

TEX と K_ET_Cindy 2つの工夫

久保 康幸 *

Two suggestions on TEX and K_ET_Cindy

Yasuyuki Kubo *

Abstract

First, I suggested how to use the mechanism to switch display of questions and answers that I prepared for TEX. Secondly, I proposed a syntax coloring idea to make it comfortable for editing text files to use when creating slides with K_ET_Cindy.

1. 問と答の表示切替

私は、過去の紀要(久保・亀田 [1], [2])において、問と答の表示切替を TEX で実現する仕組み ctoi 環境と ckai 環境の組合せを提案している。

そのときは、tex ソースの修正部分と表示切替の対応を示す例を提示したが、今回は、実際に教材を作る際には、その仕組みを利用して、過去に紹介したのとは異なる出力を得ることができるということ紹介しようと思う。

1.1 他の表示切替の紹介

自分で問と答の表示切替の仕組みを作らなくとも、既にあるのではないか? というのが当然の疑問としてある。[1], [2] で紹介していないものがあつたので、ここで紹介しておく。

emath パッケージには、emathAe.sty が含まれていて、解答の扱いを色々変更したい場合には、別に用意するファイル内に、解答を kaitou 環境内に入力し、コンパイルする tex ソースで emathAe.sty を利用することで実現するようだ [3]。

1.2 自分の提案した表示切替の紹介

環境内に入力したものが切り替わる点では、emath パッケージの kaitou 環境と自分の提案した ckai 環境は同じように思われる。ただ、その環境内に入力した答をどのように処理するかは、kaitou 環境が emathAe.sty パッケージの読み込み時に、オプションで指定するのに対して、ctoi 環境、ckai 環境の場合は、本文中で変更できる。

この仕様を活用すれば、コンパイル用の tex ファイルと別に、本文を編集するファイルを用意することで、一つの pdf ファイルに、「問の答のスペース」、「問と答」、「問のみまとめた」、「答のみまとめた」といったページを入れることができる。

一つの tex ソースで、一部を書き換えることで表示を切り替えた複数の pdf ファイルを得るのでなく、一回のコンパイルで、2つ以上の切り替えた状態のものを、一つの pdf ファイルにまとめることを提案する。

- 修正時に、修正前と修正後の切り替え表示がセットになることで違う版のものが混ざらない。
- 一つの tex ソースの一部を修正して切り替える場合でも、切り替えパターンは2つ以上であり、手元に残るファイルは3つ以上になる。切り替えるパターンが増えるごとにファイル数が増える。これに対し、pdf を一つにすれば、コンパイル用のファイル、内容を入力するファイル、出力した pdf ファイルの3つで済む。

次の節では、私が授業で利用したプリントを例に、「問と答のスペース」、「問と答」の2ページからなる pdf ファイルの作成を示す。

1.3 pdf ファイル作成の実例

コンパイル用の tex ファイル () と、内容を入力した tex ファイル () を用意する。コンパイル用の tex ファイルのドキュメント部分には、次のようなコードを入力しておく。

```
%%% 下の3行で切り替える %%%
\renewenvironment{ctoi}[1][blue]{\color{#1}}
{\relax} % 問の部分の扱いを定義

%\renewenvironment{ckai}[1][magenta]{\color{#1}}
{\relax} % 答の部分の扱いを定義

\renewenvironment{ckai}[1][white]{\color{#1}}
}{\relax} % 答の部分の扱いを定義

% 上の2行をともにコメントアウト(1行目を活か
```

す) なら, 問を詰める.

```
\input{2年181113漸化式_内容.tex}
```

%%% 下の3行で切り替える %%%

```
\renewenvironment{ctoi}[1][blue]{\color{#1}}
{\relax} % 問の部分の扱いを定義
```

```
\renewenvironment{ckai}[1][magenta]{\color{#1}}
{\relax} % 答の部分の扱いを定義
```

```
%\renewenvironment{ckai}[1][white]{\color{#1}}
{\relax} % 答の部分の扱いを定義
```

