



Jekyll theme for documentation — mydoc product

Last generated: July 10, 2016



Company
logo

Table of Contents

Overview

Release Notes

Installation

Authoring

Navigation

Formatting

Handling reviews

Publishing

Special layouts

Troubleshooting


```
{% if page.sidebar == "home_sidebar" %}
{% assign sidebar = site.data.sidebars.home_sidebar.entries %}

{% elsif page.sidebar == "product1_sidebar" %}
{% assign sidebar = site.data.sidebars.product1_sidebar.entries %}

{% elsif page.sidebar == "product2_sidebar" %}
{% assign sidebar = site.data.sidebars.product2_sidebar.entries %}

{% elsif page.sidebar == "mydoc_sidebar" %}
{% assign sidebar = site.data.sidebars.mydoc_sidebar.entries %}

{% else %}
{% assign sidebar = site.data.sidebars.home_sidebar.entries %}
{% endif %}
```

```

---
title: Alerts
tags: [formatting]
keywords: notes, tips, cautions, warnings, admonitions
last_updated: July 3, 2016
summary: "You can insert notes, tips, warnings, and important a
lerts in your content. These notes are stored as shortcodes mad
e available through the linksrefs.html include."
sidebar: mydoc_sidebar
permalink: mydoc_alerts
---

```

```

        sidebar: mydoc_sidebar
    }

    else
site.data.sidebars.home_sidebar.entries

```

i Note:

Sidebar syntax


```

entries:
- title: sidebar
  product: Jekyll Doc Theme
  version: 6.0
  folders:

  - title: Overview
    output: web, pdf
    folderitems:

    - title: Get started
      url: /index
      output: web, pdf

    - title: Introduction
      url: /mydoc_introduction
      output: web, pdf

    - title: Supported features
      url: /mydoc_supported_features
      output: web, pdf

    - title: About the theme author
      url: /mydoc_about
      output: web, pdf

    - title: Support
      url: /mydoc_support
      output: web, pdf

  - title: Release Notes
    output: web, pdf
    folderitems:

    - title: 6.0 Release notes
      url: /mydoc_release_notes_60
      output: web, pdf

    - title: 5.0 Release notes
      url: /mydoc_release_notes_50
      output: web, pdf

```

folder	subfolder	title	output
folderitem	subfolderitem	title	url
			output

```

      web      pdf
      output: web

      title:
      output: pdf
      type: frontmatter
      folderitems:
      - title:
        url: /titlepage
        output: pdf
        type: frontmatter
      - title:
        url: /tocpage
        output: pdf
        type: frontmatter

      output: pdf
```

Relative links and offline viewing

```

      .html      permalink

      .html      permalink
```

Page frontmatter

```
---
title: "Some title"
tags: [sample1, sample2]
keywords: keyword1, keyword2, keyword3
last_updated: Month day, year
summary: "optional summary here"
sidebar: sidebarname
permalink: filename.html
---
```

```
---
title: "Some title"
tags: [sample1, sample2]
keywords: keyword1, keyword2, keyword3
last_updated: Month day, year
summary: "optional summary here"
sidebar: sidebarname
permalink: filename.html
---

\

keywords

tags

toc: false

permalink
```

Where to store your documentation topics

Configure the top navigation

<code>external_url</code>		<code>url</code>	
<code>topnav</code>		<code>topnav_dropdowns</code>	<code>topnav_dropdowns</code>

Generating PDF

(<http://www.princexml.com/>)

(<http://www.princexml.com/doc/installing/>)

(<https://git-for-windows.github.io/>)

`youremail@domain.com`

Blogs / News

Markdown

(<http://kramdown.gettalong.org/>)

1. First item
2. Second item
3. Third item

F S T

1. First item
```  
alert("hello");  
```
2. Second item
Some pig!
3. Third item

Automated links

Other instructions

Overview

(<http://www.komposta.net/article/navgoco>)

(<http://startbootstrap.com/template-overviews/modern-business/>)

Supported features

Summary:

Supported feautres

Features	Supported	Notes
...
...
...
...
...

(<http://talk.jekyllrb.com>)

Features	Supported	Notes
...
...
...
...
...

Features	Supported	Notes
...
...
...

Features not available

Features	Supported	Notes
...

Features	Supported	Notes

About the theme author

Summary:

(<http://idratherbewriting.com>)

(<http://idratherbewriting.com/aboutme/>)

Support

Summary:

For more information on the Jekyll theme for documentation, please visit the [Jekyll theme for documentation GitHub repository](https://github.com/tomjohnson1492/jekyll-doc/issues) (https://github.com/tomjohnson1492/jekyll-doc/issues).

Release notes 6.0

Summary:

Mydoc product 6.0 release notes. This document contains information about the new features, bug fixes, and other changes in the 6.0 version of the Mydoc product. The release notes are organized into sections: Summary, Relative links, Subfolders for pages, Alerts templates, and Image templates.

Relative links

Relative links are used to navigate between different sections of the documentation. They are located at the bottom of each page and provide a quick way to access other parts of the documentation.

Subfolders for pages

Subfolders for pages are used to organize the documentation into a hierarchical structure. They are located at the top of each page and provide a quick way to access other parts of the documentation.

Alerts templates

Alerts templates are used to create alerts in the Mydoc product. They are located in the Alerts section of the product and provide a quick way to create alerts.

Image templates

Image templates are used to create images in the Mydoc product. They are located in the Image section of the product and provide a quick way to create images.

Automated links using Markdown formatting

Mydoc product User Guide

Mydoc product User Guide

Mydoc product User Guide

Mydoc product User Guide

Mydoc product User Guide

Workflow maps

Mydoc product User Guide

Mydoc product User Guide

Mydoc product User Guide

Mydoc product User Guide

Mydoc product User Guide

Release notes 5.0

Summary:

Release 5.0 of the mydoc product is a major update to the user interface. It includes a new design, new features, and bug fixes. The new design is more modern and easier to use. The new features include a new sidebar, a new search bar, and a new filter system. The bug fixes include fixing a number of issues that were reported by users.

Unique sidebars for each product

One of the new features in release 5.0 is the ability to have unique sidebars for each product. This allows users to see only the information that is relevant to the product they are viewing. For example, the sidebar for the "mydoc product" will show the "mydoc product" section, while the sidebar for the "mydoc product" will show the "mydoc product" section. This makes it easier for users to find the information they need and reduces the amount of clutter on the page.

Permalinks

Another new feature in release 5.0 is the ability to use permalinks. Permalinks are permanent links to a specific page or section of a document. They are useful for sharing a specific piece of information with others or for bookmarking a page. In release 5.0, we have added permalinks to all of the pages in the mydoc product. This makes it easier for users to share and bookmark specific pages.

Kramdown and Rouge

With the release of Jekyll 3.0, the default Markdown parser has changed from `Redcarpet` to `Kramdown`. This change was made to improve the rendering of Markdown documents, and to ensure that the output is consistent across different platforms and versions of Jekyll.

Blog feature

The `blog` feature has been added to the Jekyll theme, allowing you to easily create and manage a blog. This feature includes a default layout for blog posts, as well as a default set of tags and categories. You can also customize the blog feature to suit your needs, by adding additional tags and categories, or by changing the default layout.

Updated documentation

The documentation for the Jekyll theme has been updated to reflect the changes in version 5.0. This includes updates to the `README` file, as well as the `CONTRIBUTING` and `SECURITY` files.

Fixed errors

Several errors have been fixed in this release, including a bug that caused the `blog` feature to fail on some platforms, and a bug that caused the `blog` feature to not work correctly with the `blog` feature.

Accessing the old theme

(<https://github.com/tomjohnson1492/jekyllldoctheme-separate-outputs>)

About Ruby, Gems, Bundler, and other prerequisites

Summary:

About Ruby

About Ruby Gems

(<https://rubygems.org/>)

Rubygem package managers

(<http://bundler.io/>)

(<https://rubygems.org/gems/jekyll>)

(<https://rubygems.org/gems/github-pages>)

```
github-pages-health-check = 1.1.0
jekyll = 3.0.3
jekyll-coffeescript = 1.0.1
jekyll-feed = 0.4.0
jekyll-gist = 1.4.0
jekyll-github-metadata = 1.9.0
jekyll-mentions = 1.1.2
jekyll-paginate = 1.1.0
jekyll-redirect-from = 0.10.0
jekyll-sass-converter = 1.3.0
jekyll-seo-tag = 1.3.2
jekyll-sitemap = 0.10.0
jekyll-textile-converter = 0.1.0
jemoji = 0.6.2
kramdown = 1.10.0
liquid = 3.0.6
mercenary ~> 0.3
rdiscount = 2.1.8
redcarpet = 3.3.3
RedCloth = 4.2.9
rouge = 1.10.1
terminal-table ~> 1.
```

(<http://bundler.io/>)

```
gem 'github-pages'
gem 'jekyll'
```

(<https://rubygems.org>)

the number \sim last digit before the last period in

```
gem 'jekyll', '~> 3.0', '>= 3.0.3'
```

(<http://stackoverflow.com/questions/5170547/what-does-tilde-greater-than-mean-in-ruby-gem-dependencies>)

Gemfile.lock

GEM

```
remote: https://rubygems.org/
specs:
  RedCloth (4.2.9)
  activesupport (4.2.5.1)
    i18n (~> 0.7)
    json (~> 1.7, >= 1.7.7)
    minitest (~> 5.1)
    thread_safe (~> 0.3, >= 0.3.4)
    tzinfo (~> 1.1)
  addressable (2.3.8)
  coffee-script (2.4.1)
    coffee-script-source
    execjs
  coffee-script-source (1.10.0)
  colorator (0.1)
  ethon (0.8.1)
    ffi (>= 1.3.0)
  execjs (2.6.0)
  faraday (0.9.2)
    multipart-post (>= 1.2, < 3)
  ffi (1.9.10)
  gemoji (2.1.0)
  github-pages (52)
    RedCloth (= 4.2.9)
    github-pages-health-check (= 1.0.1)
    jekyll (= 3.0.3)
    jekyll-coffeescript (= 1.0.1)
    jekyll-feed (= 0.4.0)
    jekyll-gist (= 1.4.0)
    jekyll-mentions (= 1.0.1)
    jekyll-paginate (= 1.1.0)
    jekyll-redirect-from (= 0.9.1)
    jekyll-sass-converter (= 1.3.0)
    jekyll-seo-tag (= 1.3.1)
    jekyll-sitemap (= 0.10.0)
    jekyll-textile-converter (= 0.1.0)
    jemoji (= 0.5.1)
    kramdown (= 1.9.0)
    liquid (= 3.0.6)
    mercenary (~> 0.3)
    rdiscount (= 2.1.8)
    redcarpet (= 3.3.3)
    rouge (= 1.10.1)
    terminal-table (~> 1.4)
  github-pages-health-check (1.0.1)
    addressable (~> 2.3)
```

```
net-dns (~> 0.8)
octokit (~> 4.0)
public_suffix (~> 1.4)
typhoeus (~> 0.7)
html-pipeline (2.3.0)
  activesupport (>= 2, < 5)
  nokogiri (>= 1.4)
i18n (0.7.0)
jekyll (3.0.3)
  colorator (~> 0.1)
  jekyll-sass-converter (~> 1.0)
  jekyll-watch (~> 1.1)
  kramdown (~> 1.3)
  liquid (~> 3.0)
  mercenary (~> 0.3.3)
  rouge (~> 1.7)
  safe_yaml (~> 1.0)
jekyll-coffeescript (1.0.1)
  coffee-script (~> 2.2)
jekyll-feed (0.4.0)
jekyll-gist (1.4.0)
  octokit (~> 4.2)
jekyll-mentions (1.0.1)
  html-pipeline (~> 2.3)
  jekyll (~> 3.0)
jekyll-paginate (1.1.0)
jekyll-redirect-from (0.9.1)
  jekyll (>= 2.0)
jekyll-sass-converter (1.3.0)
  sass (~> 3.2)
jekyll-seo-tag (1.3.1)
  jekyll (~> 3.0)
jekyll-sitemap (0.10.0)
jekyll-textile-converter (0.1.0)
  RedCloth (~> 4.0)
jekyll-watch (1.3.1)
  listen (~> 3.0)
jemoji (0.5.1)
  gemoji (~> 2.0)
  html-pipeline (~> 2.2)
  jekyll (>= 2.0)
json (1.8.3)
kramdown (1.9.0)
liquid (3.0.6)
listen (3.0.6)
  rb-fsevent (>= 0.9.3)
  rb-inotify (>= 0.9.7)
```



```
mercenary (0.3.5)
mini_portile2 (2.0.0)
minitest (5.8.4)
multipart-post (2.0.0)
net-dns (0.8.0)
nokogiri (1.6.7.2)
  mini_portile2 (~> 2.0.0.rc2)
octokit (4.2.0)
  sawyer (~> 0.6.0, >= 0.5.3)
public_suffix (1.5.3)
rb-fsevent (0.9.7)
rb-inotify (0.9.7)
  ffi (>= 0.5.0)
rdiscount (2.1.8)
redcarpet (3.3.3)
rouge (1.10.1)
safe_yaml (1.0.4)
sass (3.4.21)
sawyer (0.6.0)
  addressable (~> 2.3.5)
  faraday (~> 0.8, < 0.10)
terminal-table (1.5.2)
thread_safe (0.3.5)
typhoeus (0.8.0)
  ethon (>= 0.8.0)
tzinfo (1.2.2)
  thread_safe (~> 0.1)
```

PLATFORMS

ruby

DEPENDENCIES

github-pages
jekyll

BUNDLED WITH

1.11.2

bundle update

(<http://bundler.io/rationale.html>)

Install Jekyll on Mac

Summary:

Ruby and RubyGems

```
curl -sSL https://rubygems.org/pages/download | \
  sudo sh -c 'which ruby && which gem && \
  rm -rf /usr/local/bin/ruby && \
  rm -rf /usr/local/bin/gem && \
  curl -sSL https://raw.githubusercontent.com/rvm/rvm/master/binscripts/alias.sh | \
  sh -s stable'
```

```
/usr/local/bin/ruby
```

```
/usr/local/bin/gem
```

```
/Users/MacBookPro/.rvm/rubies/ruby-2.2.1/bin/gem

gem install jekyll
```

```
>ERROR: While executing gem ... (Gem::FilePermissionError)
  You don't have write permissions for the /Library/Ruby/Gems/
  2.0.0 directory.
```

Install Homebrew

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```
brew update
```

Install Ruby through Homebrew

```
brew install ruby
```

```
which ruby    which gem
```

```
/usr/local/bin/ruby
```

```
/usr/local/bin/gem
```

Install the Jekyll gem

```
gem install jekyll
```

```
gem install jekyll
```

(<http://jekyllrb.com>)

Installing dependencies through Bundler

```
bundle install
```

(<http://bundler.io/>)

```
gem install bundler
```

```
sudo
```

```
...$ jekyll new my-awesome-site
...$ cd my-awesome-site
...$ jekyll serve
```

Serve the Jekyll Documentation theme

