AKNM GPTC
An AICTE approved technical institution

Markdown Language

Theory, Tools, and Live Implementations

Seminar Presentation

Presented by
Name: [My Name]
Reg. No.: [My Reg. No.]

Made with Awesome-Marp (Github)

CONTENTS

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 it's 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](<images/puppy.webp>)

<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><img src="images/puppy.webp" alt="#c" style="height:420px;" /></p>
            <div class="caption">
                Picture of a puppy
            </div>
        </section>
    </foreignObject>
</svg>
```

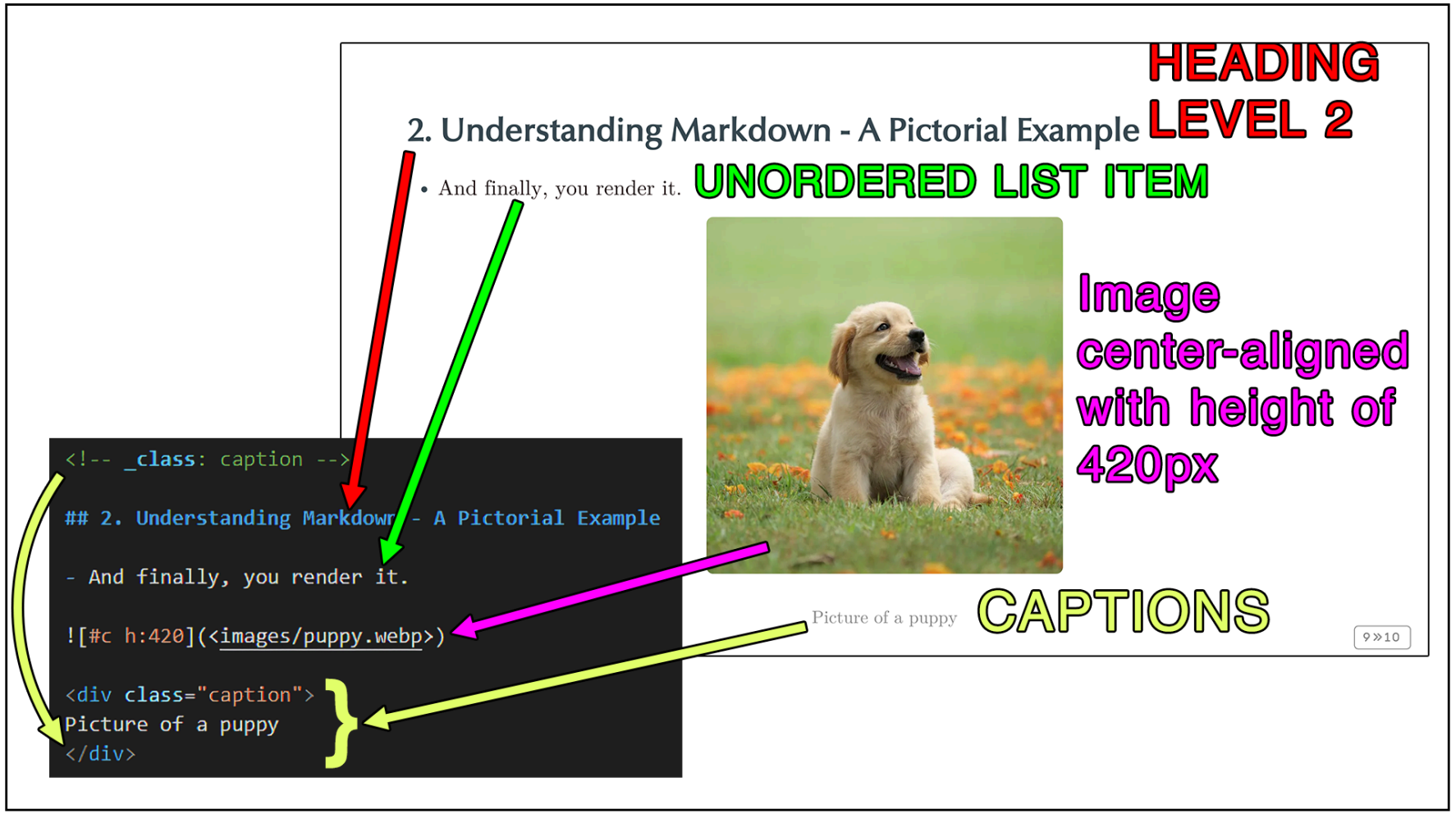
Markdown text converted to HTML using Marp

