

Kinx TT - Kinx Tiny Typesetting

Abstract

This system is a small typesetting system written in Kinx. Most of people knows the L^AT_EX¹ is being used for that purpose, but the T_EX² system is very huge. This system provides only a limited functionality, but supports some of T_EX algorithms in this small system. This would be your best partner as long as used for your personal use.

1 Overview

1.1 Goals

The goal of the Kinx Tiny Typesetting is as follows.

- Keeping it small.
- Pretty beautiful.
- Direct output to PDF.

Kinx TT has supported some kind of T_EX algorithms, so the final output would be pretty beautiful. You can check it yourself as this document was generated by this system. On the other hand, there are some known bad points below as a trade off.

- Needs a performance improvement.
- Provides only a limited functionality.

1.2 Features Overview

This system supports following algorithms.

1. Main Features
 - (a) Hyphenation & Knuth-Plass Line Breaking Algorithm.
 - (b) Widows & Orphans penalty control.
 - (c) Listing items with a bullet or numbering.
 - (d) Math formula & equation like T_EX
 - (e) Writing a program source code.
 - (f) Footnotes³.
 - (g) Images, and it can be also put with floating inside paragraph.
2. For Japanese
 - (a) Japanese hyphenation rules.
 - (b) Japanese Ruby. how to read ほんご にほんご 日本語, or 日本語 to be separated for each character.

¹ Leslie Lamport, <https://en.wikipedia.org/wiki/LaTeX>

² Donald E. Knuth, <https://en.wikipedia.org/wiki/TeX>

³ This is a footnote example.

2 Features Details

2.1 Hyphenation & Line Breaking

The figure on the right⁴ is an example of hyphenation and justification. Hyphenation will be done before applying line-breaking algorithm. The algorithm is based on `Hyper.js`, and it relies on the hyphenation algorithm by Franklin M. Liang commonly known from `LATEX`.

And also, this system is supporting **Knuth-Plass Line Breaking Algorithm** for line-break.

This is well known algorithm because the `TEX` uses it. These algorithms are known as the best way so far for a typesetting system.

This Kinx TT has supported some kind of `TEX` algorithms, so the final output would be very beautiful. You can check it on your eyes yourself as this document was generated by this system. On the other hand, there are some bad points below as a trade off.

Fig 2.1 Hyphenation and justification

2.2 Widows & Orphans

Widows and orphans control is not perfect so far. But normally it is available for a section and a paragraph. Unfortunately it is not available for listing items, images, and so on. Please check it and you can use `\columnbreak` command anywhere you need.

2.3 List items

You can use an `itemize` and an `enumerate` listing items as follows.

Here is an example of 'itemize'.

- Item level 1-1
- Item level 1-2
 - Item level 2-1
 - Item level 2-2
 - Item level 2-3
 - Item level 3-1
 - Item level 3-2
 - * Item level 4-1
 - * Item level 4-2

Here is an example of 'enumerate'.

1. Item level 1-1
2. Item level 1-2
 - (a) Item level 2-1
 - (b) Item level 2-2
 - (c) Item level 2-3
 - i. Item level 3-1
 - ii. Item level 3-2
 - A. Item level 4-1
 - B. Item level 4-2

⁴ This is an example of not only hyphenation but also a floating image.

2.4 Math formula and equation

Here is an example of Math formula and equation. You will think as if it is L^AT_EX output. Yes, you are quite right because this is came from T_EX output, but it is never using a huge T_EX system. Instead, it is using an output of K^AT_EX, which is included in this system.

$$E = mc^2$$
$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}}$$

$$\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi}$$

Fig 2.2 Math example

If you want to write it in a paragraph line, use \$...\$ style directly in paragraph. For example, $E = mc^2$ is written as \$E = mc^2\$, and another example is $\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi}$ written as \$\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}\$.

If you want to use \$ in paragraph, use \$ with \ like \\$.

2.5 Program source code

Here is an example of a source code.

```
function fib(n) {
    if (n < 3) return n;
    return fib(n-2) + fib(n-1);
}
System.println("fib(34) = ", fib(34));
```