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。
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()

```
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。

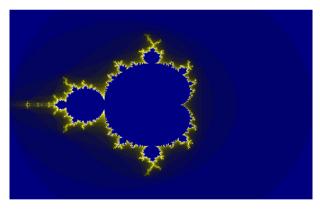


図 1.1 フラクタル

表 1.1 前付表



i

目次

| - 1 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 TeX 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 |
| nodisp で隠れた見出し・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | 4 |
| 1.3 箇条書き ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | 4 |
| 1.3.1 ナカグロ箇条書き・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | 5 |
| 1.3.2 番号箇条書き・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | 5 |
| 1.3.3 用語リスト・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | 5 |
| 第 II 部 | 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1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。

段落は普通に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!', 'Zh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.X.X) 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。

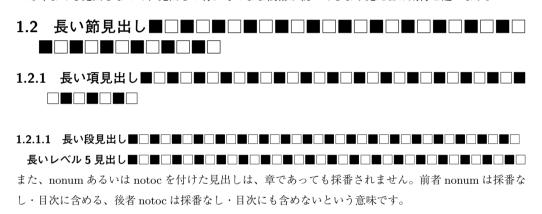
1.1.1 項見出し……に脚注を入れると TeX ではエラー

²⁾という脚注を見出し箇所に入れようとすると 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、といえばそれまでですが。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、というとた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。



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

入れ子ナカグロ筒条書きもあります。

- 簡条書き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
 - 筒条書き 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@{bold 太字}@<i>{italic イタ}@<tt>{等幅 code}

@<i>{DEC}の作っていた@{RISC CPU}。@<i>{italic イタ}@<tt>{等幅 code} 浮動小数点数演算が速い。

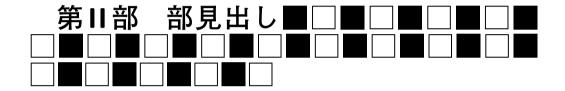
: POWER

IBM とモトローラが共同製作した RISC CPU。@
{} 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。



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 1 なるのは 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 $\mathcal O$ hello $\mathfrak I$ - $\mathfrak F^{1)}$

```
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');

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どほど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 \rightarrow 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

- 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。

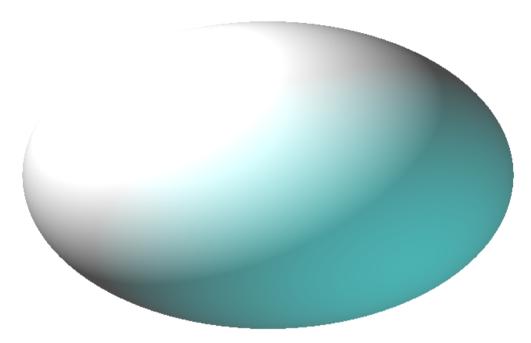


図 2.1 ボール

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。

| _ | 3 1 3 7 5 | | | | | | | |
|---|-----------|----------------------|-------------------|--|--|--|--|--|
| | A | В | C | | | | | |
| | D | E 太 字 | F G | | | | | |
| | | bold <i>italic</i> ≺ | | | | | | |
| | | タ等幅 code | | | | | | |
| | Н | Ι | 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……。

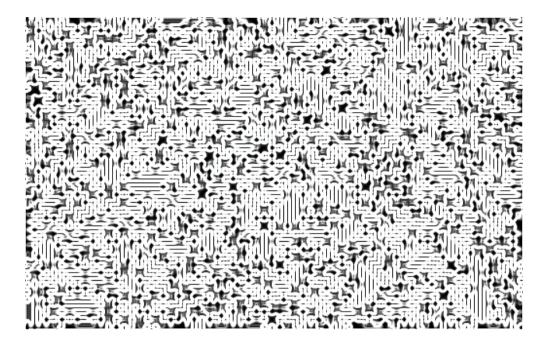
採番はしたくないけれどもキャプションは指定したいという場合は emtable があります。

 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 **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 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 **bold** *italic* イタ等幅 code

