

TEXによる問と答の表示切替2

久保 康幸 *・亀田 真澄 **

2nd Step of Indication Switching of the Question and the Answer in TEX Materials

Yasuyuki Kubo *, Masumi Kameda **

Abstract

Last year I introduced a policy on the function that change the indication of a problem and the answer. In this report, I introduce concretely a method to realize it.

1. はじめに

前回 [1] において示された、問と答の表示切替について、`comment.sty` を利用してコメントアウト、`color` を利用して解答スペースを残した問のみの印刷などの方向性を示した。目的とする機能を再び上げると、次の通りである。

試験などの印刷教材を利用するとき、印刷されるべき内容は、次のようなものではないだろうか。

- タイトル（表題）や名前欄
- 設問番号
- 設問の問題文
- 答
- 補足説明または説明付きの答など

このうち、タイトルは常に表示し、印刷すべきものである。設問番号も必ず印刷する。模範解答を別に印刷するときには問題文は必要ない。答は、試験の実施用は印刷しないが、解答スペースは必要である。説明のない答（略解）の場合と説明付きの答の場合に応じて印刷状態を変更したい。また、配布プリントでは、答の解答スペースを詰めて紙を節約したい。もし、試験が問題用紙と解答用紙を分けて印刷するならば解答用紙は、常に印刷する枠と、試験実施時には印刷しない答を選んで印刷したい。

今回は、これまでに試みた方法を紹介する。

2. 共通の考え方

解答スペースを残して答を非表示のように印刷するには、解答部分を白色に指定すればよい。色の指定には、`color` パッケージを利用する。

[1] で示した例では、解答スペースを残した印刷をするのに、`\newcommand` を利用した。ここで紹介する方法は、解答スペースを詰める印刷をする場合に、環境を定義することにしたので、それとの整合性から、`\renewenvironment` による環境の再定義で色を指定する環境を定義した。そのとき、解答部分に入る前の文字の色を見ることなく、解答部分の環境を抜ければ、自動的に解答部分に入る前の色に戻る。

解答スペースを詰める方法に合わせて、解答部分の環境を定義している。あとで紹介する事例では、表示切替をしない部分と区別するため、問の入力部分 (`ctoi` 環境) での色指定を青 (`blue`)、答の入力部分 (`ckai` 環境) での色指定をピンク色 (`magenta`) とした。Studyaid D.B. では、ピンク色に指定すると答と認識される。それにならって見た。今回紹介するのは、`ctoi` 環境と `ckai` 環境の2つである。2つ用意できれば、3つ目は同様なので省略する。

3. `comment.sty` の改変による方法

`comment` パッケージを使い、`\begin{comment}` と `\end{comment}` とではさむことにより、複数の行を一括してコメントアウトすることができる。

問と答という複数の環境を用意するため、まず考えたのは、`comment.sty` を利用して新しい環境を定義することである。まず、`ckai` 環境を定義するため、[5] に紹介された例を参考に、

```
\newenvironment{ckai}{\begin{comment}}
{\end{comment}}
```

としてみたが、ダメだった。`\begin{ckai}` で `ckai` 環境に入った後、まず `\begin{comment}` を実行することになるから、`\end{comment}` が出現するまで `tex` のコードをスキップするのかもしれないが、状況は、それほど単純ではなかった。

次に考えたのは、名前を変えて利用することであるが、当然、`comment.sty` のファイル名を変えただけではダメである。`comment.sty` がテキストファイルであるから、`comment.sty` をコピーして、`ctoi` 環境のため“`comment`”を“`ctoi`”へ置換したファイル `ctoi.sty` と、`ckai` 環境のため“`comment`”を“`ctoi`”へ置換したファイル `ckai.sty` の2つを用意した。

プリアンプルに、次の4行を挿入する。(行を区別するため行頭に番号を付けた)

```
1:\usepackage{ctoi}
2:\usepackage{ckai}
3:\renewenvironment{ctoi}[1][blue]{\color{#1}}{\relax}
4:\renewenvironment{ckai}[1][magenta]{\color{#1}}{\relax}
```

