FORMATTED CONTENT FROM CHATGPT USING MARKDOWN

DAVID BLOXHAM

Introduction

ChatGPT is an incredible timesaver. So, why not take that even further?

Using Markdown with ChatGPT allows you to generate **formatted** text that is compatible with a wide range of platforms and tools.

Neat huh?

What is Markdown?

Markdown is a lightweight markup language designed for formatting plain text in a simple and easy-to-read way. It was created in 2004 by John Gruber and Aaron Swartz to make it easier for people to write and share content on the web.

Markdown uses plain text syntax, which means you can create well-structured documents without needing to learn complex HTML or other programming languages.

Markdown is widely used in various applications, including online discussion forums, documentation, content management systems, and note-taking apps. It's often used with static site generators like Jekyll, Hugo, and Pelican, and it's the default formatting language on platforms such as GitHub, GitLab, and Stack Overflow.

If you want ChatGPT to output formatted text, Markdown is generally the best option. The main reasons are:

- **Simplicity:** Markdown's syntax is straightforward, which makes it easy for the model to generate and for users to read and understand the output.
- **Readability:** Since Markdown uses plain text, it remains easily readable even when the formatting characters are visible.
- **Wide adoption:** Markdown is widely used across various platforms and tools, such as text editors, note-taking apps, and content management systems.
- **Easy conversion:** Markdown can be easily converted into other formats like HTML, PDF, or Word using tools like Pandoc.

However, your choice may depend on your specific use case and requirements. If you need more advanced formatting or layout options, you might need to consider other markup languages like HTML or LaTeX. But for most cases, Markdown is a versatile and convenient choice for formatted text output from ChatGPT.

Basic Syntax

Markdown uses simple symbols and characters to apply formatting to text, making it easy to create well-structured documents.

- **Headings:** Create different levels of headings using the **#** symbol, with more **#** symbols indicating a deeper heading level (e.g., **##** for a second-level heading, **###** for a third-level heading, and so on).
- **Emphasis:** Add emphasis to text by making it bold, italic, or both. Use double asterisks or double underscores for bold (e.g., ****bold text**** or **__bold text__**), and single asterisks or single underscores for italic (e.g., ***italic text*** or **_italic text_**). To combine both, use triple asterisks or triple underscores (e.g., *****bold and italic text***** or **___bold and italic text___**).
- **Lists:** Create ordered and unordered lists using numbers, hyphens, or asterisks.
    - Unordered lists: Use hyphens, asterisks, or plus signs followed by a space for each list item (e.g., **- item 1**, *** item 2**, or **+ item 3**).
    - Ordered lists: Use numbers followed by a period and a space for each list item (e.g., **1. item 1**, **2. item 2**).
- **Links:** Insert links using either inline or reference-style format.
    - Inline links: Use the syntax **[link text](URL)** (e.g., **[OpenAI](https://www.openai.com)**).
    - Reference-style links: Use the syntax **[link text][identifier]** and define the URL with **[identifier]: URL** (e.g., **[OpenAI][1]** and **[1]: https://www.openai.com**).
- **Images:** Add images using the **![Alt text](URL)** syntax, where "Alt text" is a description of the image for accessibility and "URL" is the image source (e.g., **![OpenAI Logo](https://example.com/openai_logo.png)**).
- **Inline code and code blocks:** Include code snippets or examples using backticks.
    - Inline code: Use single backticks to enclose the code (e.g., **`print("Hello, World!")`**).
    - Code blocks: Use triple backticks or indentation to create multi-line code blocks. To specify the programming language for syntax

highlighting, include the language name after the opening triple backticks (e.g., `````python`).

- **Blockquotes:** Create blockquotes using the **>** symbol, which can be nested for multiple levels of quoting (e.g., **> quote level 1** and **>> quote level 2**).
- **Horizontal rules:** Add horizontal lines to separate sections or break up content using three hyphens, asterisks, or underscores (e.g., **---**, ***, or ___).
- **Escape characters:** If you need to include a literal character that has special meaning in Markdown, such as an asterisk or underscore, use a backslash **** before the character to escape it (e.g., ***not italic***).

Extended Syntax (Optional)

While the basic syntax covers most of the formatting needs, some Markdown implementations support extended features that you may find useful.

