

# reST を書かずに autodoc だけで Sphinx する

---

- @satamame (佐田和也)
- 株式会社リーディング・エッジ社

# やりたかったこと

---

- docstring からドキュメントを生成する。  
※ docstring は [NumPy 形式](#)または [Google 形式](#)とします。
- よく見かける安心感のある体裁 ([Read the Docs](#)) にする。
- GitHub Pages で公開する。
- バージョンごとのドキュメントを作る。
- [reStructuredText \(reST\)](#) を手で書かない。

# サンプル

🏠 playscript

Search docs

CONTENTS:

playscript package

📄 Other Versions

v: master ▼

🏠 » playscript - 台本データを扱うパッケージ

View page source

## playscript - 台本データを扱うパッケージ

主に日本語で書かれた台本を扱います。

### Contents:

- [playscript package](#)
  - Subpackages
  - Submodules
  - playscript.psc module
  - Module contents

### Indices and tables

- [索引](#)
- [モジュール索引](#)
- [検索ページ](#)

Next ➞

© Copyright 2021, satamame.

Built with [Sphinx](#) using a [theme](#) provided by [Read the Docs](#).

# 動作確認した環境

---

- Windows 10 Home 2004
- Python 3.8.1
- Sphinx 3.5.2
- sphinx-rtd-theme 0.5.1
- sphinx-multiversion 0.2.4

# 構成

---

ドキュメントは (中間生成物も) すべて "docs" フォルダに入るようにします。

```
my_pkg/  
├─ .git/  
├─ .venv/  
├─ docs/  
│   ├─ build/  
│   │   └─ (生成した HTML)  
│   └─ source/  
│       └─ (reST、設定やテンプレート)  
└─ src/  
    └─ my_pkg/
```

# おおまかな流れ

## 1. 準備 (最初に一回だけやる)

1. パッケージのインストール
2. リポジトリ内に Sphinx プロジェクトを作成
3. `conf.py` を編集
4. `index.rst` を編集
5. リダイレクト用の `index.html` を作成
6. バージョン切替え用のテンプレートを作成

# おおまかな流れ

## 2. 生成 (ドキュメント更新時にやる)

1. ドキュメントの対象となるバージョンを準備する
2. ソースコードから reST を生成する
3. コミットしてタグをつける
4. バージョンごとの HTML を生成する

# パッケージのインストール

---

Read the Docs で [Sphinx のバージョン](#)を固定することが推奨されているので  
仮想環境にインストールします。

```
(.venv) > pip install sphinx sphinx-rtd-theme sphinx-multiversion
```

- ["sphinx-rtd-theme"](#) : Read the Docs のテーマを使うため。
- ["sphinx-multiversion"](#) : バージョンごとのドキュメントを作るため。

今回は両方ともインストールしている前提で進めます。



# リポジトリ内に Sphinx プロジェクトを作成

---

今回は Sphinx 関連のファイルはすべて "docs" フォルダの中に作ります。

```
(.venv) > sphinx-quickstart docs
```

ビルドのソースを "source" サブフォルダに入れるか聞かれるので "y"。

```
> Separate source and build directories (y/n) [n]: y
```

プロジェクト名、著者名、リリース番号、言語を聞かれます。  
これらは後から "conf.py" というファイルを編集して変更できます。

```
> Project name: my_pkg  
> Author name(s): satamame  
> Project release []: 0.0.1  
> Project language [en]: ja
```

答え終わると、プロジェクトのためのファイルが生成されます。

(関係ないファイルやフォルダは省略しています。)

```
my_pkg/
├─ docs/
│   ├── build/
│   │   └─ (HTML の生成先。今は空)
│   ├── source/
│   │   ├── _static/
│   │   ├── _templates/
│   │   ├── conf.py
│   │   └─ index.rst
│   ├── make.bat
│   └─ Makefile
└─ src/
    └─ my_pkg/
```

# conf.py を編集

---

Sphinx や拡張用モジュールの設定は "source/conf.py" の中に書きます。

## 1. パスの指定

autodoc でパッケージのソースコードを読み込むためにパスを追加します。

```
# -- Path setup -----  
...  
  
import os      # コメント解除  
import sys     # コメント解除  
sys.path.insert(0, os.path.abspath('../..src')) # 変更
```

