Jekyll theme for documentation — designers

version 4.0

Introduction

Overview

This site provides documentation, training, and other notes for the Jekyll Documentation theme. There's a lot of iekylAation about how to do a variety of things here, and it's not all unique to this theme. But by and large, understanding how to do things in Jekyll depends on how your theme is coded. As a result, these additional detai6s are provided.

The instructions here are geared towards technical writers working on documentation. You may have a team of one or more technical writers working on documentation for multiple projects. You can use this same theme to author all of your documentation for each of your products. The theme is set up to push out documentation for multiple projects all from the same source. You can also share content across projects.

Survey of features

I'm using this theme for my documentation projects, building about 20 different outputs for various products, versions, languages, and audiences from the same

1. Build the default project

Summary: To get started with this theme, first make sure you have all the prerequisites in place; then build the theme following the sample build commands. Because this theme is set up for single sourcingthe sample

You can either download the theme files directly by clicking the Download Zip

9. Press Ctrl+C in Terminal to shp3 49h+C

2. Copy and customize the mydoc folder in configs

In the configs folder, copy the mydoc folder and its contents. Rename it to acme, and then rename each of the config_ files to the outputs you need for your acme project.

In this theme, each output requires a separate config file. If you have 10 audiences and you want separate sites for each, then then you'll have 10 config files in this directory.

More details about customizing the settings in the configuration files will be

- mydoc_4_publish.sh
- mydoc_all.sh

6. Copy the URL generator text file

In the root directory, copy urls_mydoc.txt and duplicate it. Change the suffix to urls_acme.txt.

> Tip: In this step, you're just duplicating project files. In later steps, you'll actually customize all of the settings.

Next: 3. Decide on your project's attributes (page 15)

2. Add a new proje

3. Decide on your project's attributes

Summary: Each project has attributes that define the audience, platform, product, version, and output. These attributes are used in generating the outputs. The attributes function as filtering conditions that determine what content gets included in the navigation.

[redacted]

•:

dDecrsioes b oun thefollowtiny:

FIELD	REQUIRED?	DESCRIPTION
homepage_title	Required	You set the title for your homepage via this setting. This is because multiple projects are all using the same index.md as their homepage. Because index.md has `ELJBM>DB: QORB` in the frontmatter, the "page" layout will use the `ELJBM>DB<QFQIB` property from the configuration file instead of the traditional title in T316.2375 595.. In the sample theme fileshe "pahomepage title is "Jekyll Documentation Theme — wri95.s" or "Jekyll Documentation Theme — design5.s."
si9_title	Required	Appears in T31webpage title area (on T31browser tab, not in T31page viewing area). In the sample theme fileshe "pasi95 title is rendered as `{{ page.title }}{% endif %}
port	Required	The port used in T31preview mode. This is only for T31liv31preview and doesn't affect T31published output. If you serve multiple outputs simultaneme

FIELD	REQUIRED?	DESCRIPTION
redcarpet	Required	Extensions used with redcarpet. Yon can read more about the Red Carpet extensions here (https://github.com/vmg/redcarpet).
highlighter 	Required	The syntax highlighter used. Use `OLRDB` because it has fewer dependencies on your operating system (it doesn't require Python). However, you can alsoeratting s401.1166.2375 552.6582 Td/F9 9.4500 D

FIELD	REQUIRED?	DESCRIPTION
collections	Optional	Any specific collections (custom content types that extend beyond pages or posts) that you want to define. This theme defines a collection called tooltips. You access this collection by using site.tooltips instead of site.pages or site.posts. Put the tooltip content types inside a folder in your project called _tooltips. Tooltips are useful for creating UI content. For more information about creating tooltips for UI text, see Help APIs and UI tooltips (page 130).
output	Optional	Boolean. Whether this build is `TB?` or `MAC` . This setting allows you to run conditions in your content such as `X% FC PFQB.LRQMRQ == MAC %Z` do this... `X% BKAFC %Z` . Limit the options to just `TB?` or `MAC` for this setting.
github_editme_path	Optional	A path to configure the Github Edit Me button. Put the path to the branch on Github where you want to edit the theme. Here's a sample: `QLJGLEKPLK1492/AL@RJBKQ>QFLK-QEBJB-GBHVII/BAFQ/OBSFBTP` . In this case, "re-

FIELD	REQUIRED?	DESCRIPTION
footer_image_location	Optional	The image used in the footer to brand yota000Ie. Store this image in the common_images folder so that it's not excluded by a particular project. Example: ../common_images/company_logo.png
theme_file	Optional	The theme used for the output. Currently there are two options: theme-green.css or theme-blue.css. These themes cover both web and PDF output. The themes have the same style and layott. They only differ in the accent color for the top nav bar, buttons, hyperlinks, and other smalETBT316.2375 582.99462.9

FIELD	REQUIRED?	DESCRIPTION
baseurl	The subdirectory after 2he url where 2he content is stored. In 2he sample 2heme files for 2he designers output, 2his is `/ABPFDKBOP` .	
port	The port required by 2he preview server.	
output	`TB?` or `MAC` . This setting allows you 2o construct conditional statements in your content 2o check whe2her output is web or pdf. This setting can help you filter out content 2hat doesn't fi2 well in2o a PDF (such as dynamic web elements). In particular, 2he Prince XML script conflicts with any JavaScript on 2he page, so you want 2o filter out 2he JavaScript from 2he PDF-friendly HTML output 2hat Prince consumes.	
print_title	The title for 2he PDF. In 2he sample 2heme files for designers output, 2he print title is "Jekyll Documentation Theme for Designers"	
print_subtitle	The subtitle for 2he PDF. In 2he sample 2heme files, 2he subtitle is "jE4p31.3750 332.495(dynamic we.2d 4.0.igners")	

You need to duplicate this block for each output you have.