3行目により、`\begin{ctoi}` と `\end{ctoi}` ではさまれた行 (`ctoi` 環境) が青色に指定される。4行目により、`ckai` 環境がピンク色に指定される。解答スペースを残して答を非表示に見せるには、4行目の `magenta` を `white` (白色) にすればよい。

3行目と4行目を % によりコメントアウトすれば、それぞれ1行目と2行目が有効になって、`ctoi` 環境と `ckai` 環境に入力された行はコメントアウトされる。つまり、問・答が非表示になってスペースが詰められる。

1行目と2行目により、`ctoi` 環境と `ckai` 環境が、それぞれ導入されているので、3行目と4行目での `ctoi` 環境と `ckai` 環境の定義には、再定義を意味する `\renewenvironment` を使っている。

4. `excludecomment` による方法

`comment` パッケージを使えば、`excludecomment` と `includecomment` という2つのコマンドが利用できる。[4]の「パッケージを直接書き換える方法」にも注意があるように、また、仕組みをよく分からずに `sty` ファイルのの中に書かれた単語の置換をするより安全だと思われる。

プリアンプルに挿入するのは、次の5行になる。

```
1:\usepackage{comment}
2:\excludecomment{ctoi}
```

```
3:\excludecomment{ckai}
4:\renewenvironment{ctoi}[1][blue]{\color{#1}}{\relax}
5:\renewenvironment{ckai}[1][magenta]{\color{#1}}{\relax}
```

1行目で `comment` パッケージを読み込み、それによって使えるコマンド `excludecomment` により、2行目と3行目で定義された `ctoi` 環境と `ckai` 環境は、その環境内に入力された行をコメントアウトするものである。さらに、4行目と5行目で、`ctoi` 環境と `ckai` 環境を再定義して、色を指定する環境に直している。4行目と5行目を % によりコメントアウトすれば、それぞれ2行目と3行目が有効になって、`ctoi` 環境と `ckai` 環境に入力された行はコメントアウトされる。

ここでも、4行目と5行目での `ctoi` 環境と `ckai` 環境の再定義には、`\renewenvironment` を使った。

コメントアウトの指定と色の指定の順番を逆にしてもよいが、単に2,3行目と4,5行目を入れ換えるのではなく、次のようにする。

```
1:\usepackage{comment}
2:\newenvironment{ctoi}[1][blue]{\color{#1}}{\relax}
3:\newenvironment{ckai}[1][magenta]{\color{#1}}{\relax}
4:\excludecomment{ctoi}
5:\excludecomment{ckai}
```

2,3行目までは、まだ `ctoi` 環境と `ckai` 環境が定義されていないので、`\renewenvironment` でなく、`\newenvironment` により環境と定義する。4,5行目で `\excludecomment` により `ctoi` 環境と `ckai` 環境を定義するときは、再定義かどうかは区別しないようである。

5. `excludeversion` による方法

`version.sty` および、`versions.sty` は、同じコマンド `excludeversion` が使え、`comment.sty` と、ほぼ同じ機能を実現するパッケージである。`versions.sty` の説明書(pdfファイル)には、他の2つのパッケージとの差が書かれている。私の使用目的には、3つに機能の差はないようである。

`version.sty` を使う場合、プリアンプルに挿入するのは、次の5行になる。

```
1:\usepackage{version}
2:\excludecomment{ctoi}
3:\excludecomment{ckai}
```

T_EX による問と答の表示切替 2

```
4:\renewenvironment{ctoi}[1][blue]{\color{#1}}{\relax}
```

```
5:\renewenvironment{ckai}[1][magenta]{\color{#1}}{\relax}
```

パッケージ名が `version` , 利用するコマンド名が `excludeversion` になっただけで、`comment.sty` のときと同様にする。

6. 最後に : 実例と課題

問と答についてのみ表示切替を用意した場合、問を印刷しないのに問のスペースを残すことは考えにくいので、想定されるのは、次の4通りではないだろうか。

- 問と答の両方を印刷
- 答を印刷しないが答のスペースを残す
- 答を印刷せず答のスペースを詰める
- 問を印刷せず問のスペースを詰める

もし解説も表示切替するなら場合分けはさらに多くなるが考え方は同じである。この節では、上の4パターンの実例を紹介する。また、3つの方法で比較できるように、問と答、および氏名欄は、別のファイルに入力し、`\input` コマンドで挿入した。実例は、`excludecomment` による方法で紹介する。

