



# The best test project — pdf test

version 1.0

*Last generated: August 18, 2020*

---



Company  
logo

© 2020 Your company. This is a boilerplate copyright statement... All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

# Table of Contents

## Topic 1

Page 1 .....	2
Page 2 .....	10
Page 3 .....	18

## Topic 2

Page 1 .....	26
Page 2 .....	34
Page 3 .....	42

# Page 1

**Summary:** This theme primarily uses pages. You need to make sure your pages have the appropriate frontmatter. One frontmatter tag your users might find helpful is the summary tag. This functions similar in purpose to the shortdesc element in DITA.

## Where to author content

Use a text editor such as Sublime Text, WebStorm, IntelliJ, or Atom to create pages. Atom is recommended because it's created by Github, which is driving some of the Jekyll development through Github Pages.

## Where to save pages

You can store your pages in any folder structures you want, with any level of folder nesting. The site output will pull all of those pages out of their folders and put them into the root directory. Check out the `_site` folder, which is where Jekyll is generated, to see the difference between your project's structure and the resulting site output.

The listing of all pages in the root directory (which happens when you add a permalink property to the pages) is what allows the relative linking and offline viewing of the site to work.

## Frontmatter

Make sure each page has frontmatter at the top like this:

```

---
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
---

```

Frontmatter is always formatted with three hyphens at the top and bottom. Your frontmatter must have a `title` and `permalink` value. All the other values are optional.

Note that you cannot use variables in frontmatter.

The following table describes each of the frontmatter that you can use with this theme:

Frontmatter	Required?	Description
<b>title</b>	Required	The title for the page
<b>tags</b>	Optional	Tags for the page. Make all tags single words, with underscores if needed (rather than spaces). Separate them with commas. Enclose the whole list within brackets. Also, note that tags must be added to <code>_data/tags_doc.yml</code> to be allowed entrance into the page. This prevents tags from becoming somewhat random and unstructured. You must create a tag page for each one of your tags following the pattern shown in the tags folder. (Tag pages aren't automatically created.)
<b>keywords</b>	Optional	Synonyms and other keywords for the page. This information gets stuffed into the page's metadata to increase SEO. The user won't see the keywords, but if you search for one of the keywords, it will be picked up by the search engine.

Frontmatter	Required?	Description
<b>last_updated</b>	Optional	The date the page was last updated. This information could be helpful for readers trying to evaluate how current and authoritative information is. If included, the <code>last_updated</code> date appears in the footer of the page in small font.
<b>sidebar</b>	Required	Refers to the sidebar data file for this page. Don't include the ".yml" file extension for the sidebar — just provide the file name. If no sidebar is specified, this value will inherit the <code>default</code> property set in your <code>_config.yml</code> file for the page's frontmatter.
<b>summary</b>	Optional	A 1-2 word sentence summarizing the content on the page. This gets formatted into the summary section in the page layout. Adding summaries is a key way to make your content more scannable by users (check out <a href="#">Jakob Nielsen's site</a> for a great example of page summaries.) The only drawback with summaries is that you can't use variables in them.
<b>permalink</b>	Required	The permalink <i>must</i> match the filename in order for automated links to work. Additionally, you must include the ".html" in the filename. Do not put forward slashes around the permalink (this makes Jekyll put the file inside a folder in the output). When Jekyll builds the site, it will put the page into the root directory rather than leaving it in a subdirectory or putting it inside a folder and naming the file <code>index.html</code> . Having all files flattened in the root directory is essential for relative linking to work and for all paths to JS and CSS files to be valid.

Frontmatter	Required?	Description
<b>datatable</b>	Optional	'true'. If you add <code>datatable: true</code> in the frontmatter, scripts for the <a href="#">jQuery Datatables plugin</a> get included on the page. You can see the scripts that conditionally appear by looking in the <code>_layouts/default.html</code> page.
<b>toc</b>	Optional	If you specify <code>toc: false</code> in the frontmatter, the page won't have the table of contents that appears below the title. The toc refers to the list of jump links below the page title, not the sidebar navigation. You probably want to hide the TOC on the homepage and product landing pages.

## Colons in page titles

If you want to use a colon in your page title, you must enclose the title's value in quotation marks.

## Page names and excluding files from outputs

By default, everything in your project is included in the output. You can exclude all files that don't belong to that project by specifying the file name, the folder name, or by using wildcards in your configuration file:

`exclude:`

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

Wildcards will exclude every match after the `*`.

## Saving pages as drafts

If you add `published: false` in the frontmatter, your page won't be published. You can also move draft pages into the `_drafts` folder to exclude them from the build. With posts, you can also keep them as drafts by omitting the date in the title.

## Markdown or HTML format

Pages can be either Markdown or HTML format (specified through either an `.md` or `.html` file extension).

If you use Markdown, you can also include HTML formatting where needed. But if your format is HTML, you must add a `markdown="1"` attribute to the element in order to use Markdown inside that HTML element:

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

For your Markdown files, note that a space or two indent will set text off as code or blocks, so avoid spacing indents unless intentional.

If you have a lot of HTML, as long as the top and bottom tags of the HTML are flush left in a Markdown file, all the tags inside those bookend HTML tags will render as HTML, regardless of their indentation. (This can be especially useful for tables.)

## Page names

I recommend prefixing your page names with the product, such as `"mydoc_pages"` instead of just `"pages."` This way if you have other products that also have topics with generic names such as `"pages,"` there won't be naming conflicts.

