

Hugo-theme-vivliocli

User Guide

mochimo

目次

第1章	Getting Started	1
1.1	対象環境	1
1.2	インストール	1
1.3	最初のサイトの作成	2
1.4	NextStep	4
第2章	MarkdownとShortcodes	5
2.1	Markdown	5
2.2	Header 2	5
2.3	Shortcodes	9
第3章	フロントマター	13
3.1	_index.mdのフロントマター	13
3.2	各記事のフロントマター	13
第4章	config.tomlの設定	14
4.1	[params]の設定項目	14
第5章	_pdf.mdの設定	15
5.1	_pdf.md	15
5.2	_pdfcolophon.md	16
第6章	エディション	18
6.1	エディションのconfigを作成する	18
6.2	ビルドする	19
第7章	タグ付け	20
7.1	タグ一覧ページ	20
7.2	タグ付けの方法	20

第1章 Getting Started

1.1 対象環境

本解説は Windows を対象環境としています。

1.2 インストール

Hugo `v0.94.0` 以上をインストールします。

```
hugo version
:: Hugo Static Site Generator vx.xx.x ...
```

本テーマでは `vivliostyle-cli` を利用して PDF 出力を行うため、`Node.js` 上で動作する `vivliostyle-cli` のインストールを実施します。

```
node -v
:: vX.X.X
:: note: vivliostyle-cli is not working in Node v14.0.0

npm -v
:: X.X.X

npm install -g @vivliostyle/cli
vivliostyle -v
:: cli: X.X.X
:: core: X.X.X
```

hugo-theme-vivliocliのインストール

新しいプロジェクトを作成します。ここでは `myPDF` という名称とします。

```
hugo new site myPDF
cd myPDF
```

次に本テーマを submodule として導入します。

```
git init
git submodule add https://github.com/mochimochiki/hugo-theme-vivliocli themes/hugo-theme-vivliocli
```

git を利用しない場合は、`zip` をダウンロード・解凍し、`/themes/hugo-theme-vivliocli` に配置します。

exampleSiteからファイルをコピーする

テーマの中にテンプレートとして利用できるexampleSiteがあります。exampleSiteから必要なファイルをコピーします。また、デフォルトのconfig.tomlは削除しておきます。

```
xcopy /s themes/hugo-theme-vivliocli/exampleSite/config config/
xcopy /s themes/hugo-theme-vivliocli/exampleSite/CI CI/
rm config.toml
```

1.3 最初のサイトの作成

config.tomlの編集

/config/default.tomlを開き、以下の行をコメントアウトもしくは削除します。

```
themesdir = "../.."
```

また、baseURL、titleを編集します。

```
baseURL = ""
title = "My PDF Site"
```

表紙の作成

content下にja/firstpdf、en/firstpdfディレクトリを作成します。

```
mkdir content/ja/firstpdf
mkdir content/en/firstpdf
```

次にja/firstpdf/_pdf.mdファイルを作成して以下のように編集し、文字コードUTF-8で保存します。このファイルはPDFの表紙/前書/目次となります。_pdf.mdを配置したディレクトリ下がPDF出力の単位となります。

```
---
pdfname: 'firstpdf'
doctitle: 'My first pdf'
subtitle: 'subtitle'
header: '<日付>'
author: 'auther'
pagesize: 'A4'
book: true
cover: true
toc: true
sectionNumberLevel: 2
colophon: false
outputs:
- vivlio_cover
- vivlio_config
---
```

Note

詳細は [_pdf.md](#) の設定を参照してください。

記事の作成

次に記事を作成します。 `/content/ja/firstpdf` に `_index.md` を作成して以下のように編集します。

```
---
title: "firstpdf"
weight: 10
---
```

``_index.md`` はサイト上のセクションページになります。PDF には含まれません。

また `/content/ja/firstpdf` 下に `first.md` , `second.md` を作成して以下のように編集して保存します。

```
---
title: "最初の記事"
weight: 10
---
```

最初の記事

こんにちは。これは最初の記事です。

```
---
title: "次の記事"
weight: 20
---
```

次の記事

こんにちは。これは次の記事です。

以下のコマンドで Hugo のプレビューを表示してメニューの `Languages` から `Japanese` を選択し、作成した記事が表示されることを確認します。記事を編集して保存すると、LiveReload がかかり、プレビューも更新されます。

```
hugo server --config config/default.toml
```

ビルド

`Ctrl+C` で hugo のプレビューを終了し、以下のコマンドを実行して Hugo サイトをビルドします。

```
cd CI/windows
build_pdf.bat
```

