



Small and Easy, but Beautiful Design.

KiTTy

Beautiful Design for you ...

Kray-G, Mr.Diamond Global Blue Publisher
September 18, 2020

Contents

Chapter 1 Introduction	1
1.1 What is KiTTY	1
1.1.1 Versus L ^A T _E X	1
1.1.2 Versus Word	2
1.1.3 Versus Vivliostyle	2
1.1.4 Conclusion	2
1.2 Supported Features	3
1.2.1 Typesetting Features	3
1.2.2 Typesetting Features in Japanese	3
1.2.3 PDF Features	4
Chapter 2 Getting Started	5
2.1 Installation	5
2.1.1 Linux	5
2.1.2 Windows	6
2.2 Build	6
2.2.1 Linux	6
2.2.2 Windows	6
2.3 hello, world	7
Chapter 3 Features Overview	9
3.1 Typesetting Features	9
3.1.1 Hyphenation, Justification, and Line-Breaking Algorithm	9
3.1.2 Widows and Orphans	9
3.1.3 Multiple columns	10
3.1.4 Itemization	11
3.1.5 Math formula and equations	12
3.1.6 Image	13
3.1.7 Chart	15
3.1.8 Table	17
3.1.9 Font	19
3.1.10 Color	21
3.1.11 Ligature and Special Character	21
3.1.12 Programming Code Block	22
3.1.13 Title, Cover Page, and Contents	23
3.1.14 Heading	24
3.1.15 Cross-Reference	25

3.1.16 Quotation	25
3.1.17 Footnote	26
3.2 Typesetting Features in Japanese	26
3.2.1 Japanese Hyphenation	26
3.2.2 Japanese Furigana	26
3.3 PDF Features	28
3.3.1 External Link by URL	28
3.3.2 Link by Cross-Reference	28
3.3.3 Bookmark	28
Chapter 4 Command Details	29
4.1 Markdown Commands	29
4.1.1 Paragraph Commands	29
4.1.2 Inline Commands	30
4.1.3 HTML Commands	30
4.2 KiTTy Commands	31
4.2.1 Paragraph Scope Commands	31
4.2.2 Standalone Commands	31
Chapter 5 How To Extend Features	33
5.1 Adding Style	33
5.1.1 Document Style	33
5.1.2 Title Style	34
5.1.3 Chapter Style	34
5.2 Hyphenation Rules	35
5.2.1 Add The Rule	35
5.3 Font	37
5.3.1 Add New Font	37
5.3.2 Add System Font	37
5.4 Commands	37
5.4.1 Define Standalone Commands	38
5.4.2 Define Paragraph Scope Commands	39
Appendix A Pre-defined Designs	41
A.1 Title Design	41
A.1.1 StandardArticle	41
A.1.2 StandardBook	41
A.2 Chapter Design	42

A.2.1	StandardBook	42
A.2.2	BigChapter1	42
A.2.3	BigChapter2	42
A.2.4	BigChapter3	42
Appendix B About Style Parameters		43
B.1	Style Parameters	43
Appendix C Color Examples		45
C.1	Color Names and RGB/CMYK Values	45

List of Figures

Figure 3.1	Hyphenation and Justification	9
Figure 3.2	Damselfly	14
Figure 3.3	F14 Tomcats	14
Figure 3.4	Radar Chart Example	16
Figure 3.5	Line Chart Example	16

List of Tables

Table 3.1	Image Options	14
Table 3.2	Example of Table	17
Table 3.3	Table Options	18
Table 3.4	How to write Bold, Italic, and BoldItalic	19
Table 3.5	Unit for Font Size	20
Table 3.6	The available values for box	23
Table 3.7	Headings in Markdown	24
Table 3.8	Command List for Cross-Reference	25
Table 4.1	Markdown Paragraph Commands	29
Table 4.2	Markdown Inline Commands	30
Table 4.3	HTML Commands	30
Table 4.4	Paragraph Scope Commands	31
Table 4.5	Standalone Commands	32
Table 5.1	Font File Search Path	37
Table 5.2	KiTty Command Available Object Type	38

Table B.1 Style Parameter List	43
Table C.1 Color Names and RGB/CMYK Values	45

Chapter 1

Introduction

First, this section introduces a KiTTy itself and shows a comparison to alternative softwares, and introduces a value of KiTTy and use case of KiTTy. And also, it shows supported features.

1.1 What is KiTTy

KiTTy means Kinx Tiny Typesetting, which is a simple typesetting system implemented by Kinx. It also provides a translator from Markdown, then you can typeset a Markdown document and can get a beautiful document. This document itself is also the example typeset by this system.

The objective is similar to LATEX, it is an objective to typeset beautifully for a document managed as a text file. To be concretely, it is never going to be alternative, but the objective is being more useful in the use case like your personal situation by followings.

- Keeping a small system.
- Pretty beautiful output.
- Directly output as PDF file.

KiTTy is small but it has a simple feature to typeset it beautifully, and it is a typesetting system to output PDF directly from Markdown document.

1.1.1 Versus LATEX

LATEX is a huge system. It is extendable, it is supported by many people, and it is a typesetting system which can make a beautiful document. KiTTy is also a typesetting system which has a same objective as LATEX, but it is provided as a small system with limited features.

Being huge of LATEX system causes a complexity of an installation. Besides, there are multiple distributions to provide many functionality of TEX and LATEX. That is the reason why people have to think first what distribution should be used. Everybody can use KiTTy soon because KiTTy is included in Kinx as a standard library,

Only one cons point is that limited features¹ are provided. And also the compilation speed is *slow*. It takes about 4 minutes to compile this document. The performance is one of problems, but this system is focusing what is used as a small and a personal use.

¹ About “limited features”, see [Chapter 3 Features Overview](#).

1.1.2 Versus Word

The Word is a famous WYSIWYG² Word Processor, but it is based on a different concept. The word processor can edit a document as it shows, but it usually saves a document as a binary style. Therefore, in general, you need an own software, which is “Word” in this case, to show its content. With KiTTy, same as LATEX, you can see its content directly because it is a text style, and can edit it with your familiar editor.

Saving as a text, it can be processed easily by another software, and also showing diff in a version control like Git is very easy. This is quite important and a mandatory requirement, especially for the management of source code. Besides, about the structure of a document, it is difficult from its looking to distinguish if it is structured or not. For example, you can not find its chapter number is correctly assigned, and you can not find if it would be corrected when you change a layout or a location of sentences. In particular, there is a case sometimes you have met unlucky cases as someone has not set it up correctly.

The structure is clearly written in text style in KiTTy, it can correctly realize to make a cross-reference, numbering a chapter number, a figure number, a table number, and a page number. Instead, it is a cons point not to show the document as it shows, like WYSIWYG. You have to compile first to show the actual document to be typeset.

1.1.3 Versus Vivliostyle

CSS Typesetting is based on the Web Technology, and CSS Typesetting now has a most great future. Therefore CSS Typesetting is recommended if you want to do typeset in earnest. Especially Vivliostyle³ is a very hopeful project.

As for KiTTy, it is important to be a small system and easy to use. Although time will solve, CSS Typesetting is still making it standard in progress. Therefore some of the specification has not been clarified yet, and it is possible to assume it will be changed in future. As this is a big project and making it standardized, it will be a system to use in a real commercial use, but the system itself will be getting huge.

Said again, a main point of view in KiTTy is a small and easy to use.

1.1.4 Conclusion

If you want to use a typesetting system right now in a real earnest, you should use LATEX. If you want to study a typesetting system in the future and use it, you had better use a CSS Typesetting like Vivliostyle. Moreover, in summary, KiTTy will be useful in following situations.

- Wants to use a small system, instead of the huge system like T_EX.
- Wants to manage a diff by a version control system like Git.
- Wants to control the structure of a document and make a cross-reference correctly.

² This means What You See Is What You Get.

³ <https://vivliostyle.org/>

This system is provided to the person⁴ who think the TEX system is very huge as a system for creating a small document, and who wants to manage the document as a text file. In particular, if you want to manage a document and a diff by the system like Git, it is quite difficult to manage it by Word which is realized by the style of WYSIWYG. It is a main target of use cases not to manage by a word processor and not to use a huge system.

1.2 Supported Features

1.2.1 Typesetting Features

KiTty supports following features. See “[Chapter 3 Features Overview](#)” for a concrete content of features. By the way, kerning is not supported so far.

- Hyphenation, justification, and line-breaking algorithm
- Widows and Orphans
- Multiple columns
- Itemization
- Math formula and equations
- Image
- Chart
- Table
- Font
- Color
- Ligature
- Programming Code Block
- Title, Cover Page, and Contents
- Chapter and Section
- Cross-Reference
- Quotation
- Footnote

1.2.2 Typesetting Features in Japanese

KiTty also supports following features for the own requirement in Japanese. If you want to add any other languages, I can not lead you because I am sorry I do not have any knowledge for other languages. As it has been difficult to care about the extension point for other languages, it might need to add a lot of features or many points of fixes. But it will be possible to extend features because this is an open source.

- Japanese hyphenation
- Japanese Ruby like Difficult Kanji^{how to read}

⁴ This means the person like me. In fact, the first objective was to create a user manual of my own project.

1.2.3 PDF Features

Although this is not available when it is printed on a paper, the following features are supported.

- External Link by URL
- Link by Cross-Reference
- Bookmark

Chapter 2

Getting Started

This chapter introduces about preparing to start using KiTTy in practice and how to use through simple examples. By the way, there are differences about how to prepare between Windows and Linux, but you can get a same result as long as you use prepared fonts in KiTTy. Therefore you can use both without problems.

2.1 Installation

There are 2 steps to install it.

1. Install Kinx
2. Install KiTTy package

2.1.1 Linux

Download the following modules first on Linux. The version v0.15.2 is the version which includes KiTTy library officially. When the latest version is different, rewrite the version number and do it. Make the work directory and change the directory.

```
$ mkdir temp  
$ cd temp
```

Download Kinx module and extract it first, and then download KiTTy package and extract it.

```
$ curl -L \  
      https://github.com/Kray-G/kinx/releases/download/v0.15.2/package_linux-amd64.tar.gz \  
      --output package_linux-amd64.tar.gz  
$ tar -xvf package_linux-amd64.tar.gz  
$ curl -L \  
      https://github.com/Kray-G/kinx/releases/download/v0.15.2/package_kitty.zip \  
      --output package_kitty.zip  
$ unzip package_kitty.zip
```

You will find the folder of the name of the version number, and move into it and execute an `install.sh` command.

```
$ cd v0.15.2  
$ sudo ./install.sh
```

That is all for installation. Let's confirm the location to be installed.

```
$ which kinx  
/usr/bin/kinx
```

2.1.2 Windows

Download 2 latest modules from [Release Page](#) when it is Windows.

- package_win64.zip
- package_kitty.zip

Extract each, and then copy files in `package_kitty.zip` to `lib` folder¹.

The fonts and `phantomjs` folder should be right under `lib` folder. Please note that the files in an archive file might not be that structure.

2.2 Build

You do not have to build it usually, you can use a prebuilt modules following the procedure of “[2.1 Installation](#)”. If you dare to build it, you can do it by the procedure as follows.

