

# bxluavienv パッケージ (v0.2c)

八登崇之 (Takayuki YATO; aka. “ZR”)

2012/06/05

## 1 概要

本パッケージは以下の機能を提供する。

- 中身を `verbatim` に処理するような  $\text{\LaTeX}$  環境 (以下「`verbatim` 入力環境」と呼ぶ) を容易に定義できるようにする。「環境の中身をそのまま Lua の関数に渡して処理させる」という用途を念頭に置いて、この場合、 $\text{\LaTeX}$  の範囲内で実装が可能になる。

前提環境

- $\text{\TeX}$  フォーマット:  $\text{Lua}\text{\LaTeX}$

依存パッケージ

- `etoolbox` パッケージ
- `luatexbase` パッケージ

## 2 動機

例えば、

```
\begin{timelinelist}
2010,4,10, 初版公開
2010,4,11, バグ修正 \verb|^{}|
2011,2,1, 第 2 版公開; Lua{\TeX}に対応
\end{timelinelist}
```

のように入力して、中身を Lua で整形して

- 2010 年 4 月 10 日: 初版公開
- 2010 年 4 月 11 日: バグ修正 (^{};
- 2011 年 2 月 1 日: 第 2 版公開;  $\text{Lua}\text{\TeX}$  に対応

のように出力する環境を実装したいとする。この場合、環境の中身の文字列をそのまま (改行や `\verb` 中の特殊記号も含めて) Lua の処理に渡す必要がある。一般的には、こういう処理の実現には複雑なマクロを書く

必要があるが、本パッケージの機能を用いると、この `timelinelist` 環境の定義 (Lua に処理を渡す部分) は

```
\DeclareVerbInputEnvironment{timelinelist}{\directlua{
  timelinelist(\VIEString{#1}) }}}
```

で済ませられる。

### 3 パッケージの読込

`\usepackage` で読み込む。オプションはない。

```
\usepackage{bxluavienv}
```

### 4 基本的な機能

- `\DeclareVerbInputEnvironment{<名前>}[<整数  $n$ >][<命令>]{<本体>}` :  $n - 1$  個の (非 verbatim な) 引数を伴う、「verbatim 入力環境」を指定の名前で定義する。環境の処理内容は本体部で指定され、ここは  $n$  個の引数をもつ命令の定義 (`\newcommand`) と同様の方式で記述する。一番最後の引数 ( $\#n$ ) は環境の中身の文字列を表し、それ以外の ( $n - 1$  個の) 引数は環境に渡された実際の引数を現す。 $n$  の既定値は 1 (つまり引数のない環境に対応する) である。`\命令` は引数フィルタの指定を表す (5 節を参照)。
- `\VIEString{<テキスト>}` : `luacode` パッケージの `\luastring` 命令と同様に、 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  のトークン列を Lua の文字列リテラルの形式に変換する (Lua コード記述の中で用いる)。  
ただし、`\DeclareVerbInputEnvironment` の本体部で「環境の中身を表す引数」に対して用いることが想定されていて、「先頭に  $\text{\textasciimacron}$ (CR) がある場合は除去する」「それ以外の  $\text{\textasciimacron}$ (CR) を  $\text{\textasciimacron}$ (LF) に変換する」という加工を行う。

環境の引数は verbatim の扱いにならない。

使用例 最も単純な例は 2 節のところで挙げた。この例の場合、Lua 関数 `timelinelist(s)` は環境の中身の文字列を引数  $s$  として渡されるので、それを加工して以下のような出力を `tex.print()` を用いて行えばよい。

```
\begin{list}{\textendash}{\textbullet}
\item 2010 年 4 月 10 日 : 初版公開
\item 2010 年 4 月 11 日 : バグ修正 \verb|(\textasciimacron)|
\item 2011 年 2 月 1 日 : 第 2 版公開 ; Lua{\textasciimacron}\TeX}に対応
\end{list}
```