Additionally, consider adding the product name in parentheses after the title, such as `"Pages (Mydoc)"` so that users can clearly navigate different topics for each product.



## Kramdown Markdown

Kramdown is the Markdown flavor used in the theme. This mostly aligns with Github-flavored Markdown, but with some differences in the indentation allowed within lists. Basically, Kramdown requires you to line up the indent between list items with the first starting character after the space in your list item numbering. See this [blog post on Kramdown and Rouge](#) for more details.

You can use standard Multimarkdown syntax for tables. You can also use fenced code blocks with lexers specifying the type of code. The configuration file shows the Markdown processor and extension:

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

## Automatic mini-TOCs

By default, a TOC appears at the top of your pages and posts. If you don't want the TOC to appear for a specific page, such as for a landing page or other homepage, add `toc: false` in the frontmatter of the page.

The mini-TOC requires you to use the `##` Markdown syntax for headings. If you use `<h2>` elements, you must add an ID attribute for the heading element in order for it to appear in the mini-TOC (for example, `<h2 id="mysampleid">Heading</h2>`).

## Headings

Use pound signs before the heading title to designate the level. Note that kramdown requires headings to have one space before and after the heading. Without this space above and below, the heading won't render into HTML.

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

**Result:**

## Second-level heading

---

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

**Result:**

### Third-level heading

---

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

**Result:**

#### *Fourth-level heading*

## Headings with ID Tags

If you want to use a specific ID tag with your heading, add it like this:

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

Then you can reference it with a link like this on the same page:

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

**Result:**

[Some link \(page 8\)](#)

For details about linking to headings on different pages, see [Automated links to headings on pages \(page 0\)](#).

## Specify a particular page layout

The configuration file sets the default layout for pages as the “page” layout.

You can create other layouts inside the layouts folder. If you create a new layout, you can specify that your page use your new layout by adding `layout: mylayout.html` in the page’s frontmatter. Whatever layout you specify in the frontmatter of a page will override the layout default set in the configuration file.

## Comments

Disqus, a commenting system, is integrated into the theme. In the configuration file, specify the Disqus code for the universal code, and Disqus will appear. If you don’t add a Disqus value, the Disqus form isn’t included.

## Page 2

**Summary:** This theme primarily uses pages. You need to make sure your pages have the appropriate frontmatter. One frontmatter tag your users might find helpful is the summary tag. This functions similar in purpose to the shortdesc element in DITA.

### Where to author content

Use a text editor such as Sublime Text, WebStorm, IntelliJ, or Atom to create pages. Atom is recommended because it's created by Github, which is driving some of the Jekyll development through Github Pages.

### Where to save pages

You can store your pages in any folder structures you want, with any level of folder nesting. The site output will pull all of those pages out of their folders and put them into the root directory. Check out the `_site` folder, which is where Jekyll is generated, to see the difference between your project's structure and the resulting site output.

The listing of all pages in the root directory (which happens when you add a permalink property to the pages) is what allows the relative linking and offline viewing of the site to work.

### Frontmatter

Make sure each page has frontmatter at the top like this:

```

---
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
---

```

Frontmatter is always formatted with three hyphens at the top and bottom. Your frontmatter must have a `title` and `permalink` value. All the other values are optional.

Note that you cannot use variables in frontmatter.

The following table describes each of the frontmatter that you can use with this theme:

Frontmatter	Required?	Description
<b>title</b>	Required	The title for the page
<b>tags</b>	Optional	Tags for the page. Make all tags single words, with underscores if needed (rather than spaces). Separate them with commas. Enclose the whole list within brackets. Also, note that tags must be added to <code>_data/tags_doc.yml</code> to be allowed entrance into the page. This prevents tags from becoming somewhat random and unstructured. You must create a tag page for each one of your tags following the pattern shown in the tags folder. (Tag pages aren't automatically created.)
<b>keywords</b>	Optional	Synonyms and other keywords for the page. This information gets stuffed into the page's metadata to increase SEO. The user won't see the keywords, but if you search for one of the keywords, it will be picked up by the search engine.

Frontmatter	Required?	Description
<b>last_updated</b>	Optional	The date the page was last updated. This information could be helpful for readers trying to evaluate how current and authoritative information is. If included, the <code>last_updated</code> date appears in the footer of the page in small font.
<b>sidebar</b>	Required	Refers to the sidebar data file for this page. Don't include the ".yml" file extension for the sidebar — just provide the file name. If no sidebar is specified, this value will inherit the <code>default</code> property set in your <code>_config.yml</code> file for the page's frontmatter.
<b>summary</b>	Optional	A 1-2 word sentence summarizing the content on the page. This gets formatted into the summary section in the page layout. Adding summaries is a key way to make your content more scannable by users (check out <a href="#">Jakob Nielsen's site</a> for a great example of page summaries.) The only drawback with summaries is that you can't use variables in them.
<b>permalink</b>	Required	The permalink <i>must</i> match the filename in order for automated links to work. Additionally, you must include the ".html" in the filename. Do not put forward slashes around the permalink (this makes Jekyll put the file inside a folder in the output). When Jekyll builds the site, it will put the page into the root directory rather than leaving it in a subdirectory or putting it inside a folder and naming the file <code>index.html</code> . Having all files flattened in the root directory is essential for relative linking to work and for all paths to JS and CSS files to be valid.