2.2.1 Linux

Clone it from Github, and do `make`.

```
$ git clone https://github.com/Kray-G/kinx.git  
$ cd kinx  
$ make
```

Install it.

```
$ git clone https://github.com/Kray-G/kinx.git  
$ cd kinx  
$ sudo make install  
$ sudo make kitty-install
```

2.2.2 Windows

Clone it from Github, and do `make`.

```
$ git clone https://github.com/Kray-G/kinx.git  
$ cd kinx  
$ make.cmd
```

Now no installer has been provided, but you can use this system in the build environment. Use the system in the build environment.

¹ The installer for Windows is now not provided yet, but it will be prepared in the future.

2.3 hello, world

Create the following text and save it as the name of `helloworld.md`.

```
% Hello Kinx Tiny Typesetting  
% Your name  
% October 7, 2020  
  
<param style="ArticleA4"/>  
  
# Greeting  
hello, world
```

Execute `kxkitty` command as below, and `helloworld.pdf` will be generated.

```
$ kxkitty helloworld.md
```

By the way, now you need write the document as above, but I will plan that it can be a little more simple document.

Chapter 3

Features Overview

This chapter will explain the feature's overview. As this document itself has been generated by this system, you can realize all what this document realize. Let's try various things first.

3.1 Typesetting Features

3.1.1 Hyphenation, Justification, and Line-Breaking Algorithm

This system supports a Hyphenation based on Franklin M. Liang algorithm. And also this supports a justification with that hyphenation algorithm.

Line-breaking is based on Knuth-Plass Line Breaking algorithm. This algorithm is what is controlled by Box, Glue, and Penalty, and it is same as the algorithm implemented in \TeX . These hyphenation and line-breaking algorithm are the well-known as the best practice in the typesetting so far. Note that it is not all same as the output by \TeX , because this is implemented by Kinix.

This Kinix 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 3.1 Hyphenation and Justification

3.1.2 Widows and Orphans

In some case, this system provides a part of a penalty control for Widows and Orphans. Note that this is not perfect for all cases. It is available for following cases.

- Deterrent to put a section name to the bottom of page.
 - In this case, a section name is moved to a next page.
- Deterrent to put only a top line in a paragraph to the bottom of page.
 - In this case, all lines in a paragraph is moved to a next page.
 - When only a section name is put to the bottom of page as a result, a section name is also moved to the next page.
- Deterrent to put only a last line in a paragraph to the top of a next page.
 - In this case, last 2 lines are moved to a next page.

Those above mechanisms are automatically done. You do not have to do anything in your document. But those are not perfect for all cases, therefore use `<pagebreak />` command when you feel it is not good layout, and you can insert a page break there.

3.1.3 Multiple columns

You can use multiple columns. Write `<set-column value="N"/>`, and the document is set to N columns. When you get back to only 1 column, use `<set-column value="1"/>`. The line will be back to the top of a next column when the text is reached at the bottom of page.

You can set the column height with `height` parameter. `height` parameter can be the value with unit like `10em`. Writing `<set-column value="2" height="12em" />` will set 2 columns with the height of `12em`.

About footnotes, it is not set each column. footnotes are always set against total width of a page. The following example is a part of *Alice's Adventures in Wonderland*¹ with 2 columns.

CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice "without pictures or conversations?"

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

There was nothing so VERY remark-

able in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-POCKET, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

The height is set to `30em` here. As the system does not automatically set the height, you have to set it yourself one by one if you want to set the height. The line is reached at the bottom of page, the text will be back to the top of a next column automatically. When you do not set the height, the text will get back to the top of a next column automatically when it reached at the bottom of a page.

¹ [Alice's Adventures in Wonderland - Lewis Carroll](#)

3.1.4 Itemization

There are 2 types of itemization with a symbol and a number. The following is an example of itemization with a symbol.

```
1 * Level 1
2   * Level 2
3     * Level 3
4       * Level 4
```

This is formatted as follows.

- Level 1
 - Level 2
 - Level 3
 - * Level 4

The next example is a numbered itemization. The numbered for each item will be adjusted automatically.

```
1 1. Level 1
2   1. Level 2
3     1. Level 3
4       1. Level 4
5         1. Level 4
```

This is formatted as follows.

1. Level 1
 - (a) Level 2
 - i. Level 3
 - A. Level 4
 - B. Level 4

It is possible to use both and mix it in the same list. The next example is a mixed one.

```
1 * Level 1
2   1. Level 2
3     * Level 3
4       1. Level 4
```

This is formatted as follows.

- Level 1
 - (a) Level 2
 - Level 3
 - A. Level 4

3.1.5 Math formula and equations

KiTty includes KATEX, and output Math formula and equations. There are 2 styles of stand-alone style and inline style as an output style.

3.1.5.1 Standalone Style

When it is a standalone style, it is written as a Code Block with `Math` language. This is shown in an independent line.

```
1  ```\math:label=Math1
2  \begin{aligned}
3    \int_{-\infty}^{\infty} f(x) dx &= \sqrt{\pi}
4  \end{aligned}
5````
```

For above, the output will be below. It is not necessary to set the option of `label`, but you can use `1` for the reference to Math equations if you set the `label`.

$$\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi} \quad (1)$$

Note that `label` is supported not by KATEX but by KiTTy. Therefore you have to control `label` when there are 2 or more Math equations.

```
1  ```\math:label=Math2(0.2)/Math3(0.6)
2  \begin{aligned}
3    E &= mc^2 \\
4    m &= \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \\
5  \end{aligned}
6````
```

For above, the label of Math2 and Math3 will be put at 20% and 60% from the top of the location where the Math equations is.

$$E = mc^2 \quad (2)$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (3)$$

By this, writing `\ref{Math2}` creates the reference to the Math `2`, and writing `\ref{Math3}` creates the reference to the Math `3`.

3.1.5.2 Inline Style

When using inline Math, use \$ around the Math equations. For example, writing $E = mc^2$ shows $E = mc^2$. And writing a symbol with height like integral, for example, writing $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$ as the same equation as Math 1 will show $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$. By the way, note that \ and _ have to be escaped because it is the same symbol of Markdown.

You can use `\displaystyle` inside \$ if you want to use a large type of formula. For example, writing $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$ shows $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$. But anyway it is not recommended because the height is different and not beautiful.

3.1.6 Image

For Image, you can use a Markdown syntax for Image as is, but the option will be written in the position of alt, and then the syntax will be the style of `![options](path)`. It is possible to insert an image as a standalone style, an inline style, and a floating style.

3.1.6.1 Standalone Image

To show a standalone image, write it as an independent paragraph between blank lines.

```
![scale=0.6](kinxlogo.png)
```

For above, the image will be inserted as below. The image width is adjusted to 60% for the page width by the option of `scale=0.6`. The ratio of width and height is stayed.



3.1.6.2 Inline Image

Here is an example of inline style. When inserting an image as inline style, you write it directly inside text like the following.

```
The file icon is changed to ![scale=0.08,offsetY=-5.0](zip256.png).
```

In this case, “The file icon is changed to .” will appear. You have to adjust `scale` and `offsetY` according to the original size of image.

3.1.6.3 Floating Image

To float an image, use `float=left` or `float=right` to the option. The image of “[Figure 3.1 Hyphenation and Justification](#)” at the page of [9](#) is one of example of this.



Fig 3.2 Damselfly

The picture on the left is provided as a Public Domain². The image is located as a floating image like this. The option of `scale` can be used with a floating image, but its size is limited within 70%³ of the area without margins.

And also, you can write multiple paragraphs around a floating image. When a text of a paragraph is finally reached at the bottom of a floating image, the text width of a line will be automatically back to the width of the page, and the text is located around a image naturally.

In this case, the right picture is an example of a floating image which is located to the right of the text. The image is also Public Domain as same as the above picture.

There is a note that the top of an image have to be set to the top of a paragraph. You can not put an image inside a paragraph. If there is an image you want to float when some paragraph starts, the image will be located to the left or right which you specified when paragraph starts.



Fig 3.3 F14 Tomcats

3.1.6.4 Image Options

You can use options as follows.

Table 3.1 Image Options

Option	Value	Meaning
<code>float</code>	<code>left, right</code>	A location of a floating image.
<code>scale</code>	<code>0.0~ 1.0</code>	A scale against the area between margins.
<code>caption</code>	Text	A caption of an image.
<code>box</code>	<code>BOX_NORMAL</code>	Output a box with a normal line.
	<code>BOX_THIN</code>	Output a box with a thin line.
	<code>BOX_THICK</code>	Output a box with a thick line.
<code>padding</code>	Real number	The margin between an image and a text.

² <https://free-images.com/>

³ The margin between an image and a text is included in this 70%.

3.1.7 Chart

Chart is also supported. It supports both a stanalone style and a floating style.

3.1.7.1 Standalone Chart

For a standalone style of chart, use a Code Block with the language of `chart`. For example, it is written in JSON structure as below. In addition to chart information like `width` and `height`, Chart.js⁴ data itself have to be put in the field of `options`.

```
1  ```chart
2  {
3      width: 800,
4      height: 400,
5      fontSize: 16,
6      scale: 1.0,
7      caption: "Radar Chart Example",
8      options: {
9          type: "radar",
10         data: {
11             labels: [["Eating", "Dinner"], ["Drinking", "Water"],
12                     ["Sleeping", ["Designing", "Graphics"]],
13                     ["Coding", "Cycling", "Running"]],
14             datasets: [
15                 {
16                     label: "My First dataset",
17                     backgroundColor: "rgba(255, 0, 0, 0.2)",
18                     borderColor: "red",
19                     pointBackgroundColor: "red",
20                     data: [ 10.1, 80.0, 72.2, 73.3, 55.0, 68.5, 92.0 ]
21                 },
22                 {
23                     label: "My Second dataset",
24                     backgroundColor: "rgba(0, 0, 255, 0.2)",
25                     borderColor: "blue",
26                     pointBackgroundColor: "blue",
27                     data: [ 30.9, 77.1, 49.9, 50.0, 67.8, 71.0, 22.8 ]
28                 }
29             ],
30             options: {
31                 legend: {
32                     position: "top",
33                 },
34                 scale: {
35                     ticks: {
36                         beginAtZero: true
37                     }
38                 }
39             }
40         }
41     }
42 ```
```

This will be shown as [Figure 3.4 Radar Chart Example](#).

⁴ <https://www.chartjs.org/>

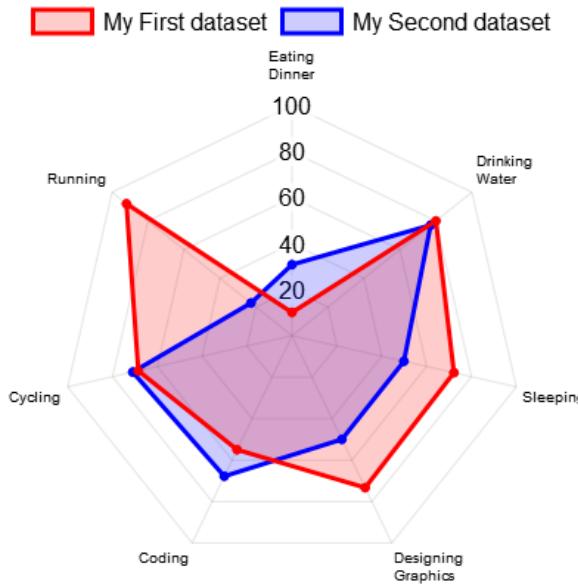


Fig 3.4 Radar Chart Example

3.1.7.2 Floating Chart

A floating chart is also supported as same as an Image.