エラーが出力された場合、PDFを開いたままにしているか確認してください。開いているとPDFの上書きに失敗します。

`/public_default/ja/firstPDF.pdf` が成果物です。確認してみましょう。

章節番号をカスタマイズする

1.1 などの章節番号の出力レベルは `_pdf.md` でカスタマイズすることができます。`_pdf.md` を開き、以下の箇所を変更してみましょう。

```
sectionNumberLevel: 2 # -> 1に変更
```

また、トップレベルの出力形式やデリミタの設定は `/config/default.toml` で設定できます。

```
[languages.ja.params]
sectionDelimiter = "."
sectionTopFormat = "第%s章" # -> "Chapter %s" に変更
```

その他の config パラメータについては [config.toml の設定](#) および [HUGO 公式ドキュメント](#)などを参照してください。

`/public_default/ja/firstpdf.pdf` で以下のことを確認します。

- `sectionNumberLevel = 1` としたことでセクション番号がトップレベルのみ表示されている
- 同時に PDF の目次にもトップレベルのみが表示されるようになっている
- `sectionTopFormat` の指定を "Chapter %s" にしたことで「第 1 章」ではなく「Chapter 1」となっている

1.4 NextStep

- 原稿で利用できる Markdown や Shortcodes を確認する
 - [Markdown と Shortcodes](#)
- PDF の表紙や奥付をカスタマイズする
 - [config.toml の設定](#)
 - [_pdf.md の設定](#)
- 複数のエディションを作る
 - [エディション](#)

第2章 Markdown と Shortcodes

本テーマで利用できる Markdown と Shortcodes のショーケースです。

2.1 Markdown

2.2 Header 2

Header 3

Header 4

Header 5

Header 6

テキスト

吾輩は猫である。名前はまだ無い。

どこで生れたかとうと見当がつかぬ。何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。吾輩はここで始めて人間というものを見た。しかもあとで聞くとそれは書生という人間中で一番獰悪な種族であったそうだ。この書生というのは時々我々を捕えて煮て食うという話である。しかしその当時は何という考もなかったから別段恐いとも思わなかった。ただ彼の掌に載せられてスーと持ち上げられた時何かフワフワした感じがあったばかりである。掌の上で少し落ちついて書生の顔を見たのがいわゆる人間というものを見始であろう。この時妙なものだと思った感じが今でも残っている。第一毛をもって裝飾されべきはずの顔がつるつるしてまるで薬缶だ。その後猫にもだいぶ逢ったがこんな片輪には一度も出会わした事がない。のみならず顔の真中があまりに突起している。そうしてその穴の中から時々ぷうぷうと煙を吹く。どうも咽せぼくて実に弱った。これが人間の飲む煙草というものである事はようやくこの頃知った。

この書生の掌の裏うちでしばらくはよい心持に坐っておったが、しばらくすると非常な速力で運転し始めた。書生が動くのか自分だけが動くのか分からないが無暗に眼が廻る。胸が悪くなる。到底助からないと思っていると、どさりと音がして眼から火が出た。それまでは記憶しているがあとは何の事やらいくら考え出そうとしても分からない。

ふと気が付いて見ると書生はいない。たくさんおった兄弟が一疋も見えぬ。肝心の母親さえ姿を隠してしまった。その上今までの所とは違って無暗に明るい。眼を明いていられぬくらいだ。はてな何でも容子がおかしいと、のそのそ這い出して見ると非常に痛い。吾輩は藁の上から急に笹原の中へ棄てられたのである。

夏目漱石(1905)『吾輩は猫である』[青空文庫](#), 底本:「[夏目漱石全集1](#)」[ちくま文庫](#)、[筑摩書房](#) 1987(昭和62)年9月29日第1刷発行

