Web Search Formula Functions — User Guide

(Un)Perplexed Spready

Three formula families provide web search capabilities, each with a different level of AI involvement:

Family	Description
WEBSEARCH	Web search + LLM summarization (always two-stage)
ASK_LOCAL_WEB	Local AI that decides whether to search
ASK_REMOTE_WEB	Remote AI that decides whether to search

The key difference from all other AI formulas (ASK_LOCAL, ASK_REMOTE, PERPLEXITY) is that these formulas can retrieve **CURRENT, LIVE information** from the internet — prices, news, regulations, stock values, etc.

WEBSEARCH — Web Search + LLM Summarization

Executes a web search, then sends the results to your local LLM for summarization and processing. The final cell value is the LLM response.

Workflow

1. WebSearch() fetches results from enabled search engines
2. Results are formatted and sent to the local LLM endpoint
3. LLM processes and returns a summarized answer

Use this when you want search results processed by your local LLM, without the complexity of tool-calling.

Syntax

```
=WEBSEARCH(query)
=WEBSEARCH0(query)                 ← same as WEBSEARCH
=WEBSEARCH1(query, filter)
=WEBSEARCH2(query, filter, maxResults)
```

Parameters

Parameter	Description
query	The search query string (required)
filter	Additional filter, e.g. "site:ollama.com" (WEBSEARCH1+)
maxResults	Maximum number of results to return, numeric (WEBSEARCH2)

Examples

```
=WEBSEARCH("current EUR/USD exchange rate")
=WEBSEARCH0("FDA pharmaceutical serialization 2025")
=WEBSEARCH1("llama3.1 changelog", "site:ollama.com")
=WEBSEARCH2("EU pharma serialization regulation", "site:ema.europa.eu", 5)
```

Typical Use Cases

- Get current prices, rates, or statistics (processed by LLM)
- Summarize search results in a specific format
- Extract specific data from web search results

ASK_LOCAL_WEB / ASKLOCALWEB — Local AI with Optional Web Search

A two-round tool-calling variant of ASK_LOCAL. In Round 1 the local AI model receives a web_search tool definition alongside your question.

The model then makes a decision:

- If the question requires current or live data, it issues a `tool_call` and the application executes WebSearch() and sends results back in Round 2, from which the model produces the final answer.
- If the question can be answered from training data alone, it responds directly — making it as fast as a regular ASK_LOCAL call.

This means you do **NOT** need to decide in advance whether a web search is needed. The AI decides for you based on the nature of the question.

REQUIRES: A local model that supports OpenAI function-calling protocol.

Recommended models: llama3.1, mistral-nemo, qwen2.5, qwen2.5-coder

Older models (llama2, mistral 0.1, etc.) will answer directly without web search — no error, just no search performed.

Syntax

```
=ASK_LOCAL_WEB(prompt)
=ASK_LOCAL_WEB0(prompt)                ← same as ASK_LOCAL_WEB
=ASK_LOCAL_WEB1(input, instruction)
=ASK_LOCAL_WEB2(input1, input2, instruction)
=ASK_LOCAL_WEB3(input1, input2, input3, instruction)
```

All names below are fully equivalent (underscore vs. no underscore):

- ASK_LOCAL_WEB = ASKLOCALWEB
- ASK_LOCAL_WEB0 = ASKLOCALWEB0
- ASK_LOCAL_WEB1 = ASKLOCALWEB1
- ASK_LOCAL_WEB2 = ASKLOCALWEB2
- ASK_LOCAL_WEB3 = ASKLOCALWEB3

Examples

Simple question (model decides to search):

```
=ASK_LOCAL_WEB("What is the current EUR/USD exchange rate?")
=ASK_LOCAL_WEB("Latest news about Ollama model releases")
=ASKLOCALWEB("Current Brent crude oil price in USD")
```

Simple question (model answers from knowledge — no search):

```
=ASK_LOCAL_WEB("What is the capital of France?")
=ASK_LOCAL_WEB("Explain what a GTIN barcode is")
```

One data input + instruction:

```
=ASK_LOCAL_WEB1(A2, "Search the web for the current price of this product")
=ASK_LOCAL_WEB1(A5, "Find the latest regulatory status for this drug")
=ASKLOCALWEB1(A3, "Is this company still active? Search if needed")
```

Two data inputs:

```
=ASK_LOCAL_WEB2(A2, B2, "Find recent news about both of these companies")
=ASK_LOCAL_WEB2(A5, B5, "Compare current market prices for both products")
```

Three data inputs:

```
=ASK_LOCAL_WEB3(A2, B2, C2, "Find and compare current regulations for all three")
```

Cross-sheet range as input:

```
=ASK_LOCAL_WEB1("RANGE:Products!A2:A50",
              "Search and find the current price for each product")
```

Settings: Uses the same endpoint, model, and API key as ASK_LOCAL. Configure in **Settings > AskLocal Settings**.