You can set `float: { right: true }` to the option, the chart will be floated and the next paragraph will be located around that chart.

In this case, it is a line chart. `scale` means a width against the area without margins as same as an image. You can put the caption for both a standalone style and a floating style, and a chart will inserted to the document as an image. This caption will be shown in the List Of Figures when setting to output Table Of Contents. And also it is same as an floating image, a paragraph is naturally located under the chart.

The following is an example of the code. By the way, in this time `options` is omitted due to space limitations, but the `options` means the option of Chart.js.

```

1  ```chart
2  {
3      float: { right: true },
4      width: 480, height: 300, scale: 0.4, caption: "Line Chart Example",
5      options: {
6          type: "line",
7          ...(omitted)
8      }
9  }```
10

```

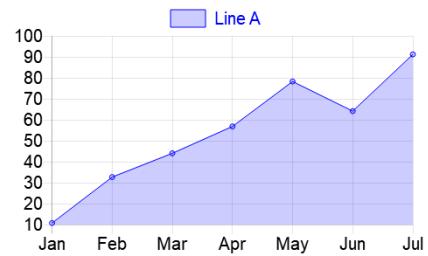


Fig 3.5 Line Chart Example

3.1.8 Table

3.1.8.1 Markdown Table

Table is also supported. The table can be written as a normal Markdown table, and the table is automatically generated.

For example, “[Table 3.1 Image Options](#)” is written as below. Use a `<context />` tag for the option which is not directly specified as a Markdown syntax.

```
1 <context label="Table:ImageOptions"/>
2 <context caption="Image Options"/>
3 | Option | Value | Meaning |
4 | ----- | ----- | ----- |
5 | `float` | `left`, `right` | A location of a floating image. |
6 | `scale` |  $0.0 \sim 1.0$  | A scale against the area between margins. |
```

Using a normal Markdown syntax, you can make it centering, or aligning to the left or right. Besides, you can use Math formula inside your table. If the cell content is becoming too long, you can use `<context cell-i-j="..." />` to write a long cell content. In this case, (i, j) means the location without a header. For example, $(0, 0)$ means A1 cell in the following table. You see i means a row number and j means a column number.

```
1 <context label="Table:TableExample"/>
2 <context caption="Example of Table"/>
3 <context vline-left="single"/>
4 <context vline-right="single"/>
5 <context vline-inside="single"/>
6 <context hline-header="double"/>
7 <context hline-inside="single"/>
8 <context cell-2-1="$\displaystyle\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$"/>
9 | Left | Center | Right |
10 | :--- | :-----: | -----: |
11 | A1 | Aligned to the center. | Aligned to the right. |
12 | A2 | Cell $(1,1)$ | Cell $(1,2)$ |
13 | A3 | - | Cell $(2,2)$ |
```

Here is the output result below.

Table 3.2 Example of Table

Left	Center	Right
A1	Aligned to the center.	Aligned to the right.
A2	Cell (1, 1)	Cell (1, 2)
A3	$\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$	Cell (2, 2)

3.1.8.2 Table Options

`<context />` tag can specify the parameters which can not be specified in a normal Markdown syntax. “[Table 3.3 Table Options](#)” shows the list of the parameters which can be specified for the table. Those parameters are used as a temporary and they will be removed right after it being used. Therefore the parameters specified once can not be used for multiple tables. You have to set it again if you want to use the same parameters for a next table.

But you can change a default value for the parameters except `label` and `caption`. For that, change the value for the name with `-default` at the end of the parameter name. For example, you can change the default value of `vline-left` by changing the value for `vline-left-default`.

Table 3.3 Table Options

Name	Value	Meaning
<code>label</code>	Text	The label for Cross-Reference.
<code>caption</code>	Text	The caption of a table.
<code>vline-left</code>	<code>single, double, false</code>	Vertical line type of the left on a table (<code>false</code> by default).
<code>vline-right</code>	Same as above	Vertical line type of the right on a table (<code>false</code> by default).
<code>vline-inside</code>	Same as above	Vertical line type inside a table (<code>false</code> by default).
<code>hline-top</code>	Same as above	Horizontal line type of the top on a table (<code>single</code> by default).
<code>hline-bottom</code>	Same as above	Horizontal line type of the bottom on a table (<code>single</code> by default).
<code>hline-header</code>	Same as above	Horizontal line type under a header (<code>single</code> by default).
<code>hline-inside</code>	Same as above	Horizontal line type inside a table (<code>false</code> by default).
<code>cell-i-j</code>	Text	The text in the cell of (i, j) . i is a row number and j is a column number, and both are 0 origin.
<code>limit-column</code>	Integer number	The column number for the minimum width to determine a cell width limitation.
<code>limit-width</code>	Real number	The minimum width of a cell width.

3.1.9 Font

3.1.9.1 Bold, Italic, BoldItalic

Bold, *Italic*, ***BoldItalic*** is shown by a normal Markdown syntax. Here is examples.

Table 3.4 How to write Bold, Italic, and BoldItalic

Markdown	Output
Bold	Bold
<i>*Italic*</i>	<i>Italic</i>
BoldItalic	<i>BoldItalic</i>

3.1.9.2 Use Fonts

If you want to use a font which is not loaded by default, load it by yourself, and you can use it. The following shows how to load a font file. It is necessary to specify 4 parameters separated by comma.

```
1 <font-load info="Name,type,shape,FileName.ttf" />
```

You can freely use the font after loading once. The parameter meaning is as follows.

- **Name** is the own name of the font you can specify.
- **type** is a font type. It should be **serif**, **sans**, **monotype**.
- **shape** is a font shape. It should be **regular**, **bold**, **italic**, **bolditalic**.
 - **regular** means a normal shape used in the normal sentence.
 - **bold**, **italic**, and **bolditalic** is for the shape meant by that name.

Use **\font** command to change the font as a following example. By the way, you have to escape \ of a command name, [, and] like \\, \\[, and \\] because it is a Markdown special character. The following example shows how to use **Parisienne-Regular.ttf** font⁵.

```
1 <font-load info="Parisienne,serif,regular,Parisienne-Regular.ttf" />
2 - Changing the font is available only with a scope like
3 ''\\font\[name=Parisienne\]{This is a pen.},'
4 and the font will be restored here.
```

This shows as below. This is available inside a scope of **\font** command as you see in the following example.

- Changing the font is available only with a scope like “*This is a pen.*,” and the font will be restored here.

⁵ This TrueType font file is already included in the KiTTy package, but it is not loaded by default. Load it by yourself to use it.

3.1.9.3 Font Size (Direct)

Font size is specified by `size` parameter of `\font` command.

```
1  ''After this, the font size will be \font[size=7pt]{shrink to 7pt}.
2  And after this, the font size will be \font[size=15pt]{extend to 15pt}. ''
3  will be shown.
4  And also, for example, '\font[size=1.2em]{1.2 times larger size}' shows
5  \font[size=1.2em]{1.2 times larger size} will be shown.
```

“After this, the font size will be shrink to 7pt. And after this, the font size will be extend to 15pt.” will be shown. And also, for example, `\font[size=1.2em]{1.2 times larger size}` shows 1.2 times larger size will be shown.

You can use the unit of size as you see in the above example. The unit you can specified is as follows.

Table 3.5 Unit for Font Size

Unit	Meaning
em	Relative size based on the current font size.
ex	Relative size based on the x height at the current font size.
px	By Pixel.
pt	By Point.
pc	By Pica, which means 1/6 inch.
mm	By Millimeter.
cm	By Centimeter.
in	By Inch.

3.1.9.4 Font Size (Relative)

Use `\bigger`、`\smaller` to change the size with a relative size. The size will be changed by +1 point for `\bigger` or -1 point for `\smaller` inside its scope.

```
1  This is an example of '\bigger'.
2  \bigger{This is, \bigger{this is, \bigger{this is} a sentence,} a sentence}.
3  For '\smaller',
4  \smaller{This is, \smaller{this is, \smaller{this is} a sentence,} a sentence}.
```

This is an example of `\bigger`. This is, this is, this is a sentence, a sentence. For `\smaller`, This is, this is, this is a sentence, a sentence.

3.1.10 Color

3.1.10.1 Text Color

Use `\color` to change the text color. Here are examples below.

```
1 * \color[red]{\bold{Red}}.  
2 This line should be colored by the name of `red`.}  
3 * \color[green]{\bold{Green}}.  
4 This line should be colored by the name of `green`.}  
5 * \color[blue]{\bold{Blue}}.  
6 This line should be colored by the name of `blue`.}  
7 * \color[cyan1]{\bold{Cyan}}.  
8 This line should be colored by the name of `cyan1`.}  
9 * \color[magenta1]{\bold{Magenta}}.  
10 This line should be colored by the name of `magenta1`.}  
11 * \color[yellow]{\bold{Yellow}}.  
12 This line should be colored by the name of `yellow`.}  
13 * \color[R=0,G=64,B=255]{\bold{RGB}}.  
14 This line should be colored by RGB value of `RGB=0,64,255`.}  
15 * \color[C=0.5,M=0.8,Y=0.2,K=0.0]{\bold{CMYK}}.  
16 This line should be colored by CMYK value of `CMYK=0.5,0.8,0.2,0.0`.
```

- **Red.** This line should be colored by the name of `red`.
- **Green.** This line should be colored by the name of `green`.
- **Blue.** This line should be colored by the name of `blue`.
- **Cyan.** This line should be colored by the name of `cyan1`.
- **Magenta.** This line should be colored by the name of `magenta1`.
- **Yellow.** This line should be colored by the name of `yellow`.
- **RGB.** This line should be colored by RGB value of `RGB=0,64,255`.
- **CMYK.** This line should be colored by CMYK value of `CMYK=0.5,0.8,0.2,0.0`.

See [Appendix C Color Examples](#) about supported color names.

3.1.11 Ligature and Special Character

3.1.11.1 Ligature

Now the following 5 ligatures is only supported.

- fi ... fi
- fl ... fl
- ff ... ff
- ffi ... ffi
- ffl ... ffl

3.1.11.2 “ and ”

both “ and ” are automatically replaced by writing 2 single quotes.

```
1 This is an ''example'' of double-quote.
```

This is an “example” of double-quote, will be shown.

3.1.11.3 Backquote

You can use a special command because backquote is same as a special character in Markdown. For this purpose, the special tag of <backqN /> where N is the number of backquote can be used. Write <backq3 /> to show ` ```, and write <backq2 /> to show ` ` . If you want to use Mototype, you can use <monotype-backqN />.

