



Markdown Language

Theory, Tools, and Live Implementations

Seminar Presentation

Presented by

Name: [My Name]

Reg. No.: [My Reg. No.]

- 
- 1 Introduction to Markup Languages
 - 2 Understanding Markdown
 - 3 Why Choose Markdown
 - 4 History & Flavors
 - 5 Markdown in Action
 - 6 Tools & Workflow
 - 7 Markdown vs GUI Tools
 - 8 Real-World Applications
 - 9 Future of Markdown and importance of AI
 - 10 References Used
 - 11 Q&A

1. Introduction to Markup Languages

1. Introduction to Markup Languages

What is a Markup Language?

- A **Markup Language** adds symbols to plain text to style and display it in a particular manner.

Common Examples of Markup Languages

- **HTML** – Used to make web pages.
 - Example: `<h1>Hello World</h1>`
- **XML** – Used to store and share data.
 - Example: `<note><to>Sreerag</to></note>`

2. Understanding Markdown

2. Understanding Markdown

What is Markdown?

- A lightweight markup language that makes writing formatted text simple.
- Easy to write and read, even without special software.
- You can just open it in a normal text editor and you can see its contents.
- The main purpose of Markdown is to be converted into HTML.

How does it work?

- You add simple symbols like `#` or `*` to style text.

Example:

```
This is normal text with no styling  
# This one is a Level 1 heading  
**This is bold text**
```

2. Understanding Markdown - A Pictorial Example

- You write Markdown text in a text editor. You save it with `.md` file extension.

```
<!-- _class: caption -->

## 2. Understanding Markdown - A Pictorial Example

- And finally, you render it.

![#c h:420](<u>images/puppy.webp</u>)

<div class="caption">
Picture of a puppy
</div>
```

Markdown text in VS Code

2. Understanding Markdown - A Pictorial Example

- Then you convert it into HTML or PDF file using an appropriate tool.
- If you are doing a presentation in Markdown, a tool like Marp ↗ is a good choice.

```
<svg data-marpit-svg="" viewBox="0 0 1280 720">
  <foreignObject width="1280" height="720">
    <section id="9" data-paginate="true" data-class="caption" data-theme="am_dark" lang="en-GB"
      class="caption" data-marpit-pagination="9" style="--paginate:true;--class:caption;--theme:am_dark;" data-marpit-pagination-total="10" data-size="16:9">
      <h2 id="2-understanding-markdown---a-pictorial-example-2">2. Understanding Markdown - A Pictorial Example</h2>
      <ul>
        <li>And finally, you render it.</li>
      </ul>
      <p></p>
      <div class="caption">
        Picture of a puppy
      </div>
    </section>
  </foreignObject>
</svg>
```

2. Understanding Markdown - A Pictorial Example

- And finally, you render it.



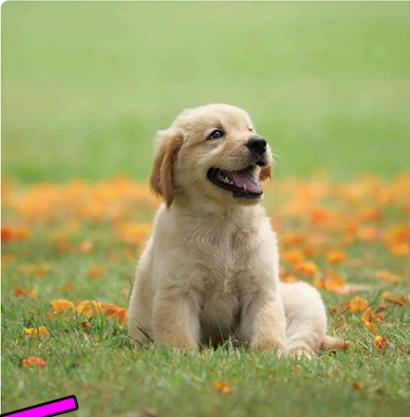
Picture of a puppy

2. Understanding Markdown - Pictorial Example Breakdown

```
<!-- _class: caption -->  
## 2. Understanding Markdown - A Pictorial Example  
  
- And finally, you render it.  
  
![#c h:420](images/puppy.webp)  
  
<div class="caption">  
Picture of a puppy  
</div>
```

2. Understanding Markdown - A Pictorial Example **LEVEL 2**

- And finally, you render it.

 Picture of a puppy

CAPTIONS

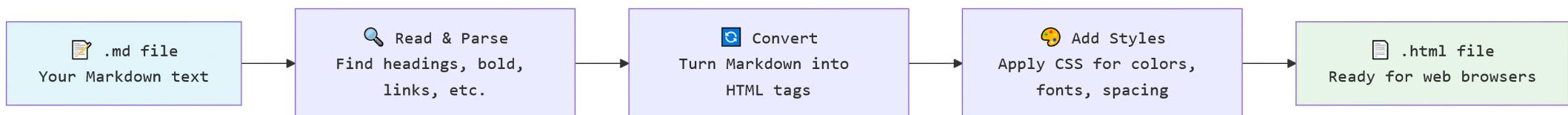
9 >> 10

Diagram illustrating the Markdown rendering process:

- A red arrow points from the heading text "2. Understanding Markdown - A Pictorial Example" to the corresponding `##` header in the code block.
- A green arrow points from the list item text "And finally, you render it." to the rendered list item in the preview.
- A pink arrow points from the image URL in the code to the rendered image in the preview.
- A yellow arrow points from the caption text in the code to the rendered caption in the preview.