Once you've duplicated the block, make a few customizations:

- In each place that "mydoc" appears, change "mydoc" to "acme".
- Use the same attributes5000project, audieOnc, verstio,0product, and•

Ift youdon'tu hav a•

All of the attributes (which are defined in the conditions.html file) must be met in order to display in the navigation. The attributes must be present on both the

```
- QFQIB:
  :
```

7. Configure the top navigation

Summary: The top navigation provides either single links or a drop-down menu. There are sohi49ther features, such as a feedback email, custom menu, and popout link.

Getting Started

Changing the top navigation

9. Set up Prince XML

> Summary: Prince XML is the utility used for creating PDFs. Though not free, this utility gets a list of links and compiles them into a PDF.

The docs folder contains the source of all your files, while the doc_outputs contains the site outputs.

Each of you32.5Cnfiguration files needs to have a destination like this:

The doc_outputs/mydoc/designers contains the site output from mydoc, so when you push content from this folder into Github, you're actually pushing the HTML site output into Github, not the mydoc source files.

Your delivery team can also grab the site output from these repos. After downloading it, the person unzips the folder and sees the website folders inside.

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 `I>VLRQ: JVI>VLRQ.EQJI` 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 code isn't included.

Posts

If you don't see the Edit File Templates option, you may need to create a file template first. Go to File > Default Settings > Editor > File and Code Templates. Create a new file template with an md extension, and then close and restart WebStorm. Then repeat this step and you will see the File Templates option appear in the right context menu.

2. In the upper-left corner of the dialog box that appears, click the + button to create a new template.
3. Name it something like Jekyll page. Insert the frontmatter you want, and save it.

   To use the Jekyll template, when you create a new file in your WebStorm project, you can select your Jekyll file template.

Disable pair quotes

By default, each time you type ' , WebStorm will pair the quote (creating two quotes). You can disable this by going to WebStorm > Preferences > Editor > Smartkeys. Clear the Insert pair quotes check box.

Conditional logic

> Summary: You can implement advanced conditional logic that includes if statements, or statements, unless, and more. This conditional logic facilitates single sourcing scenarios in which you're outputting the same content for different audiences.

About Liquid and conditional statements

If you want to create different outputs for different audiences, you can do all of this using a combination of Jekyll's Liquid markup and values in your configuration file.

You can then incorporate conditional statements that check the values in the configuration files.

> Tip: Definitely check out Liquid's documentation (http://docs.shopify.com/themes/liquid-documentation/basics) for more details about how to use operators and other liquid markup. The notes here are a small, somewhat superficial sample from the site.

Where to store filtering values

You can filter content based on values that you have set either in your config file or in a file in your _data folder. If you set the attribute in your config file, you need to r1start the Jekyll server to see the changes. If you set the value in a file in your _data folder, you don't need to r1start the server when you make changes.

Required conditional attributes

This theme r1quires you to add the following attributes in your configuration file:

- project
- audience
- product
- platform

- version

If you've ever used DITA, you probably recognize these attributes, since DITA has mostly the same ones. I've found that most single_sourcing projects I work on can be sliced and diced in the ways I need using these conditional attributes.

If you're not single sourcing and you find it annoying having to specify these attributes in your sidebar, you can rip out the logic from the sidebar.html, topnav.html file and any other places where conditions.html appears; then you wouldn't need these attributes in your configuration file.

Conditional logic based on config file value

Here's an example of conditional logic based on a value in the configs/ config_writer.yml file. In my config_writer.yml file, I have the following:

```
>RAFBK@B: TOFQBOP
```

On a page in my site (it can be HTML or markdown), I can conditionalize content using the following:

Storing conditions in the _data folder

Here's an example of using conditional logic based on a value in a data file:

```
X% FC PFQB.A>Q>.LMQFLKP.LRQMRQ == ">IME>" %Z
PELT QEFP @LKQBKQ...
X% BIPFC PFQB.A>Q>.LMQFLKP.LRQMRQ == "?BQ>" %Z
PELT QEFP @LKQBKQ...
X% BIPB %Z
QEFP PELTP FC KBFQEBO LC QEB >?LSB QTL FC @LKAFQFLKP >OB JBQ.
X% BKAFC %Z
```

To use this, I would need to have a _data folder called options where the `LRQMRQ` property is stored.

I don't really use the _data folder as much for project options. I store them in the configuration file because I usually want different projects to use different values for the same property.

For example, maybe a file or function name is called something different for different audiences. I currently single source the same content to at least two audiences in different markets.

For the first audience, the function name might be called `DBKBO>QB` , but for the second audience, the same function might be called called `BUM>KA` . In my content, I'd just use `XXPFQB.CRK@QFLKZZ` . Then in the configuration file I change its value appropriately for the audience.

Specifying the location for _data

You can also specify a `A>Q><PLRO@B` for your data location in your configuration file. Then you aren't limited to simply using `<A>Q>` to store your data files.

For example, suppose you have 2 projects: alpha and beta. You might store all the data files for alpha inside data_alpha, and all the data files for beta inside data_beta.

