Japanese (with furigana) in Org-mode

1 Proposed header

```
#+macro: furigana      @@html:<ruby> $1 ...
#  ... (same line)     <rp>(</rp><rt>$2</rt><rp>)</rp> ...
#  ... (same line)     </ruby>@@@@latex:\ruby{$1}{$2}@@
#+LATEX_HEADER: \usepackage[CJK, overlap]{ruby}
#+LATEX: \begin{CJK*}[dnp]{JIS}{min}
```

2 Proposed footer

```
#+LATEX: \clearpage\end{CJK*}
```

3 How to use it

Insert the proposed header and footer into your Org file, this will enable you to write Japanese text in your Org documents. Text will be exported correctly in HTML and PDF.

You can write Japanese text (ex.: 東京) and use the proposed macro `furigana` to add furigana signs on top of a kanji. Examples:

- `{{{furigana(水,mi zu)}}}` ⇒ 水(mi zu)
- `{{{furigana(亀,かめ)}}}` ⇒ 亀(かめ)
- `{{{furigana(犬,いぬ)}}}` ⇒ 犬(いぬ)
- `{{{furigana(猫,ねこ)}}}` ⇒ 猫(ねこ)

4 Problems and limitations

4.1 Japanese text is suppressed in PDF title

I don't know any solution or workaround for this yet; sorry :(.

4.2 PDF output fails if there is Japanese text in Org headlines

This is due to a limitation of the PDF bookmarks, but there is a couple of suggested workarounds.

4.2.1 workaround 1 (boring)

Generate output to a LaTeX file instead than PDF, then replace `\section{<japtext>}` into `\section[simpletext]{japtext}` in the .tex file, where `simpletext` is some plain ASCII text (actually I think that all ISO-8859-1 characters are supported as well). After the corrections, generate a PDF from the .tex file using `pdftex`.

4.2.2 workaround 2

Generate output to a LaTeX file instead than PDF, then in the .tex file replace

```
\usepackage{hyperref}
```

into

```
\usepackage[bookmarks=false]{hyperref}
```

After the correction, generate a PDF from the .tex file using `pdftex`.

4.2.3 workaround 3 disable hyperref

This can be done by removing `hyperref` package from the Org option `org-latex-default-packages-alist`. This will disable all hyperref features in the generated PDFs, so it's probably better **not** to remove it permanently from your Emacs init file.

4.2.4 a better workaround could be...

Subclass the Latex exporter so that it automatically behaves the same as workaround 1.