

前書き

PREDEF 内に列挙したものは前付として章採番なしです。後付の POSTDEF も同様。‘あ’ い ‘A’B、
‘あ’ い ‘A’B

PREDEF 内/POSTDEF のリストの採番表記は「リスト 1」のようになります: リスト 1
(正確には i18n.yml/locale.yml の format_number_header_without_chapter が使われます)

リスト 1: main()

```
‘あ’ い ‘A’B
int
main(int argc, char **argv)
{
    puts("OK");
    return 0;
}
```

図（図 1）、表（表 1）も同様に章番号なしです。

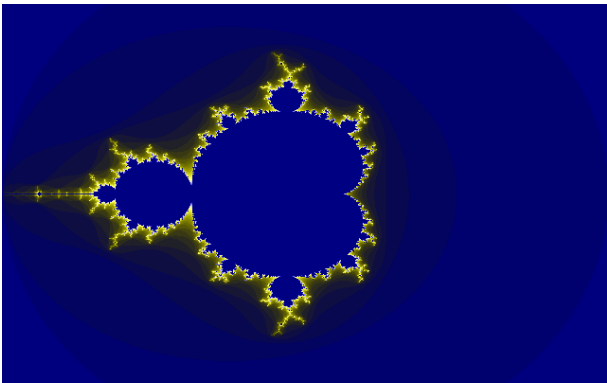


図 1.1 フラクタル

表 1.1 前付表

A	B
C	D

目次

前書き	i
第 I 部 部扉見出し	1
第 1 章 章見出し	3
1.1 節見出し	3
1.1.1 項見出し……に脚注を入れると <i>TeX</i> ではエラー	3
1.2 長い節見出し	4
1.2.1 長い項見出し	4
採番しない項見出し	4
1.2.2 採番する項見出し	4
<code>nodisp</code> で隠れた見出し	4
1.3 簡条書き	4
1.3.1 ナカグロ簡条書き	5
1.3.2 番号簡条書き	5
1.3.3 用語リスト	5
第 II 部 部見出し	7
第 2 章 長い章見出し	11
2.1 ブロック命令	11
2.1.1 ソースコード	11
2.1.2 図	12
2.1.3 表	12
2.1.4 囲み記事	13
2.2 LaTeX 式	17
2.3 インライン命令	18
2.3.1 書体	18
2.3.2 参照	19
2.3.3 参考文献	20
2.3.4 索引	20
第 3 章 コラム	21

付録 A 付録の見出し	25
A.1 付録の節	25
A.1.1 付録の項	25
参考文献	27

第Ⅰ部 部扉見出し

第 1 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 a pen. 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。

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(そもそもこういうまたぎ行為は筆者の好みではありませんが)。

1.1 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(内部的にはレベル6までであるけれども非推奨で、一部のビルダでは動かない)。

- = : 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¹⁾。デフォルトは2(X.Xまで)ですが、このリファレンスドキュメントでは一応4(X.X.X.X)まで採番を試みています。

1) 前述したように `PREDEF`、`POSTDEF` の場合は採番しません。

1.1.1 項見出し……に脚注を入れると TeX ではエラー

²⁾という脚注を見出し箇所に入れようとする TeX では (footnotemark/text 化しない限り) エラーになります。見出しにそういうものを入れるべきではない、といえはそれまでですが、見出しにはインラインでの装飾タグが入る可能性があります。

1.1.1.1 段見出し

ここまで採番することがあるケースがあるか、というたとまにあったりします。

レベル5 見出し さすがにこのあたりのレベルは採番はしないですね。紙面では、:による description 箇条書きでは大きくなりすぎるような規模の場合にこのレベルの見出しを代用することがあります。

あくまでも見出しなので、見出しの行にそのまま段落が続いてしまう見た目は期待と違います。

1.2 長い節見出し

1.2.1 長い項見出し

1.2.1.1 長い段見出し

長いレベル5 見出し 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・目次を含める、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