In your alpha configuration file, specify the data source like this:

```
A>Q><PLRO@B: A>Q><>IME>
```

How to use collections

I haven't found a huge use for collections in normal documentation. However, I did find a use for collections in generating a tooltip file that would be used for delivering tooltips 670a u1Uo1vmfac usrom texttip fions themal documentatioSeeonsfiUIing tooltonspa hu1

Sidebar Navigation

> Summary: The sidebar navigation uses a jQuery component called Navgoco. The sidebar is a somewhat complex part of the theme that remembers your current page, highlights the active item, stays in a

You can see that the `BUQBOK>I<ROI` is a condition that applies a different formatting. Althoughat is feature is available, I recommend putting any external navigation links in the top navigation bar instead of the side navigation bar.

Sidebar item highlighting

The sidebar.html file inserts an `>@QFSB` class into the sidebar element when the `ROI` attribute in the sidebar data file matches the page URL.

For example, the sidebar.html file contains the following code:

```
    X% BIPFC
```

Additionally, you must create a tag archive page similar to the other pages named

W21!t&%C$TVW!IáWyô

```
'B3@4$2ãA M>DB:
<RI>
X% CLO M>DB FK PF3 ó      %Z
X% CLO Q>D FK M>DB.Q>D  %Z
X% FC Q>D == "DB3d$2ãA"         %Z
```

%ZFC

%TS=w"DB5'$ãa.1wVG=$ U1.1wVG=$$&P]

[illegible]

```
'BQQFKD PQ>OQBA M>DBP:
<RI>
X% >PPFDK PLOQBA<M>DBP = (PFQB.M>DBP
```

With posts (instead of pages), since you can access just the posts inside `MLPQP.Q>D.Q>DK>JB`, you can be a lot more efficient with the looping.

Still, if the build times are getting long (e.g., 1 or 2 minutes per build), look into reducing the number of `CLO` loops on your site.

Empty tags?

If your page shows "tags:" at the bottom without any value, it could mean a couple of things:

- You're using a tag that isn't specified in your allowed tags list in your tags.yml file.
- You have an empty `Q>DP: 8:` property in your frontmatter.

If you don't want tags to appear at all on your page, remove the tags property from your frontmatter.

Remembering the right tags

Since you may have many tags and find it difficult to remember what tags are allowed, I recommend creating a template that prepopulates all your frontmatter with all possible tags. Then just remove the tags that don't apply.

See WebStorm Text Editor (page 50) for tips on creating file templates in WebStorm.

With weight, you could use 1, 2, 3, etc.., but Jekyll will treat 10 as coming after 1. This is why I use 1.0 and 1.1, 1.2, etc.

If you do use whole numbers, change the `MIRP: "0.1"` to `MIRP: "1"`.

Additionally, if your page names are prefaced with numbers, such as "1. Download the code," then the `XXM.TBFDEQZZ` will create a duplicate number. In that case, just remove the `XXM.TBFDEQZZ` from both code samples here.

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

On each series page, add a link to the series button at the top and a link to the next button at the bottom.

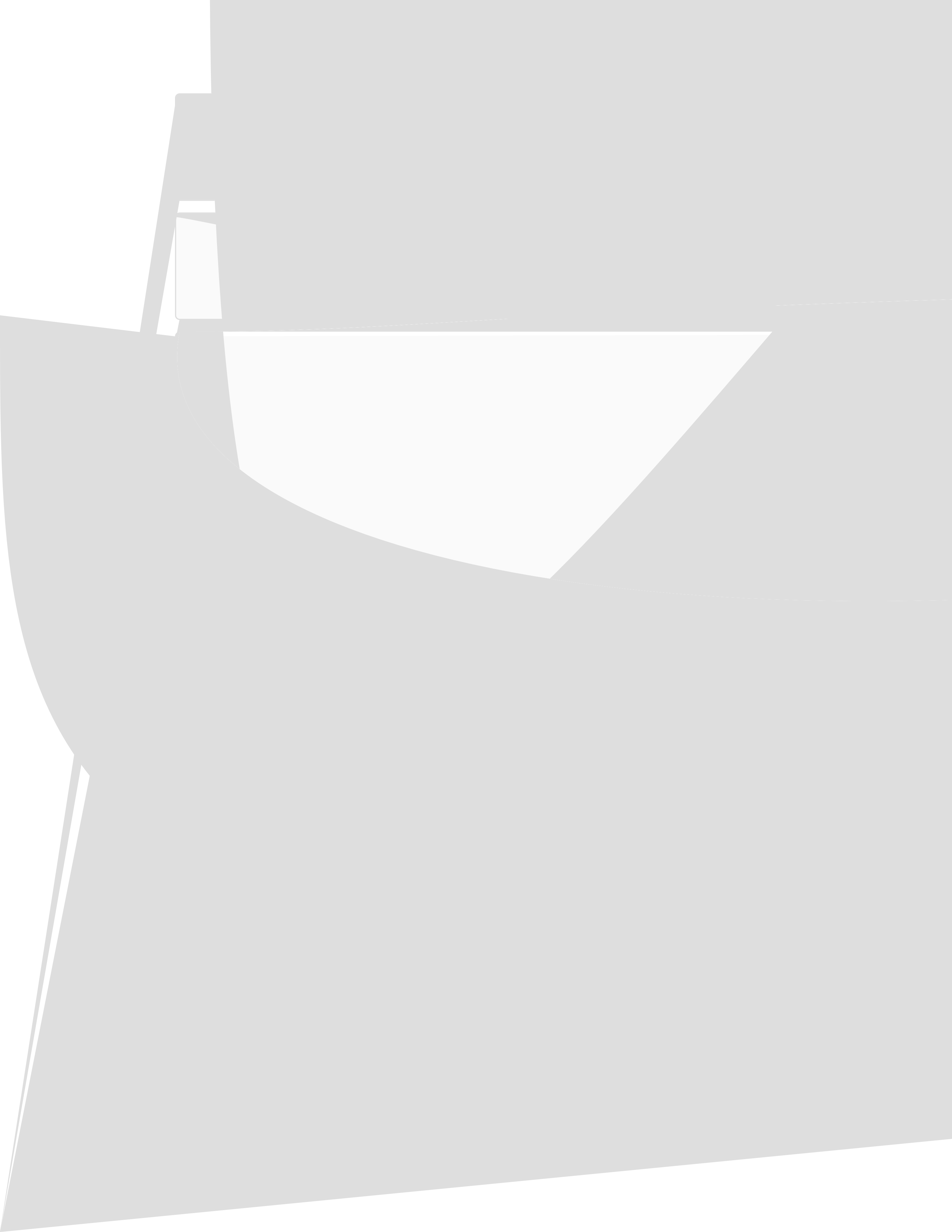
```
<!-- VLRO COLKQJ>QQBO DLBP EBOB -->



<!-- VLRO M>DB @LKQBKQ DLBP EBOB ... -->
```

