

Small and Easy, but Beautiful Design For You

# KiTTY

---

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

# 目次

## 第 1 章 イン트로ダクション

1.1	KiTTy とは	1
1.1.1	Versus L <sup>A</sup> T <sub>E</sub> X	1
1.1.2	Versus Word	1
1.1.3	結論	2
1.2	サポート機能	3

## 第 2 章 さあ始めよう

2.1	インストール	5
2.2	ビルド	5
2.3	hello, world	5

## 第 3 章 機能概要

3.1	基本機能	7
3.1.1	ハイフネーション・ジャスティフィケーション・行分割	7
3.1.2	ウィドウ／オーファン	7
3.1.3	箇条書き	8
3.1.4	数式	9
3.1.5	イメージ	10
3.1.6	グラフ（チャート）	11
3.1.7	テーブル	12
3.1.8	色	12
3.1.9	プログラム・コード	12
3.1.10	相互参照	12
3.1.11	外部リンク	12
3.1.12	脚注	12
3.1.13	合字	12
3.2	日本語用機能	12
3.2.1	日本語禁則処理	12

3.2.2 日本語ルビ . . . . .	12
-----------------------	----

## 第 4 章 コマンド

4.1 パラグラフ処理コマンド . . . . .	13
4.2 スタンドアロン・コマンド . . . . .	13

## 第 5 章 機能拡張

5.1 スタイル . . . . .	15
5.2 禁則処理 . . . . .	15

## List of Figures

3.1 ハイフネーションとジャスティフィケーション . . . . .	7
3.2 イトトンボ . . . . .	11
3.3 F14 Tomcats . . . . .	11

## List of Tables



# 第 1 章

---

## イントロダクション

### 1.1 KiTTY とは

KiTTY は Kinx Tiny Typesetting を意味し、Kinx で実装された簡易組版システムの名称です。Markdown 形式からの簡易トランスレーターを実装しているため、Markdown 形式で書かれたドキュメントを美しく組版することができます。本文書自体も Markdown で記載されているものを自動組版した一つの事例です。

考え方は L<sup>A</sup>T<sub>E</sub>X に近く、テキスト形式で管理している文書ファイルを美しく組版することを目的としています。より具体的には、本システムは L<sup>A</sup>T<sub>E</sub>X を置き換えることを目的とはしていませんが、以下を実現することによって、より個人的な利用シーンの中で、より簡単に利用できるようにすることを目的としています。

- 小さなシステムを維持すること
- それなりに美しく組版できること
- 直接 PDF ファイルを出力できること

KiTTY は小さなシステムながらある程度美しく組版できる機能を持ち、Markdown で書かれた文書から直接 PDF ファイルとして出力することができる組版システムです。

#### 1.1.1 Versus L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X は巨大なシステムです。拡張性にも優れ、多くの人々に支えられた美しい文書を作成するための組版システムです。KiTTY も L<sup>A</sup>T<sub>E</sub>X と同じ目的を持つ組版システムですが、限られた機能しか提供しない代わりに小さなシステムとして提供されます。

L<sup>A</sup>T<sub>E</sub>X の巨大さは、インストールの複雑さにもつながります。T<sub>E</sub>X、L<sup>A</sup>T<sub>E</sub>X では様々な機能を提供するためにディストリビューションそのものが複数存在しています。それにより、ユーザーはまずどのディストリビューションを使うべきかで悩むことになります。KiTTY は、Kinx パッケージに標準で組み込まれており、Kinx をインストールすることですぐに使えるようになります。

ただし、小さく、そして簡単に使える代わりにトレードオフとして限られた機能<sup>1</sup>しか提供されないといった欠点があります。また、組版スピードはそれほど速くはありません。パフォーマンスの改善も 1 つの課題です。

