Kinx TT - Kinx Tiny Typesetting

Kray-G, Mr.Diamond Global Blue Publisher

September 25, 2020

Abstract

This system is a small typesetting system written in Kinx. Most of people knows the LaTeX[1] is being used for that purpose, but the TeX[2] system is very huge. This system provides only a limited functionality, but supports some of TeX algorithms in this small system. This would be your best partner as long as used for your personal use. By the way, note that this document describes just a feature but less technical information. Please wait for the technical document, and I will prepare that as soon as possible.

Contents

[1] Leslie Lamport, https://en.wikipedia.org/wiki/LaTeX
[2] Donald E. Knuth, https://en.wikipedia.org/wiki/TeX

List of Figures

List of Tables

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 TEX 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.
- Prvides only a limited functionality.

1.2 Features Overview

This system supports following features.

1. Hyphenation & Knuth-Plass Line Breaking Algorithm, and also for some Japanese rules.
2. Widows & Orphans penalty control.
3. Listing items with a bullet or numbering.
4. Math formula & equation like TEX.
5. Tables. The header of table is automatically repeated each page.
6. Images. It can be also put with floating inside paragraph.
7. Colors. CMYK, RGB, and a lot of color names are supported.
8. Chart. All the charts of Chart.js are supported.
9. Writing a program source code.
10. Cross-reference like “**Reference to the section 2.5 Table** in the page 3.”
11. Autimatic URL hyperlink like https://en.wikipedia.org/wiki/TeX, or link text like TEX.
12. Footnotes[3].
13. Ligature. Only 5 words of fi, fl, ff, ffi, ffl are supported.
14. Japanese Ruby. 日本語(にほんご), or 日本語(how to read) to be separated for each character.

[3] This is a footnote example.

2 Features Details

2.1 Hyphenation & Line Breaking

The figure on the right[4] 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.

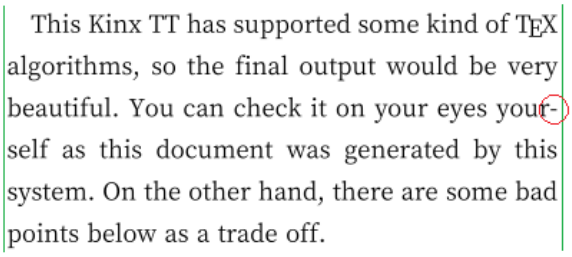

Fig 2.1 Hyphenation and justification

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.

Kinx TT is also supporting some Japanese hyphenation rules. For example, 「。」, which is like a period in English, never appears at the head of line.

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 also the example of 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 LaTeX output. Yes, you are quite right because this is came from TeX output, but it is never using a huge TeX system. Instead, it is using an output of KaTeX, which is included in this system.

$$E = mc^2 \tag{1}$$

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

$$\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi} \tag{3}$$

If you want to write a math formula inline, use `$...$` style directly in paragraph. For example, $E = mc^2$ is written as `$E = mc^2$`, and another example is $\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi}$ same as (3). As KaTeX does not support `\label` command, the label is being provided by Kinx TT. Therefore note that you need to control it by yourself a little.

If you want to use just "$" in paragraph, use $ with \ as \$.

And also you can use `\displaystyle` keyword. The Math formula can be directly put inside paragraph like $\displaystyle\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi}$, by `$\displaystyle\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$`.

2.5 Table

Here is an example of table. The ruled line can be selected from no line, a single line, or a double line. And you can set an alignment each column. Besides, the column width is also automatically calculated according to inside text size.

By the way, you can write also a Math formula inside a table cell.

Table 2.1 Table Example Crossing The Page

Left	Center	Right
A1	Aligned to the center[6].	C1
A2	$\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi}$	Aligned to the right.
A3	Aligned to the center.	On the right.
A4	Aligned to the center.	On the right.

[6] This is a footnote in table.

Left	Center	Right
A5	Aligned to the center.	On the right.
A6	Aligned to the center.	On the right.
A7	Aligned to the center.	On the right.

The table header can be automatically repeated each page.

Tables can be put on a line by using multiple columns. Here is an example of 2 tables in a line. Note that the width will be becoming narrow in this case, and also you should control to break a page yourself if crossing pages.