Changing the series drop-down color

The Bootstrap menu uses the `MOFJ>OV` class for styling. If you change this class in your theme, the Bootstrap menu should automatically change color as well. You can also just use another Bootstrap class in your button code. Instead of `?QK-MOFJ>OV`, use `?QK-FKCL` or `?QK-T>OKFKD`. See Labels (page 88) for more Bootstrap button classes.

Using a collection with your series

4500 Tf3 copying and pasting the button includes on each of your series, you could also create a collection and define a layout for the collection that has the include code. For more information on creating collections, see Collections (page 60).



Blast a warning to users

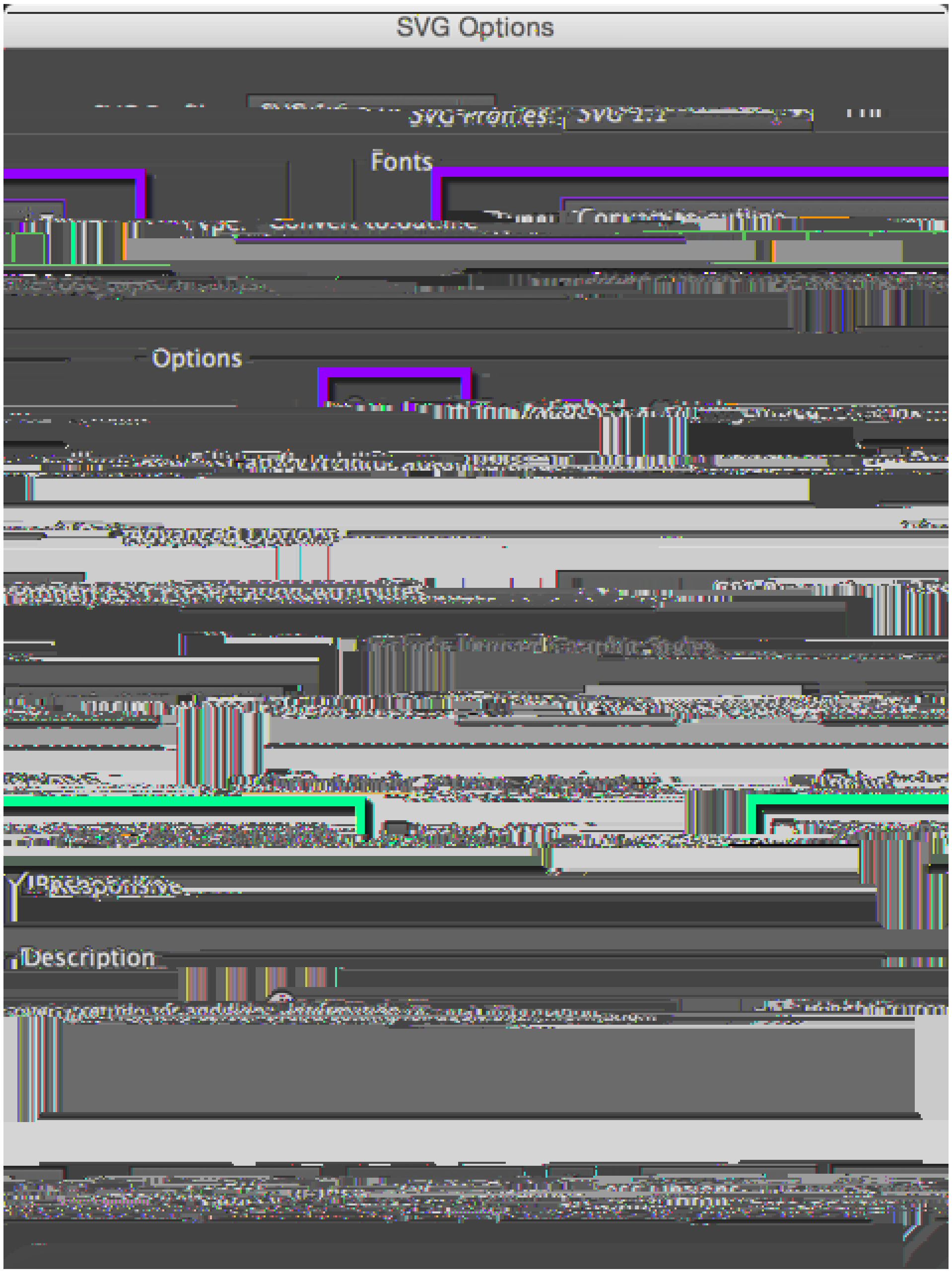
If you want to blast a warning to users on every page, add the alert or callout to the layouts/page.html page right below the frontmatter. Every page using the page layout (all, by defaut) wil4 29o 5fw mess/pagrs