#### 1.1.2 Versus Word

WYSIWYG<sup>2</sup> のワードプロセッサとして代表的な Word ですが、考え方が異なります。WYSIWYG では見たままの形式で編集可能ですが、その見た目情報を一緒に保存するために多くの場合バイナリ形式で保存されます。そのため、中に何が書かれているか知るには一般的に専用の

---

<sup>1</sup>「限られた機能」に関しては、「第 3 章 機能概要」を参照してください。

<sup>2</sup> What You See Is What You Get の頭文字をとったもの。見たままのものを実際に作成出力するという意味。

ソフトウェア（この場合 Word）が必要となります。

KiTTY は L<sup>A</sup>T<sub>E</sub>X 同様、テキストエディタさえあれば内容を知ることができ、編集することも可能です。

テキストで保存されるということは、別のソフトウェアで処理することも簡単であり、Git のようなバージョン管理システム上で差分を確認することも容易です。このことは特に、差分管理をバージョン管理システム上で実現したい場合には必須となる特徴です。

また、文書構造に関しても、Word では直接的にその見た目から「それが構造化されたもの」か「単に見た目としてそう見えているものか」の区別がつきません。例えば、章番号がきちんと章番号として設定され、文章の配置やレイアウトを変更した際に正しく番号を付け直してくれるよう配慮されているかどうか判別しづらいといった欠点もあります。特に、どのような書き方をしたとしても「見た目として正しく見えてしまっている」ということにより、他者の作成したファイルを編集する際に意図せず正しく設定してくれていなかった、といった不運もたびたび見られます。

KiTTY では、文書構造をテキストで表現する関係上、章やセクション、図、表などのリファレンスを常に正しく把握し、適切な番号付け、および相互参照機能を実現することができます。

その代わり、WYSIWYG のようにその場で出力後の体裁（見た目）を確認することができない、といった欠点があります。

### 1.1.3 結論

要約すると、以下のケースにおいて KiTTY は有益でしょう。

- T<sub>E</sub>X のような巨大なシステムではなく、小さな組版システムで簡単に利用したい
- Git のようなバージョン管理システムを使った差分管理をしたい
- 文書構造を常に適切に把握し、相互参照などを正確に実施したい
- サポートされている機能で十分、また多少の遅さは許容範囲である

ちょっとした文書作成のために T<sub>E</sub>X をフルセットで使うには巨大すぎる、と感じている方で、テキストで文書管理をしたい、と考えている方<sup>3</sup>のために本システムを用意しました。特に、Git で差分を含めた文書管理を行いたい場合、WYSIWYG で実現されているワープロソフトでの管理は大変困難ですので、主にそういった利用シーンを想定しています。

---

<sup>3</sup> つまり、私のような方。

## 1.2 サポート機能

KiTTy は、以下の機能をサポートしています。

- 基本機能
  - ハイフネーション・ジャスティフィケーション・行分割
  - ウィドウ／オーファン
  - 箇条書き
  - 数式
  - イメージ
  - グラフ（チャート）
  - テーブル
  - 色
  - プログラム・コード
  - 相互参照
  - 外部リンク
  - 脚注
  - 合字
- 日本語用処理
  - 日本語禁則処理
  - 日本語ルビ

基本機能に加え、日本語特有の処理が組み込まれています。日本語以外の言語への拡張は私自身に知見が乏しいためシステムのソースコードに影響する可能性があります。ソースコードは公開されているので必要に応じて拡張することは可能と考えています。ただし、言語ごとの固有の拡張ポイントを意識していないので、大幅な修正、もしくは機能追加が必要かもしれません。





# 第 2 章

---

## さあ始めよう

### 2.1 インストール

インストールは以下の 2 ステップを実施します。

1. Kinx のインストール
2. KiTTY 追加モジュールのインストール

### 2.2 ビルド

通常、ビルドから実施する必要はありません。既にインストーラが提供されており、手順に従ってインストールを実施することで本システムを利用することができます。あえてビルドから実行したい、といった場合は以下の手順によってビルドを実施できます。