2. Understanding Markdown - Pictorial Example Breakdown

- The following images represents what happens behind the scenes when an `.md` file is being converted to HTML.



3. Why Choose Markdown

3. Why Choose Markdown

- Simple & Fast
 - You don't need long HTML codes. A few symbols can do the same job.
- Readable Anywhere
 - Markdown is just plain text, so it works on any device or editor.
- Free & Open Source
 - No need for paid tools like Word or Canva. Markdown is completely free.
- Flexible
 - You can write notes, documents, slides, blogs, or even books with it.
- Career Boost
 - Learning Markdown also helps you learn HTML, CSS, and coding skills that many jobs need.

Markdown Code

```
- Simple & Fast
- No long HTML tags. Just a few symbols.
- Readable Anywhere
- Markdown files are plain text.
- Free & Open Source
- No costly tools like Word or Canva.
- Flexible
- Can make notes, docs, slides, blogs, and more.
- Career Boost
- Knowing Markdown helps with HTML, CSS, and coding jobs.
```

Equivalent HTML

```
<ul>
  <li>Simple & Fast
    <ul>
      <li>No long HTML tags. Just a few symbols.</li>
    </ul>
  </li>
  <li>Readable Anywhere
    <ul>
      <li>Markdown files are plain text.</li>
    </ul>
  </li>
  <li>Free & Open Source
    <ul>
      <li>No costly tools like Word or Canva.</li>
    </ul>
  </li>
  <li>Flexible
    <ul>
      <li>Can make notes, docs, slides, blogs, and more.</li>
    </ul>
  </li>
  <li>Career Boost
    <ul>
      <li>Knowing Markdown helps with HTML, CSS, and coding jobs.</li>
    </ul>
  </li>
</ul>
```

4. History & Flavors

4. History & Flavors

History of Markdown

- It was actually just a Perl ↗ script created by John Gruber in 2004
- The purpose was to make text formatting easy to read and write. ↗
- The original design idea was to make it “as readable as plain text, even before being rendered.”
- It is still used today in blogs, documentation, notes, and presentations (like this one). On the right is a screenshot from the documentation page of Obsidian ↗

Obsidian Help

Welcome to the official Obsidian Help site, where you can find tips and guides on how to use Obsidian. For API documentation visit the [Obsidian Developer Docs](#).

You can browse this site in several languages:

English

Get started

Learn the basics of note-taking with Obsidian:

- [Download and install Obsidian](#)
- [Create a vault](#)
- [Create your first note](#)
- [Link notes](#)
- [Import notes](#)
- [Sync your notes across devices](#)

Extend Obsidian

With thousands of plugins and themes, you can shape Obsidian to fit your way of thinking.

Core plugins Choose which built-in features of Obsidian you want to turn on or off.	Community plugins Enable more workflows and capabilities with features built by Obsidian users.
Themes Customize the look and feel of Obsidian with community-made color schemes.	CSS snippets Make small changes to the interface on top of your Obsidian theme.
Web Clipper Highlight web pages and save content to Obsidian using our browser extension.	

INTERACTIVE GRAPH

```
graph TD; Home --- ObsidianSync[Obsidian Sync]; Home --- WebClipper[Web Clipper]; Home --- API[API]; Home --- Plugins[Plugins]; Home --- Publish[Obsidian Publish]; Home --- Sync[Obsidian Sync]; Home --- Web[Obsidian Web Clipper]; Home --- Extending[Extending Obsidian]; Home --- Contributing[Contributing to Obsidian]; Home --- Licenses[Licenses and payment]; Home --- Teams[Teams]; Home --- Obsidian[Obsidian]; Home --- Help[Help and support]; Home --- Home
```

ON THIS PAGE

- [Obsidian Help](#)
- [Get started](#)
- [Extend Obsidian](#)
- [Add-on services](#)
- [Contribute](#)

4. History & Flavors

What Are Markdown Flavors?

- Flavors are different sets of rules that define how Markdown symbols work.
- Flavors decide what syntax applies which format to the text.