1.2.2 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) 本当は項の脚注

1.3 箇条書き

1.3.1 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）が入ることは標準では対応しておらず、どうしてもそういうのが必要な場合は途中でフックして変換後ソースを書き換えることになります。

1.3.2 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`

olnum で一応番号が変更可能なことを期待していますが、Web ブラウザだとだめなことが多いかもしれません。

- 10 箇条書き 10
- 11 箇条書き 11

1.3.3 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。

Alphabold 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。

bold 太字 *italic イタ*

```
: Alpha@<b>{bold 太字}@<i>{italic イタ}@<tt>{等幅 code}
      @<i>{DEC}の作っていた@<b>{RISC CPU}。@<i>{italic イタ}@<tt>{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。@<br>{}
      派生として POWER PC がある。

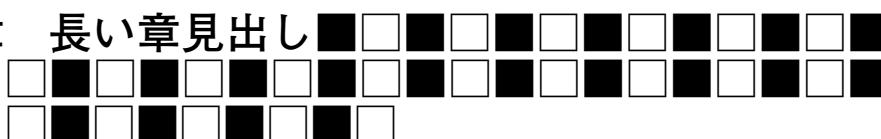
: SPARC
      Sun が作っている RISC CPU。
      CPU 数を増やすのが得意。←ふきだし説明
```

説明文に複数の段落を入れることは構文上できないので、`@
{}`を入れて改行することで代替します。

第II部 部見出し

部のリード。部は `catalog.yml` で直接指定することもできるし（その場合は見出しのみとなる）、`re` ファイルで内容を記述することもできるようにしています。部の番号表記を I にしたり 1 にしたりするのは `i18n.yml/locale.yml` の定義です。

第2章 長い章見出し



2.1 ブロック命令

2.1.1 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¹⁾

```
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。キャプションはあつたりなかつたりします。

```
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');
```

1) コードハイライトは外部パッケージに委任しています。TeX では `jlisting`、HTML では `Rouge` ?

ソースコード引用を主ターゲットにするのには一応 source というのを用意しています²⁾。

hello.rb

```
puts 'Hello'
```

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とほとの数に留めておいたほうがい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にはちょっと怖いかもなので、あまり長々したものには使わないほうがいいですね。

