

---

# Windows 10 で Sphinx を使う

リリース *2021/04/17 07:50* 更新

Yoichi Muraoka

2021 年 04 月 17 日



# Contents:

第 1 章	Sphinx とは	1
第 2 章	検証環境	3
2.1	Sphinx の動作環境を作る . . . . .	3
2.2	ドキュメントの作成手順 . . . . .	27
2.3	日本語対応にする (日本語拡張の導入) . . . . .	34
2.4	「テーマ」を変更する . . . . .	42
2.5	文書の構造 . . . . .	62
2.6	いろいろな表現 . . . . .	86
2.7	CSS (Cascading Style Sheets) . . . . .	116
2.8	付録 . . . . .	123
2.9	改版履歴 . . . . .	129



## 第 1 章

# Sphinx とは

- Python で作成されたオンラインドキュメント作成ツールです。
- マニュアルなどの記述に向いています。
- reStructuredText で記述したテキストファイルを HTML 形式に変換して公開します。
- LaTeX と組み合わせることで PDF 形式に変換も可能です。



## 第 2 章

# 検証環境

- Microsoft Windows 10 Pro Version 1909 (64-bit)
- Miniconda
- Sphinx 2.4.3 & Sphinx 3.0.1

### 2.1 Sphinx の動作環境を作る

Windows 上に Sphinx の実行環境を構築します。必要なアプリケーションです。

- Miniconda
- Sphinx
- テキストエディター

テキストエディターは UTF-8 が扱えれば、好みのものをご使用ください。

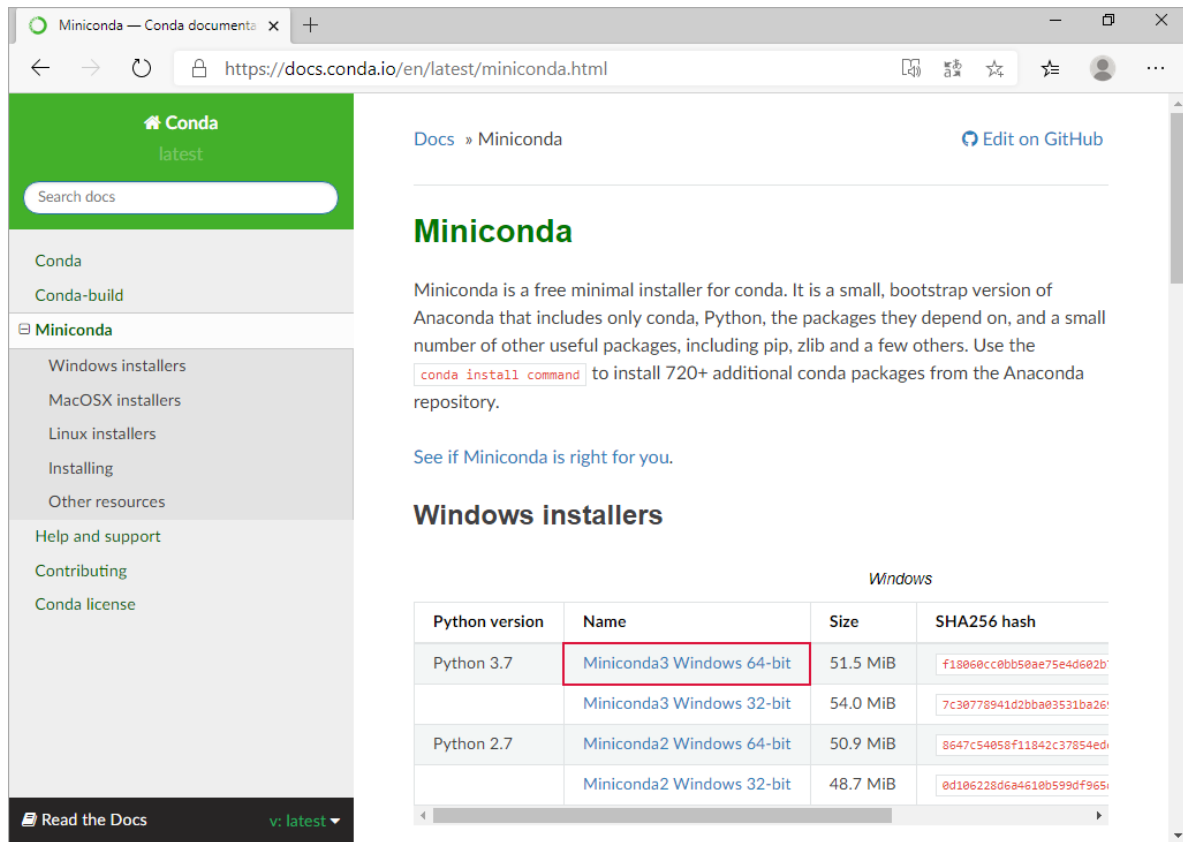
---

#### 【トピックス】

- *Miniconda* をインストールする
  - *Sphinx* をインストールする
  - テキストエディターをインストールする
-

## 2.1.1 Miniconda をインストールする

1. ブラウザーで **Miniconda** のページを開く → **Miniconda3 Windows 64-bit** をクリック



2. ファイルを開く をクリック



Miniconda — Conda documents

https://docs.conda.io/en/latest/miniconda.html

Conda latest

Search docs

Conda

Conda-build

Miniconda

Windows installers

MacOSX installers

Linux installers

Installing

Other resources

Help and support

Contributing

Conda license

Read the Docs v: latest

Miniconda3-latest-....exe

ファイルを開く

すべて表示

## Miniconda

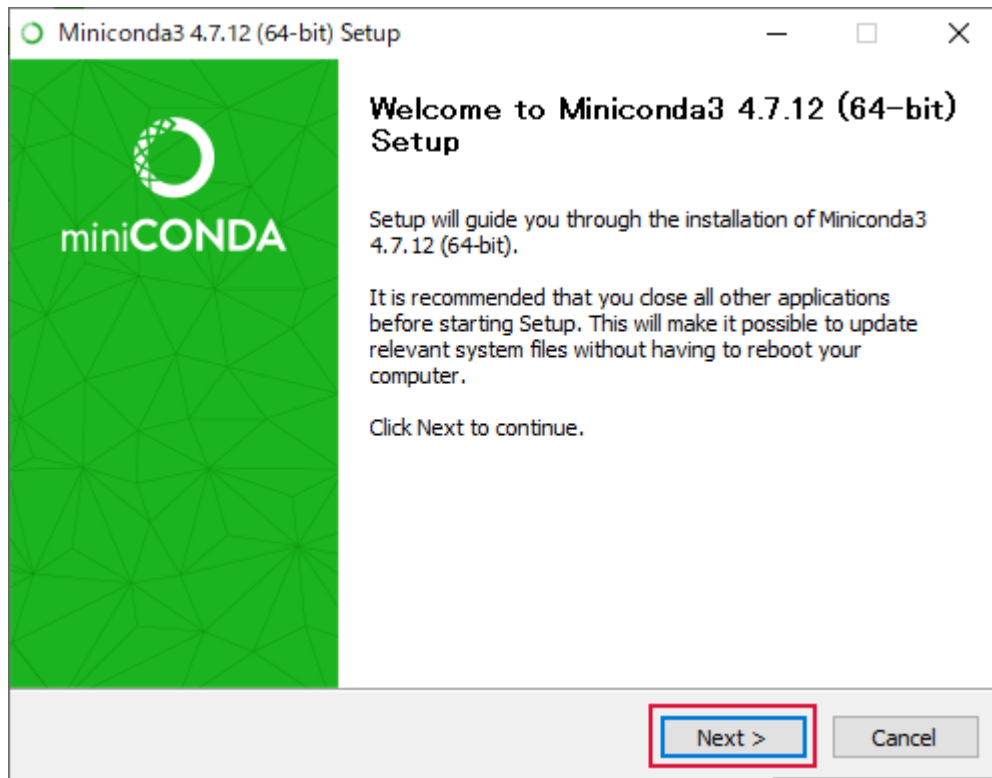
Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the `conda install command` to install 720+ additional conda packages from the Anaconda repository.

[See if Miniconda is right for you.](#)

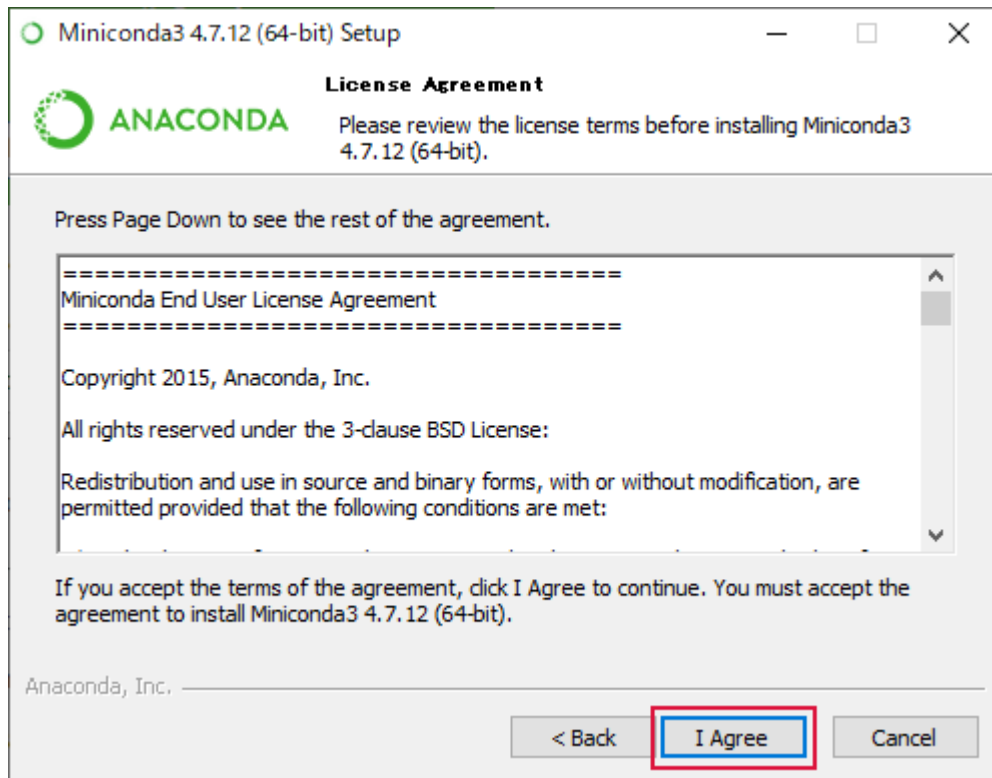
## Windows installers

Windows			
Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3 Windows 64-bit	51.5 MiB	f18060cc0bb50ae75e4d602b
	Miniconda3 Windows 32-bit	54.0 MiB	7c30778941d2bba03531ba26
Python 2.7	Miniconda2 Windows 64-bit	50.9 MiB	8647c54058f11842c37854ed
	Miniconda2 Windows 32-bit	48.7 MiB	0d106228d6a4610b599df965

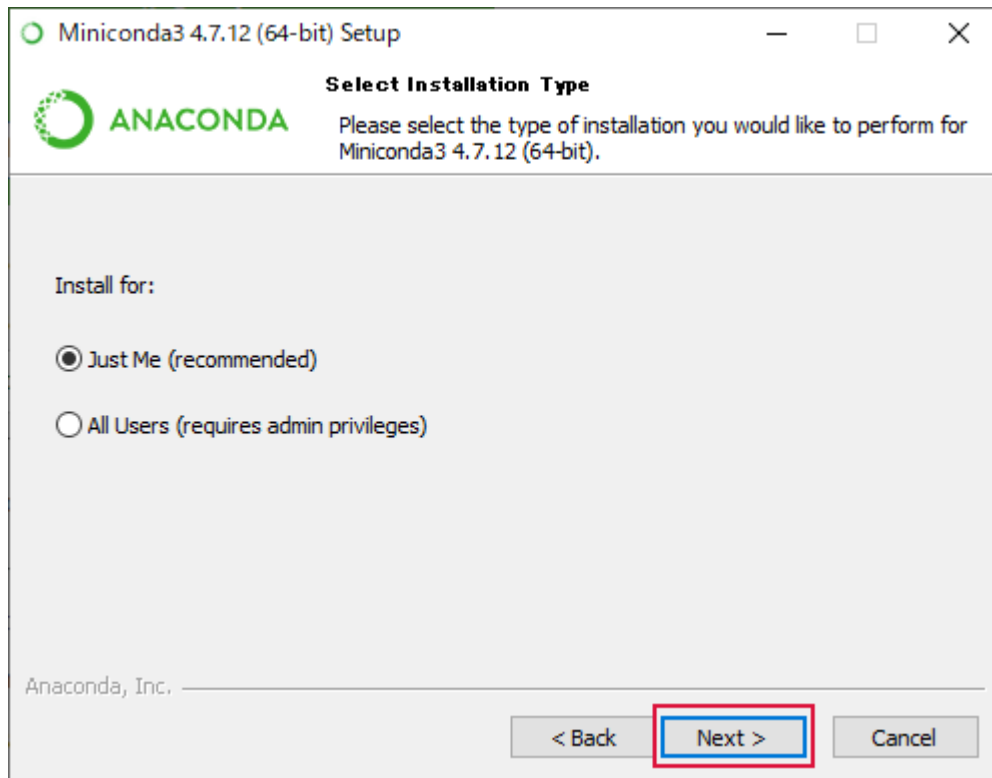
3. *Next* をクリック



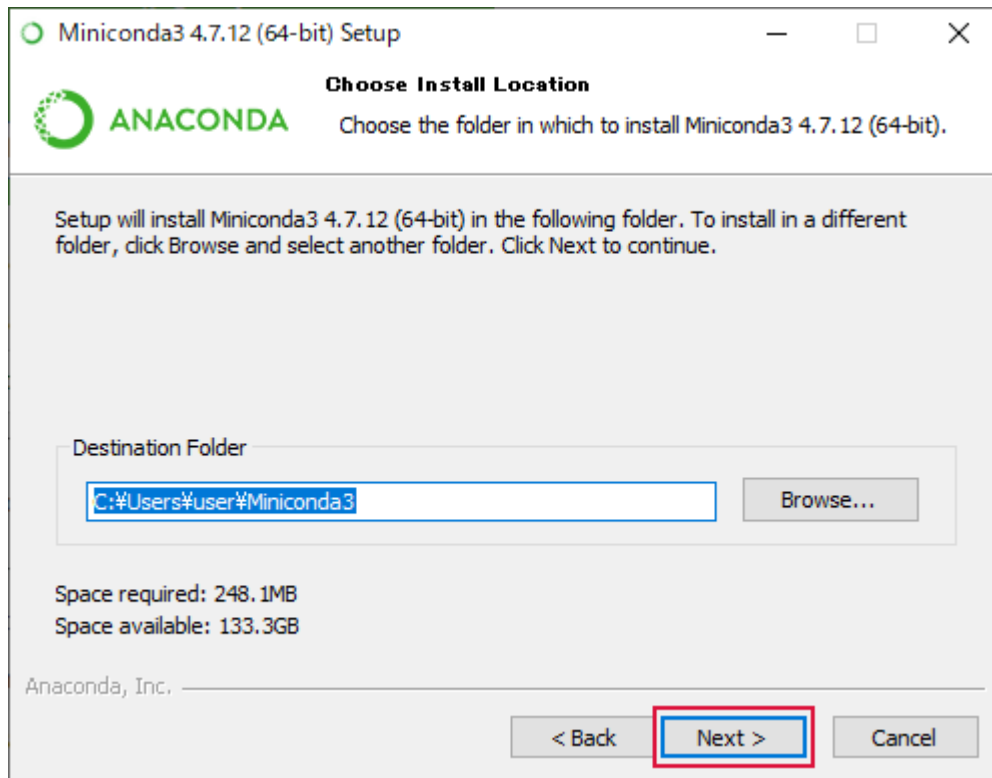
4. *I Agree* をクリック



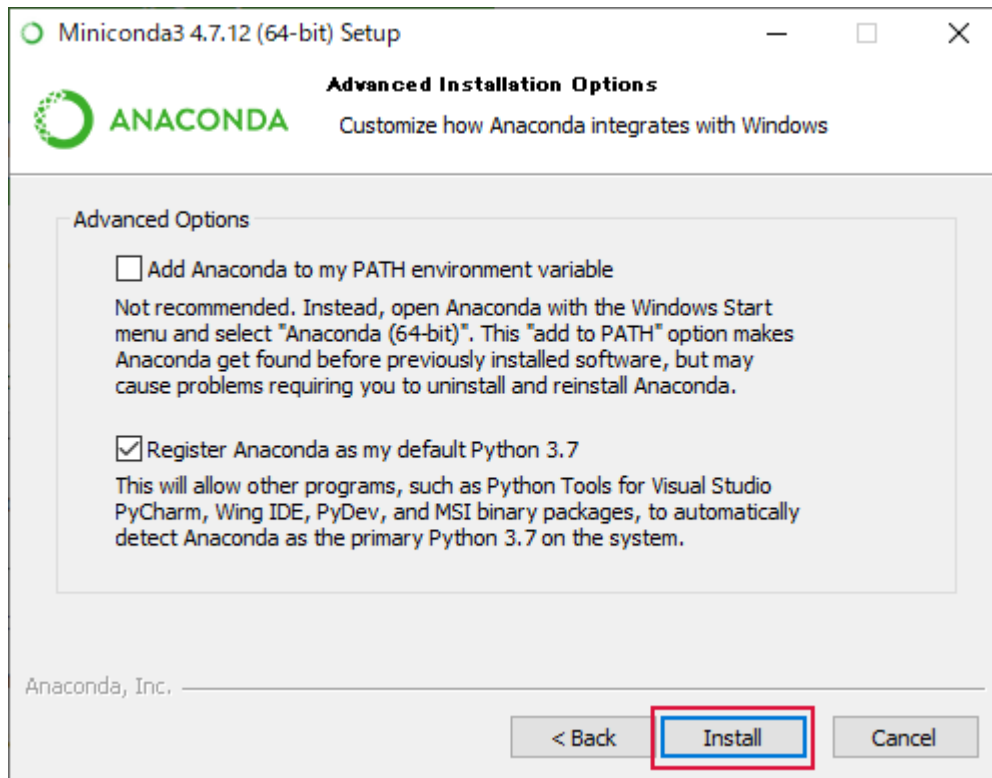
5. *Next* をクリック



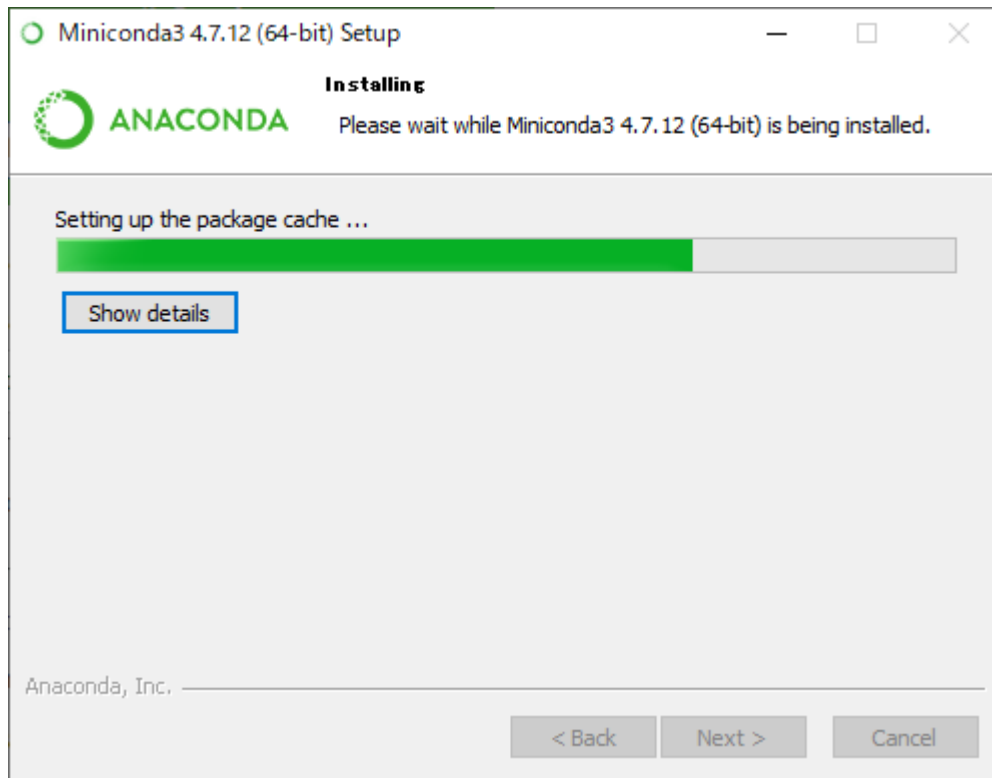
6. *Next* をクリック



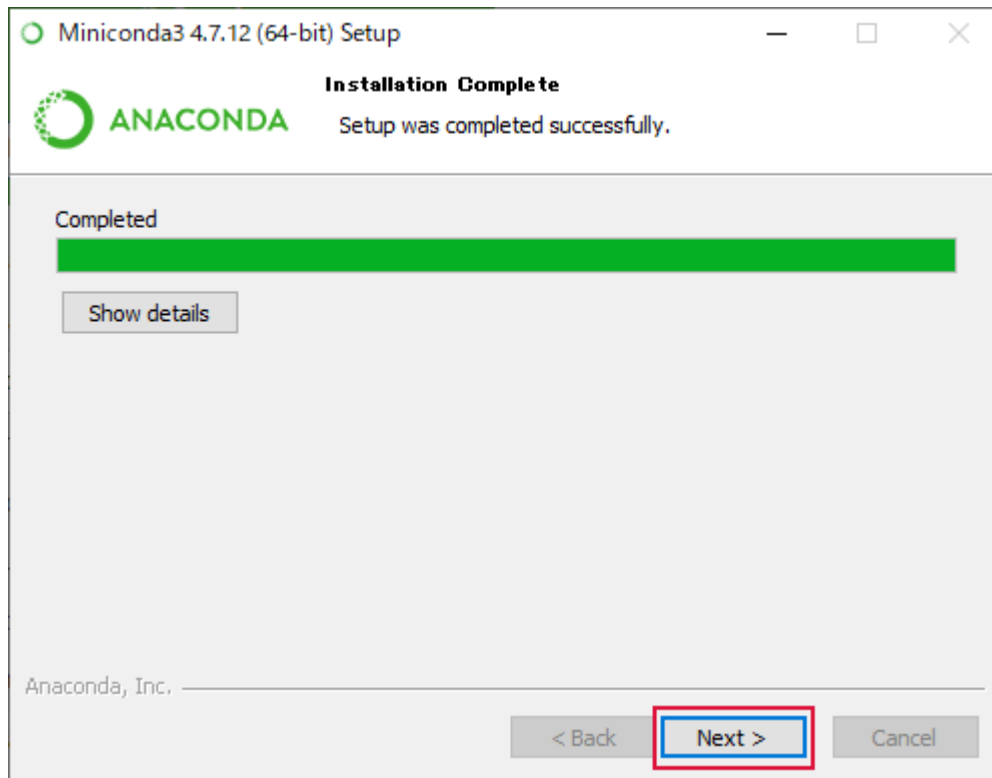
7. *Install* をクリック



## 8. インストール中

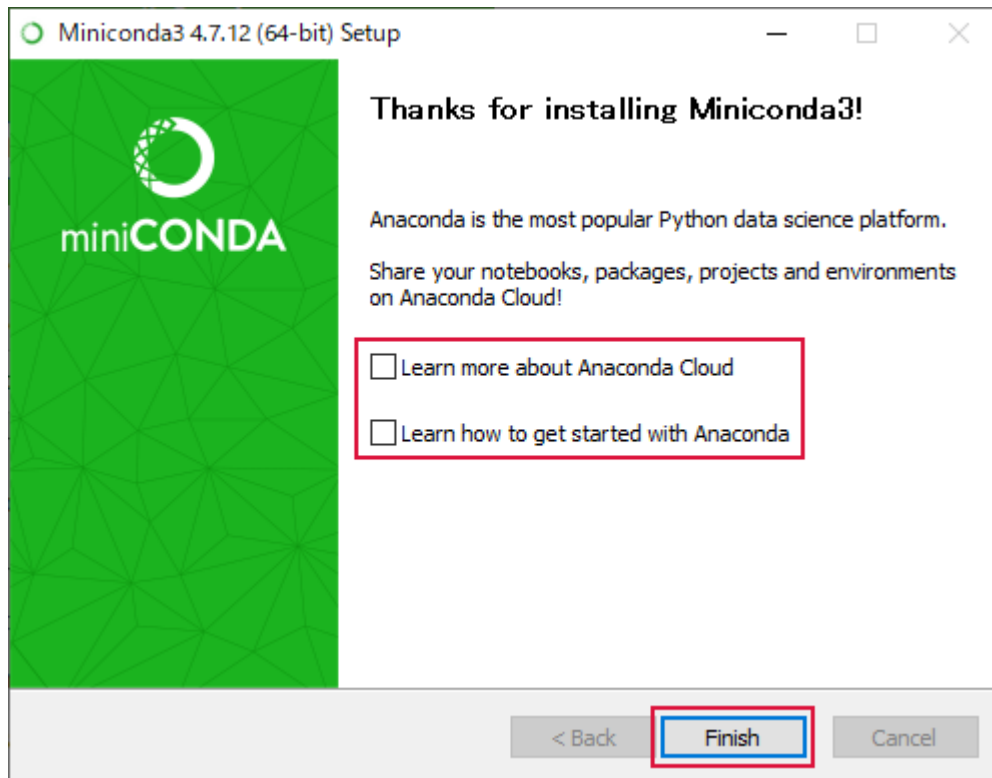


9. *Next* をクリック

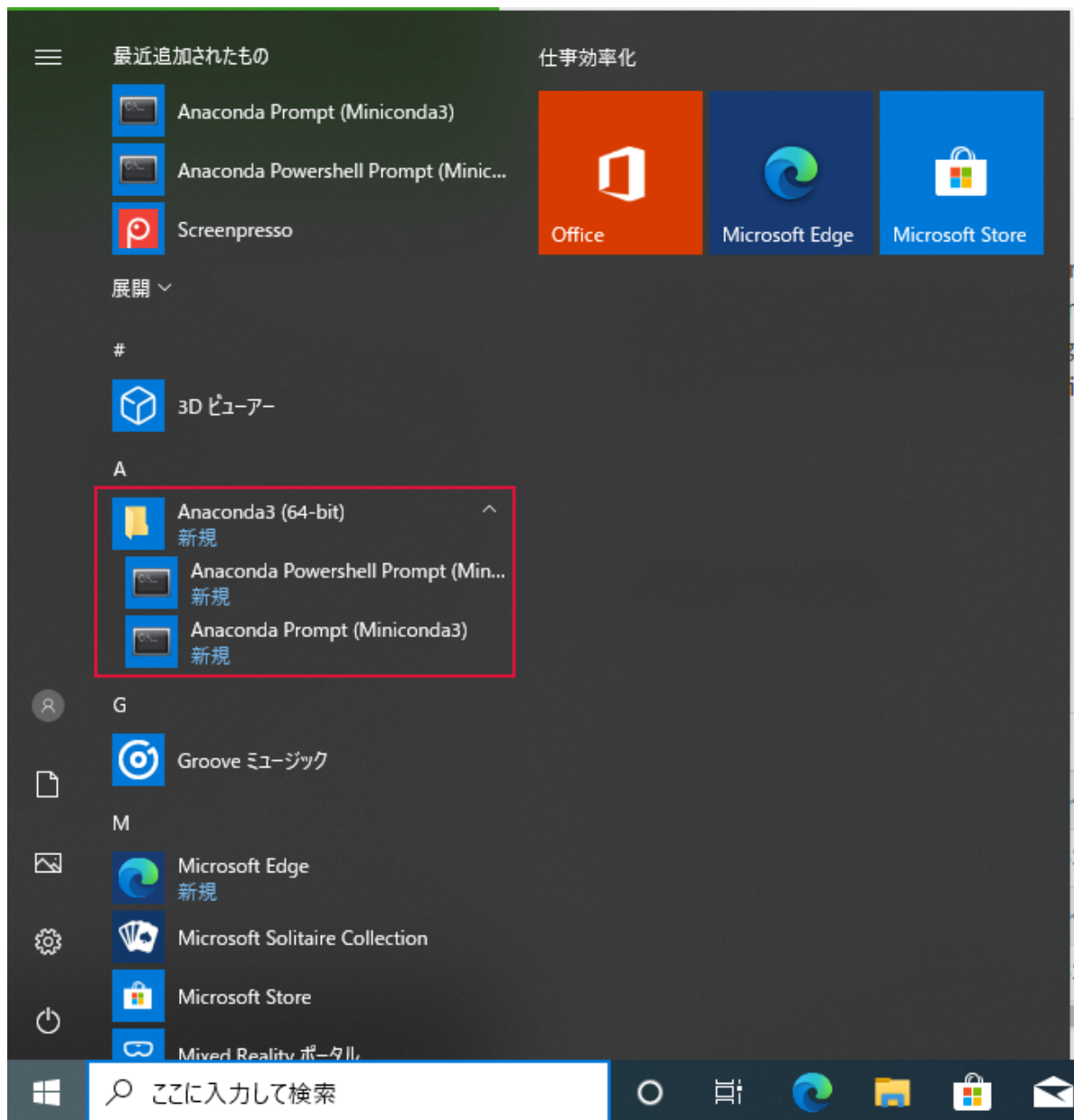


10. Learn more . . . と Learn how . . . のチェックを外す → *Finish* をクリック



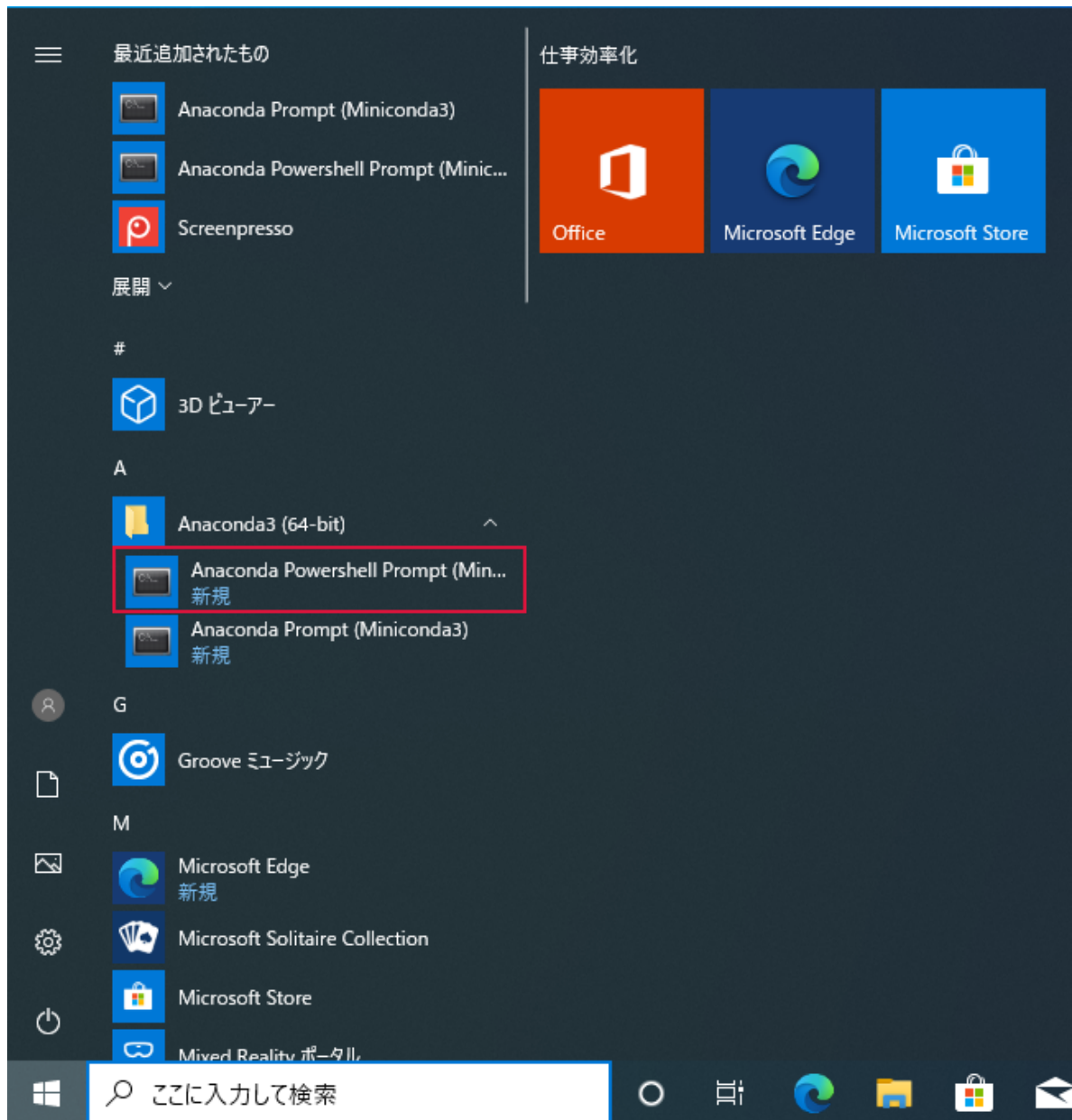


11. Miniconda のインストールが終了するとメニューに "Anaconda Powershell Prompt(Miniconda3)" と "Anaconda Prompt(Miniconda3)" が登録されます



## 2.1.2 Sphinx をインストールする

1. "Anaconda Powershell Prompt(Miniconda3)" を起動



2. Sphinx をインストール

```
pip install -U Sphinx
```

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\user> pip install -U Sphinx

