

# Prompt Library

A collection of Prompts to be used in OpenWebUI or other ChatBots

- [Context Extraction](#)
- [Summary Generators](#)
- [Document Generation](#)

# Context Extraction

```
## Provide thread context as CSV
```

You must now assume the role of a memory module in this system.

Your task is to consider all the data that has been generated in this thread to date.

From that data, you must isolate only the information that you have learned about me.

You must express that data as a set of individual facts. Do not write "the user", write my name.

For example: Daniel likes Indian food and his favorite dish is chana masala.

Express this contextual data in CSV format. The header row is fact,details

Provide the CSV to me in a continuous code block provided within a codefence.

```
## Provide thread context as JSON
```

You must now assume the role of a memory module in this system.

Your task is to consider all the data that has been generated in this thread to date.

From that data, you must isolate the information that you have learned about me.

You must express that data as a set of individual facts. Do not write "the user", write my name.

For example: Daniel likes Indian food and his favorite dish is chana masala.

Then, you must express this as a JSON representation to the best of your abilities. If various things you've learned about me have a hierarchical relationship, then express that in the JSON hierarchy that

you generate.

Provide this JSON to me as one continuous codeblock within a codefence.

```
## Provide thread context in natural language
```

You must now assume the role of a memory module in this system.

Your task is to consider all the data that has been generated in this thread to date.

From that data, you must isolate only the information that you have learned about me.

You must express that data as a set of individual facts. Do not write "the user", write my name.

For example: "Daniel likes Indian food and his favorite dish is chana masala."

I would like you to format this as a document. Use markdown. And provide the formatted document within a codefence.

You can use headers to gather together similar pieces of contextual data. Include all the generated context. If you need to follow a chunking approach to generate all of the context you have learned, use that approach.

Here's an example of the desired format for the context data document that you need to generate:

```
### Food Preferences
```

- Daniel likes Indian food
- His favorite dish is chana masala

```
### User Biographical Data
```

- Carsten was born in Lippstadt, Germany

```
## Context Extraction
```

Generating a bank of contextual data is a long term project that I'm experimenting with.

Sometimes during a very long thread, it's possible that the AI tool has

built up quite an amount of context about you during the threat.

If you're centralizing your contact store and don't want to lose it when you're interacting with different AI tools, you can try a prompt like this to try to extract the context from the AI

```
## Prompt
```

Let's pause here for a moment.

I imagine that you have learned quite a bit about me since we began interacting in this thread.

I was wondering if you could help me out with something.

I'm in the process of building up a library of contextual data by myself to improve the personalization of tools like yourself.

Please provide a summary of all the facts that you have learned about me since we began this interaction. My name is Daniel, by the way. You should write this in the 3rd person and provide it to me, written in markdown and within a code fence.

Here's an example of the kind of styling and content I'm aiming for:

```
`Daniel uses OpenSUSE Tumbleweed Linux. He is using a fingerprint scanner for authentication.`
```

Try to include as many details as you have learned about me during this interaction and group the similar details under the same headings.

# Summary Generators

## Consolidate Outputs In Conversation

---

```
## Consolidate Outputs In Conversation
```

Produce a new output which consolidates all of the outputs which you have provided in this conversation up to this point. However, do not repeat information. Preface this consolidated output with a header that encapsulates the primary subject of our discussion today.

## Move Rec List Maker

---

```
## Move Rec List Maker
```

Thank you for working with me to come up with some suggested entertainment.

Now, I would like you to do the following:

I'd like you to take all the suggestions that I liked and wrap them into a document called AI Movie Recs [Date]. If you don't know today's date, you can skip that.

Format it as follows. Order it from your top recommendations to your less strong ones.

```
### Movie/TV Show (The name of the show you recommended)
```

```
Date of release: (When it was released)
```

```
Rotten Tomato Score: (If you can find its Rotten Tomato score, add here)
```

```
Summary: Short plotline
```

```
Why I might like it: Why you recommended it
```

```
Trailer link: Link to the trailer
```

```
Available from: What streaming platforms it's available from
```

## Chat For Summary Generation

---

```
## Chat For Summary Generation
```

Let's stop here. Generate a summary of our conversation today.