```
$ ls /
```

2.1.2 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。

³⁾ 本当は image のキャプションに footnote を付けたいのですが、TeX ではエラーになりますね。厳しい……。

(いちおう、config.yml ファイルに `footnotetext: true` を追加すれば、footnotemark/footnotetext を使うモードになりますが)

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。

2.1.3 表

表は table を使います。表 2.1

table もキャプション・セル内含めて TeX では脚注できないですね… (要 footnotemark) 本当は→⁴⁾はキャプション内。TeX だとセル内の脚注は脚注文書が消えています。

表 2.1 表の例

A	B	C
D	E 太字 bolditalic イタ等幅 code	F G
H	I ⁵⁾	長いセルの折り返し

2) 書籍だと、いろいろ使い分けが必要なんですよ…… (4、5 パターンくらい使うことも)。普通の用途では list と emlist で十分だと思いますし、見た目も同じでよいのではないかと。TeX の抽象タグ名は変えてはいます。

3) GIMP のフィルタで作成。
footnote 内改行

4) 現状、表の alignment とか join とかは Re:VIEW 記法では対応していません。筆者自身の制作では <https://kmuto.jp/d/?date=20120208#p01> みたいな手法を使っています。

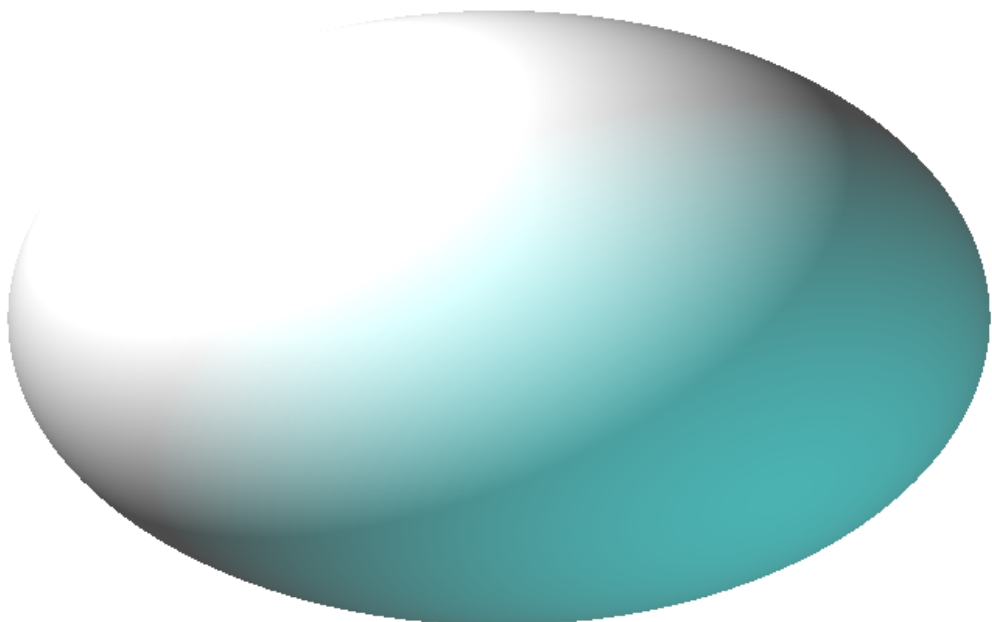


図 2.1 ボール

TeX向けには`tsize`でTeX形式の列指定自体は可能です。以下は`//tsize[|latex|p{10mm}|p{18mm}|p{50mm}]`としています。

A	B	C
D	E 太字 <i>bolditalic</i> イ タ等幅 <code>code</code>	F G
H	I	長いセルの折り返し ■■■■■■■■■■ ■■■■■■■■■■ ■■■■■■■■■■