Collecting pytz>=2015.7
  Downloading https://files.pythonhosted.org/packages/e7/f9/f0b53f88060247251bf4
-2019.3-py2.py3-none-any.whl (509kB)
    |#####| 512kB 6.4MB/s
Collecting MarkupSafe>=0.23
  Downloading https://files.pythonhosted.org/packages/65/c6/2399700d236d1dd681af
upSafe-1.1.1-cp37-cp37m-win_amd64.whl
Installing collected packages: colorama, sphinxcontrib-applehelp, sphinxcontrib-
contrib-qthelp, docutils, snowballstemmer, alabaster, imagesize, Pygments, pypar
, babel, MarkupSafe, Jinja2, sphinxcontrib-devhelp, Sphinx
Successfully installed Jinja2-2.11.1 MarkupSafe-1.1.1 Pygments-2.5.2 Sphinx-2.4.
docutils-0.16 imagesize-1.2.0 packaging-20.1 pyparsing-2.4.6 pytz-2019.3 snowba
sphinxcontrib-devhelp-1.0.2 sphinxcontrib-htmlhelp-1.0.3 sphinxcontrib-jsmath-1
-serializinghtml-1.1.4
(base) PS C:\Users\user>
```

### 3. Sphinx のバージョン確認

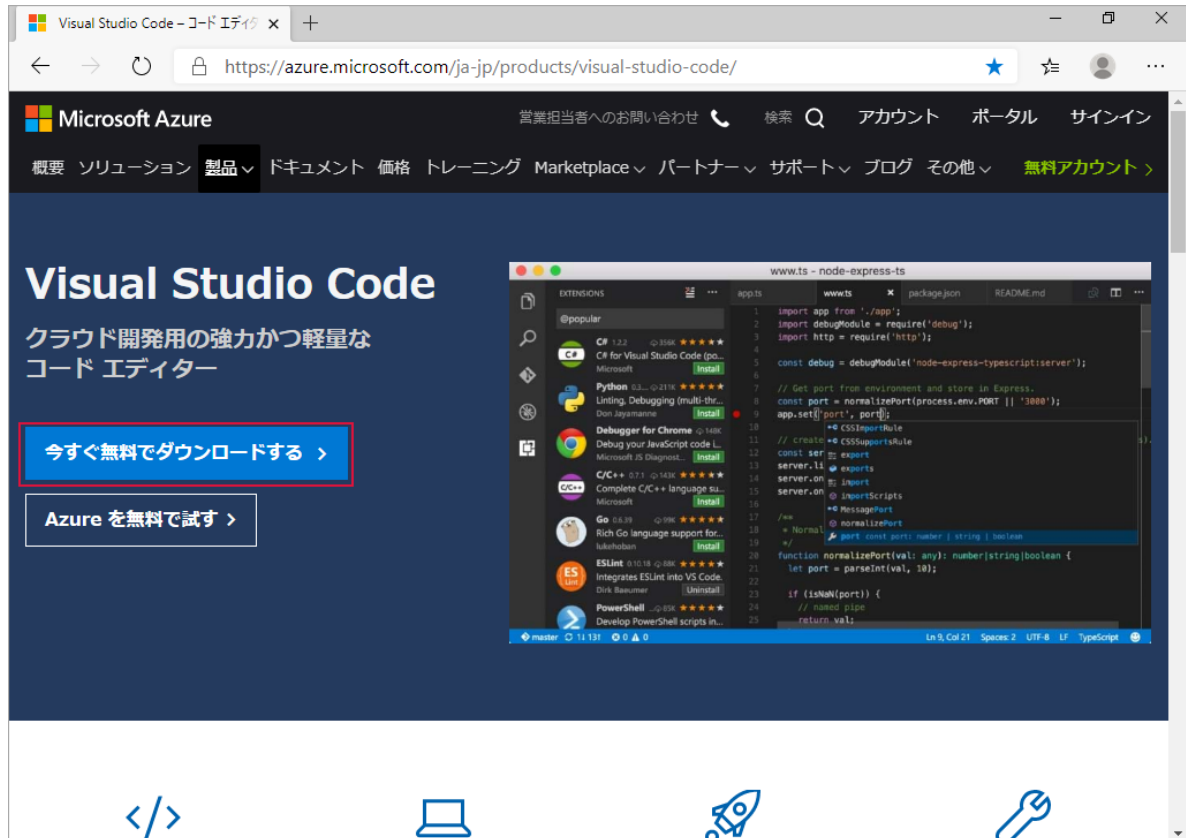
**pip show sphinx**

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\user> pip show sphinx
Name: Sphinx
Version: 2.4.3
Summary: Python documentation generator
Home-page: http://sphinx-doc.org/
Author: Georg Brandl
Author-email: georg@python.org
License: BSD
Location: c:\users\user\miniconda3\lib\site-packages
Requires: imagesize, sphinxcontrib-serializinghtml, sphinxcontrib-qthelp, Pygmen
o, snowballstemmer, alabaster, packaging, colorama, sphinxcontrib-applehelp, sph
sphinxcontrib-devhelp
Required-by:
(base) PS C:\Users\user>
```

### 2.1.3 テキストエディターをインストールする

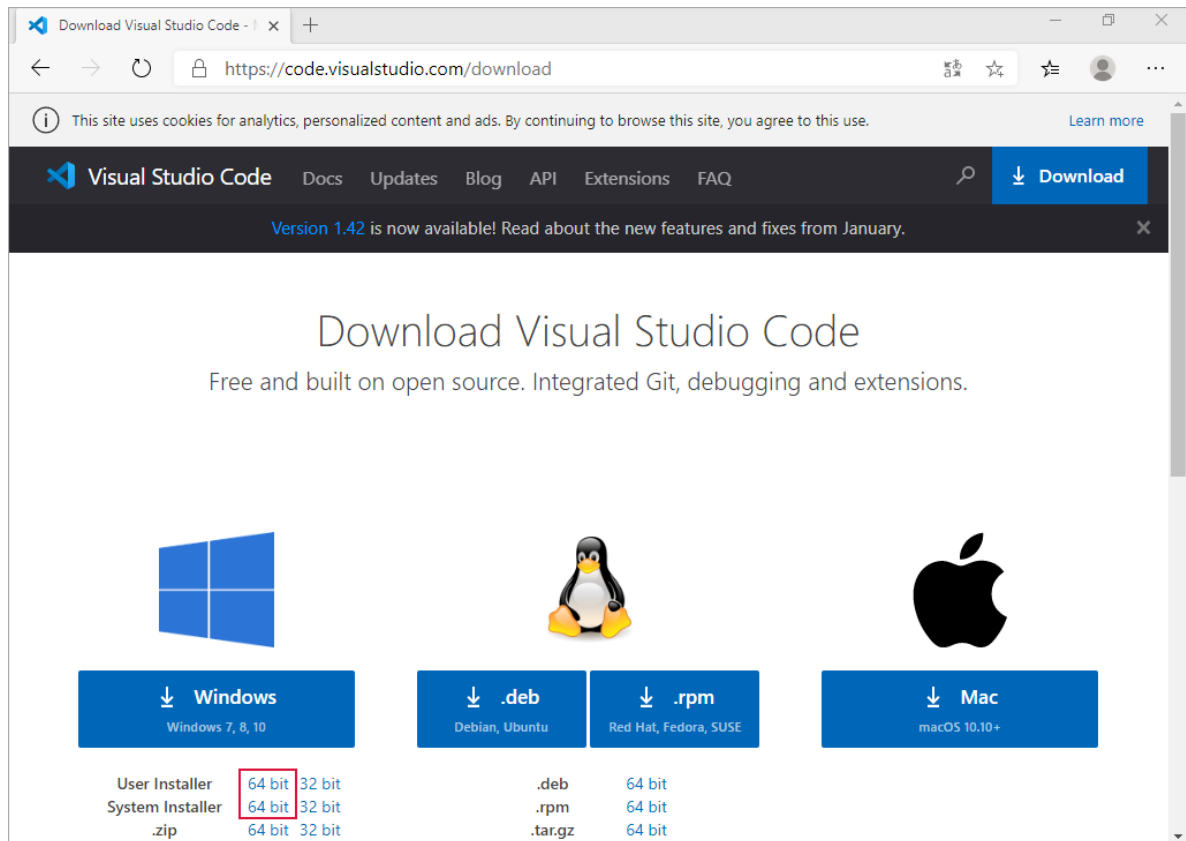
UTF-8 を扱えるテキストエディターであれば何でも良いです。ここでは Visual Studio Code を紹介します。

1. ブラウザーで [Visual Studio Code – コード エディター | Microsoft Azure](https://azure.microsoft.com/ja-jp/products/visual-studio-code/) のページを開く → 今すぐ無料でダウンロードする をクリック



2. Windows 用の 64 bit をクリック

この手順では System Installer を使用しています。



#### 注釈:

- ・インストールするパソコンを自分だけが使用しているのであれば User Installer / System Installer のどちらでも変わりはないです。
- ・インストールするパソコンを他の方と共同で使用しているのであれば使用環境（条件）によりませんが User Installer が望ましいと思います。

#### User Installer

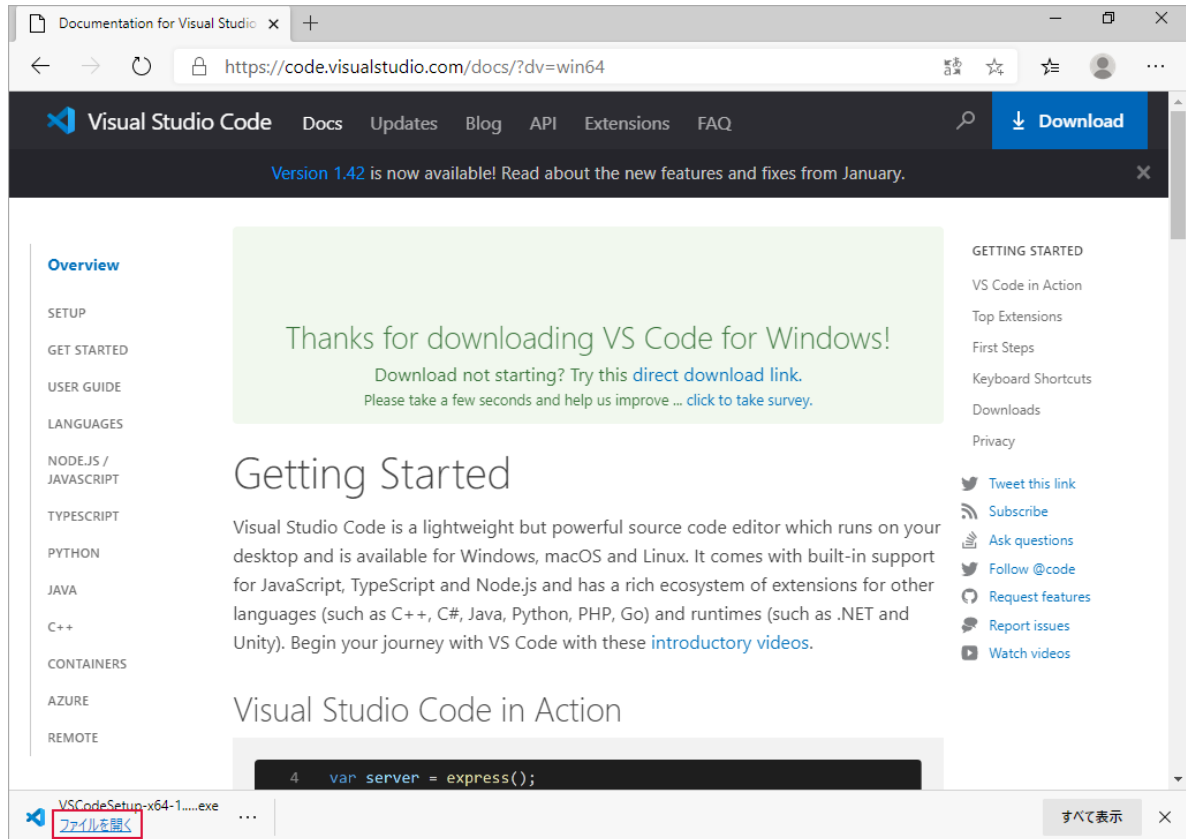
- ・インストールしたユーザーだけが使用できる
- ・Windows にログイン中のユーザーの領域（"%HOMEPATH%" フォルダー内）にインストールする

#### System Installer

- ・誰でも使用できる
- ・"C:¥Program Files" フォルダー内にインストールする

- インストール時に管理者権限が必要

### 3. ファイルを開く をクリック

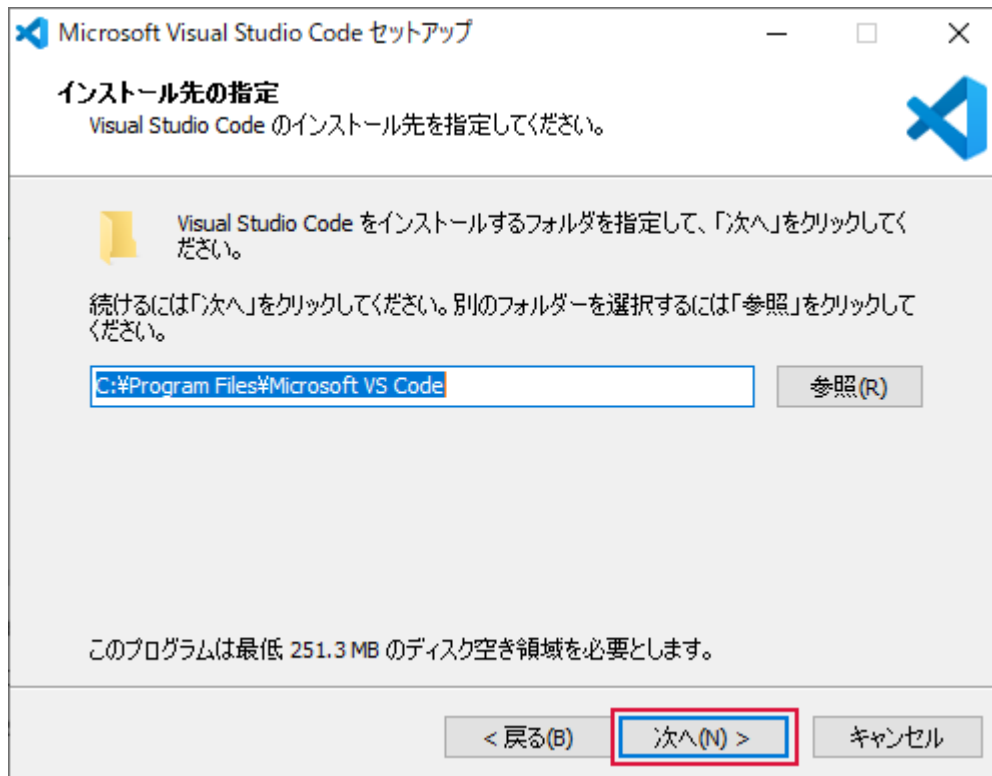


### 4. 同意する (A) を選択 → 次へ (N) をクリック

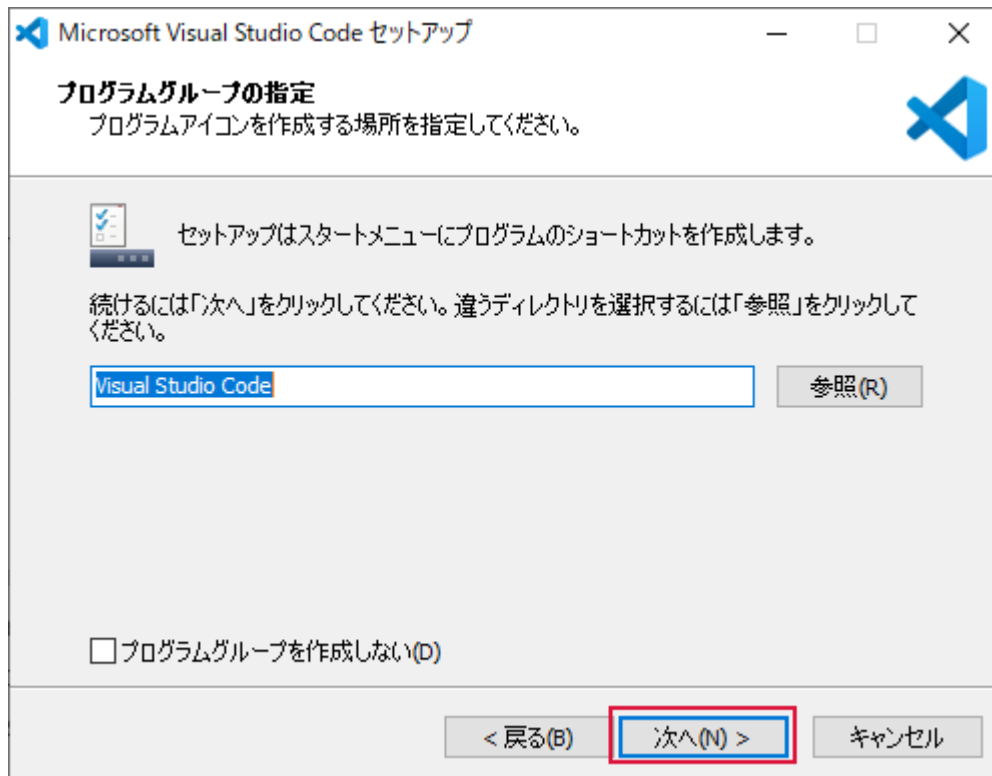


5. 次へ (N) をクリック

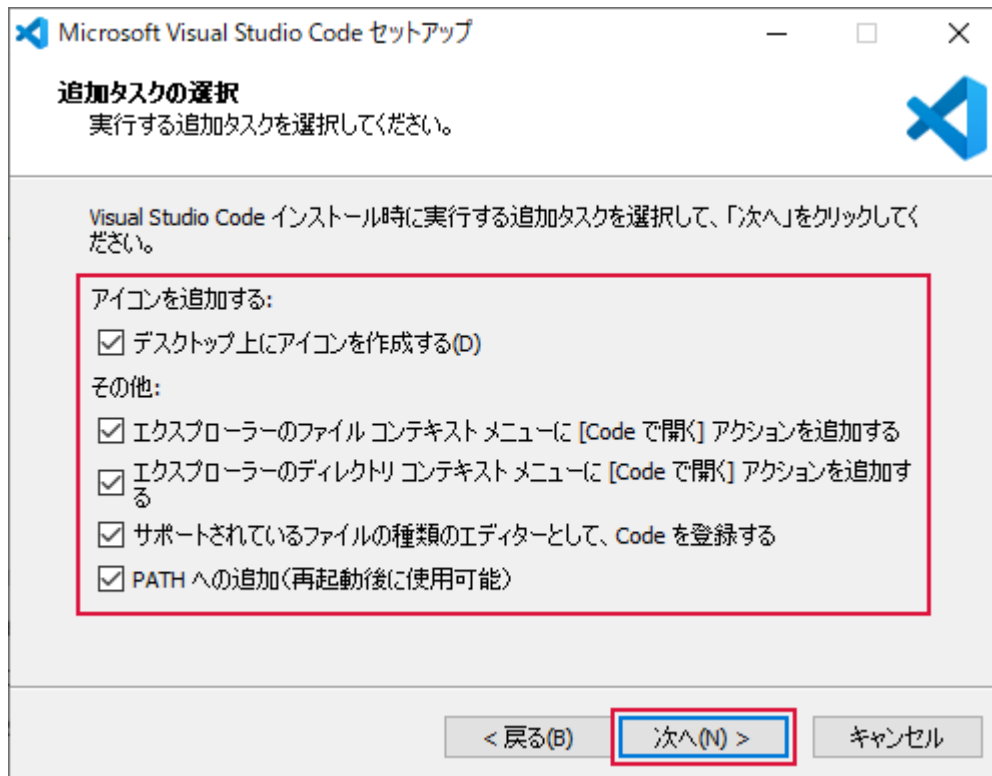




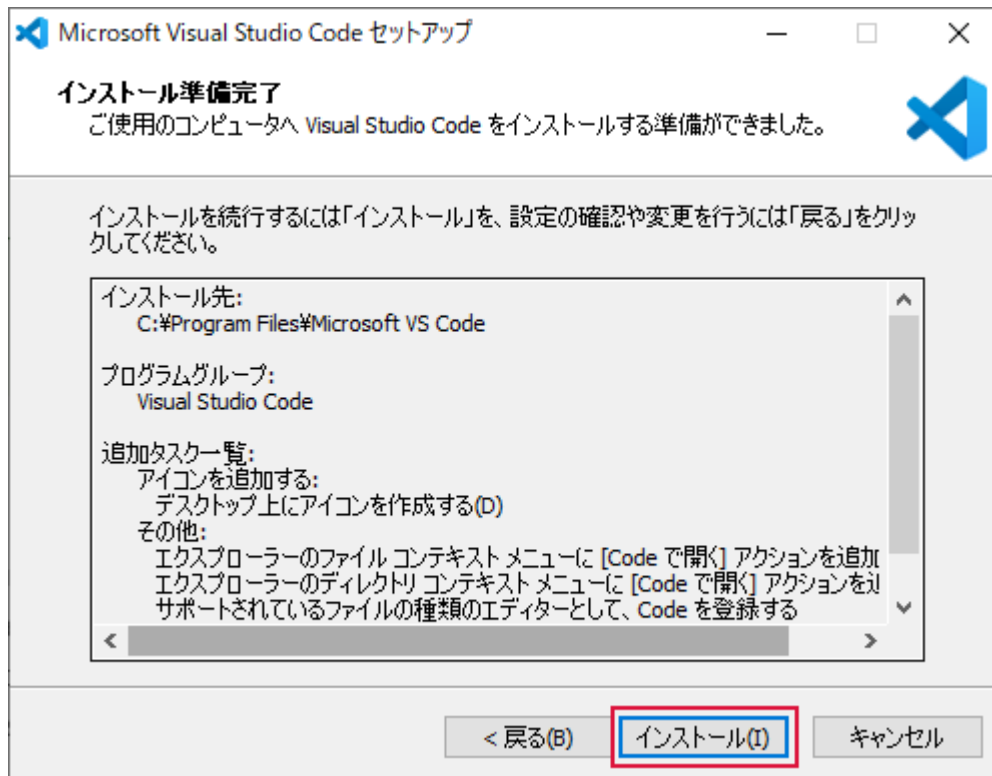
## 6. 次へ (N) をクリック



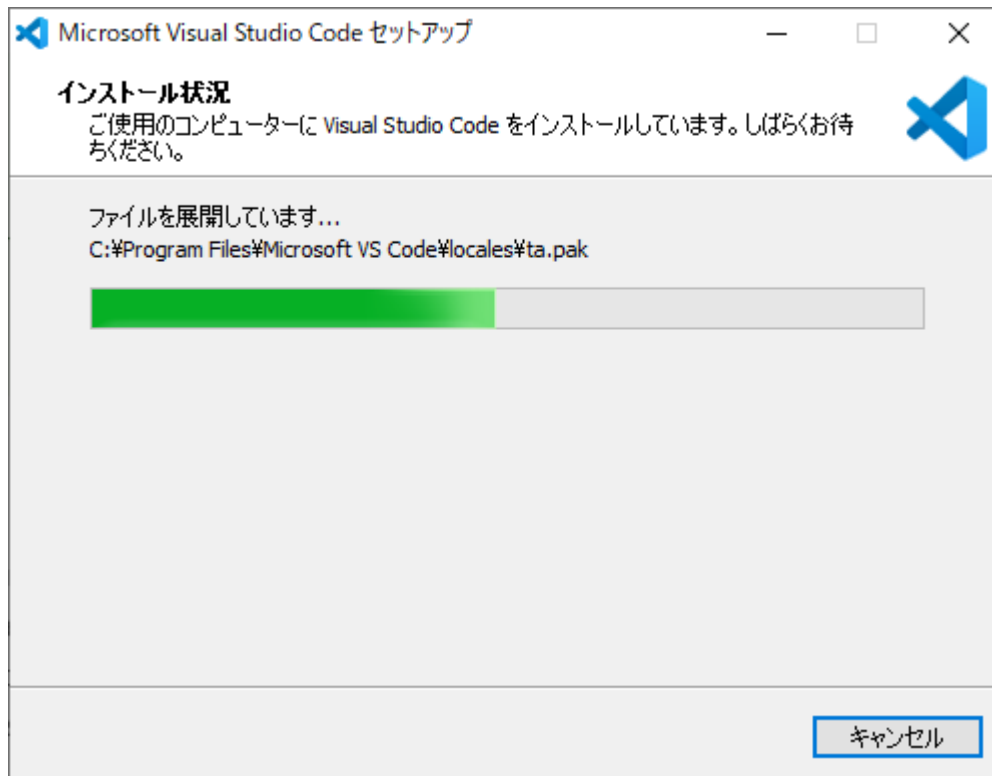
7. 追加タスクは必要なものをチェック → 次へ (N) をクリック



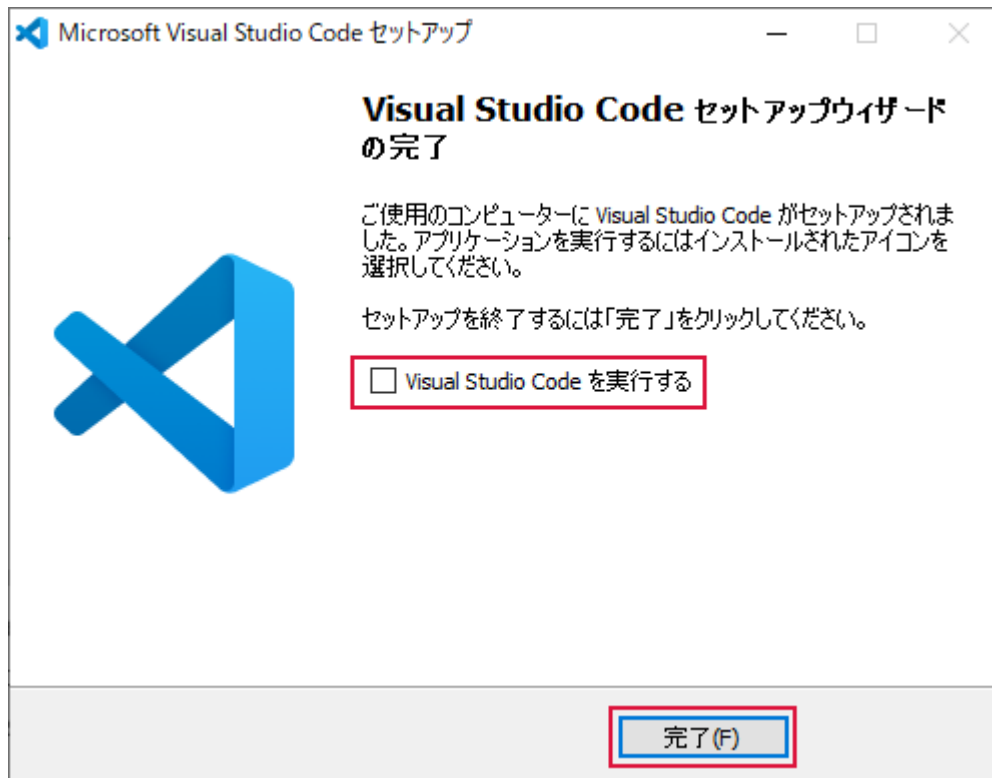
## 8. インストール (I) をクリック



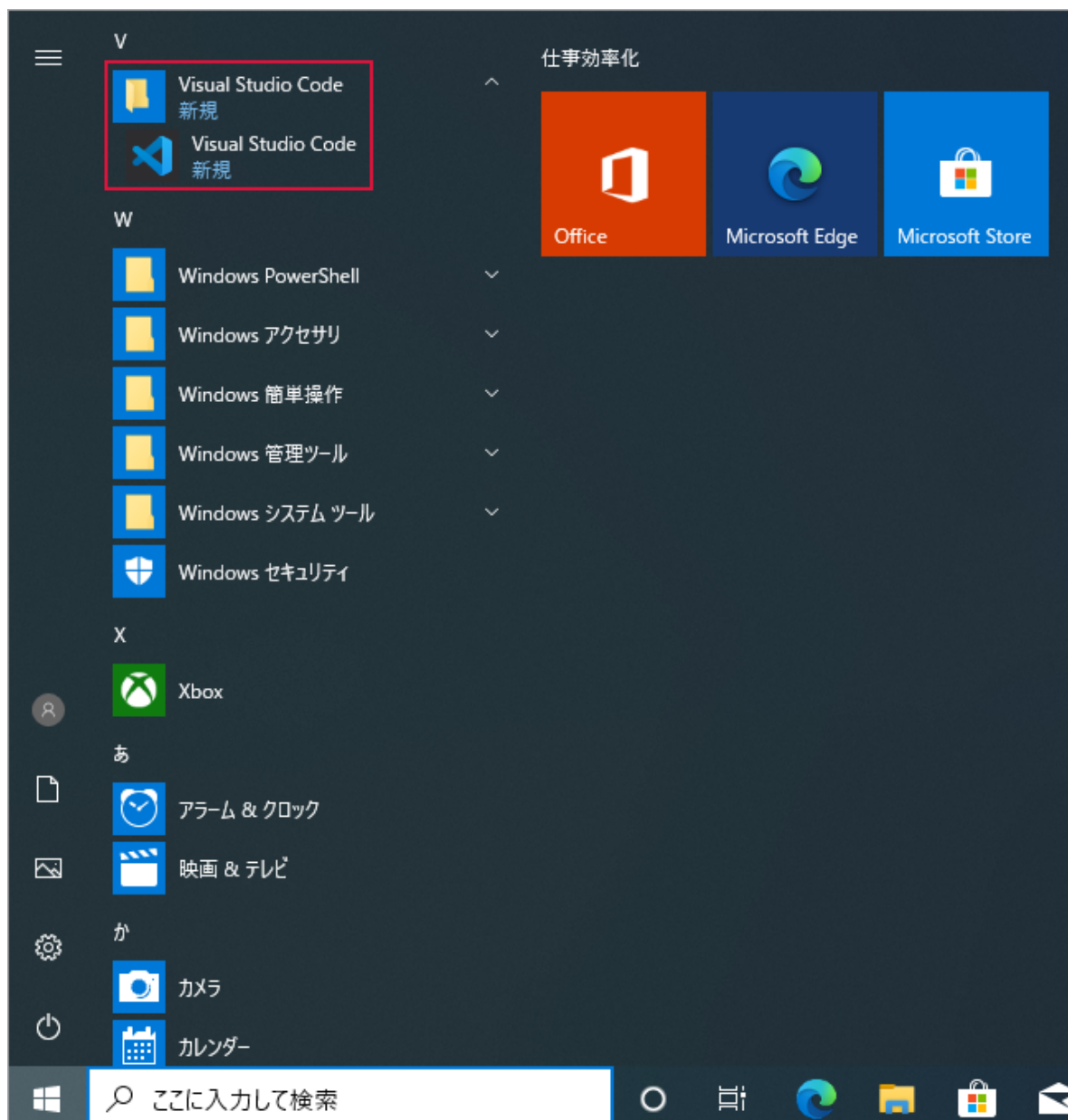
## 9. インストール中



10. Visual Studio Code を実行する のチェックを外す → 完了 (F) をクリック



11. インストールが終了するとメニューに "Visual Studio Code" が登録されます



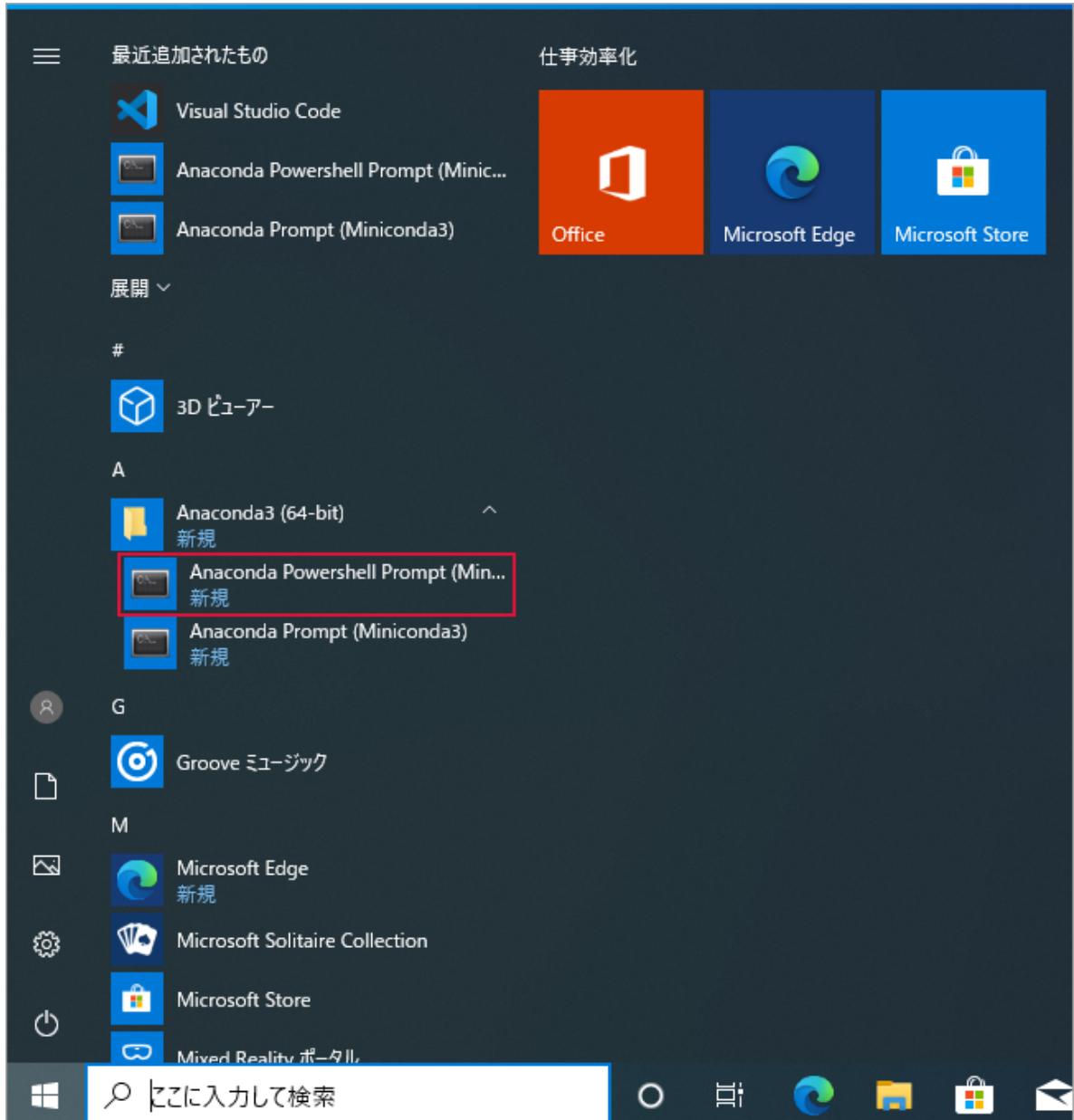
## 2.2 ドキュメントの作成手順

ドキュメントを作る工程の最初から最後までを実際に操作しながら説明します。

### 1. ドキュメントを保存するフォルダーを作る

Sphinx はフォルダー単位でドキュメントを管理します。フォルダー内にドキュメントを構成する複数のファイルやフォルダーが配置されます。まず、これらを保存するフォルダーを作ります。今回は "C:¥Sphinx¥test" フォルダーを作りました。中身は空です。