Why do they exist in the first place?

- The original Markdown was simple and lacked some useful features like tables, checklists, or math formulas.
- The newly created flavors added these features.
- Examples of such flavors include:
 - LaTeX ↗
 - Github Flavoured Markdown (GFM) ↗

Basic rules of CommonMark Flavor:

Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	bold text
Italic	<i>italicized text</i>
Blockquote	> blockquote
Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`
Horizontal Rule	---
Link	[title](https://www.example.com)
Image	![alt text](image.jpg)

4. History & Flavors

Common Flavors and Implementations

- An implementation is a program or library that follows a flavor's rules, written in a specific programming language (like C, JavaScript, Python).
- Example: CommonMark has implementations in many languages to keep behavior consistent.



CommonMark and Why it is different from the rest of them flavors

- The original Markdown script didn't have mechanism that defined what to do in certain scenarios, like nested bold text, or bold text within italics, etc.
- CommonMark was created to remove ambiguity in the original Markdown rules.
- Unlike other flavors, it does not introduce new features—it only standardizes syntax.

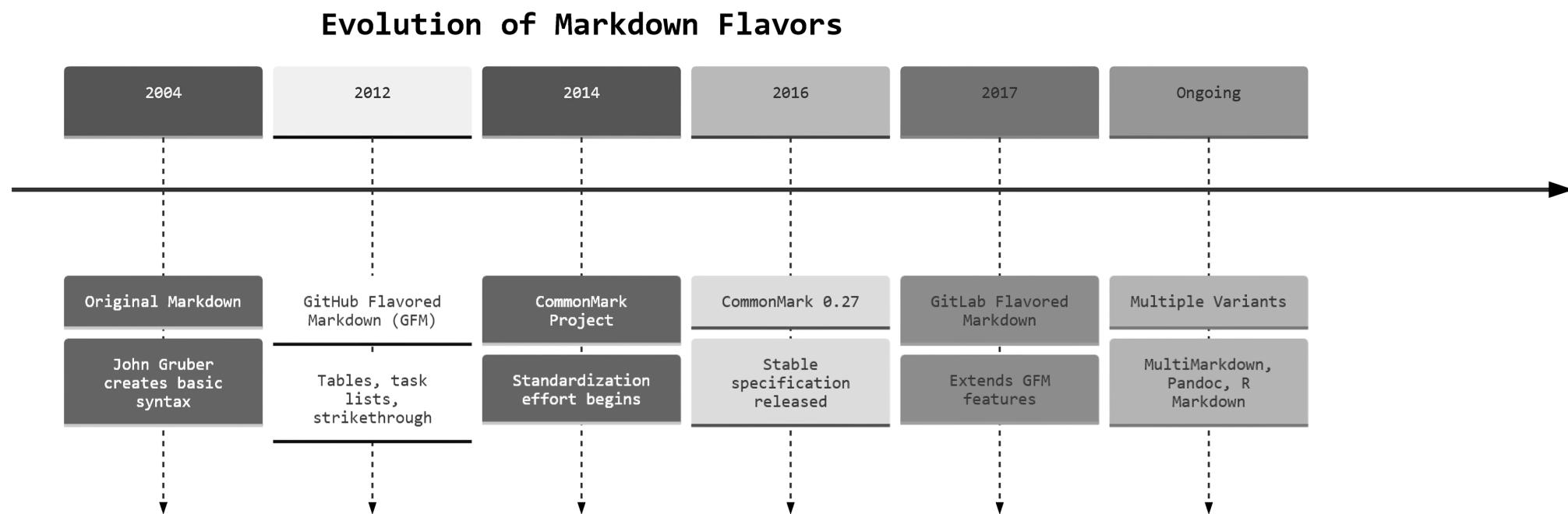
4. History & Flavors

Examples of some flavors include:

Flavor	Description
Original Markdown	First spec; loose and ambiguous
CommonMark	Standardized to remove ambiguity
GFM	GitHub's version; adds tables & lists
Markdown Extra	Adds tables, footnotes
Pandoc	Extended; supports math & slides
R Markdown	For R; runs code blocks
Markua	For writing technical books
Typora	Typora's version; adds diagrams, math
...and many more (This is already too much)	

4. History & Flavors

Some of them flavors and how they were developed over time:



4. History & Flavors

Examples of some implementations include:

Implementation	Language	Flavors Supported
cmark	C	CommonMark
commonmark.js	JavaScript	CommonMark
marked	JavaScript	GFM (supports extensions)
markdown-it	JavaScript	GFM + plugins for more
Pandoc	Haskell	Pandoc Markdown + others
Remark	JavaScript	GFM + CommonMark (extensible via plugins)
Marp	Node.js	Markdown for presentations

5. Markdown in Action

5. Markdown in Action

Basic Syntax Examples

Some of the most commonly used Markdown features are:

- **Headings** (#, ##, ###)
- **Paragraphs** and line breaks
- **Italic & Bold text** (* and **)
- **Blockquotes** (>)
- **Horizontal rules** (---

```
# Heading 1
## Heading 2
**Bold Text** and *Italic Text*
> This is a blockquote
---
```

1. **Lists** - List syntax is used to create ordered and unordered lists.

Example:

```
- Item 1
- Item 2
  - Nested Item
```

```
1. First
2. Second
```

Output

- Item 1
 - Item 2
 - Nested Item
-
1. First
 2. Second

2. Links & Images - Add clickable text pointing to a URL and add embed pictures.

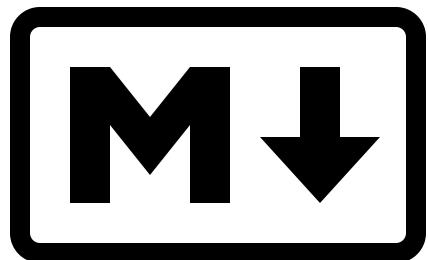
Example:

```
[Markdown](https://en.wikipedia.org/wiki/Markdown)
```

```
![Markdown Logo](https://upload.wikimedia.org/wikipedia/commons/4/48/Markdown-mark.svg)
```

Output

Markdown ↗



3. Code Blocks - Show code snippets inline or in fenced blocks.

Example:

```
```c
#include<stdio.h>
int main() {
 printf("This is a code blocks!\n");
 return 0;
}
```

```

Output

```
#include<stdio.h>
int main() {
    printf("This is a code blocks!\n");
    return 0;
}
```

4. Tables - Pretty much self-explainable.

Example:

| Vegetable | Price |
|-----------|-------|
| Onion | 100 |
| Beans | 100 |

Output

| Vegetable | Price |
|-----------|-------|
| Onion | 100 |
| Beans | 100 |

5. Task Lists - Add Checkboxes within which you can mark.

Example:

