

# My Book

published by ReVIEW

# 目次

第1章 Vivliostyleのテスト	3
1.1 仕組み	3
第2章 章見出し	4
2.1 節見出し	5
採番しない項見出し	5
採番する項見出し	5
nodispで隠れた見出し	5
2.2 箇条書き	5
ナカグロ箇条書き	5
番号箇条書き	6
用語リスト	6
2.3 ブロック命令	6
ソースコード	6
図	8
表	10
囲み記事	12
2.4 インライン命令	15
書体	15
第3章 Markdownファイルの取り込み	16

# 第1章 Vivliostyleのテスト

Vivliostyleでビルドするテストです。

## 1.1 仕組み

`rake vivliostyle` でVivliostyleによるPDFが生成されます。仕組み的には、EPUBを作った後に `vivliostyle-cli` を呼び出し、PDF変換します。

## 第2章 章見出し

章にはリードが入ることがあります。図版や表が入ることはまずありませんが、太字くらいは使われることも。

複数段落になる可能性もあります。

段落は普通書き連ねていくだけです。段落間は空行を含めます。

こんなかんじ

です。空けない場合は、1つの段落として結合されます。TeXと違って文字種によって良い塩梅にスペースを入れてくれたりはしないので、特に英文の場合は注意が必要です。this is apen.と「apen」になってしまいます。行頭行末のスペース文字も詰められてしまうので、this is a pen.は途中改行せずに記述しなければなりません(config.ymlで `join_lines_by_lang: true` とすると、unicode-eaw ライブラリによる言語判定でのスペース挿入もできます)。

通常段落は「字下げ」することを想定して表現されますが、たとえばコードをまたぐ

```
'hello!' 'こんにちはWorld!'
```

ようにしたい場合は、またいだ後の段落前に「`//noindent`」を入れておくことで字下げを抑止できます。

強制的に空行を作りたいときには `//blankline` を使います。

3行アキを入れました。

## 2.1 節見出し

`=` の数で見出しレベルを表しますが、最大 `=====` の5レベルまでの見出しがあり得ます。

- `=` : 章および部
- `==` : 節
- `===` : 項
- `====` : 段
- `=====` : レベル5見出し

「X.X.X」のように採番するか否かは `config.yml` の `secnolevel` パラメータで変動します。

また、`nonum`あるいは`notoc`を付けた見出しは、章であっても採番されません。前者`nonum`は採番なし・目次に含める、後者`notoc`は採番なし・目次にも含めないという意味です。

### 採番しない項見出し

#### 採番する項見出し

`nodisp`を付けると、紙面には表示されず目次には含まれる見出しとなります（採番なし）。節以下のレベルで使うことはほとんどなく、たとえば「献辞」のように紙面には見出しを出したくないけれども目次には入れておきたい前付名などに使うことを想定しています。

## 2.2 箇条書き

### ナカグロ箇条書き

ナカグロ箇条書き（HTMLの`ul`、TeXの`itemize`）はスペース+`*`+スペースで表現します。インラインタグが含まれることがあります。

- 箇条書き1
- 箇条書き2太字***bold italic***イタ 等幅code

入れ子ナカグロ箇条書きもあります。

- 箇条書き1
  - 箇条書き1-1
  - 箇条書き1-2
    - 箇条書き1-2-1
- 箇条書き2
  - 箇条書き2-1

箇条書きの間に別の要素（ぶらさがりの段落など）が入る場合、`//beginchild` と `//endchild` で囲むことで入れ子表現可能です。濫用は避けてください。

