

---

**Jun Ma**

***Release 01***

**Feb 19, 2020**



---

## Contents:

---

<b>1</b>	<b>Heading to Sphinx and Read the Docs</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Preparations . . . . .	1
1.3	Steps . . . . .	2
<b>2</b>	<b>Working with Sphinx</b>	<b>3</b>
2.1	Change a Theme . . . . .	3
2.2	Add a Logo Image . . . . .	3
2.3	Insert a Table . . . . .	4
2.4	Insert an Image . . . . .	4
2.5	Add the Comment Function Using Disqus . . . . .	4
<b>3</b>	<b>Working with Read the Docs</b>	<b>7</b>
3.1	Multi-version . . . . .	7
3.2	Multi-language for Localization . . . . .	7
<b>4</b>	<b>Working with Git</b>	<b>9</b>
4.1	Commit Changes to A New Branch . . . . .	9
4.2	Merge Changes into the Master Branch . . . . .	9
<b>5</b>	<b>Cooking</b>	<b>11</b>
5.1	Breakfast . . . . .	11
5.2	Lunch/Dinner . . . . .	12
	<b>Index</b>	<b>19</b>





---

## Heading to Sphinx and Read the Docs

---

---

**Note:** This page is only used to document the information I gathered and the process I went through when I did practices on Git, ReadtheDocs, and Sphinx.

---

### 1.1 Background

Treating documentation as code is becoming a major theme in the software industry.

The following advanced tools and platforms are widely used by both developers and technical writers.

- Sphinx provides a documentation generator that is best-in-class for software docs. Sphinx documents are written in the reStructuredText markup language. reStructuredText is a powerful language primarily because the syntax can be extended.
- Read the Docs is a hosting platform for Sphinx-generated documentation. It takes the power of Sphinx and adds version control, full-text search, and other useful features. It pulls down code and doc files from Git, Mercurial, or Subversion, then builds and hosts your documentation.
- GitHub is a code hosting platform for version control and collaboration.

### 1.2 Preparations

- Run **python-3.7.5-amd64.exe** to install Python 3.7.5
- Run `pip install -U Sphinx` in the command prompt to install Sphinx
- Run **Sublime Text Build 3211 x64 Setup.exe** to install Sublime
- Run **Git-2.25.0-64-bit.exe** to install Git
- Create an account in Read the Docs and Github

## 1.3 Steps

1. Run `$ sphinx-quickstart` in the command prompt to build a directory for Sphinx output.
2. Enrich the master file `index.rst` and other source files by using Sublime.

---

**Tip:** To learn more Sphinx syntax, refer to <https://www.sphinx-doc.org/en/master/usage/restructuredtext/index.html>.

---

3. Create an open-source repo in Github.
4. Commit the local directory and files to your Github Repo by running the following commands in Git Bash.
  - Verify Identity: `$ git config --global user.email "registered email address"`
  - Verify Identity: `$ git config --global user.name "registered user ID"`
  - Connect to Github Repo: `$ git remote add origin https://github.com/"UserID"/"RepoID".git`
  - Create a Pull Request and Merge: `$ git pull origin master --allow-unrelated-histories`
  - Add all files: `$ git add *`
  - Commit all files you added: `$ git commit -m "description"`
  - Push and merge updates: `$ git push -u origin master`
5. Link your GitHub repo to your Read the Docs account.
6. Build and View in Read the Docs.

### 2.1 Change a Theme

1. Run `pip install sphinx_rtd_theme` to install the *Read The Docs* theme.
2. In the `conf.py` file, add `html_theme = 'sphinx_rtd_theme'`.

### 2.2 Add a Logo Image

1. Place the **logo.png** file to the `source/_static` folder.
2. In the `conf.py` file:
  - Add `html_logo = './_static/logo.png'`.
  - Add the following argument:

```
html_theme_options = {
    'canonical_url': '',
    'analytics_id': 'UA-XXXXXXX-1', # Provided by Google in your dashboard
    'logo_only': True,                # Set as "True" to display logo only
    'display_version': True,
    'prev_next_buttons_location': 'bottom',
    'style_external_links': False,
    'vcs_pageview_mode': 'raw',
    'style_nav_header_background': '#2980B9',
    'collapse_navigation': True,
    'sticky_navigation': False,
    'navigation_depth': 4,
    'includehidden': True,
    'titles_only': False
}
```

## 2.3 Insert a Table

### 2.3.1 Input

=====	=====	=====
Header A	Header B	Header C
=====	=====	=====
A1	B1	C1
A2	B2	C2
=====	=====	=====

### 2.3.2 Result

Header A	Header B	Header C
A1	B1	C1
A2	B2	C2

## 2.4 Insert an Image

1. Place the **picture.jpg** file to the `source/_static` folder.
2. Add `.. image:: /_static/picture.jpg` in the rst file.