ln ev pagHTMLge.ndIf yocan'tto unMarkdown inside of HTML tagsgrs

With Font Awesome, you always use the `F` tag with the appropriate class. You also implement `C>` as a base class first. You can use font awesome icons inside other elements. Here I'm using a Font Awesome class inside a Bootstrap alert:

```
<AFS @I>PP=">IBOQ >IBOQ-A>KDBO" OLIB=">IBOQ"><F @I>PP="C> C>-BU
@I>J>QFLK-@FO@IB"></F> <?>6>OKFKD: </?>3EFP FP > PMB@F>I T>OKFK
D JBPP>DB.
```

Here's the result:



SVG Options

SVG Profiles: SVG 1.1

Fonts

Type: Convert to outline

Options

Advanced Options

Include Unused Graphic Styles

Responsive

Description

(& ! #$ # ' ! "& &#") % " $% % $ '

Labels

> Summary: Labels are just a simple Bootstrap component that you can includ0114 your pages as needed. They represent one of many Bootstrap options you can includ0114 your theme.

About labels

Labels might com0114 handy for adding button-like tags next to elements, such as POST, DELETE, UPDATE methods for endpoints. You can use any classes from Bootstrap 14 your content.

OR

```
<> EOBC="P>JMIB.EQJI">2>JMIB</>>
```

However, what happens when you change the page's title or link? Jekyll doesn't automatically pull in the page's title when you create links.

In my experience, coding links like this results in a lot of broken links.

Managed links

For internal links, I've found that it's a best practice to store the link in a YAML file that is derived from the table of contents.

The theme has a 5The theme hat is derived s17 7ppen_mydoc.txt. Tt of you esults in3nL6Aitle whe1mydoc

Jekyll gets around some of this link path variation by using `?>PBROI` and including code that prepends the `?>PBROI` before a link. This converts the links into absolute rather than relative links.

With absolute links, the site only displays at the `?>PBROI` you configured. This is problematidjE2 tech docs because you usually need to move files around from one folder to another based on versions you're archiving E2 when you're moving your documentation from a5 At to testing to production folders.

Limitations with links

One of the shortcomings in this .501eiis .5at the link titles in the sidebar and inline links don't necessarily have to match the titles specified on each page. You have to manually keep the page titles in sync with the titles listed in the sidebar and top navigation. Although I could potentially get rid of the titles key in the artidle topic, it would makeiit more difficult to know what page you're editing.

Navtabs

> Summary: Navtabs provide a tab-based navagation directly in your content, allowing users to click from tab to tab to see different panels of content. Navtabs are especially helpful for showing code samples for different programming languages. The only downside to using

Code

Here's the code for the above (with the filler text abbreviated):

```
<RI FA="MOLCFIB3>?P" @I>PP="K>S K>S-Q>?P">
    <IF @I>PP=">@QFSB"><> EOBC="#MOLCFIB" A>Q>-QLDDIB="Q>?">0OL
M""0̆
```

If you want the player button in the upper-left corner (which is the default), remove the from the video class.

Here are

> service such as Youtube or Vimeo. Also, note that Github's Large File Storage (which does handle large files) isn't compatible with Github Pages.

```
<Q>?IB FA="P>JMIB3>?IB" @I>PP="AFPMI>V">
   <QEB>A>
      <QO>
         <QE>0>O>JBQBO</QE>
         <QE>$BP@OFMQFLK</QE>
         <QE>3VMB</QE>
         <QE>$BC>RIQ 5>IRB</QE>
      </QO>
   </QEB>A>
   <Q?LAV>
      <QO>
         <QA>0>O>JBQBO 1</QA>
         <QA>2>JMIB ABP@OFMQFLK
         </QA>
         <QA>2>JMIB QVMB</QA>
         <QA>2>JMIB ABC>RIQ S>IRB</QA>
      </QO>

         <QA>0>O>JBQBO 2</QA>
         <QA>2>JMIB ABP@OFMQFLK
         </QA>
         <QA>2>JMIB QVMB</QA>
```

<QO>

</S$0dOOVMB

FOOD	DESCRIPTION	CATEGORY	SAMPLE TYPE
Apples	A small, somewhat round and often red-colored, crispy fruit grown on trees.	Fruit	Fuji
Bananas	A long and curved, often-yellow, sweet and soft fruit that grows in bunches in tropical climates.	Fruit	Snow
Kiwis	A small, hairy-skinned sweet fruit with green-colored insides and seeds.	Fruit	Golden
Oranges	A spherical, orange-colored sweet fruit commonly grown in Florida and California.	Fruit	Navel

Notice a few features:

siarch the table. Whd(you typ Tdwor, the table filters500 670.3KCe5jETBT770.8250 392to opich yourwor. features:) TjETBT231.375

Syntax highlighting

> Summary: You can apply syntax highlighting to your code. This theme uses pygments and applies color coding based on the lexer you specify.