2. Understanding Markdown - A Pictorial Example

- And finally, you render it.

Picture of a puppy

2. Understanding Markdown - Pictorial Example Breakdown



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

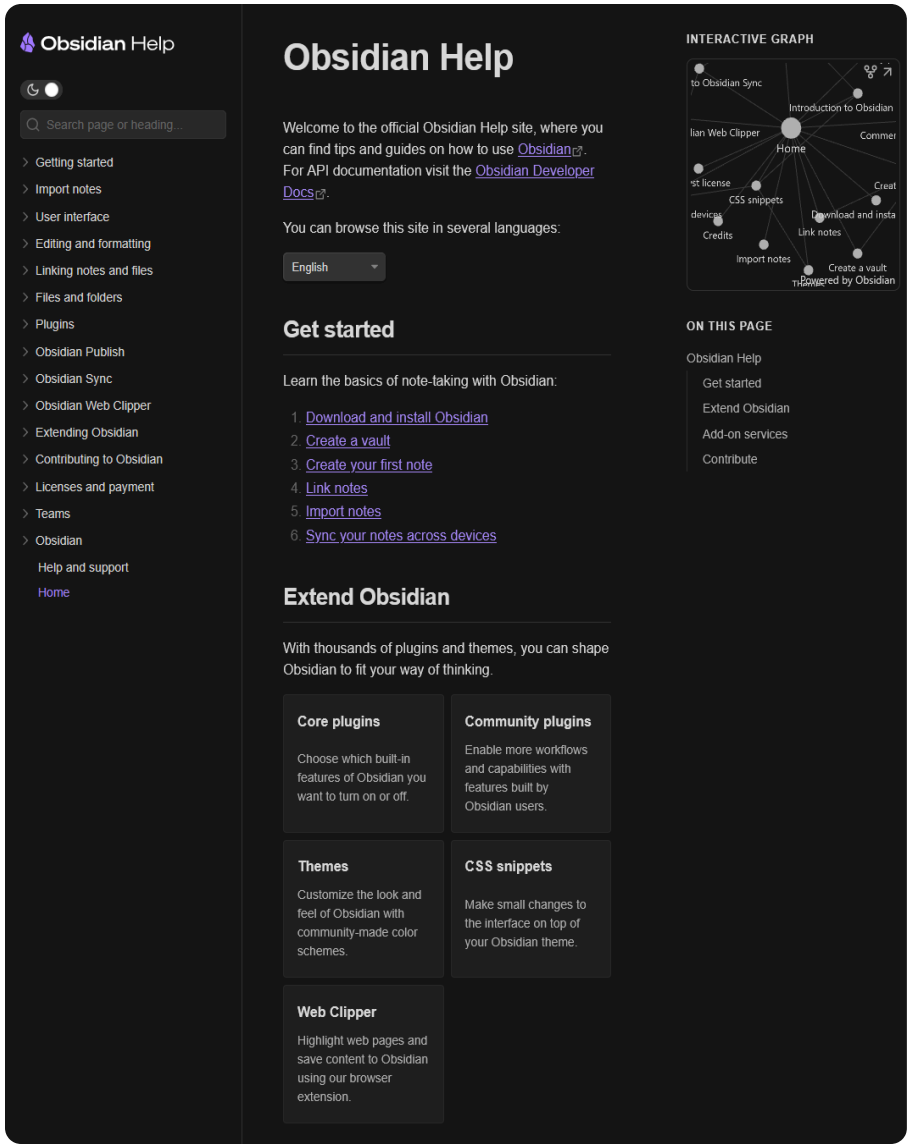
```
<ul>
  <li>Simple &amp;amp; Fast
    <ul>
      <li>No long HTML tags. Just a few symbols.</li>
    </ul>
  </li>
  <li>Readable Anywhere
    <ul>
      <li>Markdown files are plain text.</li>
    </ul>
  </li>
  <li>Free &amp;amp; Open Source
    <ul>
      <li>No costly tools like Word or Canva.</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



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	`*italicized text*`
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