---

**Tip:** To limit the width of your image, add the **width** parameter and assign a value `:width: 400`.

---

---

**Tip:** If you need to insert an image multiple times, you can define it like `.. |Name| image:: /_static/picture.jpg` and apply it by using `|Name|` anywhere.

---

## 2.5 Add the Comment Function Using Disqus

---

**Note:** Make sure you've got VPN access because Disqus can only be visited via VPN .

---

### 2.5.1 Step 1 Install the Library

In a folder parallel with your Sphinx document directory, create a **requirements-doc.txt** file with the following content:

```
sphinx
sphinx_rtd_theme>=0.2.5b2
-e git://github.com/rmk135/sphinxcontrib-disqus.git#egg=sphinxcontrib-disqus
```

Run the `pip install -r requirements-doc.txt` command to install the library.

**Note:** After the installation, move the **requirements-doc.txt** file into your Sphinx document directory so that it can be sync'ed to GitHub when you commit and push changes.

## 2.5.2 Step 2 Configure your Sphinx Documents to Adapt to Disqus

In the **conf.py** file, add the following two lines.

```
extensions = ['sphinxcontrib.disqus']
disqus_shortname = 'blackmomo'      # ``disqus_shortname`` is defined in your
↳registered Disqus page.
```

Create a new rst file **comment.rst** with the following script provided by Disqus, and change the variable values.

```
.. _comment.rst:

.. raw:: html

<div id="disqus_thread"></div>
<script>
/**
 * RECOMMENDED CONFIGURATION VARIABLES: EDIT AND UNCOMMENT THE SECTION BELOW TO
↳INSERT DYNAMIC VALUES FROM YOUR PLATFORM OR CMS.
 * LEARN WHY DEFINING THESE VARIABLES IS IMPORTANT: https://disqus.com/admin/
↳universalcode/#configuration-variables
 */
/*
var disqus_config = function () {
this.page.url = PAGE_URL; // Replace PAGE_URL with your page's canonical URL
↳variable
this.page.identifier = PAGE_IDENTIFIER; // Replace PAGE_IDENTIFIER with your page
↳'s unique identifier variable
};
*/

var disqus_shortname = 'blackmomo';           // required: replace
↳it with your Disqus shortname
(function() { // DON'T EDIT BELOW THIS LINE
var d = document, s = d.createElement('script');

s.src = '//blackmomo.disqus.com/embed.js';     // required: replace
↳it with your Disqus site name

s.setAttribute('data-timestamp', +new Date());
(d.head || d.body).appendChild(s);
})();
</script>
<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref_
↳noscript" rel="nofollow">comments powered by Disqus.</a></noscript>
```

Add the following line to the topics where the comment function needs to appear.

```
.. include:: comment.rst
```

Build your Sphinx documents locally and view the results.

---

**Note:** For security reasons, Javascript doesn't load from `file://`, only `http://` or `https://`. A simple workaround to view your HTML files through a simple web server after building your HTML files is:

1. `cd build/html`
  2. `python -m http.server 8080`
  3. Browse to <http://localhost:8080/>
- 

### 2.5.3 Step 3 Configure Read the Docs

In the **Admin > Advanced Settings** page, enter **requirements-doc.txt** in the **Requirements file** field.

*Reference*

- <https://robpol86.github.io/sphinxcontrib-disqus/>
- <https://github.com/Robpol86/sphinxcontrib-disqus/pull/7>
- <https://zhu45.org/xxks-kkk.github.io.source/kb/sphinx/disqus.html>

## 3.1 Multi-version

1. Create a new branch for your project on Github.
2. On the Versions tab of RTD, activate the new version (syncd from Github).
3. Build the new version.

## 3.2 Multi-language for Localization

### 3.2.1 Step 1 Create Translatable Files

Run `$ sphinx-build -b gettext . _build/gettext` in the **source/** directory to create **pot** files.

### 3.2.2 Step 2 Translate Text from Source Language

1. Run `$ pip install-intl` to install the **sphinx-intl** tool.
2. Run `$ sphinx-intl update -p _build/gettext -l ja_JP` to generate a directory structure like below:

```
locale
├── ja_JP
│   └── LC_MESSAGES
│       └── index.po
```

Then open those **.po** files with a text editor and translate the content in the **msgstr** argument.

