LatexCmds パッケージ

The LatexCmds Contributors

本パッケージは LaTeX に類似するコマンドを SATySFi 上で提供するものである。

1. インストール

本パッケージは Satyrographos 等を用いてインストールしたあと、

```
@require: latexcmds/latexcmds
open LatexCmds in

document (|
  title = {LatexCmds パッケージ};
  author = {The LatexCmds Contributors};
|) '<
>
```

のように使用する。もしどうしても`open LatexCmds in`をしたくない場合 (グローバルな名前空間に LatexCmds 提供のコマンドを公開したくない場合)、コマンド利用時に`\LatexCmds.clearpage`と名前空間で修飾すればよい。

2. ページ及び行分割

ページ分割には`+clearpage;`, `\clearpage;`, `+pagebreak;`, `\pagebreak;`, `+newpage;`, `\newpage;`が利用できる。(どれも同じ効果)

`+`で始まるコマンドは`<`〜`>`で囲まれた中 (縦モード), `\`で始まるコマンドは`{`〜`}`で囲まれた中 (横モード) で使用できる。いずれにせよ、最後の`;`を忘れないでほしい。

SATySFi は LaTeX の行分割コマンド (`\\`) を提供しないため、通常改行を入れたい場合は段落 (`+p { }`) を分けることになる。しかし、どうしても段落中で改行したい時のために`\linebreak;`コマンドを用意している。このように
使い
すぎると
読みにくく

なるため
注意すること。

SATySFI 文書における行は通常、適切な位置で行分割される。また、+screen等で作成した段落中ボックスも、その内部で改ページされる可能性がある。(+p {}コマンドで作成した段落も当然途中で改ページされる可能性がある。)

このような改行や改ページを防ぐために\nobreak +nobreakコマンドを用意している。例えば

```
\nobreak{あああああああああああああああああああああああああああああああああああ
ああああああああああああああああああああああああああああああああああああああああ
ああああああああああああああああああああああああああああああああああああああああ
ああああああああああああああああああああああああああああああああああああああああ
あああああああああああああああああああああああああああああああああああああ}
```

は、

あああ

となり、確かに行分割されないことがわかるだろう。また、同様に

```
+nobreak <
    +p {こんにちは}
    +p {こんにちは}
    ...
>
```

と書くと、大量に挨拶が表示されるが、途中で改ページが入り込まないことが分かるだろう。 挨拶が多すぎて直前のページに入り切らない場合は、 挨拶の直前に改ページが入り挨拶は新しいページから始まる。 しかしその場合でも挨拶が 2 ページ以上にまたがることはない。

紙資源節約の観点から+nobreakの実行結果はここには表示しない。

3. スペース

横モード、縦モード、数式モード中でそれぞれ\hspace(1pt);, \vspace(1pt);, \mspace(1pt);の

ように使用できる。space の代わりに skip としてもよい。セミコロンや括弧は省略できない。

また、現在の文字サイズを基準とした倍率を指定するスペースコマンド\rquad, \mrquadもある。例えば\rquad(1.5);とすれば文字サイズの 1.5 倍の空白が作成される。

さらに、長さを指定せずに使える以下のコマンドも用意されている。

- \quad; … 文字サイズの 1.0 倍の空白
- \hquad; … 文字サイズの 0.5 倍の空白
- \qquad; … 文字サイズの 2.0 倍の空白
- \mquad; … 文字サイズの 1.0 倍の空白 (数式モード)
- \mhquad;… 文字サイズの 0.5 倍の空白 (数式モード)
- \mqquad;… 文字サイズの 2.0 倍の空白 (数式モード)

4. ボックス

\mbox{ ... }は ... の内容が行分割されるのを防ぐ。は ... の内容を出力しないが、それを出力した場合と同じ大きさの空白を作成する。

\makeboxや\frameboxを用いると、幅を固定し中身を折り返すことができる。例えば、

```
\framebox(5cm){あいうえおかきくけこさしすせそたちつてと}
```

とすれば、 あいうえおかきくけこ
さしすせそたちつてと となる。\makeboxと\frameboxの違いは、 枠が付かないか付くかである。またボックスの行中での縦位置も指定することができる。例えば、

```
\framebox?:(Bottom)(5cm){あいうえおかきくけこさしすせそたちつてと}
```

