

📸 Page Elements: Writing content with Markdown, LaTeX, and Shortcodes

Rich content can be written in Wowchemy using **Markdown**, **LaTeX math**, and **Shortcodes**. This article gives an overview of the most common formatting options, including features that are exclusive to Wowchemy.

i *Shortcodes* are plugins which are bundled with Wowchemy or inherited from [Hugo](#). Additionally, **HTML** may be written in Markdown documents for advanced formatting.

Sub-headings

After the page title, heading 1, we can format sub-headings in the page body:

```
## Heading 2
### Heading 3
#### Heading 4
```

Emphasis

Bold, Italic, Strikethrough

```
Italics with _underscores_.

Bold with **asterisks**.

Combined emphasis with **asterisks and _underscores_**.

Strikethrough with ~~two tildes~~.
```

Italics with *underscores*.

Bold with **asterisks**.

Combined emphasis with **asterisks and underscores**.

Strikethrough with ~~two tildes~~.

Text color

Add the [HTML color](#) attribute in the form `{style="color: red"}` to the line *after* a Markdown block.

For example, to color a paragraph red:

```
Red colored text
{style="color: red"}
```

Blockquote

```
> This is a blockquote.
```

This is a blockquote.

Highlight quote

```
This is a {{< h1 >}}highlighted quote{{< /h1 >}}.
```

 This is a highlighted quote.

Lists

Ordered

```
1. First item
  1. A sub-item
2. Another item
```

1. First item

1. A sub-item

2. Another item

Unordered

```
- First item
  - A sub-item
- Another item
```

• First item

◦ A sub-item

• Another item

Todo

Todo lists can be written in Wowchemy by using the standard Markdown syntax:

```
- [x] Write math example
  - [x] Write diagram example
- [ ] Do something else
```

renders as

☒ Write math example

☒ Write diagram example

☐ Do something else

Toggle lists

Add a toggle list to a page to reveal text, such as an answer to a question, after a toggle button is clicked.

Useful for FAQs, spoilers, or hiding answers when teaching online courses.

```
{{< spoiler text="Click to view the spoiler" >}}
You found me!
{{< /spoiler >}}
```

renders as

► Click to view the spoiler

Links

Link to a page

```
[I'm an external link](https://www.google.com)
[A post]({{< relref "/post/my-page-name" >}})
[A publication]({{< relref "/publication/my-page-name" >}})
[A project]({{< relref "/project/my-page-name" >}})
[A relative link from one post to another post]({{< relref "../my-page-name" >}})
[Scroll down to a page section with heading *Hi*](#hi)
```

You can also [create a button link in the page header](#) of any page except Widget Pages (the homepage).

Link to a file



You can [create a button link to a file in the page header](#) of any page except Widget Pages (the homepage).

Otherwise, to link to a file, such as a PDF, in the **body** of your content, place the file in your `static/uploads/` folder and then link to it using the following form:

```
{{% staticref "uploads/cv.pdf" "newtab" %}}Download my CV{{% /staticref %}}
```

The optional `"newtab"` argument for `staticref` will cause the link to be opened in a new tab.

Cite

To cite a page or publication, you can use the `cite` shortcode, referencing a folder and page name that you created:

```
{{< cite page="/publication/preprint" view="4" >}}
```

where `view` corresponds to one of the available listing views used throughout Wowchemy:

- 1. Stream
- 2. Compact
- 3. Card
- 4. Traditional academic citation, configured by the `citation_style` setting in `params.yaml`

If you don't specify a view, it will default to the `compact` view.

The cite shortcode requires Wowchemy v5+.

Table of Contents

A table of contents can help users navigate long pages.

Paste the `{{< toc >}}` shortcode wherever you would like a table of contents to appear in your page.

You may also wish to consider using the [book layout](#) which displays a built-in table of contents in a right sidebar on large screens.

When using the `book` layout, you may still wish to add an in-page table of contents which only appears when the right sidebar is hidden. You can do this with `{{< toc hide_on="x1" >}}`.

Footnotes

```
I have more [^1] to say.  
  
[^1]: Footnote example.
```

I have more¹ to say.

