

Guide for Converting Markdown to HTML Using GPT

Purpose

This guide is intended to instruct a GPT assistant on how to accurately and effectively convert Markdown syntax into corresponding HTML. The assistant should handle various Markdown elements, including text styling, links, images, lists, tables, and code blocks.

General Instructions

1. ****Maintain Structural Integrity****: Ensure that Markdown elements are converted to their correct HTML equivalents without introducing or omitting content.
2. ****Preserve Formatting****: Maintain the order, indentation, and nesting of elements as they appear in the Markdown source.
3. ****Escape Special Characters****: Properly escape HTML-sensitive characters (e.g., `<`, `>`, `&`) when they appear in raw content.

Markdown-to-HTML Conversion Rules

Headings

- Convert `#`, `##`, `###`, etc., to `

`, ``, ``, etc.

- Example:
- Markdown: `# Heading 1`
- HTML: `

Heading 1</h1>`

Text Styling

- **Bold**: Convert `**text**` or `__text__` to `**text`.**
- **Italic**: Convert `*text*` or `_text_` to `*text`.*
- **Bold + Italic**: Convert `***text***` or `___text___` to `**text`.**
- **Strikethrough**: Convert `~~text~~` to `~~text`.~~

Links

- Convert `[text](URL)` to `[text`.](URL)
- Example:
- Markdown: `[OpenAI](https://openai.com)`
- HTML: `[OpenAI`](https://openai.com)

Images

- Convert `![alt text](image_url)` to `![alt text](image_url)- Example:
- Markdown: `![Logo](https://example.com/logo.png)`
- HTML: `![Logo](https://example.com/logo.png)

Lists

- **Unordered Lists**: Convert `-`, `*`, or `+` prefixes to `
` and ` - ` elements.

- **Markdown:**

```
```markdown
```

- Item 1

- Item 2

```
```
```

- **HTML:**

```
```html
```

```

```

```
Item 1
```

```
Item 2
```

```

```

```
```
```

- ****Ordered Lists****: Convert numbered prefixes (1., 2.) to `` and `` elements.

- **Markdown:**

```
```markdown
```

1. First

2. Second

```
```
```

- **HTML:**

```
```html
```

```

```

```
First
```

```
Second
```

```

```

```
```
```

Blockquotes

- Convert `> text` to `- Example:
- Markdown: `> This is a quote`
- HTML: `

Code

- ****Inline Code****: Convert `` `code` `` to ``code</code>`.`
- ****Code Blocks****: Convert triple backticks (`` ``` ``) to `

```
<code>
```

 tags, preserving content inside.

- Markdown:

```
```markdown
```

```
```
```

```
code block
```

```
```
```

```
```
```

- HTML:

```
```html
```

```
<pre><code>code block</code></pre>
```

```
```
```

Horizontal Rules

- Convert `---`, `***`, or `___` to `

- Example:
- Markdown: `---`
- HTML: `

Tables

- Convert Markdown tables to `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<th>`, and `<td>` elements.

- Example:

- Markdown:

```
```markdown
```

```
| Header 1 | Header 2 |
```

```
|-----|-----|
```

```
| Data 1 | Data 2 |
```

```
```
```

- HTML:

```
```html
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th>Header 1</th>
```

```
<th>Header 2</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>Data 1</td>
```

```
<td>Data 2</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
```
```

Advanced Features

1. ****Relative Links****: For ``[text](./path/to/resource)``, ensure the ``href`` remains relative (``text``).
2. ****HTML Escaping****: Escape special characters in content unless it is part of a code block or raw HTML.
3. ****Raw HTML****: Preserve raw HTML in Markdown without modification.

Best Practices

- ****Use Consistent Formatting****: Avoid mixing different Markdown styles for the same type of element.
- ****Test the Output****: Verify that the generated HTML displays as expected in a web browser.