```
...$ jekyll serve
...$
...$ jekyll serve
...$
...$ jekyll serve --open
...$
```

Install Jekyll on Windows

✓ **Tip:**

(<https://git-for-windows.github.io/>)

Install Ruby

(<http://rubyinstaller.org/downloads/>)

RubyInstallers

Install Ruby Development Kit

(<http://rubyinstaller.org/downloads/>)

Development Kit

For use with Ruby 2.0 and above...

C

RubyDevKit

RubyDevKit

cd foldername

dir

cd ../

```
/users /
```

```
ruby dk.rb init
```

```
ruby dk.rb install
```

(<https://github.com/oneclick/rubyinstaller/wiki/Development-Kit>)

Install the Jekyll gem

```
gem
```

```
gem install jekyll
```

(<http://jekyllrb.com>)

Installing dependencies through Bundler

(<http://bundler.io/>)

Install Bundler

```
gem install bundler
```

```
bundle init
```

```
source "https://rubygems.org"
```

```
gem 'wdm'  
gem 'jekyll'
```

(<https://rubygems.org/gems/wdm/versions/0.1.1>)

```
bundle install
```

Git Clients for Windows

(<https://git-for-windows.github.io/>)

Serve the Jekyll Documentation theme

```
jekyll serve
```

/

Pages

Summary:

Where to author content

Page names and excluding files from outputs

exclude:

- filename.md
- subfolder_name/
- mydoc_*
- gitignore

*

Frontmatter

```
---
title: Alerts
tags: [formatting]
keywords: notes, tips, cautions, warnings, admonitions
last_updated: July 3, 2016
summary: "You can insert notes, tips, warnings, and important a
lerts in your content."
sidebar: mydoc_sidebar
permalink: mydoc_alerts.html
---
```

```

    title      permalink
  
```

Frontmatter	Required?	Description
title		
tags		
keywords		
last_updated		

Frontmatter	Required?	Description
summary		<p>Summary of the page content.</p> <p>(http://www.nngroup.com/articles/corporate-blogs-front-page-structure/)</p>
permalink	<i>must</i>	<p>Permalink for the page.</p>
datatable	<code>datatable: true</code>	<p>(https://www.datatables.net/)</p>
	<code>toc: false</code>	

Colons in page titles

Saving pages as drafts

`published: false`

Markdown or HTML format

By default, the content of the pages is rendered in HTML format. However, you can also use Markdown format. To enable Markdown format, you need to set the `markdown="1"` attribute in the `<div>` tag.

```
<div markdown="1">This is a [link](http://exmaple.com).</div>
```

For example, the following code will render the text "This is a link" in HTML format, and the text "This is a link" in Markdown format.

Where to save pages

By default, the pages are saved in the `pages` directory. You can also save the pages in a different directory. To specify the directory, you need to set the `pages_dir` attribute in the `<div>` tag.

Page names

By default, the page names are generated automatically. You can also specify the page names manually. To specify the page names, you need to set the `page_names` attribute in the `<div>` tag.

Kramdown Markdown

For more information on Kramdown, see the [Kramdown website](http://kramdown.org/). For more information on the `rouge` syntax highlighter, see the [rouge website](http://rouge.jneke.net/).

(<http://idratherbewriting.com/2016/02/21/bug-with-kramdown-and-rouge-with-github-pages/>)

```
highlighter: rouge
markdown: kramdown
kramdown:
  input: GFM
  auto_ids: true
  hard_wrap: false
  syntax_highlighter: rouge
```

Automatic mini-TOCs

```
---
toc: false
---

##

<h2>

<h2 id="mysampleid">Heading</h2>
```

Headings

```
## Second-level heading
```

Result:

Second-level heading

```
### Third-level heading
```

Result:

```
#### Fourth-level heading
```

Result:

Fourth-level heading

Headings with ID Tags

```
## Headings with ID Tags {#someIdTag}
```

```
[Some link](#someIdTag)
```

Result:

Specify a particular page layout

```
layout: mylayout.html
```

Comments

Posts

Summary:

About posts

Placeholder text for the 'About posts' section.

Allowed frontmatter

Placeholder text for the 'Allowed frontmatter' section.

Frontmatter	Required?	Description
title		

Frontmatter	Required?	Description
tags		<p>Tags are used to categorize content. They are typically used to group related content together. For example, you might use the tag "blog" to group all blog posts together. Tags are typically used in the following way:</p> <pre>tags: - blog - mydoc</pre> <p>Tags are typically used in the following way:</p> <pre>tags: - blog - mydoc</pre>
keywords		<p>Keywords are used to describe the content of a document. They are typically used to help search engines find relevant content. For example, you might use the keyword "mydoc" to describe a document about mydoc. Keywords are typically used in the following way:</p> <pre>keywords: - mydoc - mydoc</pre> <p>Keywords are typically used in the following way:</p> <pre>keywords: - mydoc - mydoc</pre>
summary		<p>A summary is a brief description of the content of a document. It is typically used to help search engines find relevant content. For example, you might use the summary "mydoc" to describe a document about mydoc. A summary is typically used in the following way:</p> <pre>summary: mydoc</pre> <p>(http://www.nngroup.com/articles/corporate-blogs-front-page-structure/)</p>
permalink		<p>A permalink is a permanent URL that points to a specific document. It is typically used to help search engines find relevant content. For example, you might use the permalink "mydoc" to point to a document about mydoc. A permalink is typically used in the following way:</p> <pre>permalink: mydoc</pre>

Lists

Summary: Lists are used to group related items together.

Bulleled Lists

Example of bulleted list:

- * first item
- * second item
- * third item

Result:

Numbered list

Example of numbered list:

1. First item.
1. Second item.
1. Third item.

Result:

Complex Lists

1. Sample first item.
 - * sub-bullet one
 - * sub-bullet two
2. Continuing the list
 - 1. sub-list numbered one
 - 2. sub-list numbered two
3. Another list item.

Result:

```
1. Sample first item.  
    * sub-bullet one  
    * sub-bullet two  
2. Continuing the list  
    1. sub-list numbered one  
    2. sub-list numbered two  
3. Another list item.
```

Another Complex List

1. Sample first item.

This is a result statement that talks about something....

2. Continuing the list

```
<div markdown="span" class="alert alert-info" role="aler  
t"><i class="fa fa-info-circle"></i> <b>Note:</b> Remember to d  
o this. If you have "quotes", you must escape them.</div>
```

Here's a list in here:

- * first item
- * second item

3. Another list item.

```
```js  
function alert("hello");
```
```

4. Another item.

Result:

Note:

```
function alert("hello");
```

```
91
92 Here's a list with some intercepting text:
93
94 1. Sample first item.
95
96 .... This is a result statement that talks about something....
97
98 2. Continuing the list
99
100 .... {% include note.html content="Remember to do this. If you have \"quotes\", you must escape
101
102 .... Here's a list in here:
103
104 .... * first item
105 .... * second item
106
107 3. Another list item.
108
109 .... ``js
110 .... function alert("hello");
111 .... ``
112
113 4. Another item.
114
115 The key principle is to line up |
116
117 ## Links
118
```

Lining up the left edge ensures the list stays in tact.

Conditional logic

Summary:

About Liquid and conditional statements

✓ **Tip:** (<http://docs.shopify.com/themes/liquid-documentation/basics>)

Where to store filtering values

There are two places where you can store filtering values: the `page` object and the `config` file. The `page` object is a hash of values that are available to the `if` and `unless` tags. The `config` file is a hash of values that are available to the `if` and `unless` tags.

Conditional logic based on config file value

Conditional logic based on config file value is used to conditionally display content based on the value of a config file variable.

```
platform: mac
```

Conditional logic based on config file value is used to conditionally display content based on the value of a config file variable. The `platform` variable is set to `mac` in the config file.

```
{% if page.platform == "mac" %}
Here's some info about the Mac.
{% elsif page.platform == "windows" %}
Here's some info about Windows ...
{% endif %}
```

Conditional logic based on config file value is used to conditionally display content based on the value of a config file variable. The `platform` variable is set to `mac` in the config file.

```
if-elseif
elseif else if
if-else
```

To bake a casserole:

1. Gather the ingredients.
{% if page.audience == "writer" %}
2. Add in a pound of meat.
{% elsif page.audience == "designer" %}
3. Add in an extra can of beans.
{% endif %}
3. Bake in oven for 45 min.

```

    elsif ... else ... if ...
endif

```

Or operator

```

    {% if page.audience contains "vegan" or page.audience == "vegetarian" %}

```

(<http://docs.shopify.com/themes/liquid-documentation/basics/operators>)

```

    {% if page.audience contains "vegan" or "vegetarian" %}

```

```

{% if page.audience contains "vegan" or page.audience == "vegetarian" %}
    Then run this...
{% endif %}

```

```

{% if page.audience contains "vegan" or "vegetarian" %}
    // run this.
{% endif %}

```

Unless operator

```

    {% unless page.audience == "writer" %}

```



```
{% unless site.output == "pdf" %}
...do this
{% endunless %}
```

unless

not

Storing conditions in the `_data` folder

```
{% if site.data.options.output == "alpha" %}
show this content...
{% elsif site.data.options.output == "beta" %}
show this content...
{% else %}
this shows if neither of the above two if conditions are met.
{% endif %}
```

`output`

Specifying the location for `_data`

`data_source`

`_data`

```
data_source: data_amydoc_content_reuselpha
```

data_source: data_beta

Content reuse

Summary:

Content reuse is a feature that allows you to reuse content from one page in another page. This is useful for creating a consistent look and feel across your documentation, and for reducing the amount of content you need to maintain.

About content reuse

Content reuse is a feature that allows you to reuse content from one page in another page. This is useful for creating a consistent look and feel across your documentation, and for reducing the amount of content you need to maintain.

```
{% include custom/mypage.html %}
```

Content reuse is a feature that allows you to reuse content from one page in another page. This is useful for creating a consistent look and feel across your documentation, and for reducing the amount of content you need to maintain.

Page-level variables

```
---
title: Page-level variables
permalink: page_level_variables/
thing1: Joe
thing2: Dave
---
```

page

```
thing1: {{page.thing1}}
thing2: {{page.thing2}}
```


Collections

Summary:

What are collections

(<http://ben.balter.com/2016/02/20/jekyll-collections/>)

Create a collection

```
collections:  
  tooltips:  
    output: true
```

Interacting with collections

```
site.pages      site.posts      site.collectionname  
               collectionname  
               site.tooltips
```

(<http://jekyllrb.com/docs/collections/>)

How to use collections

...

Video tutorial on collections

...

(<http://jekyll.tips/jekyll-casts/introduction-to-collections/>)

...

WebStorm Text Editor

Summary:

About text editors and WebStorm

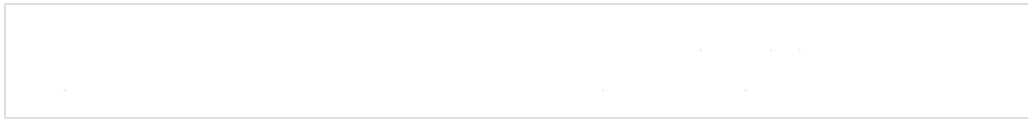
Remove unnecessary plugins

WebStorm > Preferences > Plugins

Set default tab indent to 3 spaces instead of 4

(https://www.jetbrains.com/help/webstorm/2016.1/code-style-javascript.html?origin=old_help#d658997e132)

**. Don't
select the "Use tab
character" check box.
Set **3 Tab size
Indent**



Add the Markdown Support plugin



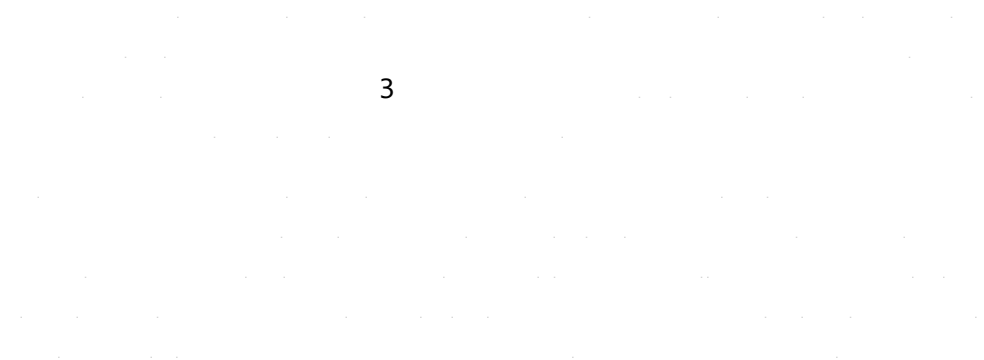
Enable Soft Wraps (word wrapping)



Exclude a directory



Set tabs to 3 spaces



(https://www.jetbrains.com/help/webstorm/2016.1/code-style-javascript.html?origin=old_help#d658997e132)

Shortcuts

| Command | Shortcuts |
|---------|-----------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

☒ **Tip:**

For more information on shortcuts, see the [Keyboard Shortcuts](#) section of the **WebStorm > Preferences > Keymap** dialog.

Finding files

File > Find in Files...
Ctrl+Shift+F
Enter

Identifying changed files

File > Recent Changes
Ctrl+Alt+Shift+F

Creating file templates

File > New > Edit File
Templates
File > Default Settings > Editor > File and Code
Templates
+

Disable pair quotes

Smartkeys
WebStorm > Preferences > Editor >
Insert pair quotes

Atom Text Editor

Summary:

(<https://atom.io/>)

Atom > Preferences

Settings

Show Invisibles

Soft Wrap

| | |
|------------|------|
| Tab Length | 4 |
| Tab Type | soft |

Atom > Preferences

Packages

autocomplete-plus

Cmd + T

Cmd + Shift + F

Cmd + Alt + S

Sidebar Navigation

Summary:

Navgoco is a jQuery plugin that allows you to create a sidebar navigation menu. It is a simple and easy to use plugin that can be used to create a sidebar navigation menu for your website. It is a jQuery plugin that allows you to create a sidebar navigation menu. It is a simple and easy to use plugin that can be used to create a sidebar navigation menu for your website.

Navgoco foundation

Navgoco is a jQuery plugin that allows you to create a sidebar navigation menu. It is a simple and easy to use plugin that can be used to create a sidebar navigation menu for your website. It is a jQuery plugin that allows you to create a sidebar navigation menu. It is a simple and easy to use plugin that can be used to create a sidebar navigation menu for your website.

active

Accordion sidebar feature

Navgoco is a jQuery plugin that allows you to create a sidebar navigation menu. It is a simple and easy to use plugin that can be used to create a sidebar navigation menu for your website. It is a jQuery plugin that allows you to create a sidebar navigation menu. It is a simple and easy to use plugin that can be used to create a sidebar navigation menu for your website.

#mysidebar

accordion

.navgoco