TeXの普通のクラスファイルだと、列指定は`l,c,r,p`（幅指定+左均等）しかないので、幅指定+左寄せ（均等なし）、幅指定+中寄せ、幅指定+右寄せの指定ができると嬉しそうです。

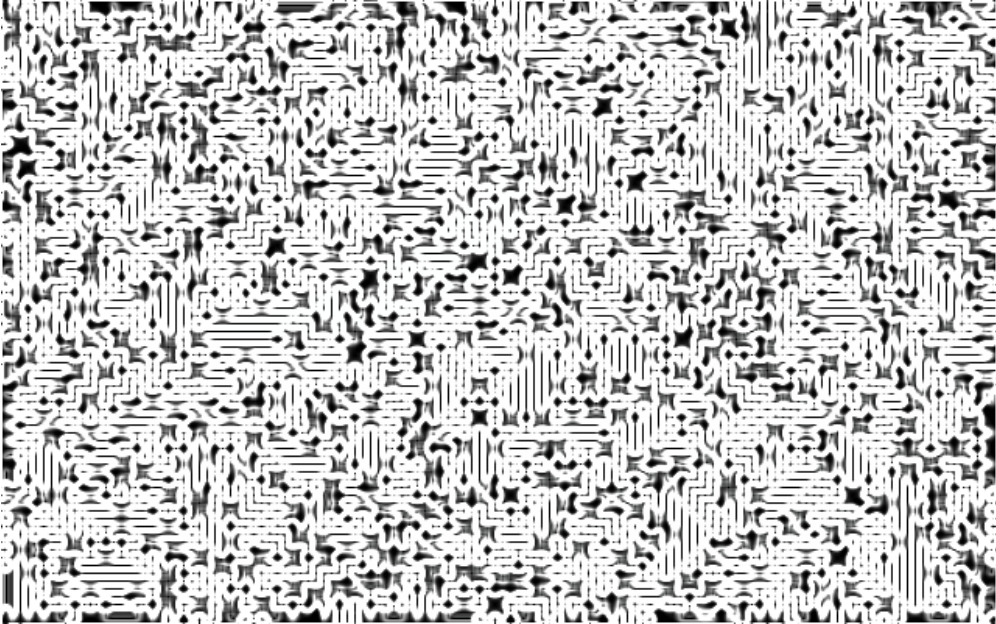
あとは縦に長い表がTeXだとそのままはみ出してしまうので`longtable`があるけれどもそれはまた問題がいろいろあり……。

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。

表 2.2 *

キャプションを指定したいが採番はしたくない表	A	B
	C	D

画像に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）。



2.1.4 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 ***bolditalic*** 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 ***bolditalic*** 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 ***bolditalic*** 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。

NOTE ノートの例太字 ***bolditalic*** イタ等幅 code

ノート 1。太字 ***bolditalic*** イタ等幅 code

ノート 2。

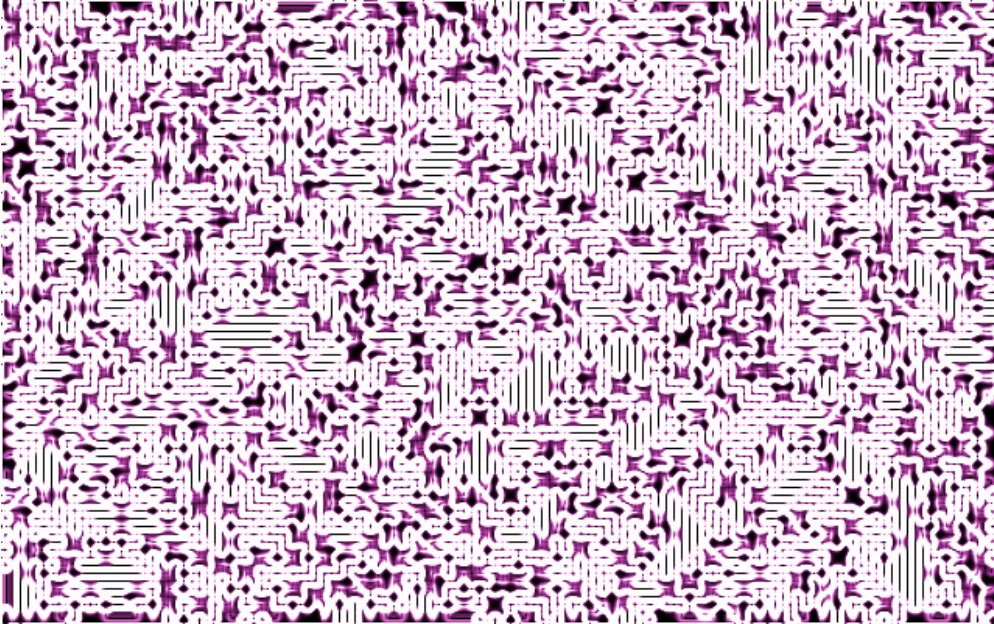


図 2.2 0

図: 採番なしキャプション

NOTE

ノート。太字 *bolditalic* イタ等幅 code

メモ memo。

MEMO メモの例太字 *bolditalic* イタ等幅 code

メモ 1。太字 *bolditalic* イタ等幅 code

メモ 2。

MEMO

メモ。太字 *bolditalic* イタ等幅 code

Tips tip。

Tips Tips の例太字 *bolditalic* イタ等幅 code

Tips1。太字 *bolditalic* イタ等幅 code