3.1.12 Programming Code Block

3.1.12.1 Code Block

Programming Code is written as a Code Block. It shows with a line number and a shadow box by default.

```
1 ` ```
2 class Test {
3     public test() {
4         # Test Method.
5     }
6 }
7 ` ````
```

Above example will show the following code⁶。

```
1 class Test {
2     public test() {
3         # Test Method.
4     }
5 }
```

Write the following to erase a line number and to be without a shadow.

```
1 ```:lineNumber=false,box=BOX_NORMAL
2 class Test {
3     public test() {
4         # Test Method.
5     }
6 }
7 ````
```

⁶ It is hard to distinguish those because the same style, but the code with ` `` ` is the example to write.

Here is the result.

```
class Test {  
    public test() {  
        # Test Method.  
    }  
}
```

3.1.12.2 Code Block Option

Specify true or false to the `lineNumber` option. The available values for the `box` option are as follows.

Table 3.6 The available values for box

Value	Meaning
BOX_NORMAL	Show a box with a normal line width.
BOX_THIN	Show a box with a thin line width.
BOX_THICK	Show a box with a thick line width.
BOX_SHADOW	Show a box with a shadow. (by default)

3.1.13 Title, Cover Page, and Contents

3.1.13.1 Title Information

KiTTy will automatically generate a title page, a cover page, and a table of contents. The necessary parameters have to be set before the first chapter title. The following example is the example of parameters used by this document. A title, author, and date is set by % at the head of a line. You should write only % for no information because its order can not be changed. The paragraph right after the setting by % will be a sub title.

```
1 % KiTTy  
2 % Kray-G, Mr.Diamond Global Blue Publisher  
3 % September 18, 2020  
4  
5 Small and Easy, but Beautiful Design For You
```

3.1.13.2 Other Parameters for Title

About the other parameters for a title, specify those by `<param />` tag before the first chapter title. The following is the example in this document.

```
1 <param style="JBookA4"/>  
2 <param titleSize="78.8"/>  
3 <param subtitleSize="14.4"/>  
4 <param backgroundImage="back.jpg"/>
```

The parameter which can be set depends on the style. This example shows parameters which can be set in the `BookA4` style. The meaning of each parameter is as follows.

- This document will use the `BookA4` style.
- The size of a title text is 78.8pt.
- The size of a sub title text is 14.4pt.
- The cover page will use the `back.jpg` for the background image.

3.1.13.3 Parameters for Table Of Contents

To set the following command, and the document will show a table of contents. By `with` parameter, you can choose if the document will show a list of figures and a list of table⁷.

```
1 <toc with="lof,lot"/>
```

3.1.13.4 Customize Style

There are default values of parameters for each style. Use `<style-info />` tag to change those default values. For example, write the tag like a following to change the chapter design. See [A.2 Chapter Design](#) for the chapter design.

```
1 <style-info name="chapter.style" value="BigChapter3" />
```

See [B.1 Style Parameters](#) for details of style parameters which you can change.

3.1.14 Heading

The heading like Chapter and Section will be shown by the line lead by # as a normal Markdown syntax. Please look at the following.

Table 3.7 Headings in Markdown

Symbol	Meaning
#	Chapter
##	Section
###	Sub Section
####	Sub Sub Section

The numbering for the above Headings will be automatically done, and Cross-Reference ([3.1.15 Cross-Reference](#)) will be available. You can use a heading text as a label for Cross-Reference.

⁷ Set `lof` for a list of figures, and `lot` for a list of tables. If you want to show both, set it separated by comma like `lof,lot`.

3.1.15 Cross-Reference

Cross-Reference is supported. The following list is the target which you can use for Cross-Reference.

Table 3.8 Command List for Cross-Reference

Command	Shown Text
\ref{label}	A chapter number, a section number, a figure and table number, or Math formula number.
\textref{label}	The text for a label.
\nameref{label}	“Number and Text” for a label.
\pageref{label}	The page number for a label.

If the reference appears before an actual source of the reference, it can not be solved. In that case, re-run KiTTy to solve the reference.

3.1.16 Quotation

Quotation will be shown by ‘>’ at the head of a line. This is also a normal Markdown syntax. You can not use the paragraph commands ([4.1.1 Paragraph Commands](#)) of Markdown, but the inline commands ([4.1.2 Inline Commands](#)) and KiTTy commands ([4.2 KiTTy Commands](#)) are available in a quotation syntax. And nesting a quotation syntax is also available. In that case, each quoted text is always recognized as a paragraph. If you do not want to do indentation, use \\noindent command at the head of paragraph. For example, here is an example of the introduction of this document.

```
1 > **KiTTy** means **Ki**nx **T**iny **Ty**pesetting,  
2 > which is a simple typesetting system implemented by Kinx.  
3 > > \\noindent It also provides a translator from Markdown,  
4 > > then you can typeset a Markdown document and can get a beautiful document.  
5 > > This document itself is also the example typeset by this system.  
6 >  
7 > The objective is similar to \\LaTeX, it is an objective to typeset beautifully for  
8 > a document managed as a text file.  
9 > To be concretely, it is never going to be alternative,  
10 > but the objective is being more useful in the use case like your personal situation  
11 > by followings.
```

The result is as below.

KiTTy means Kinx Tiny Typesetting, which is a simple typesetting system implemented by Kinx.

It also provides a translator from Markdown, then you can typeset a Markdown

document and can get a beautiful document. This document itself is also the example typeset by this system.

The objective is similar to LATEX, it is an objective to typeset beautifully for a document managed as a text file. To be concretely, it is never going to be alternative, but the objective is being more useful in the use case like your personal situation by followings.

3.1.17 Footnote

Footnote is also supported. A footnote is the style of [^label] in Markdown. Write the actual footnote text by the style of [^label]: ... in the independent paragraph. For example, see the example below.

```
1 This is a footnote[^f1] in this paragraph.  
2  
3 [^f1]: This is a footnote text.
```

Here is a result, “This is a footnote⁸ in this paragraph.” You will see the footnote will be shown at the bottom of this page.

3.2 Typesetting Features in Japanese

This is only for Japanese requirements. If you do not need this section, skip and go to the next section of [3.3 PDF Features](#).

3.2.1 Japanese Hyphenation

The following Japanse hyphenation is now supported.

- A line head Japanese hyphenation processing.
- A line end Japanese hyphenation processing.
- Line breaking is prohibited in a group ruby.

3.2.2 Japanese Furigana

3.2.2.1 About Japanese Furigana

Japanese Furigana⁹ is supported. Furigana is sometimes told as Ruby¹⁰. Here is the specification.

- The width is the longer width in a parent text and its ruby.
- Both a mono ruby and a group ruby are supported.
- Both Japanese and English can be used in both a parent text and its ruby.

⁸ This is a footnote text.

⁹ Furigana means syllabic characters to indicate pronunciation. To be simple, you can know how to read Kanji.

¹⁰ It is also called ruby because the name of the 5.5-point size imported from England in the latter half of the 19th century was ruby.

This is only for Japanese requirement. In Japanese, there is a way, which is named as “Furigana” or “Ruby”, to write a text. It is to write a small text as a way to read on the top of a main sentence. For example, it is used like “この**ほん**本はとても面白いです。” Usually, very young children can't read Kanji. And even if it were adults, there is a case they can't read a difficult Kanji or the Kanji they do not use usually. Considering those cases, Furigana is used like “**ほん**本。”

3.2.2.2 How To Use Furigana

KiTTe supports 2 types of Furigana as a mono ruby and a group ruby. When it is a mono ruby, write ‘|’ as a separator like `\ruby{[す|てき]}{素敵}`. In this case, it shows “素敵” and you will see the small words on each Kanji. On the other hand, when it is a group ruby, write `\ruby{[すてき]}{素敵}` without ‘|’. In this case, it shows “素敵” and you will see the small words are aligned at both ends in the entire kanji. You can see “て” is a center between “素” and “敵” at the example. Besides, line breaking is prohibited in a group ruby, but it is allowed in a mono ruby. By the way, “素敵” means “wonderful.”

3.2.2.3 Furigana Examples

The following text is a beginning part of *The Spider's Thread*¹¹ from Ryunosuke Akutagawa. It is shown as block quote style for easy to read. First, it is an example of a group ruby.

```
1 > \noindent ある日の事でございます。\\ruby{[おしゃかさま]}{御釈迦様}は  
2 > 極楽の\\ruby{[はすいけ]}{蓮池}のふちを、  
3 > 独りでぶらぶら御歩きになっていらっしゃいました。
```

Here is the result.

ある日の事でございます。御釈迦様は極楽の蓮池のふちを、独りでぶらぶら御歩きになっていらっしゃいました。

Next, it is an example of a mono ruby. When it is a mono ruby, insert ‘|’ as a separator.

```
1 > \noindent ある日の事でございます。\\ruby{[お|しゃ|か|さま]}{御釈迦様}は  
2 > 極楽の\\ruby{[はす|いけ]}{蓮池}のふちを、  
3 > 独りでぶらぶら御歩きになっていらっしゃいました。
```

Here is the result. You can see that the position of a ruby of “おしゃかさま” for “御釈迦様” is different. The position of “はすいけ” for “蓮池” is same, but the line break can be inserted between “蓮” and “池” as it is a mono ruby.

ある日の事でございます。御釈迦様は極楽の蓮池のふちを、独りでぶらぶら御歩きになっていらっしゃいました。

¹¹ [The Spider's Thread - Ryunosuke Akutagawa](#)

3.3 PDF Features

3.3.1 External Link by URL

Write a URL directly to generate an external link for URL automatically. For example, write `https://github.com/Kray-G/kinx` to generate the link to <https://github.com/Kray-G/kinx> on URL text itself. If you want to change the link text instead of URL, use `\url` command. For example, write `\url[https://github.com/Kray-G/kinx]{KinX}` to show [KinX](#), and you can click the link text to jump to the external web site.

3.3.2 Link by Cross-Reference

As a document a Cross-Reference ([3.1.15 Cross-Reference](#)) is supported, and the internal link will be generated for that as a PDF document. You can click that link to jump to the related document place like headings, figures, or tables.

3.3.3 Bookmark

PDF bookmark is also supported. The link for each heading is automatically generated as a PDF bookmark. Click the heading displaying on the bookmark, and you can jump to the related heading.

Note that PDF link text can not include any commands like `KiTTy` command. Therefore do not use any command in the heading text name¹².

¹² As excepted, only `TeX`, `LATEX`, `KATEX` are replaced by the plain text of `TeX`, `LaTeX`, `KaTeX`.

Chapter 4

Command Details

This chapter will describe available commands. As a command type, roughly there are 2 types of commands as Markdown command or KiTTy command. And also in Markdown commands, there are 3 types as a normal Markdown syntax, HTML, and a special command for KiTTy. KiTTy commands are basically used in a paragraph, but the command will start with \ symbol and will use \[and \]. Therefore you have to use \\ instead of \' like “\\command\\[\\]” style because those symbols are a special character in Markdown to be an escape character.

4.1 Markdown Commands

4.1.1 Paragraph Commands

KiTTy will basically use a normal Markdown syntax, but there is a case the syntax has an own condition. See the table below.

