Enumitem パッケージ

@monaqa

目次

1. パッケージの概要

enumitem は SATYSFI にて豊富な箇条書きや番号付きのリストを提供するパッケージです。標準にもすでに itemize という名前のパッケージが用意されているものの、本パッケージではより自由度の高い箇条書きを提供します。具体的には、以下のような箇条書きを書くことができます：

- デフォルトで豊富なスタイルのラベルを選択できる
- 番号付き箇条書き環境をネストさせることができる
- ネストごとに箇条書きのスタイルを変更できる
- 箇条書きのラベルの体裁を項目ごとに変更できる
- 定義リストを作成できる
- ユーザ自身がスタイルを拡張できる

本ドキュメントは enumitem パッケージ v3.0.0 の仕様及び使い方を述べたものです。旧バージョン、すなわちメジャーバージョンが 0, 1, 2 のいずれかであるものとの互換性は無く、たとえコマンド名が同じであっても引数のルールが異なることなどがあります。バージョンの違いに注意してください。

1.1. 用語集

ここでは以下のような用語を用います。

箇条書き　いくつかの単語や文章、段落などを分けて書き並べたリストのこと。ここでは本文と独立の段落で組むものを扱う。

項目　箇条書きを構成する要素。

ラベル　箇条書きの開始を表すために先頭につける記号や文字の列。番号付きの箇条書きであれば “1.” や “(1)” といった文字列を指す。

番号付き箇条書き　番号や順序のわかる記号を用いてラベルがふられた箇条書き。

番号無し箇条書き　順序のない記号など用いてラベルがふられた箇条書き。

2. enumitem をはじめよう

2.1. enumitem パッケージのインストール

enumitem パッケージは Satyrographos によってインストールすることができます。

```
opam install satysfi-enumitem
satyrographos install
```

opam list satysfi-enumitem などのコマンドでバージョン 3 系がインストールされていることを確認したら、次に進みましょう。

2.2. パッケージの読み込み

SATYSFI の文書ファイルやヘッダファイルで外部のパッケージを用いるには以下のようにします。

```
@require: enumitem/enumitem
```

さらに、EnumitemFormatAlias というモジュールも open しておきましょう。 これは enumitem パッケージで定義されているモジュールの一つであり、箇条書きの体裁を変更するための関数、定義リスト専用のコマンドなど便利な機能がまとまっています。まとめると、文書ファイルの場合は以下のような文言をファイルに書けばよいということになります。「本文」のところは、各クラスファイルの書き方に従ってください。

```
@require: enumitem/enumitem

open EnumitemFormatAlias

in

（本文）
```

準備ができたら、本文のどこかに以下のように書いてみましょう。

```
+listing{
  * foo
  * bar
    ** barfoo
    ** barbar
```

```
  * baz
}
```

もし正常にパッケージが読み込まれていれば、以下のように出力されるはずです。

- foo
- bar
    - barfoo
    - barbar
- baz

2.3. 基本的なコマンドの使用

enumitem は +listing 及び +enumerate コマンドを提供します。 標準の itemize パッケージを使ったことがある人にとっては馴染み深いでしょう。 使い方は標準のコマンドと（オプション引数を除き）ほぼ変わりません。

```
+listing{
  * hoge
  * fuga
    ** fuga1
      *** fuga11
      *** fuga12
    ** fuga2
}
```

- hoge
- fuga
    - fuga1
        - fuga11
        - fuga12
    - fuga2

```
+enumerate{
  * hoge
```

```
  * fuga
    ** fuga1
      *** fuga11
      *** fuga12
    ** fuga2
}
```

1. hoge
2. fuga
   (a) fuga1
      (i) fuga11
      (ii) fuga12
   (b) fuga2

3. enumitem の設計思想

enumitem v3.0.0 は以下の思想を元に設計されています。

- $\mathrm{SAT_YSF_I}$ の文法に備わっている箇条書き用の構文は極力尊重する。
- 箇条書き用の構文でカバーできない機能については、専用のコマンドで対応する。
- 関数を用いてラベルの体裁を指定することで、ユーザによる拡張ができるようにする。

3.1. 2 種類の表現方法

簡潔に書ける箇条書きと自由度の高い箇条書きを両立させるため、enumitem パッケージでは箇条書きの木構造を表現する 2 種類の表現方法が用意されています。

- aaa
- bbb
  - bbb の子 1
  - bbb の子 2
    - bbb の子 2 の子 1
- ccc

- ccc の子 1
- ccc の子 2

1 つは SATYSF$_{\text{I}}$ の箇条書きを表す専用の構文をそのまま用いてネストを表現する方法。こちらは標準の itemize パッケージでも採用されているインターフェースであり、簡単に記述できるのが利点です。ただし、設定できる項目の自由度はあまり高くありません。便宜上、ここではこちらを**糖衣構文ベースの記法**と呼びます。

```
+listing?:(listing-default-label){
  * aaa
  * bbb
    ** bbb の子 1
    ** bbb の子 2
      *** bbb の子 2 の子 1
  * ccc
    ** ccc の子 1
    ** ccc の子 2
}
```