## 番号箇条書き

番号箇条書き（HTMLの`ol`、TeXの`enumerate`）はスペース+数字.+スペースで表現します。

1. 箇条書き1
2. 箇条書き2太字**bold** 等幅code

## 用語リスト

用語リスト（HTMLの`dl`、TeXの`description`）はスペース+:+スペースで見出しを、説明は行頭にタブかスペースを入れて表現します。

Alpha**bold**太字*italic*イタ 等幅code

*DEC*の作っていたRISC CPU。 *italic*イタ 等幅code 浮動小数点数演算が速い。

POWER

IBMとモトローラが共同製作したRISC CPU。

派生としてPOWER PCがある。

SPARC

Sunが作っているRISC CPU。CPU数を増やすのが得意。

## 2.3 ブロック命令

### ソースコード

採番付きリストの場合はlistです（[リスト2.1](#)）。

リスト2.1: Rubyの `hello` コード

```
puts 'Hello, World!'
```

行番号と採番付きのリストはlistnumです。

リスト2.2: 行番号はリテラルな文字で特に加工はしていない

```
1 class Hello
2   def initialize
3     @msg = 'Hello, World!'
4   end
5 end
```

---

採番なしはemlistを使います。キャプションはあったりなかったりします。

```
printf("hello");
```

Python記法

```
print('hello');
```

行番号付きのパターンとしてemlistnumがあります。

```
1 printf("hello");
```

---

Python記法

```
101 print('hello');
```

---

ソースコード引用を主ターゲットにするにはsourceというのを用意しています。

hello.rb

```
puts &#39;Hello&#39;
```

実行例を示すとき用にはcmdを用意しています。いずれにせよ、商業書籍レベルでは必要なので用意しているものの、原稿レベルで書き手が使うコードブロックはほどほどの数に留めておいたほうがよいのではないかと思います。TeX版の紙面ではデフォルトは黒アミ。印刷によってはベタ黒塗りはちょっと怖いかなので、あまり長々したものには使わないほうがいいですね。

```
$ <b>ls /</b>
```

## 図

採番・キャプション付きの図の貼り付けはimageを使用します（[図2.1](#)）。図版ファイルは識別子とビルダが対応しているフォーマットから先着順に探索されます。詳細については[ImagePath](#)のドキュメントを参照してください。