箇条書き

- 箇条書き
 - 箇条書き
 - 箇条書き
 - 箇条書き
- 箇条書き

- 箇条書き

順序つきリスト

1. AAA
2. BBB
3. CCC
 1. AAA
 2. BBB
 3. CCC
 - AAA
 - BBB
4. DDD
5. EEE

リンク

[Google](#) のリンク。

表

No.	item	note
1	AAA	noteA
2	BBB	noteB
3	CCC	noteC

引用

引用文です。

インラインコード

`inlinecode` です。

コードブロック

整形済みテキストです。

整形済みテキストです。

```
// comment
if (a == b)
{
    return true;
}
```

脚注

Google¹ です。

Here is a footnote reference,²

水平線

斜体

italic 斜体

太字

bold 太字

取り消し線

~~取り消し線~~

MathJax

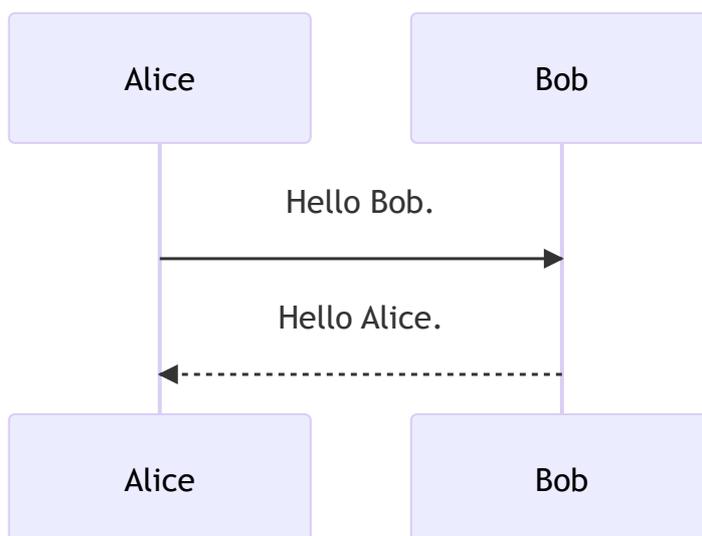
インライン数式は $y = ax^2 + \frac{b}{c}$ のように記載します。インライン数式を有効にするにはフロントマターに `math: true` を記載する必要があります。

ブロック数式は `math` コードブロックで記載します。

$$\frac{\pi}{2} = \left(\int_0^{\infty} \frac{\sin x}{\sqrt{x}} dx \right)^2 = \sum_{k=0}^{\infty} \frac{(2k)!}{2^{2k}(k!)^2} \frac{1}{2k+1} = \prod_{k=1}^{\infty} \frac{4k^2}{4k^2 - 1}$$

Mermaid

`mermaid` コードブロックは [Mermaid](#) で記載されたダイアグラムとして描画されます。



図

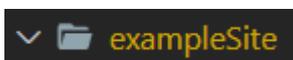
通常の図

代替テキストなしの図



インライン

インラインで小さな図を表示する



figure

代替テキストを書くと図番号とキャプションが付く。

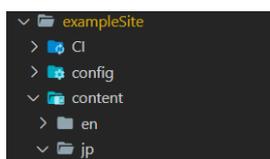


Fig. 2.1: exampleSiteのスクリーンショット

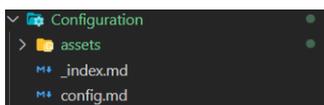


Fig. 2.2: Configurationディレクトリのスクリーンショット

代替テキストを空白文字のみにすると図番号とキャプションは付かず、中央寄せのみ。



画像サイズの変更(属性の付加)

URLクエリパラメータとして属性を付加すれば画像サイズ変更や境界線を付加可能。

```
![border=10 & width=50%](assets/2021-02-28-22-21-35.png?border=10&width=50%)
```

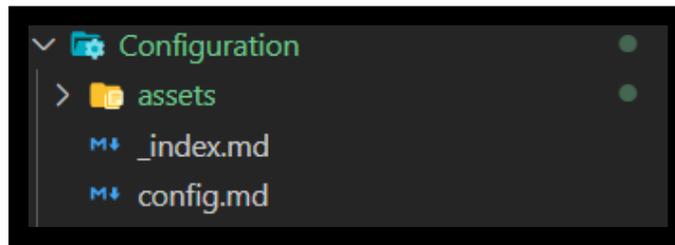


Fig. 2.3: border=10 & width=50%

2.3 Shortcodes

ShowIf

`config.toml` の `showIfs` で列挙されている場合に描画する部分を指定します。以下は `showIfs = ["edition1"]` とした場合に描画されるブロックです。

```

{{% ShowIf edition1 %}}
ここにxxxをサポートする場合に表示するコンテンツを記述。
{{% /ShowIf %}}

```

詳しくは [エディション](#) を参照してください。

HideIf

`config.toml` の `showIfs` で列挙されている場合に描画「しない」部分を指定します。以下は `showIfs = ["edition1"]` とした場合に描画されなくなるブロックです。

```

{{% HideIf edition1 %}}
ここはedition1のときのみ非表示になる。
{{% /HideIf %}}