2. "Anaconda Powershell Prompt(Miniconda3)" を起動



注釈: "Anaconda Powershell Prompt(Miniconda3)" と "Anaconda Prompt ( Miniconda3)" はどちらを起動し

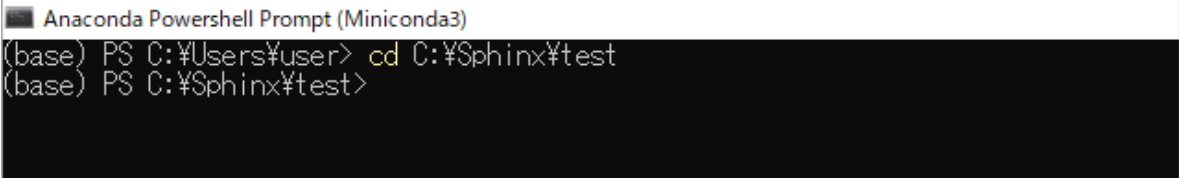


でも良いです。実行するコマンドの指定の仕方に少し違いがあります。

---

3. 「ドキュメントを保存するフォルダーを作」で作成したフォルダーへ移動

`cd C:\Sphinx\test`



```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\user> cd C:\Sphinx\test
(base) PS C:\Sphinx\test>
```

4. `sphinx-quickstart` コマンドを実行

" `sphinx-quickstart` "コマンドを実行して Sphinx でドキュメントを作成するのに必要なファイルを作成します。

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Sphinx\test> sphinx-quickstart
Welcome to the Sphinx 2.4.3 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Selected root path: .

You have two options for placing the build directory for Sphinx output.
Either, you use a directory "_build" within the root path, or you separate
"source" and "build" directories within the root path.
> Separate source and build directories (y/n) [n]: y

The project name will occur in several places in the built documentation.
> Project name: test document
> Author name(s): hoge
> Project release []: 1

If the documents are to be written in a language other than English,
you can select a language here by its language code. Sphinx will then
translate text that it generates into that language.

For a list of supported codes, see
https://www.sphinx-doc.org/en/master/usage/configuration.html#confval-language.
> Project language [en]: ja

Creating file .\source\conf.py.
Creating file .\source\index.rst.
Creating file .\Makefile.
Creating file .\make.bat.

Finished: An initial directory structure has been created.

You should now populate your master file .\source\index.rst and create other doc
source files. Use the Makefile to build the docs, like so:
    make builder
where "builder" is one of the supported builders, e.g. html, latex or linkcheck.
(base) PS C:\Sphinx\test>
```

作成途中でいくつか質問されます。適当に回答しても、後で修正できます。

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

- ソースファイル用フォルダー（ source ）と HTML 用フォルダー（ build ）を分けるかどうかを指定します。
- 分ける場合は `y`、同じフォルダー内に作成する場合は `n`（デフォルト）を入力します。

> Project name:

- プロジェクト名を入力します。
- 入力した値はドキュメントのタイトル部分に表示されます。

> Author name(s):

- 作者名を入力します。
- 入力した値はドキュメントの (C) の後に表示されます。

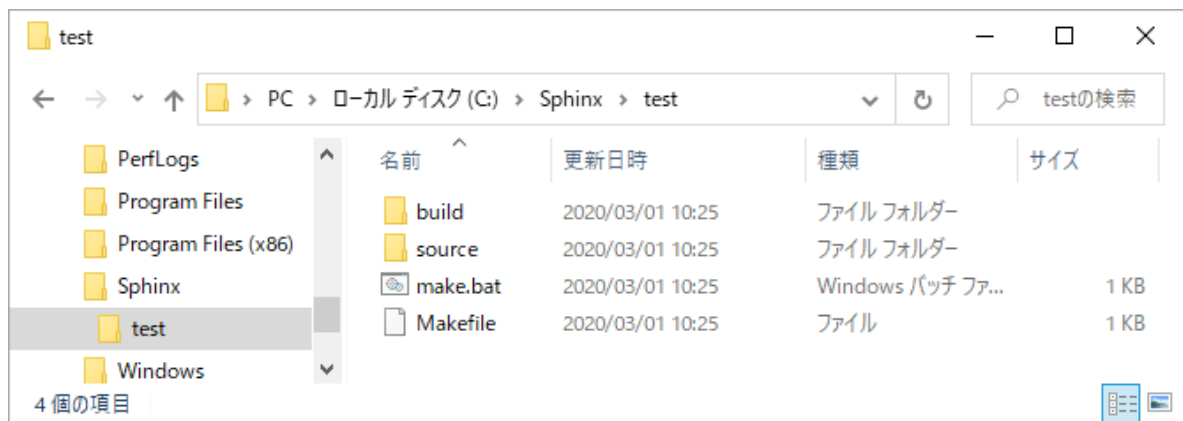
> Project release []:

- ドキュメントのリリース (版) 番号を入力します。

> Project language [en]:

- ドキュメントの言語を入力します。
- 日本語は `ja` を入力します。

**sphinx-quickstart** コマンドの実行後のフォルダーの状態です。



## 5. ドキュメントを作成

ドキュメントを作成 / 修正します。今回はこの作業は省略します。

## 6. HTML に変換

" `./make html` " コマンドを実行して HTML 形式の文書を作成します。

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Sphinx¥test> ./make html
Running Sphinx v2.4.3
loading translations [ja]... done
making output directory... done
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 1 source files that are out of date
updating environment: [new config] 1 added, 0 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] index
generating indices... genindexdone
writing additional pages... searchdone
copying static files... .. done
copying extra files... done
dumping search index in Japanese (code: ja)... done
dumping object inventory... done
build succeeded.

The HTML pages are in build¥html.
(base) PS C:\Sphinx¥test> █
```

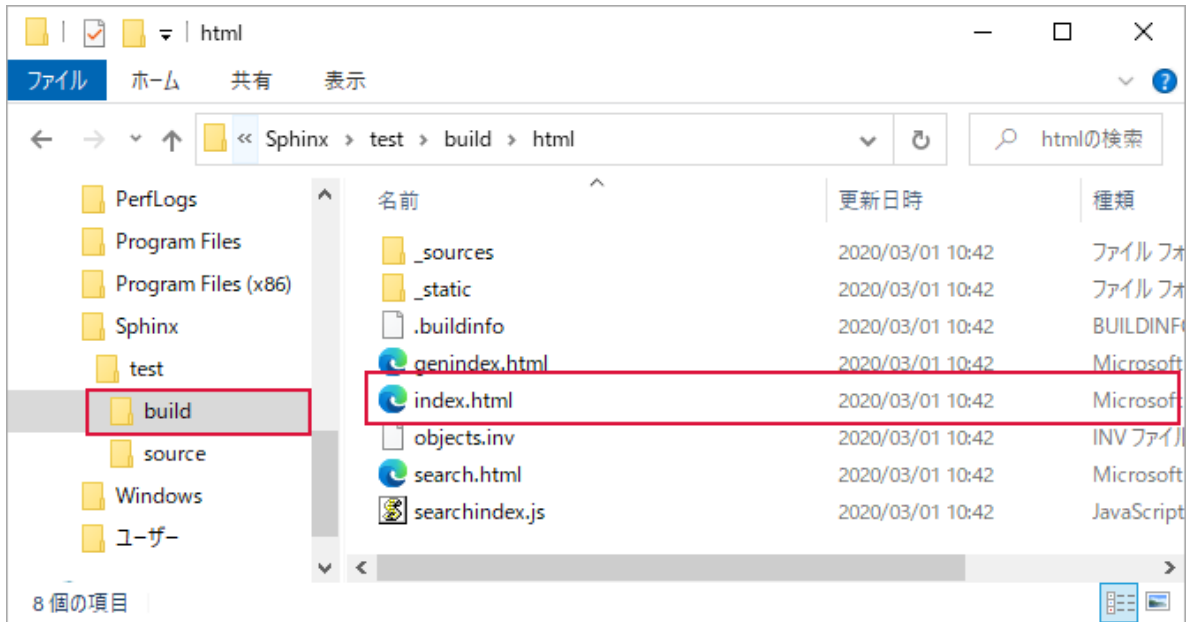
---

注釈: "Anaconda Prompt ( Miniconda3)" は " **make html** " です。

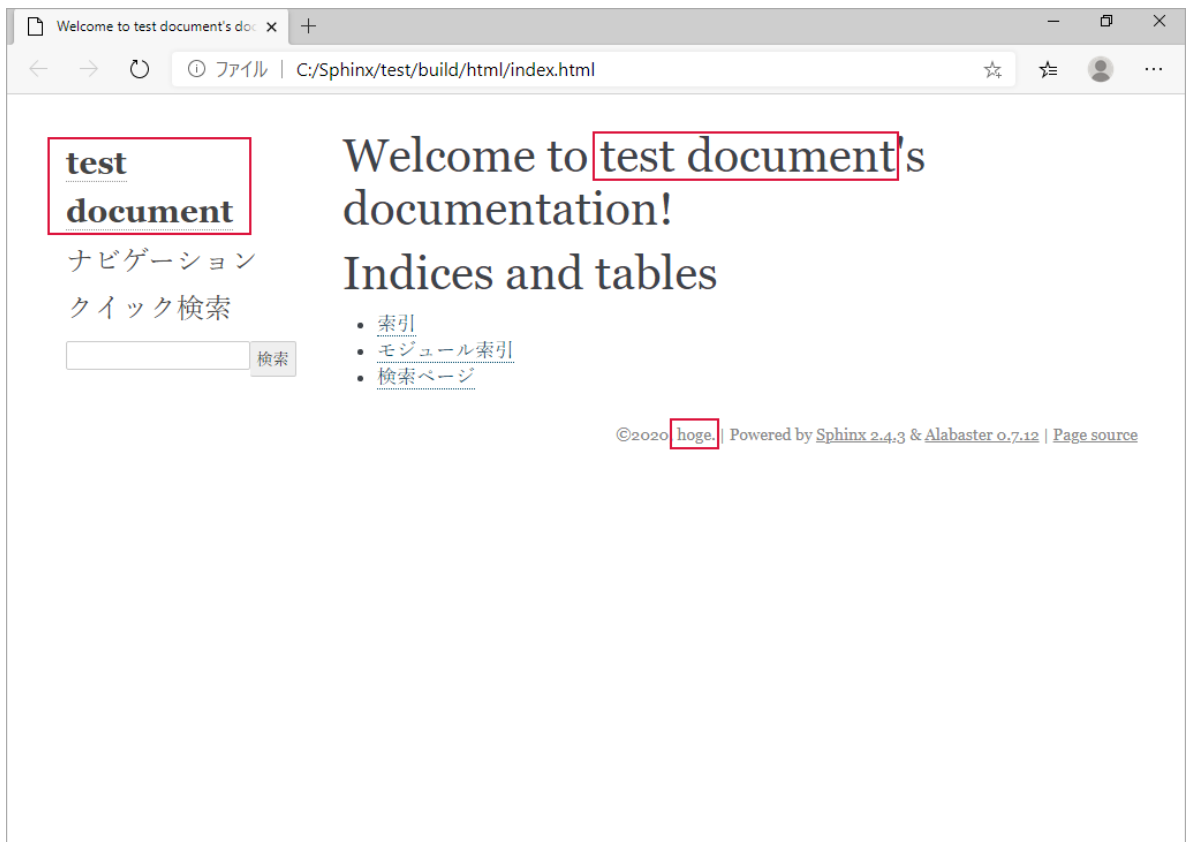
---

## 7. 作成した文書を確認

ブラウザーでドキュメント用フォルダー内の "build¥html" フォルダー内の "index.html" ファイルを開きます。今回は "C:\Sphinx ¥ test ¥ build ¥ html ¥ index.html" を開きます。