### 2.3 hello, world

次の文書を作成し、helloworld.md ファイルとして保存します。

```
hello, world
```



# 第 3 章

## 機能概要

### 3.1 基本機能

#### 3.1.1 ハイフネーション・ジャスティフィケーション・行分割

Franklin M. Liang のアルゴリズムに基づくハイフネーションをサポートしています。また、ハイフネーションに伴うジャスティフィケーション（両端揃え）機能をサポートしています。

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

Fig 3.1 ハイフネーションとジャスティフィケーション

行分割は Knuth-Plass Line Breaking アルゴリズムを採用しています。本アルゴリズムは、Box、Glue、Penalty によって分割位置をコントロールするアルゴリズムであり、TeX で実装されているアルゴリズムと同様です。これらハイフネーション・アルゴリズムも行分割アルゴリズムも、今のところ組版システムでは最良の方法として知られている方法です。ただし、実装自体は Kinx で改めて行われているため、必ずしも出力結果は TeX での出力と同一にはならない場合があります。

#### 3.1.2 ウィドウ／オーファン

ウィドウ、およびオーファンに対するペナルティ処理を一部ですが実施します。全てのケースで有効ではありませんのでご注意ください。具体的には以下のケースで有効です。

- ・ セクション名がページの最後に取り残されるケースを抑止。
  - この場合、セクション名ごと次のページに追い出されます。
- ・ 複数行パラグラフにおいて、最後の行のみ次のページに送られるケースを抑止。
  - この場合、最後の 2 行分が次のページに追い出されます。
- ・ 複数行パラグラフにおいて、最初の行のみ前のページに残るケースを抑止。
  - この場合、全ての行が次のページに追い出されます。
  - この処理の結果としてセクションが残る場合、セクション自体も次のページに追い出されます。

これらの処理は自動的に行われます。特に文書内に指示を記載する必要はありません。ただし、全てのケースで正しく動作をする訳ではありませんので、うまくレイアウトされない場合は必要に応じて `<pagebreak/>` コマンドを使って改ページを行ってください。

### 3.1.3 箇条書き

箇条書きは記号によるものと番号付きのものが利用できます。次の例は記号による箇条書きの例です。

```
1  *   レベル1
2      *   レベル2
3          *   レベル3
4              *   レベル4
```

これは以下のように整形されます。

- レベル1
  - レベル2
    - レベル3
      - \* レベル4

次の例は番号付き箇条書きの例です。数値ラベルは自動的に補正されます。

```
1  1.   レベル1
2      1.   レベル2
3          1.   レベル3
4              1.   レベル4
5              1.   レベル4
```

これは以下のように整形されます。

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

また、両者を混在させることも可能です。次の例は混在させた場合の箇条書きの例です。

```
1  *   レベル1
2      1.   レベル2
3          *   レベル3
4              1.   レベル4
```

これは以下のように整形されます。

- レベル1
  - (a) レベル2
    - レベル3
      - A. レベル4

### 3.1.4 数式

KiTTY は K<sub>A</sub>T<sub>E</sub>X を内蔵しており、数式を表現することも可能です。数式はスタンドアロン形式とインライン形式の 2 つの表現方法があります。

#### 3.1.4.1 スタンドアロン形式

スタンドアロン形式はコードブロックの形式で記載し、1 行で表現されます。その際、言語として math を指定します。

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

上記のように記載すると、以下のように表現されます。label オプションは付けなくても問題ありませんが、ラベルを付けておくことで (1) のように数式への参照を行うことが可能です。

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

ただし、K<sub>A</sub>T<sub>E</sub>X はラベル機能を持っていないため、ラベル機能は KiTTY によって実現されています。したがって、2 つの式を表現する場合にはラベルを自分自身でコントロールする必要があります。

```
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 ``
```