```

graph TD
    A[accordion] --> B[true]
    A --> C[false]
    B --> D[true]
    B --> E[false]
    C --> F[true]
    C --> G[false]
    D --> H[true]
    D --> I[false]
    E --> J[true]
    E --> K[false]
    F --> L[true]
    F --> M[false]
    G --> N[true]
    G --> O[false]
    H --> P[true]
    H --> Q[false]
    I --> R[true]
    I --> S[false]
    J --> T[true]
    J --> U[false]
    K --> V[true]
    K --> W[false]
    L --> X[true]
    L --> Y[false]
    M --> Z[true]
    M --> AA[false]
    N --> AB[true]
    N --> AC[false]
    O --> AD[true]
    O --> AE[false]
    P --> AF[true]
    P --> AG[false]
    Q --> AH[true]
    Q --> AI[false]
    R --> AJ[true]
    R --> AK[false]
    S --> AL[true]
    S --> AM[false]
    T --> AN[true]
    T --> AO[false]
    U --> AP[true]
    U --> AQ[false]
    V --> AR[true]
    V --> AS[false]
    W --> AT[true]
    W --> AU[false]
    X --> AV[true]
    X --> AW[false]
    Y --> AX[true]
    Y --> AY[false]
    Z --> AZ[true]
    Z --> BA[false]
    AA --> BB[true]
    AA --> BC[false]
    AB --> BD[true]
    AB --> BE[false]
    AC --> BF[true]
    AC --> BG[false]
    AD --> BH[true]
    AD --> BI[false]
    AE --> BJ[true]
    AE --> BK[false]
    AF --> BL[true]
    AF --> BM[false]
    AG --> BN[true]
    AG --> BO[false]
    AH --> BP[true]
    AH --> BQ[false]
    AI --> BR[true]
    AI --> BS[false]
    AJ --> BT[true]
    AJ --> BU[false]
    AK --> BV[true]
    AK --> BW[false]
    AL --> BX[true]
    AL --> BY[false]
    AM --> BZ[true]
    AM --> CA[false]
    AN --> CB[true]
    AN --> CC[false]
    AO --> CD[true]
    AO --> CE[false]
    AP --> CF[true]
    AP --> CG[false]
    AQ --> CH[true]
    AQ --> CI[false]
    AR --> CJ[true]
    AR --> CK[false]
    AS --> CL[true]
    AS --> CM[false]
    AT --> CN[true]
    AT --> CO[false]
    AU --> CP[true]
    AU --> CQ[false]
    AV --> CR[true]
    AV --> CS[false]
    AW --> CT[true]
    AW --> CU[false]
    AX --> CV[true]
    AX --> CW[false]
    AY --> CX[true]
    AY --> CY[false]
    AZ --> CZ[true]
    AZ --> DA[false]
    BA --> DB[true]
    BA --> DC[false]
    BB --> DD[true]
    BB --> DE[false]
    BC --> DF[true]
    BC --> DG[false]
    BD --> DH[true]
    BD --> DI[false]
    BE --> DJ[true]
    BE --> DK[false]
    BF --> DL[true]
    BF --> DM[false]
    BG --> DN[true]
    BG --> DO[false]
    BH --> DP[true]
    BH --> DQ[false]
    BI --> DR[true]
    BI --> DS[false]
    BJ --> DT[true]
    BJ --> DU[false]
    BK --> DV[true]
    BK --> DW[false]
    BL --> DX[true]
    BL --> DY[false]
    BM --> DZ[true]
    BM --> EA[false]
    BN --> EB[true]
    BN --> EC[false]
    BO --> ED[true]
    BO --> EE[false]
    BP --> EF[true]
    BP --> EG[false]
    BQ --> EH[true]
    BQ --> EI[false]
    BR --> EJ[true]
    BR --> EK[false]
    BS --> EL[true]
    BS --> EM[false]
    BT --> EN[true]
    BT --> EO[false]
    BU --> EP[true]
    BU --> EQ[false]
    BV --> ER[true]
    BV --> ES[false]
    BW --> ET[true]
    BW --> EU[false]
    BX --> EV[true]
    BX --> EW[false]
    BY --> EX[true]
    BY --> EY[false]
    BZ --> EZ[true]
    BZ --> FA[false]
    CA --> FB[true]
    CA --> FC[false]
    CB --> FD[true]
    CB --> FE[false]
    CC --> FF[true]
    CC --> FG[false]
    CD --> FH[true]
    CD --> FI[false]
    CE --> FJ[true]
    CE --> FK[false]
    CF --> FL[true]
    CF --> FM[false]
    CG --> FN[true]
    CG --> FO[false]
    CH --> FP[true]
    CH --> FQ[false]
    CI --> FR[true]
    CI --> FS[false]
    CJ --> FT[true]
    CJ --> FU[false]
    CK --> FV[true]
    CK --> FW[false]
    CL --> FX[true]
    CL --> FY[false]
    CM --> FZ[true]
    CM --> GA[false]
    CN --> GB[true]
    CN --> GC[false]
    CO --> GD[true]
    CO --> GE[false]
    CP --> GF[true]
    CP --> GG[false]
    CQ --> GH[true]
    CQ --> GI[false]
    CR --> GJ[true]
    CR --> GK[false]
    CS --> GL[true]
    CS --> GM[false]
    CT --> GN[true]
    CT --> GO[false]
    CU --> GP[true]
    CU --> GQ[false]
    CV --> GR[true]
    CV --> GS[false]
    CW --> GT[true]
    CW --> GU[false]
    CX --> GV[true]
    CX --> GW[false]
    CY --> GX[true]
    CY --> GY[false]
    CZ --> GZ[true]
    CZ --> HA[false]
    DA --> HB[true]
    DA --> HC[false]
    DB --> HD[true]
    DB --> HE[false]
    DC --> HF[true]
    DC --> HG[false]
    DD --> HH[true]
    DD --> HI[false]
    DE --> HJ[true]
    DE --> HK[false]
    DF --> HL[true]
    DF --> HM[false]
    DG --> HN[true]
    DG --> HO[false]
    DH --> HP[true]
    DH --> HQ[false]
    DI --> HR[true]
    DI --> HS[false]
    DJ --> HT[true]
    DJ --> HU[false]
    DK --> HV[true]
    DK --> HW[false]
    DL --> HX[true]
    DL --> HY[false]
    DM --> HZ[true]
    DM --> IA[false]
    DN --> IB[true]
    DN --> IC[false]
    DO --> ID[true]
    DO --> IE[false]
    DP --> IF[true]
    DP --> IG[false]
    DQ --> IH[true]
    DQ --> II[false]
    DR --> IJ[true]
    DR --> IK[false]
    DS --> IL[true]
    DS --> IM[false]
    DT --> IN[true]
    DT --> IO[false]
    DU --> IP[true]
    DU --> IQ[false]
    DV --> IR[true]
    DV --> IS[false]
    DW --> IT[true]
    DW --> IU[false]
    DX --> IV[true]
    DX --> IW[false]
    DY --> IX[true]
    DY --> IY[false]
    DZ --> IZ[true]
    DZ --> JA[false]
    EA --> JB[true]
    EA --> JC[false]
    EB --> JD[true]
    EB --> JE[false]
    EC --> JF[true]
    EC --> JG[false]
    ED --> JH[true]
    ED --> JI[false]
    EE --> JJ[true]
    EE --> JK[false]
    EF --> JL[true]
    EF --> JM[false]
    EG --> JN[true]
    EG --> JO[false]
    EH --> JP[true]
    EH --> JQ[false]
    EI --> JR[true]
    EI --> JS[false]
    EJ --> JT[true]
    EJ --> JU[false]
    EK --> JV[true]
    EK --> JW[false]
    EL --> JX[true]
    EL --> JY[false]
    EM --> JZ[true]
    EM --> KA[false]
    EN --> KB[true]
    EN --> KC[false]
    EO --> KD[true]
    EO --> KE[false]
    EP --> KF[true]
    EP --> KG[false]
    EQ --> KH[true]
    EQ --> KI[false]
    ER --> KJ[true]
    ER --> KK[false]
    ES --> KL[true]
    ES --> KM[false]
    ET --> KN[true]
    ET --> KO[false]
    EU --> KP[true]
    EU --> KQ[false]
    EV --> KR[true]
    EV --> KS[false]
    EW --> KT[true]
    EW --> KU[false]
    EX --> KV[true]
    EX --> KW[false]
    EY --> KX[true]
    EY --> KY[false]
    EZ --> KZ[true]
    EZ --> LA[false]
    FA --> LB[true]
    FA --> LC[false]
    FB --> LD[true]
    FB --> LE[false]
    FC --> LF[true]
    FC --> LG[false]
    FD --> LH[true]
    FD --> LI[false]
    FE --> LJ[true]
    FE --> LK[false]
    FF --> LL[true]
    FF --> LM[false]
    FG --> LN[true]
    FG --> LO[false]
    FH --> LP[true]
    FH --> LQ[false]
    FI --> LR[true]
    FI --> LS[false]
    FJ --> LT[true]
    FJ --> LU[false]
    FK --> LV[true]
    FK --> LW[false]
    FL --> LX[true]
    FL --> LY[false]
    FM --> LZ[true]
    FM --> MA[false]
    FN --> MB[true]
    FN --> MC[false]
    FO --> MD[true]
    FO --> ME[false]
    FP --> MF[true]
    FP --> MG[false]
    FQ --> MH[true]
    FQ --> MI[false]
    FR --> MJ[true]
    FR --> MK[false]
    FS --> ML[true]
    FS --> MM[false]
    FT --> MN[true]
    FT --> MO[false]
    FU --> MP[true]
    FU --> MQ[false]
    FV --> MR[true]
    FV --> MS[false]
    FW --> MT[true]
    FW --> MU[false]
    FX --> MV[true]
    FX --> MW[false]
    FY --> MX[true]
    FY --> MY[false]
    FZ --> MZ[true]
    FZ --> NA[false]
    GA --> NB[true]
    GA --> NC[false]
    GB --&
```

Fixed position sidebar

Opening sidebar links into external pages

```

    external_url      url
    a href             external_url

```

```
{% if subfolderitem.external_url %}
<li><a href="{{subfolderitem.external_url}}" target="_blank">{{deeplevel.title}}</a></li>
```

```
external_url
```

Sidebar item highlighting

```
url      active
```

```
{% elsif page.url == subfolderitem.url %}
  <li class="active"><a href="{{subfolderitem.url | prepend: site.baseurl}}">{{subfolderitem.title}}</a></li>
```

```

page.url      subfolderitem.url      active
      active
page.url      {{page.url}}
              url
              active
              url

```

```
- title: Understanding how the sidebar works
  url: /mydoc_understand_sidebar/
  output: web, pdf
```

YAML tutorial in the context of Jekyll

Summary:

YAML is a data serialization language that is easy for humans to read and write. It is often used for configuration files and data exchange between systems. In the context of Jekyll, YAML is used to define the structure and content of the site's pages and collections.

Overview

YAML is a data serialization language that is easy for humans to read and write. It is often used for configuration files and data exchange between systems. In the context of Jekyll, YAML is used to define the structure and content of the site's pages and collections. The following is a brief overview of the YAML syntax and its use in Jekyll.

YAML is a data serialization language that is easy for humans to read and write. It is often used for configuration files and data exchange between systems. In the context of Jekyll, YAML is used to define the structure and content of the site's pages and collections. The following is a brief overview of the YAML syntax and its use in Jekyll.

(<https://github.com/tefra/navgoco>)

YAML overview

YAML is a data serialization language that is easy for humans to read and write. It is often used for configuration files and data exchange between systems. In the context of Jekyll, YAML is used to define the structure and content of the site's pages and collections. The following is a brief overview of the YAML syntax and its use in Jekyll.

YAML is a data serialization language that is easy for humans to read and write. It is often used for configuration files and data exchange between systems. In the context of Jekyll, YAML is used to define the structure and content of the site's pages and collections. The following is a brief overview of the YAML syntax and its use in Jekyll.

YAML:

```
name:
  husband: Tom
  wife: Shannon
```

Markdown + Liquid:

```
Husband's name: {{site.data.samplelist.name.husband}}
```

```
Wife's name: {{site.data.samplelist.name.wife}}
```

```
mydoc      samplelist      site.data.samplelist
```

Result:

Example 2: Line breaks

YAML:

```
feedback: >
  This is my feedback to you.
  Even if I include linebreaks here,
  all of the linebreaks will be removed when the value is inserted.

block: |
  This pipe does something a little different.
  It preserves the breaks.
  This is really helpful for code samples,
  since you can format the code samples with
  the appropriate
```

Markdown:

****Feedback****

This is my feedback to you. Even if I include linebreaks here, all of the linebreaks will be removed when the value is inserted.

****Block****

This pipe does something a little different. It preserves the breaks. This is really helpful for code samples, since you can format the code samples with the appropriate white spacing.

Result:**Feedback****Block**

Example 3: Simple list

YAML

```
bikes:
  - title: mountain bikes
  - title: road bikes
  - title: hybrid bikes
```

Markdown + Liquid:

```
{% for item in site.data.samplelist.bikes %}
* {{item.title}}
{% endfor %}
```

Result:

Example 4: List items

YAML:

```
salesteams:
- title: Regions
  subfolderitems:
    - location: US
    - location: Spain
    - location: France
```

Markdown + Liquid:

```
{% for item in site.data.samplelist.salesteams %}
<h3>{{item.title}}</h3>
<ul>
{% for entry in item.subitems %}
<li>{{entry.location}}</li>
{% endfor %}
</ul>
{% endfor %}
```

Result:

```

- title: Group 1
  subfolderitems:
    - page: Thing 1
    - page: Thing 2
    - page: Thing 3
- title: Group 2
  subfolderitems:
    - page: Piece 1
    - page: Piece 2
    - page: Piece 3
- title: Group 3
  subfolderitems:
    - page: Widget 1
    - page: Widget 2 it's
    - page: Widget 3

```

item

item.subitems

-

[1]

Example 5: Table of contents

YAML:

```

toc:
  - title: Group 1
    subfolderitems:
      - page: Thing 1
      - page: Thing 2
      - page: Thing 3
  - title: Group 2
    subfolderitems:
      - page: Piece 1
      - page: Piece 2
      - page: Piece 3
  - title: Group 3
    subfolderitems:
      - page: Widget 1
      - page: Widget 2 it's
      - page: Widget 3

```

Markdown + Liquid:

```

{% for item in site.data.samplelist.toc %}
<h3>{{item.title}}</h3>
<ul>
{% for entry in item.subfolderitems %}
<li>{{entry.page}}</li>
{% endfor %}
</ul>
{% endfor %}

```

Result:

Example 6: Variables

YAML:

```
something: &hello Greetings earthling!  
myref: *hello
```

Markdown:

```
{{ site.data.samplelist.myref }}
```

Result:

```
&hello           Greetings earthling!
*hello
```

Example 7: Positions in lists

YAML:

```
about:
- zero
- one
- two
- three
```

Markdown:

```
{{ site.data.samplelist.about[0] }}
```

Result:

```
[0]
```

Example 8: Properties from list items at specific positions

YAML:

```
numbercolors:
  - zero:
    properties: red
  - one:
    properties: yellow
  - two:
    properties: green
  - three:
    properties: blue
```

Markdown + Liquid:

```
{{ site.data.samplelist.numbercolors[0].properties }}
```

Result:

```
red
yellow
green
blue
```

Example 9: Conditions

YAML:


```
mypages:
- section1: Section 1
  audience: developers
  product: acme
  url: facebook.com
- section2: Section 2
  audience: writers
  product: acme
  url: google.com
- section3: Section 3
  audience: developers
  product: acme
  url: amazon.com
- section4: Section 4
  audience: writers
  product: gizmo
  url: apple.com
- section5: Section 5
  audience: writers
  product: acme
  url: microsoft.com
```

Markdown + Liquid:

```
{% for sec in site.data.samplelist.mypages %}
{% if sec.audience == "writers" %}
* {{sec.url}}
{% endif %}
{% endfor %}
```

Result:

```

- [facebook.com](facebook.com)
- [google.com](google.com)
- [amazon.com](amazon.com)
- [apple.com](apple.com)
- [microsoft.com](microsoft.com)

audience      writers      product
```

```
{% for sec in site.data.samplelist.mypages %}
{% if sec.audience == "writers" and sec.product == "gizmo" %}
* {{sec.url}}
{% endif %}
{% endfor %}
```

More resources

(<http://www.tournemille.com/blog/How-to-create-data-driven-navigation-in-Jekyll/>)

Tags

Summary:

Add a tag to a page

For example, to add the `tags` tag to the `release_notes_2_0/` page:

```
---
title: 2.0 Release Notes
permalink: release_notes_2_0/
tags: [formatting, single_sourcing]
---
```

Tags overview

Note:

`posts.tags.tagname` returns an array of tag names for the page. For example, if the page has the tags `formatting` and `single_sourcing`, the array will be `["formatting", "single_sourcing"]`.

`posts.tags` returns an array of tag objects for the page. For example, if the page has the tags `formatting` and `single_sourcing`, the array will be `[{"tagname": "formatting", "count": 1}, {"tagname": "single_sourcing", "count": 1}]`.

Note:

The `tags` tag is a `single_sourcing` tag. It is used to format the page content. For example, the `tags` tag is used to format the `tags` tag in the `tags` tag.

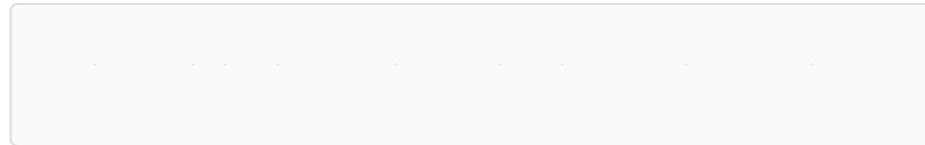
Setting up tags

```
allowed-tags:
  - getting_started
  - overview
  - formatting
  - publishing
  - single_sourcing
  - special_layouts
  - content types
```

```
---
title: "Collaboration pages"
tagName: collaboration
search: exclude
permalink: tag_collaboration/
sidebar: mydoc_sidebar
---
{% include taglogic.html %}
```

Note:

When you use the `mydoc_sidebar` tag, you must include the `mydoc_sidebar` tag in the `mydoc_sidebar` tag. This is done by adding the following line to the `mydoc_sidebar` tag:



```
<div class="tags">
  {% if page.tags != null %}
  <b>Tags: </b>
  {% assign projectTags = site.data.tags.allowed-tags %}
  {% for tag in page.tags %}
  {% if projectTags contains tag %}
  <a href="{{ "/" | prepend: site.baseurl | append: tag }}" class="btn btn-default navbar-btn cursorNorm" role="button">{{page.tagName}} {{tag}}</a>
  {% endif %}
  {% endfor %}
  {% endif %}
</div>
```

btn-info

Retrieving pages for a specific tag

Getting started pages:

```
<ul>
{% for page in site.pages %}
{% for tag in page.tags %}
{% if tag == "getting_started" %}
<li><a href="{{page.url | prepend: '..'}}">{{page.titl
e}}</a></li>
{% endif %}
{% endfor %}
{% endfor %}
</ul>
```

sort

Getting started pages:

```
<ul>
{% assign sorted_pages = (site.pages | sort: 'title') %}
{% for page in sorted_pages %}
{% for tag in page.tags %}
{% if tag == "getting_started" %}
<li><a href="{{page.url | prepend: '..'}}">{{page.titl
e}}</a></li>
{% endif %}
{% endfor %}
{% endfor %}
</ul>
```

Efficiency

```
for .
for .
for .
```

```
posts.tag.tagname
```

```
for
```

Empty tags?

```
tags: []
```

Remembering the right tags

Series

Summary:

Using series for pages

1. Create the series button

```
<div class="seriesContext">
  <div class="btn-group">
    <button type="button" data-toggle="dropdown" class="btn btn-primary dropdown-toggle">Series Demo <span class="caret"></span></button>
    <ol class="dropdown-menu">
      {% assign pages = site.pages | sort:"weight" %}
      {% for p in pages %}
      {% if p.series == "ACME series" %}
      {% if p.url == page.url %}
      <li class="active"> → {{p.weight}}. {{p.title}}</li>
      {% else %}
      <li>
        <a href="{{p.url | prepend: '..'}}">{{p.weight}}. {{p.title}}</a>
      </li>
      {% endif %}
      {% endif %}
      {% endfor %}
    </ol>
  </div>
</div>
```

⚠ Warning:

2. Create the “next” include

```
<p>{% assign series_pages = site.tags.series_acme %}  
  {% for p in pages %}  
    {% if p.series == "ACME series" %}  
      {% assign nextTopic = page.weight | plus: "1" %}  
      {% if p.weight == nextTopic %}  
        <a href="{{p.url | prepend: '..'}}"><button type="button" c  
lass="btn btn-primary">Next: {{p.weight}}  {{p.title}}</butto  
n></a>  
      {% endif %}  
    {% endif %}  
  {% endfor %}  
</p>
```

3. Add the correct frontmatter to each of your series pages

```
series: "ACME series"  
weight: 1.0
```

```
plus: "1.0"    plus: "0.1"
```

```
    {{p.weight}}  
    {{p.weight}}
```

4. Add links to the series button and next button on each page.

```
<!-- your frontmatter goes here -->  
  
{% include custom/series_acme.html %}  
  
<!-- your page content goes here ... -->  
  
{% include custom/series_acme_next.html %}
```

Changing the series drop-down color

primary

btn-primary btn-info btn-warning

Using a collection with your series

Tooltips

Summary:

Creating tooltips

There are two ways to create a tooltip in mydoc product. The first way is to use the `data-toggle="tooltip"` attribute on the element you want to create a tooltip for. The second way is to use the `data-original-title` attribute on the element you want to create a tooltip for.

```
<a href="#" data-toggle="tooltip" data-original-title="{{site.data.glossary.jekyll_platform}}">Jekyll</a> is my favorite tool for building websites.</a>
```

Jekyll

Alerts

Summary:

About alerts

Alerts are used to highlight important information or to provide a warning. They can be used to draw attention to a specific piece of content, such as a note or a warning, or to provide a general warning about a potential issue.

Alerts

Alerts are used to highlight important information or to provide a warning. They can be used to draw attention to a specific piece of content, such as a note or a warning, or to provide a general warning about a potential issue.

```
{% include note.html content="This is my note. All the content I type here is treated as a single paragraph." %}
```

Alerts are used to highlight important information or to provide a warning. They can be used to draw attention to a specific piece of content, such as a note or a warning, or to provide a general warning about a potential issue.

Note:

Alerts are used to highlight important information or to provide a warning. They can be used to draw attention to a specific piece of content, such as a note or a warning, or to provide a general warning about a potential issue.

Property	description


```
{% include note.html content="This is my note. All the content  
I type here is treated as a single paragraph. <br/><br/> Now  
I'm typing on a new line." %}
```

Note:

Types of alerts available

Note:

Tip:

Warning:

Important:

```

1  <div markdown="span" class="alert alert-info" role="alert"><i c
2  lass="fa fa-info-circle"></i> <b>Note:</b> {{include.conten
3  t}}</div>
4
5  </div>
6
7  </div>
8
9  </div>
10
11 </div>

```

```

<div markdown="span" class="alert alert-info" role="alert"><i c
lass="fa fa-info-circle"></i> <b>Note:</b> {{include.conten
t}}</div>

```

```

1  content="This is my note."
2  {{include.content}}}
3
4  </div>
5
6  </div>
7
8  </div>
9
10 </div>

```

(<http://jekyll.tips/jekyll-casts/includes/>)

(https://www.youtube.com/watch?v=TJcn_PJ2100)

Callouts

```

1  <div markdown="span" class="callout callout-info" role="callout">
2  <i class="fa fa-info-circle"></i> {{include.callout.content}}
3  </div>
4
5  </div>
6
7  </div>
8
9  </div>
10
11 </div>

```

```

{% include callout.html content="This is my callout. It has a b
order on the left whose color you define by passing a type para
meter. I typically use this style of callout when I have more i
nformation that I want to share, often spanning multiple paragr
aphs. " type="primary" %}

```

```

1  <div markdown="span" class="callout callout-info" role="callout">
2  <i class="fa fa-info-circle"></i> {{include.callout.content}}
3  </div>
4
5  </div>
6
7  </div>
8
9  </div>
10
11 </div>

```

Property	description

Property	description				
	default	primary	success	info	danger
					warning

...

danger

default

primary

success

info

warning


```
{% include callout.html content="**Important information**:  
This is my callout. It has a border on the left whose color you de  
fine by passing a type parameter. I typically use this style o  
f callout when I have more information that I want to share, of  
ten spanning multiple paragraphs. <br/><br/>Here I am starting  
a new paragraph, because I have lots of information to share. Y  
ou may wonder why I'm using line breaks instead of paragraph ta  
gs. This is because Kramdown processes the Markdown here as a s  
pan rather than a div (for whatever reason). Be grateful that y  
ou can be using Markdown at all inside of HTML. That's usually  
not allowed in Markdown syntax, but it's allowed here." type="p  
rimary" %}
```

Important information: This is my callout. It has a border on the left whose color you define by passing a type parameter. I typically use this style of callout when I have more information that I want to share, often spanning multiple paragraphs.

Here I am starting a new paragraph, because I have lots of information to share. You may wonder why I'm using line breaks instead of paragraph tags. This is because Kramdown processes the Markdown here as a span rather than a div (for whatever reason). Be grateful that you can be using Markdown at all inside of HTML. That's usually not allowed in Markdown syntax, but it's allowed here." type="primary" %}

Important information

Use Liquid variables inside parameters with includes

```
{% include callout.html content=variable %}
```

```
{% include note.html content="The {{site.company}} is pleased to  
announce an upcoming release." %}
```

Liquid Exception: Invalid syntax for include tag. File contains invalid characters or sequences: ... Valid syntax: {% include file.ext param='value' param2='value' %}

capture

company_name

```
company_name: Your company
```

```
{% capture company_note %}The {{site.company_name}} company is  
pleased to announce an upcoming release.{% endcapture %}
```

company_note include

```
{% include note.html content=company_note}
```

Note:

{{site.company_name}}

{{page.company_name}}

Markdown inside of callouts and alerts

```
markdown="span" data-bbox="111 111 281 128" data-label="Text">

markdown="span"


```

Validity checking

```
Liquid Exception: Invalid syntax for include tag: content="This is my info type callout. It has a border on the left whose color you define by passing a type parameter. type="info" Valid syntax: {% include file.ext param='value' param2='value' %} in mydoc/mydoc_alerts.md
```

Blast a warning to users on every page

Icons

Summary:

Font icon options

See Font Awesome icons available

(<http://fontawesome.github.io/Font-Awesome/icons/>)

fa-3x

fa-2x

```
<i class="fa fa-camera-retro"></i> normal size (1x)  
<i class="fa fa-camera-retro fa-lg"></i> fa-lg  
<i class="fa fa-camera-retro fa-2x"></i> fa-2x  
<i class="fa fa-camera-retro fa-3x"></i> fa-3x  
<i class="fa fa-camera-retro fa-4x"></i> fa-4x  
<i class="fa fa-camera-retro fa-5x"></i> fa-5x
```



```

<div class="alert alert-danger" role="alert"><i class="fa fa-exclamation-circle"></i> <b>Warning: </b>This is a special warning message.

```

```

<div class="alert alert-danger" role="alert"><i class="fa fa-exclamation-circle"></i> <b>Warning: </b>This is a special warning message.

```

```

<div class="alert alert-info" role="alert"><i class="fa fa-info-circle"></i> <b>Note: </b>

```

```

<div class="alert alert-warning" role="alert"><i class="fa fa-warning"></i> <b>Important: </b>

```

```

tip: '<div class="alert alert-success" role="alert"><i class="fa fa-check-square-o"></i> <b>Tip: </b>'
note: '<div class="alert alert-info" role="alert"><i class="fa fa-info-circle"></i> <b>Note: </b>'
important: '<div class="alert alert-warning" role="alert"><i class="fa fa-warning"></i> <b>Important: </b>'
warning: '<div class="alert alert-danger" role="alert"><i class="fa fa-exclamation-circle"></i> <b>Warning: </b>'
end: '</div>'

```

```

callout_danger: '<div class="bs-callout bs-callout-danger">'
callout_default: '<div class="bs-callout bs-callout-default">'
callout_primary: '<div class="bs-callout bs-callout-primary">'
callout_success: '<div class="bs-callout bs-callout-success">'
callout_info: '<div class="bs-callout bs-callout-info">'
callout_warning: '<div class="bs-callout bs-callout-warning">'

```

```

hr_faded: '<hr class="faded"/>'
hr_shaded: '<hr class="shaded"/>'

```

```

<div class="alert alert-info" role="alert"><i class="fa fa-info-circle"></i> <b>Note: </b>

```

```

{% include note.html content="Add your note here." %}

```

```
{% include tip.html content="Add your tip here." %}
```

```
{% include important.html content="Add your important info here." %}
```

```
{% include warning.html content="Add your warning here." %}
```

 Note:

 Tip:

 Important:

 Warning:

Creating your own combinations

```
<div class="alert alert-success" role="alert"><i class="fa fa-download fa-lg"></i> This is a special tip about some file to do wnload....</div>
```



(<http://fontawesome.github.io/Font-Awesome/icons/>)

```
.fa-10x{font-size:1700%;}
```

fa-10x

Glyphicon icons available

(<http://getbootstrap.com/components/#glyphicons>)

(<http://stackoverflow.com/questions/24960201/how-do-i-make-glyphicons-bigger-change-size>)

```
.gi-2x{font-size: 2em;}  
.gi-3x{font-size: 3em;}  
.gi-4x{font-size: 4em;}  
.gi-5x{font-size: 5em;}
```

gi-5x

```
<span class="glyphicon glyphicon-globe gi-5x"></span>
```



span

i

```
<span class="glyphicon glyphicon-download"></span>
```



```
<span class="glyphicon glyphicon-download gi-3x"></span>
```

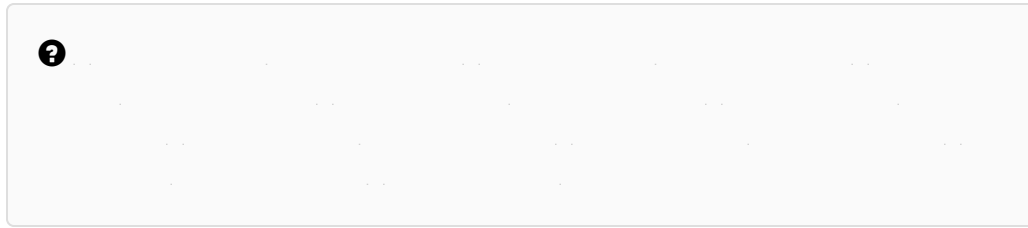