とすれば、 あいうえおかきくけこ
さしすせそたちつてと となる。先の例との違いがわかるだろうか？

横幅を指定したボックスの中に、横モードの文章だけでなく縦モードの段落を入れたくなることもある。この当りを LaTeX でやろうとすると色々アレなのであるが、SATySFi では幸いなことに簡単に実現できる。例えば

```
\parbox(5cm)<
```

```
   +p {あいうえおかきくけこさしすせそたちつてと}
>
```

とすれば、　あいうえおかきくけこさしとなるが、わかりにくいので
すせそたちつてと

```
\fbox{\parbox(5cm)<
   +p {あいうえおかきくけこさしすせそたちつてと}
>}
```

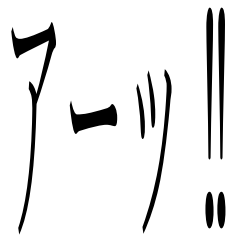
とすれば、 あいうえおかきくけこさし
すせそたちつてと となる。 先程と同様、 Bottom等のキーワードを指定することも可能である。例えば

```
\fbox{\parbox?:(Bottom)(5cm)<
   +p {あいうえおかきくけこさしすせそたちつてと}
>}
```

とすれば、 あいうえおかきくけこさし
すせそたちつてと となる。

ボックスを変形することもできる。例えば

```
\resizebox?:(3cm)?:(3cm){アーッ !!}
```

とすれば、  となる。指定しているのは言うまでもなく横方向の高さと縦方向の高さである。

```
\resizebox?:(3cm){アーッ !!}
```

や

```
\resizebox?*?:(3cm){アーッ !!}
```

とすれば、それぞれア ── ッ !!や ‖のように、横方向のみ、縦方向のみの大きさを指定することもできる。長さではなく倍率を指定する方法もあり、その場合は

```
\scalebox?:(3.0)?:(3.0){Ouch!!}
```

とすればOuch!!となる。3.0 の .0 を忘れると整数とみなされコンパイルエラーになるため注意。

ボックスを本来の位置からずらして描画するには\moveboxを用いる。例えば、

```
\movebox(2cm)(1cm){ここだよ}
```

とすれば　　ここだよ　となり、読みにくい文が完成する。

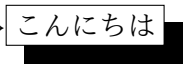
\rotateboxを用いればボックスを回転させることができる。角度は弧度法で指定し

```
\rotatebox(0.5){おむすび}\rotatebox(1.0){ころりん}\rotatebox(1.5){すっとん}\rotatebox(2.0){とん}\rotatebox(2.5){ころころ}\rotatebox(3.0){ころりん}\rotatebox(3.5){すっとん}\rotatebox(4.0){とん}
```

とすればおむすびころりんすっとんとんころころころりんすっとんとんとなる。角度として負の値を指定することもできるが、その場合は\rotatebox(0.0-.0.57)のように引き算の形で指定する必要がある。

SATySFI-LatexCmds はボックスの装飾を行うためのコマンドを提供している。例えば先程も紹介した\fboxの他、以下のものが使用できる。

- \fbox{こんにちは}→ こんにちは
- \doublebox{こんにちは}→ こんにちは
- \ovalbox{こんにちは}→ こんにちは

- \shadowbox{こんにちは}→ 

ただしこれらのボックスは途中で行分割されうることに注意。また、これらのコマンドには追加のパラメータを指定できる。例えば\fboxの場合、

```
\fbox ?:(l) ?:(p) ?:(scolor) ?:(fcolor)
```

となっている。ここで l は線の太さ、p は余白の大きさ、*scolor* は線の色、*fcolor* は塗りの色である。\doublebox \ovalboxも同様である。また、\shadowboxの場合は以下のようになる。

```
\shadowbox ?:(l) ?:(p) ?:(scolor) ?:(fcolor) ?:(drop) ?:(dropcolor)
```

ここで最初の 4 つのパラメータは先述のとおりで、*drop* は影の大きさ (はみ出す距離)、*dropcolor* は影の色である。なお、色の指定にはプリアンブル部で@require: colorが必要である。

実際にこれらのパラメータを組み合わせると、以下のように様々なデザインの箱をつくることができる。

```
+listing {
    * \fbox ?:(2pt) ?:(10pt) ?:(Color.blue) ?:(RGB(0.0, 1.0, 0.5)) {こ
んにちは}
    * \doublebox ?:(2pt) ?:(10pt) ?:(Color.blue) ?:(RGB(0.0, 1.0, 0.5))
{こんにちは}
    * \ovalbox ?:(2pt) ?:(10pt) ?:(Color.blue) ?:(RGB(0.0, 1.0, 0.5))
{こんにちは}
    * \shadowbox ?:(2pt) ?:(10pt) ?:(Color.blue) ?:(RGB(0.0, 1.0, 0.5)) ?:
(10pt) ?:(Color.red) {こんにちは}
}
```

- 
- 

- こんにちは
- こんにちは

行中ではなく縦モードで枠を使いたいこともあるだろう。そんな時は\screen及び\shadowboxを使用できる。例えば

```
+screen<
    +p {学会して　迷走迷走して}
    +p {どしたん話聞こか !?　}
>
```

学会して　迷走迷走して

どしたん話聞こか !?