Tips2。

表 2.3 ポンチ表

A	B
人	猫

Tips

Tips。太字 *bolditalic* イタ等幅 code

情報 info。

INFORMATION 情報の例太字 *bolditalic* イタ等幅 code

情報 1。太字 *bolditalic* イタ等幅 code

情報 2。

INFORMATION

情報。太字 *bolditalic* イタ等幅 code

注意 warning。

WARNING! 注意の例太字 *bolditalic* イタ等幅 code

注意 1。太字 *bolditalic* イタ等幅 code

注意 2。

WARNING!

注意。太字 ***bolditalic*** イタ等幅 code

重要 important。

IMPORTANT! 重要の例太字 *bolditalic* イタ等幅 code

重要 1。太字 ***bolditalic*** イタ等幅 code

重要 2。

IMPORTANT!

重要。太字 ***bolditalic*** イタ等幅 code

警告 caution。

CAUTION! 警告の例太字 *bolditalic* イタ等幅 code

警告 1。太字 ***bolditalic*** イタ等幅 code

警告 2。

CAUTION!

警告。太字 ***bolditalic*** イタ等幅 code

注意 notice。

NOTICE 注意の例太字 *bolditalic* イタ等幅 code

注意 1。太字 ***bolditalic*** イタ等幅 code

注意 2。

NOTICE

注意。太字 ***bolditalic*** イタ等幅 code

脚注が入ることもあり得ます。

NOTICE 脚注がある注意^a

^a notice の見出し側脚注です。

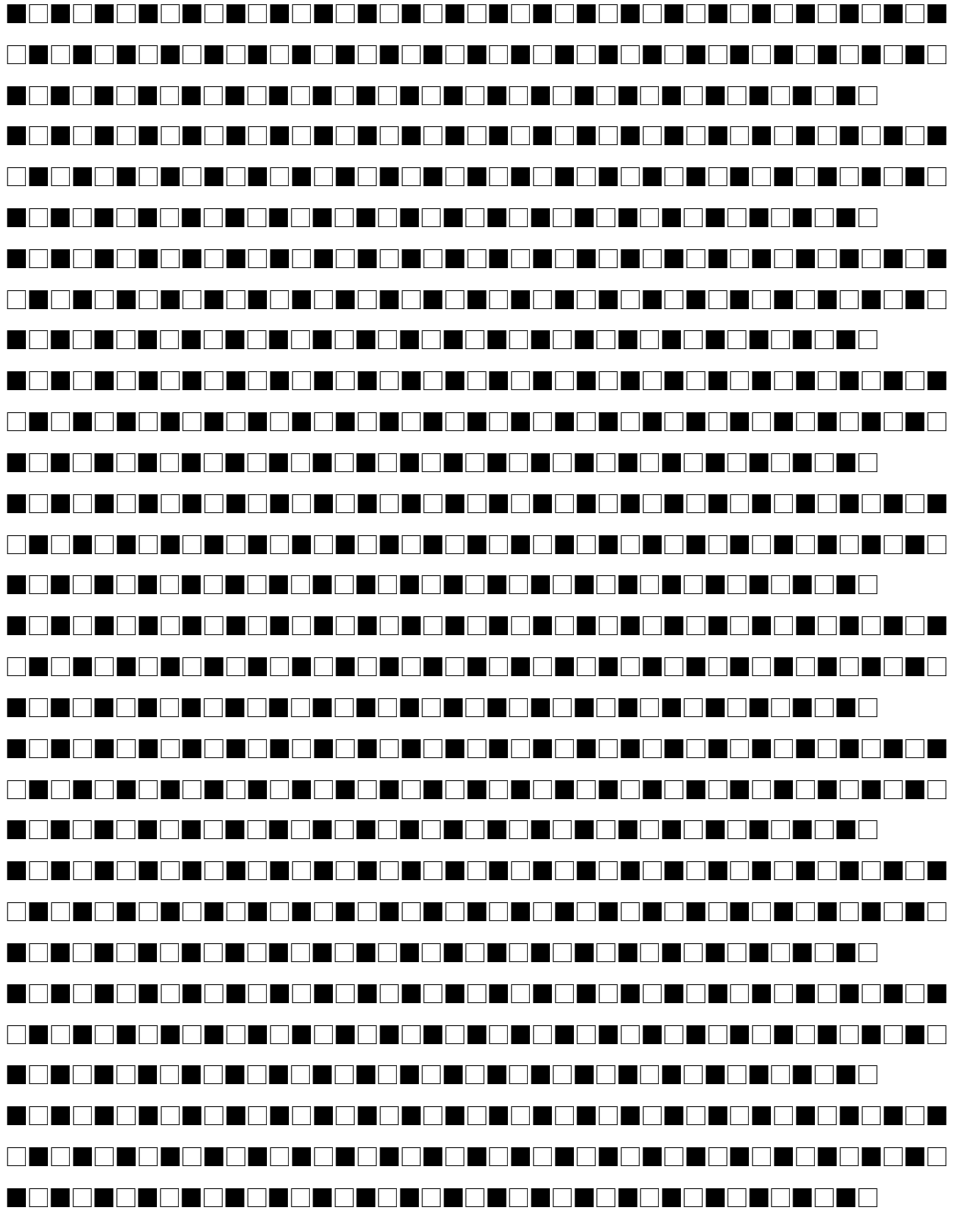
こちらにも脚注^a

^a notice の文章側脚注です。

b コラム内からの脚注です。

コラムその2

長い長いコラム。



付録 A 付録の見出し

A.1 付録の節

A.1.1 付録の項

A.1.1.1 付録の段

リスト A.1、図 A.1、表 A.1

リスト A.1: Hello