#	Outline
1	Something
2	Something

#	Overview
1	Something
2	Something
3	$\int_{-\infty}^{\infty} f(x)dx = \sqrt{\pi}$ is an example of Math formula in the table.

2.6 Image

2.6.1 Floating Image

Images can be put around a text, or put on a line as a standalone.

Fig 2.2 Barcode Icon

The first example is for a floating image put around a text. The image on the left, which is a barcode icon example, is wrapped inside text.

And also, the text around a floating image can be separated by multiple paragraphs like this. The line-break algorithm is correctly performed for a different width per line. By this mechanism, the text will wrap around a floating image and you can do it with only writing a paragraph as usual. You will be able to see this line is put under the image automatically.

By the way, the scale for a floating image means the parcentage for the actual image size. Note that it is different from a scale for a non-floating image.

You can also put a reference to an image like Figure 2.1 Hyphenation and justification.

2.6.2 Standalone Image

This is an example of an image scaled at 75%, and the image is put with a caption. This scale for a non-floating image means a parcentage for the area within the margins.

Fig 2.3 Figure Example - Kinx Logo Image

This is an example of an image scaled at 40%.

This example is the example of writing \ `image` command directly into a paragraph. Note that you can not float an image in this way, and you have to use `\floatimage` command instead.

If you want to multiple images in a line, please see below.

You will see 3 images in a line.

2.7 Colors

Changing a text color is supported by the command of \ `color`. You can specify the color by the name like red, or CMYK/RGB values directly. For example, please see below.

1. **Red**. This line should be colored by the name of `[red]`.
2. **Green**. This line should be colored by the name of `[green]`.
3. **Blue**. This line should be colored by the name of `[blue]`.
4. **Cyan**. This line should be colored by the name of `[cyan1]`.
5. **Magenta**. This line should be colored by the name of `[magenta1]`.
6. **Yellow**. This line should be colored by the name of `[yellow]`.
7. **RGB**. This line should be colored by RGB value of `[R=0,G=64,B=255]`.
8. **CMYK**. This line should be colored by CMYK value of `[C=0.5,M=0.8,Y:0.2,K=0.0]`.

2.8 Chart

Chart is now available in the document. You can use any chart of Chart.js. Please look at the web site of https://www.chartjs.org/ if you want to know details.

This is an example of chart as a floaing image. The chart can be used as a floating image or a standaone image. This example is a line chart, but so many charts are supported by Chart.js. For example, it is a bar chart, a stack chart, a area chart, a scatter chart, a pie chart, a doughnut chart, a polar area chart, a radar chart, and so on. Besides, you can also use a combination of bar and line chart if you want. See the Chart.js Web Site for details.

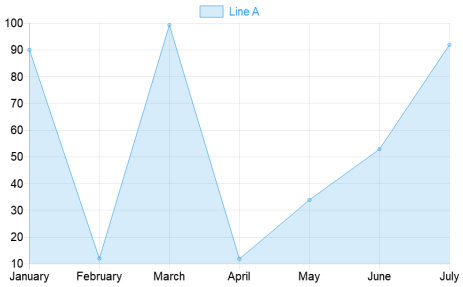

Fig 2.4 Line Chart Example

The next example is a radar chart as a standalone image.

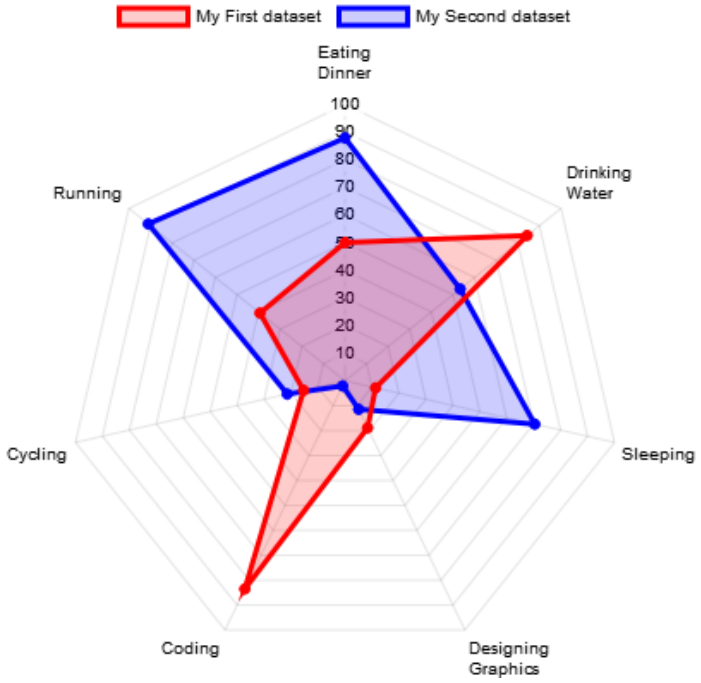

Fig 2.5 Radar Chart Example

Also about chart options, please see the Chart.js Web Site for details.