```

詳しくは [エディション](#) を参照してください。

note

注記です。以下のように `note` ショートコードで囲まれた部分が注記としてレンダリングされます。

```

{{% note %}}
ここに注記文章を記載
{{% /note %}}

```

Note

ここに注記文章を記載

`note (title)` という形式で、引数にタイトルを指定することもできます。`note` 内部に Markdown を書くことも可能です。

```

{{% note 注記 %}}
ここに注記文章を記載

```

```

* markdownも記載可能
  * 箇条書きレベル2
* 箇条書きレベル1
{{% /note %}}

```

注記

ここに注記文章を記載

- markdownも記載可能
 - 箇条書きレベル2
- 箇条書きレベル1

include

Markdown ファイル、csv ファイルの「部品」を用意しておき、原稿の任意の箇所に「挿し込む」事ができます。部品ファイルを `/content/<language>/_include` 以下に配置しておけば、以下のショートコードで include することができます。

```

{{< include "test_ja.md" >}} # /content/ja/_include/test_ja.md
{{< include "/sample/sample_ja.md" >}} # /content/ja/_include/sample/sample_ja.md
{{< include "test_ja.csv" >}} # /content/ja/_include/test_ja.csv
{{< include "./test.csv" >}} # /(md file dir path)/test_ja.csv

```

- `_include` ディレクトリ内の Markdown にはフロントマターは記載しません。
- include ショートコードは `{{< >}}` スタイル (Markdown レンダリング無し) で記述してください。`{{% %}}` スタイル (Markdown レンダリングあり) で記述すると、csv 読み込みが正しく動作しません。

csv の include

csv ファイルの include では、Markdown よりも高度な表を描画することができます。

```

{{< include
  src="./rich.csv" # source ファイルのパス
  caption="サンプル表" # キャプション
  class="gray" # table に付加する class 属性。未指定/simple/gray
  markdown=true # true にした場合表内部の Markdown をレンダリング。初期値 false
  head=1 # true にした場合1行目をヘッダーとみなす。初期値 true。数値の場合数値行数をヘッダーとみなす。
  align="left" # 表自体を左寄せ/中央寄せ/右寄せ。left / center / right
  head_align="center" # 表のヘッダーのすべての列を左寄せ/中央寄せ/右寄せ。left / center / right
  body_align="left" # 表のボディのすべての列を左寄せ/中央寄せ/右寄せ。left / center / right
  width="90%" # table の width 属性を設定。
>}}

```

表 2.1: サンプル表

id	カテゴリ	名称	備考
001	果物	りんご	
002		みかん	<ul style="list-style-type: none"> • せとか • 甘平
003		ぶどう	巨峰
004	お菓子	ポテトチップス	<ul style="list-style-type: none"> • うすしお • のりしお
005		飴	<ul style="list-style-type: none"> • みかん • ぶどう

csvの書式

縦横結合

|| で縦方向に結合, -> で横方向に結合できます。

ColumnCodes

ヘッダー(複数行の場合最終行)に ColumnCodes を埋め込むことができます。ColumnCodes は `[[@<識別子>=<値>]]` という形式のコードです。ヘッダーセルの文末に記載することで列に対して作用します。複数のコードを適用する場合 `[[@<識別子>=<値> @<識別子>=<値>]]` のようにスペース区切りで記述します。以下に ColumnCodes の一覧を列挙します。

ColumnCodes	解説
<code>[[@id=myId]]</code>	この列の全セルの <code><td></code> タグに <code>myColumnId</code> を <code>class</code> 属性として埋め込みます。
<code>[[@w=20%]]</code>	この列の幅をテーブルの20%とします。指定のない列は <code>auto</code> とみなされます。本コードが1つでもあると、 <code>include</code> ショートコードの <code>width_ratio</code> オプションは無視されます。
<code>[[@h:--]] / [[@h:--:]] / [[@h--:]]</code>	ヘッダー左/中央/右寄せ(- の数は任意)

ColumnCodes	解説
[[@:--]] / [[@:--:]] / [[@--:]]	ボディ左/中央/右寄せ(-の数は任意)

	column 1	column 2	column 3	column 4
1	id	カテゴリ	名称	備考
2	001	果物	りんご	
3	002		みかん	* せとか * 甘平
4	003		ぶどう	巨峰
5	004	お菓子	ポテトチップス	* うすしお * のりしお
6	005		飴	* みかん * ぶどう