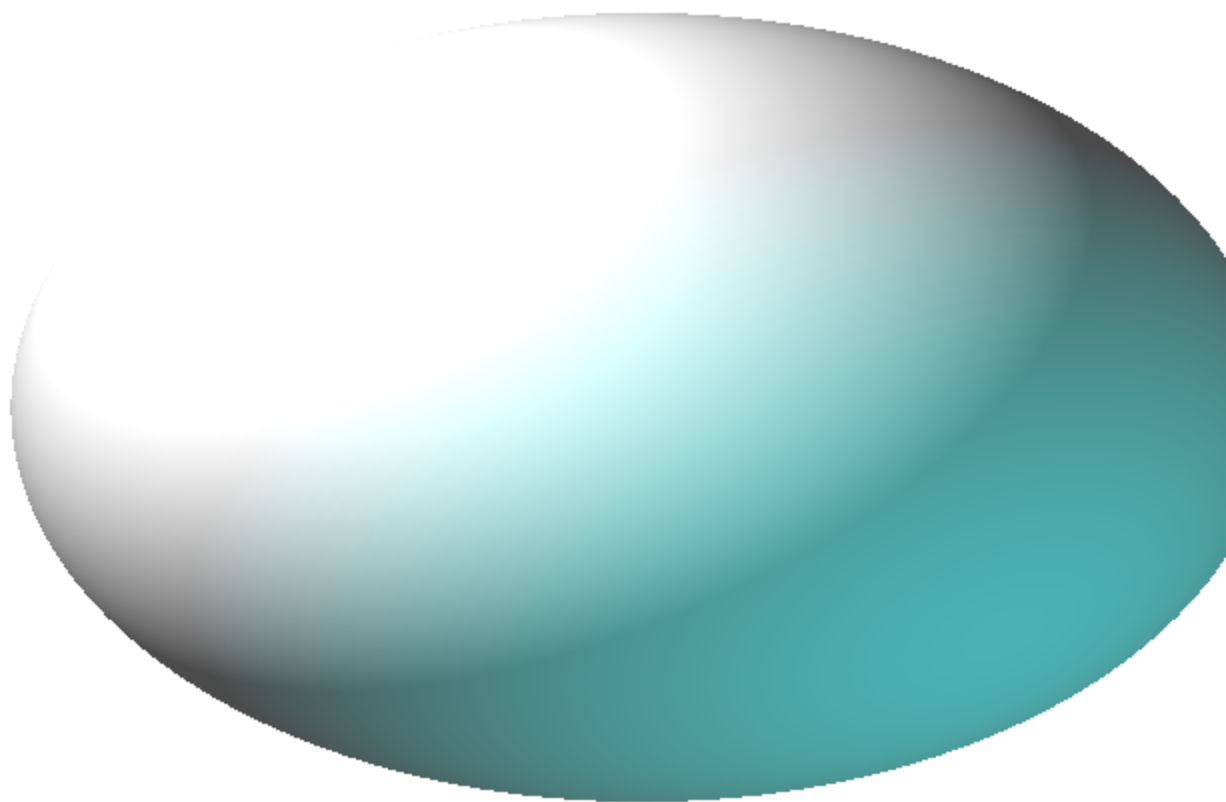
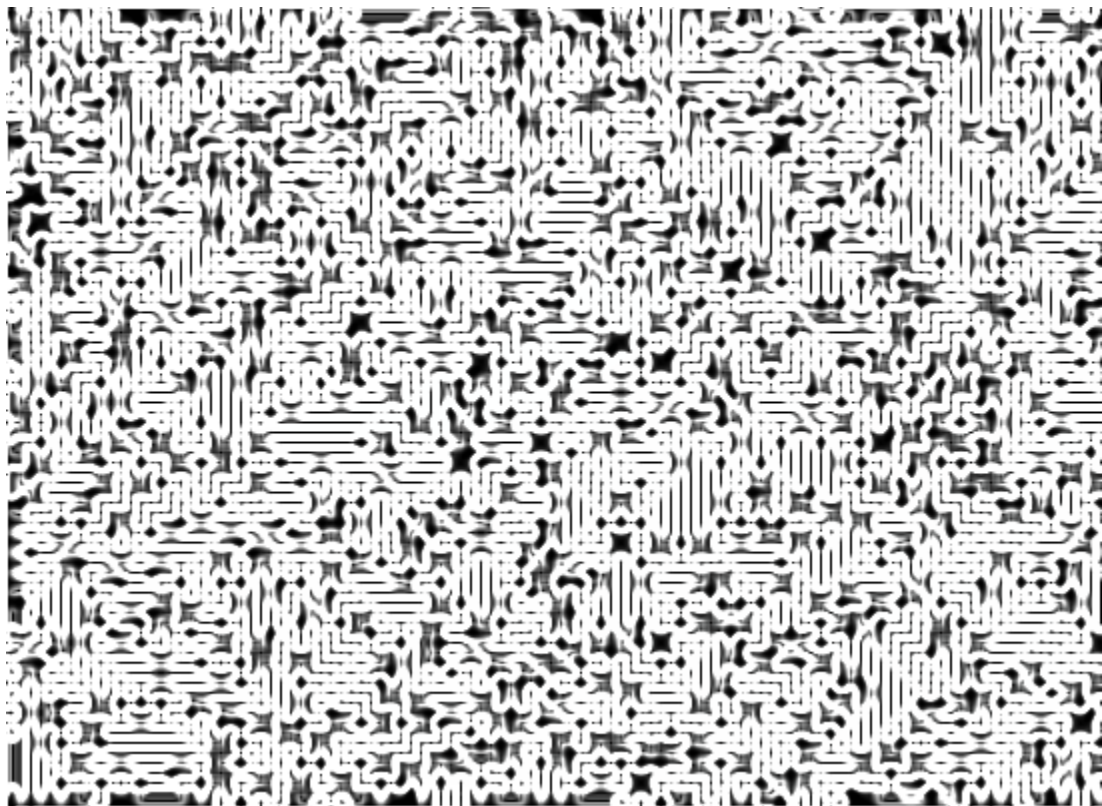