となる。タイトルをつける場合は

```
+shadowbox?:({「勇気 100\%」の調べで唄おう})<
    +p {学会して　迷走迷走して}
    +p {どしたん話聞こか !?　}
>
```

「勇気 100%」 の調べで唄おう

学会して　迷走迷走して

どしたん話聞こか !?

ちなみに、これらのボックスの途中でページ分割される可能性があるので注意。

行中での装飾枠と同様、これらの段落中の装飾枠もパラメータによって様々にカスタマイズできる。その項目は行中版とほぼ同じだが

```
+screen ?:(it) ?:(va) ?:(l) ?:(p) ?:(scolor) ?:(fcolor)
```

```
+shadowbox ?:(it) ?:(va) ?:(l) ?:(p) ?:(scolor) ?:(fcolor) ?:(drop) ?:
(dropcolor)
```

it はタイトル、*va* はタイトルの水平方向の位置で`HLeft | HCenter | HRight`などが指定できる。*l* は線の太さ、*p* は余白、*scolor* は線の色、*fcolor* は塗りの色である。また、*drop* は影のはみ出し、*dropcolor* は影の色である。

これらを駆使すれば、たとえば以下のようになる。

```
+shadowbox ?:({万年筆はいいぞ}) ?:(HRight) ?:(3pt) ?:(20pt) ?:(RGB(0.3,0.3,0.3)) ?:
(RGB(0.9,0.9,0.9)) ?* ?:(RGB(0.0, 0.5,0.0)) <
    +p {万年筆の良いところは、筆跡を見ただけで一瞬でそれと分かるところ。}
    +p {ボールペンの微妙なところは、筆跡を見ただけで一瞬でそれとバレるところ。}
>
```

万年筆はいいぞ

万年筆の良いところは、筆跡を見ただけで一瞬でそれと分かるところ。

ボールペンの微妙なところは、筆跡を見ただけで一瞬でそれとバレるところ。

`\strut`や`\mstrut`を使うと、インラインテキスト及び数式中で高さが`0.7\baselineskip`, 深さが`0.3\baselineskip`の透明な箱を置くことができる。また、`\rule ?:r w h`を用いると、幅`w`, 高さ`h`で`r`だけ持ち上げた黒塗りの箱を置くことができる。

たとえば`\rule ?:(1mm)(5mm)(2cm);`とすると  となる。

5. レイアウト

文を左、右、中央に配置する場合、`+flushleft`, `+flushright`, `+centering`を使う。ただし、$\mathrm{SATySFi}$ の制約により、以下のように中身が複数行にわたる場合はただしく配置されない場合がある。

```
+centering {
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibu-
lum lobortis nibh nec elit maximus, at mollis orci luctus. Suspendisse
eu tellus diam. Curabitur nisl nibh, tempor non arcu vitae, ullamcor-
per auctor elit.
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum lobortis nibh nec elit maximus, at mollis orci luctus. Suspendisse eu tellus diam. Curabitur nisl nibh, tempor non arcu vitae, ullamcorper auctor elit.

この場合は、若干面倒であるが`+centerings`(`+flushlefts`,`+flushrights`) を用いて手動で行を分割することで正しい結果が得られる。

```
+centerings {
    | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibu-
lum lobortis nibh
    | nec elit maximus, at mollis orci luctus. Suspendisse eu tellus diam.
Curabitur nisl
    | nibh, tempor non arcu vitae, ullamcorper auctor elit.
    |}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum lobortis nibh nec elit maximus, at mollis orci luctus. Suspendisse eu tellus diam. Curabitur nisl nibh, tempor non arcu vitae, ullamcorper auctor elit.

