anne) is looking at an unmarried person (George).

Case 2: Anne is unmarried.

• If Anne is unmarried, then Jack (married) is looking at Anne (unmarried). In this case, a married person (Jack) is looking at an unmarried person (Anne).

In both scenarios, we find that a married person is looking at an unmarried person. Therefore, the answer to the question is yes.

and then afterward provide the answer:

Yes, a married person is looking at an unmarried person. In both possible scenarios regarding Anne's marital status, we find that a married individual (either Jack or Anne) is looking at an unmarried individual (either George or Anne).

## Related resources#

Refer to LangChain's documentation on tools for more information about tools in LangChain.

View n8n's Advanced AI documentation.

## AI glossary#

• completion: Completions are the responses generated by a model like GPT.

• hallucinations: Hallucination in AI is when an LLM (large language model) mistakenly perceives patterns or objects that don't exist.

• vector database: A vector database stores mathematical representations of information. Use with embeddings and retrievers to create a database that your AI can access when answering questions.

• vector store: A vector store, or vector database, stores mathematical representations of information. Use with embeddings and retrievers to create a database that your AI can access when answering questions.