```
<div class="alert alert-danger" role="alert">
  <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
  <b>Error:</b> Enter a valid email address
</div>
```

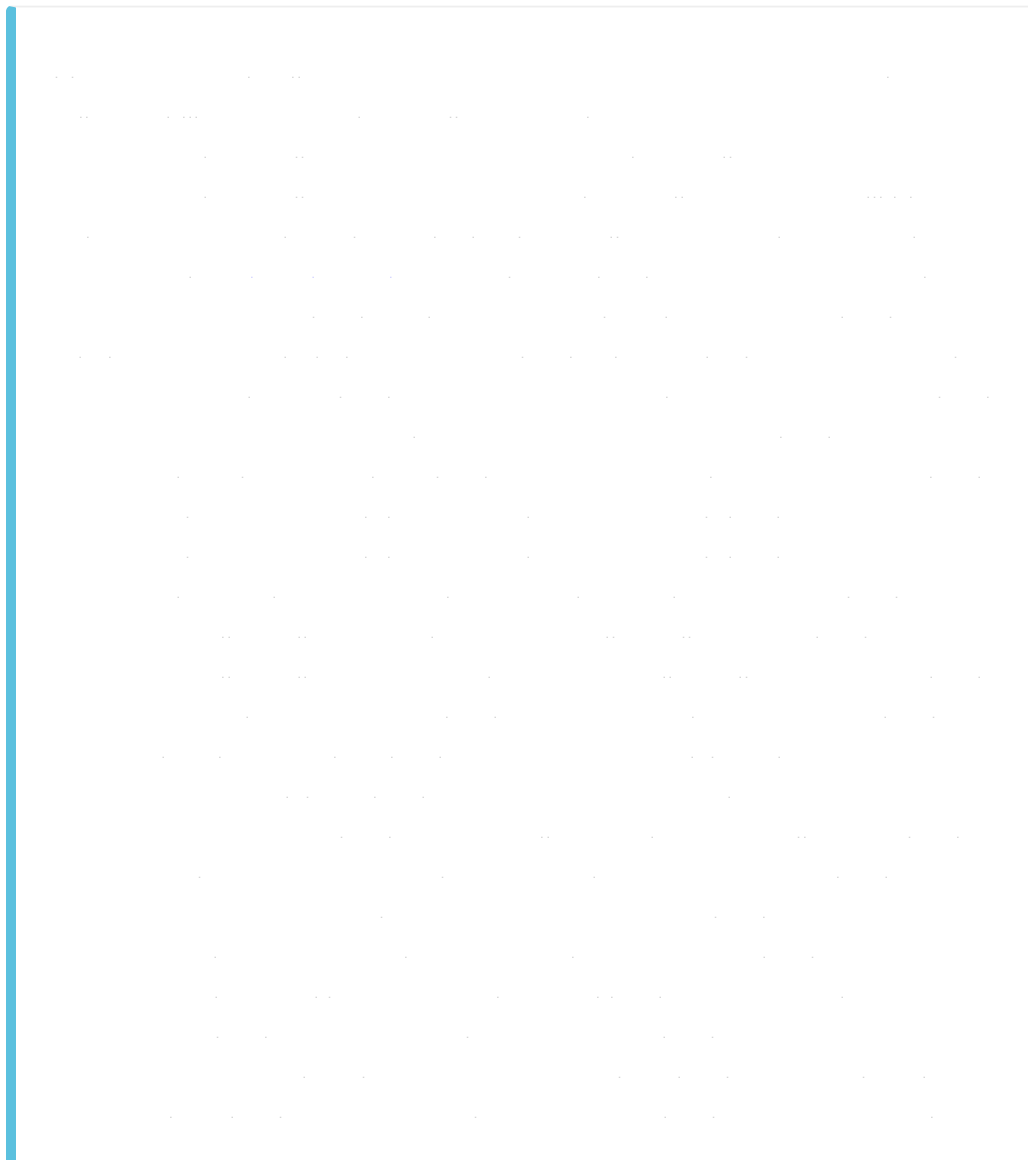
 **Error:**

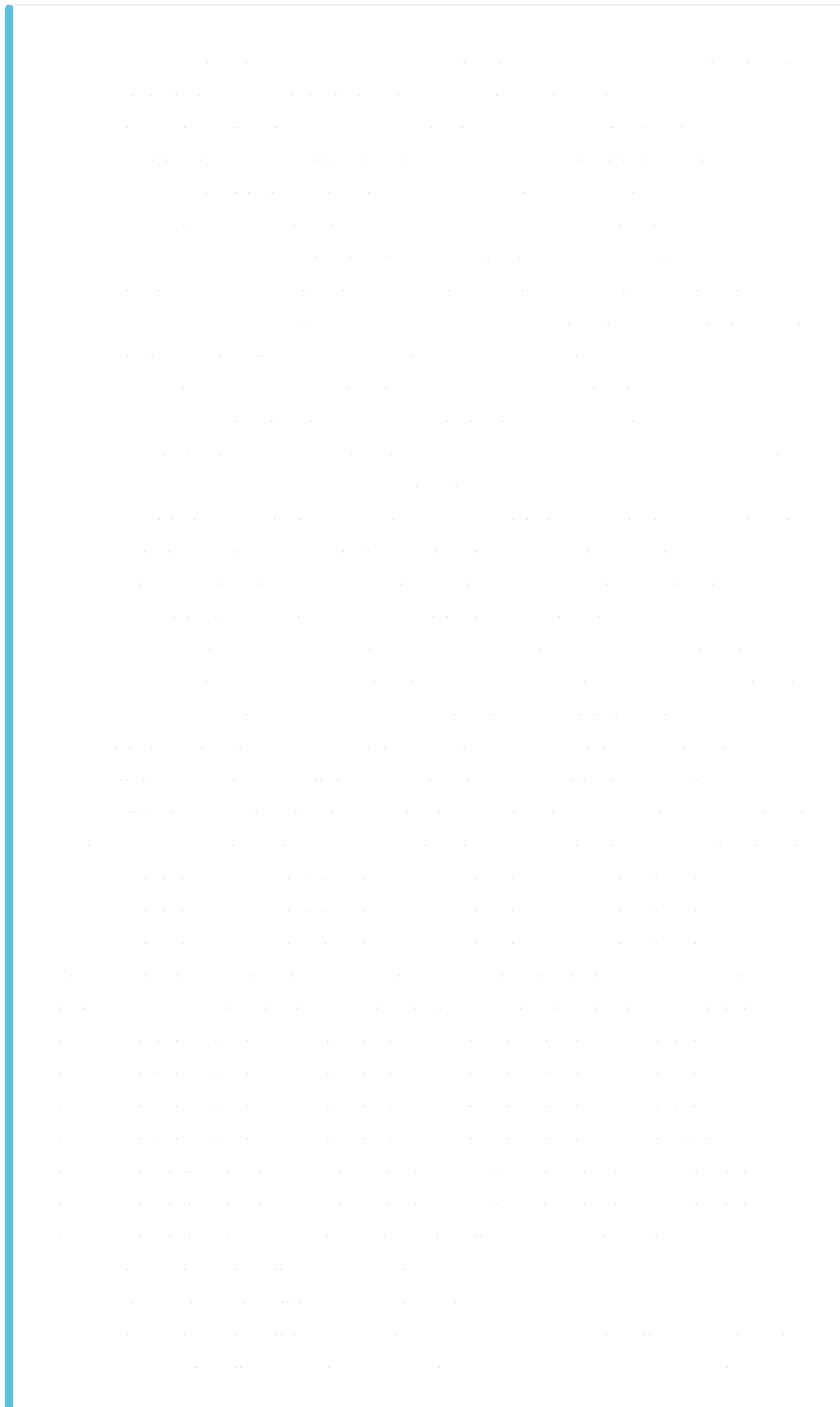
Callouts

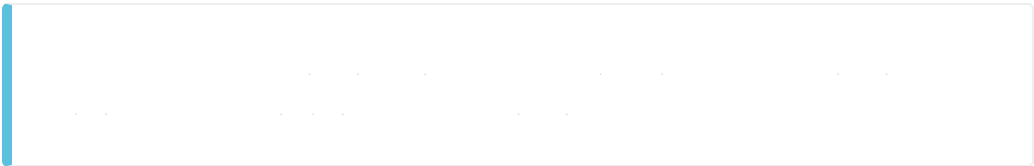
```
<div class="bs-callout bs-callout-info">
  This is a special info message. This is a special info message.
  This is a special info message. This is a special info message.
  This is a special info message. This is a special info message.
  This is a special info message. This is a special info message.
  This is a special info message. This is a special info message.
  This is a special info message. </div>
```

`{{site.data.alerts.callout_info}}`This is a special callout information message.







Images

Summary:

Images are included in the documentation using the `{% include image.html file="image.png" url="http://example.com" alt="Image alt text" caption="Image caption" %}` syntax. The `file` parameter is the name of the image file, and the `url` parameter is the URL of the image. The `alt` parameter is the alt text for the image, and the `caption` parameter is the caption for the image.

Image Include Template

The image include template is a Liquid template that is used to include an image in the documentation. It is located in the `layouts/_includes/image.html` file. The template takes the following parameters:

- `file`: The name of the image file.
- `url`: The URL of the image.
- `alt`: The alt text for the image.
- `caption`: The caption for the image.

```
{% include image.html file="jekyll.png" url="http://jekyllrb.com" alt="Jekyll" caption="This is a sample caption" %}
```

Property	description



This is a sample caption

Inline image includes

```
Click the Android SDK Manager button {% include inline_image.html
file="androidsdkmanagericon.png" alt="SDK button" %}
```

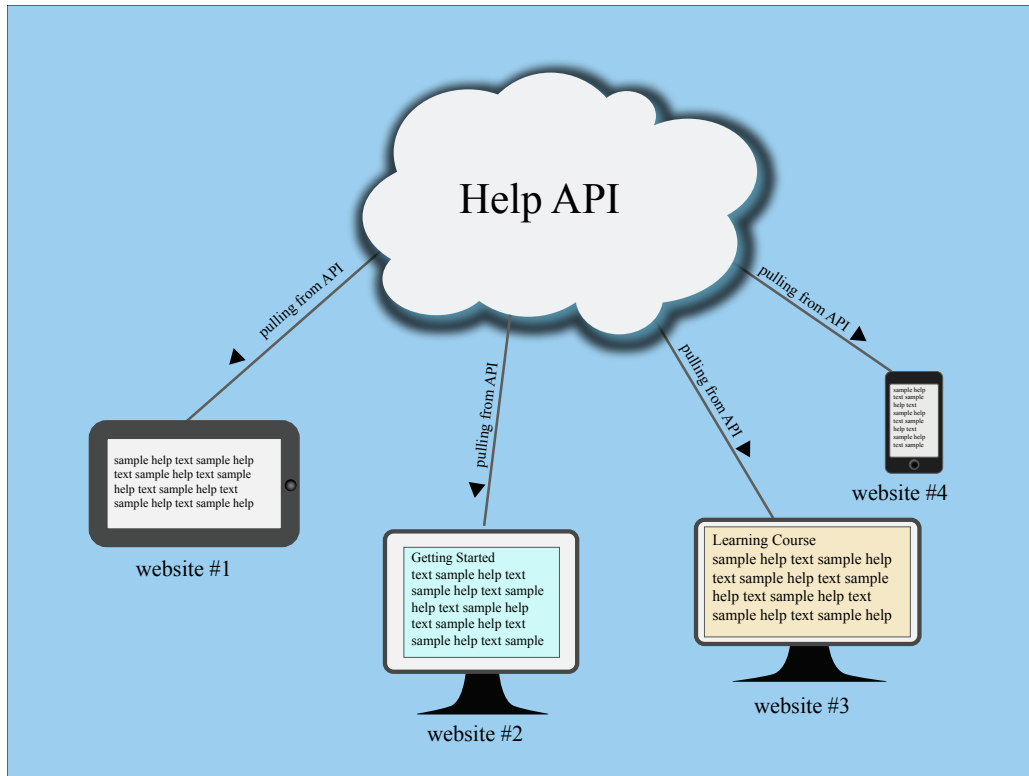
Android SDK Manager



Property	description

SVG Images

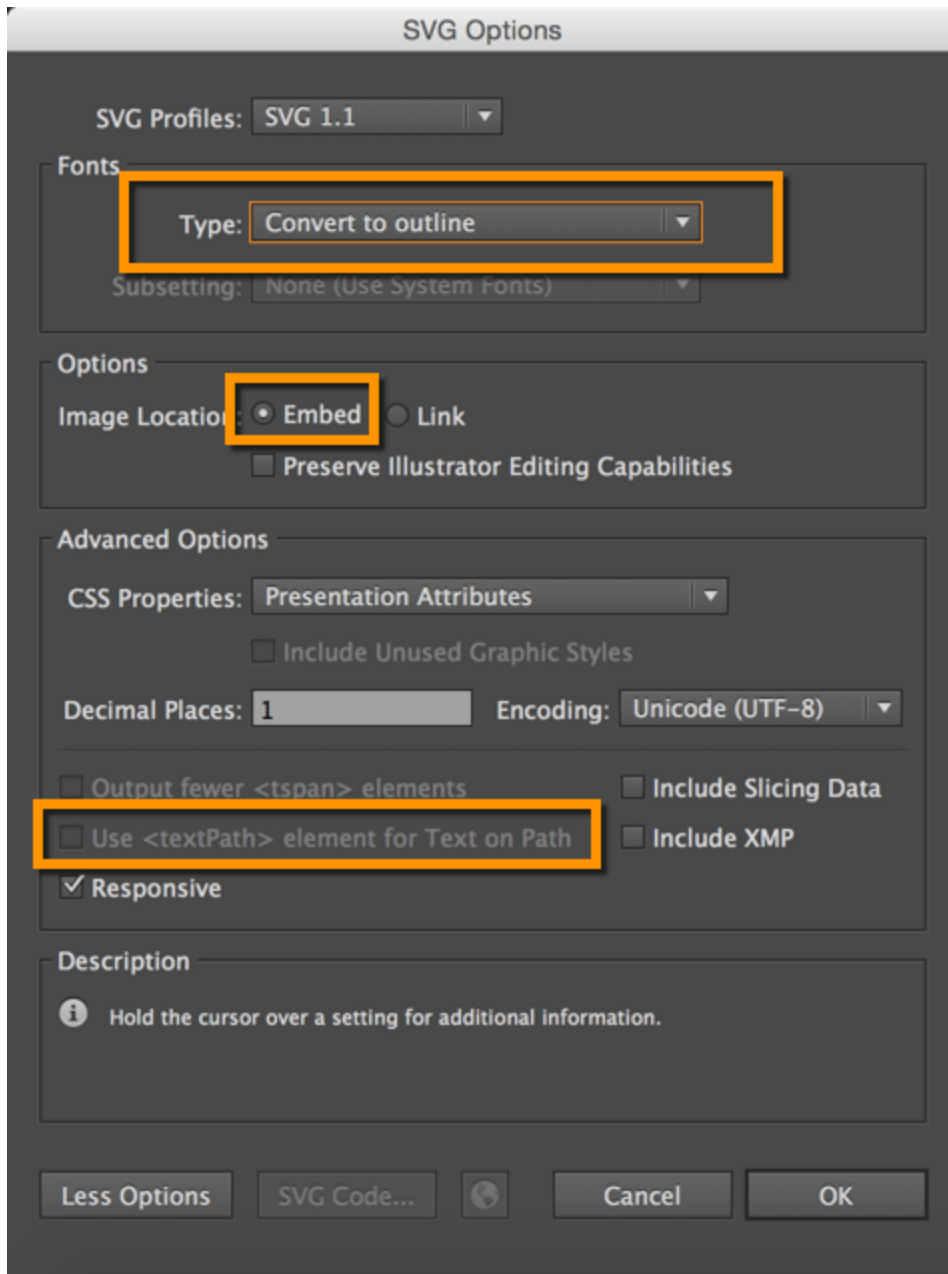
```
{% include image.html file="helpapi.svg" url="http://idratherbe
writing.com/documentation-theme-jekyll/mydoc_help_api/" alt="Bu
ilding a Help API" caption="A help API provides a JSON file at
a web URL with content that can be pulled into different target
s" max-width="600" %}
```



A help API provides a JSON file at a web URL with content that can be pulled into different targets

(<https://gist.github.com/larrybotha/7881691>)

```
/*
 * Let's target IE to respect aspect ratios and sizes for img tags
 * containing SVG files
 *
 * [1] IE9
 * [2] IE10+
 */
/* 1 */
.ie9 img[src$=".svg"] {
    width: 100%;
}
/* 2 */
@media screen and (-ms-high-contrast: active), (-ms-high-contrast:
none) {
    img[src$=".svg"] {
        width: 100%;
    }
}
```



Code samples

Summary:

Code Samples

```
```js
console.log('hello');
```
```

Result:

```
console.log('hello');
```

js

(<https://github.com/jneen/rouge/wiki/list-of-supported-languages-and-lexers>)

Labels

Summary:

About labels

Labels are used to highlight important information or to indicate the status of a component. They are available in six different styles: Default, Primary, Success, Info, Warning, and Danger.