ASK_REMOTE_WEB / ASKREMOTEWEB — Remote AI with Optional Web Search

Identical in behavior to ASK_LOCAL_WEB, but uses the remote server endpoint defined by the compile-time constants `REMOTEAPIURL` and `REMOTEAPIMODEL`. No separate settings dialog — endpoint and model are set in source constants.

Use this when:

- Your local hardware is too slow for the model required for tool calling
- You want to use a more capable remote model for web-search tasks
- Your local instance is busy and a remote server is available

Syntax

```
=ASK_REMOTE_WEB(prompt)
=ASK_REMOTE_WEB0(prompt)              ← same as ASK_REMOTE_WEB
=ASK_REMOTE_WEB1(input, instruction)
=ASK_REMOTE_WEB2(input1, input2, instruction)
=ASK_REMOTE_WEB3(input1, input2, input3, instruction)
```

All names below are fully equivalent:

- `ASK_REMOTE_WEB` = `ASKREMOTEWEB`
- `ASK_REMOTE_WEB0` = `ASKREMOTEWEB0`
- `ASK_REMOTE_WEB1` = `ASKREMOTEWEB1`
- `ASK_REMOTE_WEB2` = `ASKREMOTEWEB2`
- `ASK_REMOTE_WEB3` = `ASKREMOTEWEB3`

Examples

```
=ASK_REMOTE_WEB("Latest FDA guidance on pharmaceutical serialization")
=ASK_REMOTE_WEB0("Current WHO essential medicines list updates")
=ASKREMOTEWEB("Latest news about EU FMD regulation changes")

=ASK_REMOTE_WEB1(A5, "Find this drug's current EMA approval status")
=ASKREMOTEWEB1(A3, "Search for the latest price of this raw material")

=ASK_REMOTE_WEB2(A5, B5, "Find and compare regulatory status in both markets")

=ASK_REMOTE_WEB3(A2, B2, C2, "Find latest import regulations for all three")
```

Cross-sheet range as input:

```
=ASK_REMOTE_WEB1("RANGE:Drugs!A2:A100",
               "Find current EMA approval status for each drug")
=ASK_REMOTE_WEB2("RANGE:Products!A2:A50", "RANGE:Markets!B2:B10",
               "Check current import regulations for each combination")
```

How Two-Round Tool Calling Works

Round 1 — AI receives your question + web_search tool definition

The model evaluates the question. Two possible outcomes:

- **A)** Model issues `tool_call` → proceeds to Round 2
- **B)** Model answers directly → result is returned immediately (no search)

Round 2 — Web search executed, results fed back to AI

If a `tool_call` was issued:

1. The application calls WebSearch() with the model's search query
2. Search results are sent back to the model as a tool message
3. The model produces a final answer based on the search results
4. The final answer is written into the cell

Note: The search query in Round 2 is formulated by the AI itself, not by you. This means the model may rewrite your question into a more effective search query automatically.

Choosing Between the Three Families

Goal	Formula
I want search results processed by local LLM	**WEBSEARCH**
I want AI to answer, searching the web only if needed, using a local model	**ASK_LOCAL_WEB**
I want AI to answer, searching the web only if needed, using a remote model	**ASK_REMOTE_WEB**
I always want the web searched (cloud, paid, highest quality)	**PERPLEXITY** (see main documentation)
I never need web search (faster, no internet dependency)	**ASK_LOCAL** or **ASK_REMOTE** (see main documentation)

Important Notes

1. Recalculation

These formulas do **NOT** recalculate automatically. Use:

- Menu > Recalculate Selected Cell
- Menu > Recalculate Selected Range
- Menu > Recalculate All Formulas

Or edit and confirm the cell to trigger a single recalculation.

2. Defer Calculation

Enable **Settings > Defer Formula Calculation** while building a sheet with many web-search formulas to avoid hitting the API on every edit.

3. Error Messages

Error	Meaning
APIERR	HTTP error or network unreachable
LICENSE LOCKED	Premium features not activated
Deferred	Calculation deferred — trigger manually
N/A	Model could not produce an answer

4. Model Compatibility (ASK_LOCAL_WEB / ASK_REMOTE_WEB only)

If the configured model does not support function calling, the formula falls back silently to a direct answer (same behavior as ASK_LOCAL / ASK_REMOTE). No error is shown. If you suspect this is happening, check the application debug log for the message: `"no tool_call, falling back to CallAskLocalAI"`

5. Copy-Paste

Always use the application's **Custom Copy and Custom Paste actions** (not the OS default Ctrl+V) to ensure cell references are adjusted correctly when copying cells containing these formulas.

Happy web-searching!