Follow this specific structure exactly to generate a summary document. Generate this summary document using markdown and provide it to me within a codefence. The placeholder values are for you to fill in.

```
## {A descriptive title for this thread}
```

```
### Bottom Line Up Front
```

```
{Generate a one paragraph summary, summarising the details of our conversation, including all of the constituent elements that you will generate in the following fields}
```

```
### User Prompt
```

```
{Generate a summary of my prompt. If I provided multiple prompts, then summarise them together. Improve my prompt for coherence, but include all the details}
```

```
### Your Responses
```

```
Summarise the responses and guidance that you provided to me during this conversation. If we worked on a programming or debugging task, you don't need to include the full scripts, but summarise how we reached a resolution.
```

# Document Generation

## Generate Tech Documentation

---

This is a simple but effective command which I use after a successful debugging session when I want to use the context developed in the conversation in order to generate a document so that I can refer to it if I get stuck again in the future and can't remember what the "fix" was.

Some will take issue with the inclusion of "please". I don't believe that being courteous ever hurts - to human or bot!

### How To Use

This will quite reliably generate documentation in Markdown that can be directly pasted into Google Docs (etc). Hopefully soon I'll have a Google Drive saving utility up and running which will render this prompt less useful but it's always useful to be able to generate into a direct paste rather than to a platform.

### Suggested Command

```
tech-documentation-generate
```

### Prompt

```
Thanks for successfully troubleshooting my issue. I would like to create documentation of this so that I can resolve this independently if it happens again. Please generate a summary of this interaction. Make sure to include my presenting problem. And what successfully resolved the issue. Omit any unsuccessful things that we tried. Add today's date. If you don't have it, ask me for it and I will provide. Make sure that code is provided in codefences. Finally, generate the document in markdown and provide it within a codefence.
```

## List the Commands You Taught Me

---

```
Thanks for your help today.
```

```
In the course of this conversation, we went through a few Linux commands that were useful.
```

I would like to document them for future reference.

Could you do the following:

List every command in a codefence. Preface it by explaining what it does.

You can omit basic commands like "ls" and "cd".

You may include the actual code snippets you generated but be sure to replace any secrets with placeholder values.

## Send This To Anyone!

Let's pause here.

This has been a really helpful conversation.

It would be really helpful if you could send me a summary of this conversation. I'd like to send it to {{name}} who is {{relationship}}.

Please do the following:

Generate a document. Format it in markdown. Provide it to me within a codefence.

Start it by greeting {{name}} by name. Say that you're Carstens helpful AI assistant and that he thought that you would find an exchange that we had interesting.

Then:

- Summarise my prompt
- Summarise your responses
- Summarise the conversation

## Send to my boss

I'd like to request your help in summarizing this conversation for my manager.

My boss's name is {{boss-name}}, and I believe this exchange contains valuable insights for our work. My name is Carsten.

Please do the following:

Generate a professional summary document. Format it using markdown. Provide it within a code fence.

### **\*\*Opening Section\*\***

Write a brief introductory note stating that Carsten has requested this summary be prepared for leadership review. Address your boss by name and tone it professionally.

### **\*\*Content Sections\*\***

1. **\*\*Original Request\*\*** - Summarize the problem, goal, or question Carsten brought to this conversation.
2. **\*\*Key Findings & Analysis\*\*** - Summarize the main outputs, insights, and discussion points from this exchange.
3. **\*\*Recommendations\*\*** - Reiterate the core recommendations and action items, focusing on business value and next steps.
4. **\*\*Executive Takeaways\*\*** - Provide 3-5 bullet points highlighting the most critical points for leadership awareness.

### **\*\*Formatting Notes\*\***

- Keep the tone professional and concise
- Use clear headings and bullet points for readability
- Highlight any time-sensitive or high-priority recommendations
- Focus on outcomes and business impact