6.1 実例

氏名欄は、次の3行を入力した。

```
1:\hspace*{33mm}{\large No. ( ) 氏名
(\hfill \begin{ckai} 見本
2:\end{ckai}
3:\hfill) }% 氏名
```

`\begin{ckai}` は前後に文字があってもよいが、`\end{ckai}` は単独の行にしないとエラーが出るので、このようにした。`\hfill` の利用により、`ckai` 環境の指定で、「見本」が見えたり、消えたりしても氏名欄を示す () の幅は変化しない。

このことは、図1と図2を比較して確認できる。また、氏名欄だけを見れば、図2が答のコメントアウトか、色指定で見えなくしたのか分からないが、図2には、ピンク色が無いだけでなく、問2が問1に近いことから、コメントアウトにより答を印刷せず答のスペースを詰めていることが分かる。

さらに、答を印刷しないが答のスペースを残すには、初めに述べたように、答部分を白色に指定すればよい。図3のようになる。

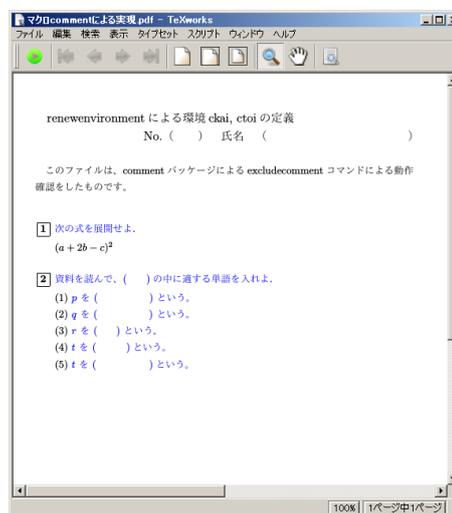


図2 答をコメントアウトした場合

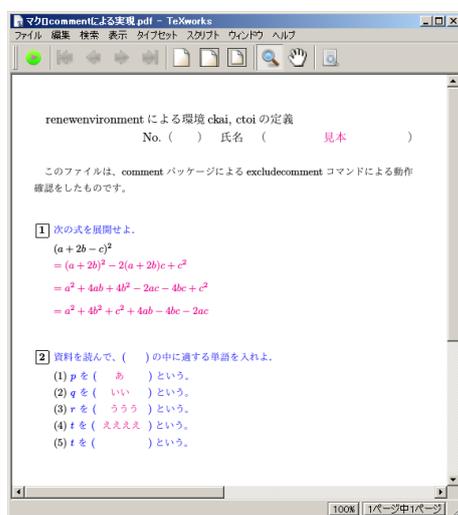


図1 問と答の両方を表示した場合

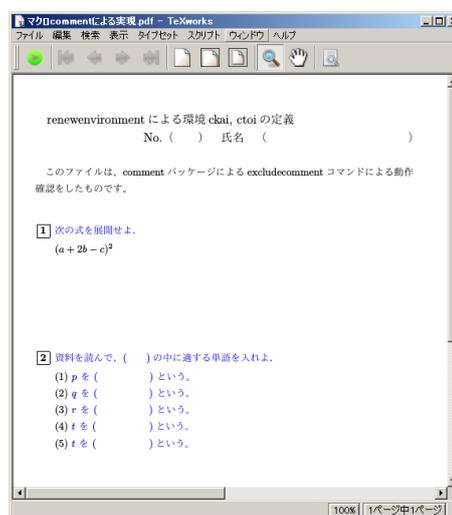


図3 答を印刷しないがスペースは残した場合

最後に答のみ印刷する場合 (図 4) を紹介する。図 1 では青く指定されていた問が表示されず、その場所に小問などが詰めて配置されていることから、図 4 が、問をコメントアウトしたものであることが分かる。図 1~図 3 と違って、図 4 では、問 2 の (1), (2) と (3)~(5) の配置に違いが確認できる。