About syntax highlighting

For syntax highlighting, use fenced code blocks optionally followed by the language syntax you want:

```
===OR?V
    ABC CLL
      MRQP 'CLL'
    BKA
===
```

This looks as follows:

```
    ABC CLL
      MRQP 'CLL'
    BKA
```

Fenced code blocks require a blank line before and after.

If you're using an HTML file, you can also use the `EFDEIFDEQ` command with Liquid markup:

```
X% EFDEIFDEQ OR?V %Z
    ABC CLL
      MRQP 'CLL'
    BKA
X% BKAEFDEIFDEQ %Z
```

It renders the same:

Commenting on files

Summary: You can add a button to your pages that allows people to

Themes

> Summary:

Generate a PDF

When you generate a PDF, Prince XML will print "page 0" for any cross references it can't find. This lets you know that a particular link is bad because the page is missing.

If you have linke 0.5000 616.37021issing.

Generating PDFs

Summary: You can generate a PDF from your Jekyll project. You do this by creating a web version of your project that is printer friendly. You then use utility called Prince to iterate through the pages and create a PDF from them. It works quite well and gives you complete control to customize the PDF output through CSS, including page directives and dynamic tags from Prince.

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

The PDF configuration file will build on the settings in the regular configuration file but will some additional fields. Here's the configuration file for the config_designers.yml12.5000 648.2s theme:

In the configuration file, customize the values for the `MOFKQ<QFQIB`

The code in the tocpage.html is nearly identical to that of the sidebar.html page except that it includes the `PFQB` and `?>PBROI` for the URLs. This is essential for Prince to create the page numbers correctly with cross references.

There's another file (in the root directory of the theme) that is critical to the PDF generation pr420.s: prince-file-list.txt. This file simply iterates through the items in your sidebar and creates a list of links. Prince will consume the list of links from

In your exclude list for your alpha project, specify it as follows:

```
BU@IRAB:
- ?BQ><*
```

If you have more sophisticated exclusion, add another level to your file names. For example, if you have different programming languages you want to filter by, add this:

- alpha_java_sample.html
- alpha_cpp_sample.html

Then you exclude files for your Alpha C++ project as follows:

```
BU@IRAB:

- >IME><G>S><*
- ?BQ><*
```

And you exclude files for your Alpha Java project as follows:

```
BU@IRAB:

- >IME><@MM<*
- >IME><?BQ><*
```

When you exclude folders, include the trailing slash at the end of the folder name:

```
BU@IRAB:
- FJ>DBP/>IME>/
```

There isn't a way to automatically exclude anything. By default, everything is included unless you explicitly list it under the exclude section.

Excluding draft content

If you're working on a draft, put it inside the _drafts folder Tj1iuT13.8571T112.5000 108.2147 Td/F9 D523E

Limitations

Help APIs and UI tooltips

> Summary: You can loop through files and generate a JSON file that developers can consume like a help API. Developers can pull in values from the JSON inte interface elements, styling them as popovers for user interface text, for example. The beauty of this method is that the UI text remains in the help system (or at least in a single JSON file delivered to the dev team) and isn't hard-coded inte the UI.

Full code deme of content API

You can create a help API that developers can use to pull in content.

For the full code deme, see the notes in the tooltip deme.

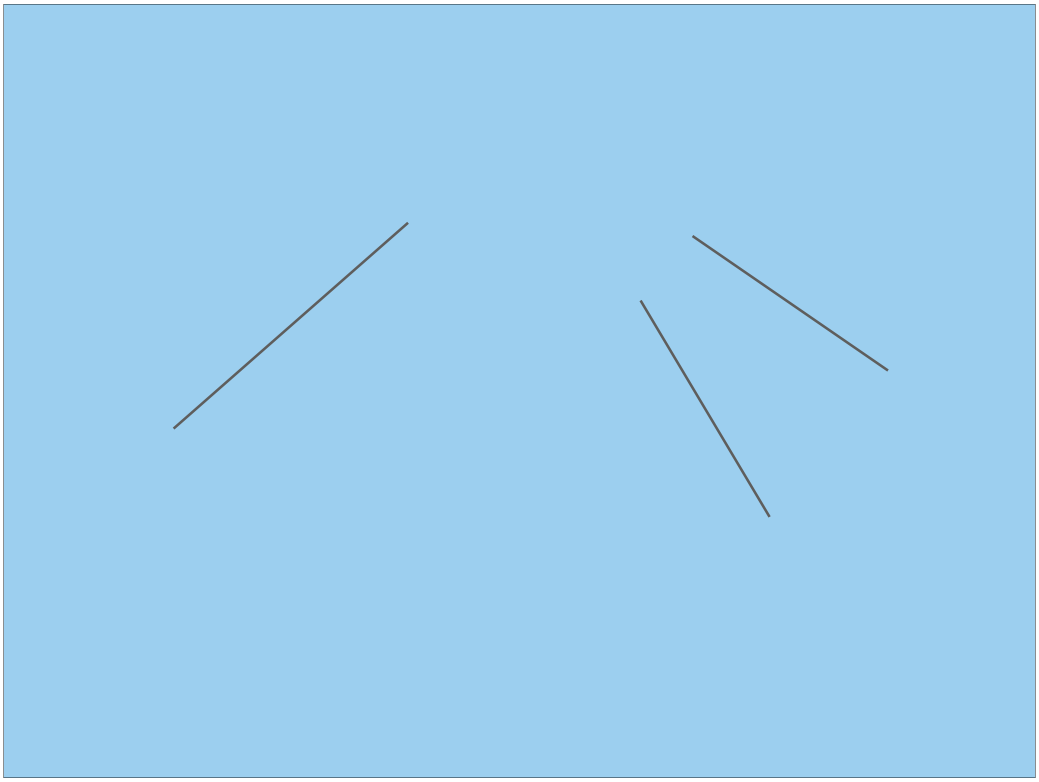
In this deme, the popovers pull in and display content from the information in a mydoc_tooltips_source.json (page 0) file located in the same directory.

