

[mdpress](#) is a tool for making HTML presentations for Markdown. Here's how it works:

Quickstart (writing your first presentation!)

```
# Chicken Chicken Chicken
## By Chicken

---

# Chicken
- Chicken
- Chicken Chicken
```

Save that as `test.md` and then use `mdpress test.md` to convert it to a presentation. `mdpress` uses [Redcarpet](#), so it can parse whatever markdown you throw at it. We use the horizontal rule, `---`, to indicate the start of a new slide.

Getting Fancy: adding transformations

So far the slide transitions haven't been that exciting. Let's spice them up. Below the `---`, type:

```
= data-x="1000"
```

And re-compile with `mdpress test.md` (you might have to delete your old presentation directory first).

Aha! Not bad! We specify transformations with `= [list of transformations]` as the first line of whatever slide we want to apply the transformation to.

We could have written

```
= data-x="1000" data-scale="2"
```

To make it scale as well as swipe.

Possible Attributes

- `data-x`, `data-y`, `data-z`: positioning
- `data-rotate-x`, `data-rotate-y`, `data-rotate-z`, `data-rotate: rotation` (`data-rotate` and `data-rotate-z` are exactly the same)
- `data-scale: scale`
- `id`: used as the slide link. For example, if you use `id=intro`, you can link to that slide using `index.html#/intro`.

See [impress.js](https://impress.js.org) for a broader description of these.

Adding this syntax to Markdown feels like kind of a hack, but it's concise. I'm open to suggestions for something better.

Auto-update

Earlier, our workflow involved having to delete the old presentation and recompile every time we made an edit. The better choice is to use the automatic mode:

```
mdpress -a [filename.md]
```

Now, `mdpress` will keep running and check for updates to your markdown file. It will automatically recompile the presentation for you, so you can seamlessly edit your markdown and view your changes instantly.

Stylesheets

`mdpress` comes with a default stylesheet, plus a few more. See the full list of available stylesheets with

```
mdpress -l
```

Or specify a stylesheet to use with

```
mdpress -s [stylesheet]
```

If you make your own stylesheets, send me a pull request and I'll add them to the list! See below.

You can also create your own local theme:

1. Use the `lib/impress_css/` files as a model to work from.
2. Create a folder called `themes` in the directory where your markdown file is. Inside it put the `[yourtheme].css` and `[yourtheme].html` files
3. Call `mdpress -s [yourtheme]` and `mdpress` will look for the two theme files in `themes/` matching that name.

Running a presentation

Someone emailed you a `mdpress` presentation. Now you want to **just run it**? Sure:

```
mdpress -r [filename.md]
```

That will automatically compile the presentation and open it in a browser window.

Latex Support

mdpress now has Latex support through [MathJax](#). To build your presentation with latex support, use the `--latex` flag.

Put your latex code between a pair of `$$` 's, like this:

```
# Latex!

$$e^{i\pi} = -1$$

```

Note: you might run into conflicts between markdown and latex. For example, `_` means emphasize in markdown. So if you want subscripts, escape `_`:

```
 $a_x$  # wrong, x will be italic
 $a\_x$  # right, x will be a subscript
```

Metadata

Metadata can be set via YAML-Frontmatter. Have a look at the examples for further information. For example, if you want to set a title, put this at the top of the markdown file:

```
---
title: All about chicken
---
```

[See the project on Github.](#)