% 上の2行をともにコメントアウト(1行目を活かす) なら, 問を詰める.

```
\input{2年181113漸化式_内容.tex}
```

このソースの中で `\input` で読み込まれた tex ファイル “2年181113漸化式_内容.tex” に実際に問と答を入力する。

なお, このソースのプリアンブル部分には, 次の2行があり, 3行のうち後ろの2行をコメントアウトすることにより, ckai 環境がコメントアウトのための環境になるようにしてある。

```
\excludeversion{ctoi}
```

```
\excludeversion{ckai}
```

クラス() No. () 氏名()

3 式 $a_1 = 2, a_{n+1} = 2a_n$ ($n = 1, 2, 3, \dots$) で定まる数列の初項から第7項までを並べよ。

解答

□, □, □, □, □, □, □

【階差数列と漸化式】
 数列 $\{a_n\}$ の第 $n+1$ 項と第 n 項との差 $b_n = a_{n+1} - a_n$ を階差という。階差で定まる数列 $\{b_n\}$ を数列 $\{a_n\}$ の階差数列という。
 (1) 数列 $\{a_n\}$ の階差数列 $\{b_n\}$ の一般項が分かっているとき, 数列 $\{a_n\}$ の一般項は, $a_n = a_1 + \sum_{k=1}^{n-1} b_k$ ($n = 2, 3, 4, \dots$) によって求められる。
 (2) 漸化式が $a_{n+1} = a_n + f(n)$ のとき, 数列 $\{b_n\}$ を $b_n = f(n)$ として定めれば, 数列 $\{b_n\}$ は, 数列 $\{a_n\}$ の階差数列となる。

4 数列 $\{a_n\}$ が, 2, 3, 6, 11, 18, 27, 38, 51, ... のとき, 階差数列 $\{b_n\}$ を初項から第7項までを並べよ。

解答

□, □, □, □, □, □, □

5 式 $a_1 = 1, a_{n+1} = a_n + 2n$ をみたす数列 $\{a_n\}$ の一般項 a_n を求めよ。

解答

階差数列 $\{b_n\}$ を $b_n = \square$ とすると,

$$a_n = \square = \square$$

$$= \square$$

図1 解答の非表示(1ページ目)

これにより作成された pdf ファイルの1ページ目(図1)と2ページ目(図2)をそれぞれ示す。

2. K_εT Cindy でのスライド作成時

K_εT Cindy でスライドを作成するとき, 内容を入力するテキストファイル(拡張子 txt) を別に用意する。そのとき, 構文による色付けがされてないと, 文章の内容も命令部分も同じ色では不便を感じた。対処法としては, スライドの内容を tex ソースを編集するソフトで編集することと, テキストファイルを編集するエディタの設定を修正することである。私の場合は, tex ソースの編集に texworks を使用している。

スライド作成時の txt ファイルには, tex コマンドのほかに K_εT Cindy 独自の命令も入力する。texworks の構文色付けに K_εT Cindy の命令も追加できれば良いが, できそうになかった。

txt ファイルの編集には, サクラエディタを使っており, 構文色付けの機能も用意されている。また, tex ソース編集向けの選択メニューまで用意されている。しかし, これは不十分だった。

(1) 登録された単語を色付けで強調する仕組みだったので, 色付けすべき命令の連続した入力に対応していなかった。例えば, 色付けされる単語に “`\sin`”, “`\theta`” の両方が登録されているのに, `\sin θ` を表示するための “`\sin\theta`” が色付けされないことが確認された。

クラス() No. () 氏名(解答例)

3 式 $a_1 = 2, a_{n+1} = 2a_n$ ($n = 1, 2, 3, \dots$) で定まる数列の初項から第7項までを並べよ。 p.22 練習 25 の再利用