Frontmatter	Required?	Description
<b>datatable</b>	Optional	'true'. If you add <code>datatable: true</code> in the frontmatter, scripts for the <a href="#">jQuery Datatables plugin</a> get included on the page. You can see the scripts that conditionally appear by looking in the <code>_layouts/default.html</code> page.
<b>toc</b>	Optional	If you specify <code>toc: false</code> in the frontmatter, the page won't have the table of contents that appears below the title. The toc refers to the list of jump links below the page title, not the sidebar navigation. You probably want to hide the TOC on the homepage and product landing pages.

## Colons in page titles

If you want to use a colon in your page title, you must enclose the title's value in quotation marks.

## Page names and excluding files from outputs

By default, everything in your project is included in the output. You can exclude all files that don't belong to that project by specifying the file name, the folder name, or by using wildcards in your configuration file:

`exclude:`

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

Wildcards will exclude every match after the `*`.

## Saving pages as drafts

If you add `published: false` in the frontmatter, your page won't be published. You can also move draft pages into the `_drafts` folder to exclude them from the build. With posts, you can also keep them as drafts by omitting the date in the title.

## Markdown or HTML format

Pages can be either Markdown or HTML format (specified through either an `.md` or `.html` file extension).

If you use Markdown, you can also include HTML formatting where needed. But if your format is HTML, you must add a `markdown="1"` attribute to the element in order to use Markdown inside that HTML element:

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

For your Markdown files, note that a space or two indent will set text off as code or blocks, so avoid spacing indents unless intentional.

If you have a lot of HTML, as long as the top and bottom tags of the HTML are flush left in a Markdown file, all the tags inside those bookend HTML tags will render as HTML, regardless of their indentation. (This can be especially useful for tables.)

## Page names

I recommend prefixing your page names with the product, such as `"mydoc_pages"` instead of just `"pages."` This way if you have other products that also have topics with generic names such as `"pages,"` there won't be naming conflicts.

Additionally, consider adding the product name in parentheses after the title, such as `"Pages (Mydoc)"` so that users can clearly navigate different topics for each product.



## Kramdown Markdown

Kramdown is the Markdown flavor used in the theme. This mostly aligns with Github-flavored Markdown, but with some differences in the indentation allowed within lists. Basically, Kramdown requires you to line up the indent between list items with the first starting character after the space in your list item numbering. See this [blog post on Kramdown and Rouge](#) for more details.

You can use standard Multimarkdown syntax for tables. You can also use fenced code blocks with lexers specifying the type of code. The configuration file shows the Markdown processor and extension:

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

## Automatic mini-TOCs

By default, a TOC appears at the top of your pages and posts. If you don't want the TOC to appear for a specific page, such as for a landing page or other homepage, add `toc: false` in the frontmatter of the page.

The mini-TOC requires you to use the `##` Markdown syntax for headings. If you use `<h2>` elements, you must add an ID attribute for the heading element in order for it to appear in the mini-TOC (for example, `<h2 id="mysampleid">Heading</h2>`).

## Headings

Use pound signs before the heading title to designate the level. Note that kramdown requires headings to have one space before and after the heading. Without this space above and below, the heading won't render into HTML.

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

**Result:**

## Second-level heading

---

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

**Result:**

### Third-level heading

---

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

**Result:**

#### *Fourth-level heading*

## Headings with ID Tags

If you want to use a specific ID tag with your heading, add it like this:

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

Then you can reference it with a link like this on the same page:

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

**Result:**

[Some link \(page 16\)](#)

For details about linking to headings on different pages, see [Automated links to headings on pages \(page 0\)](#).

## Specify a particular page layout

The configuration file sets the default layout for pages as the “page” layout.

You can create other layouts inside the layouts folder. If you create a new layout, you can specify that your page use your new layout by adding `layout: mylayout.html` in the page’s frontmatter. Whatever layout you specify in the frontmatter of a page will override the layout default set in the configuration file.

## Comments

Disqus, a commenting system, is integrated into the theme. In the configuration file, specify the Disqus code for the universal code, and Disqus will appear. If you don’t add a Disqus value, the Disqus form isn’t included.

## Page 3

**Summary:** This theme primarily uses pages. You need to make sure your pages have the appropriate frontmatter. One frontmatter tag your users might find helpful is the summary tag. This functions similar in purpose to the shortdesc element in DITA.

### Where to author content

Use a text editor such as Sublime Text, WebStorm, IntelliJ, or Atom to create pages. Atom is recommended because it's created by Github, which is driving some of the Jekyll development through Github Pages.

### Where to save pages

You can store your pages in any folder structures you want, with any level of folder nesting. The site output will pull all of those pages out of their folders and put them into the root directory. Check out the `_site` folder, which is where Jekyll is generated, to see the difference between your project's structure and the resulting site output.

The listing of all pages in the root directory (which happens when you add a permalink property to the pages) is what allows the relative linking and offline viewing of the site to work.

### Frontmatter

Make sure each page has frontmatter at the top like this:

```

---
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
---

```

Frontmatter is always formatted with three hyphens at the top and bottom. Your frontmatter must have a `title` and `permalink` value. All the other values are optional.

Note that you cannot use variables in frontmatter.

The following table describes each of the frontmatter that you can use with this theme:

Frontmatter	Required?	Description
<b>title</b>	Required	The title for the page
<b>tags</b>	Optional	Tags for the page. Make all tags single words, with underscores if needed (rather than spaces). Separate them with commas. Enclose the whole list within brackets. Also, note that tags must be added to <code>_data/tags_doc.yml</code> to be allowed entrance into the page. This prevents tags from becoming somewhat random and unstructured. You must create a tag page for each one of your tags following the pattern shown in the tags folder. (Tag pages aren't automatically created.)
<b>keywords</b>	Optional	Synonyms and other keywords for the page. This information gets stuffed into the page's metadata to increase SEO. The user won't see the keywords, but if you search for one of the keywords, it will be picked up by the search engine.