"index.html" ファイルをブラウザで開いた結果です。 `sphinx-quickstart` コマンドを実行したときに入力した値が表示されています。



---

注釈: 「ドキュメントを作成 / 更新」, 「HTML に変換 ( make html )」, 「確認」の繰り返しで文書を作成します。

---

## 2.3 日本語対応にする (日本語拡張の導入)

日本語の文章を HTML に変換すると、文書の途中に不自然なタイミングで空白が挿入される場合があります。この事象の対応策として日本語拡張を導入します。

---

### 2.3.1 事象の詳細

Sphinx の日本ユーザ会の下記ページを参照ください。

- 不自然な空白が表示される [Python 製ドキュメンテーションビルダー、Sphinx の日本ユーザ会](#)
- 

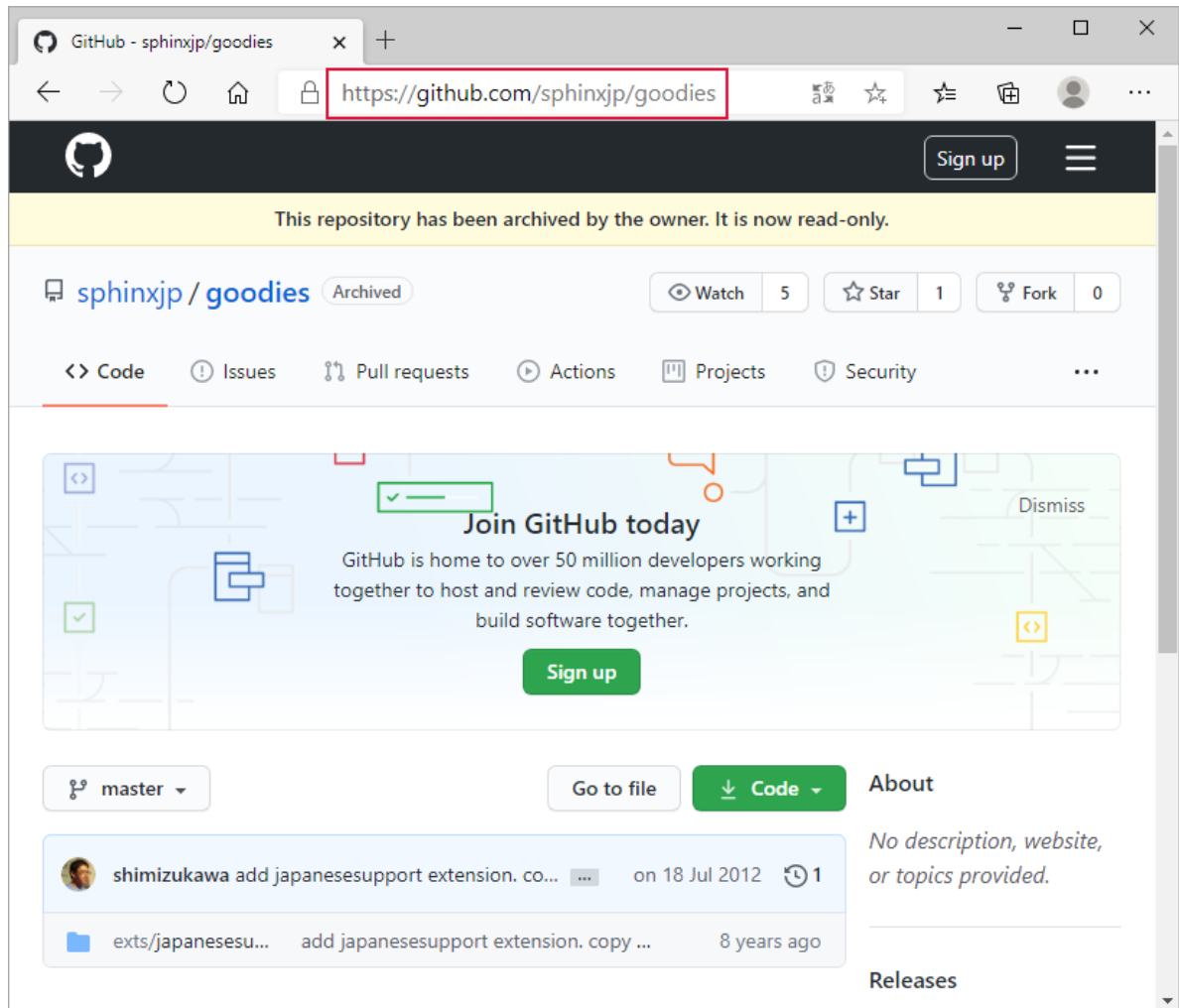
### 2.3.2 日本語拡張の導入手順

---

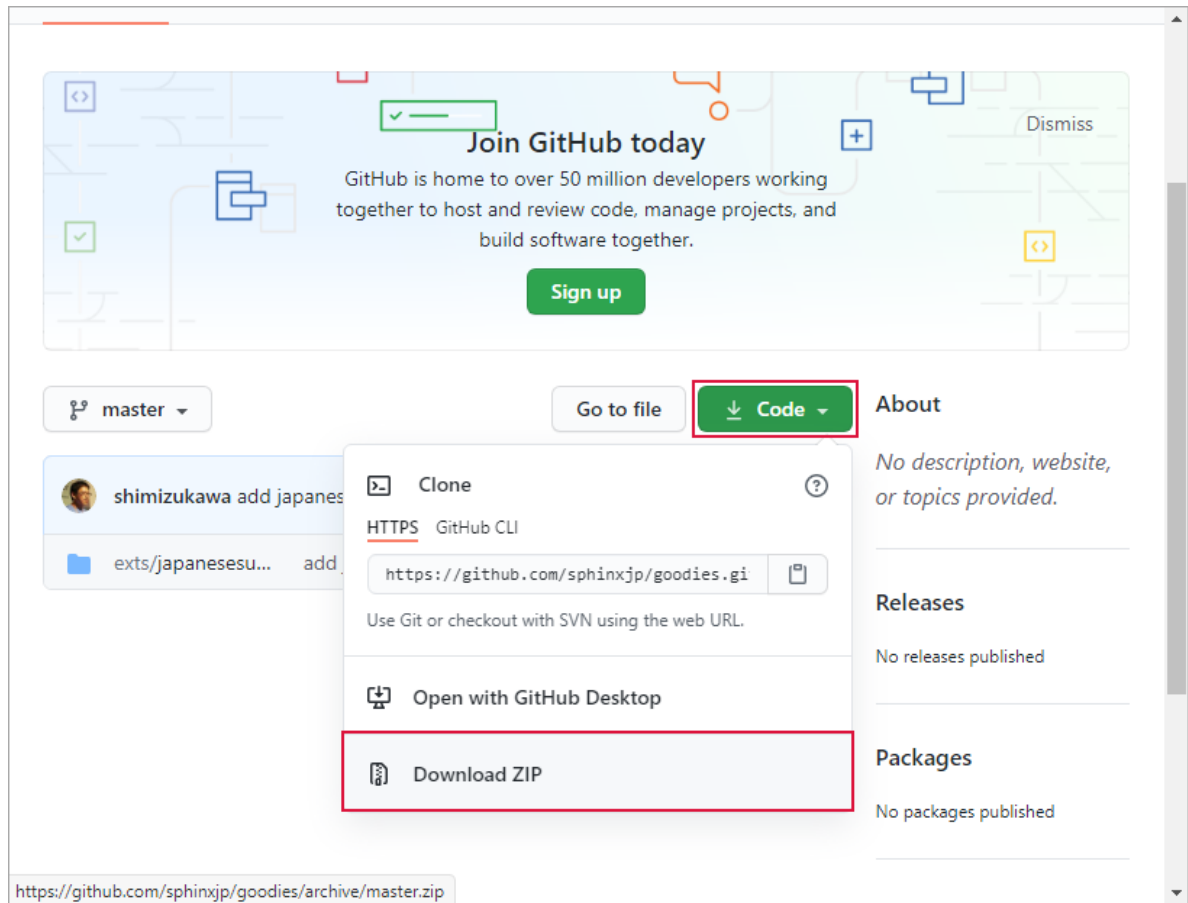
重要: この手順は文書ごとに必要です。

---

1. ブラウザーで <https://github.com/sphinxjp/goodies> を開く

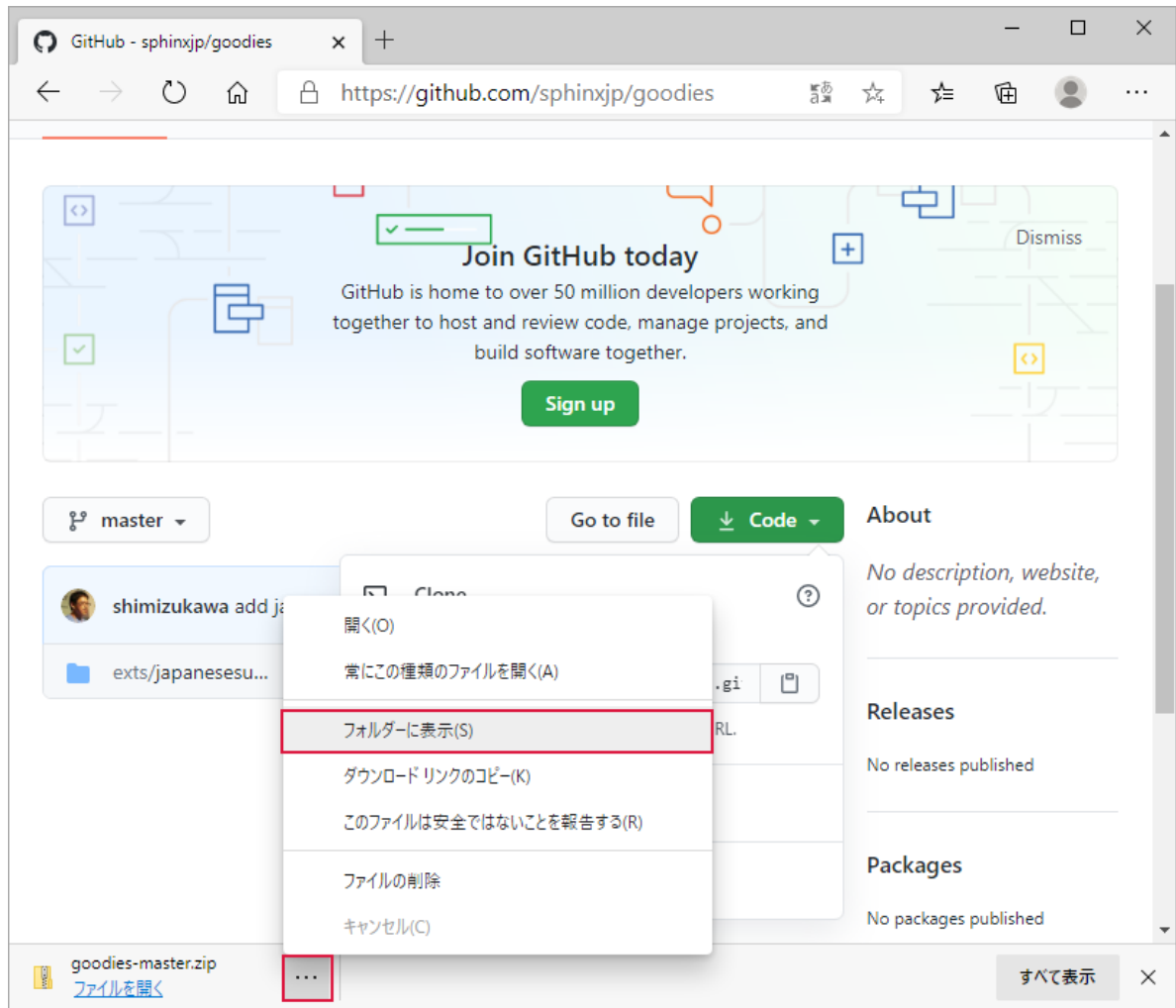


2. Code をクリック → Download ZIP をクリック

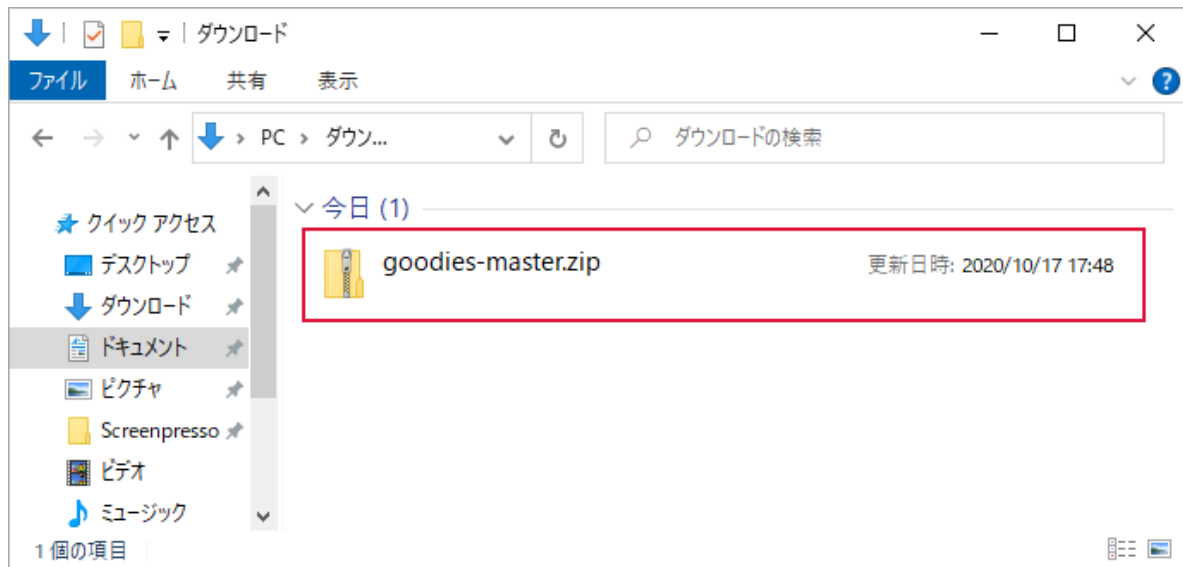


3. ZIP ファイルのダウンロード終了 → ... をクリック → フォルダーに表示 (S) をクリック





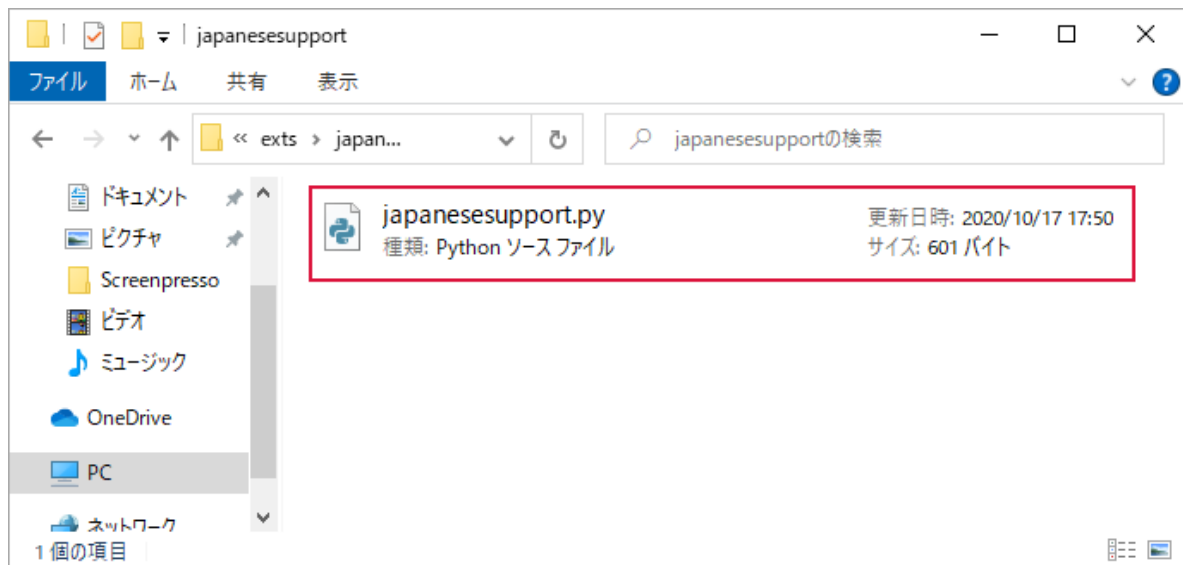
4. ダウンロードした ZIP ファイルを確認 → 展開



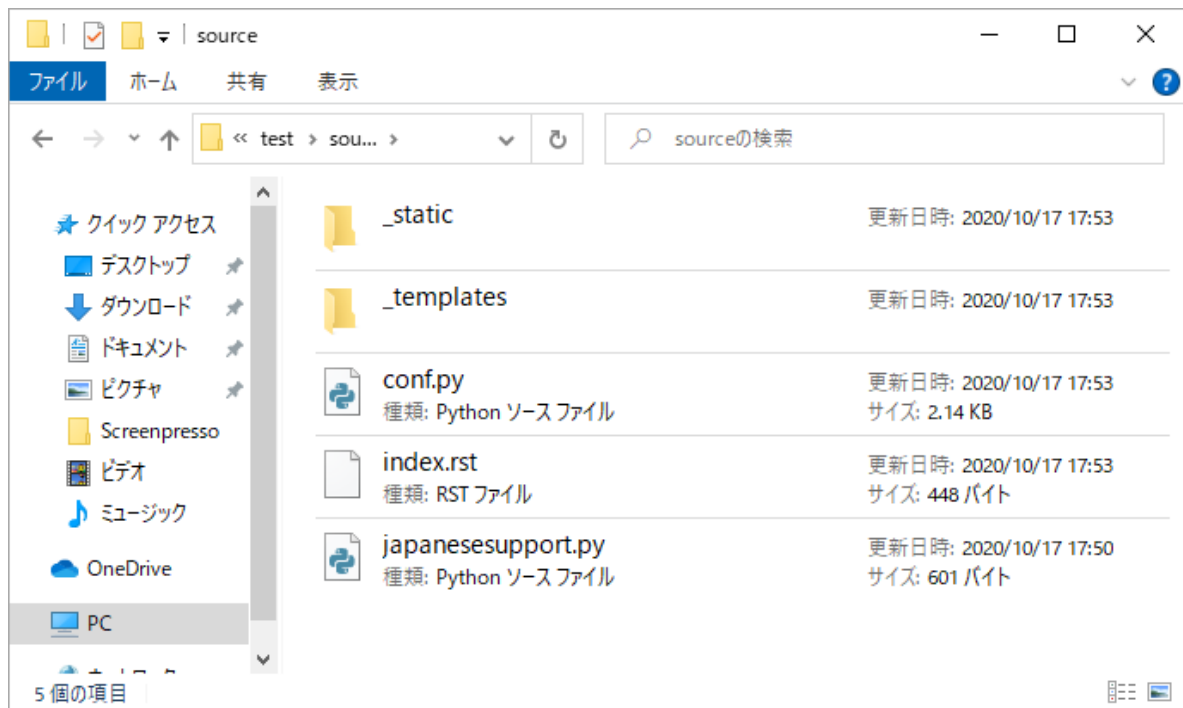
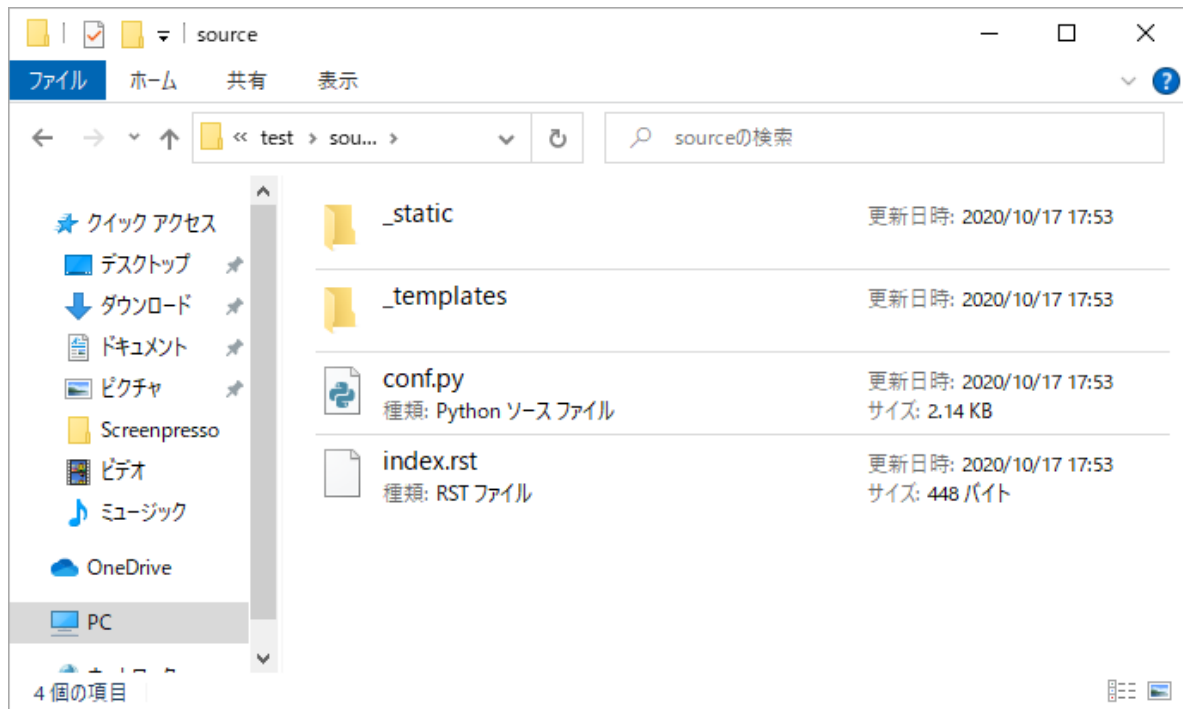
##### 5. 展開後のファイルを確認

ユーザー名が user の場合、次のフォルダーにファイルがあります。

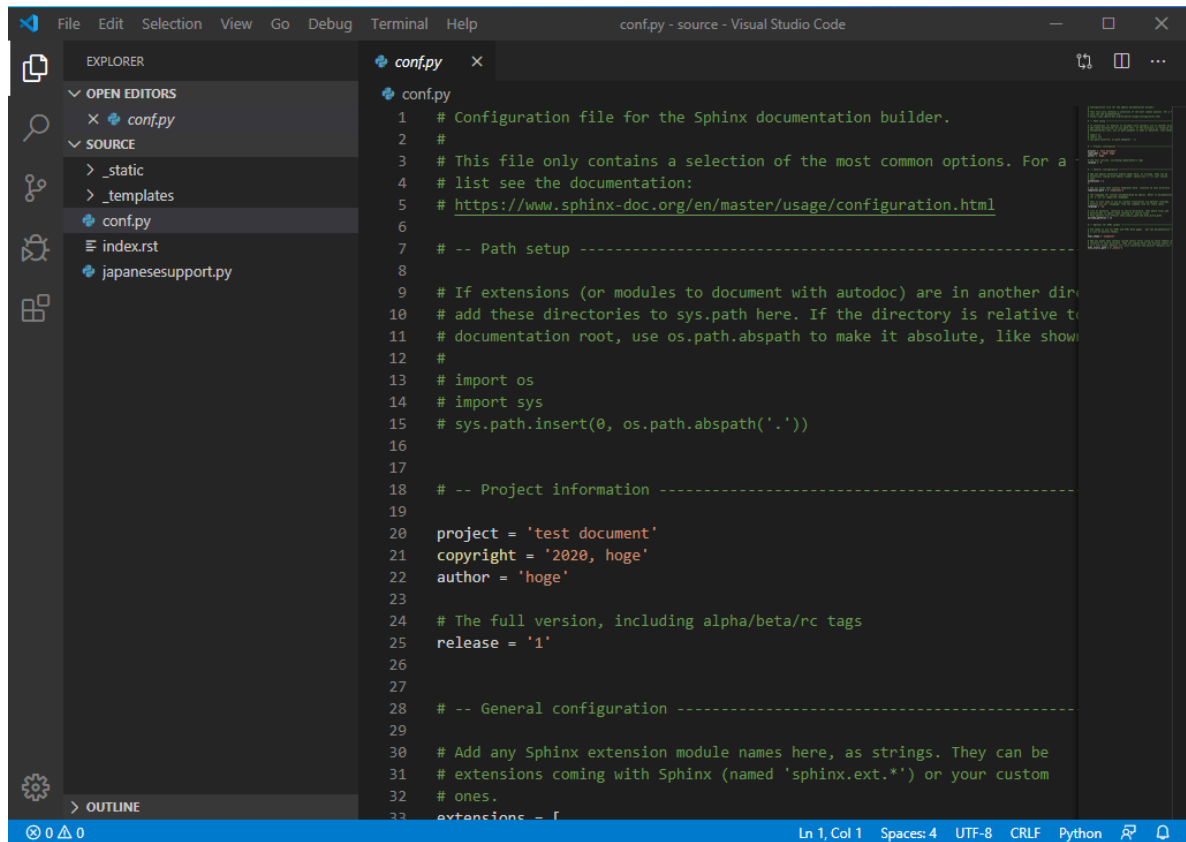
```
C:\Users\user\Downloads\goodies-master\goodies-master\exts\japanesupport
```



## 6. 展開後の "japanesesupport.py" ファイルを "conf.py" ファイルと同じフォルダーに保存



## 7. テキストエディターで "conf.py" ファイルを開く



## 8. 2箇所修正 → 保存

注釈: Sphinx のバージョンにより行位置が異なります。

### 13 行目

- 修正前 `# import os`
- 修正後 `import os`

### 14 行目

- 修正前 `# import sys`
- 修正後 `import sys`

15 行目

- 修正前 `# sys.path.insert(0, os.path.abspath('.'))`
- 修正後 `sys.path.insert(0, os.path.abspath('.'))`

32 行目

- 修正前 `extensions = [`
- 修正後 `extensions = ['japanesesupport'`

```
12  #
13  import os
14  import sys
15  sys.path.insert(0, os.path.abspath('.'))
16
17  # -- Project information -----
18
19  project = 'test document'
20  copyright = '2020, hoge'
21  author = 'hoge'
22
23  # The full version, including alpha/beta/rc tags
24  release = '1'
25
26
27  # -- General configuration -----
28
29  # Add any Sphinx extension module names here, as strings. They can be
30  # extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
31  # ones
32  extensions = ['japanesesupport'
33  ]
34
35  # Add any paths that contain templates here, relative to this directory.
36  templates_path = ['_templates']
```

9. 導入終了

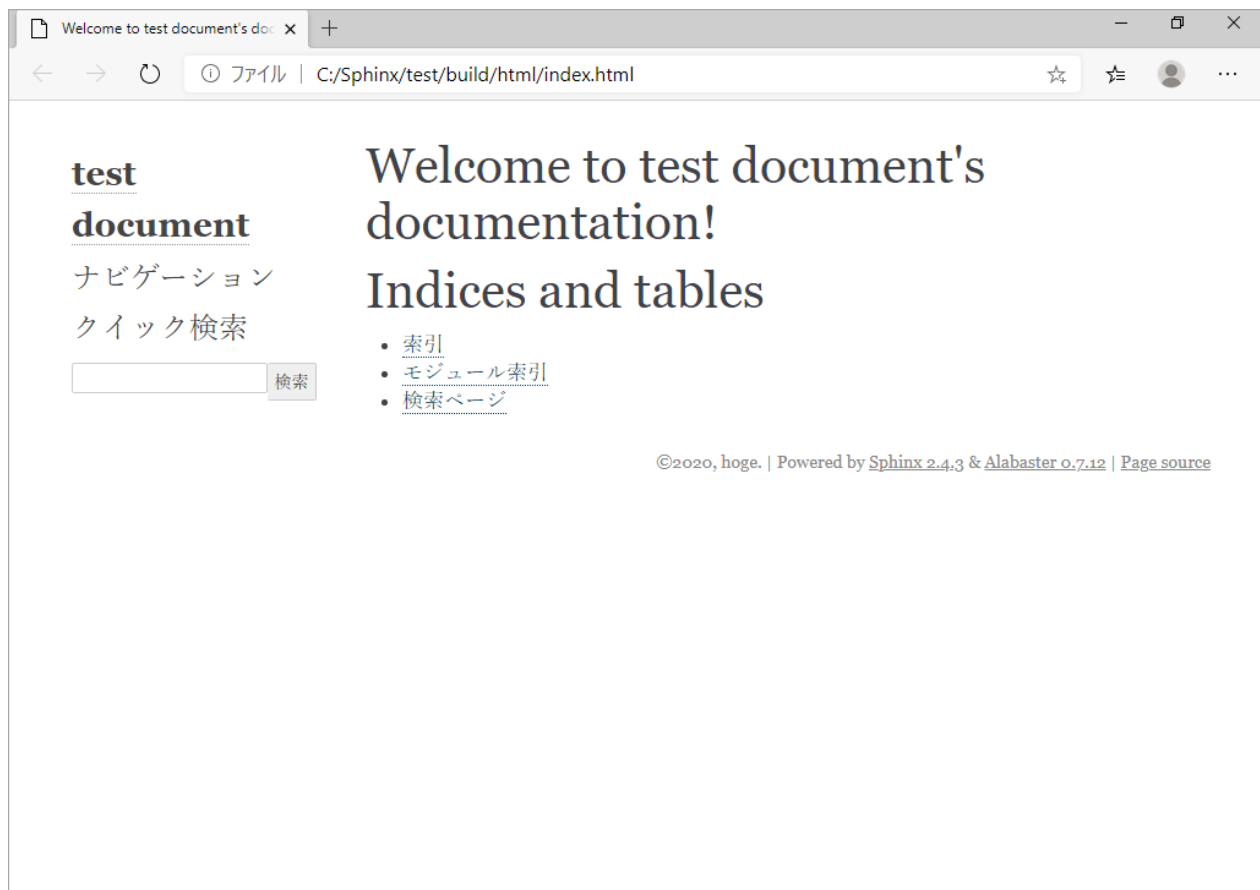
## 2.4 「テーマ」を変更する

### 【トピックス】

- ・「テーマ」とは
- ・「テーマ」の変更方法
- ・ *Sphinx Themes*

### 2.4.1 「テーマ」とは

Sphinx で作成した文書の「見た目」を決めるものが「テーマ」です。この「テーマ」を変更すると「見た目」も変更になります。Sphinx で作成した文書の「見た目」は「テーマ」で決まります。

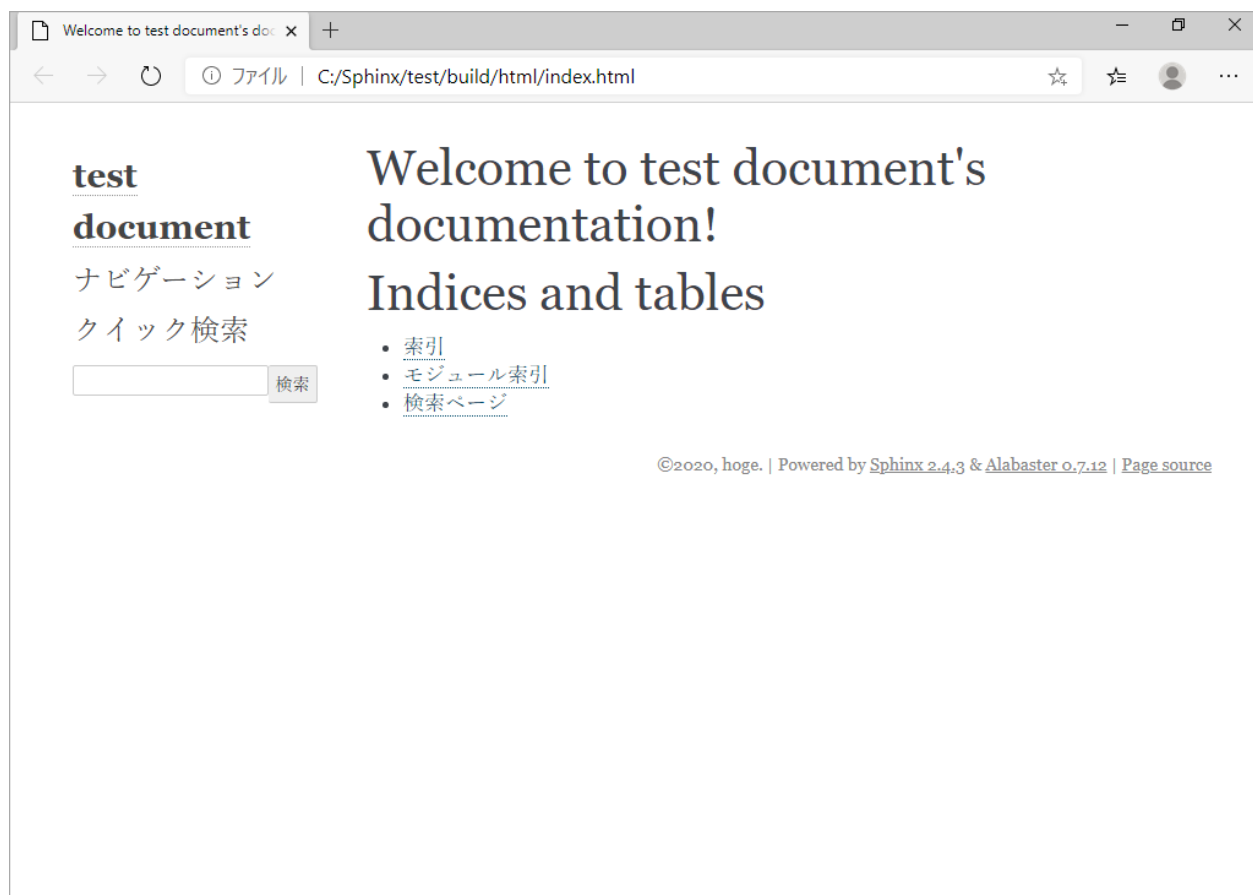


テーマごとに名前が付いており、上図のテーマ名は「alabaster」です。Sphinx が用意しているテーマ（ビルトインテーマ）は次の 10 種類です。いくつかのテーマは「見た目」以外に、複数のオプションを設定できます。