## 2. 拡張用モジュールの指定

```
# -- General configuration -----  
...  
  
extensions = [  
    'sphinx.ext.autodoc',      # ソースコード読み込み用  
    'sphinx.ext.napoleon',    # docstring パース用  
    'sphinx_rtd_theme',       # Read the Docs (不要 *1)  
    'sphinx_multiversion',    # マルチバージョン用  
]
```

`autodoc` と `napoleon` は Sphinx に付属していますので、別途インストールの必要はありません。

(\*1)

Read the Docs Sphinx Theme のインストール手順に従って  
"sphinx\_rtd\_theme" を拡張用モジュールとして追加していますが、  
これは**不要**です。

追加するのはローカライズの機能を取り込むためです。  
しかしこの機能が現状で日本語に対応していないので意味がないのです。  
(言語をデフォルトの "en" にした場合も不要。)

また、Read the Docs Sphinx Theme のインストール手順では  
"conf.py" の中で `import sphinx_rtd_theme` をするとありますが、  
これも不要かと思います。

### 3. テーマの指定

```
# -- Options for HTML output -----  
...  
  
html_theme = 'sphinx_rtd_theme' # 変更
```

このテーマに固有の設定も、"conf.py" に追加することで可能です。

<https://sphinx-rtd-theme.readthedocs.io/en/latest/configuring.html>

## 4. マルチバージョンの設定

個々の設定については [sphinx-multiversion のサイト](#) に仕様があります。

ここでは、「Git の履歴から "1.0" や "12.34" のようなタグと "master" ブランチをドキュメントのバージョンとして取り出す」という設定をします。

```
# -- Options for sphinx-multiversion -----  
  
smv_tag_whitelist = r'^\d+\.\d+$'      # マッチしたタグを抽出  
smv_branch_whitelist = r'^master$'     # マッチしたブランチを抽出
```



ここに表示され、選択できるようになる。



The screenshot shows the documentation for the 'playscript' package. The left sidebar has a blue header with the 'playscript' logo and a search bar. Below the search bar, the 'CONTENTS:' section lists 'playscript package'. Further down, the 'Other Versions' section shows 'v: 0.2' with a dropdown arrow. The 'Tags' section lists '0.2', 'Branches', and 'master'. A large red arrow points from the '0.2' tag to the main content area. The main content area has a breadcrumb trail '» playscript - 台本データを扱うパッケージ', a title 'playscript - 台本データを扱うパッケージ', a description '主に日本語で書かれた台本を扱います。', and a 'Contents:' section with a list of links: 'playscript package', 'Subpackages', 'Submodules', 'playscript.psc module', and 'Module contents'.

🏠 playscript

Search docs

**CONTENTS:**

playscript package

📖 Other Versions v: 0.2 ▼

Tags

0.2

Branches

master

🏠 » playscript - 台本データを扱うパッケージ

## playscript - 台本データを扱うパッケージ

主に日本語で書かれた台本を扱います。

### Contents:

- [playscript package](#)
  - Subpackages
  - Submodules
  - [playscript.psc module](#)
  - [Module contents](#)

# index.rst を編集

---

"index.rst" から、目次に表示したいパッケージ (の reST) を参照します。

```
Welcome to my_pkg's documentation!  
=====
```

```
.. toctree::  
    :maxdepth: 2  
    :caption: Contents:  
  
    my_pkg      (これを追加)
```

# リダイレクト用の index.html を作成

---

```
<!DOCTYPE html>
<html>
  <head>
    <title>Redirecting to master branch</title>
    <meta charset="utf-8">
    <meta http-equiv="refresh" content="0;
      url=./master/index.html">
    <link rel="canonical"
      href="https://my_pkg.com/master/index.html">
  </head>
</html>
```

```
my_pkg/
└─ docs/
    ├─ build/
    │   │   └─ index.html  <-- これを作って
    │   │   └─ 0.2/
    │   │       └─ index.html
    │   │       └─ (index 以外のページ, CSS)
    │   └─ master/
    │       └─ index.html  <-- ここへリダイレクト
    │       └─ (index 以外のページ, CSS)
    └─ source/
    └─ make.bat
    └─ Makefile
```

※バージョンごとのフォルダ ("0.2", "master") はあとで生成されます。

# バージョン切替え用のテンプレートを 作成

---

バージョン切替えの UI を表示するために、"docs/source/\_templates" フォルダの下に "versions.html" というファイルを作ります。内容は sphinx-multiversion のサイトからコピーしてきます。