### 3.2.3 Step 3 Build the Documentation in Target Language

Run `$ sphinx-build -b html -D language=ja_JP . _build/html/ja_JP` to build the documentation in Japanese.

#### *Reference*

- <https://docs.readthedocs.io/en/stable/guides/manage-translations.html>
- <https://docs.readthedocs.io/en/stable/localization.html>
- <https://www.icanlocalize.com/site/tutorials/how-to-translate-with-gettext-po-and-pot-files/>
- <https://www.drupal.org/node/1814954>



### 4.1 Commit Changes to A New Branch

For example, the current branch I'm working on is named **master**, and the other branch is named **stable**.

1. Run `$ git checkout -b stable` to switch to the new branch.
2. Edit files and save.
3. Run `$ git commit <file> -m "sync with master"` to commit a single change to your local repo.
4. Run `$ git push origin stable` to push the change to the new branch on the Github.

### 4.2 Merge Changes into the Master Branch

The current branch I'm working on is named **stable**.

1. Run `$ git checkout master` to switch to the master branch.
2. Run `$ git merge stable`.



## CHAPTER 5

---

### Cooking

---

### 5.1 Breakfast

#### 5.1.1 Day 1

- Fried Egg
- Fried Dumpling
- Coffee



### 5.1.2 Day 2

- Hand-grasping Cake + Pork Luncheon Meat + Scrambled Egg
- Coffee



### 5.1.3 Day 3

- Sumai
- Steamed Bun
- Coffee

## 5.2 Lunch/Dinner

### 5.2.1 Meal 1

- Rice
- Cucumber + Shrimp
- Green Pepper + Egg



### 5.2.2 Meal 2

- Rice
- Fish Fillet
- Spinach Soup





### 5.2.3 Meal 3

- 
- 
- 

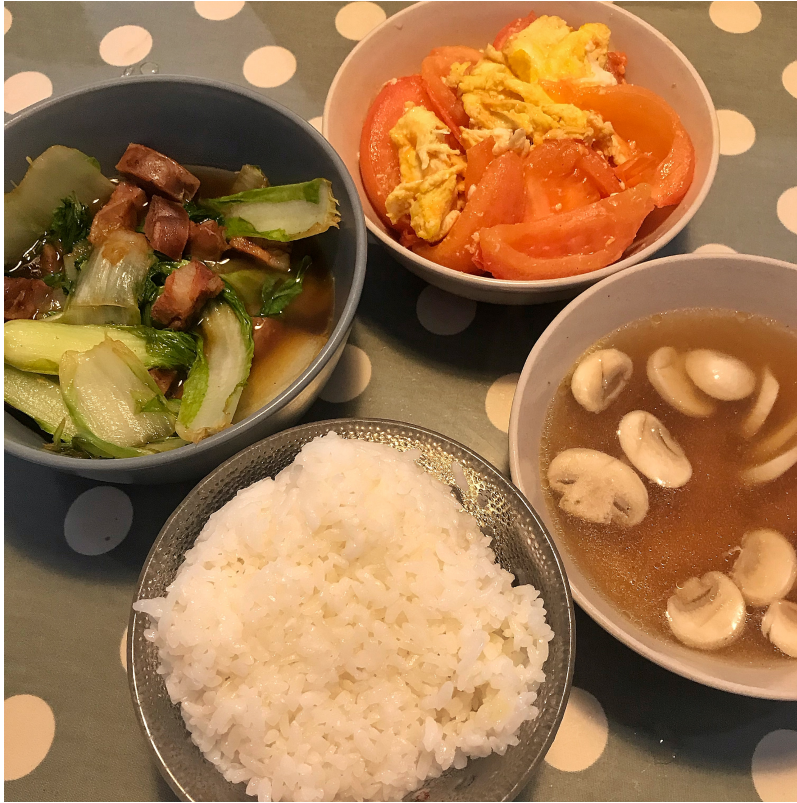
### 5.2.4 Meal 4

- Rice
- Sausage + Dried Tofu
- Mushroom + Shrimp
- Soup



### 5.2.5 Meal 5

- Rice
- Tomato + Eggs
- Cabbage + Sausage
- Mushroom Soup



### 5.2.6 Meal 6

- Noodles
- Green Vegetables
- Pork Luncheon Meat
- Egg





### 5.2.7 Meal 7

- Noodles
- Tomato + Cabbage
- Pork Luncheon Meat
- Egg



## M

menu, 18

## W

worklog1, 2

worklog2, 6

worklog3, 8