CommonMark Syntaxes and their Equivalent in HTML

Element	Markdown Syntax	HTML Equivalent

Heading	`# H1`
	`## H2`
	`### H3`
		`<h1>H1</h1>`
		`<h2>H2</h2>`
		`<h3>H3</h3>`

Bold	`**bold text**`	`<strong>bold text</strong>`
Italic	`*italicized text*`	`<em>italicized text</em>`
Blockquote	`> blockquote`	`<blockquote>blockquote</blockquote>`

Ordered List	`1. First item`
	`2. Second item`
	`3. Third item`
		`<ol><li>First item</li>`
		`<li>Second item</li>`
		`<li>Third item</li></ol>`

continued in next slide...		

4. History & Flavors

CommonMark Syntaxes and their Equivalent in HTML

Element	Markdown Syntax	HTML Equivalent

Unordered List	`- First item`
	`- Second item`
	`- Third item`
		`<ul><li>First item</li>`
		`<li>Second item</li>`
		`<li>Third item</li></ul>`

Code	`` `code` ``	`<code>code</code>`
Horizontal Rule	`---`	`<hr>`
Link	`[title](https://www.youtube.com/)`	`<a href="https://www.youtube.com/">title</a>`
Image	`![alt text](image.jpg)`	`<img src="image.jpg" alt="alt text">`

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

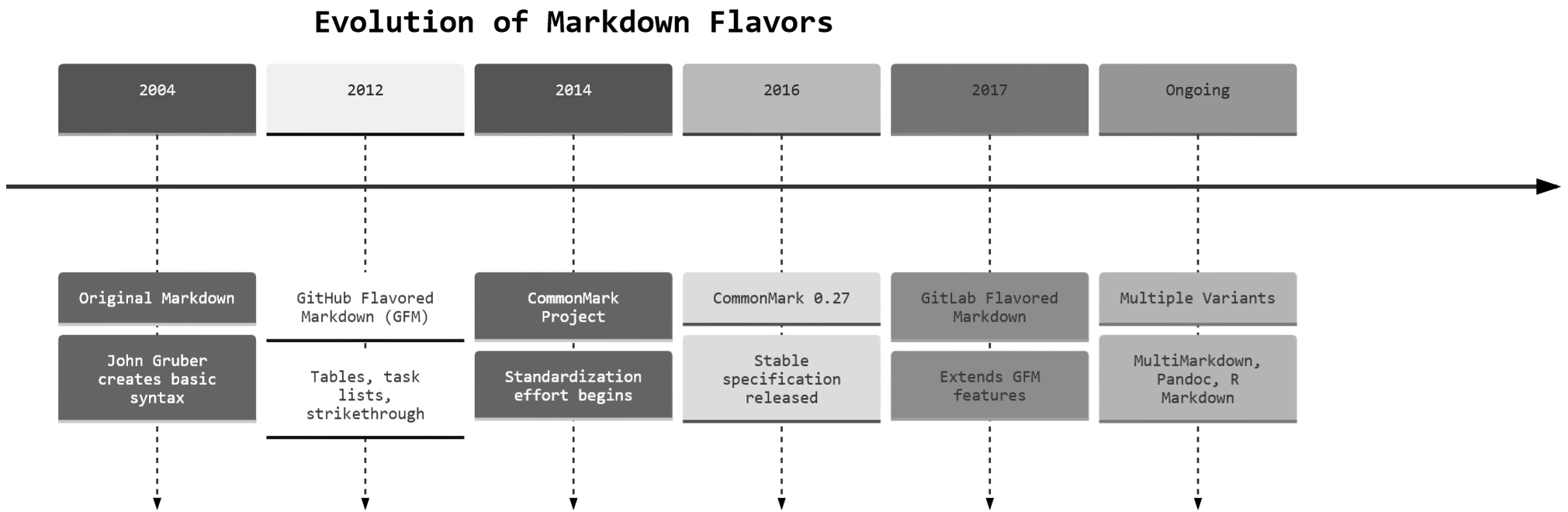
Examples of some flavors include:

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.