<https://holzhaus.github.io/sphinx-multiversion/master/templates.html#readthedocs-theme>

※このやり方は Read the Docs テーマを使っている場合の特別なやり方です。一般的には、"conf.py" の中で html\_sidebars にテンプレートを追加するようですが、Read the Docs テーマを使う場合はその方法では表示できないようです。

# ドキュメントの対象となるバージョンを準備する

---

ここからは、バージョンごとのドキュメントを作成する際に毎回実施する作業となります。

まず、「Git 上のタグを見てドキュメントを作るのだ」という事を理解する必要があります。

今回の設定では、"1.0" などのタグをつけたコミットはドキュメント生成の対象になりますから、混乱しないように**パッケージ自体がバージョン 1.0 相当になった時点で以降の作業**するようにします。

# ソースコードから reST を生成する

---

パッケージが準備できて「これでリリースする」と決めたら、**ソースコードからドキュメントのソースを生成**します。

```
(.venv) > sphinx-apidoc -f -o docs/source src
```

このコマンドのリファレンスは Sphinx のサイトにあります。

<https://www.sphinx-doc.org/ja/master/man/sphinx-apidoc.html>

※**注意**：ここで参照したパッケージは、ドキュメントのビルド時に **import** (実行) されます。

```
my_pkg/
├── docs/
│   ├── build/
│   │   └── index.html
│   ├── source/
│   │   ├── _static/
│   │   ├── _templates/
│   │   ├── conf.py
│   │   ├── index.rst
│   │   ├── modules.rst <-- 生成される
│   │   └── my_pkg.rst <-- 生成される
│   ├── make.bat
│   └── Makefile
└── src/
```



# コミットしてタグをつける

---

sphinx-multiversion が **Git** の履歴からドキュメントのソースを取ってくるので、HTML を生成する前にコミットしてタグをつける必要があります。

```
> git add .  
> git commit -m 'v0.2 ドキュメントソース'  
> git tag 0.2
```

`smv_tag_whitelist` にマッチするタグ名をつけます。

逆にマッチしないタグ名 ("1.0.5" など) をつけば、ドキュメントのバージョンは更新せずにパッチ等のリリース用のタグにできます。

# バージョンごとの HTML を生成する

---

`sphinx-multiversion` というコマンドを使います。

```
(.venv) > sphinx-multiversion docs/source docs/build
```

