

# The kanbun package

テらてふヲルガヲためぱつけーぢ  
以羅蝶作漢文訓讀用包

Yuanhao Chen (陳元鎬)

15 JANUARY 2022, V1.0

## Contents

### 1 Introduction

### 2 Basic example with Lua $\text{\LaTeX}$

### 3 Usage

3.1 Package options . . . . .	2
3.2 Basic usage without Lua $\text{\LaTeX}$	4
3.3 Usage with Lua $\text{\LaTeX}$ . . . . .	5

## 1 Introduction

The kanbun package, like other *kanbun-kundoku* (漢文訓讀)  $\text{\LaTeX}$  packages (such as gckanbun), allows users to manually input macros for elements in a *kanbun-kundoku* paragraph.

More importantly, it accepts input with light markup in the “*kanbun* annotation” form when used with Lua $\text{\LaTeX}$ , which allows typesetting *kanbun-kundoku* paragraphs efficiently<sup>1</sup>.

## 2 Basic example with Lua $\text{\LaTeX}$

As seen in the following example, typesetting a *kanbun-kundoku* paragraph with the kanbun package requires only light annotations — it automatically transforms the annotated plain text into  $\text{\LaTeX}$  macros through Lua, rather than having users type in macros themselves.

<sup>1</sup>The idea comes from 漢文 HTML by UntPhesoca, which is a JavaScript and css implementation.

```

\documentclass{ltjarticle}
\usepackage[kumi=aki, tateaki=1]{kanbun}
\begin{document}
\Kanbun
月落チ烏啼キテ霜満ツ[レ]天ニ,
江楓漁火対ス[二]愁眠ニ[一]。
姑(こ)蘇(そ)城外ノ寒山寺,
夜半ノ鐘声到ル[二]客船ニ[一]。
\EndKanbun
\printkanbun
\end{document}

```

夜 姑<sup>こ</sup> 江 月  
 半<sup>ル</sup> 蘇<sup>そ</sup> 楓 落<sup>チ</sup>  
 鐘 城 漁 烏  
 聲 外<sup>ノ</sup> 火 啼<sup>キテ</sup>  
 到<sup>ル</sup> 寒 對<sup>ス</sup> 霜  
 客 山 愁 満<sup>ッ</sup>  
 船<sup>ニ</sup> 寺 眠<sup>ニ</sup> 天<sup>ニ</sup>

Note that if you want to use this functionality, you have to run this document with Lua<sup>®</sup>TeX.

## 3 Usage

### 3.1 Package options

Load the package with

```

\usepackage[<kanbun options>]{kanbun}

```

This package provides a variety of customisable features in *kanbun-kundoku*.

`scale=<ratio>`

Sets the ratio of the size of *kanji* to that of ruby texts. Default: 2.

`fontcmd`= $\langle font\ command \rangle$

Sets the font command to use for *kanji*. If `luatexja-fontspec` is loaded, it is set default to `\addfontfeatures{RawFeature={+trad}}` to obtain traditional *kanji* if applicable.

`rubyfontcmd`= $\langle font\ command \rangle$

Sets the font command to use for ruby texts. If `luatexja-fontspec` is loaded, it is set default to `\addfontfeatures{RawFeature={+ruby}}` to obtain ruby glyphs when applicable.

`unit`= $\langle length \rangle$

Sets the base size (size of *kanji*). Default: `\kanbun_zw`, which is initialised as 1em.

`yokoaki`= $\langle ratio \rangle$

Sets the horizontal space between *kanji* with respect to the size of ruby texts. Default: 2.

`tateaki`= $\langle ratio \rangle$

Sets the vertical space between *kanji* with respect to the size of ruby texts. Default: 2.

`okuriintrusion`= $\langle ratio \rangle$

Sets the intrusion of *okurigana* (how much *okurigana* should be vertically tucked into the space of *kanji*) with respect to the size of ruby texts, if that does not cause an overlap with *furigana*. Default: 1.

`kumi`=aki, beta

Sets whether to use *aki-gumi* (typeset with uniform inter-character spacing) or *beta-gumi* (typeset with no inter-character space between adjacent character frames). Or simply call `aki` or `beta` without `kumi`=. Default: `aki`.