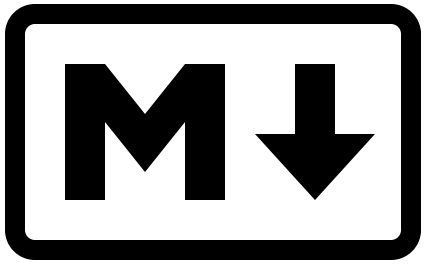
Example:

```
[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

```c
#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
```

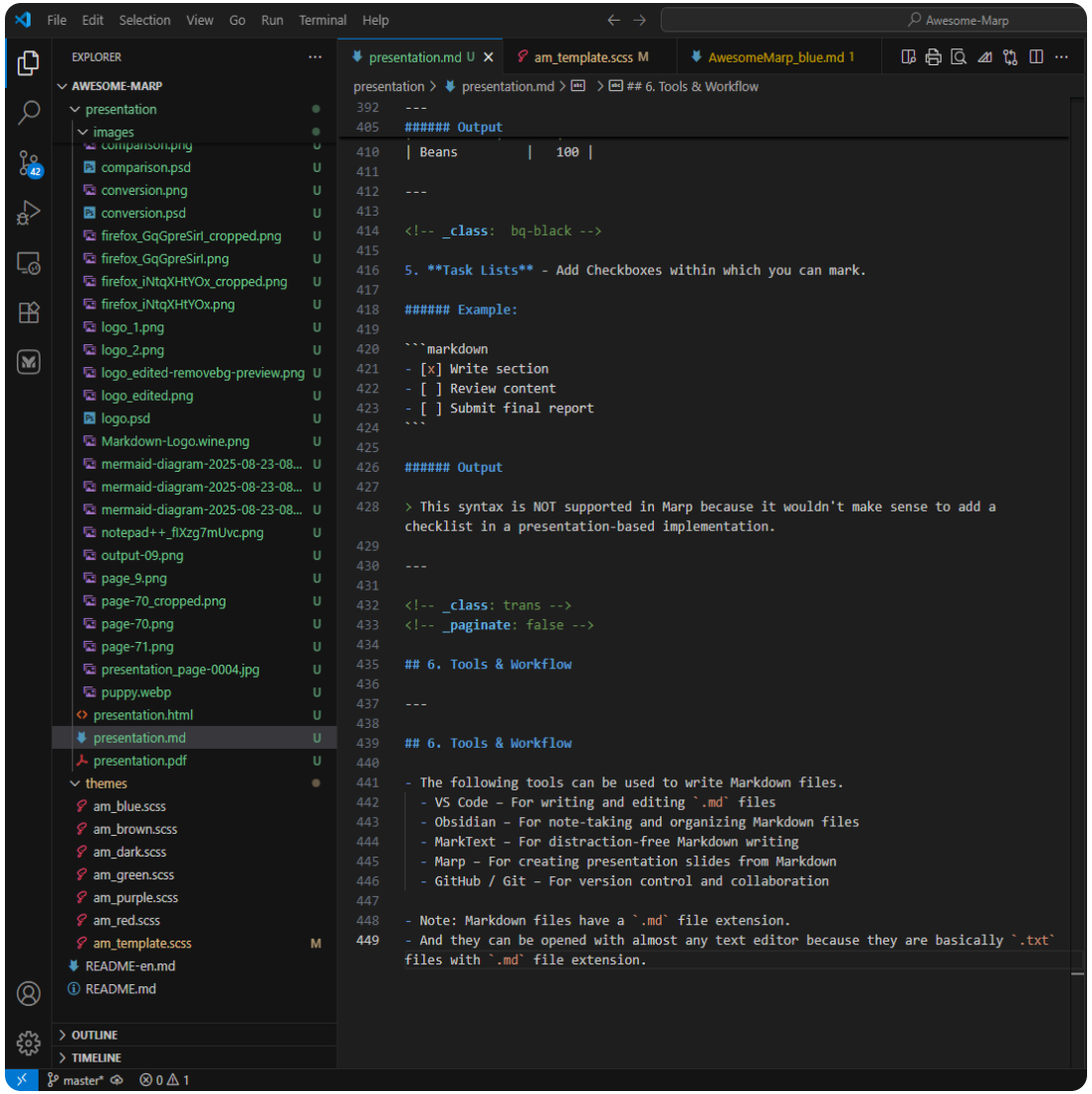
Output

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



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 but **pivotal**: 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
  - Works well with coding workflows
  - 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~

Link to the project repo