Figures

To cross-reference a figure, provide it with an ID, for example: `{{< figure src="image.jpg" id="wowchemy" >}}`.

The figure can now be cross-referenced with a link in the form `[A Figure](#figure-wowchemy)`.

Tags and Categories

Use `{{< list_tags >}}` to provide a list of linked tags or `{{< list_categories >}}` to provide a list of linked categories.



Call-to-Action Buttons

Create buttons and call-to-actions (CTA) with the *CTA* shortcode.

Example CTA button:

```
{{< cta cta_text="Do something" cta_link="/" cta_new_tab="false" >}}
```

Example CTA button with a link to an alternate action:

```
{{< cta cta_text="Do something" cta_link="/" cta_new_tab="false" cta_alt_text="Alternative action" cta_alt_link="/alt" >}}
```

Mention a user

To mention someone, type `{{% mention "username" %}}` where `username` corresponds to a user account in Wowchemy.

List child pages


To list child pages, such as to list book sections from a book chapter, use `{{< list_children >}}`.

Images

Single image

Images may be added to a page by either placing them in your `assets/media/` media library or in your [page's folder](#), and then referencing them using one of the following notations.

Figures are **themeable** according to light/dark mode and can also be **cross-referenced**.



Prior to Wowchemy v5.1, the media library was located at `static/img/` or `static/media/`.

A figure from your page's folder or your `assets/media/` media library:

```
![[screen reader text]](image.jpg "caption")
```

For more customizable figures, we can use the built-in Figure shortcode rather than the Markdown figure syntax above.

A numbered figure with a caption:

```
{{< figure src="image.jpg" caption="A caption" numbered="true" >}}
```

Using light/dark mode? Did you know that you can create dynamically themed light/dark images?

Dynamically theme images according to the user's light/dark preference

To **dynamically theme images** according to the user's light or dark theme

- `{{< figure src="image.jpg" caption="test" theme="light" >}}` inverts image when browsing in dark mode
- `{{< figure src="image.jpg" caption="test" theme="dark" >}}` inverts image when browsing in light mode

Alternatively, paste a **themeable inline SVG image** in the form `<svg fill="currentColor">...</svg>` directly into your Markdown.

To add an image gallery to a page:

- 1. Create a photo album folder in `assets/media/albums/`
 - Or, **prior to v5.3**, create the album folder within your page’s folder instead
 - Album names must be **lowercase** due to a [Hugo bug](#) – consider up-voting the bug with a 👍
- 2. Add images to your album folder
- 3. Paste `{{< gallery album="<ALBUM-FOLDER>" >}}` where you would like the gallery to appear in the page content, changing the album parameter to match the name of your album folder

How can I change the size or shape of gallery images?

Simply add `resize_options` when you call the gallery shortcode. This option lets you control the shape/size of gallery images using the standard [Hugo resize options](#). For example:

```
{{< gallery album="<ALBUM-FOLDER>" resize_options="250x250" >}}
```

How can I add captions to gallery images?

Optionally, to add captions for your images, add the following instances to the end of your page’s front matter:

```
gallery_item:
- album: <ALBUM FOLDER>
  image: <IMAGE 1 NAME>.jpg
  caption: Write your image 1 caption here
- album: <ALBUM FOLDER>
  image: <IMAGE 2 NAME>.jpg
  caption: Write your image 2 caption here
```

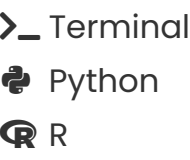
Inline image

Wowchemy enables you to use a wide range of [inline images \(aka icons\)](#) from *Font Awesome* and *Academicons* in addition to [emojis](#). There is also an [issue](#) open to add support for custom images.

Here are some examples using the *icon* shortcode to render icons:

```
{{< icon name="terminal" pack="fas" >}} Terminal
{{< icon name="python" pack="fab" >}} Python
{{< icon name="r-project" pack="fab" >}} R
```

renders as



Optionally, left and right padding can be added to an icon using the `padding_left="3"` and `padding_right="3"` options, respectively.

