The \texttt{pxrubrica} package

Takayuki YATO (aka “ZR”)

v1.3d  [2021/03/06]

1 Package Loading

There is no options available.

\usepackage{pxrubrica}

2 Basic Usage

2.1 Very Basic

- Mono ruby (\texttt{m} option): one ruby block per one kanji.
  
  \begin{itemize}
  \item E.g. \texttt{\ruby[m]{鷹}{たか}} \rightarrow 鷹 \texttt{\ruby[m]{鷹}{うぐいす}} \rightarrow 鷹
  \end{itemize}

- Group ruby (\texttt{g} option): one ruby block per kanji sequence.
  
  \begin{itemize}
  \item E.g. \texttt{\ruby[g]{雲雀}{ひばり}} \rightarrow 雲雀 \texttt{\ruby[g]{不和}{ほととぎす}} \rightarrow 不和
  \end{itemize}

- Jukugo ruby (\texttt{j} option): one ruby block per kanji, but read as whole.
  
  \begin{itemize}
  \item E.g. \texttt{\ruby[j]{孔雀}{くにじゃく}} \rightarrow 孔雀 \texttt{\ruby[j]{七面鳥}{しずめんちょう}} \rightarrow 七面鳥
  \end{itemize}

The current implementation uses the following strategy: if none of the ruby string per a kanji causes protrusion, then the whole is typeset as mono-ruby; otherwise, the whole is typeset as group-ruby, ignoring the ruby grouping.

- A \(\mid\) symbol in a ruby string denotes the boundary of reading for each kanji (i.e. 孔 = く and 雀 = じゃく). It is not needed in group ruby.

- Comparison of typeset results:
  
  \begin{itemize}
  \item Mono \(\texttt{\ruby[m]{小鳩}{こ|ばと}}\) \rightarrow 小鳩 鶴鶴 雷鳥 燕
  \item Group \(\texttt{\ruby[g]{小鳩}{こ|ばと}}\) \rightarrow 小鳩 鶴鶴 雷鳥 燕
  \item Jukugo \(\texttt{\ruby[j]{小鳩}{こ|ばと}}\) \rightarrow 小鳩 鶴鶴 雷鳥 燕
  \end{itemize}

Usually When a jukugo has per-character reading then jukugo ruby (\texttt{j}) is preferred, otherwise (\texttt{g}) is. If you particularly intend to show the per-character reading for a jukugo, you might use the \texttt{m} option. Note that all of \texttt{m}, \texttt{g} and \texttt{j} lead to the same result for ruby to a single kanji.

- You can give a default value of option using the \texttt{\rubysetup} command; \texttt{\rubysetup{g}\ruby{鳥}{しゃも}} is equivalent to \texttt{\ruby[g]{鳥}{しゃも}}. The “default of default” is \texttt{cjPeF}.

1
2.2 Intrusion/Protrusion

- Control of ruby intrusion:

<table>
<thead>
<tr>
<th>Intrusion Type</th>
<th>Ruby Command</th>
<th>Kana</th>
<th>Kanji</th>
</tr>
</thead>
<tbody>
<tr>
<td>No intrusion</td>
<td>\ruby[</td>
<td>-&gt;]</td>
<td>この 鶴 の この 鶴 の この 鶴 の</td>
</tr>
<tr>
<td>Small intrusion</td>
<td>\ruby[(-)]</td>
<td>この 鶴 の この 鶴 の この 鶴 の</td>
<td></td>
</tr>
<tr>
<td>Big intrusion</td>
<td>\ruby[&lt;-&gt;]</td>
<td>この 鶴 の この 鶴 の この 鶴 の</td>
<td></td>
</tr>
</tbody>
</table>

- If you require “ruby output may intrude to kana but not to kanji,” then you might want to do “この\ruby[<-|]\{鶴\}{かさぎ}等” to get “この鶴等.”

- To specify a basic mode (m/g/j) and intrusion at a time, you can use option strings such as |g| and |m>. In fact, the symbol - works as placeholder for basic mode and means the use of default value.

- Control of ruby protrusion: the string || supresses the protrusion.

```
雀の… ← \ruby[|->]{雀}{すずめ} vs. 雀の… ← \ruby[|->]{雀}{すずめ}
インコの
```

2.3 More Commands

- \aruby: attaches ruby to an alphabet (non-CJK) string.
  E.g. \aruby{Get out}{ゲラウッ}! → Get out!

- \rubyfontsetup: specifies the font used for ruby output. For example, when you want to attach mincho-font ruby to gothic-font kanji string you can do as follows:

```
\rubyfontsetup{\mcfamily}この\{\gtfamily \ruby[j]{明朝体}{みんなちょうたい}} → この 明朝体
```