

# pdffill パッケージ

前田一貴

2019 年 1 月 29 日

## 目次

1	概要	1
2	依存パッケージ	1
3	とりあえず使ってみよう	2
4	ユーザ用コマンドの説明	8
5	実装	8
5.1	パッケージの読み込み	8
5.2	オプションの定義と変数の用意	8
5.3	デフォルト値	8
5.4	ユーザ用コマンドの定義	9

## 1 概要

申請書や報告書といった事務書類を作成するとき、慣れない Word や Excel の様式を送られて編集する必要に迫られることが多いです。特にある程度の長さの文章を記述する欄がある場合、Word や Excel では無様な仕上がりになってしまったり、枠が崩れてしまったり、ページ数が不意に増えたりしてイライラします（使い方が下手なせいかもしれないが）。あぐくの果てに記入して保存したファイルを修正のためにもう一度開こうとしたらクラッシュした日には……（←実体験）。

pdffill は使い慣れた  $\text{\LaTeX}$  で美しく事務書類を作成するためのパッケージです。目標は科研費  $\text{\LaTeX}$ <sup>\*1</sup> の方法を一般化することです。

---

<sup>\*1</sup> <http://osksn2.hep.sci.osaka-u.ac.jp/~taku/kakenhiLaTeX/>

## 2 依存パッケージ

tikz, expl3, xparse, pdfpages. さらに，日本語を書くために，LuaTeX ならば LuaTeX-ja<sup>\*2</sup>，XeTeX ならば ZXjatype<sup>\*3</sup> を用意して下さい．なお，このドキュメントでは LuaTeX-ja を用いています．

## 3 とりあえず使ってみよう

細かいことを書く前にまずは使い方を見てみましょう．例として，次ページにあるような申請書に記入することを考えます．

---

<sup>\*2</sup> <http://sourceforge.jp/projects/luatex-ja/wiki/FrontPage>

<sup>\*3</sup> <http://zrbabbler.sp.land.to/zxjatype.html>

ほげほげ申請書

平成 年 月 日

ほげほげ協会 会長 殿

下記の通りほげほげすることを申請します。

フリガナ						
申請者氏名						
会員番号						

ほげほげするに至った理由

この申請書のファイル名を `sampleform.pdf` とします．ここでは `LuaTeX-j` を用いることとし，次のように書いたファイルと同じディレクトリに用意しましょう．

```
\documentclass{ltjsarticle}
\usepackage{pdffill}
%% ここに日本語関連のパッケージの読み込み・設定を書く
\begin{document}
\pdfdefaultoption{grid,draft}
\pdffill{sampleform}{}
\end{document}
```

これを `lualatex` コマンドでコンパイルすると次ページのような出力が得られます (2011/10/14 以降の `LuaTeX-j` をインストールしておく必要あり)．

(注意) PDF ビューワによってはこれで描かれるグリッドが意図通り点線にならずに，見辛くなってしまうことがあるようです．

出力を見ながら `\pdffill{sampleform}{}` の部分を次のように変えてみましょう．

```
\pdffill{sampleform}{
% 日付
\pfnode[right] (13.75, 24.85) {\splitboxes{14.5mm}{23}{5}{9}}
% 名前のフリガナ
\pfnode[right] (5.6, 21.85) {\footnotesize ホゲ タロウ}
% 名前
\pfnode[right] (5.6, 21.3) {保毛 太郎}
% 会員番号
\pfnode[right] (5.38, 20.7) {\splitboxes{7.3mm}{1234567}}
% 申請理由
\pfnode[below right,text width=44\zw] (3.3, 20) {\parindent=1\zw\par
これまでに多くの場でほげほげの経験を積み，その結果あらゆる場面でほげほげする
能力を身に付けることができました．
この能力をより広く社会で生かしていくためには，貴協会ではほげほげすることが
何よりも効果的であると判断しました．
申請が認められた後には，全世界でほげほげするほげほげツアーを実施し，
さらにほげほげをほげほげしていく所存でほげほげ．

ほげほげほげほげほげほげほげほげほげほげほげほげほげほげほげほげほげほげほげ
(以下略)
}
}
```