Emojis

An emoji can be inserted using either an emoji keyboard or by typing the name of the emoji.

Don’t have an emoji keyboard?

Copy and paste emoji names from the [Emoji cheat sheet](#). The following serves as an example, but remember to remove the added spaces between each emoji name and surrounding colons:

```
I : heart : Wowchemy : smile :
```



Twitter

Embed a feed

[Design your timeline on Twitter](#) and paste it into a Markdown page.

Embed a single tweet

To include a single tweet, pass the tweet's ID from the tweet's URL as parameter to the shortcode:

```
{{< tweet 666616452582129664 >}}
```

Callouts

Wowchemy supports a Markdown extension for callouts, also referred to as *alerts* or *asides*.

Callouts are a useful feature to **draw attention to important or related content** such as notes, hints, or warnings in your articles. They are especially handy when writing educational tutorial-style articles or documentation.


A callout can be created by using the *Callout* shortcode below. (For older Wowchemy versions prior to v5, replace *callout* in the examples below with *alert*.)

Wowchemy comes built-in with a few different styles of callouts.

The paragraph will render as a callout with the default *note* style:

```
{{% callout note %}}
A Markdown callout is useful for displaying notices, hints, or definitions to your readers.
{{% /callout %}}
```

This will display the following *note* block:

 A Markdown callout is useful for displaying notices, hints, or definitions to your readers.

Alternatively, a warning can be displayed to the reader using the *warning* option:

```
{{% callout warning %}}
Here's some important information...
{{% /callout %}}
```

This will display the following *warning* notice to the reader:

 Here's some important information...

Audio

You can add a podcast or music to a page by placing the MP3 file in the page's folder and then referencing the audio file using the *audio* shortcode:

```
{{< audio src="markvard.mp3" >}}
```

Videos

The following kinds of video may be added to a page.

Local video file



Videos may be added to a page by either placing them in your `static/media/` media library or in your `page's folder`, and then referencing them using one of the following notations.

 Prior to Wowchemy v5, the media library is located at `static/img/`.

A video from your `static/media/` media library:

```
{{< video library="true" src="my_video.mp4" controls="yes" >}}
```

A video within a `page's folder` (e.g. `content/post/hello/`):

```
{{< video src="my_video.mp4" controls="yes" >}}
```

Youtube:

```
{{< youtube w7Ft2ymGmfc >}}
```

Vimeo:

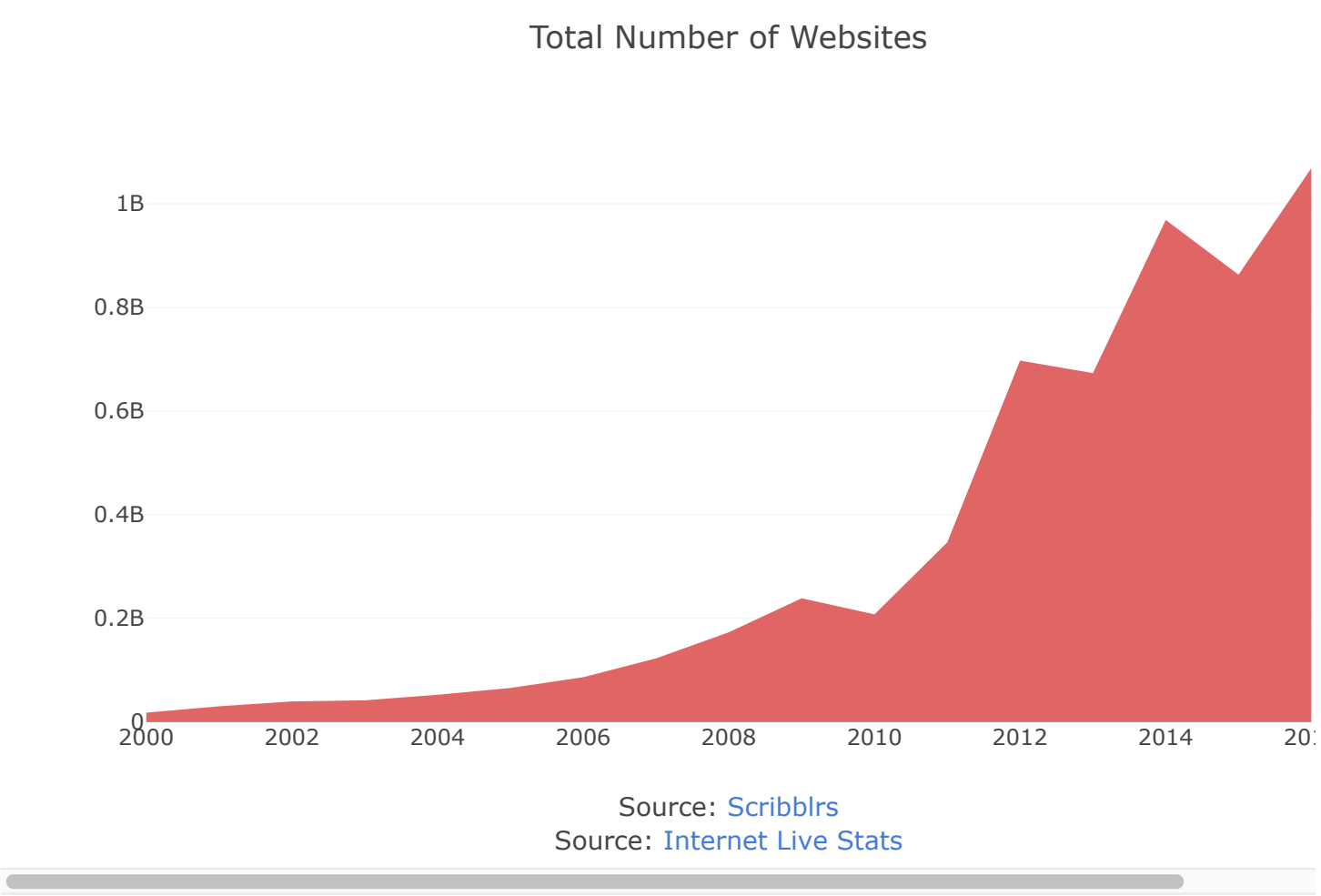
```
{{< vimeo 146022717 >}}
```

Charts

Wowchemy supports the popular `Plotly` chart format.

Save your Plotly JSON in your page folder, for example `chart.json`, and then add the `{{< chart data="chart" >}}` shortcode where you would like the chart to appear.

Demo:



You might also find the `Plotly JSON Editor` useful.

The `chart` shortcode requires Wowchemy v5+.

Embed Documents

The following kinds of document may be embedded into a page.

To embed **Google Documents** (e.g. slide deck), click *File > Publish to web > Embed* in Google Docs and copy the URL within the displayed `src="..."` attribute. Then paste the URL in the form:



```
{{< gdocs src="https://docs.google.com/..." >}}
```

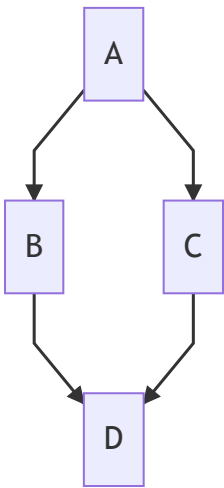
Diagrams

Wowchemy supports a Markdown extension for diagrams. You can enable this feature by toggling the `diagram` option in your `config/_default/params.yaml` file or by adding `diagram: true` to your page front matter. Then insert your [Mermaid diagram syntax](#) within a `mermaid` code block as seen below and that's it.