ノート 2。

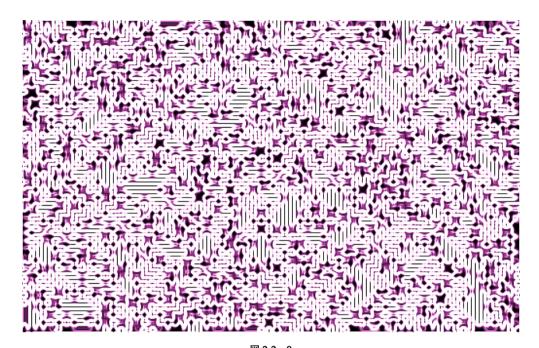


図 2.2 0 図: 採番なしキャプション

NOTE

ノート。太字 bold italic 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 **bold** italic イタ等幅 code

メモ 2。

мемо

メモ。太字 bolditalic イタ等幅 code

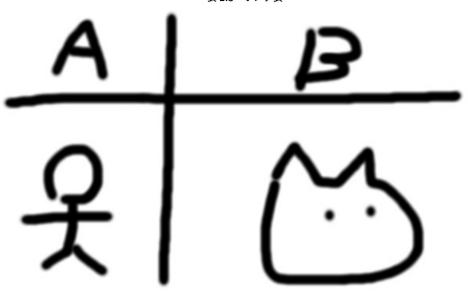
$Tips\ tip_\circ$

Tips Tips の例太字 bolditalic イタ等幅 code

Tips1。太字 bolditalic イタ等幅 code

 $Tips2_{\circ}$

表 2.3 ポンチ表



Tips

Tips。太字 **bold** *italic* 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 bold italic イタ等幅 code 情報 2。

INFORMATION

情報。太字 bold italic 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 bold italic イタ等幅 code 注意 2。

WARNING!

注意。太字 bold italic 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 **bold** italic 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 **bold** *italic* イタ等幅 code 警告 2。

CAUTION!

警告。太字 bold italic イタ等幅 code

注意 notice。

NOTICE 注意の例太字 bolditalic イタ等幅 code

NOTICE

注意。太字 bold italic 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。

2.2 LaTeX 式

LaTeX 式は TeX 紙面以外は保証されません。EPUB では MathML (mathml: true) を使えますが、表現や互換性が不足しており、LaTeX をバックエンドとして画像化する imgmath: true のほうがよさそうです。

$$\sum_{i=1}^{n} f_n(x)$$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

式採番がほしいケースは多々発生しているので、標準の文法を拡張する必要があるように思っています (現状、review-ext.rb でなんとかしていることが多いです)。

段落中の式は $E=mc^2$ というシンプルなものならまだよいのですが、 $\sinh^{-1}x=\log(x+\sqrt{x^2+1}$ のような形だと $\}$ のエスケープで読みにくめです。今のところ Ruby にあるようなフェンス文法を実装するのも難しいですね。 $\sum_{i=1}^n\sum_{j=1}^n$

2.3 インライン命令

2.3.1 書体

本文での……キーワード**キーワード**(keyword) 6)、太字 \mathbf{b} 太字、イタリック i イタリック、等幅 $\mathbf{t}\mathbf{t}$ 等幅、強調 \mathbf{strong} 強調、強調 \mathbf{em} 強調、下線<u>u</u>下線、等幅 \mathbf{code} 等幅、等幅太字 $\mathbf{t}\mathbf{t}\mathbf{b}$ 等幅太字、等幅イタリッ

- kw, b, strong, em は同じ書体でよいでしょう。
- tt、code は同じ書体でよいでしょう。
- ami はコードブロックの中で使うことを想定しています。

⁶⁾ キーワードのカッコは太字にしないほうがいいのかなと思いつつあります(手元の案件では太字にしないよう挙動を変えて しまっているほうが多い)。

箇条書き内での……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 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、16 進 UTF 文字指定あ、インラインアイコ



表内での……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>u</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、16 進 UTF 文字指定あ、

コードブロック内では対応装飾は減らしてよいと考えます。代わりに balloon が追加されます。

キャプション内での……キーワード**キーワード**(keyword)、太字 \mathbf{b} 太字、イタリック i イタリック、等幅 \mathbf{t} t 等幅、強調 \mathbf{strong} 強調、強調 \mathbf{em} 強調、下線 \mathbf{u} 下線、等幅 \mathbf{code} 等幅、等幅太字 \mathbf{ttb} 等幅**イタリック** \mathbf{tti} 等幅**イタリック**、網カケ \mathbf{ami} アミ、16 進 \mathbf{UTF} 文字指定あ、インラインアイコン