Fig. 2.4: rich.csv

csvの特定の値のみ参照する

csvファイルについては行を決定するためのkeyと列名を指定することで特定の値のみを参照することもできます。keyには最も左の列が使われます。重複する値がある場合は最初に見つかったものが優先されます。

```
{{< include "test_ja.csv" "003" "Name" >}} # /content/ja/_include/test_ja.csv の "003" に  
マッチした行の "Name" 列の値
```

上記の場合、1列目の値が003である行のName列の値がショートコードの位置に挿し込まれます。

1. 脚注テキスト <https://www.google.co.jp/> ↗

2. Here is the footnote. ↗

第3章 フロントマター

セクションや各記事の設定について解説します。

3.1 `_index.md`のフロントマター

`_index.md` はデフォルトでは各記事と同等レベルのセクションとして扱われますが、`role` を設定することでパート/チャプターなどのより大きな枠とすることができます。

role

`role: doc-part` : `_index.md` はパートの表紙ページとして PDF 出力されます。

`role: doc-chapter` : `_index.md` はチャプターのタイトル及び最初の記事として PDF 出力されます。

3.2 各記事のフロントマター

ShowIf

ShowIf フロントマターを指定することで、`config.toml` の `showIfs` で列挙されているキーワードにマッチした場合にのみ記事を描写することができます。以下のフロントマターを指定した記事は `showIfs = ["edition1"]` である場合に描写されず。

```
---
title: edition1の説明
ShowIf: ["edition1"]
---
```

詳しくは [エディション](#) を参照してください。

HideIf

HideIf フロントマターを指定すると、そのエディションでのみ非表示とする記事/セクションを作ることができます。詳しくは [エディション](#) を参照してください。

math

`math: true` としたページにはインライン数式を記述することができます。インライン数式は `$E = mc^2$` のように記載します。なおブロック数式は本フラグの値にかかわらず常に以下のように記載できます。

```
```math
y = ax^2 + bx + \frac{c}{d}
```
```

第4章 config.toml の設定

config.toml の主な項目の説明をします。設定項目については [HUGO 公式ドキュメント](#) や default.toml のコメントも合わせて参照してください。

4.1 [params] の設定項目

[params]以下に設定する設定値です。

isPDF

PDFかどうか。trueにすると各htmlページでの menu 生成が抑制されます。PDFビルド時は true に設定します。

theme_css

組版に使用するcssを指定します。/static/css/yourtheme.css を使用する場合 /css/yourtheme.css と、指定します。デフォルトではテーマの static/css 下にある style-main.css が参照されています。

showlfs

ShowIf / HideIf ショートコード・フロントマターで表示/非表示切り替える条件の一覧。

```
showIfs = ["edition1", "edition2"]
```

詳しくは [エディション](#) を参照してください。

sectionNumberLevel

デフォルトの章番号を付加するレベルを指定します。0の場合章番号の付加は行われません。

sectionDelimiter

sectionNumberLevel >= 2 の場合の章・節番号のデリミタを指定します。設定されていない場合、. がデリミタとなります。

sectionTopFormat

トップレベルのセクションの番号をフォーマットすることができます。フォーマット中に %s を1つ指定してください。その位置にセクション番号が挿入されます。言語ごとに変更することも可能です。

例) sectionTopFormat = "Chapter %s"

```
[languages.ja.params]
sectionTopFormat = "第%s章"
[languages.en.params]
sectionTopFormat = "Chapter %s"
```

第5章 _pdf.md の設定

本テーマでは vivliostyle で利用される構成ファイル/カバーページ(目次)/奥付を生成するために、以下の config ファイルを利用しています。

- _pdf.md
- _pdfcolophon.md

サイト内の PDF 生成したいレベルのディレクトリにこの 2 ファイルを配置することで、PDF 化するための構成ファイルを生成できます(サイトルート以外も可)。

5.1 _pdf.md

`_pdf.md` は vivliostyle の config ファイルと表紙のテンプレートです。以下のように記述します。

```

---
pdfname :
  default: "pdf ファイル名"
  # editionA: "editionA ファイル名"
doctitle:
  default: "表紙タイトル"
  # editionA: "editionA 表紙タイトル"
subtitle:
  default: "表紙サブタイトル"
  # editionA: "editionA 表紙サブタイトル"
doc_number:
  default: "文書番号"
  # editionA: "editionA 文書番号"
author: "著者"
company: "会社"
pagesize: "A4"
colophon: true
outputs:
- vivlio_cover
- vivlio_config
---

# はじめに

```

(ここに目次前に表示したい文章を記述)