Table 4.1 Markdown Paragraph Commands

Markdown	Syntax Detail
Paragraph	Separated by blank line.
Line Break	Put double spaces at the end of line.
Code Block	The line which is started with 4 spaces, or a paragraph between ```.
Blockquote	The line which is started with '>'.
Heading	The line which is started with '#'. The style of underline, which is == and -- under the heading, is not supported.
Itemization	The line which is started with '*', or the number like 1. etc. Blank lines can not be included between items.
Table	Normal Markdown table is supported. Use <context /> command for additional information for table.
Figure	When there is only one line of !... in a paragraph. Otherwise it will be dealt with an inline command.
Footnote	The line which is started with “[^name]:” in a paragraph, and name is anything you can name.

4.1.2 Inline Commands

KiTty will basically use a normal Markdown syntax also for inline commands, but there is a case the syntax has an own condition. See the table below.

Table 4.2 Markdown Inline Commands

Purpose	Syntax
Italic	The string wrapped between '*' and '*'.
Bold	The string wrapped between '***' and '***'.
Bold Italic	The string wrapped between '****' and '****'.
Inline Code	The string wrapped between ` and `.
Inline Image	Use !... style.
Link	Automatically generated by URL. Now the style of ... is not supported.
Footnote Tag	Write “[^name]” in a paragraph. name is anything you can name.

4.1.3 HTML Commands

KiTty will use some HTML tags as a command. It is used mainly for setting a parameter and controlling a typesetting without showing in the document.

Table 4.3 HTML Commands

Command	Meaning
<toc />	To show a Table of Contents. This have to be set before the first chapter. The value of lot and lof can be set to the with option.
<param />	To set an initial value of page. This have to be set before the first chapter.
<context />	To set an additional parameter temporarily for some commands. The parameter depends on the command.
<clear-float />	To clear floating setting and to stop wrapping around an image.
<pagebreak />	To insert a page break forcibly, and to do processing of page break.
<set-column />	To set the number of column. The number of columns is set to value attribute, and column's height is set to height attribute.
<style-info />	To change the parameter of a style. The parameter is set to name and value attribute.
<appendix />	To indicate the start of Appendix.

4.2 KiTTY Commands

4.2.1 Paragraph Scope Commands

Paragraph scope commands will do process the text inside {} as a paragraph. The following table shows a description of commands.

Table 4.4 Paragraph Scope Commands

Command	Meaning
\bigger{}	To change the font size by +1pt.
\smaller{}	To change the font size by -1pt.
\bold{}	To set the font shape to bold. It can be also shown by enclosing a text by **.
\italic{}	To set the font shape to italic. It can be also shown by enclosing a text by *.
\color[params]{}	To change a color of a text. You can use a color name like red, R=r, G=g, B=b, or C=c, M=m, Y=y, K=k.
\font[params]{}	To change a font information like \font[size=1em]{}. size ... to change the font size. name ... to change the font.
\footnote{}	To generate a footnote. It can be also shown by [^name].
\monotype{}	To set to use a non-proportional font.
\raise[params]{}	To change an offset of Y, and to move a text to upside. The offset is set by height attribute.
\lower[params]{}	To change an offset of Y, and to move a text to downside. The offset is set by height attribute.
\sans{}	To set to use Sans Serif.
\url[params]{}	To generate a link text. URL should be written in params.

In the {} of all commands you can write a paragraph although just {} is written in above examples. For example, if you write \\sans{''This is Sans Serif Font.''}, it will show “This is Sans Serif Font.”

4.2.2 Standalone Commands

Standalone command will not process the text inside {} as a paragraph. The text inside {} is used as one of parameters. By the way you can omit both [] and {}, but you have to

insert a space or only {} to distinguish from a next word if necessary. For example, you can write `\noindent{}`.

Table 4.5 Standalone Commands

Command	Meaning
<code>\TeX</code>	<code>\TeX</code> logo will be shown.
<code>\LaTeX</code>	<code>\LaTeX</code> logo will be shown.
<code>\KaTeX</code>	<code>\KaTeX</code> logo will be shown.
<code>\noindent</code>	Indentation will be erased. The paragraph without indentation will be generated.
<code>\apos</code>	Apostrophe will be shown.
<code>\hs</code>	A space, which width is the width of a small <i>i</i> , will be shown.
<code>\hspace{width}</code>	To change the X position by the value of <code>width</code> . To move to the right by a plus value.
<code>\vspace{height}</code>	To change the Y position by the value of <code>height</code> . To move to the down by a plus value.
<code>\ref{label}</code>	To generate a Cross-Reference with the number specified by <code>label</code> .
<code>\pageref{label}</code>	To generate a Cross-Reference to the page specified by <code>label</code> .
<code>\textref{label}</code>	To generate a Cross-Reference with the text specified by <code>label</code> .
<code>\nameref{label}</code>	To generate a Cross-Reference with the number and the text specified by <code>label</code> .
<code>\pack{text}</code>	To make it be recognized as no line breaking in <code>text</code> .
<code>\ruby[Ruby]{Parent-Text}</code>	To show <code>Ruby</code> on the top of <code>Parent-Text</code> . For example of the left, <code>Parent-Text</code> .

Chapter 5

How To Extend Features

This chapter describe how to extend features in KiTTy. However all information can not be described in this chapter, and some other document will be provided in the future. Now this describe only an outline of a rough way to extend and add features.

5.1 Adding Style

5.1.1 Document Style

The style definition file of a document will be stored to the following path when the install path is \$INSTALL.

```
$INSTALL/lib/std/typesetting/style/additional/*.kx
```

The files here are loaded in no particular order, so styles cannot be related, for example, class inheritance. The standard styles, which can be parent classes, are stored below.

```
$INSTALL/lib/std/typesetting/style/basic/*.kx
```

You can use the style just by placing a file under the folder of **additional**. When adding it under **basic**, please note that you need to add the load process to the following file.

```
$INSTALL/lib/std/typesetting/style/Styles.kx
```

Therefore, you should normally add the style definition file under **additional**.

5.1.1.1 Style Definition File Example

Here is the example of ArticleA4_2Cols which is provided by default.

```
1 using typesetting.style.basic.ArticleA4;
2
3 namespace Typesetting {
4     namespace Style {
5
6         class ArticleA4_2Cols : Typesetting.Style.ArticleA4 {
7             @style.columns = 2;
8         }
9
10    } # namespace Style
11 } # namespace Typesetting
```

The class name is the style name. And the class have to be joining in the namespace of `Typesetting.Style`. This example shows the class `ArticleA4_2Cols` is inherit from `ArticleA4` style, and set the number of columns to be 2 columns. The style of `ArticleA4` which is the base of all styles, are located to the path of `basic/ArticleA4.kx`. The parameters that you can set are all described in that file. Please see the file of `ArticleA4` to know what parameters you can set.

5.1.2 Title Style

You can create a plug-in of title design. It can be also an independent page as a cover page. The style definition of a title page is placed as below.

```
$INSTALL/lib/std/typesetting/style/title/*.kx
```

In your document, use the following tag to change a style of a title. The `yourstylename` means a file name without an extension of the style definition file, when you follow how to create as described below.

```
1 <style-info name="title.style" value="yourstylename" />
```

5.1.2.1 Title Style Example

For example, please look at the definition of `StandardArticle`.

By `Typesetting.Style.Title[__FILE__.stem()]`, the property name will be the name without an extension. And write it to add a function object which defines a title design to that property.

```
1 namespace Typesetting {
2   namespace Style {
3
4     Typesetting.Style.Title[__FILE__.stem()]
5       = function(info, context, core, title, opts) {
6         # drawing a Title Design
7       };
8
9   } # namespace Style
10 } # namespace Typesetting
```

See [A.1 Title Design](#) about the pre-designed title style.

5.1.3 Chapter Style

You can also create a plug-in of a chapter design. The chapter design definition file is placed to the foloowing path.

```
$INSTALL/lib/std/typesetting/style/chapter/*.kx
```

If you want to change the chapter style, use the following tag. The `yourstylename` means a file name without an extension of the style definition file, when you follow how to create as described below.

```
1 <style-info name="chapter.style" value="yourstylename" />
```

5.1.3.1 Chapter Style Example

For example, please look at the example of `StandardBook`.

By `Typesetting.Style.Chapter[__FILE__.stem()]`, the property name will be the name without an extension. And write it to add a function object which defines a chapter design to that property.

```
1 namespace Typesetting {
2   namespace Style {
3     ...
4     Typesetting.Style.Chapter[__FILE__.stem()]
5       = function(info, context, core, text, opts) {
6         # drawing a Chapter Design
7       };
8
9   } # namespace Style
10 } # namespace Typesetting
```

See [A.2 Chapter Design](#) about the pre-designed chapter style.

5.2 Hyphenation Rules

5.2.1 Add The Rule

The rule file is stored under the following path. By the way, the rule for Japanese is absolutely same as the rule for English, because it is no differences between them.

```
$INSTALL/lib/std/typesetting/lang/*.kx
```

If you want to change the chapter style, use the following tag. The `yourprocname` means a file name without an extension of the style definition file, when you follow how to create as described below.

```
1 <style-info name="hyphenationRule" value="yourprocname" />
```

By `Typesetting.insertGlue[__FILE__.stem()]`, the property name will be the name without an extension. And write it to add a function object which defines a process of hyphen-

ation to that property. For example, if the file name is ja.kx, Typesetting.insertGlue.ja will be defined.

```
1 Typesetting.insertGlue[__FILE__.stem()] = _function(info, Linebreak, wordlist, nodes) {
2     # Hyphenation Rules
3
4     wordlist.each { &(node, i):
5         # ...
6         nodes.push(Linebreak.Glue(0, Linebreak.Infinity, 0));
7         nodes.push(Linebreak.Penalty(0, -Linebreak.Infinity, 0));
8         nodes.push(node);
9         # ...
10    };
11};
```

5.2.1.1 Create The Rule

wordlist as an argument is the list of words to be analyzed. By its content, add a word object to a nodes array. The following is the basic rule to add a node.

- To prohibit a line break before the node, a Penalty with Linebreak.Infinity should be inserted before the node.

```
1 nodes.push(Linebreak.Penalty(0, Linebreak.Infinity, 0));
2 nodes.push(node); # Target Node
```

- To prohibit a line break after the node, a Penalty with Linebreak.Infinity should be inserted after the node.

```
1 nodes.push(node); # Target Node
2 nodes.push(Linebreak.Penalty(0, Linebreak.Infinity, 0));
```

- To do a line break before the node, the list will be as below.

```
1 nodes.push(Linebreak.Glue(0, Linebreak.Infinity, 0));
2 nodes.push(Linebreak.Penalty(0, Linebreak.Infinity, 0));
3 nodes.push(node); # Target Node
```

- To insert a Glue before the node, the Glue will be inserted as below.

```
1 nodes.push(Linebreak.Glue(width, stretch, shrink));
2     # width ..... Width of Glue
3     # stretch ... Maximum additional width limitation.
4     # shrink ... Minimum shrunked width limitation.
5 nodes.push(node); # Target Node
```

Finally, `nodes` will be a list including hyphenation rules. Hyphenation will be done based on this information.