An example **flowchart**:

```
```mermaid
graph TD;
 A-->B;
 A-->C;
 B-->D;
 C-->D;
```
```

renders as

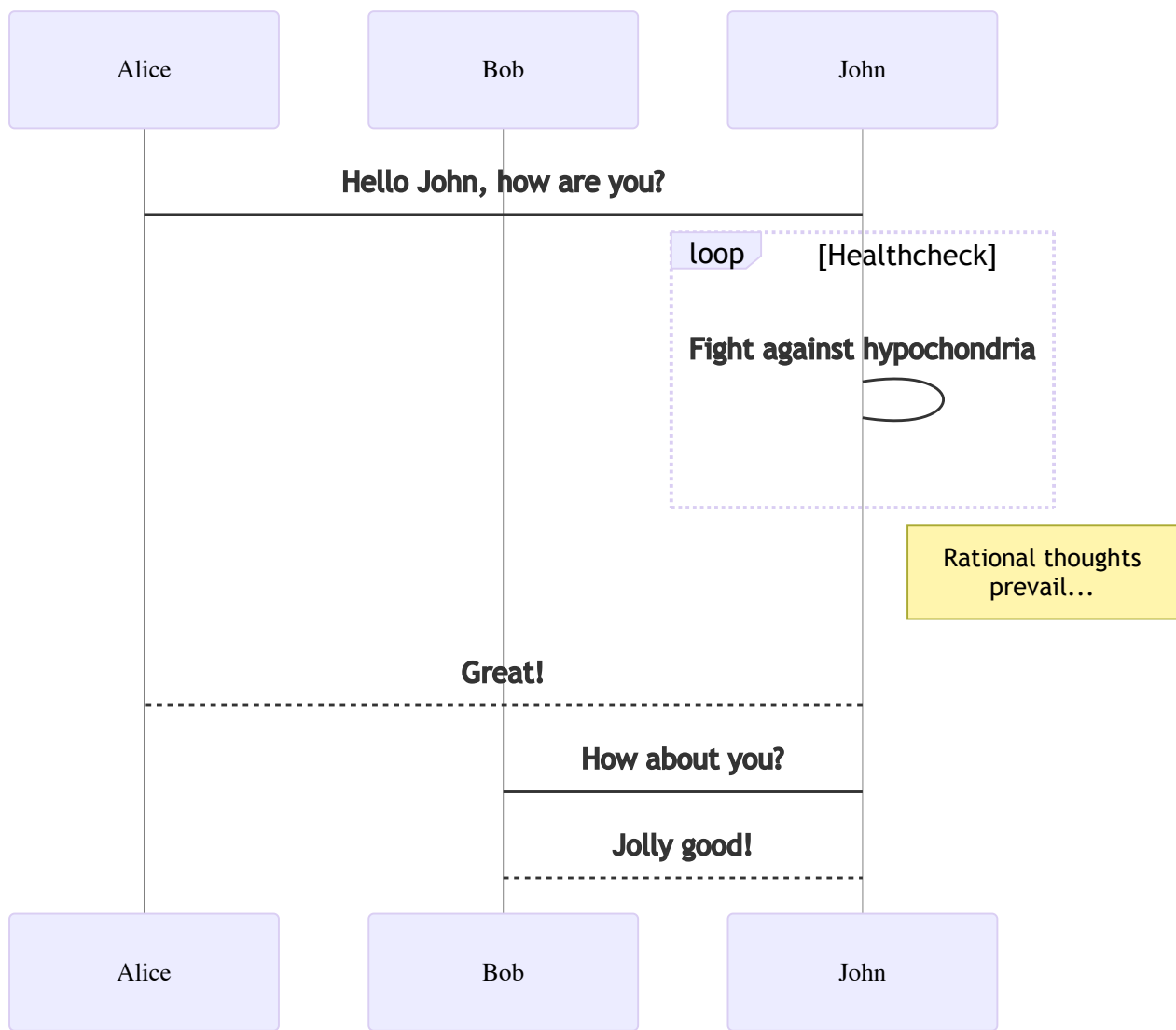


An example **sequence diagram**:

```
```mermaid
sequenceDiagram
 participant Alice
 participant Bob
 Alice->>John: Hello John, how are you?
 loop Healthcheck
 John->>John: Fight against hypochondria
 end
 Note right of John: Rational thoughts
prevail...
 John-->>Alice: Great!
 John->>Bob: How about you?
 Bob-->>John: Jolly good!
```
```