なお、+listing のオプション引数のデフォルトは listing-default-label であるため、上のコードの ?:(listing-default-label) に相当する箇所は省略することができます（次の例との比較のためあえてつけています）。

もう 1 つは SATYSF$_{\text{I}}$ の箇条書き専用構文を用いることなく、+itemize 及び +item のみを用いてブロックテキストベースで箇条書きのネストを表現する方法。星印 * だけでアイテムの始まりを示すことができた糖衣構文ベースの記法とは異なり、項目のたびに +item を書く必要があります。記法としては冗長になるものの、箇条書きを記述する際の自由度が高いというメリットがあります。こちらを**ブロックベースの記法**と呼びます。

```
+itemize(listing-default-label)<
  +item{aaa}<>
  +item{bbb}<
    +item{bbb の子 1}<>
    +item{bbb の子 2}<
      +item{bbb の子 2 の子 1}<>
    >
  >
  +item{ccc}<
```

```
    +item{ccc の子 1}<>
    +item{ccc の子 2}<>
  >
>
```

ここで示した 2 つのコードは、どちらも最終的には同様の結果となることに注意してください。実は 1 番目の糖衣構文ベースの記法は、まず内部でブロックベースの記法へと展開されてから処理されます。

3.2. ブロックベース記法の使いみち

上で示したコードは、糖衣構文ベースの記法でもブロックベースの記法でも、どちらでも実現可能な箇条書きでした。その場合は、より簡潔に書ける糖衣構文ベースの記法を用いるべきでしょう。 ブロックベースの記法の利点は自由度の高さにあります。 つまり、 糖衣構文ベースでは実現できない箇条書きも表現することができるのです。

1 つは箇条書きの途中で項目の体裁を変更すること。

- aaa

2. bbb
    - bbb の子 1
    - bbb の子 2
        - bbb の子 2 の子 1

3.ccc
    - ccc の子 1
    - ccc の子 2

- aaa
- bbb
    1. bbb の子 1
    2. bbb の子 2
        1. bbb の子 2 の子 1
- ccc

3.1.ccc の子 1

3.2.ccc の子 2

もう 1 つは、項目の下に任意のブロックテキストを配置できること。

```
+itemize(listing-default-label)<
  +item{ここは最初の段落です。ラベルは最初の段落の冒頭に付きます。}<
    +pn{
      ここは 2 番目の段落です。
      2 番目以降であれば自由にブロックテキストを挿入できます。
      以下はコードブロックを挿入する例。
    }
    +code(`aaa`);
    +item{このように、途中で箇条書きを挟むことができます。}<
      +item{当然ネストさせることもできます。}<>
    >
    +pn{
      箇条書きを挟んだ後、元の階層に戻って再び段落を再開することができます。
    }
  >
>
```

- ここは最初の段落です。ラベルは最初の段落の冒頭に付きます。

  ここは 2 番目の段落です。2 番目以降であれば自由にブロックテキストを挿入できます。以下はコードブロックを挿入する例。

  ```
  aaa
  ```

  - このように、途中で箇条書きを挟むことができます。
    - 当然ネストさせることもできます。

  箇条書きを挟んだ後、元の階層に戻って再び段落を再開することができます。

これらを応用すれば、定義リストのような箇条書きを実現することもできます。

定義項目 1　これが定義内容です。

ちょっと長めの定義項目 2　これがちょっと長めの定義内容です。 項目だけでなく本文も

少しだけ長めです。

だいぶ長いから改行挟んだほうが良い定義項目 3

こちらは改行を挟んでおり、本文も長めです。このように通常の段落を入れることができます。

3.3. 2 つの記法の併用

2 つの記法は組み合わせて使用することもできます。

```
+itemize(listing-default-label)<
  +item{aaa}<>
  +item{bbb}<
    +sublist(listing-default-label){
      * bbb の子 1
      * bbb の子 2
        ** bbb の子 2 の子 1
    }
  >
  +sublist(listing-default-label){
    * ccc
      ** ccc の子 1
      ** ccc の子 2
  }
>
```

```
+listing{
  * aaa
  * bbb
    \subitem(listing-default-label)<
      +item{bbb の子 1}<>
      +item{bbb の子 2}<
        +item{bbb の子 2 の子 1}<>
      >
    >
  * ccc
    \subitem(listing-default-label)<
      +item{ccc の子 1}<>
      +item{ccc の子 2}<>
```

```
    >
}
```