- [x] Write section
- [] Review content
- [] Submit final report

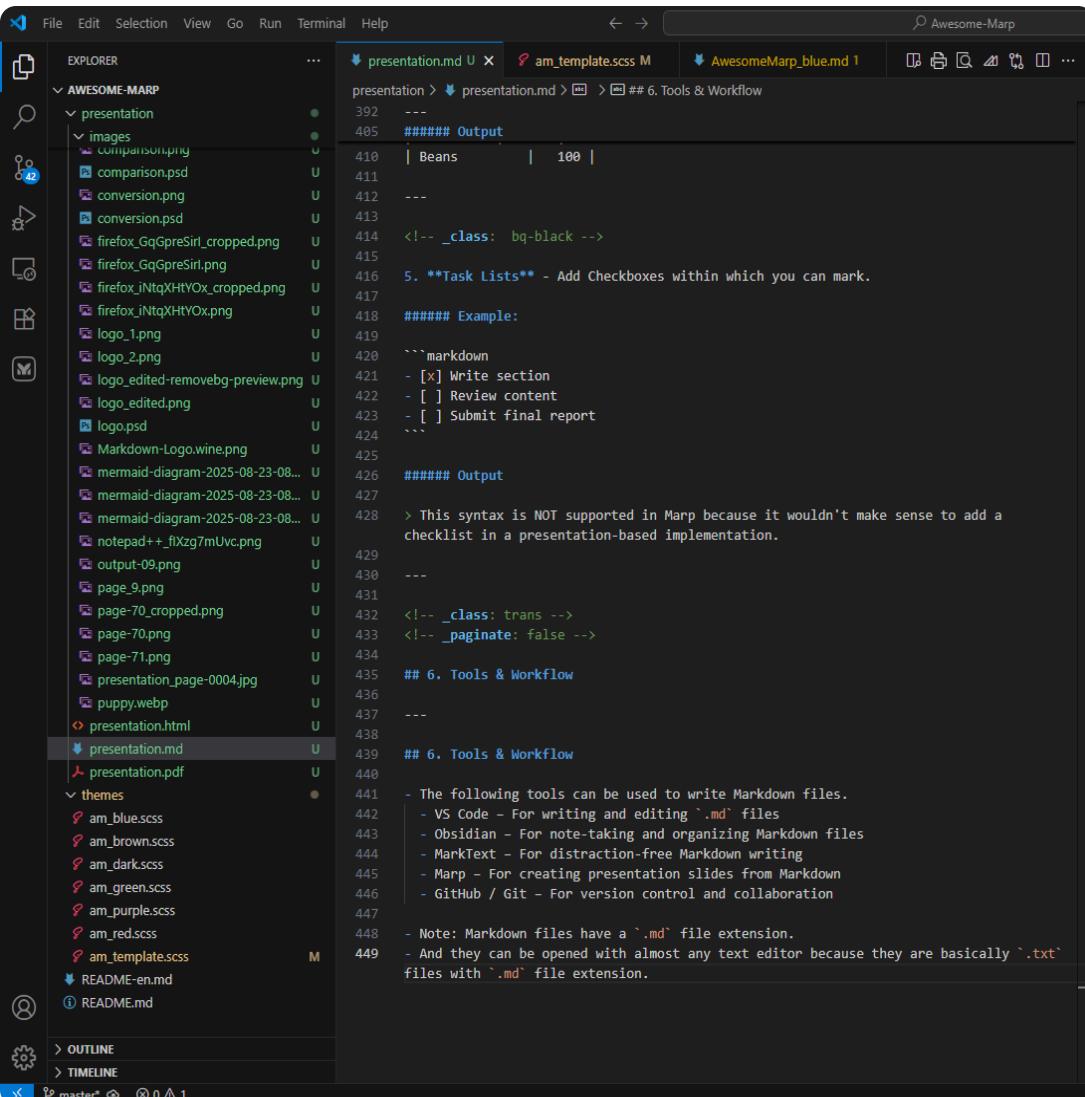
Output

 This syntax is NOT supported in Marp because it wouldn't make sense to add a checklist in a presentation-based implementation.

6. Tools & Workflow

6. Tools & Workflow

- The following **tools** can be used to write Markdown files.
 - **VS Code** – For writing and editing `.md` files
 - **Obsidian** – For note-taking and organizing Markdown files
 - **MarkText** – For distraction-free Markdown writing
 - **Marp** – For creating presentation slides from Markdown
 - **GitHub / Git** – For version control and collaboration



A screenshot of the VS Code interface showing a Marp presentation file. The left sidebar shows a file tree with a folder named 'AWESOME-MARP' containing a 'presentation' folder. Inside 'presentation' are several image files (comparison.png, comparison.psd, conversion.png, conversion.psd, firefox_GqGpreSir_cropped.png, firefox_GqGpreSir.png, firefox_INtqxHTYOx_cropped.png, firefox_INtqxHTYOx.png, logo_1.png, logo_2.png, logo_edited-removebg-preview.png, logo_edited.png, logo.psd, Markdown-Logo.wine.png, mermaid-diagram-2025-08-23-08..., mermaid-diagram-2025-08-23-08..., mermaid-diagram-2025-08-23-08..., notepad++_fIXzg7mUvc.png, output-09.png, page_9.png, page-70_cropped.png, page-70.png, page-71.png, presentation_page-0004.jpg, puppy.webp, presentation.html, presentation.md, presentation.pdf). The main editor area displays the content of 'presentation.md'. The code includes sections for 'Output' and 'Task Lists', and a note about checklist support. At the bottom, there's a list of tools for writing Markdown files, a note about file extensions, and a note about GitHub/Git.

```
File Edit Selection View Go Run Terminal Help
presentation.md U X am_template.scss M AwesomeMarp_blue.md 1
presentation > presentation.md > ## 6. Tools & Workflow
392 ---
405 ##### Output
410 | Beans | 100 |
411 ---
412 ---
413 <!-- _class: bq-black -->
414 5. **Task Lists** - Add Checkboxes within which you can mark.
415
416 5. **Task Lists** - Add Checkboxes within which you can mark.
417
418 ##### Example:
419
420 ---markdown
421 - [x] Write section
422 - [ ] Review content
423 - [ ] Submit final report
424
425
426 ##### Output
427
428 > This syntax is NOT supported in Marp because it wouldn't make sense to add a
429 checklist in a presentation-based implementation.
430
431
432 <!-- _class: trans -->
433 <!-- _paginate: false -->
434
435 ## 6. Tools & Workflow
436
437
438
439 ## 6. Tools & Workflow
440
441 - The following tools can be used to write Markdown files.
442 - VS Code - For writing and editing `*.md` files
443 - Obsidian - For note-taking and organizing Markdown files
444 - MarkText - For distraction-free Markdown writing
445 - Marp - For creating presentation slides from Markdown
446 - GitHub / Git - For version control and collaboration
447
448 - Note: Markdown files have a `*.md` file extension.
449 - And they can be opened with almost any text editor because they are basically `*.txt` files with `*.md` file extension.
```