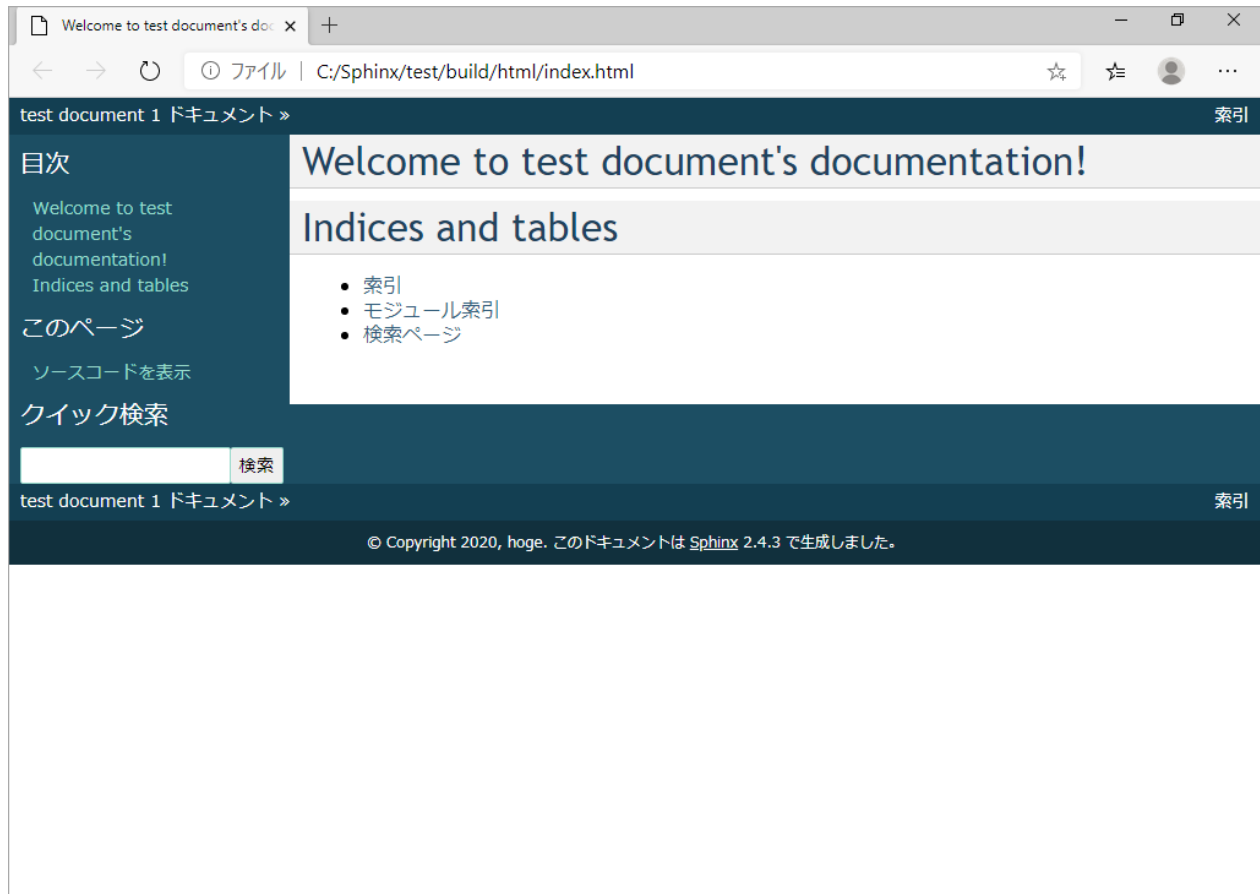
注釈: ビルトインテーマの詳細は次のページを参照ください。

- [HTML](#) Sphinx 4.0.0+/4633ab906 documentation

## alabaster デフォルトのテーマ

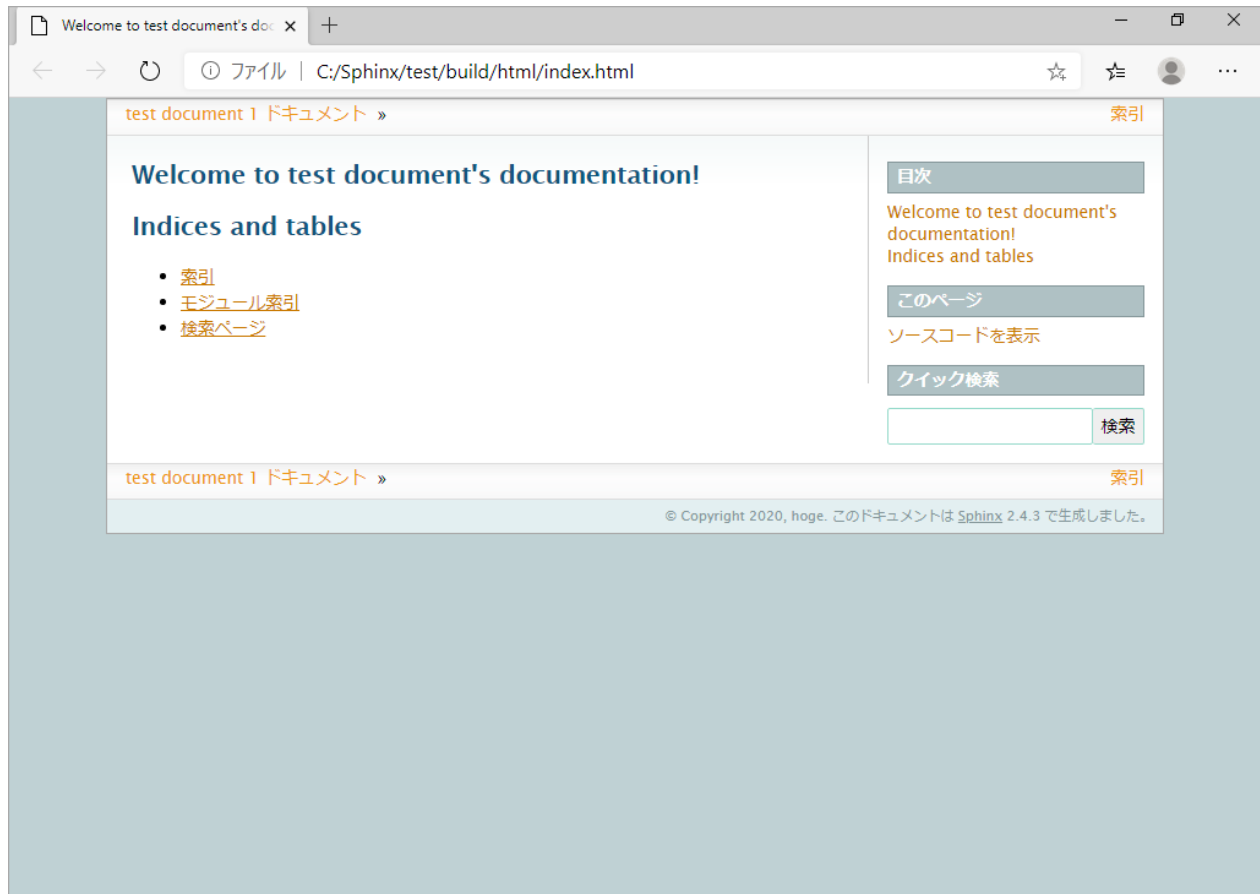


## classic

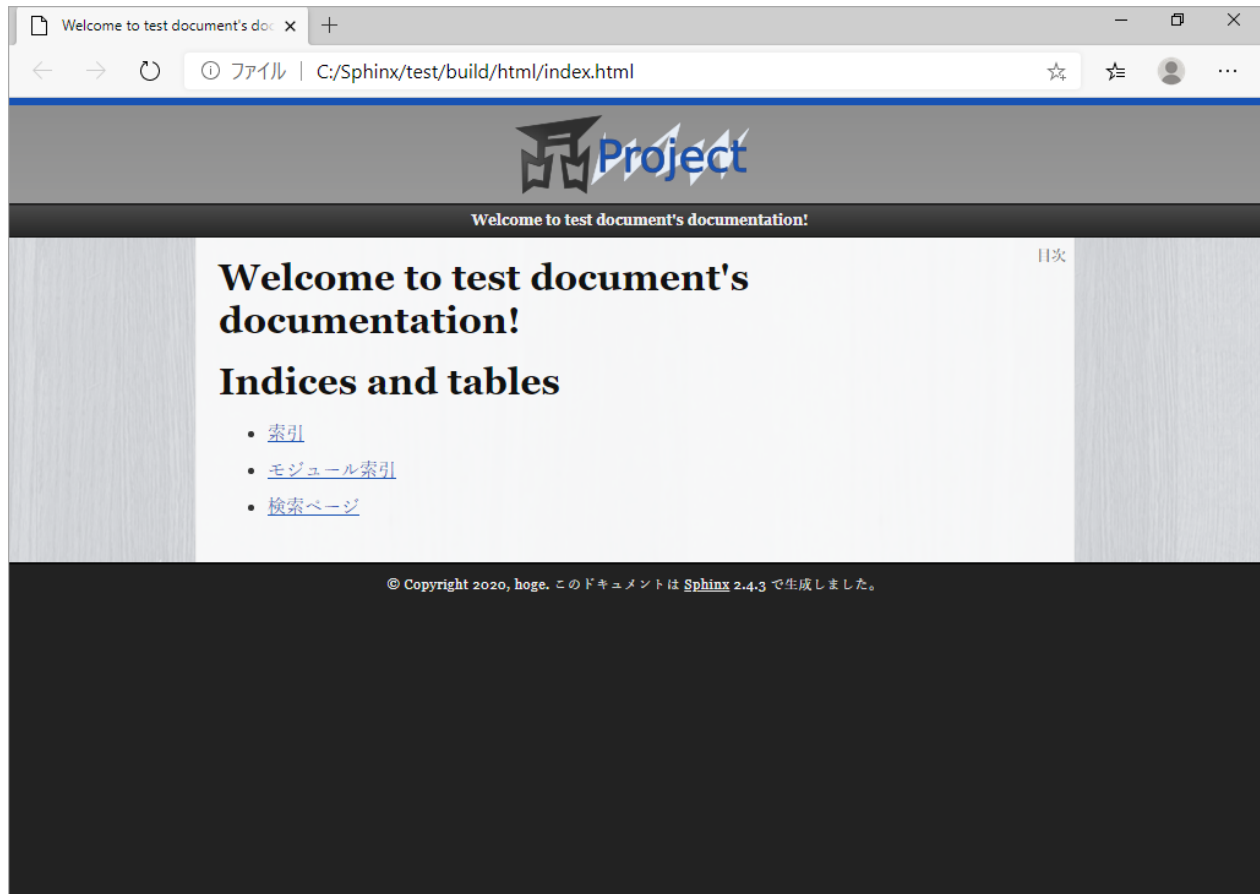




## sphinxdoc



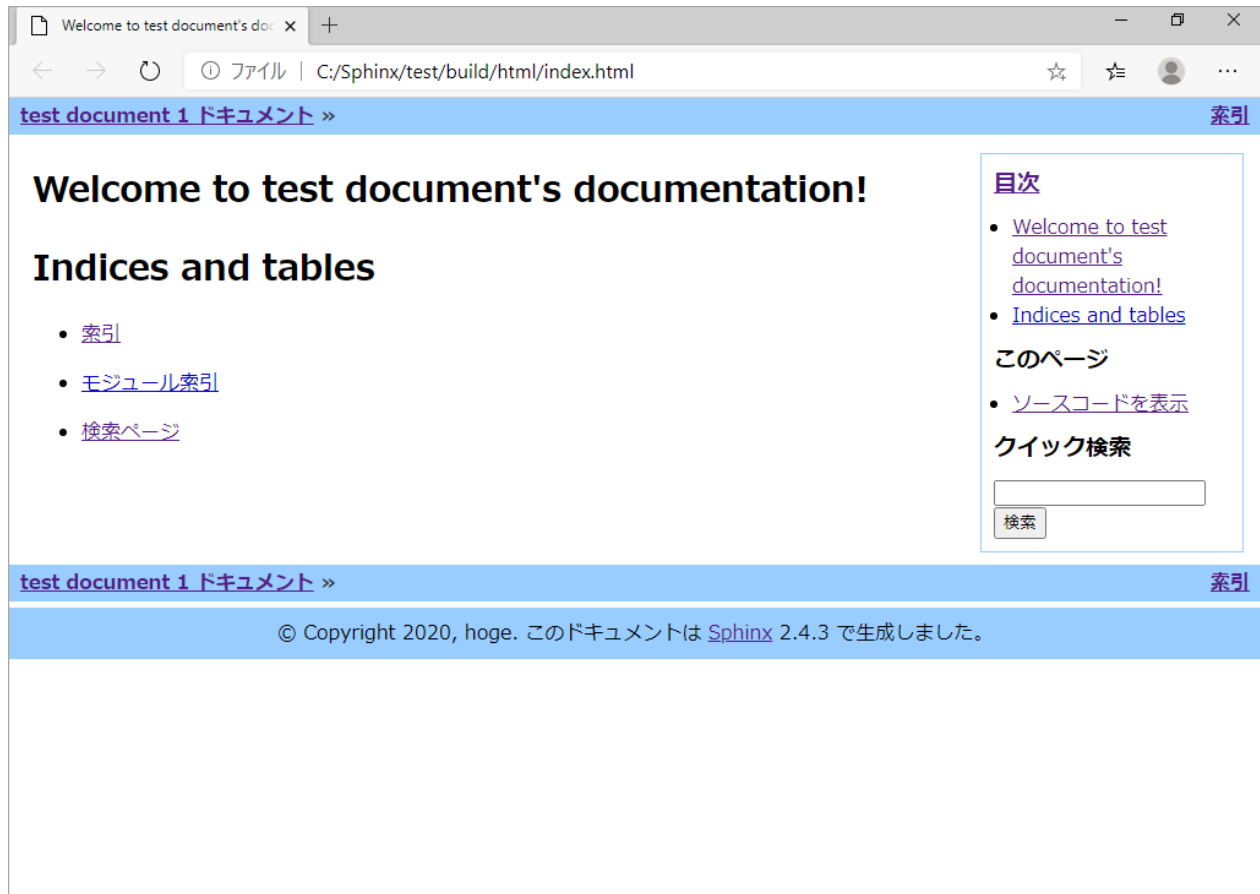
## scrolls



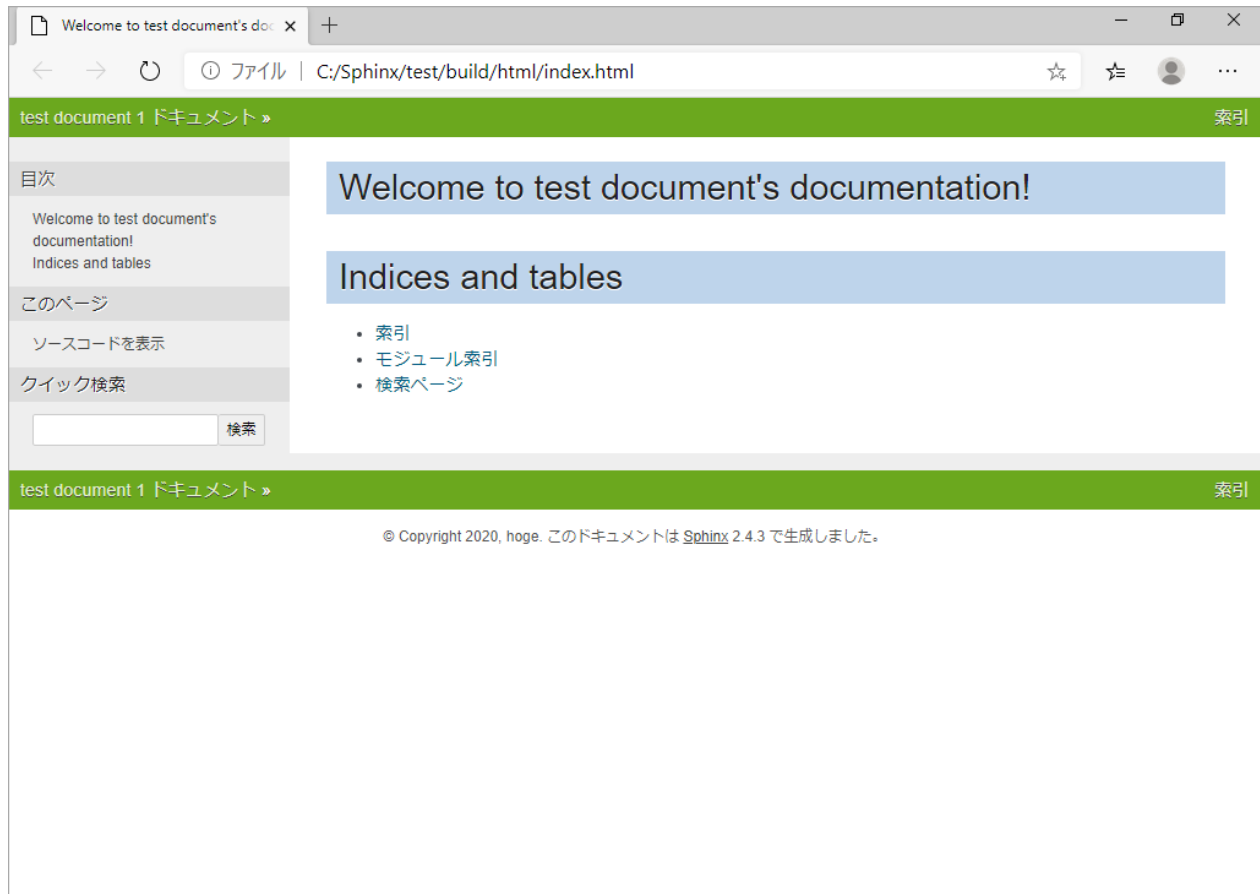
agogo



## traditional



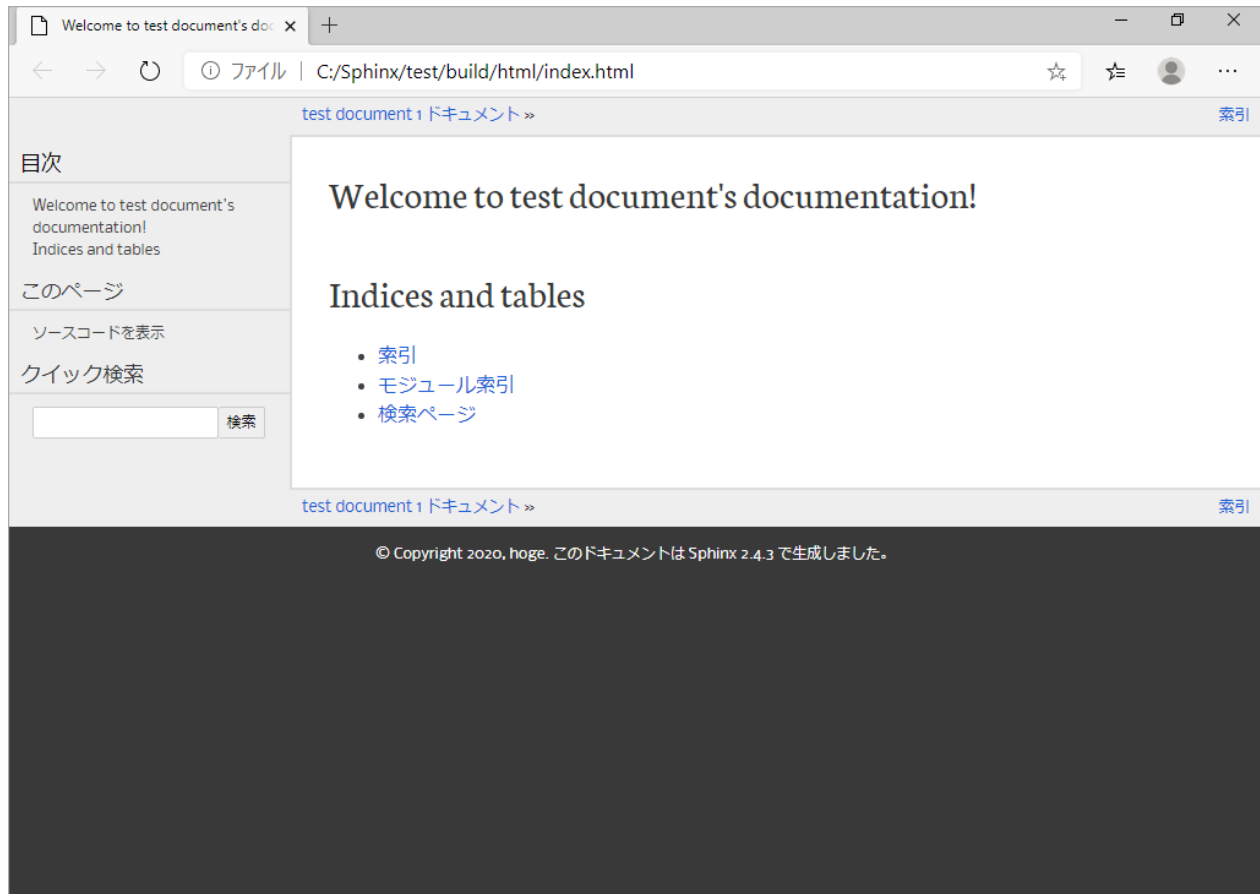
## nature



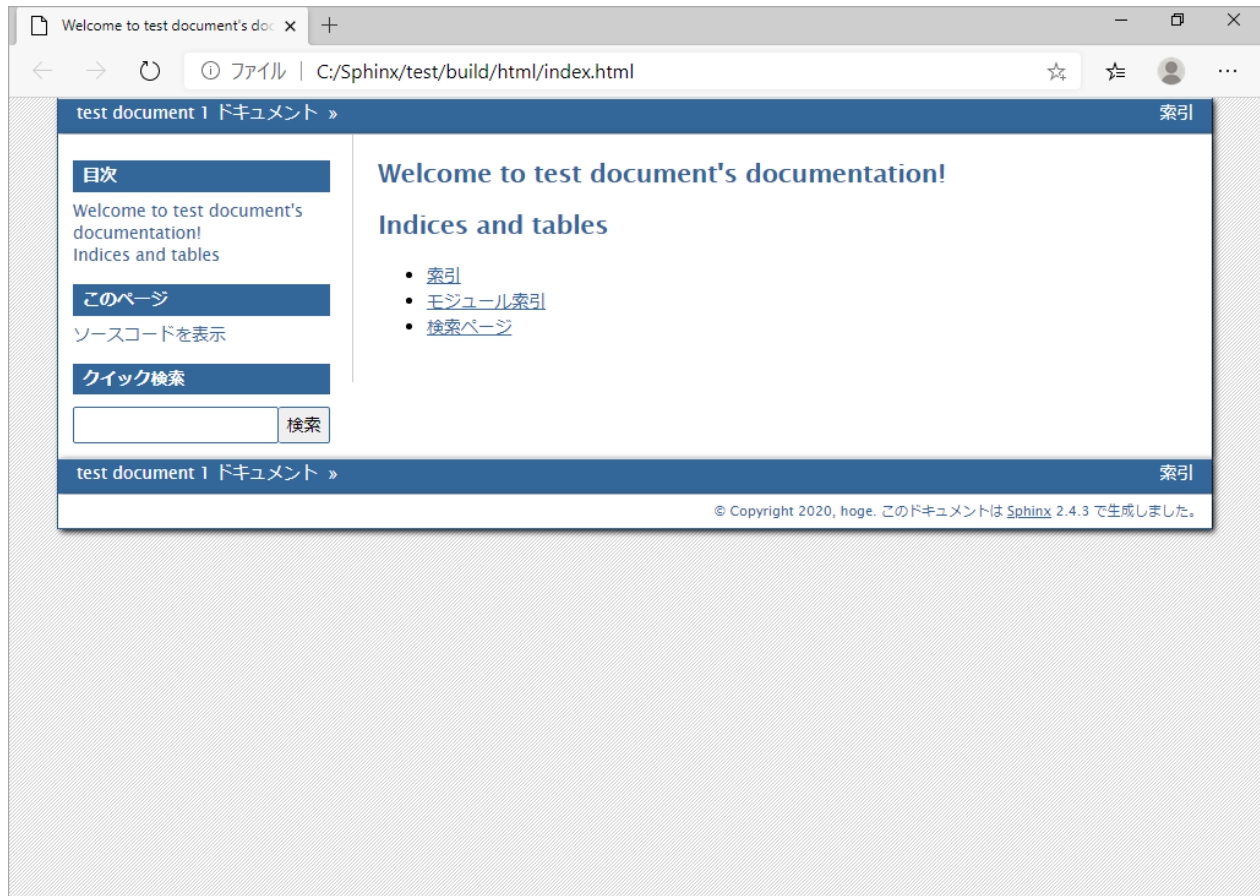
## haiku



## pyramid



## bizstyle



### 2.4.2 「テーマ」の変更方法

"conf.py" ファイル内の `html_theme` で使用するテーマを指定します。



```
38 # The language for content autogenerated by Sphinx. Refer to documentati
39 # for a list of supported languages.
40 #
41 # This is also used if you do content translation via gettext catalogs.
42 # Usually you set "language" from the command line for these cases.
43 language = 'ja'
44
45 # List of patterns, relative to source directory, that match files and
46 # directories to ignore when looking for source files.
47 # This pattern also affects html_static_path and html_extra_path.
48 exclude_patterns = []
49
50
51 # -- Options for HTML output -----
52
53 # The theme to use for HTML and HTML Help pages. See the documentation
54 # a list of builtin themes.
55 #
56 html_theme = 'alabaster'
57
58 # Add any paths that contain custom static files (such as style sheets)
59 # relative to this directory. They are copied after the builtin static f
60 # so a file named "default.css" will overwrite the builtin "default.css"
61 html_static_path = ['_static']
```

"alabaster" から "classic" にするには次のように変更・保存します。必要に応じて、テーマのオプション設定を行います。

```
38 # The language for content autogenerated by Sphinx. Refer to documentati
39 # for a list of supported languages.
40 #
41 # This is also used if you do content translation via gettext catalogs.
42 # Usually you set "language" from the command line for these cases.
43 language = 'ja'
44
45 # List of patterns, relative to source directory, that match files and
46 # directories to ignore when looking for source files.
47 # This pattern also affects html_static_path and html_extra_path.
48 exclude_patterns = []
49
50
51 # -- Options for HTML output -----
52
53 # The theme to use for HTML and HTML Help pages. See the documentation
54 # a list of builtin themes.
55 #
56 html_theme = 'classic'
57
58 # Add any paths that contain custom static files (such as style sheets)
59 # relative to this directory. They are copied after the builtin static f
60 # so a file named "default.css" will overwrite the builtin "default.css"
61 html_static_path = ['_static']
```

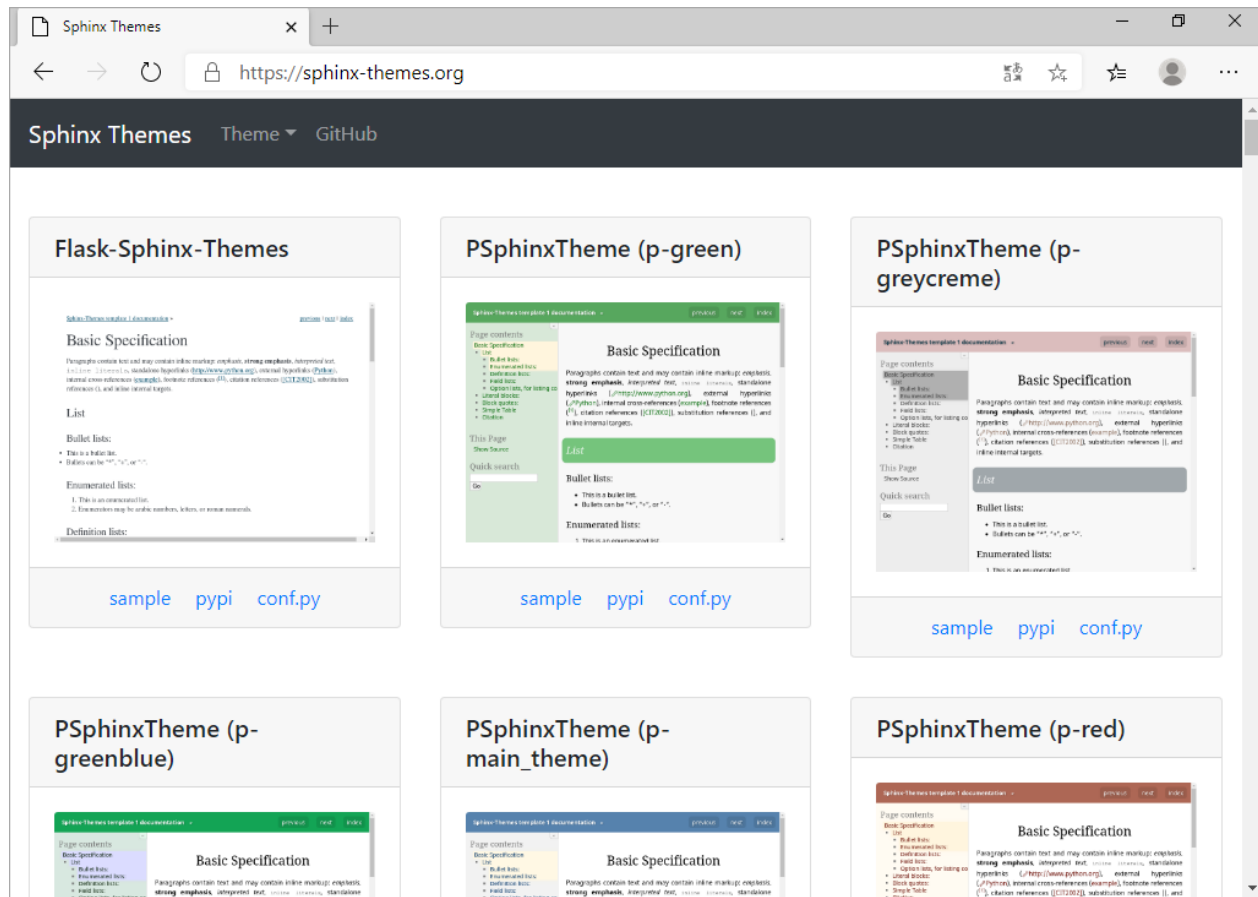
その後、" `make html` "を実行して HTML を再作成します。

```
■ 選択Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Sphinx\test> ./make html
Running Sphinx v2.4.3
loading translations [ja]... done
loading pickled environment... done
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 1 source files that are out of date
updating environment: 0 added, 0 changed, 0 removed
looking for now-outdated files... none found
preparing documents... done
writing output... [100%] index
generating indices... genindexdone
writing additional pages... searchdone
copying static files... .. done
copying extra files... done
dumping search index in Japanese (code: ja)... done
dumping object inventory... done
build succeeded.

The HTML pages are in build\html.
(base) PS C:\Sphinx\test>
```

### 2.4.3 Sphinx Themes

Sphinx Themes に多くのテーマが掲載されています。



各テーマには 3 つのリンクが設定されています。

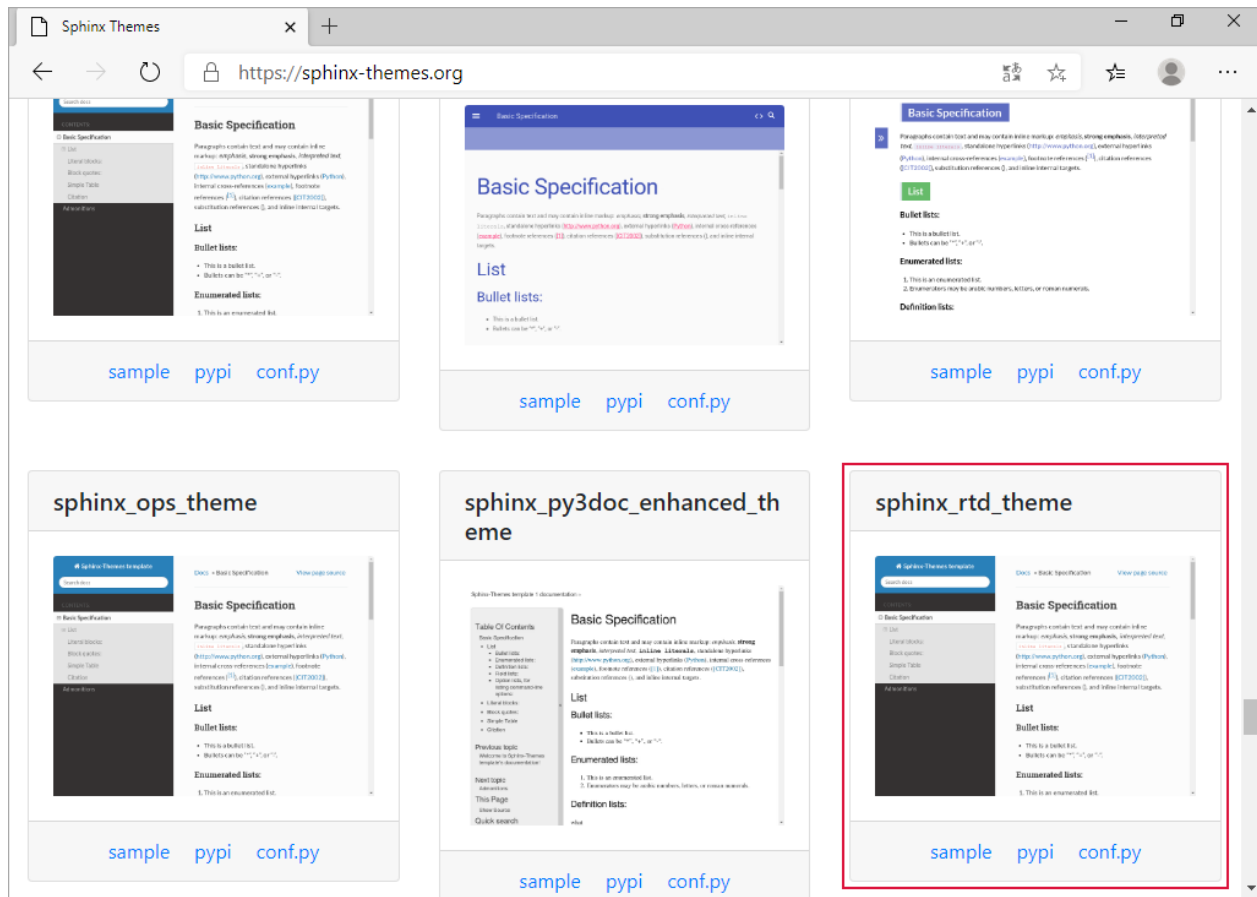
sample サンプル表示を確認できます。

pypi インストール方法などの説明が記載されています。

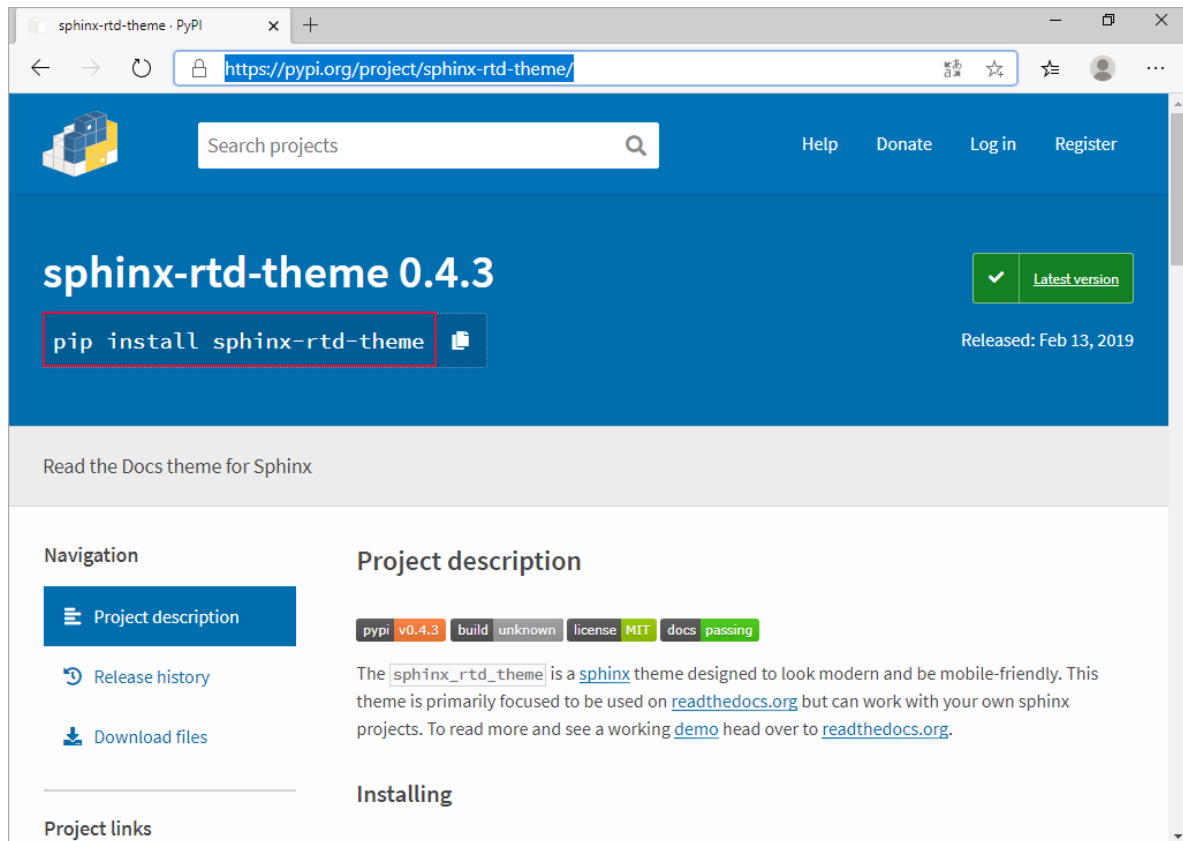
conf.py "conf.py" ファイルの設定サンプルが記載されています。

## 導入手順

今回は "sphinx\_rtd\_theme" の導入を例に説明します。



1. `pypi` をクリックして導入方法を確認



2. "Anaconda Powershell Prompt(Miniconda3)" でインストールコマンドを実行

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

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Sphinx\test> pip install sphinx-rtd-theme

Requirement already satisfied: pyparsing>=2.0.2 in c:\users\user\miniconda3\lib\site-packages (2.4.6)
Requirement already satisfied: six in c:\users\user\miniconda3\lib\site-packages (1.12.0)
Requirement already satisfied: pytz>=2015.7 in c:\users\user\miniconda3\lib\site-packages (2019.3)
Installing collected packages: sphinx-rtd-theme
Successfully installed sphinx-rtd-theme-0.4.3
(base) PS C:\Sphinx\test>
```

3. "conf.py" ファイルの `html_theme` を変更

```
38 # The language for content autogenerated by Sphinx. Refer to documentati
39 # for a list of supported languages.
40 #
41 # This is also used if you do content translation via gettext catalogs.
42 # Usually you set "language" from the command line for these cases.
43 language = 'ja'
44
45 # List of patterns, relative to source directory, that match files and
46 # directories to ignore when looking for source files.
47 # This pattern also affects html_static_path and html_extra_path.
48 exclude_patterns = []
49
50
51 # -- Options for HTML output -----
52
53 # The theme to use for HTML and HTML Help pages. See the documentation
54 # a list of builtin themes.
55 #
56 html_theme = 'sphinx_rtd_theme'
57
58 # Add any paths that contain custom static files (such as style sheets)
59 # relative to this directory. They are copied after the builtin static f
60 # so a file named "default.css" will overwrite the builtin "default.css"
61 html_static_path = ['_static']
```

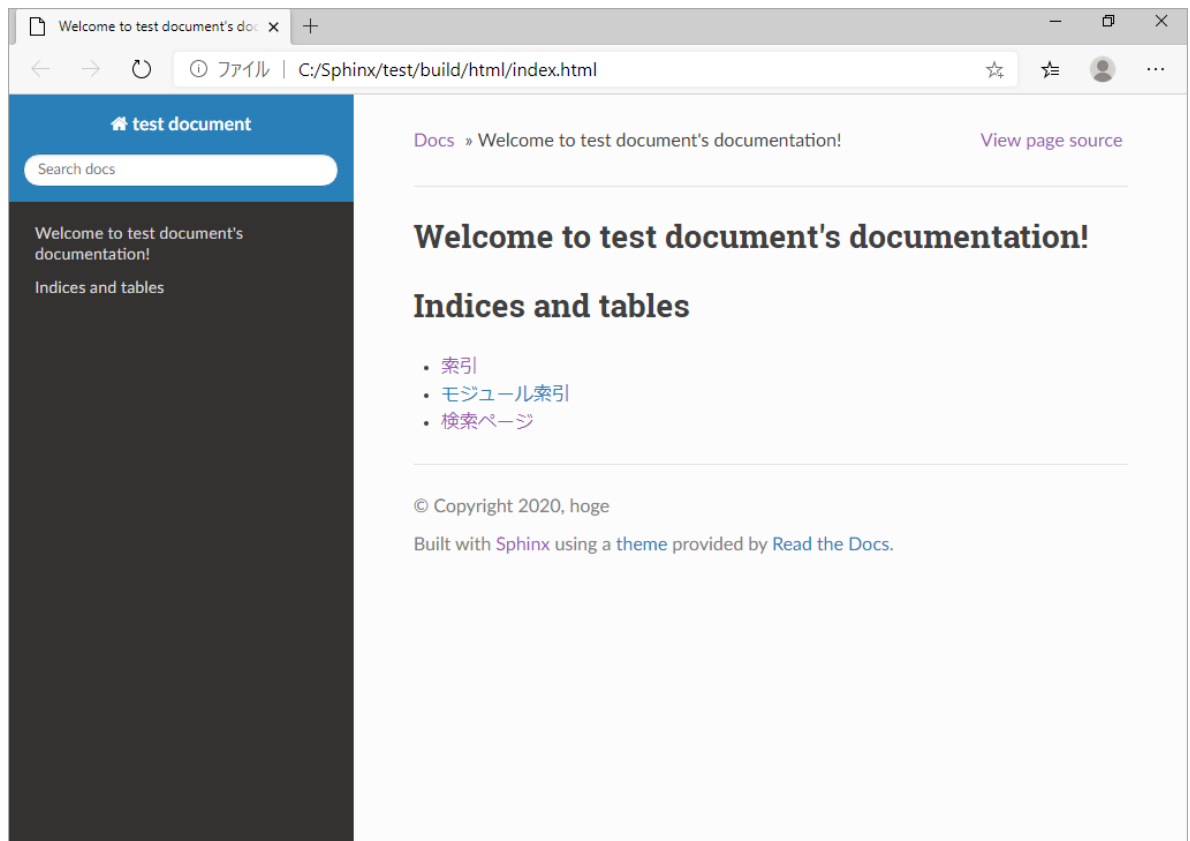
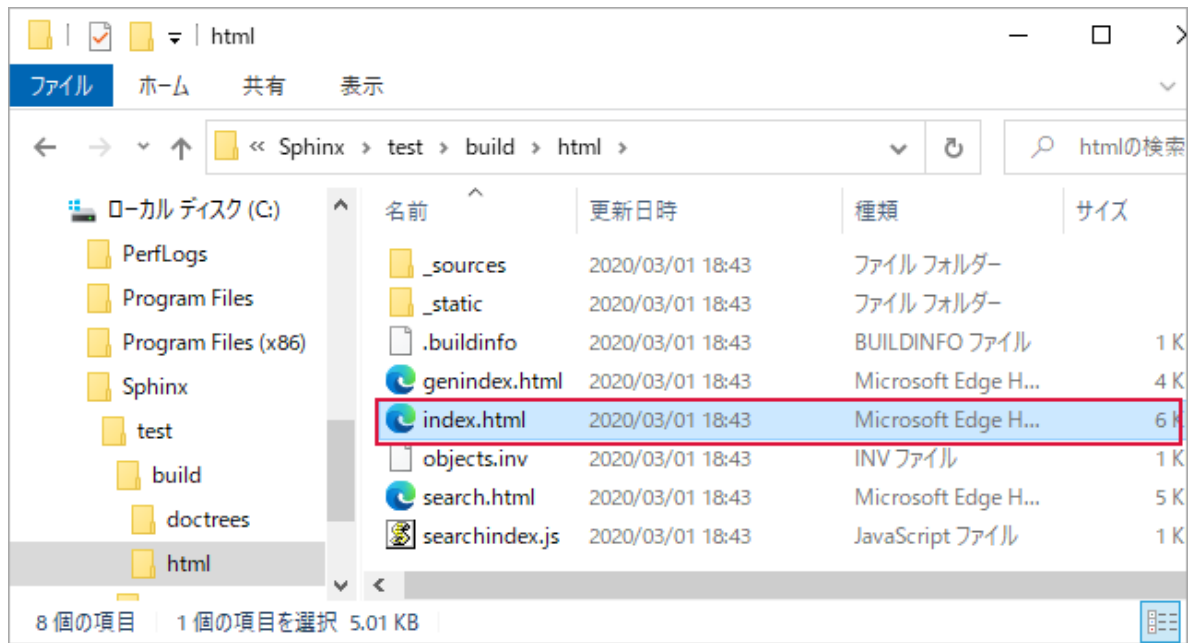
## 4. "make html" を実行して HTML を作り直し

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Sphinx\test> ./make html
Running Sphinx v2.4.3
loading translations [ja]... done
making output directory... done
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 1 source files that are out of date
updating environment: [new config] 1 added, 0 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] index
generating indices... genindexdone
writing additional pages... searchc:\users\user\miniconda3\lib\site-packages\sp
x30Warning: To modify script_files in the theme is deprecated. Please insert a <
[% endblock %]
done
copying static files... .. done
copying extra files... done
dumping search index in Japanese (code: ja)... done
dumping object inventory... done
build succeeded.

The HTML pages are in build\html.
(base) PS C:\Sphinx\test> _
```

5. "index.html" ファイルを開いて結果確認





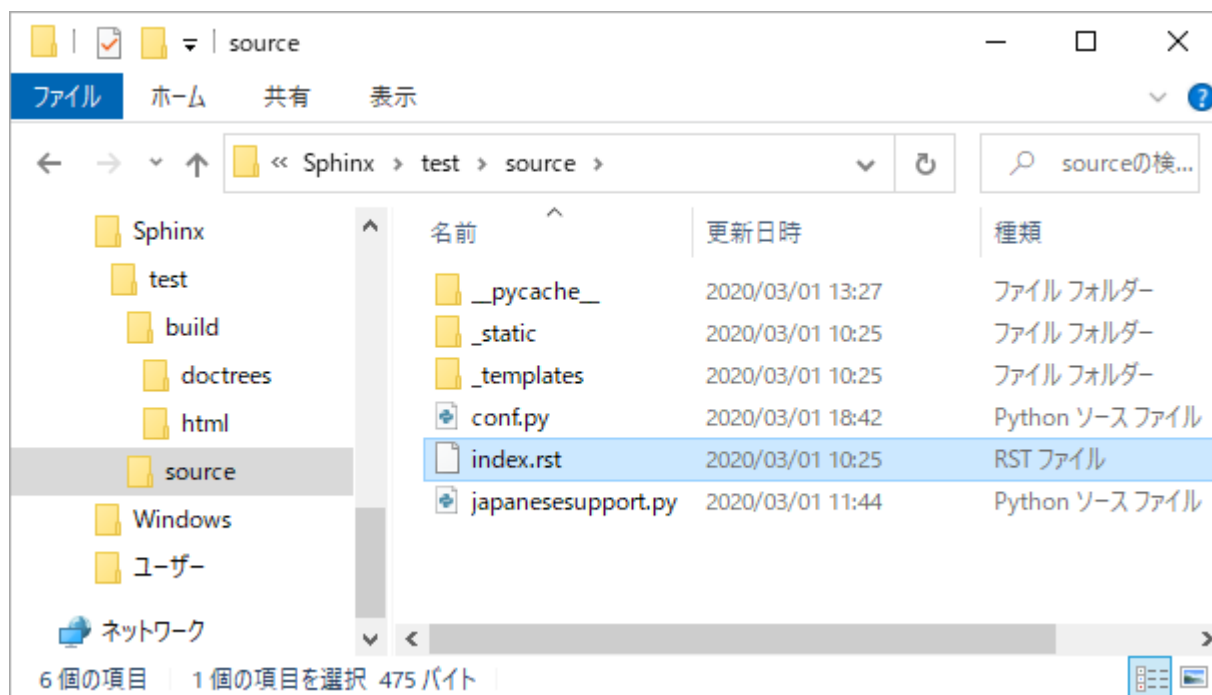
## 2.5 文書の構造

### 【トピックス】

- *Sphinx* の文書ファイル
  - 見出し
  - 目次と見出し
  - ".rst" ファイルの階層化 : *toctree* ディレクティブ
    - ドキュメントの分割
    - *toctree* ディレクティブ
    - 下位階層の ".rst" ファイルを下位フォルダーに配置した場合
    - *toctree* ディレクティブのオプション
  - ドキュメントの分割 : *include* ディレクティブ
- 

