

Sphinxプロジェクトを始める

プロジェクトのディレクトリを作成する

`sphinx-quickstart` コマンドを使うことで、簡単にプロジェクトの雛形を作ることができる。

```
$ mkdir sphinx-sample && sphinx-sample
$ sphinx-quickstart
Sphinx 4.1.2 クイックスタートユーティリティへようこそ。
```

以下の設定値を入力してください (Enter キーのみ押した場合、
かっこで囲まれた値をデフォルト値として受け入れます)。

選択されたルートパス: .

Sphinx 出力用のビルドディレクトリを配置する方法は2つあります。
ルートパス内にある "_build" ディレクトリを使うか、
ルートパス内に "source" と "build" ディレクトリを分ける方法です。
> ソースディレクトリとビルドディレクトリを分ける (y / n) [n]: y

プロジェクト名は、ビルドされたドキュメントのいくつかの場所にあります。
> プロジェクト名: test_sphinx
> 著者名 (複数可): masataka
> プロジェクトのリリース [:]: 1.0.0

ドキュメントを英語以外の言語で書く場合は、
言語コードで言語を選択できます。Sphinx は生成したテキストをその言語に翻訳します。

サポートされているコードのリストについては、
<https://www.sphinx-doc.org/en/master/usage/configuration.html#confval-language> を参照してください。
> プロジェクトの言語 [en]: ja

ファイル /Users/masataka/Junk/test_sphinx/source/conf.py を作成しています。
ファイル /Users/masataka/Junk/test_sphinx/source/index.rst を作成しています。
ファイル /Users/masataka/Junk/test_sphinx/Makefile を作成しています。
ファイル /Users/masataka/Junk/test_sphinx/make.bat を作成しています。

終了: 初期ディレクトリ構造が作成されました。

マスターファイル /Users/masataka/Junk/test_sphinx/source/index.rst を作成して
他のドキュメントソースファイルを作成します。次のように Makefile を使ってドキュメントを作成します。
make builder
"builder" はサポートされているビルダーの 1 つです。例: html, latex, または linkcheck。

以下のようなディレクトリが作られている

```
.
├── Makefile
├── build
├── make.bat
├── source
│   ├── _static
│   ├── _templates
│   ├── conf.py
│   └── index.rst
```

4 directories, 4 files

文章のビルド

htmlの作成

プロジェクトのホームディレクトリ以下 (**Makefile**が置いてあるディレクトリ)

```
$ make html
```

とすると、**build/html**にHTMLページが作成される。

pdfの作成

プロジェクトのホームディレクトリ以下で

```
$ make latexpdf
```

を実行すると、**build/latex**以下にpdfが作成される。

テーマの変更

テーマをpipでインストールして、**source/conf.py**を書き換える

```
$ pip install sphinx-rtd-theme
```

source/conf.pyを以下のように変更する。

```
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.viewcode',
    'sphinx.ext.todo',
    'sphinx.ext.napoleon',
    'sphinx_rtd_theme',
```

```
    'myst_parser'  
]  
html_theme = 'sphinx_rtd_theme'
```

その後, htmlを生成する

```
$ make html
```

Markdownを使う

MyST-Parserを使う方法 (推奨)

SphinxプロジェクトでMarkdownをサポートするためには以下の手順にしたがって設定をする

1. Markddwonパーサーである**MyST-Parser**をインストールする

```
$ pip install --upgrade myst-parser
```

2. **source/conf.py**の**extensions**にMyST-Parserを加える

```
extensions = ['myst_parser']
```

3. **source/conf.py**に拡張子の設定をする

```
source_suffix = {  
    '.rst': 'restructuredtext',  
    '.txt': 'markdown',  
    '.md': 'markdown',  
}
```

その他の手段

Markdownで書いて, pandocで

```
pandoc -f markdown -t rst -o ../main.rst main.md
```

みたいにreStructuredTextに変換すればいける。

recommonmark, **commonmark**を使う方法 (現在はサポートされていない)

ネットで調べると `recommonmark` を使うことで, `Markdown` を使うことができるようになるがあるが、公式によると `CommonMark` はもうサポートされていないため

```
$ make htm
WARNING: 拡張機能のセットアップ中 recommonmark: 拡張 'recommonmark' には setup()
関数がありません。これは本当にSphinx拡張ですか？
```

のようなエラーが出力されて、うまくいかない。

参考文献

1. [Sphinx Documentation: Markdown](#)