これで次々ページの出力が得られます．さらに `\pdfdefaultoption{grid,draft}` をコメントアウトすれば完成です．

ほげほげ申請書

平成 年 月 日

ほげほげ協会 会長 殿

下記の通りほげほげすることを申請します。

フリガナ																	
申請者氏名																	
会員番号																	
ほげほげするに至った理由																	

ほげほげ申請書

平成 23 年 5 月 9 日

ほげほげ協会 会長 殿

下記の通りほげほげすることを申請します。

フリガナ	ホゲ タロウ						
申請者氏名	保毛 太郎						
会員番号	1	2	3	4	5	6	7

## ほげほげするに至った理由

これまでに多くの場でほげほげの経験を積み、その結果あらゆる場面でほげほげする能力を身に付けることができました。この能力をより広く社会で生かしていくためには、貴協会でほげほげすることが何よりも効果的であると判断しました。申請が認められた後には、全世界でほげほげするほげほげツアーを実施し、さらにほげほげをほげほげしていく所存でほげほげ。

[illegible]

# ほげほげ申請書

平成 23 年 5 月 9 日

ほげほげ協会 会長 殿

下記の通りほげほげすることを申請します。

[illegible]

## 4 ユーザ用コマンドの説明

そのうち書きます…….

## 5 実装

### 5.1 パッケージの読み込み

まず、必要なパッケージを読み込み、明るい未来のために \ExplSyntaxOn します.

```
1 \RequirePackage{tikz}
2 \RequirePackage{expl3,xparse}
3 \ExplSyntaxOn
4 \RequirePackage{pdfpages}
```

### 5.2 オプションの定義と変数の用意

\l\_pf\_page\_int \pdffill コマンドのオプションを l3keys を用いて定義します.

```
\l_pf_draft_bool 5 \keys_define:nn {pdffill} {
\l_pf_draftcolor_tl 6   page      .int_set:N = \l_pf_page_int,
\l_pf_grid_bool 7   draft     .bool_set:N = \l_pf_draft_bool,
\l_pf_gridcolor_tl 8   draft     .default:n = true,
\l_pf_tics_dim 9   draftcolor .tl_set:N = \l_pf_draftcolor_tl,
\l_pf_gridstep_int 10  grid      .bool_set:N = \l_pf_grid_bool,
\l_pf_labelstep_int 11  grid      .default:n = true,
12  gridcolor .tl_set:N = \l_pf_gridcolor_tl,
13  tics      .dim_set:N = \l_pf_tics_dim,
14  gridstep .int_set:N = \l_pf_gridstep_int,
15  labelstep .int_set:N = \l_pf_labelstep_int,
16 }
```

\g\_pf\_defaultoption\_tl その他、必要な変数を定義します.

```
\l_pf_tmpx_dim 17 \tl_new:N \g_pf_defaultoption_tl
\l_pf_tmpy_dim 18 \dim_new:N \l_pf_tmpx_dim
\l_pf_gridstep_dim 19 \dim_new:N \l_pf_tmpy_dim
20 \dim_new:N \l_pf_gridstep_dim
```

### 5.3 デフォルト値

上で準備した変数のデフォルト値を設定します.

```
21 \int_set:Nn \l_pf_page_int \c_one
22 \tl_set:Nn \l_pf_draftcolor_tl {red}
23 \tl_set:Nn \l_pf_gridcolor_tl {blue!40}
24 \dim_set:Nn \l_pf_tics_dim {2mm}
25 \int_set:Nn \l_pf_gridstep_int \c_one
26 \int_set:Nn \l_pf_labelstep_int \c_ten
```



## 5.4 ユーザ用コマンドの定義

`\pfdefaultoption` `\pdffill` のオプションのデフォルト値を設定するコマンドです。 `\g_pf_defaultoption_tl` に指定された値をセットするだけです。

```
27 \ProvideDocumentCommand \pfdefaultoption { m } {
28   \tl_gset:Nn \g_pf_defaultoption_tl {#1}
29 }
```

`\pfnode` `\pdffill` 中で使用するための TikZ の `\node` に相当するコマンドです。このコマンドを用いることで、`draft` オプションが機能します。

```
30 \ProvideDocumentCommand \pfnode { 0{} u(u,u) +m } {
31   \bool_if:NTF \l_pf_draft_bool {
32     \node[#1,draw=\l_pf_draftcolor_tl] at (#3, #4) {#5};
33     \draw[draw=\l_pf_draftcolor_tl,fill=\l_pf_draftcolor_tl]
34       (#3, #4) circle (0.5mm);
35   } {
36     \node[#1] at (#3, #4) {#5};
37   }
38 }
```