図2.1: ボール



採番なし、あるいはキャプションもなしのものはindepimageを使います。



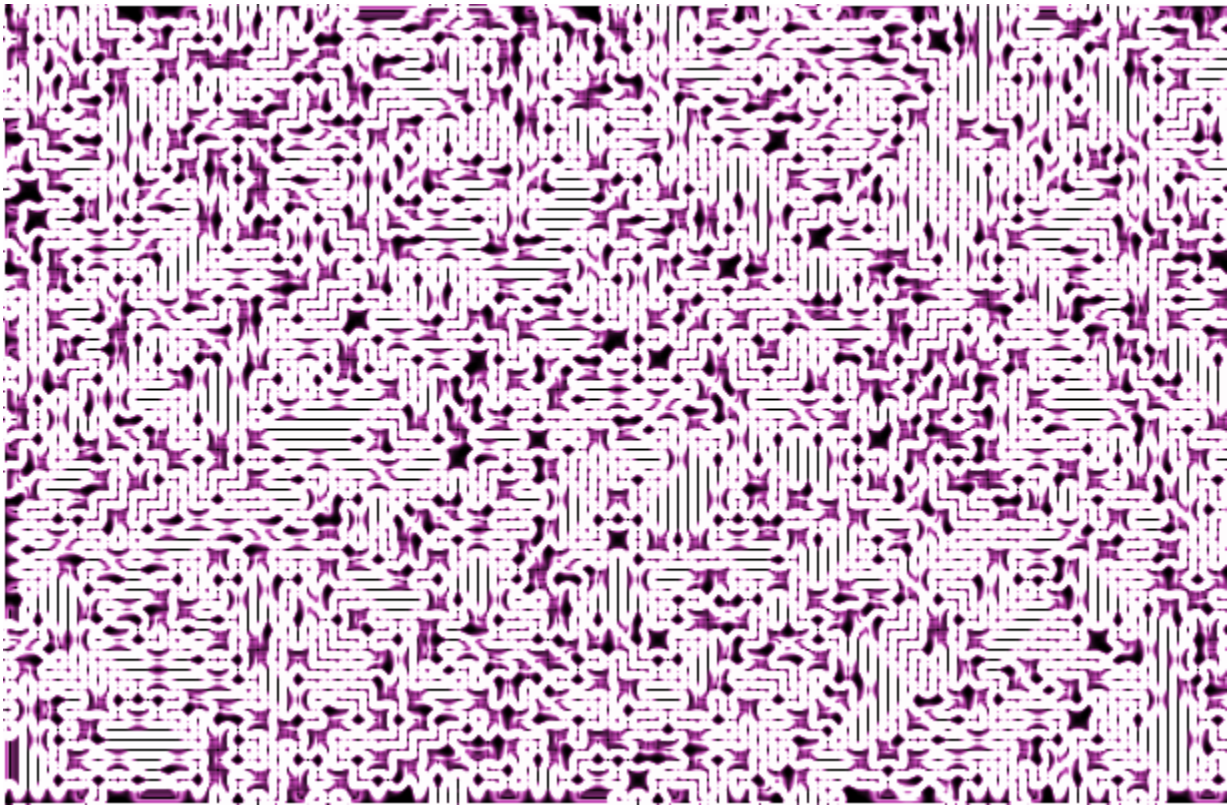


図: 採番なしキャプション

## 表

表はtableを使います。

表2.1: 表の例

[illegible]