5.3 Font

5.3.1 Add New Font

If you want to add the font, place the font file at the following path. By the way, only TrueType font is supported so far¹.

```
$INSTALL/lib/fonts
```

5.3.2 Add System Font

If you want to use a system font provided by an operating system, specify the file name of the font you want to use. When doing so, a search path is different between Windows and Linux.

Table 5.1 Font File Search Path

OS	Search Path
Windows	C:/Windows/Fonts
Linux	/usr/share/fonts/truetype

You can specify the file name as a relative path from above search path. For example, when you want to use Times New Roman on Windows, do the following.

```
1 <font-load info="Times,serif,regular,times.ttf" />
2
3 * This is a Regular Style of a default font.
4 * \font\[name=Times\]{Times New Roman of Regular Style}
```

The font information is embedded to a PDF file, so the PDF file can be read and displayed correctly even on the other OS. However, as a matter of course, note that if you use a font that can not be detected on a different OS, the typesetting process itself will fail because the font can not be found.

5.4 Commands

You can also add KiTTy commands. KiTTy commands are located to the following path.

```
$INSTALL/lib/std/typesetting/command/inline/*.kx
$INSTALL/lib/std/typesetting/command/paragraph/*.kx
```

¹ This is a limitation of libharu.

Standalone Commands ([4.2.2 Standalone Commands](#)) are stored under `inline` folder, and Paragraph Scope Commands ([4.2.1 Paragraph Scope Commands](#)) are stored under `paragraph` folder.

Put your additional command file, and you can use the new command automatically. By the way, the file name seems to be a command name in almost all cases, but the class name is actually used as a command name. So far a class name and a file name is same in almost all commands. A command is recognized when the name is absolutely matched or matched with a lower case.

5.4.1 Define Standalone Commands

Here is the template for Standalone Commands. It is necessary to implement an `exec` method.

Though `WordSet` object is returned in this case, but you can return `Word` object, `LineBreak.Box` object, `LineBreak.Glue` object, `LineBreak.Penalty` object, or null. You should return one of those if necessary.

```
1 namespace Typesetting {
2     namespace Command {
3
4         class Command(info_, context_) {
5             public exec(params) {
6                 var ws = new Typesetting.WordSet(info_);
7                 # ...
8                 return ws;
9             }
10        }
11    }
12 } # namespace Command
13 } # namespace Typesetting
```

Table 5.2 KiTTy Command Available Object Type

Object	Meaning
<code>WordSet</code>	To return multiple words.
<code>Word</code>	To return a word.
<code>LineBreak.Box</code>	To return a Box instead of Word or WordSet. For example, use it when the text width is fixed.
<code>LineBreak.Glue</code>	To insert a Glue.
<code>LineBreak.Penalty</code>	To insert a Penalty.
<code>null</code>	Nothing to return when only changing the context, etc. This means no word in there.

5.4.2 Define Paragraph Scope Commands

Here is the template for Paragraph Scope Commands. The namespace is `ParagraphCommand` and it is different from Standalone Commands. It is necessary to implement `start`, `end`, and `translate` method. You do not need to implement if the method is not used. If not implemented, KiTTy will just ignore it.

```
1  namespace Typesetting {
2    namespace ParagraphCommand {
3
4      class Command(info_, context_) {
5        public start() {
6          # Procedure when started by '{'
7        }
8        public end() {
9          # Procedure when ended by '}'
10       }
11       public translate(value) {
12         # A paragraph between '{' and '}' is received as value.
13         # value is an array of WordSet or something because this is called after
14         # processing a paragraph. If you want to translat from value to another
15         # word set, do process it here.
16         # This method is called before calling end().
17       }
18     }
19
20   } # namespace Command
21 } # namespace Typesetting
```


Appendix A

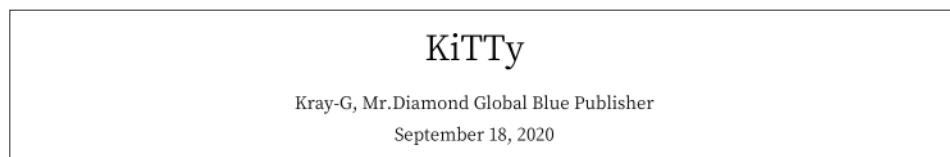
Pre-defined Designs

A.1 Title Design

As a title design, there are 2 types of designs by default.

A.1.1 StandardArticle

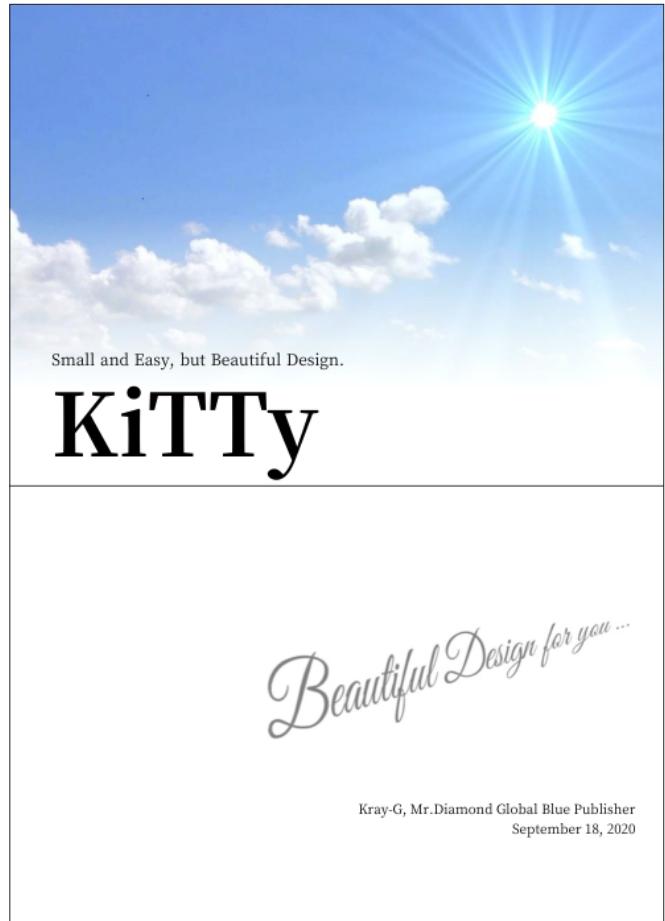
This is a title design for article. The title is shown at the upside of a top page of a document.



A.1.2 StandardBook

This will create an independent cover page. This is the style used in this document. It is available to set a sub title and a background image.

The image on the right is the example of the cover page of this document. The sub title is shown on the top of the title text, and there is underline under the title text. The author and date is shown at the right side under the underline. The other place is covered by the background image. The background image is drawn from the top with the margin 0.

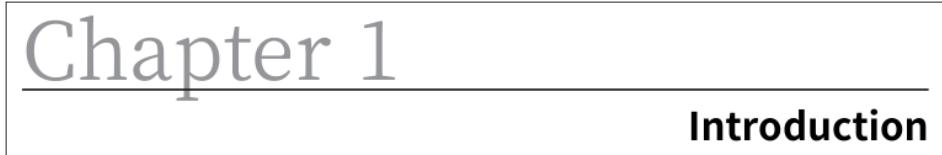


A.2 Chapter Design

As a chapter design, there are 4 types of designs by default.

A.2.1 StandardBook

This is the style used in this document. This has a largish chapter number with an underline, and put a bold type of a heading to the right side under the underline.



A.2.2 BigChapter1

This style has a chapter number which is a little bigger than `StandardBook`, and it changes the location of a number and a heading.



A.2.3 BigChapter2

This style has a chapter number which is a little smaller than `StandardBook`, and it puts a heading which is a little bigger with a regular font on the underline.



A.2.4 BigChapter3

This style has a quite big chapter number and puts a heading with a regular font under the number.



Appendix B

About Style Parameters

B.1 Style Parameters

You can change the style parameters by `<style-info />`, and it is specified by the format of `<style-info name="name" value="value" />`. Here is the default value of the styles for an English document.

Table B.1 Style Parameter List

Name	ArticleA4	BookA4
title.style	"StandardArticle"	"StandardBook"
abstract.title.text	"Abstract"	"Abstract"
chapter.style	-	"StandardBook"
hyphenationRule	"en"	"en"
toc.header	"Contents"	"Contents"
toc.lof	"List Of Figures"	"List Of Figures"
toc.lot	"List Of Tables"	"List Of Tables"
toc.appendix	"Appendix "	"Appendix "
table.label	"Table"	"Table"
image.label	"Fig"	"Fig"
image.fulllabel	"Figure"	"Figure"

The parameters which you can change will be added in the future.

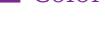
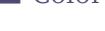
Appendix C

Color Examples

C.1 Color Names and RGB/CMYK Values

You can use the color names below with the `\color` command. This table also shows a value of RGB and CMYK for the color name.

Table C.1 Color Names and RGB/CMYK Values

Color Name	RGB Value	CMYK Value	Color Example
aqua	[0x00, 0xff, 0xff]	[1.000, 0.000, 0.000, 0.000]	 Color
aquamarine1	[0x87, 0xff, 0xd7]	[0.471, 0.000, 0.157, 0.000]	 Color
aquamarine3	[0x5f, 0xd7, 0xaf]	[0.471, 0.000, 0.157, 0.157]	 Color
black	[0x00, 0x00, 0x00]	[0.000, 0.000, 0.000, 1.000]	 Color
blue	[0x00, 0x00, 0xff]	[1.000, 1.000, 0.000, 0.000]	 Color
blue1	[0x00, 0x00, 0xff]	[1.000, 1.000, 0.000, 0.000]	 Color
blue3	[0x00, 0x00, 0xd7]	[0.843, 0.843, 0.000, 0.157]	 Color
blueviolet	[0x5f, 0x00, 0xff]	[0.627, 1.000, 0.000, 0.000]	 Color
cadetblue	[0x5f, 0xaf, 0xaf]	[0.314, 0.000, 0.000, 0.314]	 Color
chartreuse1	[0x87, 0xff, 0x00]	[0.471, 0.000, 1.000, 0.000]	 Color
chartreuse2	[0x87, 0xd7, 0x00]	[0.314, 0.000, 0.843, 0.157]	 Color
chartreuse3	[0x5f, 0xd7, 0x00]	[0.471, 0.000, 0.843, 0.157]	 Color
chartreuse4	[0x5f, 0x87, 0x00]	[0.157, 0.000, 0.529, 0.471]	 Color
cornflowerblue	[0x5f, 0x87, 0xff]	[0.627, 0.471, 0.000, 0.000]	 Color
cornsilk1	[0xff, 0xff, 0xd7]	[0.000, 0.000, 0.157, 0.000]	 Color
cyan1	[0x00, 0xff, 0xff]	[1.000, 0.000, 0.000, 0.000]	 Color
cyan2	[0x00, 0xff, 0xd7]	[1.000, 0.000, 0.157, 0.000]	 Color
cyan3	[0x00, 0xd7, 0xaf]	[0.843, 0.000, 0.157, 0.157]	 Color
darkblue	[0x00, 0x00, 0x87]	[0.529, 0.529, 0.000, 0.471]	 Color
darkcyan	[0x00, 0xaf, 0x87]	[0.686, 0.000, 0.157, 0.314]	 Color
darkgoldenrod	[0xaf, 0x87, 0x00]	[0.000, 0.157, 0.686, 0.314]	 Color
darkgreen	[0x00, 0x5f, 0x00]	[0.373, 0.000, 0.373, 0.627]	 Color