解答

2, 4, 8, 16, 32, 64, 128

【階差数列と漸化式】
 数列 $\{a_n\}$ の第 $n+1$ 項と第 n 項との差 $b_n = a_{n+1} - a_n$ を階差という。階差で定まる数列 $\{b_n\}$ を数列 $\{a_n\}$ の階差数列という。
 (1) 数列 $\{a_n\}$ の階差数列 $\{b_n\}$ の一般項が分かっているとき, 数列 $\{a_n\}$ の一般項は, $a_n = a_1 + \sum_{k=1}^{n-1} b_k$ ($n = 2, 3, 4, \dots$) によって求められる。
 (2) 漸化式が $a_{n+1} = a_n + f(n)$ のとき, 数列 $\{b_n\}$ を $b_n = f(n)$ として定めれば, 数列 $\{b_n\}$ は, 数列 $\{a_n\}$ の階差数列となる。

4 数列 $\{a_n\}$ が, 2, 3, 6, 11, 18, 27, 38, 51, ... のとき, 階差数列 $\{b_n\}$ を初項から第7項までを並べよ。

解答

1, 3, 5, 7, 9, 11, 13

5 式 $a_1 = 1, a_{n+1} = a_n + 2n$ をみたす数列 $\{a_n\}$ の一般項 a_n を求めよ。

解答

階差数列 $\{b_n\}$ を $b_n = 2n$ とすると,

$$a_n = a_1 + \sum_{k=1}^{n-1} b_k = 1 + \sum_{k=1}^{n-1} 2k$$

$$= n^2 - n + 1$$

図2 解答を表示(2ページ目)

(2) texworks と同様に K_FT_Cindy 独自の命令が色付けされない。

2.1 実際の対応

まず、一般の T_EX コマンド色付けのためには、サクラエディタに用意されたものを使用する。

次に、(1) の状況に単語の色付けで対応するには、コマンドの結合をした数多くの単語の登録が必要となる。サクラエディタのヘルプ (リンク先 [4]) によれば、最大で 15000 個までのキーワードを登録できるが、コマンドの結合を組み合わせて対応するのは現実的でない。そのため、サクラエディタに用意された、もう一つの方法である、正規表現による色付けを利用した。

正規表現は、次のようにした。

```
/(\)[A-Za-z]+/k
```

これにより、エスケープコード “\” で始まる単語に対応できる。

また、数式モードの切り替えに使う “\$” が色付けされてなかったため、これも正規表現

```
/[\$]/k
```

で対応した。なお、texworks に合わせて茶色を指定した。

T_EX における環境の始まりと終わりを示すコマンド \begin, \end は、後ろに必ず “{” に続ける環境名を伴うので、次のような正規表現にした。

```
/(\begin|begin|end)(?=[\{])/k
```

```
/(\end|begin|end)(?=[\{])/k
```

正規表現による色付けは、100 個までのキーワードにしかな対応していないため、登録単語による色付けで対応できないものに限った。

2.2 スライド作成のためのリスト

ここで、(2) の状況に対応するため、K_FT_Cindy によるスライド作成時の命令について、順に紹介する。

main::文字列 //(章タイトルの指定)

強調キーワードへ “main:” を登録することで対応。

new::文字列 //(節タイトルの指定)

強調キーワードへ “new:” を登録することで対応。

new::[数値]::文字列 (節タイトルに行下げを指定)

上とは別に、/(new::|begin|end)(?=[\w\{])/k という正規表現で対応する必要があった。

%repeat=数値 //(new 含むスライドの枚数)

強調キーワードへ “% repeat” を登録することで対応。

\slidepage //(スライドにページ番号を表示)

/(\slidepage|begin|end)(?=[\w\{])/k という正規表現で対応。

enumerate //(enumerate 環境の始まり)

/(enumerate|begin|end)(?=[\w\{])/k という正規表現で対応。

itemize //(itemize 環境の始まり)