2.9 Program Code

Here is an example of a source code. This code is a simple Kinx code of fibonacci function. The code is vary basic and it is often used for a benchmark.

```
 1  function fib(n) {
 2      if (n < 3) return n;
 3      return fib(n-2) + fib(n-1);
 4  }
 5
 6  native nfib(n) {
 7      if (n < 3) return n;
 8      return nfib(n-2) + nfib(n-1);
 9  }
10
11  var tmr = new SystemTimer();
12  var r = fib(34);
13  var e = tmr.elapsed();
14  System.println("fib(34) = %8d, Normal Function: elapsed: %f" % r % e);
15
16  tmr.restart();
17  r = nfib(34);
18  e = tmr.elapsed();
19  System.println("fib(34) = %8d, Native Function: elapsed: %f" % r % e);
20
21  # fib(34) =  9227465, Normal Function: elapsed: 0.590539
22  # fib(34) =  9227465, Native Function: elapsed: 0.052000
```

This example is also an example of `native` Keyword in Kinx. The `native` Keyword means the function is directly assembled to x64 machine code, so the performance is very fast. In the above example, the native function is 10 times faster than the normal function.

The code can be also shown with a line number and a box around the code. There are some box line types you can select as follows.

Table 2.2 The Box Line Type

Type	**Meaning**
`Typesetting.BOX_NORMAL`	Normal line, by default.
`Typesetting.BOX_THIN`	Thin line.
`Typesetting.BOX_THICK`	Thick line.
`Typesetting.BOX_SHADOW`	The shadow box. This is used in the example above.

2.10 Cross Reference

Cross references are supported with \ label. The reference will be applied by \ ref, \ pageref, \ textref, and \ nameref. \ ref means just a number. \ pageref means a page number. \ textref means a text only. \ nameref means a number and a text.

For example, write "\nameref{Sec:Table} in the page \pageref{Sec:Table}", and "2.5 Table in the page 3" will appear.

2.11 Links

When writing just a URL like https://github.com/Kray-G/kinx, it will be there with a hyperlink automatically. If you want to write a link text instead of writing a hyperlink URL directly, you can use \ url command for that. For example, \url[https://github.com/Kray-G/kinx]{Kray-G/kinx} will be Kray-G/kinx with a hyperlink to the web site of https://github.com/Kray-G/kinx.

2.12 Footnotes

You can use **footnote** for some additional explanation of the sentence. For example, this is a main sentence[7] and footnote will appear at the bottom of the page.

2.13 Ligature

The ligature for only 5 words of fi, fl, ff, ffi, ffl is now supported. For example, "difficult" has a "ffi" and you will see ffi instead of ffi.

2.14 Japanese Ruby

This is an requirement only for Japanese. In Japanese, there is a writing style called "Ruby" which is the style of a small text written on the top of a main sentence to inform how to read a Kanji of a main sentence. For example,「この本はとても面白いです。」, which means that "this book is very interesting." In general, so young children can not read Kanji, and also there is a case that even adult people can not read a difficult Kanji or the Kanji which they do not use usually. So ruby is used like 本 for them.

Kinx TT provides 2 types of ruby. One of those is like「素敵」. In this case, the ruby is for total words. You can see the「て」is just on the center between「素」and「敵」. Another one is like「素敵」. In this case, the ruby is for each Kanji one by one, and it allowed to break the line between「素」and「敵」. By the way,「素敵」means "wonderful" in Japanese.

As another example, the long ruby like「狼」is okay. In an opposite way, the short ruby like「無花果」is also okay. You can also use English as a ruby such as「個人化」, or「personalization」.

Anyway you can free to use a ruby in your document.

[7] This is a footnote for the main sentence.