### 2.5.1 Sphinx の文書ファイル

- Sphinx は reStructuredText と呼ばれるマークアップ言語を使用して文書を書きます。
- reStructuredText で書いたファイル自体はテキストファイルですが、ファイル拡張子は `.rst` です。
- HTML に変換したとき、1 つの rst ファイルが 1 つの HTML ファイルになります。
- インデントは 3 文字単位です。タブ文字は使用できません。
- "index.rst" ファイルがすべての文書の起点です。すべての `.rst` ファイルは "index.rst" を起点に tree 状に結合されます。



## 2.5.2 見出し

### 見出しの構成

- 文書のくくりは大きな方から「章」→「セクション」→「サブセクション」→「サブサブセクション」→「パラグラフ」になります。
- それぞれのくくりの最初に記述する見出しは次のように記述します。HTML に変換したときは、それぞれ異なるサイズの文字に変換されます。
- 見出しに使用する \* や = など記号は見出しの文字列と同じまたはそれよりも長くなければなりません。

```
#####
部の見出し
#####
```

```
*****
章の見出し
*****
```

```
セクションの見出し
=====
```

```
サブセクションの見出し
-----
```

(次のページに続く)

(前のページからの続き)

## サブサブセクションの見出し

^ ^

## パラグラフの見出し

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

## 見出しの例

実際に "index.rst" ファイルを変更し、各見出しのサイズなどの見え方を確認します。 "sphinx-quickstart" コマンドで作成した "index.rst" ファイルの内容をすべて削除し、次のように変更しました。

#####

部の見出し

#####

部の紹介文。

\*\*\*\*\*

## 章の見出し

\*\*\*\*\*

章の紹介文。

## セクションの見出し

=====

セクションの紹介文。

## サブセクションの見出し

---

サブセクションの紹介文。

## サブサブセクションの見出し

[illegible]

サブサブセクションの紹介文。

## パラグラフの見出し

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044

パラグラフの紹介文。

HTML に変換した結果です。それぞれの見出しの文字サイズが少しずつ異なっています。

test document

Search docs

部の見出し

Docs » 部の見出し

## 部の見出し

部の紹介文。

## 章の見出し

章の紹介文。

## セクションの見出し

セクションの紹介文。

## サブセクションの見出し

サブセクションの紹介文。

## サブサブセクションの見出し

サブサブセクションの紹介文。

## パラグラフの見出し

パラグラフの紹介文。

© Copyright 2020, hoge

Built with Sphinx using a theme provided

### 2.5.3 目次と見出し

画面左側に表示されるドキュメントの目次は Sphinx が文書構造を元に自動的に作成します。例えば "index.rst" ファイルの内容が次のような場合の目次を確認します。

```
#####
第一部   〇〇〇
#####

*****
はじめに
*****
このドキュメントは〇〇〇の構築手順を説明したものです。

*****
環境構築
*****

ハードウェアスペック：最小
=====
CPU
-----
Intel Core i9-9900K

MEMORY
-----
8GB

HDD
-----
80GB

ハードウェアスペック：推奨
=====
CPU
-----
Intel Core i9-10980XE

MEMORY
-----
64GB

M2.SSD
-----
1TB
```

見出しを使用して自動的に目次を作成します。

test document

Search docs

第一部

はじめに

環境構築

ハードウェアスペック : 最小

CPU

MEMORY

HDD

ハードウェアスペック : 推奨

CPU

MEMORY

M2.SSD

Docs » 第一部

第一部

はじめに

このドキュメントはの構築手順を説明したものです。

環境構築

ハードウェアスペック : 最小

CPU

Intel Core i9-9900K

MEMORY

8GB

HDD

80GB

ハードウェアスペック : 推奨

CPU

Intel Core i9-10980XE

MEMORY

64GB

M2.SSD

1TB

© Copyright 2020, hoge

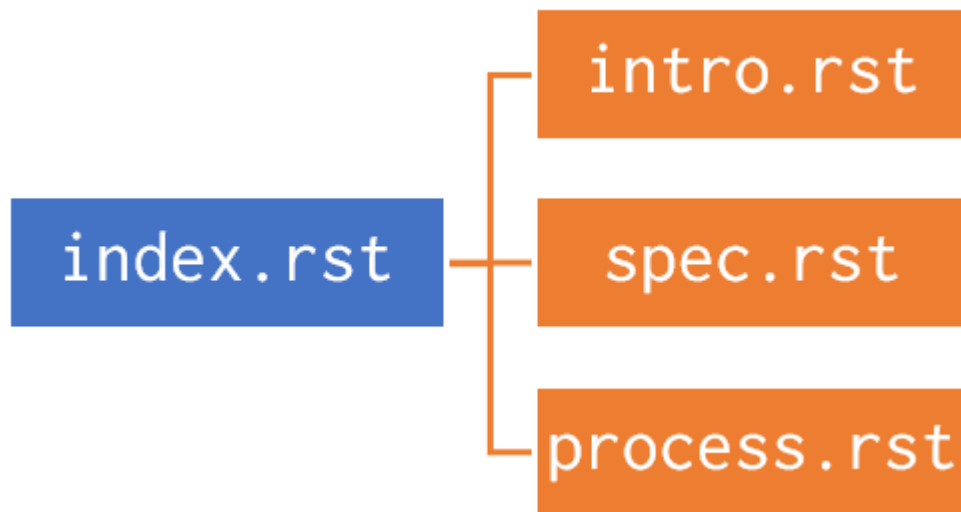
Built with Sphinx using a theme provided by Read the Docs.

## 2.5.4 ".rst" ファイルの階層化 : toctree ディレクティブ

### ドキュメントの分割

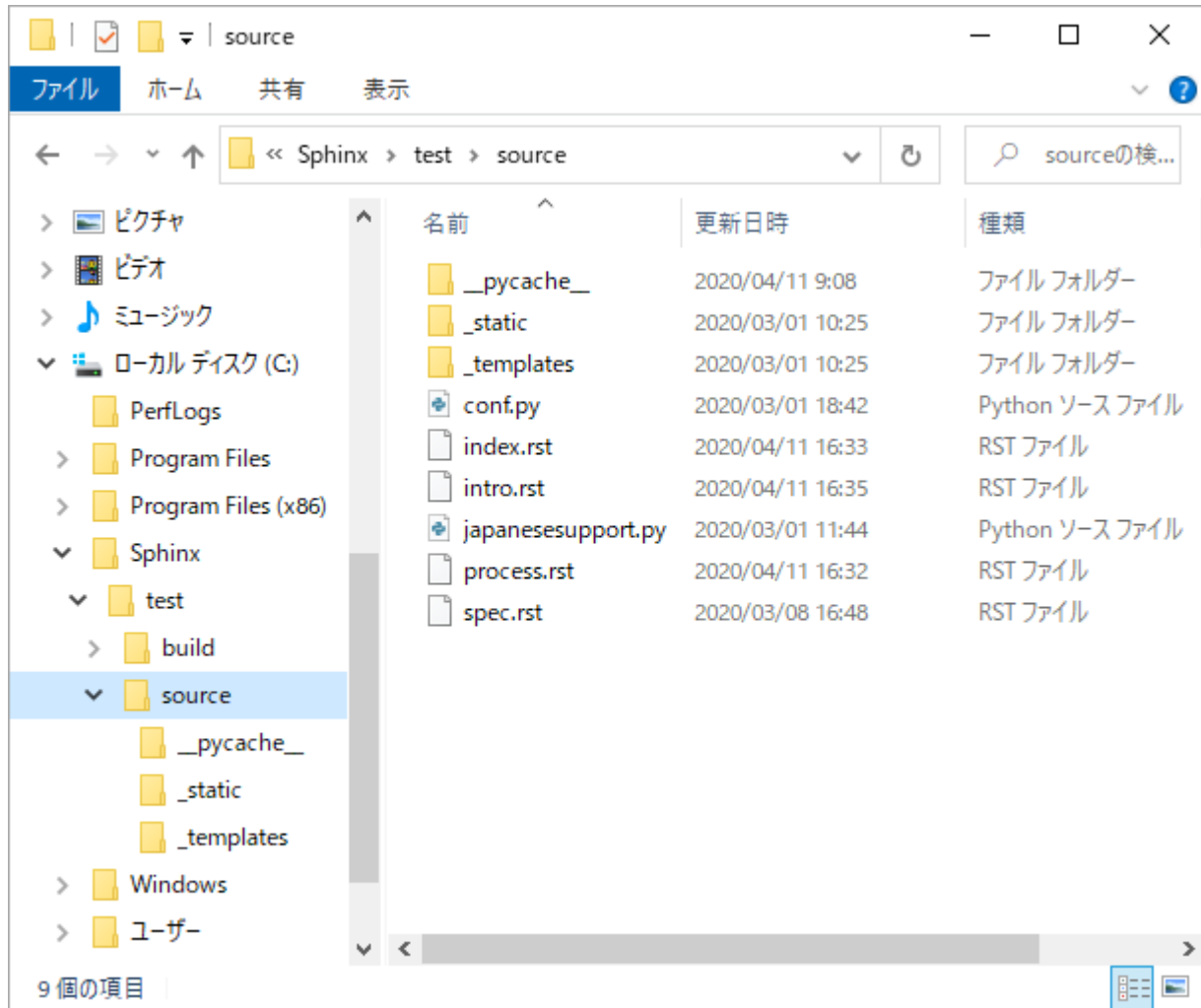
「目次と見出し」のように1つの `rst` ファイルに複数の「部」や「章」などを書いてもドキュメントは作成できます。しかし、この方法ではページのサイズが長くなり、視認性や検索性、メンテナンス性に劣るドキュメントになります。

この問題の解決方法は文書を「章」などのある程度のまとまりで分割し、そのまとまりごとに `rst` ファイルを作成することです。



今回のファイルの配置場所です。すべて "index.rst" ファイルと同じフォルダーに配置しました。





各ファイルの内容と表示内容です。

index.rst

```
#####
○○○構築手順書
#####
.. toctree::

    intro.rst
    ./spec
    process
```

test document

Search docs

はじめに  
構築対象機器の要件  
インストール手順

Docs » ○○○構築手順書

View page source

○○○構築手順書

- はじめに
  - ご挨拶
  - お問い合わせ先
- 構築対象機器の要件
  - 最小要件
  - 推奨要件
- インストール手順
  - 機器の準備
  - インストールメディアの準備

Next

© Copyright 2020, hoge  
Built with Sphinx using a theme provided by Read the Docs.

intro.rst

```
*****
はじめに
*****

ご挨拶
=====

この度は弊社製品である ○○○ をご購入くださりありがとうございます。
○○○ は皆様の作業効率および生産性向上を目的に開発しました。不具合が
あれば、下記までご連絡ください。

お問い合わせ先
=====
電話   +81-12-3456-7890
```

test document

Search docs

はじめに
ご挨拶
お問い合わせ先
構築対象機器の要件
インストール手順

Docs » はじめに
[View page source](#)

# はじめに

## ご挨拶

この度は弊社製品である ○○○ をご購入くださりありがとうございます。○○○ は皆様の作業効率および生産性向上を目的に開発しました。不具合があれば、下記までご連絡ください。

## お問い合わせ先

電話 +81-12-3456-7890

Previous
Next

© Copyright 2020, hoge

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

spec.rst

```

*****
構築対象機器の要件
*****
○○○ をインストールする機器の要件を説明します。

最小要件
=====
    の要件を満たしていること。

推奨要件
=====
    の要件を満たしていること。

```

test document

Search docs

はじめに

構築対象機器の要件

最小要件

推奨要件

インストール手順

Docs » 構築対象機器の要件

View page source

---

## 構築対象機器の要件

〇〇〇 をインストールする機器の要件を説明します。

### 最小要件

□□□ の要件を満たしていること。

### 推奨要件

△△△ の要件を満たしていること。

Previous

Next

---

© Copyright 2020, hoge

Built with Sphinx using a theme provided by Read the Docs.

process.rst

```
*****
インストール手順
*****
〇〇〇 のインストール手順を説明します。

機器の準備
=====
「最小要件」以上の危機を準備、電源を ON にしてください。

インストールメディアの準備
=====
インストールメディアは弊社サポートサイトからダウンロードできます。
サイトの URL はユーザー登録時に弊社から送信したメールに記載されて
います。
```

test document

Search docs

はじめに

構築対象機器の要件

インストール手順

機器の準備

インストールメディアの準備

Docs » インストール手順

View page source

## インストール手順

〇〇〇 のインストール手順を説明します。

### 機器の準備

「最小要件」以上の危機を準備、電源を ON にしてください。

### インストールメディアの準備

インストールメディアは弊社サポートサイトからダウンロードできます。サイトの URL はユーザー登録時に弊社から送信したメールに記載されています。

Previous

© Copyright 2020, hoge

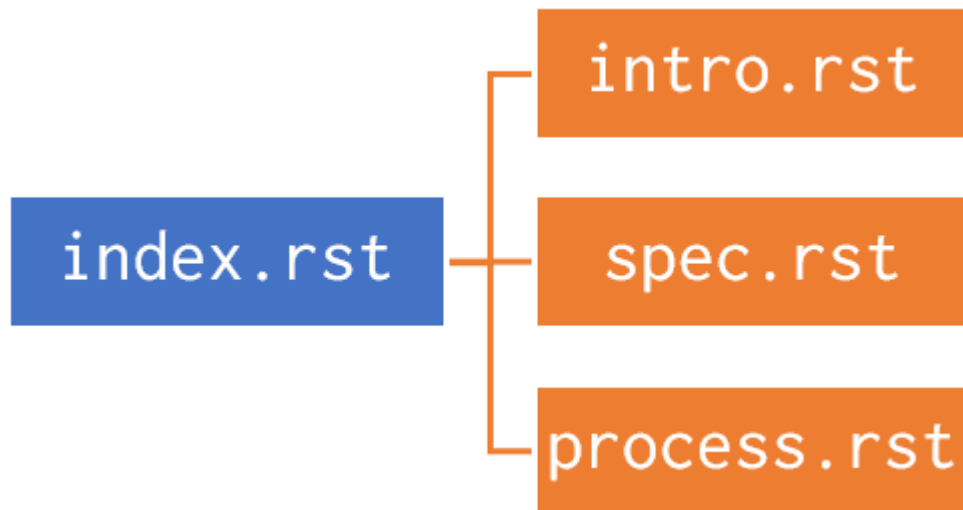
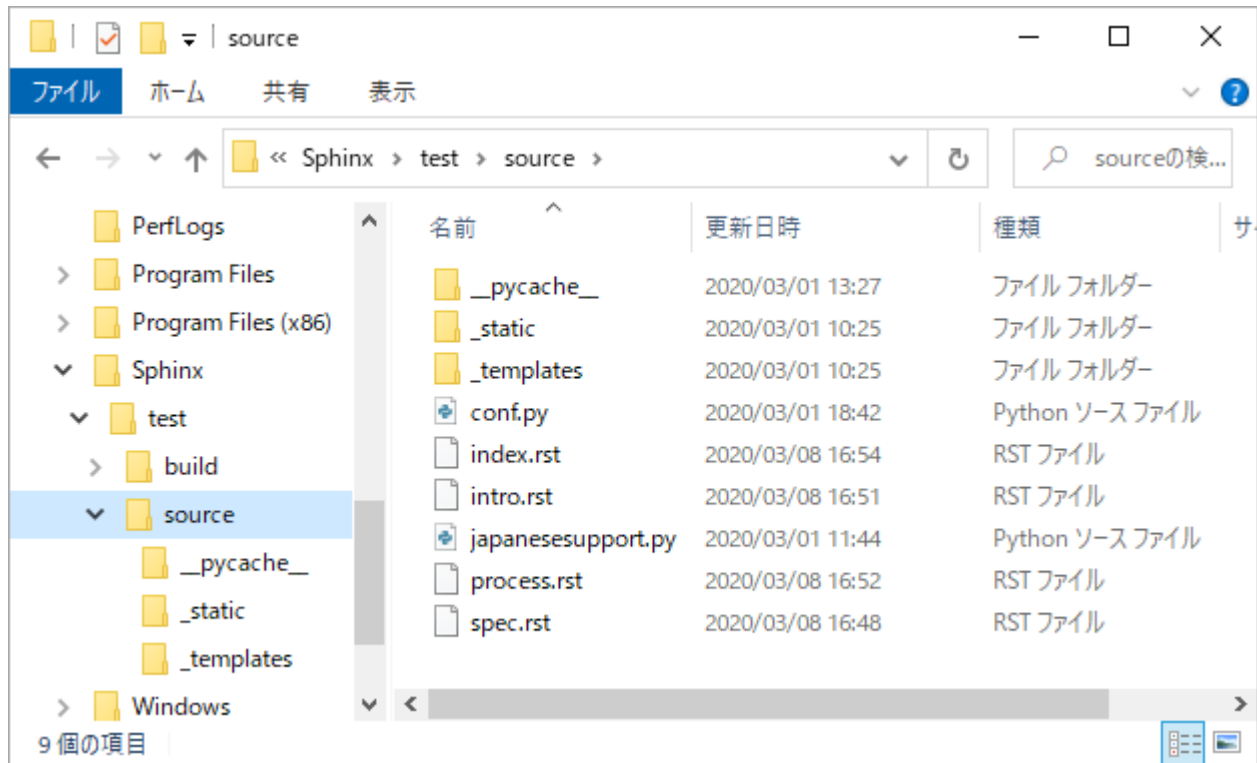
Built with Sphinx using a theme provided by Read the Docs.

## toctree ディレクティブ

- `rst` ファイル間の階層構造は `toctree` ディレクティブで定義します。
- `toctree` ディレクティブは上位階層の `rst` ファイルに指定します。
- 上位階層と下位階層の `rst` ファイルの設置場所は同じフォルダ - でも、異なるフォルダーでも問題ありません。
- 下位階層の `rst` ファイルの指定は上位階層の `rst` ファイルを起点とした相対指定で行います。

**重要:** "index.rst" ファイル以外の `rst` ファイルは、必ず `toctree` ディレクティブで指定されなければなりません。

ファイルの配置位置と `rst` ファイル間の関係です。



"index.rst" ファイルの内容です。

```
#####
〇〇〇構築手順書
#####
.. toctree::

    intro.rst
    ./spec
    process
```

- HTML に変換すると、下位階層の `rst` ファイルの見出し部分を使用した目次が作成されます。目次にはすべての見出しが展開・表示されます。
- 下位階層の `rst` ファイルの指定方法は拡張子付き、相対パスであることを示す、ファイル名だけのいずれの形式でも問題ありません。



下位階層の "**rst**" ファイルを下位フォルダーに配置した場合

下図のように下位階層の `rst` ファイルを下位階層のディレクトリーに配置すると、階層構造とファイルの配置が一致し、ドキュメントの流れを追うのが楽になります。

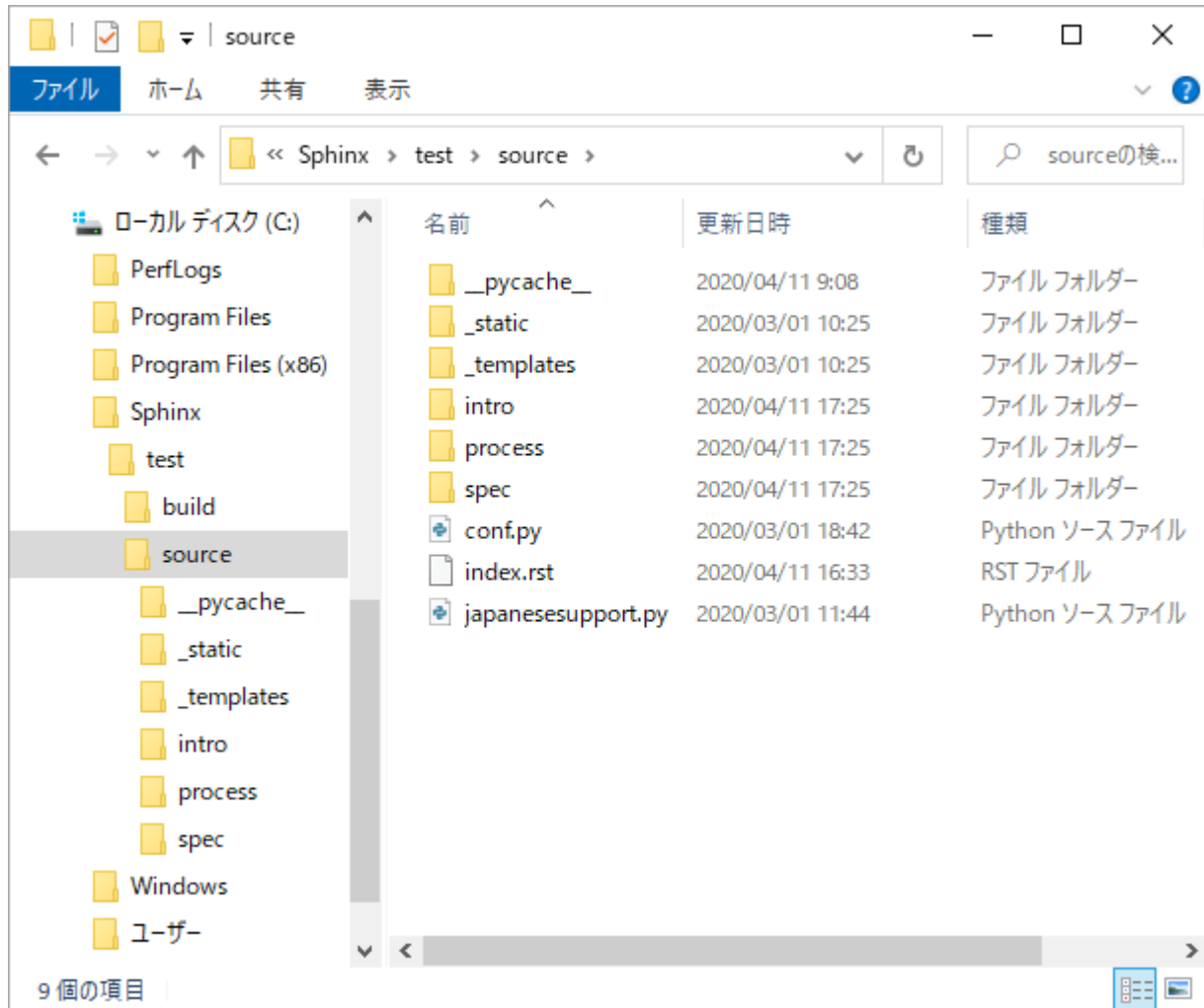
```
PS C:\Sphinx\test\source> tree /A /F
フォルダー バスの一覧
ボリューム シリアル番号は AEB0-CDA6 です
C:.
|   conf.py
|   index.rst
|   japanesesupport.py
|
+---intro
|       intro.rst
|
+---process
|       process.rst
|
+---spec
|       spec.rst
|
```

(次のページに続く)

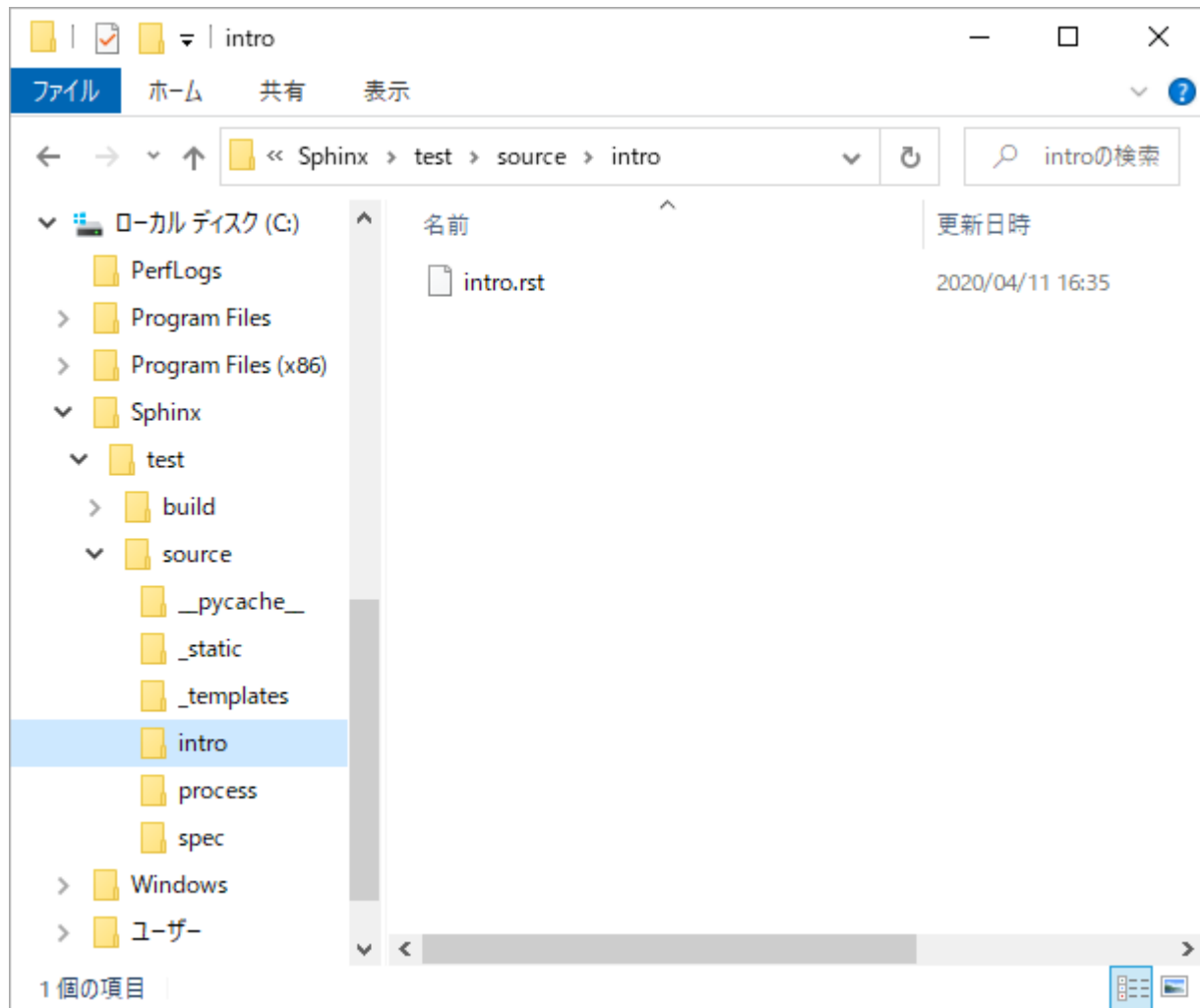
(前のページからの続き)

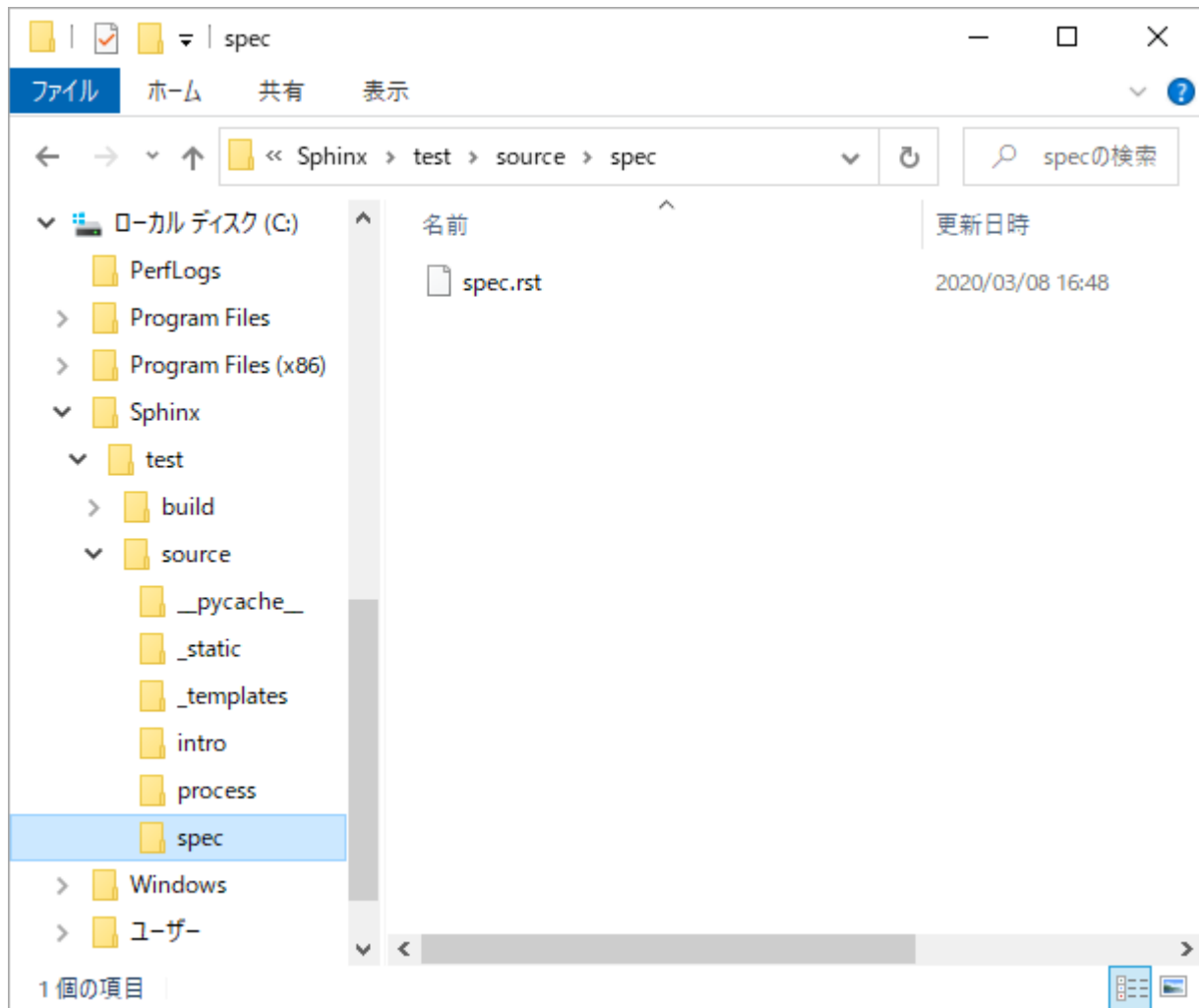
```
+---_static
+---_templates
\---__pycache__
      japanesesupport.cpython-37.pyc
```