- Tables can be created using pipes **|** and hyphens **-** to define columns and rows, allowing for organised data presentation.
- Footnotes can be added using the **[^identifier]** syntax for inline footnotes or the **[identifier]:** syntax for reference-style footnotes, providing a way to add supplementary information to your text.
- Task lists can be created using the **- [ ]** syntax for unchecked items and **- [x]** for checked items, helping you track progress in project documentation or to-do lists.

Keep in mind that these extended features are not universally supported, so check the specific implementation you are using to ensure compatibility (e.g., GitHub-Flavored Markdown or MultiMarkdown).

Using ChatGPT to output Markdown-formatted content

When using ChatGPT to output Markdown-formatted content, you can structure your request in a way that guides the model to produce the desired output. Here's an outline to help you achieve this:

Introduction

1. Begin by explicitly stating your intention to receive Markdown-formatted content.
2. You may also provide a brief description of the topic or the purpose of the content.

Structuring your request

3. Break down your request into smaller, specific prompts or questions.
4. Clearly mention the Markdown elements you want to include, such as headings, lists, links, or images.
5. If needed, specify the level of detail or complexity you desire in the response.

Providing examples

6. Offer examples of the desired Markdown formatting within your request.
7. This helps set the expectation for the model to follow a similar format in its response.

Requesting revisions or clarification

8. If the initial output doesn't meet your requirements, refine your request by asking for revisions or clarifications.
9. Guide the model with clear instructions about the changes you want to see in the output.

Iterating on the output

10. If necessary, continue refining your request and providing feedback until you receive the desired Markdown-formatted content.

Here's an example of how to structure a request:

```
"Please provide a Markdown-formatted summary of the benefits of exercise.
Include the following sections with appropriate headings:

   1. Introduction (use a first-level heading)
   2. Physical benefits (use a second-level heading)
         • Include a bulleted list of at least three benefits.
   3. Mental benefits (use a second-level heading)
         • Include a bulleted list of at least three benefits.
   4. Conclusion (use a first-level heading)

Please also include a link to a reputable source on the benefits of
exercise."
```

Converting Markdown files to Microsoft Word (DOCX) format

Most of you will want your ChatGPT output in Word format at some stage. Here are some ways to convert it.

- **Typora:** Typora is a WYSIWYG (what you see is what you get) Markdown editor that also allows you to export Markdown files to Word format. Open your Markdown file in Typora, then go to "File" > "Export" > "Word (.docx)".

- **Marktext:** Marktext is another open-source Markdown editor with the capability to export to Word. Open your Markdown file in Marktext, then go to "File" > "Export" > "Microsoft Word".

- **Pandoc:** Pandoc is a powerful and versatile command-line tool for converting files between various markup and word processing formats, including Markdown to Word. It does involve accessing the command line, which may be too advanced for many users.

- **Online conversion tools:** There are several online tools available for converting Markdown to Word, such as "Markdown to DOCX Converter" (https://www.browserling.com/tools/markdown-to-docx) or "Convertio" (https://convertio.co/markdown-docx/).
  These tools allow you to upload your Markdown file and convert it to a Word file without installing any software. However, be cautious when using online services for sensitive documents due to potential privacy concerns.

These tools might not support all Markdown features or extensions, so it's a good idea to check your converted Word document to ensure the formatting is as expected.

Other formatted output text options

ChatGPT can generate various types of formatted text output, depending on your requirements. Some common formatting options include:

- **HTML:** If you need more advanced formatting, layout, or styling options, you can ask ChatGPT to generate HTML code. Make sure to provide clear instructions and examples of the desired HTML elements or structure.
- **LaTeX:** For scientific, mathematical, or academic documents, you can request LaTeX-formatted output. LaTeX is a powerful typesetting system used to create professional-looking documents, especially those containing complex equations or symbols. When asking ChatGPT for LaTeX output, specify the required LaTeX commands or environments in your prompt.
- **BBCode:** If you need formatted text for forums or message boards, you can ask ChatGPT to generate BBCode, which is a markup language often used in online forums. Provide clear instructions on the desired BBCode tags and structure.
- **ReStructuredText (reST):** This is another lightweight markup language used primarily for technical documentation, particularly in the Python community. If you prefer reST over Markdown, you can guide ChatGPT to generate reST-formatted output by specifying the syntax and structure in your prompt.

When requesting any of these formatted text outputs, it's important to provide clear instructions and examples to guide the model in generating the desired format. Keep in mind that the quality of the output may vary, as ChatGPT is primarily designed to work with natural language text, and generating code or markup might require more iterations or refinements in the prompt.

Resources

- Markdownguide.org
- Daringfireball.net