Marpee

*Ah, the magic of *checks notes* compiling slides*

Marpee: I am angry therefore I rant on markdown

Every machine I would move my slides to would display them differently. Even pdfs where not immune to this

Marpee: I am angry therefore I rant on markdown

Some unexpected requirements: missing fonts or mishaps in vectorial objects would also lead to incorrect display of slides and/or images within the slides

Marpee: I am angry therefore I rant on markdown

Portability of tools: powerpoint? which version? on what OS? and libreoffice? what about conversion to/from pptx? But google slides is universal! No, it's not

Marpee: I am angry therefore I rant on markdown

Limited expressivity: "Yes, slide software is extremely flexible, allows for so many things to do!" Said nobody ever. You know what's expressive? The **Web**. I want to be able to insert html/css/js in my presentation to leverage an incredibly large set of tools which exist on the one platform which is constantly looked at and interacted with. I want to put my slides on my website and people can interact with a plot, a slider, a sidenote, a whatever.

Marpee: I am angry therefore I rant on markdown

Plain text format is **the** portable format. Material in the slides is not "copy-pasted", rather it is part of the data ecosystem that I live with (thank you, Obsidian). Having a single language and format for notes, project management, slides, handbooks, etc. makes it so that any piece of information can be referenced without duplication, which is *essential* when your information has to be updated and/or replicated several times.

Imagine writing a handbook for documentation, and then having to create the slides from it. Imagine copy-pasting to power-point from word. Couldn't be me.

Marpee: I am angry therefore I rant on markdown

I bet you want to add some formula and use powerpoint's formulas. Or maybe a screenshot. Or maybe LateX. Thank you LateX, but you're too much, sorry.

Marpee: I am angry therefore I rant on markdown

Versioning systems are great. Online repositories of versioning systems are even better. "Oh, I need to modify this presentation, then give it to Annah to modify, then Adam has to shorten it too". Git, ladies. Github, wenches. Gitlab, even.

Marpee: getting started

Detailed instructions for compilation are available at the project's README.

Briefly:

1. Install `marp` (requires node): `npm install -g @marp-team/marp-cli` (Linux) or `brew install marp-cli` (brew, Mac)
2. Compile your slides in the desired format, e.g., html through a local (`marp --html slides.md`) or remote installation (`npx @marp-team/marp-cli@latest slides.md`)

That's it! Now you can compile your slides in `html`, `pdf`, `pptx`

Alternatively, a Docker image is also available.

If you are working on a tablet, and don't have any of these available, you should stop working.

Marpee and markdown

`markdown` is a template language thought for ease of writing, which includes syntax for

- **bold** through `**{text}**`, and *italic* through `*{text}*`
- headings through `#`: with `k` `#` you get a `h{k}` tags
- links through `[link text](https://whatever.com)`
- bullet list through `-`
- tables
- inline LateX with the `${latex}$` and `$${latex}$$` tags

As a markup language, `markdown` has several applications that allow to turn your `markdown` files into books (Leanpub, mdbook), notes (obsidian, logseq, boost note), websites (jekyll, quarto). Moreover, markdown allows you to embed native `html` within your `markdown` files.

Reference.

Marpee

Building on marp, which provides a compiler for `markdown` slides, `marpee` provides a set of utilities to ease construction of slide packs. This includes:

- import everything
- slide themes
- css customization
- vectorial assets

Moreover, being `markdown`, you can mixin `html` code!

Import everything

Fonts

Forget presentations ruined by missing fonts! You can simply import web fonts instead. This presentation mainly features the Arvo font, which was imported with

```
<!-- Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Arvo:ital,wght@0,400;0,700;1,400;1,700&display=swap" rel="stylesheet">
```

Even if this requires a network connection to correctly import, it's a far safer option than manually look for and configure fonts yourself.

Import everything

Themes

Rather than compile directly to pdf, you can go through an html step, thus compiling `markdown` to `html`, which you can then style with `CSS`! `marp` has a default theme (with more you can choose from), with `marpee` adding two more (which I added because I needed them, and no, I don't take theme requests):

- Unipi: an unofficial theme for University of Pisa
- xai: an unofficial theme for the XAI research project

Import everything

Objects and css

The default `marpee` template includes an import to Semantic UI a CSS-based UI components library.

Dividers

```
<div class="ui center aligned basic segment">
  One choice
  <div class="ui horizontal divider">
    Or
  </div>
  Another choice
</div>
```

One choice

OR

Another choice

Import everything

Objects and css

The default `marpee` template includes an import to Semantic UI a CSS-based UI components library.