Frontmatter	Required?	Description
<b>last_updated</b>	Optional	The date the page was last updated. This information could be helpful for readers trying to evaluate how current and authoritative information is. If included, the <code>last_updated</code> date appears in the footer of the page in small font.
<b>sidebar</b>	Required	Refers to the sidebar data file for this page. Don't include the ".yml" file extension for the sidebar — just provide the file name. If no sidebar is specified, this value will inherit the <code>default</code> property set in your <code>_config.yml</code> file for the page's frontmatter.
<b>summary</b>	Optional	A 1-2 word sentence summarizing the content on the page. This gets formatted into the summary section in the page layout. Adding summaries is a key way to make your content more scannable by users (check out <a href="#">Jakob Nielsen's site</a> for a great example of page summaries.) The only drawback with summaries is that you can't use variables in them.
<b>permalink</b>	Required	The permalink <i>must</i> match the filename in order for automated links to work. Additionally, you must include the ".html" in the filename. Do not put forward slashes around the permalink (this makes Jekyll put the file inside a folder in the output). When Jekyll builds the site, it will put the page into the root directory rather than leaving it in a subdirectory or putting it inside a folder and naming the file <code>index.html</code> . Having all files flattened in the root directory is essential for relative linking to work and for all paths to JS and CSS files to be valid.

Frontmatter	Required?	Description
<b>datatable</b>	Optional	'true'. If you add <code>datatable: true</code> in the frontmatter, scripts for the <a href="#">jQuery Datatables plugin</a> get included on the page. You can see the scripts that conditionally appear by looking in the <code>_layouts/default.html</code> page.
<b>toc</b>	Optional	If you specify <code>toc: false</code> in the frontmatter, the page won't have the table of contents that appears below the title. The toc refers to the list of jump links below the page title, not the sidebar navigation. You probably want to hide the TOC on the homepage and product landing pages.

## Colons in page titles

If you want to use a colon in your page title, you must enclose the title's value in quotation marks.

## Page names and excluding files from outputs

By default, everything in your project is included in the output. You can exclude all files that don't belong to that project by specifying the file name, the folder name, or by using wildcards in your configuration file:

`exclude:`

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

Wildcards will exclude every match after the `*`.

## Saving pages as drafts

If you add `published: false` in the frontmatter, your page won't be published. You can also move draft pages into the `_drafts` folder to exclude them from the build. With posts, you can also keep them as drafts by omitting the date in the title.

## Markdown or HTML format

Pages can be either Markdown or HTML format (specified through either an `.md` or `.html` file extension).

If you use Markdown, you can also include HTML formatting where needed. But if your format is HTML, you must add a `markdown="1"` attribute to the element in order to use Markdown inside that HTML element:

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

For your Markdown files, note that a space or two indent will set text off as code or blocks, so avoid spacing indents unless intentional.

If you have a lot of HTML, as long as the top and bottom tags of the HTML are flush left in a Markdown file, all the tags inside those bookend HTML tags will render as HTML, regardless of their indentation. (This can be especially useful for tables.)

## Page names

I recommend prefixing your page names with the product, such as `"mydoc_pages"` instead of just `"pages."` This way if you have other products that also have topics with generic names such as `"pages,"` there won't be naming conflicts.

Additionally, consider adding the product name in parentheses after the title, such as `"Pages (Mydoc)"` so that users can clearly navigate different topics for each product.



## Kramdown Markdown

Kramdown is the Markdown flavor used in the theme. This mostly aligns with Github-flavored Markdown, but with some differences in the indentation allowed within lists. Basically, Kramdown requires you to line up the indent between list items with the first starting character after the space in your list item numbering. See this [blog post on Kramdown and Rouge](#) for more details.

You can use standard Multimarkdown syntax for tables. You can also use fenced code blocks with lexers specifying the type of code. The configuration file shows the Markdown processor and extension:

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

## Automatic mini-TOCs

By default, a TOC appears at the top of your pages and posts. If you don't want the TOC to appear for a specific page, such as for a landing page or other homepage, add `toc: false` in the frontmatter of the page.

The mini-TOC requires you to use the `##` Markdown syntax for headings. If you use `<h2>` elements, you must add an ID attribute for the heading element in order for it to appear in the mini-TOC (for example, `<h2 id="mysampleid">Heading</h2>`).

## Headings

Use pound signs before the heading title to designate the level. Note that kramdown requires headings to have one space before and after the heading. Without this space above and below, the heading won't render into HTML.

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

**Result:**

## Second-level heading

---

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

**Result:**

### Third-level heading

---

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

**Result:**

#### *Fourth-level heading*

## Headings with ID Tags

If you want to use a specific ID tag with your heading, add it like this:

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

Then you can reference it with a link like this on the same page:

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

**Result:**

[Some link \(page 24\)](#)

For details about linking to headings on different pages, see [Automated links to headings on pages \(page 0\)](#).

## Specify a particular page layout

The configuration file sets the default layout for pages as the “page” layout.

You can create other layouts inside the layouts folder. If you create a new layout, you can specify that your page use your new layout by adding `layout: mylayout.html` in the page’s frontmatter. Whatever layout you specify in the frontmatter of a page will override the layout default set in the configuration file.