数式中で文字や式を上下に配置する場合、`\overset`, `\underset`が使用できる。たとえば、

```
${i\hbar\frac{\partial}{\partial t}\Psi = \sqbracket{-\underset{\tex-
t!{運動エネルギーを表す}}{\frac{1}{2m}\frac{\partial^2}{\partial x^2}}+V\-
paren{x}} \Psi}
```

$$i\hbar\frac{\partial}{\partial t}\Psi = \left[\underset{\text{運動エネルギーを表す}}{-\frac{1}{2m}\frac{\partial^2}{\partial x^2}} + V(x)\right]\Psi$$

となる。ただし、これらのコマンド中では行分割されないことに注意。また、数式モードではなく横モードで同じ機能を使いたい場合\normal-overset, \normal-undersetを用いる。例えば

```
\normal-overset{そんなところで}{閑話休題}
```

を表示すれば、 そんなところで 閑話休題 となる。ただし、文字にルビを振りたい場合は\normal-oversetではなく、satysfi-rubyなどのパッケージを用いるほうがよいだろう。

\underbrace等を用いれば数式の上下に括弧を書くこともできる。例えば、

```
${\underbrace?:{2N + 1}{-N, -N + 1, \ldots, 0, \ldots, N}}
```

とすれば、$\underbrace{-N, -N+1, ..., 0, ..., N}_{2N+1}$ となる。\underbraceの他にも、\overbrace, \underparen, \overparen, \underangle \overangle, \undersqbracket, \oversqbracketが提供されている。また数式モードではなく横モードで使いたい場合、\normal-overbrace等を使えばよい。

これらのコマンドは簡単なカスタマイズ機能をサポートしている。例えば\overparenの場合

```
\overparen ?:it0 ?:color ?:r
```

となっている。ここで $it0$ は括弧の上に表示するサブテキストであり、$color$ は括弧の色である。また r はサブテキストのフォントサイズと現在のフォントサイズの比率で、デフォルトで 0.5 である。例えば、

```
${\overbrace?:{\bigcirc\ ..\ \bigcirc}?:!(Color.red)?:!(1.0){\under-
brace?*?:!(Color.red){-------}}}
```

とすれば $\overbrace{\bigcirc\ ..\ \bigcirc}^{\text{---------}}$ となる。

6. フォント

\fontsize, +fontsize, \mfontsizeを用いると、それぞれ横モード中、縦モード中、数式モード中で文字サイズを変更することができる。また、\rfontsizeや\mrfontsizeを用いると、現在の文字サイズとの比で文字サイズを指定することができる。

\textcolor, +textcolor, \mtextcolorを用いると、それぞれ横モード中、縦モード中、数式モード中で文字色を変更することができる。また、\colorbox, \mcolorboxを用いるとテキストの背景色を指定できる。例えば、

```
+fontsize(20pt) <
    +textcolor(Color.blue) <
        +p {
            \colorbox?:(5pt)(RGB(0.8,0.8,0.8)){あ \rfontsize(1.2){い \rfont-
size(1.2){う \textcolor(Color.red){え}\colorbox(RGB(0.5,1.0,0.5)){お}}}}
        }
    >
>
```

とすると、

あいうえお

となる。

SATySFI では分数の分母と分子が偏っている場合、例えば $x' = (x - vt)\left(\frac{1}{1 - \left(\frac{v}{c}\right)^2}\right)$ のように括弧内の縦位置がアンバランスになることがある。この場合、\math-middleコマンドを用いて

```
${x' = \paren{x-vt}\paren{\math-middle{\frac{1}{1-\paren{\frac{v}{c}}
^2}}}}
```

とすれば、$x' = (x - vt)\left(\dfrac{1}{1 - \left(\dfrac{v}{c}\right)^2}\right)$ となり、縦位置がバランスされる。

7. シンボル

SATYSF$_{\mathrm{I}}$ で数式を扱う時は`satysfi-azmath`等のパッケージを使うことが推奨されるが、こういった有名パッケージに含まれない特殊文字を使いたくなることがある。SATYSF$_{\mathrm{I}}$-LatexCmds では以下のシンボルをサポートしている。

- `\hbar` …$\hbar$
- `\varphi` …φ
- `\normalphi` …ϕ
- `\sumT` …$\sum$ (displaystyle 版の和記号。`\sumT_a^b`と書くと $\sum_a^b$ となる。一方`\sum_a^b`と書くと $\displaystyle\sum_a^b$ となる。)
- `\quote-sl; hello \quote-sr;` … ‘ hello ’
- `\quote-dl; hello \quote-dr` … “ hello ”