ここでもし、

```
\begin{timelinelist}{\textbullet}
2010,4,10, 初版公開
\end{timelinelist}
```

のように項目頭の記号を引数として指定可能にする場合は、`timelinelist` 環境の定義を

```
\DeclareVerbInputEnvironment{timelinelist}[2]{\directlua{
  timelinelist(\luastringN{#1},\VIEString{#2}) }}
```

のようにする。この場合、Lua 関数 `timelinelist(i,s)` は引数 `i` に `timelinelist` 環境の引数（項目頭の記号）を受け取ることになる。

## 5 引数フィルタ

- `\DeclareVIEArguments{<引数列>}` : 引数フィルタのマクロの中で使用し、`verbatim` 入力環境の本体に渡る引数を指定する。

引数フィルタ（`\DeclareVerbInputEnvironment` の第 2 引数）が無指定の場合、環境に与える引数は単純な必須引数の列に限られる。例えば、

```
\DeclareVerbInputEnvironment{foobar}[3]{%
  \showtokens{#1/#2/#3}}
\begin{foobar}{XXX}{YYY}ZZZ\end{foobar}
```

の場合、本体部では `#1=XXX`、`#2=YYY`、`#3=ZZZ` となる。

引数フィルタを用いると、環境の引数としてオプション引数やその他の特殊な形式の引数（`xparse` パッケージや、`TEX` のマクロ定義を使って実現できる）を扱うことが可能になる。「引数フィルタ」とは、環境に与えられた引数列を読み取って、それを単純な必須引数の列に変換した上で `verbatim` 入力関数の本体に渡す命令（マクロ）のことである。例えば、

```
\newcommand{\foobarargs}[2][ABC]{%
  \DeclareVIEArguments{{#1}{#2}}}%
\DeclareVerbInputEnvironment{foobar}[3][\foobarargs]{%
  \showtokens{#1/#2/#3}}
\begin{foobar}{YYY}ZZZ\end{foobar}
```

というコードの場合、まず環境の引数列 `{YYY}` が引数フィルタに渡される、つまり `\foobarargs{YYY}` により

```
\foobarargs{YYY}
```

が行われる。すると、デフォルト引数の機構のため、

```
\DeclareVIEArguments{{ABC}{YYY}}
```

が実行される。`\DeclareVIEArguments` の引数の後に環境の中身を加えた

```
{ABC}{YYY}{ZZZ}
```

が本体部に渡されるという結果になる。

使用例 先の `timelinelist` の例で、項目頭の記号を「`\textendash`」を既定値とするオプション引数とする場合は以下のようにする。

```
\newcommand*{\timelinelistargs}[1][\textendash]{%
```

```

\DeclareVIEArguments{{#1}}
\DeclareVerbInputEnvironment{timelinelist}[2][\timelinelistargs]{\directlua{
  timelinelist(\luastring{#1},\VIEString{#2}) }}

```

## 6 その他の機能

- `\ParseAsLaTeX{<文字列>}` : 文字列を(現在のカテゴリコード設定の下で)トークン化して (La)TeX のコードとして実行する。

この機能を使うと、「環境を命令の形式で定義する」ことができ、かつ環境内に `verbatim` なものを含めることも可能になる。

使用例 次の命令はテキストを指定された色の枠囲いで出力するものであるが、命令に `\verb` を入れることができない。

```

% \specialbox{前景色}{背景色}{テキスト}
\newcommand{\specialbox}[3]{%
  \fcolorbox{#1}{#2}{\Large\textcolor{#1}{#3}}}
%\specialbox{myred}{mypink}{Lua{\TeX} \verb|^~;|} % ダメ

```

次のように `verbatim` 引数環境として定義すればよい。

```

\DeclareVerbInputEnvironment{specialbox}[3]{%
  \fcolorbox{#1}{#2}{\Large\textcolor{#1}{\ParseAsLaTeX{#3}}}}
\begin{specialbox}{myred}{mypink}
Lua{\TeX} \verb|^~;|
\end{specialbox}

```