renders as

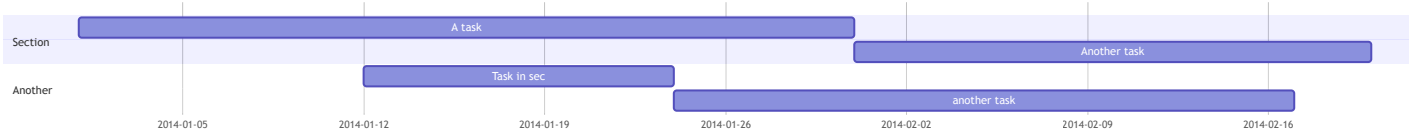




An example **Gantt diagram**:

```
```mermaid
gantt
 dateFormat YYYY-MM-DD
 section Section
 A task :a1, 2014-01-01, 30d
 Another task :after a1 , 20d
 section Another
 Task in sec :2014-01-12 , 12d
 another task : 24d
    ```
```

renders as



Advanced diagrams

More advanced diagrams can be created in the open source [draw.io](#) editor. The editor has support for almost any type of diagram, from simple to complex. A diagram can be easily embedded in Wowchemy by choosing **File > Embed > SVG** in the [draw.io](#) editor and pasting the generated code into your page.

Alternatively, a diagram can be exported as an [image](#) from any drawing software, or a [document/slide](#) containing a diagram can be embedded.

Code highlighting

Pass the *language* of the code, such as `python`, as a parameter after three backticks:



```
```python
Example of code highlighting
input_string_var = input("Enter some data: ")
print("You entered: {}".format(input_string_var))
```
```

Result:

```
# Example of code highlighting
input_string_var = input("Enter some data: ")
print("You entered: {}".format(input_string_var))
```

Highlighting options

The Wowchemy integrates with [HighlightJS](#) for source code highlighting, the powerful technology behind some of the most popular tech sites such as StackOverflow.

Configuration options are supported that allow finer-grained control over HighlightJS.

The following table lists the supported options for configuring highlight.js, along with their expected type and a short description. A “yes” in the **config.yaml** column means the value can be set globally in `params.yaml`, and a “yes” in the **front matter** column means that the value can be set locally in a particular page’s front matter.

| option | type | description | params.yaml | front matter |
|----------------------------------|---------|-----------------------------|-------------|--------------|
| <code>highlight</code> | boolean | enable/disable highlighting | yes | yes |
| <code>highlight_languages</code> | slice | choose additional languages | yes | no |
| <code>highlight_style</code> | string | choose a highlighting style | yes | no |

Option `highlight`

The `highlight` option allows enabling or disabling the inclusion of highlight.js, either globally or for a particular page. If the option is unset, it has the same effect as if you had specified `highlight = true`. That is, every page will be parsed by HighlightJS.

If you’d like to only include highlight.js files on pages that actually require source code highlighting, you can set `highlight = false` in `params.yaml`, and then override it by setting `highlight: true` in the front matter of any pages that require source code highlighting. Conversely, you could enable highlighting globally, and disable it locally for pages that do not require it.

Option `highlight_languages`

The `highlight_languages` option allows you to specify additional languages that are supported by HighlightJS, but are not considered “common” and therefore are not supported by default. For example, if you want source code highlighting for R and Python in all pages, set `highlight_languages = ["r", "python"]` in `params.yaml`.

If the `highlight_languages` option is set, then the corresponding javascript files will be served from the [cdnjs server](#). To see a list of available languages, visit the [cdnjs page](#) and search for links with the word “languages”.

The `highlight_languages` option provides an easy and convenient way to include support for additional languages to be severed from a CDN. If serving unmodified files from cdnjs doesn’t meet your needs, you can include javascript files for additional language support via one of the methods described in the [customization guide](#).

Option `highlight_style`

The `highlight_style` option allows you to select an alternate css style for highlighted code. For example, if you wanted to use the solarized-dark style, you could set `highlight_style = "solarized-dark"` in `params.yaml`.




If the `highlight_style` option is unset, the default is to use the file `/css/highlight.min.css`, either the one provided by the Wowchemy theme, or else the one in your local `static` directory. The `/css/highlight.min.css` file provided by Wowchemy is equivalent to the `github` style from highlight.js.

If the `highlight_style` option *is* set, then `/css/highlight.min.css` is ignored, and the corresponding css file will be served from the [cdnjs server](#). To see a list of available styles, visit the [cdnjs page](#) and search for links with the word “styles”.

See the [highlight.js demo page](#) for examples of available styles.



Not all styles listed on the [highlight.js demo page](#) are available from the [cdnjs server](#). If you want to use a style that is not served by cdnjs, just leave `highlight_style` unset, and place the corresponding css file in `/static/css/highlight.min.css`.



If you don't want to change the default style that ships with Wowchemy but you do want the style file served from the [cdnjs server](#), set `highlight_style = "github"` in `params.yaml`.

The `highlight_style` option is only recognized when set in `params.yaml`. Setting `highlight_style` in your page's preamble has no effect.

Jupyter Notebook

[View the guide to blogging with Jupyter Notebooks.](#)

GitHub gist