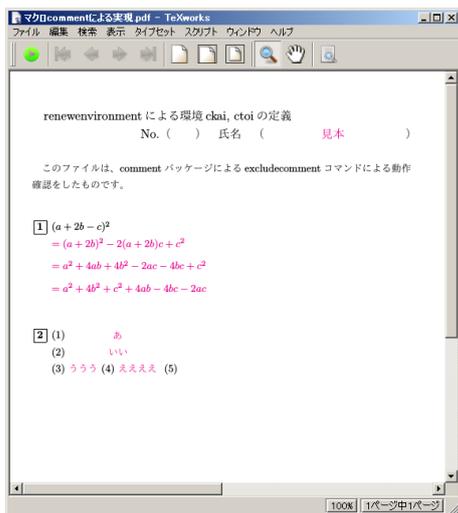


図 4 問をコメントアウトした場合

まず、比較的単純な問 2(3) の入力を紹介する。

```
01:(3) \begin{ctoi}
02:\( r \) を ( \hspace*{.8mm}
03:\end{ctoi}
04:\begin{ckai}
05:うろう
06:\end{ckai}
07:\begin{ctoi}
08:\hspace*{.8mm} ) という。 \par
09:\end{ctoi}
```

ここから先は、入力ソースの行数が多い例が出るので、行番号は 2 けたで示した。答の部分 (04~06 行目) を ckai 環境に入れ、その前後を ctoi 環境に入れた問の文章ではさんだけである。前後の小問の () の幅に合わせるため、 $\hspace*{.8mm}$ を入れてあるが、コメントアウトした図 2 では、(3) のみ、答の単語の幅だけ狭くなっている。

このように、問と答の行をそれぞれ、ctoi 環境と ckai 環境に入力しただけでは、問 2 のような、空欄を埋める設問が美しくない。

問 2(1), (2) では、答のコメントアウト時に () の幅が変化しないよう ketlayer を利用してみた。問 2(1), (2) の入力は次のようになる。

なお、入力する行数は増えるが、 \begin{layer} の前と、 \end{layer} の後ろに空の行を入れるのが基本である。

```
01:(1) \begin{ctoi}
02:\( p \) を ( \hspace*{5.3zw} ) という。
03:\end{ctoi}
04:
05:\begin{ckai}
06:
07:\begin{layer}{30}{0}
08:\putnotec{22}{-1}{あ}
09:\end{layer}
10:
11:\end{ckai}
12:(2) \begin{ctoi}
13:\( q \) を ( \hspace*{5.3zw} ) という。
14:\end{ctoi}
15:
16:\begin{layer}{30}{0}
17:\begin{ckai}
18:\putnotec{21}{-1}{いい}
19:\end{ckai}
20:\end{layer}
```

小問 (1) (01~11 行目) は、ckai 環境の中に ketlayer による環境 (layer 環境) を入れた。答のコメントアウト時には、layer 環境ごとコメントアウトされる。

小問 (2) (12~20 行目) は、layer 環境の中に ckai 環境を入れた。

こうして、ketlayer の利用で、図 2 のように答をコメントアウトしても () の幅は変化しないようにできる。なお、08 行目では、ketlayer の命令により座標 (22, -1) を中心として文字列“あ”を配置しているが、18 行目では、座標 (21, -1) を中心として文字列“いい”を配置している。文字列の違いではなく、ckai 環境と layer 環境のどちらが内側にあるかによって差があることが分かった。

次に、答をコメントアウトしても () の幅は変化しないで、問をコメントアウトした場合 (図 4) に、(3)~(5) のように答を詰めて 1 行に収めることを考えた。問をコメントアウトしたときに改行の命令もコメントアウトし、答の位置 (特に横の位置) が変化するので、layer 環境によらず () の幅が変わらないようにしたい。