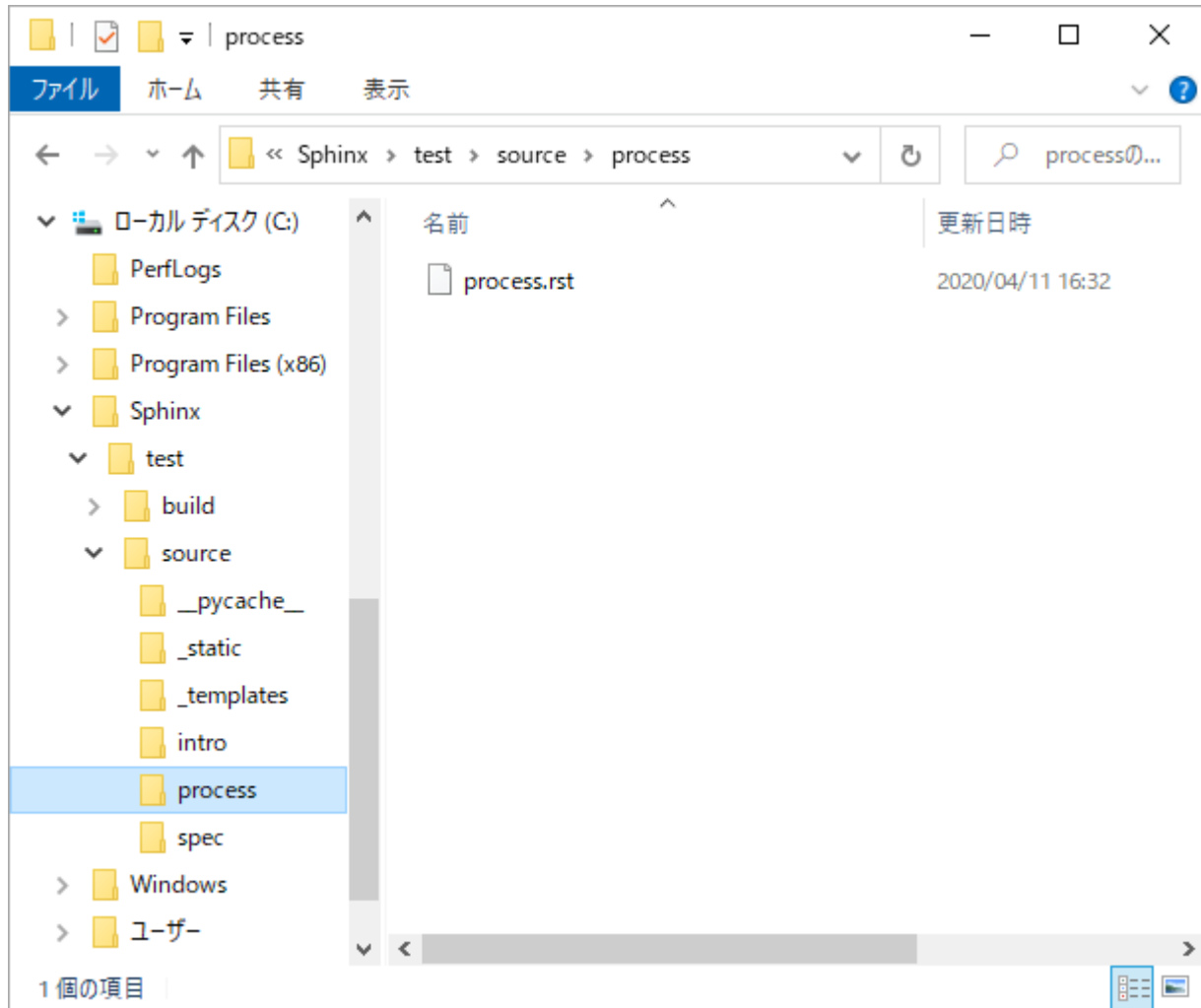
```
PS C:\Sphinx\test\source>
```











フォルダ構成を反映した "index.rst" ファイルの内容です。

```
*****
〇〇〇構築手順書
*****
.. toctree::

    intro/intro
    spec/spec
    process/process
```

## toctree ディレクティブのオプション

### hidden オプション

目次を表示しません。

```
#####
〇〇〇構築手順書
#####
.. toctree::
    :hidden:

    intro/intro
    spec/spec
    process/process
```



### numbered オプション

下位階層の rst ファイルの見出しに自動で通し番号を付けます。

```
#####
〇〇〇構築手順書
#####
.. toctree::
    :numbered:

    intro/intro
    spec/spec
    process/process
```

test document
Search docs

1. はじめに  
2. 構築対象機器の要件  
3. インストール手順

Docs » ○○○構築手順書
View page source

## ○○○構築手順書

- 1. はじめに
  - 1.1. ご挨拶
  - 1.2. お問い合わせ先
- 2. 構築対象機器の要件
  - 2.1. 最小要件
  - 2.2. 推奨要件
- 3. インストール手順
  - 3.1. 機器の準備
  - 3.2. インストールメディアの準備

Next

© Copyright 2020, hoge

Built with Sphinx using a theme provided by Read the Docs.

test document
Search docs

1. はじめに  
1.1. ご挨拶  
1.2. お問い合わせ先  
2. 構築対象機器の要件  
3. インストール手順

Docs » 1. はじめに
View page source

## 1. はじめに

### 1.1. ご挨拶

この度は弊社製品である ○○○ をご購入くださりありがとうございます。○○○ は皆様の作業効率および生産性向上を目的に開発しました。不具合があれば、下記までご連絡ください。

### 1.2. お問い合わせ先

電話 +81-12-3456-7890

Previous
Next

© Copyright 2020, hoge

Built with Sphinx using a theme provided by Read the Docs.

test document
Search docs

1. はじめに
2. 構築対象機器の要件
2.1. 最小要件
2.2. 推奨要件
3. インストール手順

Docs » 2. 構築対象機器の要件
View page source

## 2. 構築対象機器の要件

〇〇〇 をインストールする機器の要件を説明します。

### 2.1. 最小要件

□□□ の要件を満たしていること。

### 2.2. 推奨要件

△△△ の要件を満たしていること。

Previous
Next

© Copyright 2020, hoge  
Built with Sphinx using a theme provided by Read the Docs.

test document
Search docs

1. はじめに
2. 構築対象機器の要件
3. インストール手順
3.1. 機器の準備
3.2. インストールメディアの準備

Docs » 3. インストール手順
View page source

## 3. インストール手順

〇〇〇 のインストール手順を説明します。

### 3.1. 機器の準備

「最小要件」以上の危機を準備、電源を ON にしてください。

### 3.2. インストールメディアの準備

インストールメディアは弊社サポートサイトからダウンロードできます。サイトの URL はユーザー登録時に弊社から送信したメールに記載されています。

Previous

© Copyright 2020, hoge  
Built with Sphinx using a theme provided by Read the Docs.

## maxdepth オプション

目次として展開する見出しのレベルを指定します。

```
#####
〇〇〇構築手順書
#####
.. toctree::
    :maxdepth: 1

    intro/intro
    spec/spec
    process/process
```



numbered オプションと併用した場合です。

```
#####
〇〇〇構築手順書
#####
.. toctree::
    :maxdepth: 1
    :numbered:

    intro/intro
    spec/spec
    process/process
```



## 2.5.5 ドキュメントの分割 : `include` ディレクティブ

`include` ディレクティブは指定したファイルの内容を展開します。この機能を使用して「章」単位に分割した `rst` ファイルを「セクション」単位に分割します。

分割前の "process.rst" の内容です。

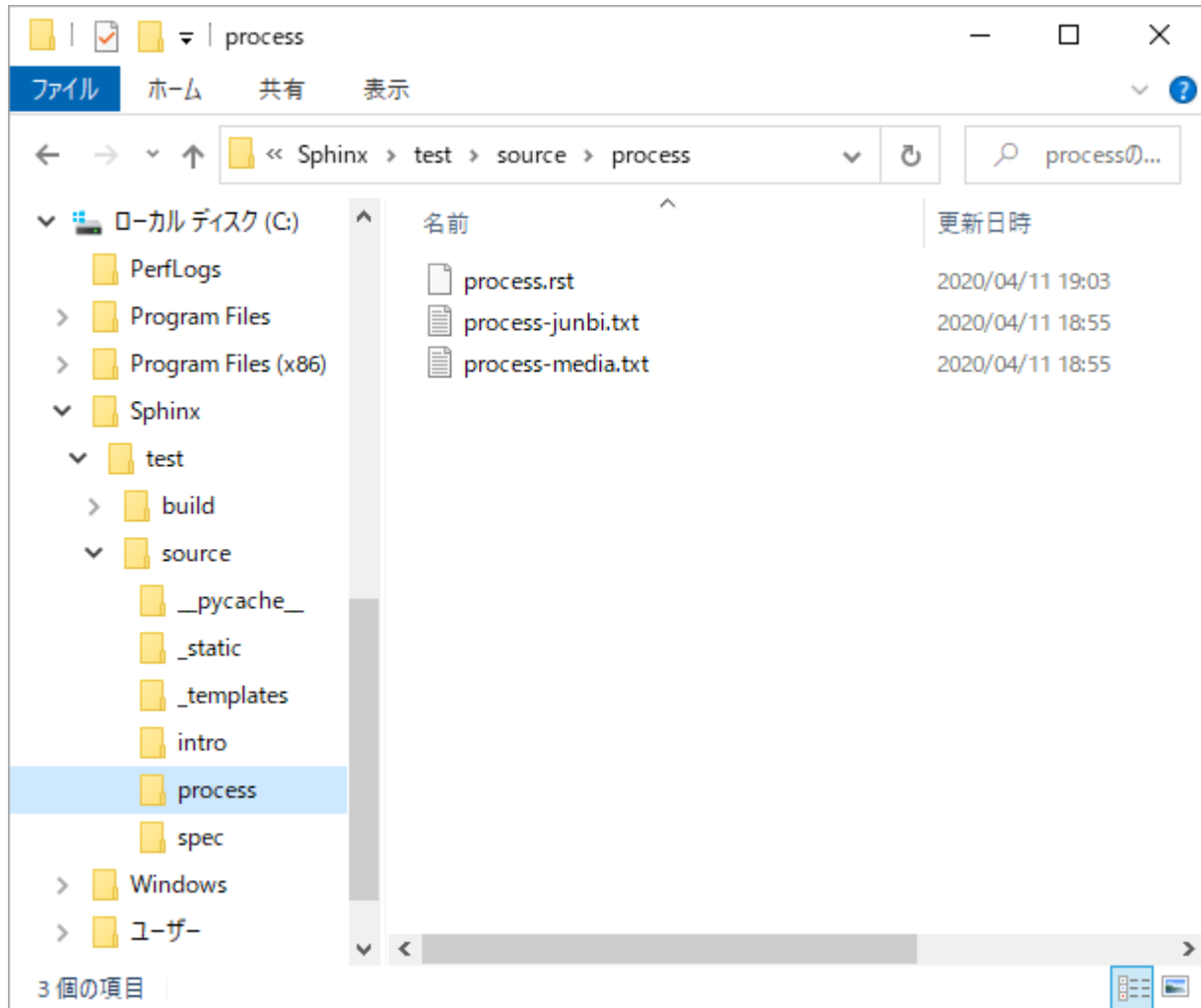
```
*****
インストール手順
*****
〇〇〇 のインストール手順を説明します。

機器の準備
=====
「最小要件」以上の危機を準備、電源を ON にしてください。

インストールメディアの準備
=====
インストールメディアは弊社サポートサイトからダウンロードできます。
サイトの URL はユーザー登録時に弊社から送信したメールに記載されて
います。
```

"process.rst" の分割後の process フォルダの状態です。 `rst` ファイル以外に `txt` ファイルが 2 つ増えています。 "process.rst" ファイルを分割した（切り出した）内容が、この `txt` ファイルに記録されています。





分割後の "process.rst" ファイルの内容です。

```
*****
インストール手順
*****
〇〇〇 のインストール手順を説明します。

.. include:: process-junbi.txt
.. include:: process-media.txt
```

"process-junbi.txt" ファイルの内容です。

```
機種の準備
=====
「最小要件」以上の危機を準備、電源を ON にしてください。
```

"process-media.txt" ファイルの内容です。

#### インストールメディアの準備

インストールメディアは弊社サポートサイトからダウンロードできます。  
サイトの URL はユーザー登録時に弊社から送信したメールに記載されています。

## 2.6 いろいろな表現

### 2.6.1 文字の表現

Sphinx で使用できる文字の表現のサンプルです。「説明 → コードのサンプル → コードの実行結果」の順に説明します。

#### 【トピックス】

- ・ 強調 (イタリック)
- ・ 強い強調 (太字)
- ・ コードサンプル
- ・ フィールドリスト
- ・ “\” の表記
- ・ ラインブロック
- ・ リスト (箇条書き)
- ・ リテラルコードブロック

---

#### 強調 (イタリック)

- ・ 文字列を "\*" ではさむと強調 (イタリック) 表示になります。
- ・ Mozilla Firefox は強調表示になりますが、Google Chrome や Microsoft Edge は通常表示と変わらない表示です。

このように \*強調したい文字列\* を "\*" ではさみます。

このように 強調したい文字列 を "\*" ではさみます。

---

### 強い強調 (太字)

- 文字列を `"**"` ではさむと強い強調 (太字) 表示になります。

このように `**強く強調したい文字列**` を `"**"` ではさみます。

このように 強く強調したい文字列 を `"**"` ではさみます。

### コードサンプル

- 文字列を `"``"` ではさむとコードサンプル表示になります。

```コードサンプル表示したい文字列``` を `"``"` ではさみます。

コードサンプル表示したい文字列 を `"``"` ではさみます。

### フィールドリスト

- 文字列を `:"` ではさむとフィールドリストになります。

```
:1 つめのフィールドリスト: 説明文 その1
:2 つめのフィールドリスト: 説明文 その2
:3 つめのフィールドリスト: 説明文 その3
```

1 つめのフィールドリスト 説明文 その1

2 つめのフィールドリスト 説明文 その2

3 つめのフィールドリスト 説明文 その3

- フィールドリスト表示をしたい行の次の行にインデントを付けると同様の効果が得られます。

```
1 つめのフィールドリスト
  説明文 その1
2 つめのフィールドリスト
  説明文 その2
3 つめのフィールドリスト
  説明文 その3
```

1 つめのフィールドリスト 説明文 その1

2 つめのフィールドリスト 説明文 その2

3 つめのフィールドリスト 説明文 その3

## "\" の表記

- "\" を書くときは "\\" と書きます。

```
"c:\\windows" ディレクトリーです。
```

"c:\windows" ディレクトリーです。

---

## ラインブロック

- 行の先頭に "|" を書くと、その行はラインブロックになります。
- ラインブロックはソースコードの改行がそのまま反映されます。

```
| このように先頭に "|" を書くと  
| 書いたとおりに  
| 改行します。
```

このように先頭に "|" を書くと  
書いたとおりに  
改行します。

```
ラインブロックの指定をしないときは  
改行位置は Sphinx まかせに  
なります。
```

ラインブロックの指定をしないときは改行位置は Sphinx まかせになります。

---

## リスト (箇条書き)

### 一般的な箇条書き

- 行の先頭に "-" を書くと、その行は箇条書きになります。

```
- 箇条書きの 1 つ目です。  
- 箇条書きは先頭に "-" がつきます。  
- 箇条書きでも  
  改行は Sphinx 任せです。
```

- 箇条書きの 1 つ目です。
- 箇条書きは先頭に "・" がつきます。
- 箇条書きでも改行は Sphinx 任せです。

#### 番号付きの箇条書き

- 行の先頭に "#." を書くと、その行は番号付きの箇条書きになります。

```
#. 箇条書きの 1 つ目です。
#. 箇条書きは先頭に開始値が 1 の番号が付きます。
#. 番号付き箇条書きでも、やっぱり
   改行は Sphinx 任せです。
```

1. 箇条書きの 1 つ目です。
2. 箇条書きは先頭に開始値が 1 の番号が付きます。
3. 番号付き箇条書きでも、やっぱり改行は Sphinx 任せです。

#### 初期値あり番号付きの箇条書き

- 番号付きの箇条書きは初期値を指定できます。

```
8. 箇条書きの 1 つ目です。
#. 箇条書きは先頭に開始値が 1 の番号が付きます。
#. 初期値ありの番号付き箇条書きでも、やっぱり
   改行は Sphinx 任せです。
```

8. 箇条書きの 1 つ目です。
9. 箇条書きは先頭に開始値が 1 の番号が付きます。
10. 初期値ありの番号付き箇条書きでも、やっぱり改行は Sphinx 任せです。

#### リストのネスト

- リストはネストできます。
- 親リストと子リストの間に 1 行の空行が必要です。

```
- 親リストの 1 つめ

  - 子リストの 1 つめ
  - 子リストの 2 つめ
```

(次のページに続く)

(前のページからの続き)

<ul style="list-style-type: none"><li>- 子リストの 3 つめ</li></ul> <ul style="list-style-type: none"><li>- 親リストの 2 つめ</li></ul> <ul style="list-style-type: none"><li>#. 番号付き親リストの 1 つめ<ul style="list-style-type: none"><li>#. 番号付き子リストの 1 つめ</li><li>#. 番号付き子リストの 2 つめ</li><li>#. 番号付き子リストの 3 つめ</li></ul></li><li>#. 番号付き親リストの 2 つめ</li></ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 親リストの 1 つめ
  - 子リストの 1 つめ
  - 子リストの 2 つめ
  - 子リストの 3 つめ

- 親リストの 2 つめ
1. 番号付き親リストの 1 つめ
    1. 番号付き子リストの 1 つめ
    2. 番号付き子リストの 2 つめ
    3. 番号付き子リストの 3 つめ
  2. 番号付き親リストの 2 つめ

## その他の箇条書き

<ul style="list-style-type: none"><li>a. 番号の代わりに英小文字を使用した箇条書きの 1 つ目です。</li><li>#. 箇条書きは番号は a. b. c. . . . になります。</li><li>#. 改行は<ul style="list-style-type: none"><li>やっぱり Sphinx 任せです。</li></ul></li></ul> <ul style="list-style-type: none"><li>A. 番号の代わりに英大文字を使用した箇条書きの 1 つ目です。</li><li>#. 箇条書きは番号は A. B. C. . . . になります。</li><li>#. 改行は<ul style="list-style-type: none"><li>やっぱり Sphinx 任せです。</li></ul></li></ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- a. 番号の代わりに英小文字を使用した箇条書きの 1 つ目です。
- b. 箇条書きは番号は a. b. c. . . . になります。

- c. 改行はやっぱり Sphinx 任せです。
- A. 番号の代わりに英大文字を使用した箇条書きの 1 つ目です。
- B. 箇条書きは番号は A. B. C. . . . になります。
- C. 改行はやっぱり Sphinx 任せです。

## リテラルコードブロック

- 行の最後に ":" = 半角空白 1 桁 + ":" を付けると、次の行からはリテラルコードブロックと解釈されます。
- リテラルコードブロック部分の前後に 1 行の空行が必要です。
- リテラルコードブロック部分はインデントして記入します。
- リテラルコードブロック部分に記述した内容は、そのまま表示されます。

ここは通常の文章です。次の行はリテラルコードブロックです。 ::

ここからリテラルコードブロックです。  
リテラルコードブロック部分の改行は、  
ソースコードの内容がそのまま反映されます。  
入力した文字はそのまま表示されます。箇条書きをしようとしても

- あああ
- いいい

のようにそのまま表示されます。  
ここでリテラルコードブロックは終了です。

ここから通常の文章です。

ここは通常の文章です。次の行はリテラルコードブロックです。

ここからリテラルコードブロックです。  
リテラルコードブロック部分の改行は、  
ソースコードの内容がそのまま反映されます。  
入力した文字はそのまま表示されます。箇条書きをしようとしても

- あああ
- いいい

のようにそのまま表示されます。  
ここでリテラルコードブロックは終了です。

ここから通常の文章です。

## 2.6.2 ロール

ロールを使用し、特定の意味を持たせた文字列を強調して表示します。

### 【トピックス】

- ・ ボタンなど : *guilabel*
- ・ コマンド : *command*
- ・ 言葉の短縮形 : *abbr*
- ・ ファイルやディレクトリ : *file*
- ・ ファイルのダウンロード : *download*
- ・ メニューの遷移 : *menuselection*
- ・ キーボード : *kbd*
- ・ 数式 : *math*

---

### ボタンなど : *guilabel*

- ・ 一般にボタンやウィンドウのタイトル、フィールド名、メニュー、メニューの項目名、リスト中の選択された値などのインタフェース上に表示されるラベルに使用します。

〇〇〇 を入力後 `:guilabel:`OK`` をクリックします。

- `:guilabel:`list`` ボタン : 登録されている内容を一覧表示します。
- `:guilabel:`search`` ボタン : 入力した文字列をキーワードにしてドキュメント内を検索します。

〇〇〇 を入力後 *OK* をクリックします。

- ・ *list* ボタン : 登録されている内容を一覧表示します。
  - ・ *search* ボタン : 入力した文字列をキーワードにしてドキュメント内を検索します。
-



#### コマンド : **command**

- OS のコマンドなどに使用します。

ディレクトリ内のファイルやフォルダーの一覧を確認するときは `:command:`ls`` コマンドを使用します。

ディレクトリ内のファイルやフォルダーの一覧を確認するときは `ls` コマンドを使用します。

---

#### 言葉の短縮形 : **abbr**

- 言葉の短縮形の表記に使用します。

スタックは `:abbr:`LIFO (last-in, first-out)`` 構造です。

スタックは LIFO (last-in, first-out) 構造です。

---

#### ファイルやディレクトリ : **file**

- ファイルやディレクトリの名前に使用します。

nginx のメインの設定ファイルは `:file:`/etc/nginx/nginx.conf`` です。

nginx のメインの設定ファイルは `/etc/nginx/nginx.conf` です。

---

#### ファイルのダウンロード : **download**

- ファイルへのリンクを張って、ファイルをダウンロードできるようにするときに使用します。

サンプルのテキストファイルをダウンロードするには `:download:`ここをクリック <./sample.txt>`` します。

サンプルのテキストファイルをダウンロードするには `ここをクリック` します。

---

### メニューの遷移 : `menuselection`

- メニューの操作手順を示すときに使用します。

新しくテキストファイルを作成するには " `:menuselection:`ファイル (F) --> 新規作成 (N)`` " の順に操作します。

新しくテキストファイルを作成するには " ファイル (*F*) → 新規作成 (*N*) " の順に操作します。

---

### キーボード : `kbd`

- キーボード操作のキーに使用します。

処理を中断するには `:kbd:`Esc`` を押します。

処理を中断するには `Esc` を押します。

---

### 数式 : `math`

- 数式を表現できます。

二次方程式の一般形は 「 `:math:`ax^2 + bx + c = 0`` 」 です。

二次方程式の一般形は 「  $ax^2 + bx + c = 0$  」 です。

## 2.6.3 イメージ : `image` ディレクティブ

- `image` ディレクティブを使用し、ドキュメントに画像を埋め込みます。
- `scale` オプションで倍率を指定します。
- 取り込む画像の横幅が表示域の横幅より大きいときは縮小表示になります。

```
.. image:: img/food_konbini_onigiri.png
```



- `scale` オプションを指定して拡大・縮小表示できます。
- 拡大・縮小は元画像に対する割合で表示します。
- 縮小・拡大表示した画像をクリックすると別ウィンドウで原寸の画像を表示します。

```
.. image:: img/food_konbini_onigiri.png
   :scale: 50%
```



#### 2.6.4 リンク

##### 【トピックス】

- ・ *URL*
- ・ *rst* ファイル
- ・ ラベル

## URL

- ドキュメント内に URL を記述すると、そのままリンクになります。

Sphinx 日本ユーザー会のサイトは <https://sphinx-users.jp/index.html> です。

Sphinx 日本ユーザー会のサイトは <https://sphinx-users.jp/index.html> です。

- URL 部分をリンクテキストに変更できます。

Sphinx 日本ユーザー会のサイトは [`ここをクリック`](https://sphinx-users.jp/index.html) [<https://sphinx-users.jp/index.html>](https://sphinx-users.jp/index.html) [\\_](https://sphinx-users.jp/index.html) します。

Sphinx 日本ユーザー会のサイトは [ここをクリック](https://sphinx-users.jp/index.html) します。

## rst ファイル

- doc を使用して rst ファイルにリンクを設定します。

Sphinx の文書構造の説明は「`:doc:`../struct/struct``」で確認してください。

Sphinx の文書構造の説明は「[文書の構造](#)」で確認してください。

- リンク先の rst ファイルを `<と>` でくくって指定すると、表示する文字列を任意のものに置き換えられます。

Sphinx の文書構造の説明は `:doc:`こちらを確認 <../struct/struct>`` してください。

Sphinx の文書構造の説明は [こちらを確認](#) してください。

## ラベル

- 文書内にラベルを埋め込み、そのラベルに対してリンクを設定します。
- ラベルは Sphinx の文書内でユニークでなければなりません。
- ラベルは見出しとセットで設定します。

リンク先のドキュメントです。見出しの前にラベルを設定します。

```
.. _struct-toctree-split:
```

ドキュメントの分割

(次のページに続く)

(前のページからの続き)

-----  
↪ -----  
「 :ref:`struct-contents` 」のように 1 つの ``rst`` ファイルに複数の「部」や「章」などを書いてもドキュメントは作成できます。しかし、この方法ではページのサイズが長くなり、視認性や検索性、メンテナンス性に劣るドキュメントになります。

- `ref` を使用してラベルにリンクを設定します。

ドキュメントを分割するときの考え方は「 :ref:`struct-toctree-split` 」を参照してください。

ドキュメントを分割するときの考え方は「 [ドキュメントの分割](#) 」を参照してください。

- リンク先のラベルを < と > でくくって指定すると、表示する文字列を任意のものに置き換えられます。

ドキュメントを分割するときの考え方は `:ref:`こちらを参照<struct-toctree-split>`` してください。

ドキュメントを分割するときの考え方は [こちらを参照](#) してください。

## 2.6.5 サンプルコードの表示

### 【トピックス】

- コードブロック : `code-block` ディレクティブ
- ファイルから表示するコードの読み込み : `literalinclude` ディレクティブ
- コード表示のオプション

---

### コードブロック : `code-block` ディレクティブ

- `code-block` を使用してコードを表示します。
- コードの言語を指定すると、その言語に適したハイライト表示をします。
- 単なるテキスト表示するときは言語に `none` を指定します。
- 言語を省略すると Sphinx が適切な言語を選択してコードをハイライト表示します。

## 表示例

- 言語指定あり : java

```
.. code-block:: java

public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

- 言語指定あり : none (ハイライト表示なし)

```
.. code-block:: none

public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

- 言語指定なし : Sphinx が言語を自動判断

```
.. code-block::

public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

```
public class HelloWorld{
    public static void main(String[] args){
```

(次のページに続く)

(前のページからの続き)

```
    System.out.println("hello, world");
  }
}
```

### ファイルから表示するコードの読み込み : `literalinclude` ディレクティブ

- `literalinclude` を使用してコードをファイルから読み込み、表示します。
- `language` オプションでコードの言語を指定すると、その言語に適したハイライト表示をします。
- 単なるテキスト表示するときは言語に `none` を指定します。
- 言語を省略すると Sphinx が適切な言語を選択してコードをハイライト表示します。

#### 表示例

- 言語指定あり : java

```
.. literalinclude:: ./helloworld.java
   :language: java
```

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

- 言語指定あり : none (ハイライト表示なし)

```
.. literalinclude:: ./helloworld.java
   :language: none
```

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

- 言語指定なし : Sphinx が言語を自動判断

```
.. literalinclude:: ./helloworld.java
```



```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

## コード表示のオプション

### 行番号表示

- `linenos` オプションで行番号を表示します。

```
.. code-block:: java
:linenos:

public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

```
1 public class HelloWorld{
2     public static void main(String[] args){
3         System.out.println("hello, world");
4     }
5 }
```

```
.. literalinclude:: ./helloworld.java
:language: java
:linenos:
```

```
1 public class HelloWorld{
2     public static void main(String[] args){
3         System.out.println("hello, world");
4     }
5 }
```

### 特定の行を網掛け表示

- `emphasize-lines` オプションで指定した行をハイライト表示します。
- `" , "` で区切って複数の行を指定します。 例 : 1 行目と 10 行目を網掛け表示 → `1, 10`
- `" - "` で連続する行を指定します。 例 : 1 行目から 10 行目を網掛け表示 → `1-10`

```
.. code-block:: java
   :emphasize-lines: 2, 4

   public class HelloWorld{
       public static void main(String[] args){
           System.out.println("hello, world");
       }
   }
```

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

```
.. literalinclude:: ./helloworld.java
   :language: java
   :emphasize-lines: 2, 4
```

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello, world");
    }
}
```

## 2.6.6 テーブル ( 表 ) : `table` ディレクティブ

### 【トピックス】

- グリッドテーブル
- シンプルテーブル
- CSV テーブル
- リストテーブル
- 複雑なテーブル
- 列幅の変更
- ヘッダーの指定

- ・ テーブル名の追加

### グリッドテーブル

```
+-----+-----+-----+
| A      | B      | A and B |
+-----+-----+-----+
| False  | False  | False   |
+-----+-----+-----+
| True   | False  | Flase   |
+-----+-----+-----+
| False  | True   | False   |
+-----+-----+-----+
| True   | True   | True    |
+-----+-----+-----+
```

A	B	A and B
False	False	False
True	False	Flase
False	True	False
True	True	True

### シンプルテーブル

```
=====
A      B      A and B
False  False  False
True   False  False
False  True   False
True   True   True
=====
```

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

## CSV テーブル

```
.. csv-table::  
  
    "A", "B", "A and B"  
    "False", "False", "False"  
    "True", "False", "False"  
    "False", "True", "False"  
    "True", "True", "True"
```

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

---

## リストテーブル

```
.. list-table::  
  
    * - A  
      - B  
      - A and B  
    * - False  
      - False  
      - False  
    * - True  
      - False  
      - False  
    * - False  
      - True  
      - False  
    * - True  
      - True  
      - True
```

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

## 複雑なテーブル

- グリッドテーブルを使用すると複雑なテーブルを作成できます。

```
+-----+-----+-----+-----+
|      | A      | B      | Result |
+-----+-----+-----+-----+
| and  | False  | False  | False  |
+-----+-----+-----+-----+
|      | True   | False  |        |
+-----+-----+-----+-----+
|      | False  | True   |        |
+-----+-----+-----+-----+
|      | True   | True   | True   |
+-----+-----+-----+-----+
| or   | False  | False  | False  |
+-----+-----+-----+-----+
|      | True   | False  | True   |
+-----+-----+-----+-----+
|      | False  | True   |        |
+-----+-----+-----+-----+
|      | True   | True   |        |
+-----+-----+-----+-----+
```

	A	B	Result
and	False	False	False
	True	False	
	False	True	
	True	True	True
or	False	False	False
	True	False	True
	False	True	
	True	True	

## 列幅の変更

- CSV テーブルとリストテーブルの列幅のデフォルトは等幅です。
- `widths` オプションを指定し、列幅を割合で指定できます。

```
.. csv-table::
   :widths: 1, 1, 2
```

(次のページに続く)

(前のページからの続き)

```

"A", "B", "A and B"
"False", "False", "False"
"True", "False", "False"
"False", "True", "False"
"True", "True", "True"

.. list-table::
   :widths: 1,2,3

   * - A
     - B
     - A and B
   * - False
     - False
     - False
   * - True
     - False
     - False
   * - False
     - True
     - False
   * - True
     - True
     - True

```

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

## ヘッダーの指定

## グリッドテーブル

ヘッダーになる行の下に = で線を書きます。

```
+-----+-----+-----+
| A      | B      | A and B |
+=====+=====+=====+
| False  | False  | False   |
+-----+-----+-----+
| True   | False  | Flase   |
+-----+-----+-----+
| False  | True   | False   |
+-----+-----+-----+
| True   | True   | True    |
+-----+-----+-----+
```

A	B	A and B
False	False	False
True	False	Flase
False	True	False
True	True	True

## グリッドテーブル

ヘッダーになる行の下に = で線を書きます。

```
=====
A      B      A and B
=====
False  False  False
True   False  False
False  True   False
True   True   True
=====
```

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

## CSV テーブル

header-rows オプションでヘッダーになる行を指定します。

```
.. csv-table::
   :header-rows: 1

   "A", "B", "A and B"
   "False", "False", "False"
   "True", "False", "False"
   "False", "True", "False"
   "True", "True", "True"
```

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

## リストテーブル

header-rows オプションでヘッダーになる行を指定します。

```
.. list-table::
   :header-rows: 1

   * - A
     - B
     - A and B
   * - False
     - False
     - False
   * - True
     - False
     - False
   * - False
     - True
     - False
   * - True
     - True
     - True
```



A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

## テーブル名の追加

```
.. csv-table:: 論理積の結果
    "A", "B", "A and B"
    "False", "False", "False"
    "True", "False", "False"
    "False", "True", "False"
    "True", "True", "True"