## Comments

Disqus, a commenting system, is integrated into the theme. In the configuration file, specify the Disqus code for the universal code, and Disqus will appear. If you don’t add a Disqus value, the Disqus form isn’t included.

# Page 1

**Summary:** This theme primarily uses pages. You need to make sure your pages have the appropriate frontmatter. One frontmatter tag your users might find helpful is the summary tag. This functions similar in purpose to the shortdesc element in DITA.

## Where to author content

Use a text editor such as Sublime Text, WebStorm, IntelliJ, or Atom to create pages. Atom is recommended because it's created by Github, which is driving some of the Jekyll development through Github Pages.

## Where to save pages

You can store your pages in any folder structures you want, with any level of folder nesting. The site output will pull all of those pages out of their folders and put them into the root directory. Check out the `_site` folder, which is where Jekyll is generated, to see the difference between your project's structure and the resulting site output.

The listing of all pages in the root directory (which happens when you add a permalink property to the pages) is what allows the relative linking and offline viewing of the site to work.

## Frontmatter

Make sure each page has frontmatter at the top like this:

```

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

```

Frontmatter is always formatted with three hyphens at the top and bottom. Your frontmatter must have a `title` and `permalink` value. All the other values are optional.

Note that you cannot use variables in frontmatter.

The following table describes each of the frontmatter that you can use with this theme:

Frontmatter	Required?	Description
<b>title</b>	Required	The title for the page
<b>tags</b>	Optional	Tags for the page. Make all tags single words, with underscores if needed (rather than spaces). Separate them with commas. Enclose the whole list within brackets. Also, note that tags must be added to <code>_data/tags_doc.yml</code> to be allowed entrance into the page. This prevents tags from becoming somewhat random and unstructured. You must create a tag page for each one of your tags following the pattern shown in the tags folder. (Tag pages aren't automatically created.)
<b>keywords</b>	Optional	Synonyms and other keywords for the page. This information gets stuffed into the page's metadata to increase SEO. The user won't see the keywords, but if you search for one of the keywords, it will be picked up by the search engine.

Frontmatter	Required?	Description
<b>last_updated</b>	Optional	The date the page was last updated. This information could be helpful for readers trying to evaluate how current and authoritative information is. If included, the <code>last_updated</code> date appears in the footer of the page in small font.
<b>sidebar</b>	Required	Refers to the sidebar data file for this page. Don't include the ".yml" file extension for the sidebar — just provide the file name. If no sidebar is specified, this value will inherit the <code>default</code> property set in your <code>_config.yml</code> file for the page's frontmatter.
<b>summary</b>	Optional	A 1-2 word sentence summarizing the content on the page. This gets formatted into the summary section in the page layout. Adding summaries is a key way to make your content more scannable by users (check out <a href="#">Jakob Nielsen's site</a> for a great example of page summaries.) The only drawback with summaries is that you can't use variables in them.
<b>permalink</b>	Required	The permalink <i>must</i> match the filename in order for automated links to work. Additionally, you must include the ".html" in the filename. Do not put forward slashes around the permalink (this makes Jekyll put the file inside a folder in the output). When Jekyll builds the site, it will put the page into the root directory rather than leaving it in a subdirectory or putting it inside a folder and naming the file <code>index.html</code> . Having all files flattened in the root directory is essential for relative linking to work and for all paths to JS and CSS files to be valid.

Frontmatter	Required?	Description
<b>datatable</b>	Optional	'true'. If you add <code>datatable: true</code> in the frontmatter, scripts for the <a href="#">jQuery Datatables plugin</a> get included on the page. You can see the scripts that conditionally appear by looking in the <code>_layouts/default.html</code> page.
<b>toc</b>	Optional	If you specify <code>toc: false</code> in the frontmatter, the page won't have the table of contents that appears below the title. The toc refers to the list of jump links below the page title, not the sidebar navigation. You probably want to hide the TOC on the homepage and product landing pages.

## Colons in page titles

If you want to use a colon in your page title, you must enclose the title's value in quotation marks.

## Page names and excluding files from outputs

By default, everything in your project is included in the output. You can exclude all files that don't belong to that project by specifying the file name, the folder name, or by using wildcards in your configuration file:

`exclude:`

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

Wildcards will exclude every match after the `*`.

## Saving pages as drafts

If you add `published: false` in the frontmatter, your page won't be published. You can also move draft pages into the `_drafts` folder to exclude them from the build. With posts, you can also keep them as drafts by omitting the date in the title.

## Markdown or HTML format

Pages can be either Markdown or HTML format (specified through either an `.md` or `.html` file extension).

If you use Markdown, you can also include HTML formatting where needed. But if your format is HTML, you must add a `markdown="1"` attribute to the element in order to use Markdown inside that HTML element:

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

For your Markdown files, note that a space or two indent will set text off as code or blocks, so avoid spacing indents unless intentional.

If you have a lot of HTML, as long as the top and bottom tags of the HTML are flush left in a Markdown file, all the tags inside those bookend HTML tags will render as HTML, regardless of their indentation. (This can be especially useful for tables.)

## Page names

I recommend prefixing your page names with the product, such as `"mydoc_pages"` instead of just `"pages."` This way if you have other products that also have topics with generic names such as `"pages,"` there won't be naming conflicts.

Additionally, consider adding the product name in parentheses after the title, such as `"Pages (Mydoc)"` so that users can clearly navigate different topics for each product.