採番はしたくないけれどもキャプションは指定したいという場合はemtableがあります。

キャプションを指定したいが採番はしたくない表

<u>AB</u>
<u>CD</u>

画像としておいて貼り付けたほうがよさそうな場合はimgtableを使います ([表2.2](#))。

表2.2: ポンチ表

A	B
人	猫

## 囲み記事

//{~//}の囲み記事の中には段落のみしか入らないものと想定します（箇条書きなどを入れたい場合はユーザーの責任で適宜、変換後ソースを加工してもらうことを前提とします）。

引用はquoteで表現します。上下アキ、左インデント（2文字くらい？）が入るのが一般的でしょうか。

ここに引用文。太字***bold italic***イタ等幅code

2行目の引用文。

中寄せはcenteringです。

中寄せ本文。太字***bold italic***イタ 等幅code

2行目の中寄せ本文。

右寄せはflushrightです。

右寄せ本文。太字***bold italic***イタ 等幅code

2行目の右寄せ本文。

ノートnote。以降、キャプションあり/なしのパターンがあります。表現については結局紙面デザインに応じて千差万別になるものだと思いますので、基本デザインとしては何か囲み要素だとわかって、カスタマイズしやすければよい、という程度です。

ノートの例太字***bold italic***イタ 等幅code

ノート1。太字***bold italic***イタ 等幅code

ノート2。

ノート。太字***bold italic***イタ 等幅code

メモ memo。

メモの例太字**bold***italic*イタ 等幅code

メモ1。太字**bold***italic*イタ 等幅code

メモ2。

メモ。太字**bold***italic*イタ 等幅code

Tips tip。

Tipsの例太字**bold***italic*イタ 等幅code

Tips1。太字**bold***italic*イタ 等幅code

Tips2。

Tips。太字**bold***italic*イタ 等幅code

情報 info。

情報の例太字**bold***italic*イタ 等幅code

情報1。太字**bold***italic*イタ 等幅code

情報2。

情報。太字**bold***italic*イタ 等幅code

注意 warning。

注意の例太字***italic***イタ 等幅code

注意1。太字***italic***イタ 等幅code

注意2。

注意。太字***italic***イタ 等幅code

重要 important。

重要の例太字***italic***イタ 等幅code

重要1。太字***italic***イタ 等幅code

重要2。

重要。太字***italic***イタ 等幅code

警告 caution。

警告の例太字***italic***イタ 等幅code

警告1。太字***italic***イタ 等幅code

警告2。

警告。太字***italic***イタ 等幅code

注意 notice。

注意の例太字***italic***イタ 等幅code

注意1。太字***italic***イタ 等幅code

注意2。

注意。太字**bold***italic*イタ 等幅code

## 2.4 インライン命令

### 書体

本文での……キーワード**キーワード (keyword)**、太字**b**太字、イタリック*i*イタリック、等幅 **tt**等幅、強調**strong**強調、強調*em*強調、下線u下線、等幅 **code**等幅、等幅太字 **ttb**等幅太字、等幅イタリック *tti*等幅イタリック、網カケamiアミ、挿入ins挿入、削除~~del~~削除、16進UTF文字指定あ

ルビ<sup>がくぜん</sup>愕然

## 第3章 Markdownファイルの取り込み

pandoc および pandoc2review を使って、Markdown 形式で書かれたファイルをあたかも Re:VIEW の1つの章のように扱うことができます。

Installing pandoc に従って pandoc をまずご利用の環境にインストールしておいてください。

pandoc2review 機能を有効にするには、config.yml の contentdir パラメータを以下のように設定します (設定済み)。

```
contentdir: _refiles
```

そして、catalog.yml に Markdown ファイルの拡張子を `.re` に変更した名前で登録します。たとえばこのファイル自体を登録するのは以下のようになります。

PREDEF:

- preface.re

CHAPS:

- article.re
- markdown.re

APPENDIX:

POSTDEF:

- contributors.re

pandoc の Markdown 記法を逸脱しない範囲で、Re:VIEW 文法の大半を代替できます。pandoc2review における Markdown 形式処理の注意事項 を参照してください。