

### What's on this Page

[List of content formats](#)[External Helpers](#)[Learn Markdown](#)

## CONTENT MANAGEMENT

# Content Formats

Both HTML and Markdown are supported content formats.

You can put any file type into your /content directories, but Hugo uses the markup front matter value if set or the file extension (see Markup identifiers in the table below) to determine if the markup needs to be processed, e.g.:

- Markdown converted to HTML
- [Shortcodes](#) processed
- Layout applied

## List of content formats

The current list of content formats in Hugo:

Name	Markup identifiers	Comment
Goldmark	md, markdown, goldmark	Note that you can set the default handler of md and markdown to something else, see <a href="#">Configure Markup</a> . <span>New in v0.60.0</span>
Blackfriday	blackfriday	Blackfriday will eventually be deprecated.
MMark	mmark	Mmark is deprecated and will be removed in a future release.

Name	Markup identifiers	Comment
Emacs Org-Mode	org	See <a href="#">go-org</a> .
AsciiDoc	asciidocext, adoc, ad	Needs <a href="#">Asciidoctor</a> installed.
RST	rst	Needs <a href="#">RST</a> installed.
Pandoc	pandoc, pdc	Needs <a href="#">Pandoc</a> installed.
HTML	html, htm	To be treated as a content file, with layout, shortcodes etc., it must have front matter. If not, it will be copied as-is.

The markup identifier is fetched from either the markup variable in front matter or from the file extension. For markup-related configuration, see [Configure Markup](#).

## External Helpers [↗](#)

Some of the formats in the table above need external helpers installed on your PC. For example, for AsciiDoc files, Hugo will try to call the `asciidoctor` command. This means that you will have to install the associated tool on your machine to be able to use these formats.

Hugo passes reasonable default arguments to these external helpers by default:

- `asciidoctor`: `--no-header-footer -`
- `rst2html`: `--leave-comments --initial-header-level=2`
- `pandoc`: `--mathjax`

Because additional formats are external commands, generation performance will rely heavily on the performance of the external tool you are using. As this feature is still in its infancy, feedback is welcome.

## External Helper AsciiDoc [↗](#)

[AsciiDoc](#) implementation EOLs in Jan 2020 and is no longer supported. AsciiDoc development is being continued under [Asciidoctor](#). The format AsciiDoc remains of course. Please continue with the implementation Asciidoctor.

## External Helper Asciidoctor [↗](#)

The Asciidoctor community offers a wide set of tools for the AsciiDoc format that can be installed additionally to Hugo. [See the Asciidoctor docs for installation instructions](#). Make sure that also all optional extensions like `asciidoctor-diagram` or `asciidoctor-html5s` are installed if required.

External `asciidoctor` command requires Hugo rendering to disk to a specific destination directory. It is required to run Hugo with the command option `--destination`.

Some [Asciidoctor](#) parameters can be customized in Hugo:

Parameter	Comment
<code>backend</code>	Don't change this unless you know what you are doing.
<code>doctype</code>	Currently, the only document type supported in Hugo is <code>article</code> .
<code>extensions</code>	Possible extensions are <code>asciidoctor-html5s</code> , <code>asciidoctor-bibtex</code> , <code>asciidoctor-diagram</code> , <code>asciidoctor-interdoc-reftext</code> , <code>asciidoctor-katex</code> , <code>asciidoctor-latex</code> , <code>asciidoctor-mathematical</code> , <code>asciidoctor-question</code> , <code>asciidoctor-rouge</code> .
<code>attributes</code>	Variables to be referenced in your AsciiDoc file. This is a list of variable name/value maps. See <a href="#">Asciidoctor's attributes</a> .
<code>noHeaderOrFooter</code>	Output an embeddable document, which excludes the header, the footer, and everything outside the body of the document. Don't change this unless you know what you are doing.
<code>safeMode</code>	Safe mode level <code>unsafe</code> , <code>safe</code> , <code>server</code> or <code>secure</code> . Don't change this unless you know what you are doing.
<code>sectionNumbers</code>	Auto-number section titles.
<code>verbose</code>	Verbosely print processing information and configuration file checks to <code>stderr</code> .
<code>trace</code>	Include backtrace information on errors.
<code>failureLevel</code>	The minimum logging level that triggers a non-zero exit code (failure).

Hugo provides additional settings that don't map directly to Asciidoctor's CLI options:

### **workingFolderCurrent**

Sets the working directory to be the same as that of the AsciiDoc file being processed, so that [include](#) will work with relative paths. This setting uses the `asciidoctor` cli parameter `--base-dir` and attribute `outdir=`. For rendering diagrams with [asciidoctor-diagram](#), `workingFolderCurrent` must be set to `true`.

### **preserveTOC**

By default, Hugo removes the table of contents generated by Asciidoctor and provides it through the

built-in variable [.TableOfContents](#) to enable further customization and better integration with the various Hugo themes. This option can be set to true to preserve Asciidoctor's TOC in the generated page.

Below are all the AsciiDoc related settings in Hugo with their default values:

config. **yaml** toml json

```
markup:
  asciidocExt:
    attributes: {}
    backend: html5
    extensions: []
    failureLevel: fatal
    noHeaderOrFooter: true
    preserveTOC: false
    safeMode: unsafe
    sectionNumbers: false
    trace: false
    verbose: false
    workingFolderCurrent: false
```

Notice that for security concerns only extensions that do not have path separators (either \, / or .) are allowed. That means that extensions can only be invoked if they are in one's ruby's \$LOAD\_PATH (ie. most likely, the extension has been installed by the user). Any extension declared relative to the website's path will not be accepted.

Example of how to set extensions and attributes:

```
[markup.asciidocExt]
extensions = ["asciidoctor-html5s", "asciidoctor-diagram"]
workingFolderCurrent = true
[markup.asciidocExt.attributes]
my-base-url = "https://example.com/"
my-attribute-name = "my value"
```

In a complex Asciidoctor environment it is sometimes helpful to debug the exact call to your external helper with all parameters. Run Hugo with -v. You will get an output like

```
INFO 2019/12/22 09:08:48 Rendering book-as-pdf.adoc with C:\Ruby26-x64\bin\asciidoctor.bat using as
```

# Learn Markdown

Markdown syntax is simple enough to learn in a single sitting. The following are excellent resources to get you up and running:

- [Daring Fireball: Markdown](#), John Gruber (Creator of Markdown)
- [Markdown Cheatsheet](#), Adam Pritchard
- [Markdown Tutorial \(Interactive\)](#), Garen Torikian
- [The Markdown Guide](#), Matt Cone

## See Also

- [Shortcodes](#)
- [markdownify](#)
- [.RenderString](#)
- [anchorize](#)

“Content Formats” was last updated: August 21, 2021: [highlight: Remove some pygments references \(8fcf2c55d\)](#)

IMPROVE THIS PAGE

By the [Hugo Authors](#)

[File an Issue](#) [Get Help](#) [Discuss Source Code](#)

[@GoHugoIO](#) [@spf13](#) [@bepsays](#)



## Hugo Sponsors



linode

The Hugo logos are copyright © Steve Francia 2013–2021.

The Hugo Gopher is based on an original work by Renée French.

Menu

Docs Menu