Reveal animations

```
<div class="ui fade reveal">
  <div class="visible content">
    <img src="https://semantic-ui.com/images/wireframe/square-image.png" class="ui small image">
  </div>
  <div class="hidden content">
    <img src="https://semantic-ui.com/images/avatar/large/ade.jpg" class="ui small image">
  </div>
</div>
```

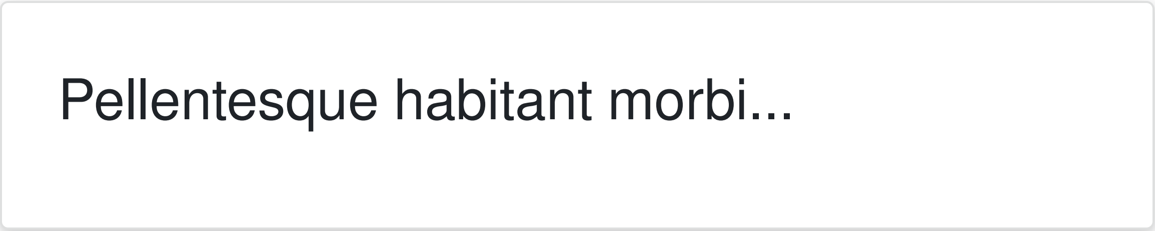
Import everything

Objects and css

The default `marpee` template includes an import to Semantic UI a CSS-based UI components library.

Segments

```html
<div class="ui raised segment">
  <p>Pellentesque habitant morbi...</p>
</div>
```



Pellentesque habitant morbi...

CSS

You can import the base theme I'm using for these slides with

```
<link
    rel="stylesheet"
    type="text/css",
    href="https://cdn.jsdelivr.net/gh/msetzu/marpee@latest/css/themes/base.css">
```

but remember that `marp` allows you to also insert `html` yourself. You can create your own theme, or apply different `CSS` to different files with a `style` tag. Even more refined, you can apply `CSS` to a single slide with the `style scoped` tag.

Stretching and pulling

`marpee` provides some convenience `CSS` classes for managing width and heights of objects (`/css/scaling/sizing.css`).

class	css
`wX`	`{width: X !important}`
`hX`	`{height: X !important}`

You can use this, e.g. to stretch and shrink columns by assigning classes `w60` and `w40` to a two-column layout to have the first column span 60% of the space, and the other one the 40% remaining. Supported are values `10, 20, 25, 33, 35, 40, etc.` You can find a complete list on sizing.css.

Highlighting

To remark some points inline, `marpee` provides the `highlight` class. Apply class `highlight` to a `span` tag to highlight it!

- not highlighted
- highlighted

Boxes

To create an box, e.g., to display an important theorem, use the `ui raised segment question` classes on a `div`. Optionally, give it a title with a `<p class="question"></p>`

Important theorem. Poggi e buoi fanno pari.

Templates

Columns

You can create two-columns slides with

```
<div class="ui two column doubling stackable grid container bottom">
<div class="column">
</div>
<div class="column">
</div>
</div>
```

Want more columns? Simply swap the `two` class in the outer `div` to `three`, `four`, etc., and add as many `<div class="column"></div>`

To render content inside `html` scopes as `markdown`, leave two blank lines between the `markdown` and `html`.

Columns

You can create two-columns slides

Column 1

Column 2

Columns and rows

You can also compose columns to create grids.

```
<!-- First row -->
<div class="ui two column doubling stackable grid container bottom" style="width: 50%;">
<div class="column">
</div>
<div class="column">
</div>
</div>

<!-- Second row -->
<div class="ui two column doubling stackable grid container bottom" style="width: 50%;">
<div class="column">
</div>
<div class="column">
</div>
</div>
```

Columns and rows

1, 1	1, 2
2,1	2,2

VS

One thing!

vs

The other thing!

Boxes

```
<div class="ui raised segment question">
<p class="question" style="display: inline;">Important theorem.</p>
Poggi e buoi fanno pari.
</div>
```

Boxes

Important theorem. Poggi e buoi fanno pari.

Footers

```
<!-- footer: "This is the footer of the page. Note that once set, it stays! If you want to remove it in the next slides insert an empty footer." -->
```

Much more

You can further customize this through marp directives.

You can find a list of presentations I've made with `marpee` in the gallery list.