```
{{< gist USERNAME GIST-ID >}}
```

L^AT_EX math

Wowchemy supports a Markdown extension for *L^AT_EX* math. You can enable this feature by toggling the `math` option in your `config/_default/params.yaml` file.

To render *inline* or *block* math, wrap your LaTeX math with `$...$` or `$$...$$`, respectively.

Example **math block**:

```
$$\gamma_{n} = \frac{
\left | \left (\mathbf{x}_{n} - \mathbf{x}_{n-1} \right )^T
\left [\nabla F (\mathbf{x}_{n}) - \nabla F (\mathbf{x}_{n-1}) \right ] \right |}
{\left ||\nabla F(\mathbf{x}_{n}) - \nabla F(\mathbf{x}_{n-1}) \right ||^2}$$
```

renders as

$$\gamma_n = \frac{\left | \left (\mathbf{x}_n - \mathbf{x}_{n-1} \right)^T \left [\nabla F (\mathbf{x}_n) - \nabla F (\mathbf{x}_{n-1}) \right] \right |}{\left ||\nabla F(\mathbf{x}_n) - \nabla F(\mathbf{x}_{n-1}) \right ||^2}$$

Example **inline math** `$\nabla F(\mathbf{x}_n)$` renders as $\nabla F(\mathbf{x}_n)$.

Example **multi-line math** using the `\\` math linebreak:

```
$$f(k;p_0^{*}) = \begin{cases}p_0^{*} & \text{if } k=1, \\
1-p_0^{*} & \text{if } k=0.\end{cases}$$
```

renders as

$$f(k;p_0^{\ast}) = \begin{cases}p_0^{\ast} & \text{if } k=1, \\ 1-p_0^{\ast} & \text{if } k=0.\end{cases}$$

Math in front matter



Math is primarily intended for use within the page body, but can also be added to page front matter variables such as a publication abstract with some additional checks.

As Hugo and Wowchemy can attempt to parse YAML, Markdown, and LaTeX content in the front matter, Markdown special characters need to be escaped in any math within the front matter fields by using a backslash to prevent the math being parsed as Markdown. The following tips may help:

- escape each LaTeX backslash (\) with an extra backslash, yielding \\
- escape each LaTeX underscore (_) with two backslashes, yielding _

Hence, `abstract: "${O(d_{\max})}$"` becomes `abstract: "${O(d_\\max})}$"`.

Subscripts & superscripts

Subscripts and superscripts can be written direct from a (unicode) keyboard or [pasted from a reference](#).

Alternatively, you can enable `math` in `params.yaml` and use the Latex notation, for example $\$^2\$$.

Tables

Markdown Table

For quick, simple tables, use the [Markdown table generator](#) and paste the resulting Markdown into your page.

CSV Table

For larger tables, save your spreadsheet as a CSV file in your page’s folder and then render it by adding the `Table` shortcode to your page:

```
{{< table path="results.csv" header="true" caption="Table 1: My results" >}}
```

Requires Wowchemy v5.4

Google Sheets

See the [Embed Documents](#) section.

1. Footnote example. ↩

PREVIOUS

[Page Features](#)

NEXT

[Create a landing page](#)

Last updated on 19 Jan, 2022