```
<span class="label label-default">Default</span>  
<span class="label label-primary">Primary</span>  
<span class="label label-success">Success</span>  
<span class="label label-info">Info</span>  
<span class="label label-warning">Warning</span>  
<span class="label label-danger">Danger</span>
```

Default **Primary** **Success** **Info** **Warning** **Danger**

Labels are used to highlight important information or to indicate the status of a component. They are available in six different styles: Default, Primary, Success, Info, Warning, and Danger.

Links

Summary:

Create an external link

[Google](http://google.com)

Linking to internal pages

```
[Icons](mydoc_icons.html)
```

Automated links

```
url
```

Note:

```
permalink      permalink
               somefile.html
               somefile.html
```

```
sidebars
```

```
sidebars:
- home_sidebar
- mydoc_sidebar
- product1_sidebar
- product2_sidebar
- other
```

```
{% include links.html %}
```

See the [Icon][mydoc_icons] file.

Automated links to headings on pages

Example: mydoc_pages.html#someIdTag

```
## Some heading {#someheading}
```

Example: mydoc_pages.html#someIdTag

```
- title: Some link bookmark  
  url: /mydoc_pages.html#someIdTag
```

Example: mydoc_pages.html#someIdTag

```
This is [Some link][mydoc_pages.html#someIdTag].
```

Result:

Example: mydoc_pages.html#someIdTag

Navtabs

Summary:

Common uses

Navtabs demo



Profile

Code

```
<ul id="profileTabs" class="nav nav-tabs">
  <li class="active"><a href="#profile" data-toggle="tab">Pro
file</a></li>
  <li><a href="#about" data-toggle="tab">About</a></li>
  <li><a href="#match" data-toggle="tab">Match</a></li>
</ul>
<div class="tab-content">
<div role="tabpanel" class="tab-pane active" id="profile">
  <h2>Profile</h2>
<p>Praesent sit amet fermentum leo....</p>
</div>

<div role="tabpanel" class="tab-pane" id="about">
  <h2>About</h2>
  <p>Lorem ipsum ...</p></div>

<div role="tabpanel" class="tab-pane" id="match">
  <h2>Match</h2>
  <p>Vel vehicula ....</p>
</div>
</div>
```

Design constraints

Appearance in the mini-TOC

Must use HTML

Match up ID tags

```
href          id          div
href="#acme"      acme
<div role="tabpanel" class="tab-pane" id="acme">
```

Set an active tab

```
<div role="tabpanel" class="tab-pane active" id="acme">
```

Sets a cookie

(<http://stackoverflow.com/questions/10523433/how-do-i-keep-the-current-tab-active-with-twitter-bootstrap-after-a-page-reload>)

Functionality to implement

Tables

Summary:

Multimarkdown Tables

```
| Priority apples | Second priority | Third priority |
|-----|-----|-----|
| ambrosia | gala | red delicious |
| pink lady | jazz | macintosh |
| honeycrisp | granny smith | fuji |
```

Result:

Priority apples	Second priority	Third priority

Note:

HTML Tables

```
markdown="1"td
```

```

<table>
<colgroup>
<col width="30%" />
<col width="70%" />
</colgroup>
<thead>
<tr class="header">
<th align="left">Field</th>
<th align="left">Description</th>
</tr>
</thead>
<tbody>
<tr>
<td align="left" markdown="1">First column **fields**</td>
<td align="left" markdown="1">Some descriptive text. This is a
markdown link to [Google](http://google.com). Or see [some lin
k][mydoc_tags].</td>
</tr>
<tr>
<td align="left" markdown="1">Second column **fields**</td>
<td align="left" markdown="1">Some more descriptive text. Here
we have a Markdown-formatted list:

* first item
* second item
* third item
</td>
</tr>
</tbody>
</table>

```

Result:

jQuery datatables

jQuery datatables is a plugin that allows you to turn a table into a datatables. It is a very powerful plugin that can be used to create a datatables with many features. You can find more information about it on the jQuery datatables website (<https://www.datatables.net/>).

Here is an example of how to use it:

```

<table>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
<tr>
<td>First column **fields**</td>
<td>Some descriptive text. This is a
markdown link to [Google](http://google.com). Or see [some lin
k][mydoc_tags].</td>
</tr>
<tr>
<td>Second column **fields**</td>
<td>Some more descriptive text. Here
we have a Markdown-formatted list:

* first item
* second item
* third item
</td>
</tr>
</table>

```

And here is the jQuery datatables code:

```

<script>
$(document).ready(function() {
    $('#table').dataTable({
        datatables: true
    });
});
</script>

```

```
<script>
$(document).ready(function(){

    $('table.display').DataTable( {
        paging: true,
        stateSave: true,
        searching: true
    }
    );
});
</script>
```

(<https://www.datatables.net/manual/options>)

display

table.display

```
<table id="sampleTable" class="display">
  <thead>
    <tr>
      <th>Parameter</th>
      <th>Description</th>
      <th>Type</th>
      <th>Default Value</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Parameter 1</td>
      <td>Sample description
      </td>
      <td>Sample type</td>
      <td>Sample default value</td>
    </tr>
    <tr>
      <td>Parameter 2</td>
      <td>Sample description
      </td>
      <td>Sample type</td>
      <td>Sample default value</td>
    </tr>
    <tr>
      <td>Parameter 3</td>
      <td>Sample description
      </td>
      <td>Sample type</td>
      <td>Sample default value</td>
    </tr>
    <tr>
      <td>Parameter 4</td>
      <td>Sample description
      </td>
      <td>Sample type</td>
      <td>Sample default value</td>
    </tr>
  </tbody>
</table>
```

Field	Description
fields	http://google.com
fields	

Food	Description	Category	Sample type

(<https://www.datatables.net/manual/options>)

i Note:

Syntax highlighting

Summary:

About syntax highlighting

MyDoc uses `highlight` to highlight code blocks in the documentation. The code blocks are highlighted using the `highlight` command.

```
```ruby
def foo
 puts 'foo'
end
```
```

```
def foo
  puts 'foo'
end
```

The `highlight` command is used to highlight the code blocks. The command is used as follows:

```
{% highlight ruby %}
def foo
  puts 'foo'
end
{% endhighlight %}
```

```
def foo
  puts 'foo'
end
```

```
highlighter: rouge
```

Available lexers

```
ruby
```


Workflow maps

Summary:

Workflow maps overview

Simple workflow maps

```

<div id="userMap">
<div class="content"><a href="p2_sample1.html"><div clas
s="box box1">Connect to ADB</div></a></div>
<div class="arrow">→</div>
<div class="content"><a href="p2_sample2.html"><div clas
s="box box2">Download and Build the Starter Kit</di
v></a></div>
<div class="arrow">→</div>
<div class="content"><a href="p2_sample3.html"><div clas
s="box box3">Take a Tour</div></a></div>
<div class="arrow">→</div>
<div class="content"><a href="p2_sample4.html"><div clas
s="box box4">Load Your Widgets</div></a></div>
<div class="arrow">→</div>
<div class="content"><a href="p2_sample5.html"><div clas
s="box box5">Query for Something</div></a></div>
<div class="clearfix"></div>
</div>

```

```

---
title: Sample 1 Topic
keywords: sample
summary: "This is just a sample topic..."
sidebar: product2_sidebar
permalink: p2_sample1
folder: product2
simple_map: true
map_name: usermap
box_number: 1
---

```

```
simple_map      true
```

```
{% if page.simple_map == true %}

<script>
    $(document).ready ( function(){
        $('#.box{{page.box_number}}').addClass('active');
    });
</script>

{% include custom/{{page.map_name}}.html %}

{% endif %}
```

[illegible]

Complex workflow maps

```
<button type="button" class="btn btn-default btn-lg modalButton3" data-toggle="modal" data-target="#myModal3">Publish your app
```

```
<div class="modal fade" id="myModal3" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
```

```
---
title: Sample 6 Topic
keywords: sample
summary: "This is just a sample topic..."
sidebar: product2_sidebar
permalink: p2_sample6
complex_map: true
map_name: usermapcomplex
box_number: 1
toc: false
folder: product2
---
```

```
complex_map      true
```

```
map
```

```
true
```

```
<script>
    $(document).ready ( function(){
        $('.modalButton{{page.box_number}}').addClass('active');
    });
</script>

{% include custom/{{page.map_name}}.html %}

{% endif %}
``````
```

## Commenting on files

### Summary:

## About the review process

```
{% unless jekyll.environment == "production" %}

 {% if site.github_editme_path %}

 <a target="_blank" href="https://github.com/{{site.github_e
ditme_path}}/{{page.folder}}{{page.url | append: ".md"}}{% endi
f %}" class="btn btn-default githubEditButton" role="butto
n"><i class="fa fa-github fa-lg"></i> Edit me
```

(<http://jekyllrb.com/docs/configuration/>)

```
JEKYLL_ENV=production jekyll serve
```

```
github_editme_path
```

## Add reviewers as collaborators

## Settings

**i Note:**

## Workflow

**! Warning:**

## Prose.io

(<http://prose.io>)



## Build arguments

### Summary:

## How to build Jekyll sites

```
jeekyll build
```

```
serve
```

```
jeekyll serve
```

```
_site
```

```
jeekyll serve --config configs/myspecialconfig.yml --destination
n ../doc_outputs
```

```
configs/myspecialconfig.yml
```

```
_config.yml
```

```
../doc_outputs
```

## Shortcuts for the build arguments

```
. myscript.sh
```

## Stop a server

**Ctrl+C**

```
ps aux | grep jekyll
```

```
kill -9 22298
```

```
kill -9 $(ps aux | grep '[j]ekyll' | awk '{print $2}')
```

GeneralColorsTextWindowTerminalSessionKeysAdvanced

Basics

Name: Kill all Jekyll

Shortcut key: ^⌘

Tags: Example: linux, dark bg, tall window

Command

☒ Login shell

☐ Command:

Send text at start: kill -9 \$(ps aux | grep '[j]ekyll' | awk '{print \$2}')

Working Directory

☐ Home directory

☐ Reuse previous session's directory

☒ Directory: /Users/tjohnson/projects/docs

☐ Advanced Configuration Edit...

URL Schemes

Schemes handled: Select URL Schemes...

# Themes

## Summary:

## Theme options

There are two options for the theme file. The first is to use the default theme file, which is `theme-green.css`. The second is to use a custom theme file, which is `theme-blue.css`. The theme file is located in the `css` directory of the `mydoc-pdf` package. The theme file is a CSS file that defines the look and feel of the PDF document. The theme file can be customized to match the look and feel of your organization's branding.

```
<link rel="stylesheet" href="/mydoc-pdf/css/theme-green.css">
```

## Theme differences

The theme file defines the look and feel of the PDF document. The theme file can be customized to match the look and feel of your organization's branding. The theme file is a CSS file that defines the look and feel of the PDF document. The theme file can be customized to match the look and feel of your organization's branding.

# Generating PDFs

## Summary:

PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations.

## PDF overview

PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations.

## Demo

PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations.

## 1. Set up Prince

PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations. PrinXML is a software tool that generates PDFs from XML documents. It is a powerful tool for creating PDFs from XML data, and it is used by many companies and organizations.

## 2. Create a new configuration file for each of your PDF targets

For example, to create a configuration file for a PDF target, you can use the following configuration:

```
destination: _site/
url: "http://127.0.0.1:4010"
baseurl: "/mydoc-pdf"
port: 4010
output: pdf
product: mydoc
print_title: Jekyll theme for documentation – mydoc product
print_subtitle: version 5.0
output: pdf
defaults:
 -
 scope:
 path: ""
 type: "pages"
 values:
 layout: "page_print"
 comments: true
 search: true
```

### **Note:**

The `output` field in the configuration file is used to specify the output format for the target. In this example, the output format is `pdf`.