6. Tools & Workflow

- Here is how you manage and process Markdown files.
 - i. Write in Markdown using VS Code, Obsidian, or MarkText
 - ii. Preview and Style content in chosen tool
 - iii. Export or Convert to desired format (e.g., PDF, HTML, PPTX)
 - iv. Optional: Add CSS for custom styling
 - v. Use Marp to generate presentation slides from `.md` files
- The best thing is that:
 - All tools are free and open source
 - Cross-platform and lightweight compared to Word or Canva
 - Markdown files are plain text files, the only difference being it has a `.md` file extension. This means you can open them in almost any text editor and voila! you get your content.

7. Markdown vs GUI Tools

7. Markdown vs GUI Tools

| Feature | Markdown Tools (e.g., Marp, Obsidian) | GUI Tools (e.g., Word, PowerPoint, Canva) |
|-----------------|---|---|
| Cost | Free & open-source | Often paid or subscription-based |
| Portability | Plain text files; open anywhere | Requires specific software |
| File Size | Very small (.md files) | Large (.docx, .pptx) |
| Ease of Writing | Uses simple syntax | WYSIWYG, no syntax knowledge needed |
| Flexibility | Can integrate HTML/CSS/JS for custom styles | Limited design customization |
| Version Control | Works great with Git | Harder to track changes |
| Learning Curve | Requires learning Markdown syntax | Easier for beginners |
| Output Formats | Can export to PDF, HTML, slides easily | Limited export formats |
| Collaboration | Easy via GitHub or text editors | Built-in features like comments, co-authoring |

7. Markdown vs GUI Tools

In short...

- Advantages of Markdown Tools
 - Free, lightweight, and portable
 - Free, lightweight, and portable
 - Supports automation and templates
- Advantages of GUI Tools
 - Beginner-friendly, no syntax knowledge
 - Rich visual interface with drag-and-drop
 - Comes with built-in templates and assets

 GUI tools are great for quick, design-heavy tasks, while Markdown tools excel in speed, flexibility, and technical workflows.

8. Real-World Applications

8. Real-World Applications

- Markdown can be used for many real-world applications.
- It can be used to create and manage notes, blogs, reports, presentations, and more.
- It's used by developers, writers, and teams worldwide.

Common Use Cases

- **Note-taking** → Obsidian ↗, Joplin
- **Reports & Docs** → MarkText ↗, Typora, GitHub Wiki ↗
- **Blogs & Books** → Ghost, Leanpub
- **Presentations** → Marp
- **Project Management** → GitHub Issues, Trello

9. Future of Markdown and importance of AI

9. Future of Markdown and importance of AI

- We live in the age of AI. Everything is becoming smarter.
- AI can take your raw ideas and turn them into clean, formatted Markdown.
- Perfect for docs, reports, blogs, and technical content.
- Saves time → Less typing, more creating.

Future Possibilities of Markdown with AI

- **AI-assisted Slides** → Write a topic, AI creates a Markdown-based presentation.
- **Smart Formatting** → AI converts plain text into tables, charts, and diagrams.
- **Content Generation** → AI + Markdown = quick blogs, wikis, and notes.
- **Interactive Docs** → AI brings dynamic elements inside Markdown files.

9. Future of Markdown and importance of AI

CONCLUSION

“ Why This Matters

- Markdown is simple. AI makes it smarter and faster.
- In today's world, speed and efficiency matter more than ever.
- Combining Markdown with AI means:
 - No more repetitive formatting.
 - Focus on ideas, not syntax.
 - Create professional content in minutes, not hours.
- The future of writing, teaching, and documentation is AI-powered Markdown.
- Learning Markdown today means staying ahead tomorrow.

10. References Used

10. References Used

1. The Markdown Guide by Matt Cone (book) ↗
2. Obsidian Documentation ↗
3. Marpit Documentation ↗
4. Mermaid Documentation on Flowchart and Timeline syntax ↗
5. MarkText Documentation ↗
6. Reddit
7. ChatGPT
8. Claude

THANK YOU