\sublist を用いれば、糖衣構文ベースの記法を（糖衣構文を直接使わず）ネストさせることができます。

```
+listing{
  * aaa
  * bbb
    \sublist(listing-default-label){
      * bbb の子 1
      * bbb の子 2
        \sublist(listing-default-label){
          * bbb の子 2 の子 1
        }
    }
  * ccc
    \sublist(listing-default-label){
      * ccc の子 1
      * ccc の子 2
    }
}
```

4. Gallery

4.1. ラベルの変更

```
```

- 水馬、赤いな。ア、イ、ウ、エ、オ。
  - 浮藻に小蝦もおよいでる。
- 柿の木、栗の木。カ、キ、ク、ケ、コ。
  - 啄木鳥こつこつ、枯れけやき。
- 大角豆に醋をかけ、サ、シ、ス、セ、ソ。
  - その魚淺瀬で刺しました。

一．　水馬、赤いな。ア、イ、ウ、エ、オ。

　　一．　浮藻に小蝦もおよいでる。

一．　柿の木、栗の木。カ、キ、ク、ケ、コ。

　　一．　啄木鳥こつこつ、枯れけやき。

一．　大角豆に醋をかけ、サ、シ、ス、セ、ソ。

　　一．　その魚淺瀬で刺しました。

一．　水馬、赤いな。ア、イ、ウ、エ、オ。

　　二．　浮藻に小蝦もおよいでる。

一．　柿の木、栗の木。カ、キ、ク、ケ、コ。

　　二．　啄木鳥こつこつ、枯れけやき。

一．　大角豆に醋をかけ、サ、シ、ス、セ、ソ。

　　二．　その魚淺瀬で刺しました。

1. 水馬、赤いな。ア、イ、ウ、エ、オ。

    1. 浮藻に小蝦もおよいでる。

4. 柿の木、栗の木。カ、キ、ク、ケ、コ。

    1. 啄木鳥こつこつ、枯れけやき。

9. 大角豆に醋をかけ、サ、シ、ス、セ、ソ。

    1. その魚淺瀬で刺しました。

1. 水馬、赤いな。ア、イ、ウ、エ、オ。

    1. 浮藻に小蝦もおよいでる。

4. 柿の木、栗の木。カ、キ、ク、ケ、コ。

    1. 啄木鳥こつこつ、枯れけやき。
9. 大角豆に醋をかけ、サ、シ、ス、セ、ソ。
    1. その魚淺瀬で刺しました。

1. 水馬、赤いな。ア、イ、ウ、エ、オ。
   続き：浮藻に小蝦もおよいでる。
4. 柿の木、栗の木。カ、キ、ク、ケ、コ。
   続き：啄木鳥こつこつ、枯れけやき。
9. 大角豆に醋をかけ、サ、シ、ス、セ、ソ。
   続き：その魚淺瀬で刺しました。

4.2. ラベルのユーザ定義

4.3. 複雑な箇条書き

- ✓：0x2713
- □：0x2501

4.4. 応用：定義リスト

5. 機能一覧

5.1. Enumitem モジュール

Enumitem モジュールは本パッケージの根幹となるコマンドを定義します。最も大切なのは +item コマンドであり、Enumitem パッケージで提供される主要なコマンドは全て +item を用いたブロックボックス列に展開されるようになっています。

+item の引数は以下のようになっています。

```
+item?:( itemfmt-self )({ text-body })?:( itemfmt-child )< children >
```

itemfmt-self （itemfmt、オプション）

自分自身の項目の体裁。省略した場合は、自身の親で設定されたフォーマットが用いられる。あくまで自分自身の項目のみに影響があり、後述の通り、自身の子要素には反映されない。

text-body (`inline-text`、必須)

自分自身の項目の本文。ラベルは本文の左端に付く。空のインラインテキストを指定することもできる。その場合はフォーマットの `display-label-with-empty-body` オプションによってラベルが付くかどうか変化する。

itemfmt-child (`itemfmt`、オプション)

子要素の項目の体裁のデフォルト値。省略した場合、子要素は自身の親で設定されたフォーマットを引き継ぐ（`itemfmt-self` の値は用いられない）。

5.2. `itemfmt` 型

`itemfmt` 型は `enumitem` パッケージに欠かせない型です。

foo

1.1.bar

- □ あああああ
- □ あああああ
- ☑ あああああ
- □ あああああ