上記のようにすることで、ラベルの配置位置を数式の上端からそれぞれ 20%、60% の位置に Math2、Math3 のラベルを配置します。

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

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

これによって、`\ref{Math2}` と記載することで (2) への参照を、`\ref{Math3}` と記載することで (3) への参照を作成することが可能となります。

### 3.1.4.2 インライン形式

インラインで数式を扱う場合は \$ で囲みます。例えば、 $E = mc^2$  と記載すると、 $E = mc^2$  と表現されます。また、インテグラルなどの高さのある表記をインラインで記載すると、例えば (1) と同じ  $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$  を記載すると、 $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$  と表現されます。尚、\$ で括った中では Markdown の記法と重なるため、\ や \_ を \ でエスケープする必要があることにご注意ください。

仮に大きな形式で表現したい場合は `\displaystyle` を先頭につけて記載します。例えば、 $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$  と `\displaystyle` を付けて記載すると、 $\int_{-\infty}^{\infty} f(x) dx = \sqrt{\pi}$  と表現されます。ただし、行の高さが揃わないためあまりお勧めするものではありません。

### 3.1.5 イメージ

イメージは Markdown のイメージ形式で記載しますが、alt 部分にオプションを指定し、`![options](path)` の形で記載します。スタンドアロン形式での挿入、インラインでの図の挿入、およびテキストを周りに配置する形でのフローティング形式で挿入することが可能です。

#### 3.1.5.1 スタンドアロン・イメージ

全幅で表示させるには前後を空行の形にし、独立したパラグラフで記載します。

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


上記のように記載すると、以下のように図が挿入されます。scale=0.6 と指定していることにより、版面の横幅の 60% の大きさに補正されて表示されます。その際、縦横の比率は維持されます。



#### 3.1.5.2 インライン・イメージ

インラインの例です。インラインで図を挿入する場合、文中に直接以下のように書きます。

```
ファイルアイコンは ![scale=0.08,offsetY=-0.8](zip256.png) になります。
```

この場合、「ファイルアイコンは  になります。」と表現されます。図の元のサイズに応じて scale と offsetY を適宜調整してください。

### 3.1.5.3 フローティング・イメージ

イメージをフローティングさせるには、オプションに `float=left` または `float=right` を指定します。p.7 に示している [Figure 3.1 ハイフネーションとジャスティフィケーション](#) はフローティング・イメージの一例です。



Fig 3.2 イトトンボ

イメージの下までパラグラフの文章が到達した場合、自動的にテキスト幅が版面の幅に戻り、自然な形でテキストが配置されます。

また、右のイメージのようにパラグラフの右側に配置することも可能です。イメージは上記と同様 Public Domain のものを使わせていただいています。

1 つ注意点としては、パラグラフの先頭とイメージの上端の位置を合わせる必要があります。パラグラフの途

中にフローティングさせることはできません。あるパラグラフを開始する際にフローティングすべきイメージがあれば、パラグラフの左右どちらか指定した場所にイメージを配置します。



Fig 3.3 F14 Tomcats

### 3.1.5.4 オプション

オプションは以下のものを使用できます。

### 3.1.6 グラフ（チャート）

グラフ（チャート）も挿入可能です。グラフもスタンドアロン形式とフローティング形式の両形式をサポートしています。

<sup>4</sup> <https://free-images.com/>

<sup>5</sup> この 70% にはイメージとテキストの間の余白を含みます。

3.1.6.1 スタンドアロン・グラフ

3.1.6.2 フローティング・グラフ

3.1.7 テーブル

3.1.8 色

3.1.9 プログラム・コード

3.1.10 相互参照

3.1.11 外部リンク

3.1.12 脚注

3.1.13 合字

3.2 日本語用機能

3.2.1 日本語禁則処理

3.2.2 日本語ルビ



# 第 4 章

---

## コマンド

### 4.1 パラグラフ処理コマンド

### 4.2 スタンドアロン・コマンド



# 第 5 章

---

## 機能拡張

### 5.1 スタイル

### 5.2 禁則処理