$\makebox[幅][c]{文字列}$ により指定した幅の箱を用意して、その中央に文字列を配置できるが、このコマンドは、コメントアウトの環境で使うとエ

ラーが出るようだ。その対策として、答の文字列の長さの半分 (w) を戻り、答を配置してから、また、長さ w を戻すことで、答を入力しなかった場合と同じ位置から、後の文章が始まるようにした。文字列の長さを計測して利用するには、`\settowidth` コマンドを使用した。小問 (3)~(5) は次のように入力した。

```
01:(3) \begin{ctoi}\( r \) を ( \hspace*
    {.8mm}
02:\end{ctoi}
03:\begin{ckai}うろう
04:\end{ckai}
05:\begin{ctoi}\hspace*{.8mm} ) という。
    \par
06:\end{ctoi}
07:(4) \begin{ctoi}\( t \) を (
08:\end{ctoi}
09:\begin{ckai}\hspace*{2zw}
10:\settowidth{\kailen}{ \scalebox{.5}{
    ええええ}}
11:\hspace*{-\kailen}ええええ\hspace*{-
    \kailen}
12:\end{ckai}
13:\hspace*{2zw}\begin{ctoi} ) という。
    \par
14:\end{ctoi}
15:(5) \begin{ctoi}\( t \) を (
16:\end{ctoi}
17:\hspace*{2zw}\begin{ctoi}\hspace*
    {2zw} ) という。
18:\end{ctoi}
```

環境に入る`\begin{ }`は、行の途中にあってもかまわないので、01, 03, 05, 09, 13, 15, 17 行目のような入力をして行数を減らしたが、かえって分かりにくいかも知れない。

`\settowidth` コマンドを使用したのは、10 行目である。プリアンブルで新しく定義してある長さの単位 `\kailen` に文字列“ええええ”の幅の半分の長さを定める。`\scalebox` を使って、文字を横方向に半分にして長さを測っている。11 行目で、“ええええ”の前後を `\hspace*{-\kailen}` ではさんだ。10, 11 行目を次のようにしても同じはずである。

```
10:\settowidth{\kailen}{ ええええ}
11:\hspace*{-.5\kailen}ええええ\hspace*
    {-.5\kailen}
```

これは、文字列“ええええ”の幅をそのまま取得し、11 行目で、`\hspace*{-\kailen}` の代わりに

`\hspace*{-.5\kailen}` として、`\kailen` の 0.5 倍に戻している。今回は、上で紹介したようにしてみた。このようにして、() に入れる文字列の幅により () の幅が変わることはない。

次に、(3)~(5) を行ごとに分けたり、1 行にまとめたりする切替のため、05, 13 行目の最後に段落の区切り (`\par`) を入れた。これらは `ctoi` 環境の中にあるので、問をコメントアウトすれば、(3)~(5) の答が段落を区切らず 1 行に入る。

6.2 課題

(1) 今回、目的に応じて環境を定義した行から選んで、`%` でコメントアウトにより表示切替を実現している。マクロパッケージ化し、オプションによる切替といった、もっと簡潔で分かりやすい方法で実現したいと思っている。

今回は、`excludecommand` を利用した場合で例を示した。そのとき、

(2) 環境の終了を告げる `\end` コマンドを独立した行にする必要がある。正確には、`\end` コマンドの後ろに半角空白のみ許容する。そのため、`tex` ソースの行数が増える。また、環境の中に入れた文字列の前後に半角スペースが加わる。

(3) `mekabox` とのバッティングにより、上で示したような入力 (小問 (4)) をした。

(2), (3) について、`version.sty` や `versions.sty` での検証が不十分である。(2), (3) が解消しても、小問 (4) で示した文字列の半分の長さを戻す方法が `makebox` を使うより有効なら、その作業をコマンドにして入力作業を単純化することも考えられる。

参考文献および参考 URL

- [1] 久保康幸・亀田真澄：T_EX 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, (2015).
- [2] CAST_EX 応用研究会：K_ET_EX でらくらく T_EX グラフ，イーテキスト研究所，(2011).
- [3] 奥村晴彦ほか：[改訂 第 6 版] L^AT_EX₂ ϵ 美文書作成入門，技術評論社，(2013).
- [4] 「T_EX Wiki パッケージ」(<https://texwiki.texjp.org/?%E3%83%91%E3%83%83%E3%82%B1%E3%83%BC%E3%82%B8>)
- [5] 「目的別 L^AT_EX ガイド [一般] マクロ:新しいコマンド/環境」(<http://utsukemononi.gozaru.jp/latex/datg022.html>)
- [6] 「Ketpic.com」(<http://ketpic.com/>)