`default` や `editionA` とリスト表記している設定値は config.toml の `showIfs` で指定したエディション名を指定することでそのエディション用に設定できます。

pdfname

PDF ファイル名です。必須です。`/content/<language>` 以下では競合しないよう一意の名称となるようにします。

doctitle

表紙のタイトルです。必須です。エディションごとに切替可能です。

subtitle

表紙に表示するサブタイトルです。エディションごとに切替可能です。

doc_number

表紙に表示する文書番号です。エディションごとに切替可能です。

author

表紙に表示する著者です。

company

表紙に表示する会社名です。

pagesize

ページサイズです。必須です。vivliostyle.config.js の pagesize 項目になります。A4/A5 など。

colophon

奥付を生成するかどうか。true で生成します。true の場合に後述の _pdfcolophon.md が必要になります。

outputs

config ファイルを生成するために必要な記述です。変更せずそのまま記載します。必須です。

5.2 _pdfcolophon.md

奥付生成のためのテンプレートです。このファイルは _pdf.md のフロントマターで colophon: true の場合に必要になります。下記のように記述します。

```

---
title: "奥付"
outputs:
- vivlio_colophon
---

## 奥付のタイトル

(奥付の本文)

<div role="doc-colophon">

## 本のタイトル

xxx年x月x日 初版発行

```

```
著者	著者名
発行者	発行者名
印刷・製本	印刷社名

© コピーライト記述

</div>
```

title

奥付のタイトルです。本文には表示されません。ヘッダーにタイトルを表示している場合、このタイトルが表示されます。必須です。

outputs

奥付用ファイルを生成するために必要な記述です。変更せずそのまま記載します。必須です。

本文

奥付ページの本文欄です。通常のファイルと同様です。

`<div role="doc-colophon">` タグ内

本タグ内は下揃えで描画されます。本のタイトル及び出版情報の表を記述することを想定しています。

第6章 エディション

内容が異なるPDFのファミリーを生成する必要がある場合、HUGOの [Configuration Directory](#) の仕組みを利用してエディションを作り分け、構成や記事内容を切り替えてビルドできます。

6.1 エディションの config を作成する

`/config/myedition.toml` を作成し、ビルド時に `default.toml` とともに指定することで簡単に `myedition` エディションを作成することができます。

特定要素の表示/非表示を切り替える

`ShowIf/HideIf` ショートコード/フロントマターと共に `showIfs` を使用することで表示/非表示を切り替えます。

```
[params]
showIfs = ["edition1"]
```

この場合、`.md` の記事で以下のように記載した要素が表示されます。

```
{{% ShowIf edition1 %}}
ここに edition1 をサポートする場合には表示するコンテンツを記述。
{{% /ShowIf %}}
```

以下のように記載した要素は `edition1` でのみ非表示になります。

```
{{% HideIf edition1 %}}
ここに edition1 の場合には非表示にするコンテンツを記述。
{{% /HideIf %}}
```

特定記事・セクションの表示/非表示を切り替える

記事/セクション単位では以下のようにフロントマターを書いている場合にPDFに含まれます。必ずリスト形式 (`["xxx", "yyy"]`) で書く必要があります。

```
[params]
showIfs = ["edition1"]
```

`config.toml` に上記の用に `"edition1"` を含むとき、以下のページはPDFに含まれることになります。

```
---
title: edition1の説明
ShowIf: ["edition1"]
---
```

```
config.toml
```

で `showIfs = ["@all-pages"]` のように、`@all-pages` を含めると、各記事に ShowIf フロントマターが書かれていても強制的に全ページを PDF 描画対象とします。

`HideIf` を使用し以下のようにすると、`edition1` の PDF には含まれなくなります。

```
---  
title: edition1 以外で表示  
HideIf: ["edition1"]  
---
```

6.2 ビルドする

エディションを指定してビルドするには、引数にエディションを指定します。例えば `pdf_other` エディションをビルドするには以下のように指定します。

```
.\CI\build.bat pdf_other
```

第7章 タグ付け

ページにタグ付けすることができます。タグ付けされたページは、タグごとに分類されたタグ一覧ページからリンクされます。

7.1 タグ一覧ページ

タグ付けされたページが一つでも存在する場合、メニューの言語セレクタの下にタグ一覧ページへのリンクが表示されます。

7.2 タグ付けの方法

タグを付けるために必要なのは、ページのフロントマターで `tags: ["tagA", "tagB"]` のようにページに付加するタグのリストを宣言することだけです。これによりタグ一覧ページにタグが列挙されます。

```
---  
title: ページにタグ付けする  
weight: 70  
---
```