正規表現/(itemize|begin|end)(?=[\w\{])/k で対応。

layer::[数値]{[数値]} //(layer 環境の始まり)

正規表現/(layer::|begin|end)(?=[\{])/k で対応。

end //(上の 3 環境の終わり)

強調キーワードへ “end/” を登録することで対応。

item::文 //(item を追加)

正規表現/(item::|begin|end)(?=[\w\{])/k で対応。

putnote::方向と位置など //(layer 環境に図などを入れる)

強調キーワードへ “putnote:” を登録することで対応。

\setthin{濃さ} //(文字の濃さ)

T_EX コマンドと同じ扱いとし、特に何もしなかった。

他にも、Setslidebody, Setslidehyper, Setslide-main, Setslidepage などのスライドの設定に関するコマンドがあるが、これらは、後ろに “(” を伴うので、強調キーワードへの単語登録で対応できた。

なお、2018 年 11 月 12 日現在の K_FT_Cindy には、2018 年 7 月 29 日版の houtouseslidej.txt が含まれており、それを参考に必要なコマンドと使用方法を確認した。また、スライドの内容を入力するテキストファイルの全ての行の最後には、“//” をつけることになっている。これがサクラエディタの単語の区切りの認識に影響を与える。また、後ろに “{” に続ける可能性のあるものも正規表現で対応しなければならなかった。K_FT_Cindy の開発が進み、スライダの内容を入力するテキストファイルの行末に “//” をつけてもつけなくてもよくなったことが、2018 年 11 月 27 日にメーリングリスト KeTpic.com で知らされた。ここで紹介した正規表現キーワードによる対応のいくつかは、次第に不要になると思われる。

2.3 コメントアウトについて

K_FT_Cindy によるスライド作成で、%repeat により繰り返し枚数を指定したページでは、“%[:]:文章” により文章を表示したいページを指定する。(ページ指定の構文は HoutouseslideJ.txt を参照。)

そのため、スライドの内容を編集するテキストファ

イルは、先頭に % をつける行が多く、texworks のような、% によるコメントアウト行への色付けでは、構文の色付けが意味をなさない。

スライドの内容を編集するテキストファイルでは、コメントアウトは、2つ以上の % で指定することになっており、サクラエディタでは、コメントスタイルの設定画面で、コメントアウトの色付けは“%%”を指定した。

この事情からも、スライド作成時のテキストファイルは、texworks よりも、今回のように設定したサクラエディタで編集することにした。なお、 $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$ が配布されている Dropbox からは、関連ソフトとして Terapad というエディタが入手できるが、Terapad には、構文による色付けの機能はないようである。[6]

2.4 他の PC との共有

ここで紹介したサクラエディタの色付け設定（強調キーワード、正規表現キーワード）は、インポート、エクスポートにより他の PC にインストールしたサクラエディタに反映できる。

参考文献および参考 URL

- [1] 久保康幸・亀田真澄： $\text{T}_{\text{E}}\text{X}$ による問と答の表示切替, 弓削商船高等専門学校紀要, 第 38 号, pp.87–89, (2016).
- [2] 久保康幸・亀田真澄： $\text{T}_{\text{E}}\text{X}$ による問と答の表示切替 2, 弓削商船高等専門学校紀要, 第 39 号, pp.29–33, (2017).
- [3] sampleae.pdf(emathAe.sty ver.0.36):
ダウンロードサイト (2018/11/28 確認):
<http://emath.s40.xrea.com/allinone.htm>.
- [4] サクラエディタ ヘルプ最終更新日 2013/03/31
: <http://sakura-editor.sourceforge.net/htmlhelp/HLP000086.html>.
- [5] 「Ketpic.com」(<http://ketpic.com/>)
- [6] TeraPad で指定した箇所の文字色を変更したい:
(サイト確認 2018/11/28):
<https://bzizi.exblog.jp/17444266/>.