Instead of placing the JSON source in the same directory, you could also host the JSON file on another site.

Additionally, instead of tooltip popovers, you could also print content directly te the page. Basically, whatever you can stuff inte a JSON file, developers can integrate it onte a page.

Diagram overview

Here's a diagram showing the basic idea of the help API:



```
X
  "BKQOFBP": 8
    X
      "FA": "?>PB?>II",
      "?LAV": "">PB?>II FP @LKPFABOBA !JBOF@>'P M>PQQFJB PMLO
Q, QELRDE QE>Q J>V ?B JLOB LC > EFPQLOF@>I QBOJ QE>K > @ROOBKQ
LKB. 3EBOB'P > ILQ JLOB BU@FQBJBKQ >?LRQ CLLQ?>II QE>K ?>PB?>I
```

You can store your mydoc_tooltips_source.json file anywhere you want, but to me it make sense to store it inside a tooltips folder fort2112.ecific project. Thisu wye

[illegible]

Warning: Note: Make sure your JSON file is valid. Otherwise, this method won't work. I use the JSON Formatter extension for Chrome (https://chrome.google.com/webstore/detail/json-formatter/

Reuse Demo

SPORT	COMMENTS
Basketball	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 groundo(Barequiress of - circular) TjETBT211.3750 641.1rsc

[illegible]) TjETBT493ET50 656.snoozecompwatch. fl et amosttpart.(COMMENTS) TjETBT4

Search configuration

Summary: The search feature uses JavaScript to look for keyword matches in a JSON file. The results show instant matches, but it doesn't provide a search results page like Google. Also, sometimes invalid formatting can break the JSON file.

About search

The search is configured through the search.json file in the root directory. Take a look at that code if you want to change what fields are included.

The search is a simple search that looks at content in pages. It looks at titles, summaries, keywords, tags, and bodies.

However, the search doesn't work like google — you can't hit return and see a list of results on the search results page, with the keywords in bold. Instead, this search shows a list of page titles that contain keyword matches. It's fast, but simple.

Excluding pages form search

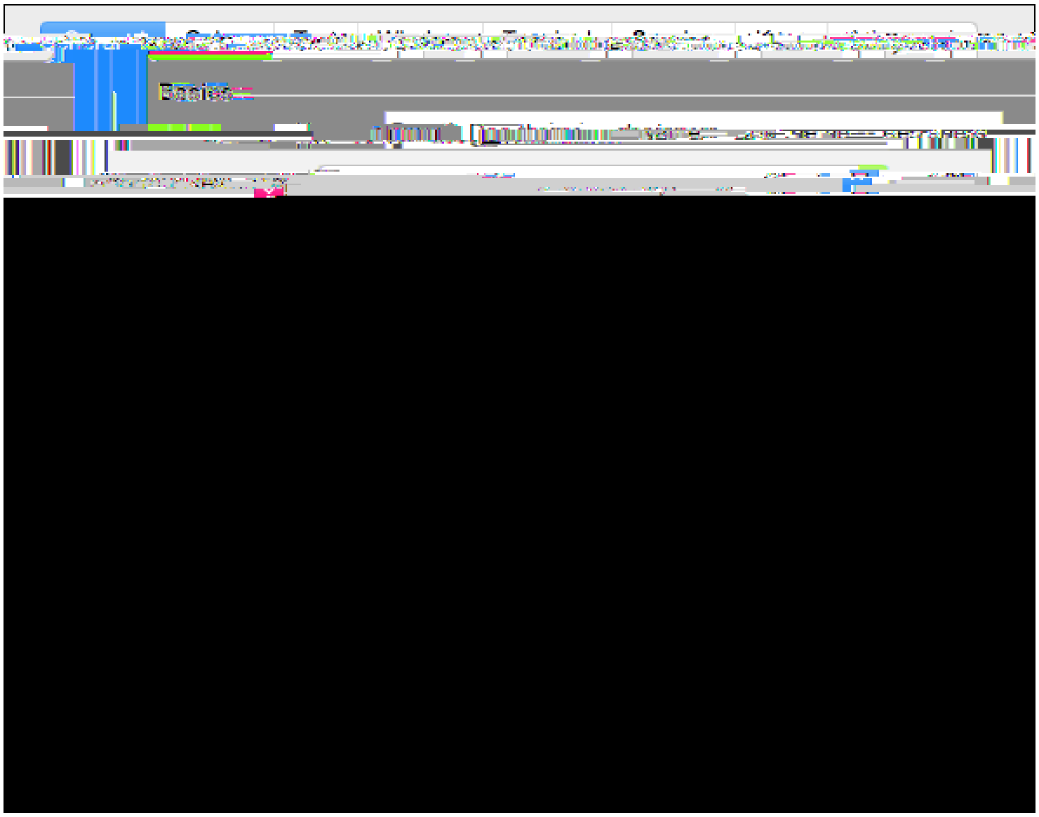
By default, every page is included in the search. Depending on the type of content

If this happens, go directly to the search.json file in your browser, and then copy the content. Go to a JSON validator (http://jsonlint.com/) and paste in the content. Look for the line causing trouble. Edit the file to either exclude it from search or fix the syntax so that it doesn't invalidate the JSON.

