

answers.sty (Ver.2.10) に関するメモ

大石 勝*

概要

answers.sty は、試験問題の解答を巻末に書き出すためのスタイルファイルである。これは、別ファイルに書き出した後、それを取り込むことにより実現している。以下では、answers.sty の使用方法などを解説する。なお、本書の作成に当たっては、answers.dtx を \LaTeX で処理してできる answers.dvi を参考にした。

目次

1	インストール	1
2	使用方法	2
2.1	<code>\Newassociation</code>	2
2.2	<code>\Opensolutionfile</code> , <code>\Closesolutionfile</code> , <code>\Readsolutionfile</code>	2
2.3	<code>\Writetofile</code> と Filesave 環境	3
3	カスタマイズ	3
3.1	<code>yyy</code> 環境の再定義	3
3.2	<code>\Openzzzhook</code> , <code>\Closezzzhook</code> の再定義	4
3.3	外部ファイル名の変更	4

1 インストール

answers package は、CTAN から answers.ins, answers.dtx を入手すると良い。answers.sty も配布されているようであるが、MS-DOS プロンプトまたはコマンドプロンプトから

```
latex answers.ins
```

とすれば、(サンプルファイルとともに) answers.sty を取り出すことができる。また、マニュアルを取り出すには

```
latex answers.dtx
```

とすれば良い。answers.sty をはじめ全てのファイルは、`C:\%...%texmf%tex%latex%answers` などの適切な場所へインストールしておくが良い。なお、answers.sty は、内部で verbatim.sty を呼び出しているため、これもあわせてインストールしておかなければならないが、tools package に含まれているので改めてインストールする必要は少ないと思う。

*ohishi@grn.mmtr.or.jp

2 使用方法

answers.sty を使用するには、プリアンブルに

```
¥usepackage{answers}
```

と指定するだけで良い。package option としては nosolutionfiles がある。これを指定すると、外部ファイルに書き出さずにその場 (*xxx* 環境¹のあるところ) に書き出すことになる。

2.1 ¥Newassociation

answers.sty を読み込むだけでは何ら環境は定義されていない²。したがって、プリアンブルに次の宣言をしなければならない。

```
¥Newassociation{xxx}{yyy}{zzz}
```

これによって、*xxx* 環境、*yyy* 環境が定義される。ここで、*xxx* はソースファイル (main の TeX file) で用いる環境名、*yyy* は外部ファイルに書き出される内容 (*xxx* 環境内が外部ファイルに書き出される) の周りに追加される環境名、*zzz* は外部ファイル名のハンドル (外部ファイル名とは限らない。cf. ¥Opensolution コマンドのオプション引数) である。これらの *xxx*, *yyy*, *zzz* は自分で好きな名前を付ければ良い。ただし、数字、句読点、半角スペースなどを含んではいけない。

xxx 環境は、

```
¥Opensolutionfile{zzz}
...
¥Closesolutionfile{zzz}
```

の中に書くことになる (次節参照)。このコマンドが複数回現れると、(オプションでファイル名をかえなければ) 最後のものしか有効にならないようである。

2.2 ¥Opensolutionfile, ¥Closesolutionfile, ¥Readsolutionfile

¥Opensolutionfile, ¥Closesolutionfile の各コマンドは、document 環境内に記述する。いずれも 1 つの引数を取り、外部ファイルのハンドル名 *zzz* を指定する。¥Opensolutionfile, ¥Closesolutionfile の各コマンドは文字通り外部ファイルを開く、閉じるに対応している。

¥Readsolutionfile コマンドは、外部ファイルを読み込むコマンドで、¥Closesolutionfile コマンドよりも後で指定する。

```
¥Readsolutionfile{zzz}
```

とすれば、ハンドル名 *zzz* のファイル (デフォルトでは *zzz.tex*) を読み込むことができる。¥input や ¥include でももちろん読み込むことができる。

¥Opensolutionfile コマンドはオプション引数として外部ファイル名を指定することができる。オプション引数を省略した場合は、ハンドル名が外部ファイル名となる。例えば、

```
¥Opensolutionfile{zzz}[www]
```

¹次節参照。

²実は、Filesave 環境は定義されている。これについては後述する。

とすると、`zzz.tex` ではなく `www.tex` が (存在しなければ) 作成される。`www` にはファイル名として (通常) 許される文字しか使用してはならない。

`\Opensolutionfile` コマンドを異なるオプション引数で複数指定すれば、複数の外部ファイルを扱うことが可能となる³。この場合、`\Closesolutionfile` コマンドも同じ数だけ必要となる。例えば、章ごとに `www` の部分を変えて

```
\Opensolutionfile{zzz}[www]
...
\Closesolutionfile{zzz}
```