コードブロック内での……

太字 b 太字

イタリック i イタリック

下線u下線

網カケ ami アミ ←ふきだし説明

2.3.2 参照

• 章番号:第1章、付録A、第II部、

| • | 章題:章見出し、部見出し■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□ |
|---|--|
| | ■□■□、付録の見出し、参考文献 |
| • | 章番号+題:第2章「長い章見出し■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□■□ |
| | |

■□■□■□■□■□■□■□■□」、付録 A「付録の見出し」、「参考文献」

節や項への参照は hd を使います。

- 「2.1 ブロック命令」の「2.1.2 図」
- 「2.3.2 参照 |
- コラム参照 コラム 「コラムその 2」

他章への図表リスト参照の例です (リスト 1、図 1、表 1、リスト A.1、図 A.1、表 A.1)。

なお、この「.」区切りなどのフォーマットは i18n.yml あるいは locale.yml でカスタマイズされ得る (format_number、format_number_header、format_number_without_chapter、format_number_header_without_chapter) ので、スタイルで固定化するのは避けるべきです。

label で定義したラベルへの参照の例です。EPUB だと?? TeX だと@<href>{inlineop} 。互換性がないのは気味が悪いですね。

2.3.3 参考文献

参考文献 bib.re ファイルへの文献参照は、[1] とします。

2.3.4 索引

索引は TeX と IDGXML 以外では妥当な動作を定義していません。idx は文中にも表示し、hidx は文中からは隠した形の索引にします。読みは Mecab があればそちらを使いますが、辞書ファイルを直接定義することもできます。

idx, hidx いずれも=見出しの中には入れないようにし、後続の段落先頭に hidx で入れるように注意します (入れてしまうと目次などがおかしくなります)。

第3章 コラム

コラムは見出しを流用する形で作成され、//note などのブロックでは普通には表現できない、 筒条書きや図表・リストなどを内包する囲み記事を作成することを目的としています。

コラム見出し1太字b太字、イタリックiイタリック、等幅 tt 等幅

aを見出しに入れたときになぜか webmaker は失敗するようです。

コラム段落のキーワード**キーワード**(keyword)、太字 \mathbf{b} 太字、イタリック i イタリック、等幅 $\mathbf{t}\mathbf{t}$ 等幅、強調 \mathbf{strong} 強調、強調 \mathbf{em} 強調、下線 $\mathbf{a}\mathbf{m}$ 、等幅 \mathbf{code} 等幅、等幅太字 $\mathbf{t}\mathbf{t}\mathbf{b}$ 等幅



字、等幅イタリック tti 等幅イタリック、インラインアイコン



リスト 3.1: コラムのリスト

puts "Re:VIEW is #{impression}."

//image[img3-1][適当に作ったコラム内画像]{ は TeX では現状、エラーになります。

//indepimage[img3-1][適当に作ったコラム内画像]{ //}

//table[tab3-1][コラム表]{ もだめですね。

A B

C D

- ・ 箇条書き1
 - 箇条書き 1-1
 - * 箇条書き 1-1-1
- 1 番号箇条書き1
- 2 番号簡条書き2

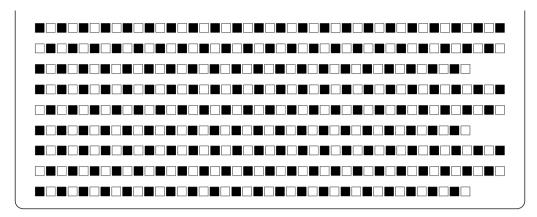
説明文見出し

説明文の説明

a コラム見出しの脚注です。

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。



参照はコラム 「コラムその 2」です。

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 1b 2

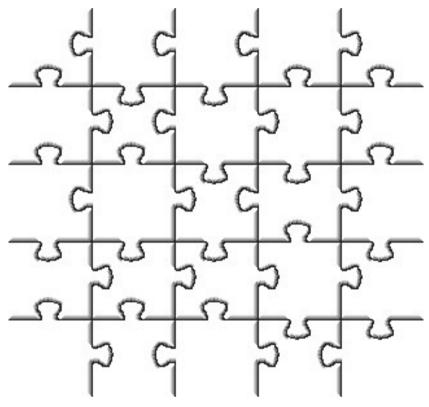


図 A.1 パズル

参考文献

[1] Lins, 1991

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