Color Name	RGB Value	CMYK Value	Color Example
darkkhaki	[0xaf, 0xaf, 0x5f]	[0.000, 0.000, 0.314, 0.314]	 Color
darkmagenta	[0x87, 0x00, 0xaf]	[0.157, 0.686, 0.000, 0.314]	 Color
darkolivegreen1	[0xd7, 0xff, 0x87]	[0.157, 0.000, 0.471, 0.000]	 Color
darkolivegreen2	[0xaf, 0xff, 0x5f]	[0.314, 0.000, 0.627, 0.000]	 Color
darkolivegreen3	[0xaf, 0xd7, 0x5f]	[0.157, 0.000, 0.471, 0.157]	 Color
darkorange	[0xff, 0x87, 0x00]	[0.000, 0.471, 1.000, 0.000]	 Color
darkorange3	[0xd7, 0x5f, 0x00]	[0.000, 0.471, 0.843, 0.157]	 Color
darkred	[0x87, 0x00, 0x00]	[0.000, 0.529, 0.529, 0.471]	 Color
darkseagreen	[0x87, 0xaf, 0x87]	[0.157, 0.000, 0.157, 0.314]	 Color
darkseagreen1	[0xd7, 0xff, 0xaf]	[0.157, 0.000, 0.314, 0.000]	 Color
darkseagreen2	[0xaf, 0xff, 0xaf]	[0.314, 0.000, 0.314, 0.000]	 Color
darkseagreen3	[0xaf, 0xd7, 0x87]	[0.157, 0.000, 0.314, 0.157]	 Color
darkseagreen4	[0x5f, 0xaf, 0x5f]	[0.314, 0.000, 0.314, 0.314]	 Color
darkslategray1	[0x87, 0xff, 0xff]	[0.471, 0.000, 0.000, 0.000]	 Color
darkslategray2	[0x5f, 0xff, 0xff]	[0.627, 0.000, 0.000, 0.000]	 Color
darkslategray3	[0x87, 0xd7, 0xd7]	[0.314, 0.000, 0.000, 0.157]	 Color
darkturquoise	[0x00, 0xd7, 0xd7]	[0.843, 0.000, 0.000, 0.157]	 Color
darkviolet	[0xaf, 0x00, 0xd7]	[0.157, 0.843, 0.000, 0.157]	 Color
deeppink1	[0xff, 0x00, 0xaf]	[0.000, 1.000, 0.314, 0.000]	 Color
deeppink2	[0xff, 0x00, 0x5f]	[0.000, 1.000, 0.627, 0.000]	 Color
deeppink3	[0xd7, 0x00, 0x87]	[0.000, 0.843, 0.314, 0.157]	 Color
deeppink4	[0xaf, 0x00, 0x5f]	[0.000, 0.686, 0.314, 0.314]	 Color
deepskyblue1	[0x00, 0xaf, 0xff]	[1.000, 0.314, 0.000, 0.000]	 Color
deepskyblue2	[0x00, 0xaf, 0xd7]	[0.843, 0.157, 0.000, 0.157]	 Color
deepskyblue3	[0x00, 0x87, 0xd7]	[0.843, 0.314, 0.000, 0.157]	 Color
deepskyblue4	[0x00, 0x5f, 0xaf]	[0.686, 0.314, 0.000, 0.314]	 Color
dodgerblue1	[0x00, 0x87, 0xff]	[1.000, 0.471, 0.000, 0.000]	 Color
dodgerblue2	[0x00, 0x5f, 0xff]	[1.000, 0.627, 0.000, 0.000]	 Color
dodgerblue3	[0x00, 0x5f, 0xd7]	[0.843, 0.471, 0.000, 0.157]	 Color
fuchsia	[0xff, 0x00, 0xff]	[0.000, 1.000, 0.000, 0.000]	 Color
gold1	[0xff, 0xd7, 0x00]	[0.000, 0.157, 1.000, 0.000]	 Color

Color Name	RGB Value	CMYK Value	Color Example
gold3	[0xd7, 0xaf, 0x00]	[0.000, 0.157, 0.843, 0.157]	 Color
green	[0x00, 0x80, 0x00]	[0.502, 0.000, 0.502, 0.498]	 Color
green1	[0x00, 0xff, 0x00]	[1.000, 0.000, 1.000, 0.000]	 Color
green3	[0x00, 0xd7, 0x00]	[0.843, 0.000, 0.843, 0.157]	 Color
green4	[0x00, 0x87, 0x00]	[0.529, 0.000, 0.529, 0.471]	 Color
greenyellow	[0xaf, 0xff, 0x00]	[0.314, 0.000, 1.000, 0.000]	 Color
grey	[0x80, 0x80, 0x80]	[0.000, 0.000, 0.000, 0.498]	 Color
grey0	[0x00, 0x00, 0x00]	[0.000, 0.000, 0.000, 1.000]	 Color
grey100	[0xff, 0xff, 0xff]	[0.000, 0.000, 0.000, 0.000]	
grey11	[0x1c, 0x1c, 0x1c]	[0.000, 0.000, 0.000, 0.890]	 Color
grey15	[0x26, 0x26, 0x26]	[0.000, 0.000, 0.000, 0.851]	 Color
grey19	[0x30, 0x30, 0x30]	[0.000, 0.000, 0.000, 0.812]	 Color
grey23	[0x3a, 0x3a, 0x3a]	[0.000, 0.000, 0.000, 0.773]	 Color
grey27	[0x44, 0x44, 0x44]	[0.000, 0.000, 0.000, 0.733]	 Color
grey3	[0x08, 0x08, 0x08]	[0.000, 0.000, 0.000, 0.969]	 Color
grey30	[0x4e, 0x4e, 0x4e]	[0.000, 0.000, 0.000, 0.694]	 Color
grey35	[0x58, 0x58, 0x58]	[0.000, 0.000, 0.000, 0.655]	 Color
grey37	[0x5f, 0x5f, 0x5f]	[0.000, 0.000, 0.000, 0.627]	 Color
grey39	[0x62, 0x62, 0x62]	[0.000, 0.000, 0.000, 0.616]	 Color
grey42	[0x6c, 0x6c, 0x6c]	[0.000, 0.000, 0.000, 0.576]	 Color
grey46	[0x76, 0x76, 0x76]	[0.000, 0.000, 0.000, 0.537]	 Color
grey50	[0x80, 0x80, 0x80]	[0.000, 0.000, 0.000, 0.498]	 Color
grey53	[0x87, 0x87, 0x87]	[0.000, 0.000, 0.000, 0.471]	 Color
grey54	[0x8a, 0x8a, 0x8a]	[0.000, 0.000, 0.000, 0.459]	 Color
grey58	[0x94, 0x94, 0x94]	[0.000, 0.000, 0.000, 0.420]	 Color
grey62	[0x9e, 0x9e, 0x9e]	[0.000, 0.000, 0.000, 0.380]	 Color
grey63	[0xaf, 0x87, 0xaf]	[0.000, 0.157, 0.000, 0.314]	 Color
grey66	[0xa8, 0xa8, 0xa8]	[0.000, 0.000, 0.000, 0.341]	 Color
grey69	[0xaf, 0xaf, 0xaf]	[0.000, 0.000, 0.000, 0.314]	 Color
grey7	[0x12, 0x12, 0x12]	[0.000, 0.000, 0.000, 0.929]	 Color
grey70	[0xb2, 0xb2, 0xb2]	[0.000, 0.000, 0.000, 0.302]	 Color

Color Name	RGB Value	CMYK Value	Color Example
grey74	[0xbc, 0xbc, 0xbc]	[0.000, 0.000, 0.000, 0.263]	█ Color
grey78	[0xc6, 0xc6, 0xc6]	[0.000, 0.000, 0.000, 0.224]	█ Color
grey82	[0xd0, 0xd0, 0xd0]	[0.000, 0.000, 0.000, 0.184]	█ Color
grey84	[0xd7, 0xd7, 0xd7]	[0.000, 0.000, 0.000, 0.157]	█ Color
grey85	[0xda, 0xda, 0xda]	[0.000, 0.000, 0.000, 0.145]	█ Color
grey89	[0xe4, 0xe4, 0xe4]	[0.000, 0.000, 0.000, 0.106]	█ Color
grey93	[0xee, 0xee, 0xee]	[0.000, 0.000, 0.000, 0.067]	█ Color
honeydew2	[0xd7, 0xff, 0xd7]	[0.157, 0.000, 0.157, 0.000]	█ Color
hotpink	[0xff, 0x5f, 0xd7]	[0.000, 0.627, 0.157, 0.000]	█ Color
hotpink2	[0xd7, 0x5f, 0xaf]	[0.000, 0.471, 0.157, 0.157]	█ Color
hotpink3	[0xd7, 0x5f, 0x87]	[0.000, 0.471, 0.314, 0.157]	█ Color
indianred	[0xd7, 0x5f, 0x5f]	[0.000, 0.471, 0.471, 0.157]	█ Color
indianred1	[0xff, 0x5f, 0x87]	[0.000, 0.627, 0.471, 0.000]	█ Color
khaki1	[0xff, 0xff, 0x87]	[0.000, 0.000, 0.471, 0.000]	█ Color
khaki3	[0xd7, 0xd7, 0x5f]	[0.000, 0.000, 0.471, 0.157]	█ Color
lightcoral	[0xff, 0x87, 0x87]	[0.000, 0.471, 0.471, 0.000]	█ Color
lightcyan1	[0xd7, 0xff, 0xff]	[0.157, 0.000, 0.000, 0.000]	█ Color
lightcyan3	[0xaf, 0xd7, 0xd7]	[0.157, 0.000, 0.000, 0.157]	█ Color
lightgoldenrod1	[0xff, 0xff, 0x5f]	[0.000, 0.000, 0.627, 0.000]	█ Color
lightgoldenrod2	[0xff, 0xd7, 0x87]	[0.000, 0.157, 0.471, 0.000]	█ Color
lightgoldenrod3	[0xd7, 0xaf, 0x5f]	[0.000, 0.157, 0.471, 0.157]	█ Color
lightgreen	[0x87, 0xff, 0x87]	[0.471, 0.000, 0.471, 0.000]	█ Color
lightpink1	[0xff, 0xaf, 0xaf]	[0.000, 0.314, 0.314, 0.000]	█ Color
lightpink3	[0xd7, 0x87, 0x87]	[0.000, 0.314, 0.314, 0.157]	█ Color
lightpink4	[0x87, 0x5f, 0x5f]	[0.000, 0.157, 0.157, 0.471]	█ Color
lightsalmon1	[0xff, 0xaf, 0x87]	[0.000, 0.314, 0.471, 0.000]	█ Color
lightsalmon3	[0xd7, 0x87, 0x5f]	[0.000, 0.314, 0.471, 0.157]	█ Color
lightseagreen	[0x00, 0xaf, 0xaf]	[0.686, 0.000, 0.000, 0.314]	█ Color
lightskyblue1	[0xaf, 0xd7, 0xff]	[0.314, 0.157, 0.000, 0.000]	█ Color
lightskyblue3	[0x87, 0xaf, 0xd7]	[0.314, 0.157, 0.000, 0.157]	█ Color
lightslateblue	[0x87, 0x87, 0xff]	[0.471, 0.471, 0.000, 0.000]	█ Color