`page_print`

```

 output: pdf
 site.output pdf web pdf
```

```
print_subtitle : "Introduction"
print_title : "Introduction"
```

### 3. Make sure your sidebar\_doc.yml file has a titlepage.html and tocpage.html

```
titlepage.html
tocpage.html
```



**Note:**

```
output: web
output: pdf

output: pdf
```

## 4. Customize your headers and footers

```
youremail@domain.com
```

```
a[href]::after {
 content: " (page " target-counter(attr(href), page) ")"
}
```

```
a[href*="mailto"]::after, a[data-toggle="tooltip"]::after, a[hr
ef].noCrossRef::after {
 content: "";
}
```

**Tip:**

```
noCrossRef
```



```
a[href^="http:"]::after, a[href^="https:"]::after {
 content: " (" attr(href) ")";
}
```

```
@page {
 margin: 60pt 90pt 60pt 90pt;
 font-family: sans-serif;
 font-style:none;
 color: gray;
}
```

```

type: title

```

```

type: frontmatter

```

```

type: first_page

```

body

```
<body class="{% if page.type == "title"%}title{% elsif page.type == "frontmatter" %}frontmatter{% elsif page.type == "first_page" %}first_page{% endif %} print">
```

```
body.title { page: title }
```

```
body class="title"
@page title
```

```
@page title {
 @top-left {
 content: " ";
 }
 @top-right {
 content: " "
 }
 @bottom-right {
 content: " ";
 }
 @bottom-left {
 content: " ";
 }
}
```

type: frontmatter

```
body.frontmatter { page: frontmatter }
body.frontmatter {counter-reset: page 1}

@page frontmatter {
 @top-left {
 content: prince-script(guideName);
 }
 @top-right {
 content: prince-script(datestamp);
 }
 @bottom-right {
 content: counter(page, lower-roman);
 }
 @bottom-left {
 content: "youremail@domain.com"; }
}
```

counter(page, lower-roman)

```
body.first_page {counter-reset: page 1}

h1 { string-set: doctitle content() }

@page {
 @top-left {
 content: string(doctitle);
 font-size: 11px;
 font-style: italic;
 }
 @top-right {
 content: prince-script(datestamp);
 font-size: 11px;
 }

 @bottom-right {
 content: "Page " counter(page);
 font-size: 11px;
 }
 @bottom-left {
 content: prince-script(guideName);
 font-size: 11px;
 }
}
```

prince-script

```
<script>
 Prince.addScriptFunc("datestamp", function() {
 return "PDF last generated: July 10, 2016";
 });
</script>

<script>
 Prince.addScriptFunc("guideName", function() {
 return "Jekyll theme for documentation – mydoc product
User Guide";
 });
</script>
```

```
content: string(doctype);
```

```
content: "Page " counter(page);
```

```
--javascript
```

## 5. Customize the PDF script

```
echo 'Killing all Jekyll instances'
kill -9 $(ps aux | grep '[j]ekyll' | awk '{print $2}')
clear

echo "Building PDF-friendly HTML site for Mydoc ...";
jekyll serve --detach --config _config.yml,pdfconfigs/config_my
doc_pdf.yml;
echo "done";

echo "Building the PDF ...";
prince --javascript --input-list=_site/pdfconfigs/prince-list.t
xt -o _pdf/mydoc.pdf;
echo "done";
```

```
.. jekyll serve
```

```
.. --javascript
```

```
.. --input-list
```

```
.. -o
```

**Note:**

## 6. Add conditions for your new builds in the sidebarconfigs.html file

```
{% if site.product == "mydoc" %}
{% assign sidebar_pdf = site.data.sidebars.mydoc_sidebar.entries %}
{% endif %}

{% if site.product == "product1" %}
{% assign sidebar_pdf = site.data.sidebars.product1_sidebar.entries %}
{% endif %}

{% if site.product == "product2" %}
{% assign sidebar_pdf = site.data.sidebars.product2_sidebar.entries %}
{% endif %}
```

```
sidebar_pdf
```

## 7. Add a download button for the PDF

```
<button type="button" class="btn btn-default" aria-label="Left Align"> PDF Download</button>
```



## JavaScript conflicts

```
error: TypeError: value is not an object
```

```
output: pdf
```

```
{% unless site.output == "pdf" %}
javascript content here ...
{% endunless %}
```

```
unless
```

## Overriding Bootstrap Print Styles

```
@media print{*,:after,:before{color:#000!important;text-shado
w:none!important;background:0 0!important;-webkit-box-shadow:no
ne!important;box-shadow:none!important}}
```

```
*
```

```
@media print{*,:after,:before{/*color:#000!important;*/text-sha
dow:none!important;*/background:0 0!important*/;-webkit-box-sha
dow:none!important;box-shadow:none!important}
```

\*

pre

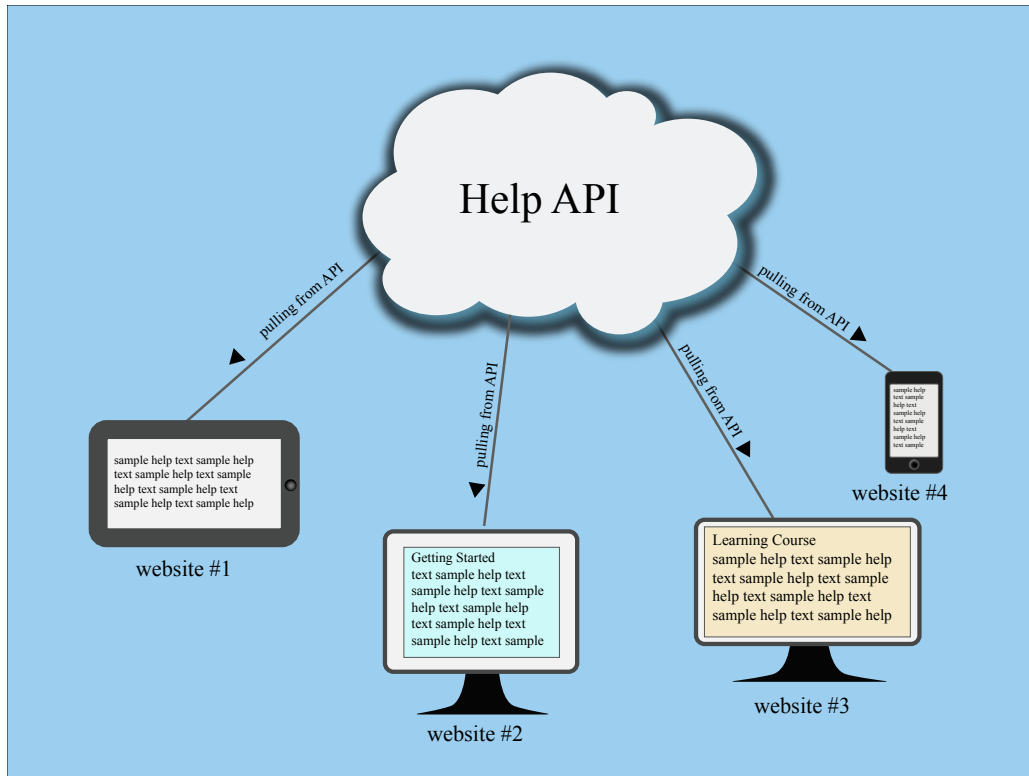


## Help APIs and UI tooltips

### Summary:

## Full code demo of content API

## Diagram overview



## 1. Create a “collection” for the help content

```
collections:
 tooltips:
 output: false
```

<pre> --- id: basketball product: mydoc output: true </pre>	<pre> --- id: basketball product: mydoc output: false </pre>
-------------------------------------------------------------	--------------------------------------------------------------

## 2. Create tooltip definitions in a YAML file

```

basketball: "Basketball is a sport involving two teams of five
players each competing to put a ball through a small circular r
im 10 feet above the ground. Basketball requires players to be
in top physical condition, since they spend most of the game ru
nning back and forth along a 94-foot-long floor."

```

## 3. Create pages in your collection

```

id: basketball
product: mydoc

```

```

```liquid
---
id: basketball
product: mydoc
---

{{site.data.definitions.basketball}}
```

```

## 4. Create a JSON file that loops through your collection pages

```

layout: none
search: exclude

{
 "entries":
 [
 {% for page in site.tooltips %}
 {% if page.product == "mydoc" %}
 {
 "id" : "{{ page.id }}",
 "body": "{{ page.content | strip_newlines | replace: '\',
 '\\\\' | replace: '\"', '\\\"' }}"
 } {% unless forloop.last %},{% endunless %}
 {% endif %}
 {% endfor %}
]
}

```

```

layout: none
search: exclude

{
 "entries":
 [
 {% for page in site.tooltips %}
 {% if page.product == "mydoc" %}
 {
 "id" : "{{ page.id }}",
 "body": "{{ page.content | strip_newlines | replace: '\',
 '\\\\' | replace: '\"', '\\\"' }}"
 } {% unless forloop.last %},{% endunless %}
 {% endif %}
 {% endfor %}
]
}

```

```
{
 "entries": [
 {
 "id": "baseball",
 "body": "Baseball is considered America's pasttime sport, though that may be more of a historical term than a current one. There's a lot more excitement about football than baseball. A baseball game is somewhat of a snooze to watch, for the most part."
 },
 {
 "id": "basketball",
 "body": "Basketball is a sport involving two teams of five players each competing to put a ball through a small circular rim 10 feet above the ground. Basketball requires players to be in top physical condition, since they spend most of the game running back and forth along a 94-foot-long floor."
 },
 {
 "id": "football",
 "body": "No doubt the most fun sport to watch, football also manages to accrue the most injuries with the players. From concussions to blown knees, football players have short sport lives."
 },
 {
 "id": "soccer",
 "body": "If there's one sport that dominates the world landscape, it's soccer. However, US soccer fans are few and far between. Apart from the popularity of soccer during the World Cup, most people don't even know the name of the professional soccer organization in their area."
 }
]
}
```

**Tip:**

(<https://google-styleguide.googlecode.com/svn/trunk/jsoncstyleguide.xml>)

```

type: getting_started

```

## 5. Build your site and look for the JSON file

## 6. Allow CORS access to your help if stored on a remote server

```
Header set Access-Control-Allow-Origin "*"

```

```
curl -I http://idratherassets.com/wp-content/apidemos/tooltips.json
```

```
-I
```

```
Access-Control-Allow-Origin: *
```

### Add CORS Configuration

```
<CORSConfiguration>
 <CORSRule>
 <AllowedOrigin>*</AllowedOrigin>
 <AllowedMethod>GET</AllowedMethod>
 </CORSRule>
</CORSConfiguration>
```

(<http://enable-cors.org/server.html>)

✔ **Tip:** `$.get()` is a shorthand for `$.ajax({url: url, dataType: "json", success: success})`.

## 7. Explain how developers can access the help

Below is an example of how to use `$.get()` to load the help content for the `basketball` page.

```
<script type="text/javascript">
$(document).ready(function(){

var url = "mydoc_tooltips_source.json";

$.get(url, function(data) {

 $.each(data.entries, function(i, page) {
 if (page.id == "basketball") {
 $("#basketball").attr("data-content", page.body
);
 }
 });
});

});
</script>
```

The following table shows the parameters for the `$.get()` function.

<code>url</code>	The URL to the JSON file containing the help content.
<code>each</code>	A function to be called for each entry in the <code>data.entries</code> array.
<code>page.id</code>	The ID of the page to be loaded.
<code>basketball</code>	The ID of the page to be loaded.
<code>#basketball</code>	The ID of the page to be loaded.
<code>data-content</code>	The content of the page to be loaded.

⚠ **Warning:** The `$.get()` function is only supported in the `jQuery` library.



(<https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpajjjmafapmmgkkhgoa?hl=en>)

(<http://jsonformatter.curiousconcept.com/>)

data-content

(<http://getbootstrap.com/javascript/#popovers>)

data-content

```
<p>Basketball </p>
```

id="basketball"

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
```

```
<script type="text/javascript">
$(document).ready(function(){
 $('[data-toggle="popover"]').popover({
 placement : 'right',
 trigger: 'hover',
 html: true
 });
});
```

## 8. Create easy links to embed the help in your help site

```
<h2>Reuse Demo</h2>
```

```
<table>
<thead>
<tr>
<th>Sport</th>
<th>Comments</th>
</tr>
</thead>
<tbody>

<tr>
<td>Basketball</td>
<td>{{site.data.definitions.basketball}}</td>
</tr>

<tr>
<td>Baseball</td>
<td>{{site.data.definitions.baseball}}</td>
</tr>

<tr>
<td>Football</td>
<td>{{site.data.definitions.football}}</td>
</tr>

<tr>
<td>Soccer</td>
<td>{{site.data.definitions.soccer}}</td>
</tr>
</tbody>
</table>
```

# Reuse Demo

Sport	Comments
Baseball	Baseball is a team sport played on a field with a diamond-shaped infield and a pitcher's mound. The game involves batting and fielding, with the goal of scoring runs by hitting the ball into the field and running the bases.
Baseball	Baseball is a team sport played on a field with a diamond-shaped infield and a pitcher's mound. The game involves batting and fielding, with the goal of scoring runs by hitting the ball into the field and running the bases.
Baseball	Baseball is a team sport played on a field with a diamond-shaped infield and a pitcher's mound. The game involves batting and fielding, with the goal of scoring runs by hitting the ball into the field and running the bases.
Baseball	Baseball is a team sport played on a field with a diamond-shaped infield and a pitcher's mound. The game involves batting and fielding, with the goal of scoring runs by hitting the ball into the field and running the bases.

# Search configuration

## Summary:

## About search

## Excluding pages from search

`search: exclude`

## Troubleshooting search

(<http://jsonlint.com/>)

## Including the body field in search

body

```

 "body": "{{ page.content | strip_html | strip_newlines |
replace: '\\', '\\\\' | replace: '\"', '\\\"' | replace: '^t',
' ' }}",

```

```
| replace: '^t', ' '
```

([http://www.ultraedit.com/support/tutorials\\_power\\_tips/ultraedit/regular\\_expressions.html](http://www.ultraedit.com/support/tutorials_power_tips/ultraedit/regular_expressions.html))

## Customizing search results

```

title: search
layout: none
search: exclude

[
{% for page in site.pages %}
{% unless page.search == "exclude" %}
{
"title": "{{ page.title | escape }}",
"tags": "{{ page.tags }}",
"keywords": "{{page.keywords}}",
"url": "{{ page.url | prepend: site.baseurl }}",
"summary": "{{page.summary | strip }}"
},
{% endunless %}
{% endfor %}

{% for post in site.posts %}

{
"title": "{{ post.title | escape }}",
"tags": "{{ post.tags }}",
"keywords": "{{post.keywords}}",
"url": "{{ post.url | prepend: site.baseurl }}",
"summary": "{{post.summary | strip }}"
}
{% unless forloop.last %},{% endunless %}
{% endfor %}