The search.json file already tries to strip out content that would otherwise make the JSON invalid:

```
       "?LAV": "XX M>DB.@LKQBKQ Y PQOFM<EQJI Y PQOFM<KBTIFKBP Y
```

```
<!-- PQ>OQ PB>O@E -->
<AFS FA="PB>O@E-ABJL-@LKQ>FKBO">
<FKMRQ QV>MB="QBUQ" FA="PB>O@E-FKMRQ" MI>@BELIABO="PB>O@E..."
<RI FA="OBPRIQP-@LKQ>FKBO"></RI>
</AFS>
<P@OF>MQ/P@@E="GP/GBHVII-PB>O@E.GP" QVMB="QBUQ/G>S>P@OFMQ"
MQ>
<P@OFMQ QVMB="QBUQ/G>S>P@OFMQ">
2FJMIB*BHVII2B>O@E.FKFQ(X
PB>O@E)KMRQ: AL@RJBKQ.DBQ%IBJBKQ"V)A('PB>O@E-FKMRQ'),
OBPRIQP#LKQ>FKBO: AL@RJBKQ.DBQ%IBJBKQ"V)A('B#Ñ
```



Launching a profile

Getting around the password prompts in SCP

You can publish your docs via SSH through a Terminal window or more likely, via a shell script that you simply execute as part of the publishing process. However, you will be prompted for your password with each file transfer unless you configure passwordless SSH.

The basic process for setting up password less SSH is to create a key on your own machine that you also transfer to the remote machine. When you use the SCP command, the remote machine checks that you have the authorized key and allows access without a password prompt.

To remove the password prompts when connecting to servers via SSH:

1. On your local machine, go to your .ssh directory:

```
@A [/.PPE
```

Note that any directory that starts with a dot, like .ssh, is hidden. You can view hidden folders by enabling them on your Mac. See this help topic (http://ianlunn.co.uk/articles/quickly-showhide-hidden-files-mac-os-x-mavericks/). Additionally, when you look at the files in a directory, use ls -a instead of just ls to view the hidden files.

If you don't have an .ssh directory, create one with `JHAFO .PPE`

Create a new key inside your .ssh directory:

```
PPE-HBVDBK -Q OP>
```

Press Enter. When prompted about "Enter file in which to save the key ...", press Enter again.

This will create a file called id_rsa.pub (the key) and id_rsa (your identification) in this .ssh folder.

When prompted for a passphrase for the key, just leave it empty and press Enter twice. You should see something like this:

```
P@M -O ../AL@<LRQMRQP/JVAL@/TOFQBOP <OTBå3$ä#$$Ä$3%4 O:@2>O@///P4FZI/
```

Knowledge-base layout

Summary: This shows a sample layout for a knowledge base. Each square could link to a tag archive page. In this example, font icons

single_sourcing

Publishing

Special layouts

falafel

Praesent auctor a enim non lacinia. Integer sodales aliquet mi vel dapibus. Donec consequat justo eget nisi lacinia, eu sodales ligula molestie. Sed sapien nulla, rhoncus at elementum a,

giraffe

Nullam venenatis at lectus sed pharetra. Sed hendrerit ligula lectus, non pellentesque diam faucibus sit amet. Aliquam dictum hendrerit pellentesque. Cras eu nisl sagittis, faucibus velit sit amet, sagittis odio. Donec vulputate ex vitae purus

hippo

Cras nec pretium nulla. Suspendisse tempus tortor vel venenatis pulvinar.

paranoia

xylophone

> Note: This was mostly an experiment to see if there was a better way to document a long JSON code example. I haven't actually used this approach in my own documentation.

- Pages (page 44)
- Sidebar Navigation (page 62)
- Support (page 0)
- Supported features (page 3)
- Troubleshooting (page 170)
- WebStorm Text Editor (page 50)

If you squish your screen small enough, at a certain breakpoint this style reverts to the regular `AI` class.

Although I like the side-by-side view for shorter definitions, I found it problematic with longer definitions.

```
GBHVII PBOSB --@LKCFD @LKCFD<?>PB.VO-@LKCFD [illegible]
```

Look in _includes/custom/conditions.html and make sure the conditional values alure match up with the values declared in the configuration file. Specifically, you need to make sure you've declared a value for project, product, platform, and version.

If you don't have any values for these properties, you still need to keep them in your configuration file. Just put something like `>II` as the value.

> Note: This theme is designed for single sourcing. If you're only building o8e site, you can remove these values from the _includes/sidebar.html file and _data/sidebar.yml files.

Understanding how the theme works can be helpful in troubleshooting. The _includes/sidebar.html file loops through the values in the _data/sidebar.yml file. Tlure are `FC` statements that check whealur the conditions (as specified in the conditions.html file) are met. If the sidebar.yml item has the right product,