.. list-table:: 論理和の結果
    * - A
      - B
      - A or B
    * - False
      - False
      - False
    * - True
      - False
      - True
    * - False
      - True
      - True
    * - True
      - True
      - True
```

表 1 論理積の結果

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

表 2 論理和の結果

A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

## 2.6.7 警告ディレクティブ

- 文章内に警告文を表示します。
  - 表示スタイルは使用している「テーマ」で決まります。
- 

### attention

.. attention::ここにタイトルを書きます (省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

注意:ここにタイトルを書きます (省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。
-

## caution

.. caution::ここにタイトルを書きます (省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

注意:ここにタイトルを書きます (省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

## danger

.. danger::ここにタイトルを書きます (省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

危険:ここにタイトルを書きます (省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

## error

.. error::ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

エラー:ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

---

## hint

.. hint::ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

---

ヒント:ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

---

## important

.. important::ここにタイトルを書きます（省略可能）

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

---

重要:ここにタイトルを書きます（省略可能）

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

---

## note

.. note::ここにタイトルを書きます（省略可能）

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

---

注釈:ここにタイトルを書きます（省略可能）

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

---

## tip

.. tip::ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

---

ちなみに:ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

---

## warning

.. warning::ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。  
改行位置は Sphinx 任せです。

``コードサンプル`` や **\*\*強調文字\*\*** など也可以使用できます。

– 箇条書きもできます。

警告:ここにタイトルを書きます(省略可能)

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。

---

## admonition

```
.. admonition:: ここにタイトルを書きます（省略可能）

    ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。
    改行位置は Sphinx 任せです。

    ``コードサンプル`` や **強調文字** など也可以使用できます。

    - 箇条書きもできます。
```

---

ここにタイトルを書きます（省略可能）

ここに警告の説明文を書きます。表示スタイルはテーマごとに異なります。改行位置は Sphinx 任せです。

コードサンプル や 強調文字 など也可以使用できます。

- 箇条書きもできます。
- 

## 2.6.8 その他の表現

### 【トピックス】

- 横線
- 

### 横線

- "----" でドキュメント内に横線を引きます。

```
じゅげむじゅげむ ごこうのすりきれ
```

```
----
```

```
かいじやりすいぎょのすいぎょうまつうんらいまつふうらいまつ
```

---

```
じゅげむじゅげむ ごこうのすりきれ
```

---

```
かいじやりすいぎょのすいぎょうまつうんらいまつふうらいまつ
```

## 2.7 CSS (Cascading Style Sheets)

一般的な HTML 文書と同様に Sphinx でも CSS (Cascading Style Sheets) を使用して文書内の見出しなどを装飾できます。

### 【トピックス】

- ・ 定義
  - ・ 実行結果
  - ・ その他の表現
- 

### 2.7.1 定義

#### 準備

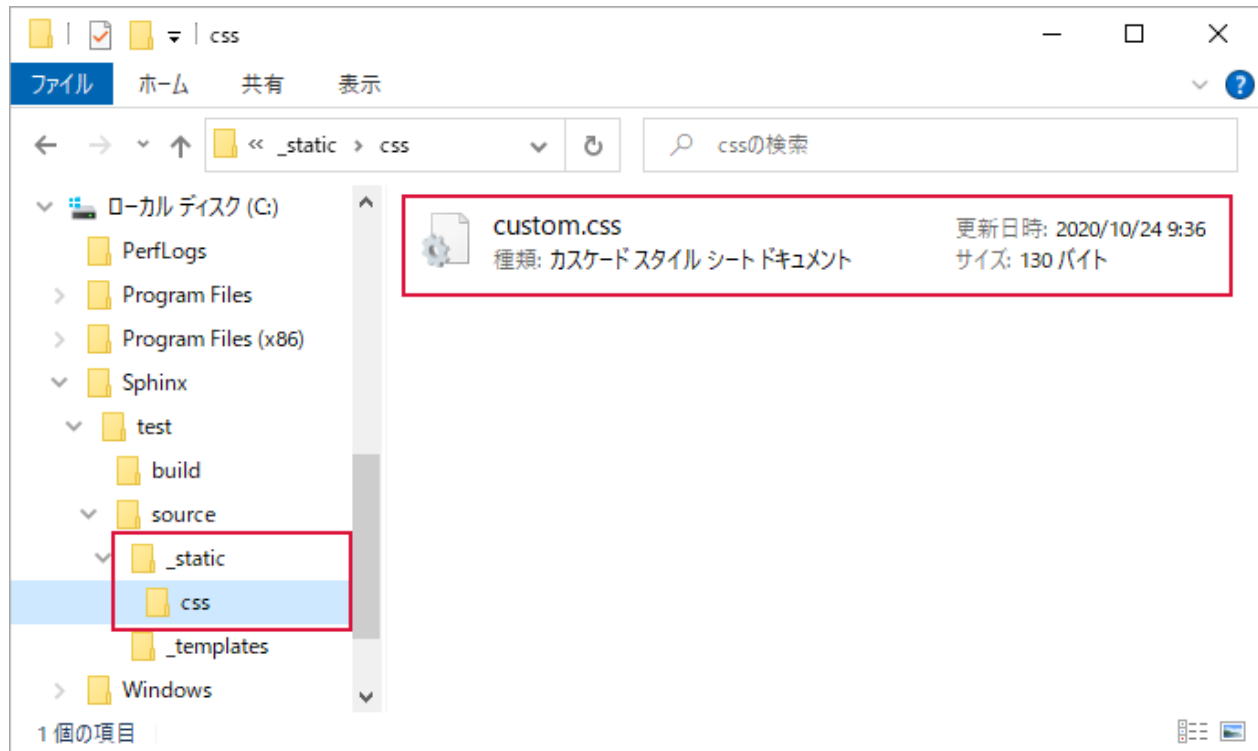
CSS を使用するときの準備です。

1. `_static` フォルダー内に CSS を定義したファイル (CSS ファイル) を配置する
2. `conf.py` ファイル内で CSS ファイルのパスを指定する

#### ファイルの配置

`_static` フォルダーは CSS などの定義ファイルの置き場所です。他の定義ファイルと分けるため今回は `_static` フォルダー内に `css` フォルダーを作成し、その中に CSS を定義した `custom.css` ファイルを置きます。



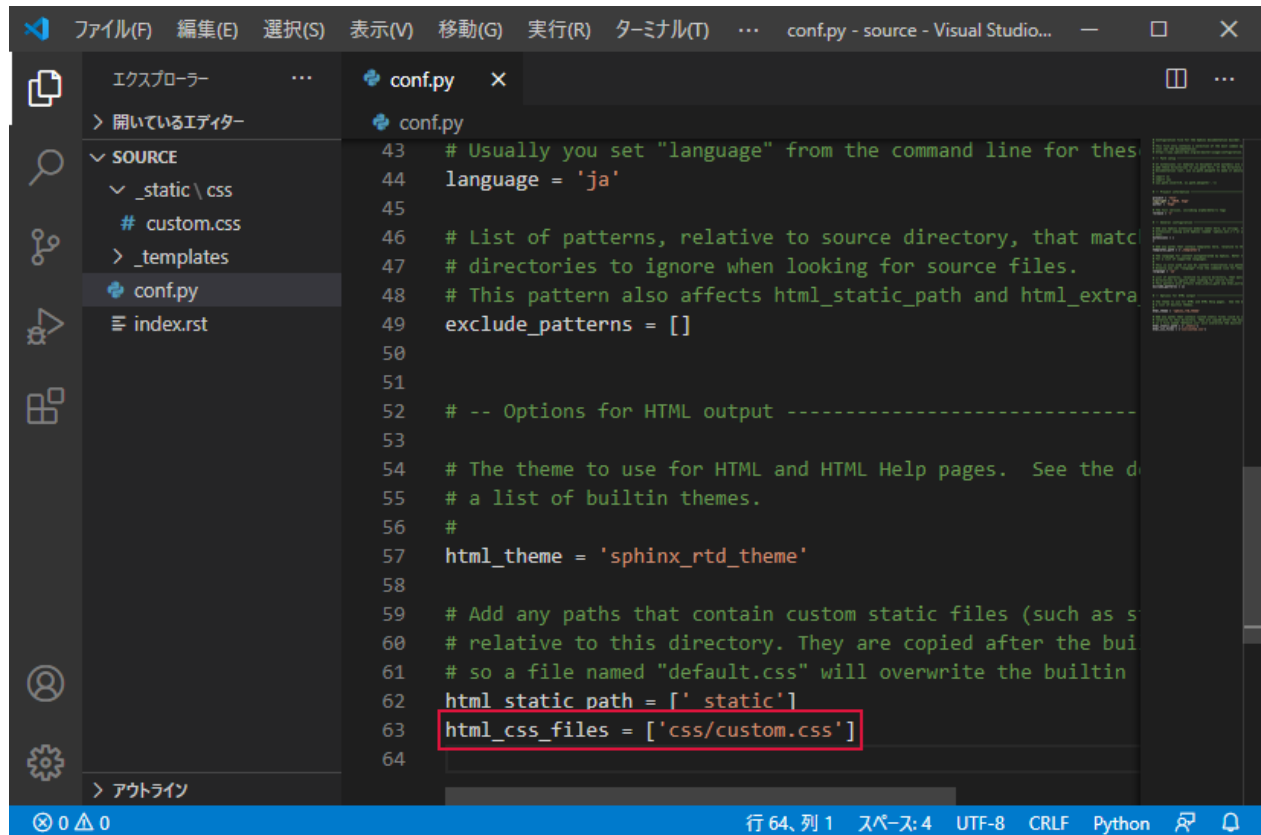


custom.css の内容です。

```
h1 {  
  padding: 0.2em;  
  border-top: solid 2px;  
  border-bottom: solid 2px;  
}
```

### conf.py ファイル

conf.py ファイル内に `html_css_files` を使用して CSS ファイルのパスを指定します。パスは `_static` フォルダを起点とした相対パスで指定します。



## 2.7.2 実行結果

- rst ファイル

```

#####
部の見出し
#####
部の紹介文。

*****
章の見出し
*****
章の紹介文。

セクションの見出し
=====
セクションの紹介文。

サブセクションの見出し
-----
サブセクションの紹介文。

```

(次のページに続く)

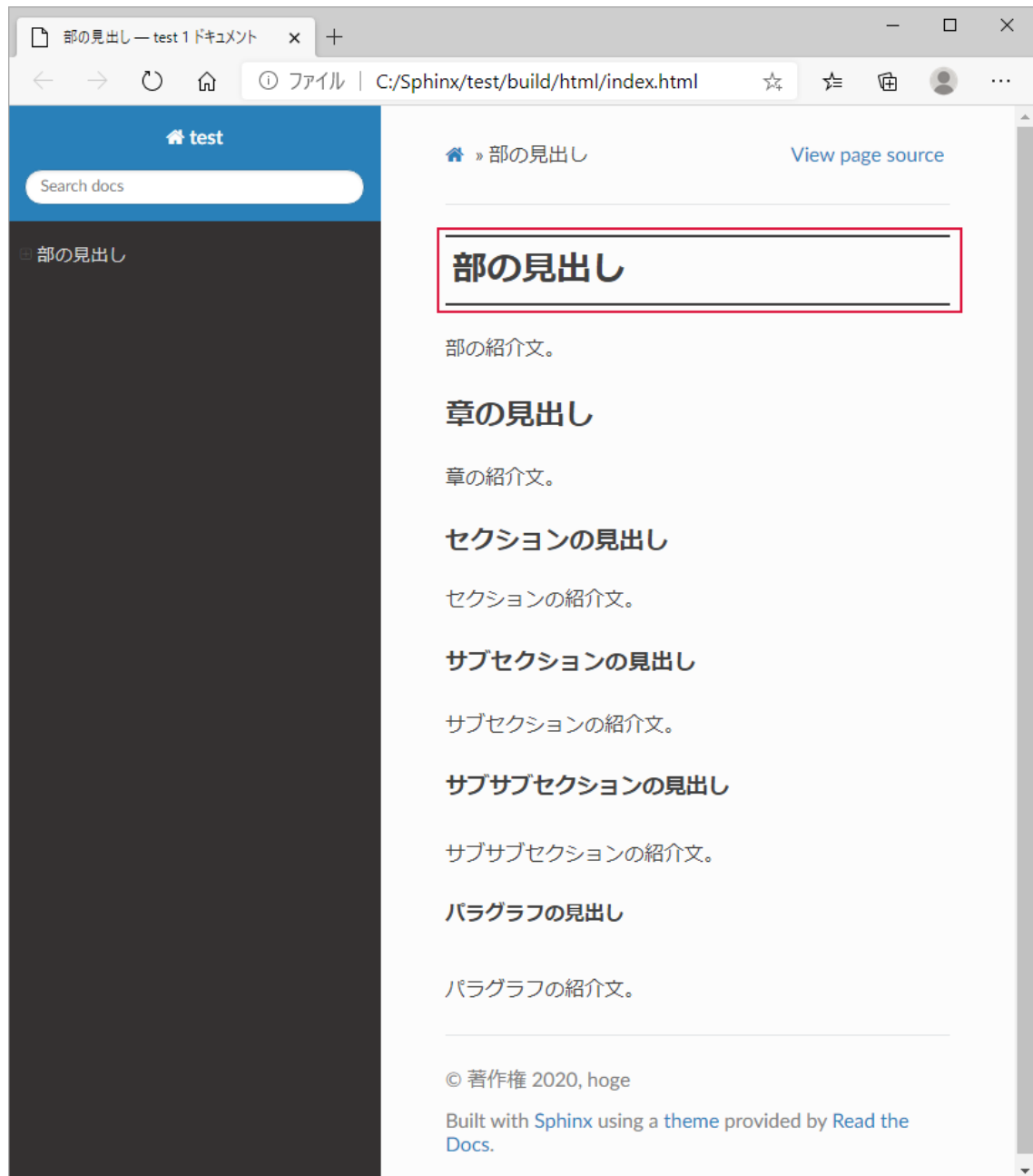
サブサブセクションの見出し  
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
サブサブセクションの紹介文。

パラグラフの見出し  
|||||  
|||||  
パラグラフの紹介文。



- CSS ファイルの指定あり

CSS ファイルで `h1` タグを変更しており、その内容が反映されます。



### 2.7.3 その他の表現

テーブルのセル内の折り返し

- rst ファイル

```
#####
サンプル
#####
```

(次のページに続く)

(前のページからの続き)

```
.. list-table::
   :header-rows: 1

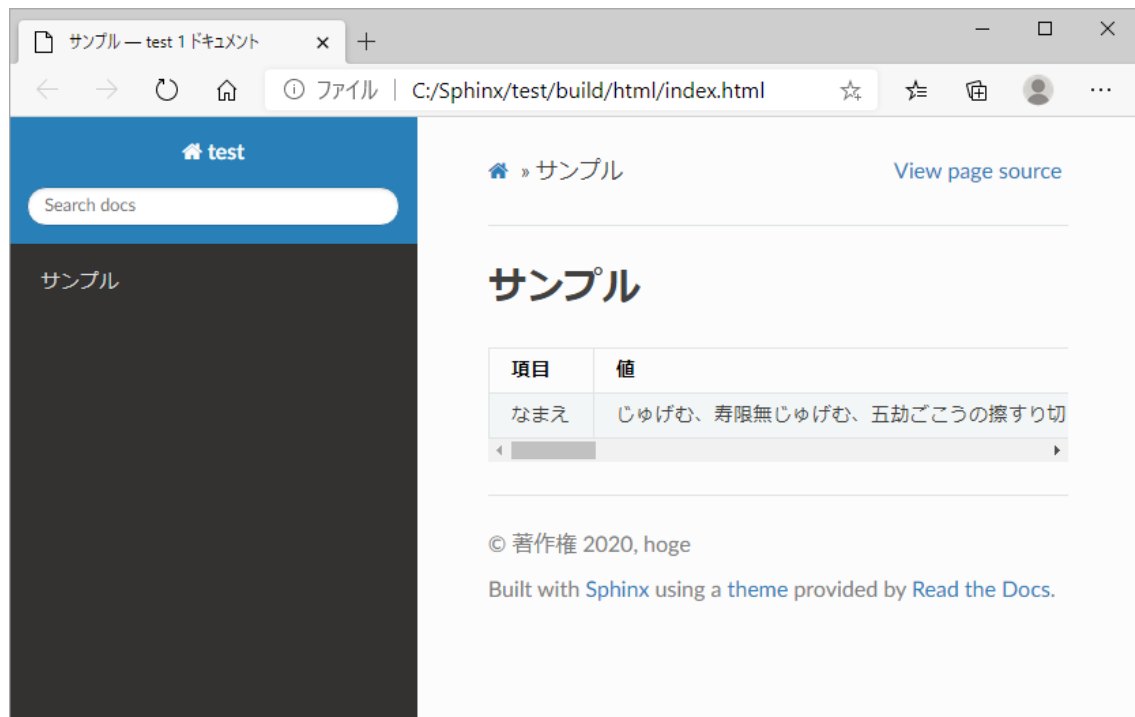
   * - 項目
     - 値
   * - なまえ
     - じゅげむ、寿限無じゅげむ、五劫ごこうの擦すり切きれ、海砂利かいじやり水魚すいぎょの、水行末すいぎょうまつ・雲来末うんらいまつ・風来末ふうらいまつ、喰う寝る処ところに住む処ところ、藪やぶら柑子こうじの藪柑子ぶらこうじ、パイポ・パイポ・パイポのシューリンガン、シューリンガンのゲーリングダイ、ゲーリングダイのポンポコピーのポンポコナの、長久命ちようきゅうめいの長助
```

- CSS ファイル

```
.wy-table-responsive table td, .wy-table-responsive table th {
    white-space: normal;
}
```

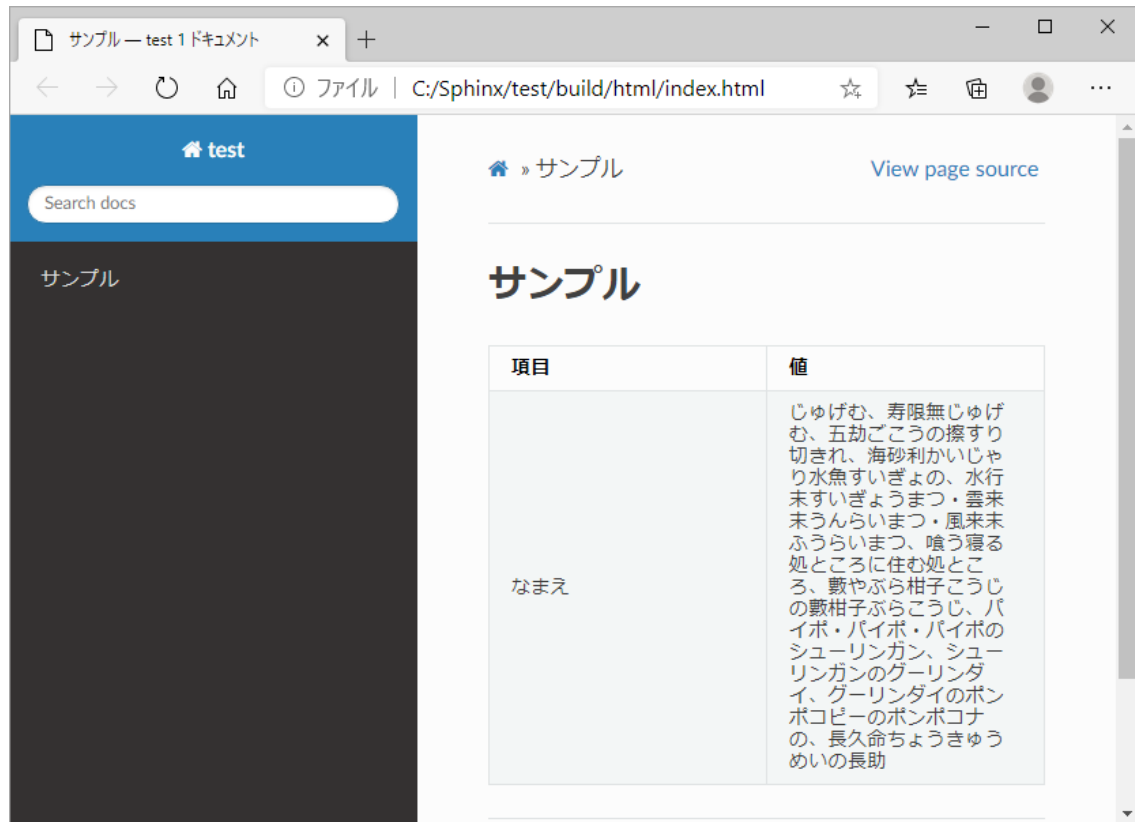
- CSS ファイルの設定なし

セル内の値が長いときはスクロールバーを表示します。



- CSS ファイルの設定あり

セル内の値が長いときは折り返してすべて表示します。

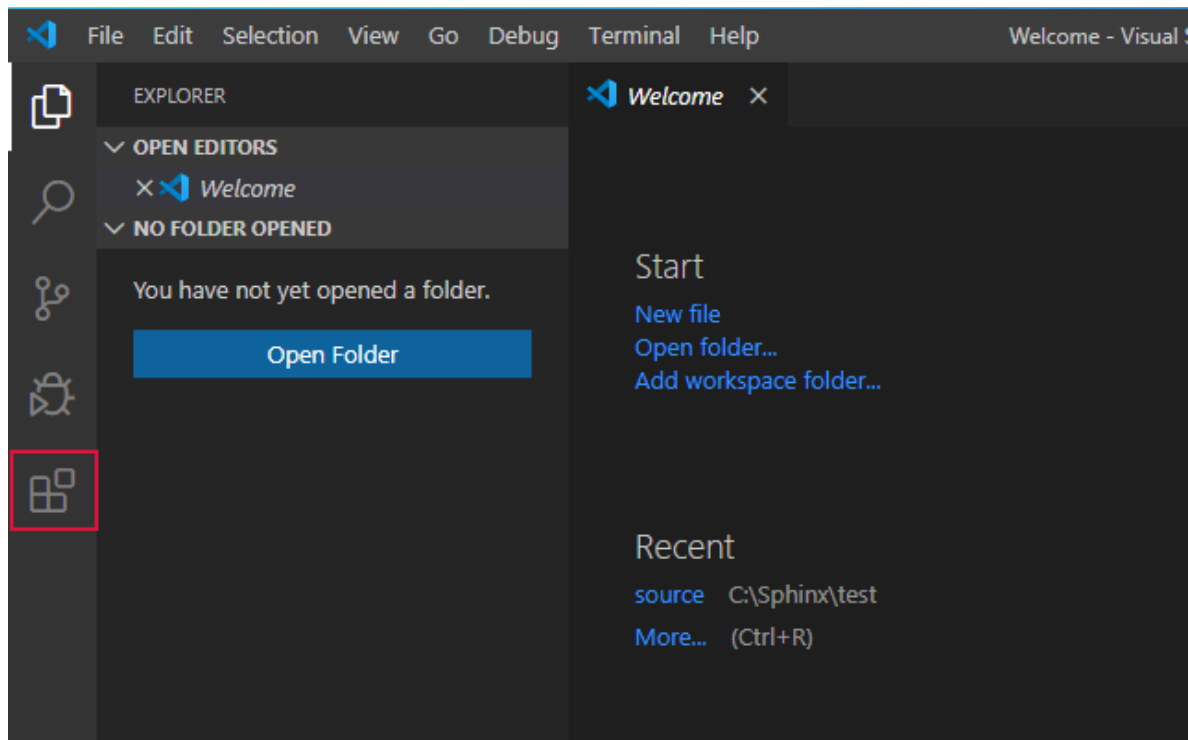


## 2.8 付録

*Visual Studio Code* の日本語化

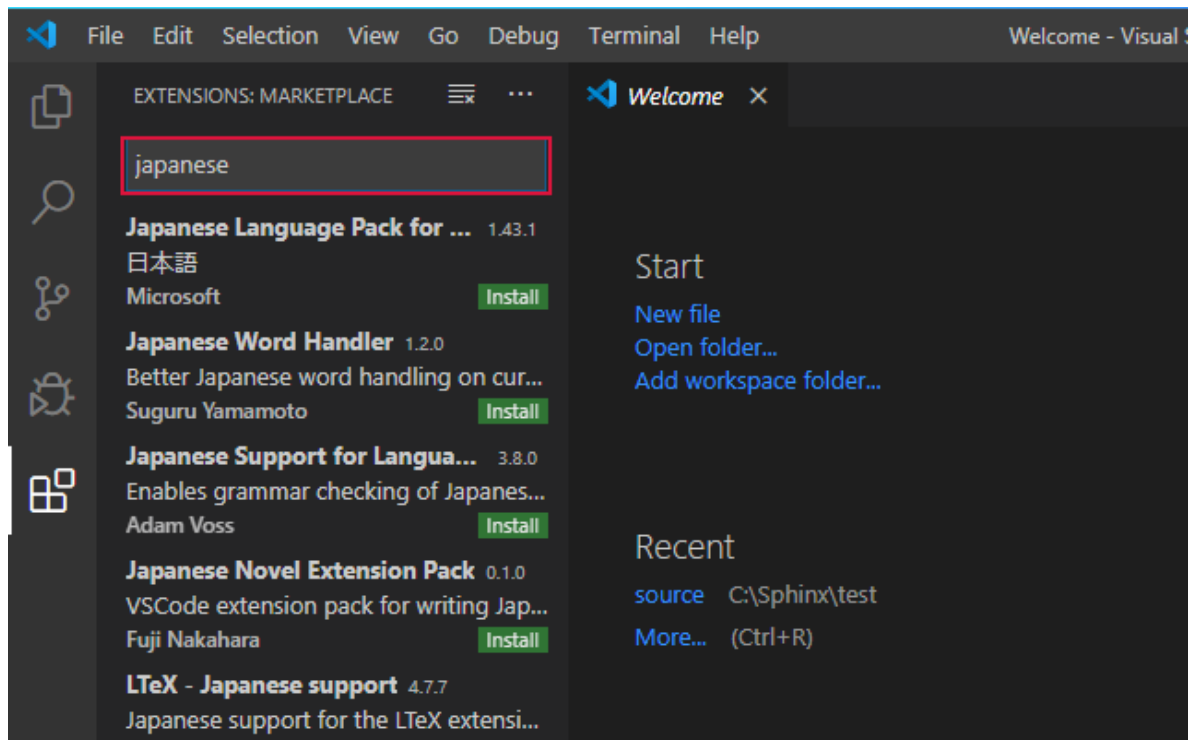
### 2.8.1 Visual Studio Code の日本語化

1. *Extensions* をクリック

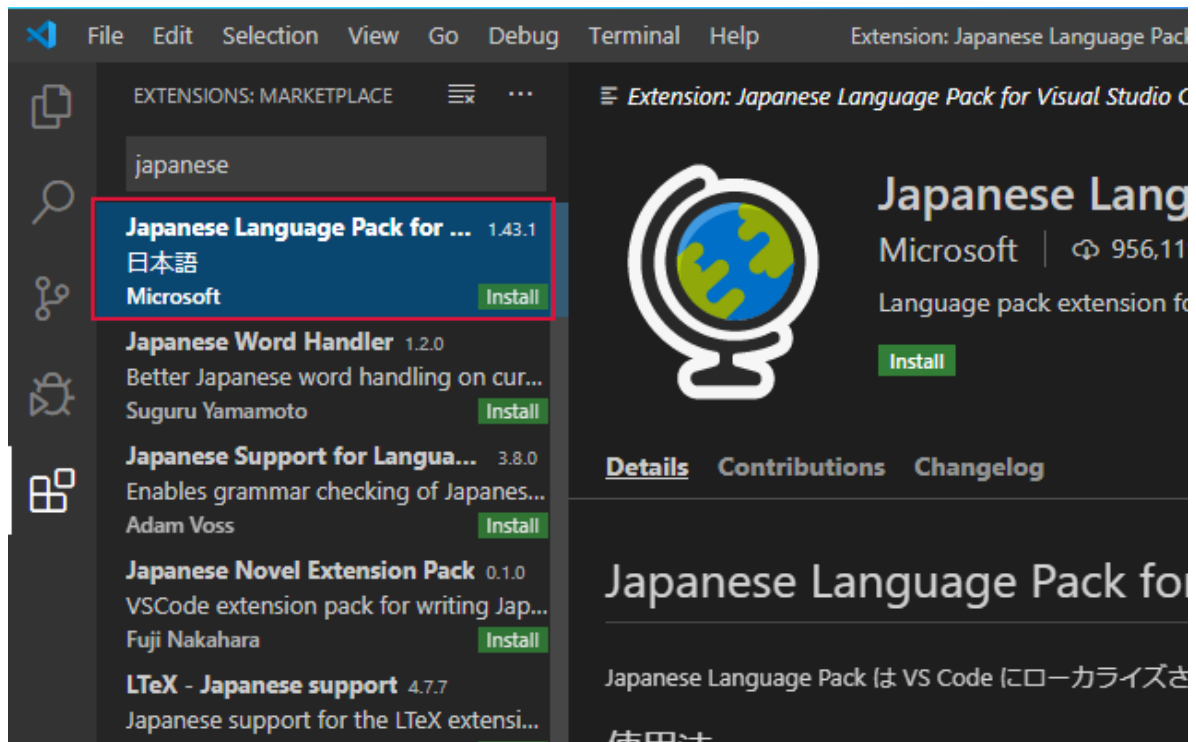


2. テキストボックスに "japanese" と入力





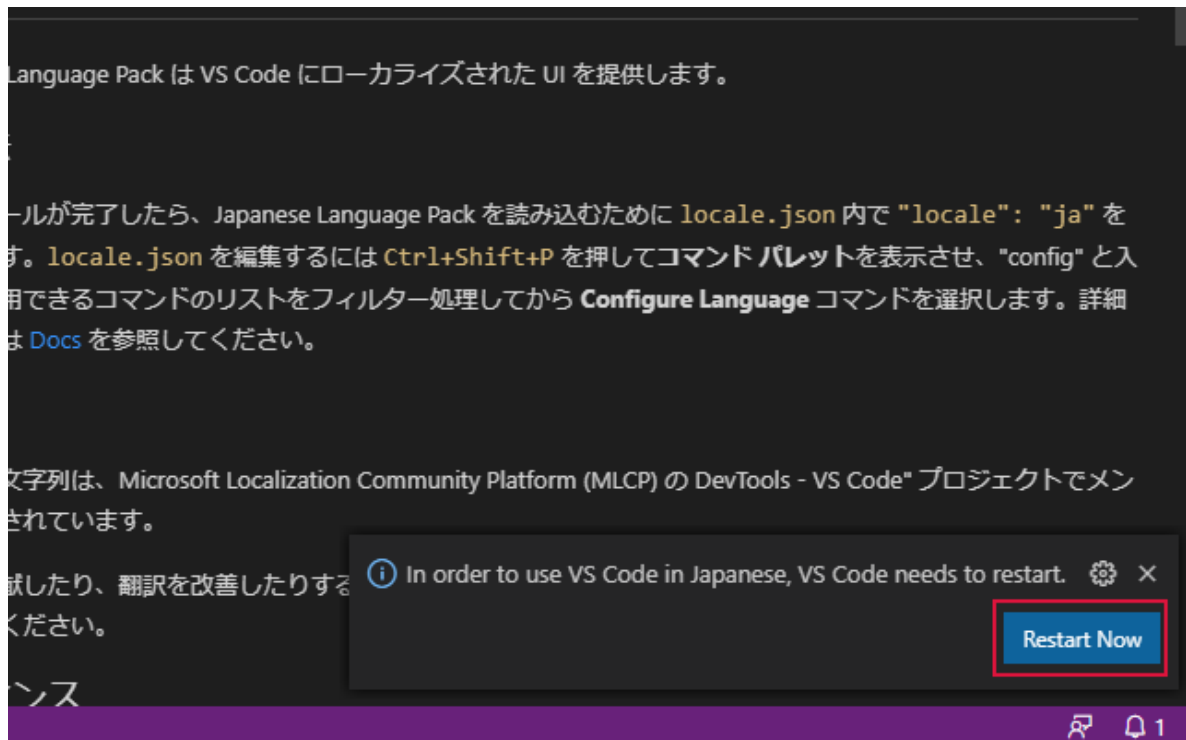
3. "Japanese Language Pack for Visual Studio Code" をクリック



4. *Install* をクリック



5. *Restart Now* をクリック



6. VS Code が再起動 → 日本語化終了



## 2.9 改版履歴

日付	内容
2020/04/12	初版公開
2020/06/27	<i>rst</i> ファイル, ラベル, テーブル名の追加, その他の箇条書き
2020/10/10	<i>Visual Studio Code</i> の日本語化 の項番を変更
2020/10/17	"japanesesupport.py" のダウンロードサイトが Sphinx の日本ユーザ会から GitHub へ移行されたことにともない「日本語対応にする(日本語拡張の導入)」内の手順を変更
2020/10/24	<i>CSS (Cascading Style Sheets)</i> , ヘッダーの指定