## Kramdown Markdown

Kramdown is the Markdown flavor used in the theme. This mostly aligns with Github-flavored Markdown, but with some differences in the indentation allowed within lists. Basically, Kramdown requires you to line up the indent between list items with the first starting character after the space in your list item numbering. See this [blog post on Kramdown and Rouge](#) for more details.

You can use standard Multimarkdown syntax for tables. You can also use fenced code blocks with lexers specifying the type of code. The configuration file shows the Markdown processor and extension:

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

## Automatic mini-TOCs

By default, a TOC appears at the top of your pages and posts. If you don't want the TOC to appear for a specific page, such as for a landing page or other homepage, add `toc: false` in the frontmatter of the page.

The mini-TOC requires you to use the `##` Markdown syntax for headings. If you use `<h2>` elements, you must add an ID attribute for the heading element in order for it to appear in the mini-TOC (for example, `<h2 id="mysampleid">Heading</h2>`).

## Headings

Use pound signs before the heading title to designate the level. Note that kramdown requires headings to have one space before and after the heading. Without this space above and below, the heading won't render into HTML.

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

**Result:**

## Second-level heading

---

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

**Result:**

### Third-level heading

---

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

**Result:**

#### *Fourth-level heading*

## Headings with ID Tags

If you want to use a specific ID tag with your heading, add it like this:

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

Then you can reference it with a link like this on the same page:

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

**Result:**

[Some link \(page 32\)](#)

For details about linking to headings on different pages, see [Automated links to headings on pages \(page 0\)](#).

## Specify a particular page layout

The configuration file sets the default layout for pages as the “page” layout.

You can create other layouts inside the layouts folder. If you create a new layout, you can specify that your page use your new layout by adding `layout: mylayout.html` in the page’s frontmatter. Whatever layout you specify in the frontmatter of a page will override the layout default set in the configuration file.

## Comments

Disqus, a commenting system, is integrated into the theme. In the configuration file, specify the Disqus code for the universal code, and Disqus will appear. If you don’t add a Disqus value, the Disqus form isn’t included.

## Page 2

**Summary:** This theme primarily uses pages. You need to make sure your pages have the appropriate frontmatter. One frontmatter tag your users might find helpful is the summary tag. This functions similar in purpose to the shortdesc element in DITA.

### Where to author content

Use a text editor such as Sublime Text, WebStorm, IntelliJ, or Atom to create pages. Atom is recommended because it's created by Github, which is driving some of the Jekyll development through Github Pages.

### Where to save pages

You can store your pages in any folder structures you want, with any level of folder nesting. The site output will pull all of those pages out of their folders and put them into the root directory. Check out the `_site` folder, which is where Jekyll is generated, to see the difference between your project's structure and the resulting site output.

The listing of all pages in the root directory (which happens when you add a permalink property to the pages) is what allows the relative linking and offline viewing of the site to work.

### Frontmatter

Make sure each page has frontmatter at the top like this:

```

---
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
---

```

Frontmatter is always formatted with three hyphens at the top and bottom. Your frontmatter must have a `title` and `permalink` value. All the other values are optional.

Note that you cannot use variables in frontmatter.

The following table describes each of the frontmatter that you can use with this theme:

Frontmatter	Required?	Description
<b>title</b>	Required	The title for the page
<b>tags</b>	Optional	Tags for the page. Make all tags single words, with underscores if needed (rather than spaces). Separate them with commas. Enclose the whole list within brackets. Also, note that tags must be added to <code>_data/tags_doc.yml</code> to be allowed entrance into the page. This prevents tags from becoming somewhat random and unstructured. You must create a tag page for each one of your tags following the pattern shown in the tags folder. (Tag pages aren't automatically created.)
<b>keywords</b>	Optional	Synonyms and other keywords for the page. This information gets stuffed into the page's metadata to increase SEO. The user won't see the keywords, but if you search for one of the keywords, it will be picked up by the search engine.

Frontmatter	Required?	Description
<b>last_updated</b>	Optional	The date the page was last updated. This information could be helpful for readers trying to evaluate how current and authoritative information is. If included, the <code>last_updated</code> date appears in the footer of the page in small font.
<b>sidebar</b>	Required	Refers to the sidebar data file for this page. Don't include the ".yml" file extension for the sidebar — just provide the file name. If no sidebar is specified, this value will inherit the <code>default</code> property set in your <code>_config.yml</code> file for the page's frontmatter.
<b>summary</b>	Optional	A 1-2 word sentence summarizing the content on the page. This gets formatted into the summary section in the page layout. Adding summaries is a key way to make your content more scannable by users (check out <a href="#">Jakob Nielsen's site</a> for a great example of page summaries.) The only drawback with summaries is that you can't use variables in them.
<b>permalink</b>	Required	The permalink <i>must</i> match the filename in order for automated links to work. Additionally, you must include the ".html" in the filename. Do not put forward slashes around the permalink (this makes Jekyll put the file inside a folder in the output). When Jekyll builds the site, it will put the page into the root directory rather than leaving it in a subdirectory or putting it inside a folder and naming the file <code>index.html</code> . Having all files flattened in the root directory is essential for relative linking to work and for all paths to JS and CSS files to be valid.

Frontmatter	Required?	Description
<b>datatable</b>	Optional	'true'. If you add <code>datatable: true</code> in the frontmatter, scripts for the <a href="#">jQuery Datatables plugin</a> get included on the page. You can see the scripts that conditionally appear by looking in the <code>_layouts/default.html</code> page.
<b>toc</b>	Optional	If you specify <code>toc: false</code> in the frontmatter, the page won't have the table of contents that appears below the title. The toc refers to the list of jump links below the page title, not the sidebar navigation. You probably want to hide the TOC on the homepage and product landing pages.