After initialising the options, you can still change the option values through `expl3` syntax, with a prefix `\kanbun_` to option names. For example, to switch to *beta-gumi*, you could use

```
\ExplSyntaxOn
\tl_set:Nn \kanbun_kumi { beta }
\ExplSyntaxOff
```

### 3.2 Basic usage without Lua<sup>A</sup>T<sub>E</sub>X (not recommended)

When not using the advanced *kanbun*-annotation functionality, it is possible to typeset *kanbun* with any engine with CJK support, such as using X<sub>Y</sub><sup>A</sup>T<sub>E</sub>X with the xeCJK package, or using up<sup>A</sup>T<sub>E</sub>X with utarticle or other appropriate class.

```
\kanjiunit{ \furiokuri{⟨right furigana⟩}{⟨right okurigana⟩} }
{⟨left (top) punctuation (e.g. ‘’’)⟩}
{⟨kanji⟩}
{⟨other punctuation⟩}
{⟨kaeriten⟩}
{ \furiokuri{⟨left furigana⟩}{⟨left okurigana⟩} }
```

`\kanbunfont`

Sets the font size of *kanji*. Use when the unit option is set different to the document’s default font size.

Use `\multifuriokuri` instead of `\furiokuri` if you are putting *furigana* to multiple kanji.

```
\multifuriokuri[⟨length by which ruby is raised⟩]{⟨furigana⟩}{⟨okurigana⟩}
```

For example, the code

```
% example text from https://phesoca.com/kanbun-html/
\kanbunfont
\kanjiunit{}{}{子}{}{}{}
\kanjiunit{\furiokuri{}{ク}}{}{曰}{,}{}{}
\kanjiunit{\furiokuri{}{ソ}}{}{盍}{}{三}{\furiokuri{}{ル}}
\kanjiunit{}{}{各}{ゝ}{}{}
\kanjiunit{\furiokuri{}{ハ}}{}{言}{}{二}{}
\kanjiunit{\furiokuri{}{ノ}}{}{爾}{}{}{}
\kanjiunit{\furiokuri{}{ヲ}}{}{志}{。}{一}{}
\par
```

outputs

言<sup>ハ</sup> 子  
<sup>ニ</sup>爾<sup>ノ</sup> 曰<sup>ク</sup>  
<sup>ヲ</sup>志<sup>ル</sup> 盍<sup>ソ</sup>  
<sup>一</sup>。 各<sup>ニ</sup>  
<sup>ゝ</sup>

with `tateaki` set to 1.

### 3.3 Usage with Lua<sup>A</sup>T<sub>E</sub>X

*Kanbun* annotation uses the following brackets to mark different elements in *kanbun-kundoku* (as described in [漢文 HTML](#) by UntPhesoca).

- ( ): *furigana* (振り仮名)
- { }: *okurigana* (送り仮名) (these brackets can be omitted)
- ‹ ›: *furigana* (振り仮名) of *saidoku-moji* (再読文字)
- « »: *okurigana* (送り仮名) of *saidoku-moji* (再読文字)
- [ ]: *kaeriten* (返り点)
- ‘ ’: multiple *kanji*, potentially with *tateten* inserted, as a ruby base; group ruby (グループルビ)
- no annotation: *kanji* (漢字) and punctuation

*Tateten* (豎点) can be input with either — (U+2015), — (U+2014) or | (U+3190).

`\Kanbun`  
`\EndKanbun`

Write the annotated *kanbun* between the commands `\Kanbun` and `\EndKanbun`, and it will be processed and saved, ready to be used later.

`\printkanbun`

Where you would like to use the most recently saved *kanbun-kundoku* paragraph, use `\printkanbun`.

`\printkanbuncode`

If you wish to make modifications on the result or to use the result with a non-Lua<sup>A</sup>T<sub>E</sub>X engine, it is possible to obtain the macros using `\printkanbuncode` (prints in the terminal), and continue to work from there.

You can always save `\printkanbun` to a macro and start a new annotated *kanbun* block, as in the following example.

```
% example text from https://phesoca.com/kanbun-html/
\documentclass{ltjtarticle}
\usepackage[kumi=beta]{kanbun}

\Kanbun
此レ乃チ信(しん)之‘所一[三]以’(ゆゑん)為ル[二]陛下ノ禽(とりこ)ト[一]也。
\EndKanbun
\let\信\printkanbun

\Kanbun
孤之有ルハ[二]孔明[一], 猶ホ<ごと>«キ»[二]魚之有ルガ[一レ]水也。
\EndKanbun
\let\孔明\printkanbun

\begin{document}
\孔明\par\bfseries\信
\end{document}
```

孤<sup>ルハ</sup>之<sup>二</sup>有<sup>一</sup>孔明<sup>一</sup>, 猶<sup>ホ</sup>ホ<sup>二</sup>魚<sup>二</sup>之<sup>一</sup>有<sup>ルガ</sup>水<sup>レ</sup>也<sup>。</sup>  
 此<sup>レ</sup>乃<sup>チ</sup>信<sup>しん</sup>之<sup>三</sup>所<sup>一</sup>以<sup>ゆゑん</sup>為<sup>ルキ</sup>陛<sup>二</sup>下<sup>一</sup>禽<sup>ノ</sup>ト<sup>とりこ</sup>也<sup>。</sup>