実行すると、"docs/build" 以下に HTML が生成されます。

"docs/build/index.html" を開くと (リダイレクトされて) master ブランチから生成したドキュメントが表示されます。

そこからサイドバーを使って他のバージョンに切替えられます。

# GitHub Pages にプッシュする

---

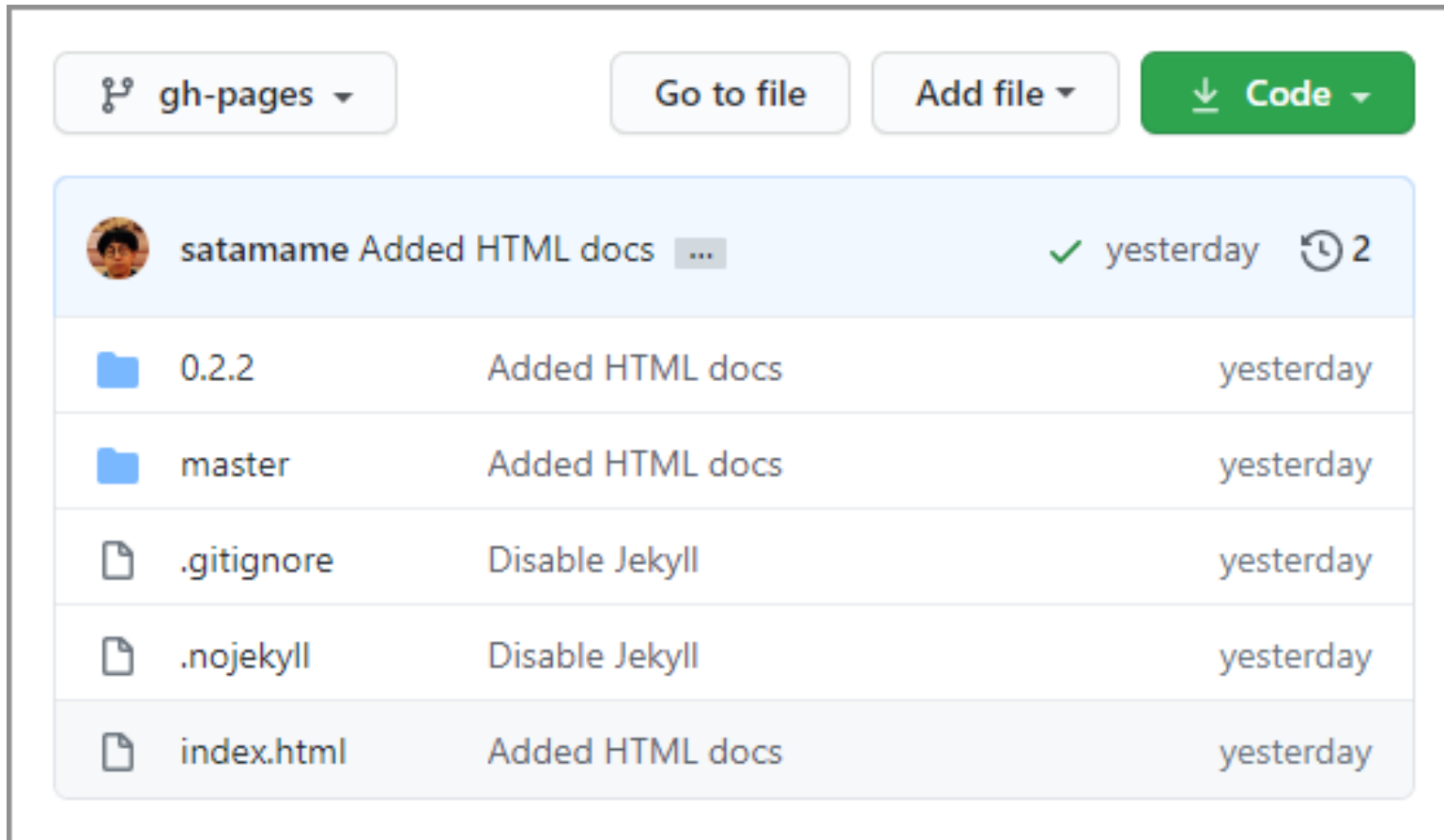
sphinx-multiversion のサイトに手順が載っています。

[https://holzhaus.github.io/sphinx-multiversion/master/github\\_pages.html](https://holzhaus.github.io/sphinx-multiversion/master/github_pages.html)

ちょっと手順は違いますが、こんな感じです。

1. gh-pages ブランチを --orphan で作っておく。
2. ".nojekyll" というファイルを作っておく。
3. master ブランチで HTML を生成し、stash する。
4. gh-pages ブランチに切り替え、さっきの stash を apply する。
5. "docs/build" の中身をルートに移動。
  - ソースコード等は削除して良い。
6. gh-pages を commit して push する。

参考 : gh-pages ブランチの中身の例



The screenshot shows the GitHub interface for the 'gh-pages' branch. At the top, there are buttons for 'Go to file', 'Add file', and 'Code'. Below this, a commit by 'satamame' is highlighted, showing the message 'Added HTML docs' and a timestamp of 'yesterday'. A table below the commit lists the files added in this commit:

File	Commit Message	Time
0.2.2	Added HTML docs	yesterday
master	Added HTML docs	yesterday
.gitignore	Disable Jekyll	yesterday
.nojekyll	Disable Jekyll	yesterday
index.html	Added HTML docs	yesterday

※ "https://username.github.io/reponame/" のような URL で公開される。

# 参考文献

---

※リンク先の情報を保証するものではありません。

- [Sphinx documentation](#)
- [sphinx.ext.autodoc -- docstringからのドキュメントの取り込み](#)
- [sphinx-multiversion 0.2.4 documentation](#)
- [Getting Started with Sphinx — Read the Docs](#)
- [Read the Docs Sphinx Theme 0.5.1 documentation](#)
- [Sphinxの使い方. docstringを読み込んで仕様書を生成 - Qiita](#)
- [PythonプロジェクトのドキュメントをSphinxで作成する - Qiita](#)