## Colons in page titles

If you want to use a colon in your page title, you must enclose the title's value in quotation marks.

## Page names and excluding files from outputs

By default, everything in your project is included in the output. You can exclude all files that don't belong to that project by specifying the file name, the folder name, or by using wildcards in your configuration file:

`exclude:`

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

Wildcards will exclude every match after the `*`.

## Saving pages as drafts

If you add `published: false` in the frontmatter, your page won't be published. You can also move draft pages into the `_drafts` folder to exclude them from the build. With posts, you can also keep them as drafts by omitting the date in the title.

## Markdown or HTML format

Pages can be either Markdown or HTML format (specified through either an `.md` or `.html` file extension).

If you use Markdown, you can also include HTML formatting where needed. But if your format is HTML, you must add a `markdown="1"` attribute to the element in order to use Markdown inside that HTML element:

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

For your Markdown files, note that a space or two indent will set text off as code or blocks, so avoid spacing indents unless intentional.

If you have a lot of HTML, as long as the top and bottom tags of the HTML are flush left in a Markdown file, all the tags inside those bookend HTML tags will render as HTML, regardless of their indentation. (This can be especially useful for tables.)

## Page names

I recommend prefixing your page names with the product, such as `"mydoc_pages"` instead of just `"pages."` This way if you have other products that also have topics with generic names such as `"pages,"` there won't be naming conflicts.

Additionally, consider adding the product name in parentheses after the title, such as `"Pages (Mydoc)"` so that users can clearly navigate different topics for each product.



## Kramdown Markdown

Kramdown is the Markdown flavor used in the theme. This mostly aligns with Github-flavored Markdown, but with some differences in the indentation allowed within lists. Basically, Kramdown requires you to line up the indent between list items with the first starting character after the space in your list item numbering. See this [blog post on Kramdown and Rouge](#) for more details.

You can use standard Multimarkdown syntax for tables. You can also use fenced code blocks with lexers specifying the type of code. The configuration file shows the Markdown processor and extension:

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

## Automatic mini-TOCs

By default, a TOC appears at the top of your pages and posts. If you don't want the TOC to appear for a specific page, such as for a landing page or other homepage, add `toc: false` in the frontmatter of the page.

The mini-TOC requires you to use the `##` Markdown syntax for headings. If you use `<h2>` elements, you must add an ID attribute for the heading element in order for it to appear in the mini-TOC (for example, `<h2 id="mysampleid">Heading</h2>`).

## Headings

Use pound signs before the heading title to designate the level. Note that kramdown requires headings to have one space before and after the heading. Without this space above and below, the heading won't render into HTML.

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

**Result:**

## Second-level heading

---

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

**Result:**

### Third-level heading

---

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

**Result:**

#### *Fourth-level heading*

## Headings with ID Tags

If you want to use a specific ID tag with your heading, add it like this:

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

Then you can reference it with a link like this on the same page:

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

**Result:**

[Some link \(page 40\)](#)

For details about linking to headings on different pages, see [Automated links to headings on pages \(page 0\)](#).

## Specify a particular page layout

The configuration file sets the default layout for pages as the “page” layout.

You can create other layouts inside the layouts folder. If you create a new layout, you can specify that your page use your new layout by adding `layout: mylayout.html` in the page’s frontmatter. Whatever layout you specify in the frontmatter of a page will override the layout default set in the configuration file.

## Comments

Disqus, a commenting system, is integrated into the theme. In the configuration file, specify the Disqus code for the universal code, and Disqus will appear. If you don’t add a Disqus value, the Disqus form isn’t included.

## Page 3

**Summary:** This theme primarily uses pages. You need to make sure your pages have the appropriate frontmatter. One frontmatter tag your users might find helpful is the summary tag. This functions similar in purpose to the shortdesc element in DITA.

### Where to author content

Use a text editor such as Sublime Text, WebStorm, IntelliJ, or Atom to create pages. Atom is recommended because it's created by Github, which is driving some of the Jekyll development through Github Pages.

### Where to save pages

You can store your pages in any folder structures you want, with any level of folder nesting. The site output will pull all of those pages out of their folders and put them into the root directory. Check out the `_site` folder, which is where Jekyll is generated, to see the difference between your project's structure and the resulting site output.

The listing of all pages in the root directory (which happens when you add a permalink property to the pages) is what allows the relative linking and offline viewing of the site to work.

### Frontmatter

Make sure each page has frontmatter at the top like this:

```

---
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
---

```

Frontmatter is always formatted with three hyphens at the top and bottom. Your frontmatter must have a `title` and `permalink` value. All the other values are optional.

Note that you cannot use variables in frontmatter.

The following table describes each of the frontmatter that you can use with this theme:

Frontmatter	Required?	Description
<b>title</b>	Required	The title for the page
<b>tags</b>	Optional	Tags for the page. Make all tags single words, with underscores if needed (rather than spaces). Separate them with commas. Enclose the whole list within brackets. Also, note that tags must be added to <code>_data/tags_doc.yml</code> to be allowed entrance into the page. This prevents tags from becoming somewhat random and unstructured. You must create a tag page for each one of your tags following the pattern shown in the tags folder. (Tag pages aren't automatically created.)
<b>keywords</b>	Optional	Synonyms and other keywords for the page. This information gets stuffed into the page's metadata to increase SEO. The user won't see the keywords, but if you search for one of the keywords, it will be picked up by the search engine.

