

Creating a Website With Jekyll

Contents

- What is Jekyll?.....3**
- Create a Configuration File.....3**
 - Config File Settings.....4
 - What are Variables?.....4
 - Create Default Variables.....4
- Write A Post.....5**
- Override Variables..... 5**
 - What is YAML Front Matter?..... 5

What is Jekyll?

Jekyll is a program that takes static content files and creates a website from them. Jekyll provides the ability to reuse website layouts, reuse data, and customize URLs without access to a formal backend.

The layout of a Jekyll website is stored separately from the content of the website. This means that the layout can be updated from only one location, rather than needing to update many pages individually. Jekyll also allows for multiple layouts, meaning a website can have several designs to fit various types of content. If a layout needs bit of unique data from content pages, such as a blog post title, that data can be stored in a variable and accessed from the layout.

Variables store data that can be reused throughout the website. These variables can store data from the site URL to specific information for a blog post. By storing data in a variable, the data can be updated across the site with one simple change to the variable, instead of having to update every page on the website. Other site information is also stored in variables, such as all of the site's posts, pagination, and a posts URL.

URLs fairly customizable with the help of Jekyll. Jekyll does link posts via their paths, but links posts via their filename. This allows for the path to each post to be changed easily when desired. The URL pattern for posts are fully customizable. Customized URLs are impossible without a backend, so this feature is very useful for a static website.

For more information, the official Jekyll documentation is available at <http://jekyllrb.com/>.

Create a Configuration File

The configuration file has settings that can affect many aspects of the website, including customization of URLs and declaring custom variables.

The configuration file is required for a Jekyll site to function. It can be customized so the Jekyll site works as desired. Default variables, build settings, server settings and extensions are all configured in the configuration file.

1. Create a `_config.yml` in your site's root directory.
2. Create a section to store your default Jekyll variables, as described in [Create Default Variables](#) on page 4

```
defaults:
  -
    scope:
      path: ""
      type: "pages"
    values:
      layout: "default"
```

3. Set any language processing variables, such as Github Flavored Markdown (GFM).

```
kramdown:
  input: GFM
```

4. Add any other customized settings, such as extensions, into the file. These settings are available in [Config File Settings](#) on page 4

Config File Settings

The configuration file can be customized so the Jekyll site works as desired. These settings can affect many aspects of the website, including customization of URLs and declaring custom variables.

Configuration Settings

Setting	Description	Options
source	Jekyll will look for files in this directory.	<code>source: DIR</code>
destination	Jekyll will create the website in this directory.	<code>destination: DIR</code>
exclude	Jekyll will ignore these files or directories.	<code>exclude: [DIR, FILE, ...]</code>
include	Jekyll will explicitly include these files or directories.	<code>include: [DIR, FILE, ...]</code>
timezone	Overrides the time set by the computer with a different <i>IANA Time Zones</i>	<code>timezone: TIMEZONE</code>
host	hostname of server	<code>host: HOSTNAME</code>
port	Website Port	<code>port: PORT</code>
baseurl	URL the website will be served from	<code>baseurl: URL</code>

What are Variables?

Jekyll provides the ability to pass variables between files. Variables can be used to pass customized data from a post, or used to set a default value for the entire website.

Jekyll variables are stored in two ways: as a variable belonging to the site, or as a variable belonging to the page.

Site variables will be consistent across all pages. The title of the site or home URL would be good site variables: they don't change depending on the page being viewed. These variables can be declared in the `config.yml` file.

Page variables do change based on an individual page. The title of a blog post or the date the post was written would be effective uses of a page variables. Page variables are declared in the YAML Front Matter.

Create Default Variables

Default variables can be declared in the configuration file to be reused throughout the website.

Default variables can be anything that is reused throughout the website. Items such as the title of the website, certain layout settings, or the author make good default variables.

Default variables can have their values overwritten by individual posts. If the post does not override the default variable, the default variable's value will be used for that post.

1. Pick a setting that will be reused on your website.
2. Decide the appropriate value to be the default of this setting.

This should be a value that will not look out of place if a post does not have an override.

3. Open the website's Jekyll configuration file. On a new line, enter the name of the variable and set it equal to the desired default value.



Attention: Be sure the variable is not nested under other variables, or it may cause the site to build incorrectly.

An variable named `backgroundColor` set to default to red would look like this in the configuration file:

```
backgroundColor: red
```

The variable should now be accessible from the website's posts and layout.

Write A Post

A post is a file filled with site content. Posts are inserted into the layout by Jekyll when it creates the website.

Posts can be a variety of file types. This means the content can be in whatever format suits it the best. All posts into the `_posts` folder. Jekyll will take these posts and convert them into a web-friendly format.

1. Create a file with a filename fitting the required format and of of an appropriate file type in the `_posts` folder.

Jekyll can use a variety of file types. Markdown and HTML are the most common. The filename of a post must follow a specific format. This format indicates the date the post was made, and the topic of the post. The following format for the filename is required: `YEAR-MONTH-DAY-title.extension`

An example filename would be `2015-12-25-DITA.html`

2. Open the file in a text editor of your choice.
3. Write content for your post and save it.
4. Open the post in your browser and double check that it looks correct.

Override Variables

Variables declared in the configuration file can be overwritten in posts. This lets data be customized based on the content the reader is viewing.

Variables should be overwritten in the YAML front matter of a post. This will override that variable only for that post. For instance, if a post is written by a new author, the author variable may need to be overwritten with the new author's name.

1. Be sure you have your default variable declared, as described in [Create Default Variables](#) on page 4
2. Create your post, as described in [Write A Post](#) on page 5
3. In the YAML front matter of the post created in step 2, write the name of the variable and set it equal to the value needed for this post.

An variable named `backgroundColor` set to be red would look like this in the YAML front matter:

```
---
backgroundColor: red
---
```

When viewing this post on the website, the post should reflect the changed setting. This changed setting will not be reflected on other posts.

What is YAML Front Matter?

YAML Font Matter is used to tell Jekyll how to display and use a post. Front Matter is used to override default variables, and set the values of other properties.

Front Matter must be the first few lines in a file. Jekyll detects that a file starts with Front Matter and knows that the file needs to be processed. Jekyll then parses the YAML found in the Front Matter and deals with the variables.

Front matter is separated from the rest of the file by three dashes on their own line at the start and the conclusion of the YAML. There may only be one section of Front Matter in each file.