とすると、章ごとの解答が得られるはずだ。この場合、外部ファイルの読み込みには、`\Readsolution` コマンドではなく、`\input` や `\include` コマンドで読み込むことが可能である。

2.3 `\Writetofile` と `Filesave` 環境

`\Writetofile` コマンドを用いると、(`yyy` 環境などが付加されないで) ファイルに直接書き込むことができる。

```
\Writetofile{zzz}{...}
```

このコマンドは、2つの引数を取る。最初の引数は外部ファイルのハンドル名 `zzz` を、また第2引数はファイルに書くべき内容である。第2引数にコマンドを書く場合は、そのコマンドに `\protect` を冠しなければならない⁴。また、第2引数には段落も許される (改行を含んでも良い)。さらに、コマンドの後の空白は消え去るので、`\xx\yyy` と (外部ファイルに) 書き込みたい場合は、`\protect\xx\space\yyy` とするよう注意が必要である。

これでは不便なので、上記の制限をなくした `Filesave` 環境があらかじめ定義されている。

```
\begin{Filesave}{zzz}
...
\end{Filesave}
```

3 カスタマイズ

3.1 `yyy` 環境の再定義

`\Newassociation` コマンドが実行されると、第2引数である `yyy` 環境が定義される⁵ ようになっている (内部で `\newsolution` が実行され、更にその中で `yyy` 環境が定義されている)。その `yyy` 環境の前後では、それぞれ `\preyyy` と `\postyyy` が実行され、`yyy` 環境では `\yyylabel` や `\yyyparams` とともに `\trivlist` や `\item` が付加されるようになっている。

したがって、`\preyyy` や `\postyyy` を定義⁶、また、`\yyylabel` や `\yyyparams` を再定義すれば、出力形式を変更できる。`\preyyy` は、

```
\newcommand{\preyyy}{...}
```

³`\Newassociation` を複数宣言することで複数の外部ファイルを扱うことも可能。

⁴ \TeX の `\write` コマンドは拡張しているようだ。

⁵もちろん、`xxx` 環境も定義される。

⁶これらはデフォルトで定義されていない。

のように定義しておけば良い。`¥postyyy` も同様である。`¥yyylabel` は、1つの引数をとるので、

```
¥renewcommand{¥yyylabel}[1]{...#1...}
```

の形式で再定義できる。`¥yyyparams` は引数をとらないが、同様に再定義できる。`¥yyylabel` の実体は `¥solutionstyle` なので、これを再定義しても良い。デフォルトでは、

```
¥newcommand{¥solutionstyle}[1]{¥textbf{#1}}
```

となっている。また、`¥yyyparams` の方の実体は、`¥Currentlabel` であり、これを再定義しても良い。デフォルトでは、

```
¥newcommand{¥Currentlabel}{¥@currentlabel}
```

となっている。これらの例は、`ansexam2.tex` が参考になるだろう。しかし、直接 `yyy` 環境を再定義の方が手っ取り早いかもしれない (*cf.* `ansexam3.tex`)。

3.2 ¥Openzzzhook, ¥Closezzzhook の再定義

`¥Opensolutionfile`, `¥Closesolutionfile` コマンドでは、それぞれ `¥Openzzzhook`, `¥Closezzzhook` が実行される。これらを定義⁷すれば、`¥Opensolutionfile`, `¥Closesolutionfile` コマンド実行時の振る舞いを変更できる。これらの第1引数は、常に外部ファイルのハンドル名が入るようである⁸。マニュアル (`answers.dvi`) の9ページには、以下のような例が載っている。

```
¥newcommand{¥Openxxxhook}[2]{%
  ¥Writetofile{#1}{¥protect¥section{#2}}%
}%
¥newcommand{¥Closexxxhook}[1]{%
  ¥Readsolutionfile{#1}%
}

¥Opensolutionfile{xxx}{Answers to selected problems}
...
¥Closesolutionfile{xxx} The default
```

3.3 外部ファイル名の変更

外部ファイル `zzz` の拡張子はデフォルトで `tex` であるが、`¥solutionextension` コマンドで変更できる。また、拡張子の前はデフォルトが `.` で、これは `¥solutionpoint` を再定義することで変更できる。`answers.sty` には、

```
¥newcommand{¥solutionpoint}{.}
¥newcommand{¥solutionextension}{tex}
```

と定義されているが、これらを再定義する必要性は低いと思われる。

⁷再定義ではない。

⁸したがって、以下の例の `xxx` は `zzz` と思った方がよい。