]
```

```


 <!--start search-->
 <div id="search-demo-container">
 <input type="text" id="search-input" placeholder="search...">
 <ul id="results-container">
 </div>
 <script src="/mydoc-pdfjs/jekyll-search.js" type="text/javascript"></script>
 <script type="text/javascript">
 SimpleJekyllSearch.init({
 searchInput: document.getElementById('search-input'),
 resultsContainer: document.getElementById('results-container'),
 dataSource: '/mydoc-pdf/search.json',
 searchResultTemplate: '{title}',
 noResultsText: 'No results found.',
 limit: 10,
 fuzzy: true,
 })
 </script>
 <!--end search-->


```

{url}      {title}

## More robust search

(<http://swifttype.com>)      (<http://algolia.com>)



# iTerm profiles

## Summary:

## About iTerm profiles

## Set up profiles

**Profiles > Open Profiles.**

### Edit Profiles

#### Name

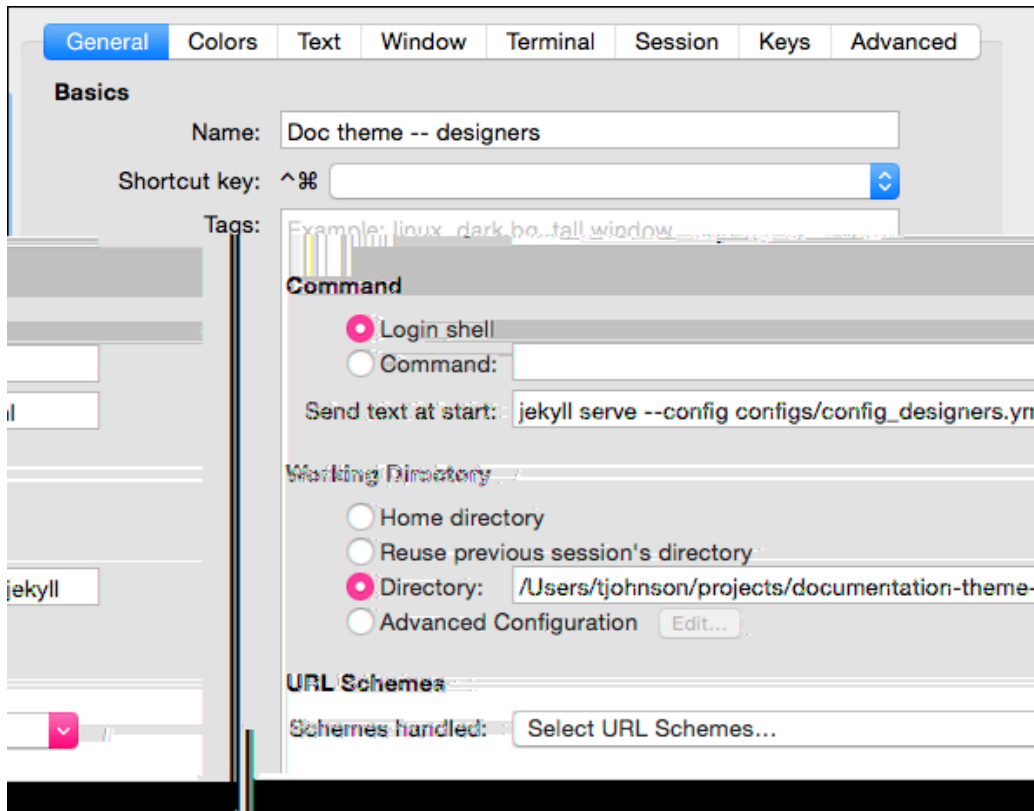
Doc theme -- designers

#### Send text at start

```
JEKYLL_ENV=production jekyll serve
```

#### Directory

**theme-jekyll** **/Users/tjohnson/projects/documentation-**



## Launching a profile

**View > Toggle Toolbar**

**New**

☒ **Tip:**

**Ctrl+C**

## Pushing builds to server

### Summary:

## Pushing to AWS S3

```
aws s3 cp ~/users/tjohnson/projects/mydocproject/ s3://[aws path]docpath/mydocproject --recursive
```

```
aws s3 cp ~/users/tjohnson/projects/anotherdocproject2/ s3://[aws path]docpath/anotherdocproject --recursive
```

## Pushing to a regular server

scp

```
scp -r /users/tjohnson/projects/mydocproject/ name@domain:/var/www/html/mydocproject
```

# Getting around the password prompts in SCP

## Summary:

SCP is a tool for copying files between a local host and a remote host. It is a wrapper around the SSH protocol. When you use SCP to copy a file, you are prompted for the password of the user on the remote host. This can be annoying if you have to enter the password every time you use SCP. There are a few ways to get around this. One way is to use the `scp -P` option. Another way is to use the `scp -r` option. A third way is to use the `scp -o` option. This document will show you how to use these options to get around the password prompts in SCP.

## Get rid of password prompts

There are a few ways to get around the password prompts in SCP. One way is to use the `scp -P` option. Another way is to use the `scp -r` option. A third way is to use the `scp -o` option. This document will show you how to use these options to get around the password prompts in SCP.

```
cd ~/.ssh
```

There are a few ways to get around the password prompts in SCP. One way is to use the `scp -P` option. Another way is to use the `scp -r` option. A third way is to use the `scp -o` option. This document will show you how to use these options to get around the password prompts in SCP.

(<http://ianlunn.co.uk/articles/quickly-showhide-hidden-files-mac-os-x-mavericks/>)

There are a few ways to get around the password prompts in SCP. One way is to use the `scp -P` option. Another way is to use the `scp -r` option. A third way is to use the `scp -o` option. This document will show you how to use these options to get around the password prompts in SCP.

```
mkdir .ssh
```

There are a few ways to get around the password prompts in SCP. One way is to use the `scp -P` option. Another way is to use the `scp -r` option. A third way is to use the `scp -o` option. This document will show you how to use these options to get around the password prompts in SCP.

```
ssh-keygen -t rsa
```

**Enter**

Enter

**Enter**

```
tjohnson-mbpr13:.ssh tjohnson$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/yourname/.ssh/id_rsa.
Your public key has been saved in /Users/yourname/.ssh/id_rsa.pub.
The key fingerprint is:
9a:8f:b5:495:39:78:t5:dc:19:d6:29:66:02:e8:02:a0 yourname@yourname-mbpr99.local
```

```
+--[RSA 2048]-----+
| |
|+ |
|E |
|o. . |
|.. = o S |
|.&^ + 7i = o |
| = B . |
| o 0 + |
| *.o |
+-----+
```

```
ssh <your_username>@remoteserver.com
```

```
<your_username>
```

```
/home/remoteserver/<your_username>
pwd dir
```

```
/home/remoteserver/<your_username>
```

```
mkdir -p .ssh
```

```
ls -a
```

```
cd ~/.ssh
```

```
scp id_rsa.pub <your-username>@yourserver.com:/home/remo
teserver/<your-username>/.ssh
```

```
authorized_keys
```

```
mv id_rsa.pub authorized_keys
```

```
chmod 700 authorized_keys
```

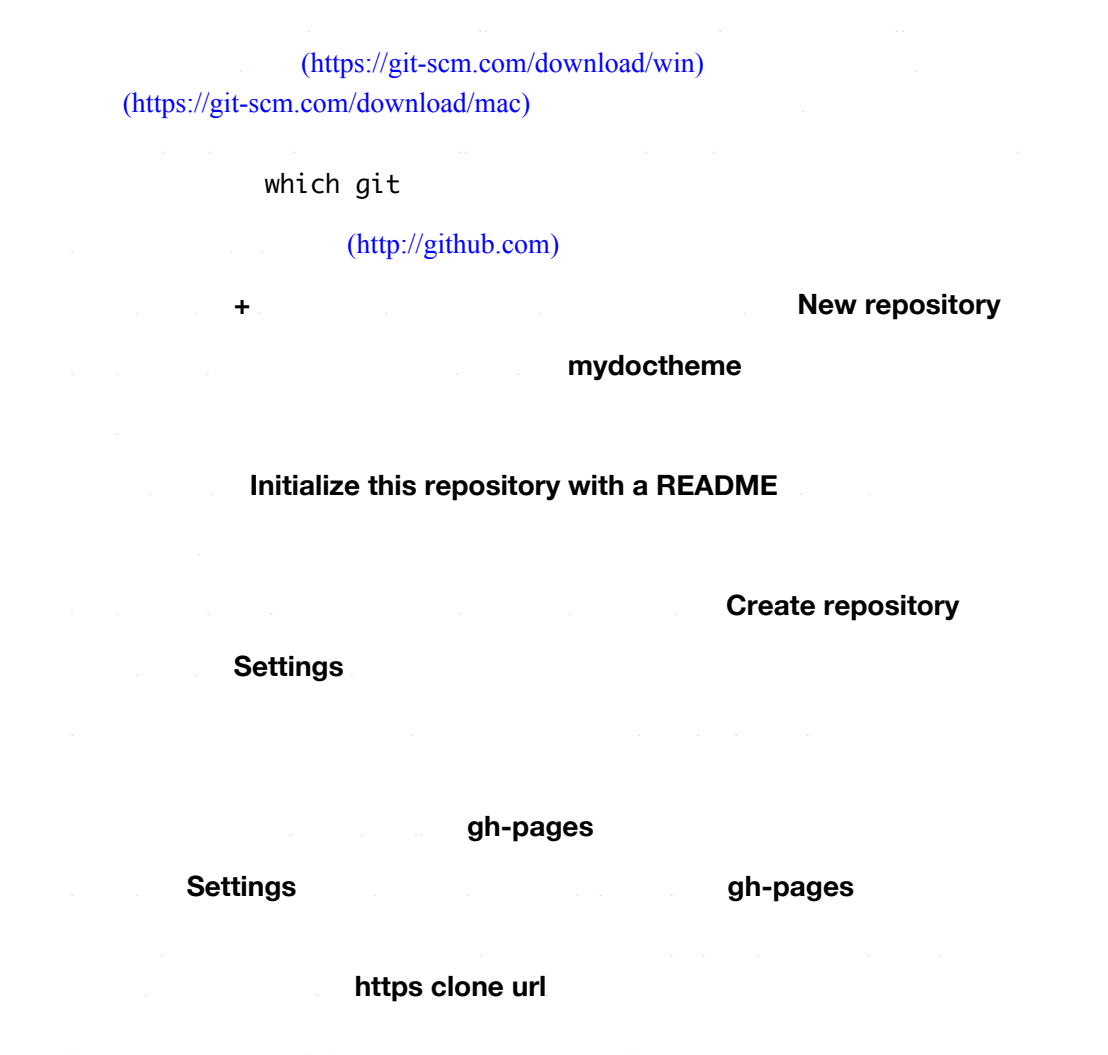
```
ssh <your_username>@remoteserver.com
```

```
scp -r ../doc_outputs/mydoc/writers <your-username>@remo
teserver:/var/www/html/
```

# Publishing on Github Pages

## Summary:

## Set up your Github repo





```
git clone https://github.com/tomjohnson1492/myreponame.git
cd myreponame
https://github.com/tomjohnson1492/myreponame.git
```

```
url: tomjohnson1492.github.io baseurl: /myreponame
```

## Install Bundler

Install Bundler using Homebrew (Mac OS X) or RubyGems (Linux).

For Mac OS X, see <http://brew.sh/>

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```
gem install bundler
```

## Add the github pages gem

```
bundle init
```

```
open gemfile
```

```
source 'https://rubygems.org'
gem 'github-pages'
```

```
bundle install
```

```
git add --all
git commit -m "committing my jekyll theme"
git push
```

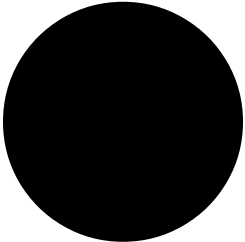
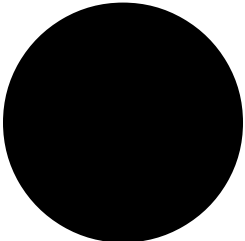
## Customize your URL

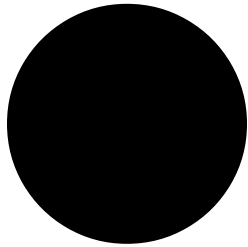
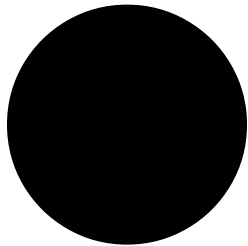
## Knowledge-base layout

### Summary:

Knowledge base layout is a set of rules that define the structure of the knowledge base. It is used to organize the knowledge base into categories and subcategories. The layout is defined by a set of rules that are applied to the knowledge base. The rules are defined in the Knowledge Base Categories section.

## Knowledge Base Categories


<p><i>Getting started</i></p> <p><input type="text"/></p>


<div data-bbox="734 210 888 245"><i>Navigation</i></div> <div data-bbox="657 319 961 388"></div>
<div data-bbox="686 499 932 743"></div> <div data-bbox="704 827 915 867"><i>Single sourcing</i></div> <div data-bbox="657 940 961 1010"></div>
<div data-bbox="686 1119 932 1362"></div> <div data-bbox="734 1446 888 1486"><i>Formatting</i></div> <div data-bbox="657 1560 961 1629"></div>

## Generating a list of all pages with a certain tag

Getting started pages:

```

{% assign sorted_pages = (site.pages | sort: 'title') %}
{% for page in sorted_pages %}
{% for tag in page.tags %}
{% if tag == "getting_started" %}
{{ page.title }}
{% endif %}
{% endfor %}
{% endfor %}

```

# Glossary layout

## Summary:

Summary of the glossary layout. The glossary is a list of terms and their definitions. It is organized alphabetically. The terms are listed in the left column, and the definitions are listed in the right column. The definitions are written in a clear, concise, and easy-to-understand manner. The glossary is a valuable resource for users who need to understand the terminology used in the documentation.

## fractious

fractious: adjective. Easily irritated or annoyed; fractious.

## gratuitous

gratuitous: adjective. Not free of charge; gratuitous. Also, adjective. Not necessary or useful; gratuitous.

## haughty

haughty: adjective. Proud or arrogant; haughty. Also, adjective. Disdainful or contemptuous; haughty.

## gratuitous

gratuitous: adjective. Not free of charge; gratuitous. Also, adjective. Not necessary or useful; gratuitous.

## impertinent

impertinent: adjective. Not showing respect or consideration for others; impertinent. Also, adjective. Not relevant or pertinent; impertinent.

```
fractious
: {{site.data.glossary.fractious}}

gratuitous
: {{site.data.glossary.gratuitous}}

haughty
: {{site.data.glossary.haughty}}

gratuitous
: {{site.data.glossary.gratuitous}}

impertinent
: {{site.data.glossary.intrepid}}
```

## Horizontally styled definiton lists

```
<div class="glossary">
 <dl class="dl-horizontal">
 <dt>fractious</dt>
```

**fractious**

```
<dd>{{site.data.glossary.fractious}}</dd>
 </dl>
 <dt>gratuitous</dt>
```

**gratuitous**

```
<dd>{{site.data.glossary.gratuitous}}</dd>
 </dl>
 <dt>haughty</dt>
```

**haughty**

```
<dd>{{site.data.glossary.haughty}}</dd>
 </dl>
 <dt>gratuitous</dt>
```

**gratuitous**

```
<dd>{{site.data.glossary.gratuitous}}</dd>
 </dl>
</div>
```

**impertinent****intrepid**

```
<dl class="dl-horizontal">

<dt id="fractious">fractious</dt>
<dd>{{site.data.glossary.fractious}}</dd>

<dt id="gratuitous">gratuitous</dt>
<dd>{{site.data.glossary.gratuitous}}</dd>

<dt id="haughty">haughty</dt>
<dd>{{site.data.glossary.haughty}}</dd>

<dt id="benchmark_id">gratuitous</dt>
<dd>{{site.data.glossary.gratuitous}}</dd>

<dt id="impertinent">impertinent</dt>
<dd>{{site.data.glossary.impertinent}}</dd>

<dt id="intrepid">intrepid</dt>
<dd>{{site.data.glossary.intrepid}}</dd>

</dl>
```

dl



## FAQ layout

### Summary:

div





# Troubleshooting

### Summary:

## Issues building the site

```

jekyll 2.5.3 | Error: Address already in use - bind(2)

```

```
ps aux | grep jekyll
```

```
kill -9 22298
```

```
kill -9 $(ps aux | grep '[j]ekyll' | awk '{print $2}')
```

```
chmod +x build_writer.sh
```

## shell file not runnable

search: exclude