```
os.println("Hello");
```

表 A.1 付録表、見出し行なし（左 1 列目を見出しと見なす）

a	1
b	2
c	3

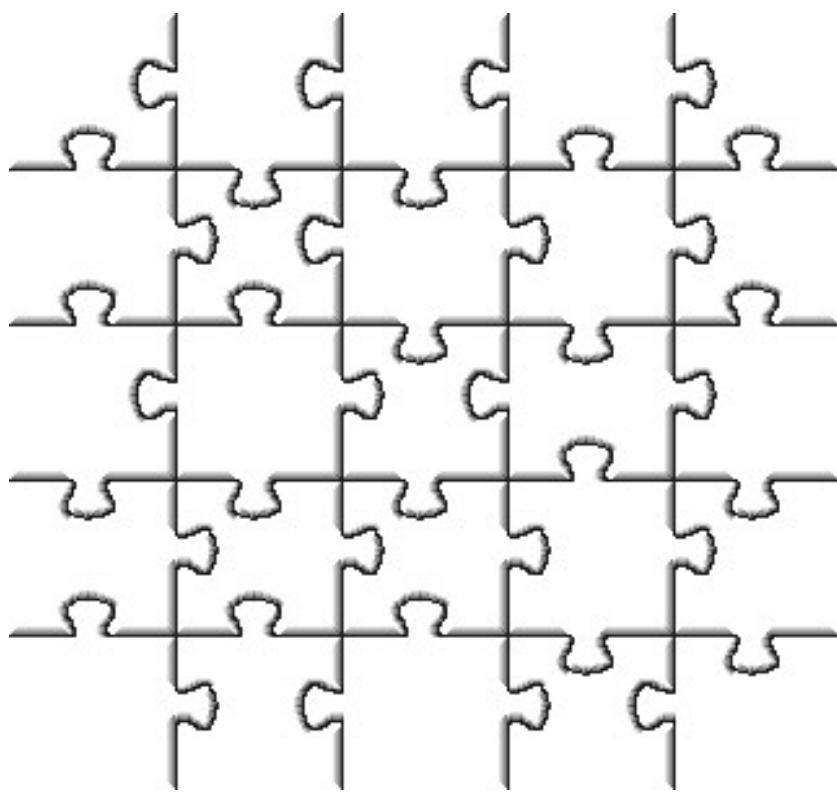


図 A.1 パズル

参考文献

[1] Lins, 1991

Refael D. Lins. A shared memory architecture for parallel study of algorithms for cyclic reference_counting. Technical Report 92, Computing Laboratory, The University of Kent at Canterbury , August 1991

索引

!<>"\$%&()~|=|,./\{}?_[]*.;+-%#()’“^, 20

index, 20

Mecab, 20

索引, 20

 hidx, 20

 idx, 20

辞書ファイル, 20