

[DOCS](#)[NEWS](#)[HELP](#)

Directory structure

[✎ Improve this page](#)

Jekyll is, at its core, a text transformation engine. The concept behind the system is this: you give it text written in your favorite markup language, be that Markdown, Textile, or just plain HTML, and it churns that through a layout or series of layout files. Throughout that process you can tweak how you want the site URLs to look, what data gets displayed in the layout, and more. This is all done through editing text files, and the static web site is the final product.

A basic Jekyll site usually looks something like this:

GETTING STARTED

- [Welcome](#)
- [Quick-start guide](#)
- [Installation](#)
- [Basic Usage](#)
- [Directory structure](#)
- [Configuration](#)

YOUR CONTENT

- [Front Matter](#)
- [Writing posts](#)
- [Working with drafts](#)
- [Creating pages](#)
- [Static Files](#)
- [Variables](#)
- [Collections](#)
- [Data Files](#)
- [Assets](#)
- [Blog migrations](#)

CUSTOMIZATION

```
.
├── _config.yml
├── _drafts
│   ├── begin-with-the-crazy-ideas.textile
│   └── on-simplicity-in-technology.markdown
├── _includes
│   ├── footer.html
│   └── header.html
├── _layouts
│   ├── default.html
│   └── post.html
├── _posts
│   ├── 2007-10-29-why-every-programmer-should-love-jekyll
│   └── 2009-04-26-barcamp-boston-4-roundup
├── _data
│   └── members.yml
├── _site
├── .jekyll-metadata
└── index.html
```

An overview of what each of these does:

FILE / DIRECTORY	DESCRIPTION
<code>_config.yml</code>	Stores configuration data. Many of these options can be specified from the command line executable but it's easier to specify them here so you don't have to remember them.

[Templates](#)
[Permalinks](#)
[Pagination](#)
[Plugins](#)
[Extras](#)

DEPLOYMENT

[GitHub Pages](#)
[Deployment methods](#)
[Continuous Integration](#)

MISCELLANEOUS

[Troubleshooting](#)
[Sites using Jekyll](#)
[Resources](#)
[Upgrading](#)

META

[Contributing](#)
[History](#)

FILE / DIRECTORY	DESCRIPTION
<code>_drafts</code>	Drafts are unpublished posts. The format of these files is without a date: <code>title.MARKUP</code> . Learn how to work with drafts .
<code>_includes</code>	These are the partials that can be mixed and matched by your layouts and posts to facilitate reuse. The liquid tag <code>{% include file.ext %}</code> can be used to include the partial in <code>_includes/file.ext</code> .
<code>_layouts</code>	These are the templates that wrap posts. Layouts are chosen on a post-by-post basis in the YAML Front Matter , which is described in the next section. The liquid tag <code>{{ content }}</code> is used to inject content into the web page.
<code>_posts</code>	Your dynamic content, so to speak. The naming convention of these files is important, and must follow the format: <code>YEAR-MONTH-DAY-title.MARKUP</code> . The permalinks can be customized for each post, but the date and markup language are determined solely by the file name.
<code>_data</code>	Well-formatted site data should be placed here. The jekyll engine will autoload all YAML files in this

FILE / DIRECTORY	DESCRIPTION
	directory (using either the <code>.yml</code> , <code>.yaml</code> , <code>.json</code> or <code>.csv</code> formats and extensions) and they will be accessible via <code>site.data</code> . If there's a file <code>members.yml</code> under the directory, then you can access contents of the file through <code>site.data.members</code> .
<code>_site</code>	This is where the generated site will be placed (by default) once Jekyll is done transforming it. It's probably a good idea to add this to your <code>.gitignore</code> file.
<code>.jekyll-metadata</code>	This helps Jekyll keep track of which files have not been modified since the site was last built, and which files will need to be regenerated on the next build. This file will not be included in the generated site. It's probably a good idea to add this to your <code>.gitignore</code> file.
<code>index.html</code> and other HTML, Markdown, Textile files	Provided that the file has a YAML Front Matter section, it will be transformed by Jekyll. The same will happen for any <code>.html</code> , <code>.markdown</code> , <code>.md</code> , or <code>.textile</code> file in your site's root directory or directories not listed above.
Other Files/Folders	Every other directory and file except for those listed above—such as <code>css</code>

FILE / DIRECTORY	DESCRIPTION
	and <code>images</code> folders, <code>favicon.ico</code> files, and so forth—will be copied verbatim to the generated site. There are plenty of sites already using Jekyll if you're curious to see how they're laid out.
<div><div>◀ BACK</div><div>NEXT ▶</div></div>	

The contents of this website

are © 2015 Tom Preston-Werner under the terms of the MIT License.

Proudly hosted by 