Frontmatter	Required?	Description
<b>last_updated</b>	Optional	The date the page was last updated. This information could be helpful for readers trying to evaluate how current and authoritative information is. If included, the <code>last_updated</code> date appears in the footer of the page in small font.
<b>sidebar</b>	Required	Refers to the sidebar data file for this page. Don't include the ".yml" file extension for the sidebar — just provide the file name. If no sidebar is specified, this value will inherit the <code>default</code> property set in your <code>_config.yml</code> file for the page's frontmatter.
<b>summary</b>	Optional	A 1-2 word sentence summarizing the content on the page. This gets formatted into the summary section in the page layout. Adding summaries is a key way to make your content more scannable by users (check out <a href="#">Jakob Nielsen's site</a> for a great example of page summaries.) The only drawback with summaries is that you can't use variables in them.
<b>permalink</b>	Required	The permalink <i>must</i> match the filename in order for automated links to work. Additionally, you must include the ".html" in the filename. Do not put forward slashes around the permalink (this makes Jekyll put the file inside a folder in the output). When Jekyll builds the site, it will put the page into the root directory rather than leaving it in a subdirectory or putting it inside a folder and naming the file <code>index.html</code> . Having all files flattened in the root directory is essential for relative linking to work and for all paths to JS and CSS files to be valid.

Frontmatter	Required?	Description
<b>datatable</b>	Optional	'true'. If you add <code>datatable: true</code> in the frontmatter, scripts for the <a href="#">jQuery Datatables plugin</a> get included on the page. You can see the scripts that conditionally appear by looking in the <code>_layouts/default.html</code> page.
<b>toc</b>	Optional	If you specify <code>toc: false</code> in the frontmatter, the page won't have the table of contents that appears below the title. The toc refers to the list of jump links below the page title, not the sidebar navigation. You probably want to hide the TOC on the homepage and product landing pages.

## Colons in page titles

If you want to use a colon in your page title, you must enclose the title's value in quotation marks.

## Page names and excluding files from outputs

By default, everything in your project is included in the output. You can exclude all files that don't belong to that project by specifying the file name, the folder name, or by using wildcards in your configuration file:

`exclude:`

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

Wildcards will exclude every match after the `*`.

## Saving pages as drafts

If you add `published: false` in the frontmatter, your page won't be published. You can also move draft pages into the `_drafts` folder to exclude them from the build. With posts, you can also keep them as drafts by omitting the date in the title.

## Markdown or HTML format

Pages can be either Markdown or HTML format (specified through either an `.md` or `.html` file extension).

If you use Markdown, you can also include HTML formatting where needed. But if your format is HTML, you must add a `markdown="1"` attribute to the element in order to use Markdown inside that HTML element:

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

For your Markdown files, note that a space or two indent will set text off as code or blocks, so avoid spacing indents unless intentional.

If you have a lot of HTML, as long as the top and bottom tags of the HTML are flush left in a Markdown file, all the tags inside those bookend HTML tags will render as HTML, regardless of their indentation. (This can be especially useful for tables.)

## Page names

I recommend prefixing your page names with the product, such as `"mydoc_pages"` instead of just `"pages."` This way if you have other products that also have topics with generic names such as `"pages,"` there won't be naming conflicts.

Additionally, consider adding the product name in parentheses after the title, such as `"Pages (Mydoc)"` so that users can clearly navigate different topics for each product.



## Kramdown Markdown

Kramdown is the Markdown flavor used in the theme. This mostly aligns with Github-flavored Markdown, but with some differences in the indentation allowed within lists. Basically, Kramdown requires you to line up the indent between list items with the first starting character after the space in your list item numbering. See this [blog post on Kramdown and Rouge](#) for more details.

You can use standard Multimarkdown syntax for tables. You can also use fenced code blocks with lexers specifying the type of code. The configuration file shows the Markdown processor and extension:

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

## Automatic mini-TOCs

By default, a TOC appears at the top of your pages and posts. If you don't want the TOC to appear for a specific page, such as for a landing page or other homepage, add `toc: false` in the frontmatter of the page.

The mini-TOC requires you to use the `##` Markdown syntax for headings. If you use `<h2>` elements, you must add an ID attribute for the heading element in order for it to appear in the mini-TOC (for example, `<h2 id="mysampleid">Heading</h2>`).

## Headings

Use pound signs before the heading title to designate the level. Note that kramdown requires headings to have one space before and after the heading. Without this space above and below, the heading won't render into HTML.

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

**Result:**

## Second-level heading

---

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

**Result:**

### Third-level heading

---

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

**Result:**

#### *Fourth-level heading*

## Headings with ID Tags

If you want to use a specific ID tag with your heading, add it like this:

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

Then you can reference it with a link like this on the same page:

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

**Result:**

[Some link \(page 48\)](#)

For details about linking to headings on different pages, see [Automated links to headings on pages \(page 0\)](#).

## Specify a particular page layout

The configuration file sets the default layout for pages as the “page” layout.

You can create other layouts inside the layouts folder. If you create a new layout, you can specify that your page use your new layout by adding `layout: mylayout.html` in the page’s frontmatter. Whatever layout you specify in the frontmatter of a page will override the layout default set in the configuration file.

## Comments

Disqus, a commenting system, is integrated into the theme. In the configuration file, specify the Disqus code for the universal code, and Disqus will appear. If you don’t add a Disqus value, the Disqus form isn’t included.