Color Name	RGB Value	CMYK Value	Color Example
lightslategrey	[0x87, 0x87, 0xaf]	[0.157, 0.157, 0.000, 0.314]	█ Color
lightsteelblue	[0xaf, 0xaf, 0xff]	[0.314, 0.314, 0.000, 0.000]	█ Color
lightsteelblue1	[0xd7, 0xd7, 0xff]	[0.157, 0.157, 0.000, 0.000]	█ Color
lightsteelblue3	[0xaf, 0xaf, 0xd7]	[0.157, 0.157, 0.000, 0.157]	█ Color
lightyellow3	[0xd7, 0xd7, 0xaf]	[0.000, 0.000, 0.157, 0.157]	█ Color
lime	[0x00, 0xff, 0x00]	[1.000, 0.000, 1.000, 0.000]	█ Color
magenta1	[0xff, 0x00, 0xff]	[0.000, 1.000, 0.000, 0.000]	█ Color
magenta2	[0xff, 0x00, 0xd7]	[0.000, 1.000, 0.157, 0.000]	█ Color
magenta3	[0xd7, 0x00, 0xd7]	[0.000, 0.843, 0.000, 0.157]	█ Color
maroon	[0x80, 0x00, 0x00]	[0.000, 0.502, 0.502, 0.498]	█ Color
mediumorchid	[0xaf, 0x5f, 0xd7]	[0.157, 0.471, 0.000, 0.157]	█ Color
mediumorchid1	[0xff, 0x5f, 0xff]	[0.000, 0.627, 0.000, 0.000]	█ Color
mediumorchid3	[0xaf, 0x5f, 0xaf]	[0.000, 0.314, 0.000, 0.314]	█ Color
mediumpurple	[0x87, 0x87, 0xd7]	[0.314, 0.314, 0.000, 0.157]	█ Color
mediumpurple1	[0xaf, 0x87, 0xff]	[0.314, 0.471, 0.000, 0.000]	█ Color
mediumpurple2	[0xaf, 0x87, 0xd7]	[0.157, 0.314, 0.000, 0.157]	█ Color
mediumpurple3	[0x87, 0x5f, 0xd7]	[0.314, 0.471, 0.000, 0.157]	█ Color
mediumpurple4	[0x5f, 0x5f, 0x87]	[0.157, 0.157, 0.000, 0.471]	█ Color
mediumspringgreen	[0x00, 0xff, 0xaf]	[1.000, 0.000, 0.314, 0.000]	█ Color
mediumturquoise	[0x5f, 0xd7, 0xd7]	[0.471, 0.000, 0.000, 0.157]	█ Color
mediumvioletred	[0xaf, 0x00, 0x87]	[0.000, 0.686, 0.157, 0.314]	█ Color
mistyrose1	[0xff, 0xd7, 0xd7]	[0.000, 0.157, 0.157, 0.000]	█ Color
mistyrose3	[0xd7, 0xaf, 0xaf]	[0.000, 0.157, 0.157, 0.157]	█ Color
navajowhite1	[0xff, 0xd7, 0xaf]	[0.000, 0.157, 0.314, 0.000]	█ Color
navajowhite3	[0xaf, 0xaf, 0x87]	[0.000, 0.000, 0.157, 0.314]	█ Color
navy	[0x00, 0x00, 0x80]	[0.502, 0.502, 0.000, 0.498]	█ Color
navyblue	[0x00, 0x00, 0x5f]	[0.373, 0.373, 0.000, 0.627]	█ Color
olive	[0x80, 0x80, 0x00]	[0.000, 0.000, 0.502, 0.498]	█ Color
orange1	[0xff, 0xaf, 0x00]	[0.000, 0.314, 1.000, 0.000]	█ Color
orange3	[0xd7, 0x87, 0x00]	[0.000, 0.314, 0.843, 0.157]	█ Color
orange4	[0x87, 0x5f, 0x00]	[0.000, 0.157, 0.529, 0.471]	█ Color

Color Name	RGB Value	CMYK Value	Color Example
orangered1	[0xff, 0x5f, 0x00]	[0.000, 0.627, 1.000, 0.000]	 Color
orchid	[0xd7, 0x5f, 0xd7]	[0.000, 0.471, 0.000, 0.157]	 Color
orchid1	[0xff, 0x87, 0xff]	[0.000, 0.471, 0.000, 0.000]	 Color
orchid2	[0xff, 0x87, 0xd7]	[0.000, 0.471, 0.157, 0.000]	 Color
palegreen1	[0xaf, 0xff, 0x87]	[0.314, 0.000, 0.471, 0.000]	 Color
palegreen3	[0x87, 0xd7, 0x87]	[0.314, 0.000, 0.314, 0.157]	 Color
paleturquoise1	[0xaf, 0xff, 0xff]	[0.314, 0.000, 0.000, 0.000]	 Color
paleturquoise4	[0x5f, 0x87, 0x87]	[0.157, 0.000, 0.000, 0.471]	 Color
palevioletred1	[0xff, 0x87, 0xaf]	[0.000, 0.471, 0.314, 0.000]	 Color
pink1	[0xff, 0xaf, 0xd7]	[0.000, 0.314, 0.157, 0.000]	 Color
pink3	[0xd7, 0x87, 0xaf]	[0.000, 0.314, 0.157, 0.157]	 Color
plum1	[0xff, 0xaf, 0xff]	[0.000, 0.314, 0.000, 0.000]	 Color
plum2	[0xd7, 0xaf, 0xff]	[0.157, 0.314, 0.000, 0.000]	 Color
plum3	[0xd7, 0x87, 0xd7]	[0.000, 0.314, 0.000, 0.157]	 Color
plum4	[0x87, 0x5f, 0x87]	[0.000, 0.157, 0.000, 0.471]	 Color
purple	[0xaf, 0x00, 0xff]	[0.314, 1.000, 0.000, 0.000]	 Color
purple3	[0x5f, 0x00, 0xd7]	[0.471, 0.843, 0.000, 0.157]	 Color
purple4	[0x5f, 0x00, 0xaf]	[0.314, 0.686, 0.000, 0.314]	 Color
red	[0xff, 0x00, 0x00]	[0.000, 1.000, 1.000, 0.000]	 Color
red1	[0xff, 0x00, 0x00]	[0.000, 1.000, 1.000, 0.000]	 Color
red3	[0xd7, 0x00, 0x00]	[0.000, 0.843, 0.843, 0.157]	 Color
rosybrown	[0xaf, 0x87, 0x87]	[0.000, 0.157, 0.157, 0.314]	 Color
royalblue1	[0x5f, 0x5f, 0xff]	[0.627, 0.627, 0.000, 0.000]	 Color
salmon1	[0xff, 0x87, 0x5f]	[0.000, 0.471, 0.627, 0.000]	 Color
sandybrown	[0xff, 0xaf, 0x5f]	[0.000, 0.314, 0.627, 0.000]	 Color
seagreen1	[0x5f, 0xff, 0xaf]	[0.627, 0.000, 0.314, 0.000]	 Color
seagreen2	[0x5f, 0xff, 0x5f]	[0.627, 0.000, 0.627, 0.000]	 Color
seagreen3	[0x5f, 0xd7, 0x87]	[0.471, 0.000, 0.314, 0.157]	 Color
silver	[0xc0, 0xc0, 0xc0]	[0.000, 0.000, 0.000, 0.247]	 Color
skyblue1	[0x87, 0xd7, 0xff]	[0.471, 0.157, 0.000, 0.000]	 Color
skyblue2	[0x87, 0xaf, 0xff]	[0.471, 0.314, 0.000, 0.000]	 Color

Color Name	RGB Value	CMYK Value	Color Example
skyblue3	[0x5f, 0xaf, 0xd7]	[0.471, 0.157, 0.000, 0.157]	 Color
slateblue1	[0x87, 0x5f, 0xff]	[0.471, 0.627, 0.000, 0.000]	 Color
slateblue3	[0x5f, 0x5f, 0xd7]	[0.471, 0.471, 0.000, 0.157]	 Color
springgreen1	[0x00, 0xff, 0x87]	[1.000, 0.000, 0.471, 0.000]	 Color
springgreen2	[0x00, 0xff, 0x5f]	[1.000, 0.000, 0.627, 0.000]	 Color
springgreen3	[0x00, 0xd7, 0x5f]	[0.843, 0.000, 0.471, 0.157]	 Color
springgreen4	[0x00, 0x87, 0x5f]	[0.529, 0.000, 0.157, 0.471]	 Color
steelblue	[0x5f, 0x87, 0xaf]	[0.314, 0.157, 0.000, 0.314]	 Color
steelblue1	[0x5f, 0xd7, 0xff]	[0.627, 0.157, 0.000, 0.000]	 Color
steelblue3	[0x5f, 0x87, 0xd7]	[0.471, 0.314, 0.000, 0.157]	 Color
tan	[0xd7, 0xaf, 0x87]	[0.000, 0.157, 0.314, 0.157]	 Color
teal	[0x00, 0x80, 0x80]	[0.502, 0.000, 0.000, 0.498]	 Color
thistle1	[0xff, 0xd7, 0xff]	[0.000, 0.157, 0.000, 0.000]	 Color
thistle3	[0xd7, 0xaf, 0xd7]	[0.000, 0.157, 0.000, 0.157]	 Color
turquoise2	[0x00, 0xd7, 0xff]	[1.000, 0.157, 0.000, 0.000]	 Color
turquoise4	[0x00, 0x87, 0x87]	[0.529, 0.000, 0.000, 0.471]	 Color
violet	[0xd7, 0x87, 0xff]	[0.157, 0.471, 0.000, 0.000]	 Color
wheat1	[0xff, 0xff, 0xaf]	[0.000, 0.000, 0.314, 0.000]	 Color
wheat4	[0x87, 0x87, 0x5f]	[0.000, 0.000, 0.157, 0.471]	 Color
white	[0xff, 0xff, 0xff]	[0.000, 0.000, 0.000, 0.000]	
yellow	[0xff, 0xff, 0x00]	[0.000, 0.000, 1.000, 0.000]	 Color
yellow1	[0xff, 0xff, 0x00]	[0.000, 0.000, 1.000, 0.000]	 Color
yellow2	[0xd7, 0xff, 0x00]	[0.157, 0.000, 1.000, 0.000]	 Color
yellow3	[0xd7, 0xd7, 0x00]	[0.000, 0.000, 0.843, 0.157]	 Color
yellow4	[0x87, 0xaf, 0x00]	[0.157, 0.000, 0.686, 0.314]	 Color