`\pdffill` このパッケージにおいて最も重要なコマンドです。まずはオプションの処理をします。

```
39 \cs_generate_variant:Nn \keys_set:nn {nx}
40 \ProvideDocumentCommand \pdffill { 0{} m +m } {
41   \group_begin:
42   \keys_set:nx {pdffill} {\g_pf_defaultoption_tl,#1}
```

`\l_pf_picturecommand_tl` `\includepdf` に渡す `picturecommand` を定義します。実際には `picture` 環境ではなく `tikzpicture` 環境を使いたいのので、`\put(0, 0)` の中にパディングなしで `overlay` な `tikzpicture` を入れます。そして、冒頭に指定されたコマンドを流し込みます。

```
43 \tl_set:Nn \l_pf_picturecommand_tl {
44   \put(0, 0){\noindent
45     \begin{tikzpicture}[inner~sep=0pt,overlay]
46     #3
```

次に、もし `grid` オプションが指定されていたらグリッドを描きます。

```
47   \bool_if:NT \l_pf_grid_bool {
48     \dim_set:Nn \l_pf_gridstep_dim
49       {1cm * \l_pf_gridstep_int}
50     \draw[color=\l_pf_gridcolor_tl,densely~dotted,step=\l_pf_tics_dim]
51       (0, 0) grid (\paperwidth, \paperheight);
52     \draw[color=\l_pf_gridcolor_tl,densely~dotted,
53       step=\l_pf_gridstep_dim,thick]
54       (0, 0) grid (\paperwidth, \paperheight);
```

目盛を書きます。まずは `x` 方向。

```
55     \dim_set:Nn \l_pf_tmpy_dim {\l_pf_gridstep_dim / 2}
56     \dim_while_do:nNnn \l_pf_tmpy_dim < \paperheight {
```

```

57         \dim_set:Nn \l_pf_tmpx_dim \l_pf_gridstep_dim
58         \int_set_eq:NN \l_tmpa_int \l_pf_gridstep_int
59         \dim_while_do:nNnn \l_pf_tmpx_dim < \paperwidth {
60             \node[color=\l_pf_gridcolor_tl] at (\l_pf_tmpx_dim, \l_pf_tmpy_dim)
61                 {\scriptsize \int_use:N \l_tmpa_int};
62             \dim_add:Nn \l_pf_tmpx_dim \l_pf_gridstep_dim
63             \int_add:Nn \l_tmpa_int \l_pf_gridstep_int
64         }
65         \dim_add:Nn \l_pf_tmpy_dim {\l_pf_gridstep_dim * \l_pf_labelstep_int}
66     }

```

同様にして  $y$  方向も書きます。

```

67         \dim_set:Nn \l_pf_tmpx_dim {\l_pf_gridstep_dim / 2}
68         \dim_while_do:nNnn \l_pf_tmpx_dim < \paperwidth {
69             \dim_set:Nn \l_pf_tmpy_dim \l_pf_gridstep_dim
70             \int_set_eq:NN \l_tmpa_int \l_pf_gridstep_int
71             \dim_while_do:nNnn \l_pf_tmpy_dim < \paperheight {
72                 \node[color=\l_pf_gridcolor_tl] at (\l_pf_tmpx_dim, \l_pf_tmpy_dim)
73                     {\scriptsize \int_use:N \l_tmpa_int};
74                 \dim_add:Nn \l_pf_tmpy_dim \l_pf_gridstep_dim
75                 \int_add:Nn \l_tmpa_int \l_pf_gridstep_int
76             }
77             \dim_add:Nn \l_pf_tmpx_dim {\l_pf_gridstep_dim * \l_pf_labelstep_int}
78         }
79     }
80 \end{tikzpicture}
81 }
82 }

```

最後に目的の PDF ファイルを `\includepdf` して、その上から `picturecommand` を描きます。

```

83 \includepdf[pages=\int_use:N \l_pf_page_int,
84             picturecommand=\l_pf_picturecommand_tl,
85             fitpaper]
86             {#2}
87 \group_end:
88 }

```

`\splitboxes` トークンを等間隔に配置するための簡単なコマンドです。

```

89 \ProvideDocumentCommand \splitboxes { 0 {c} m m } {
90     \tl_map_inline:nn {#3} {\hbox_to_wd:nn {#2} {
91         \bool_if:nT {\str_if_eq_p:nn {#1} {c} || \str_if_eq_p:nn {#1} {r}} {
92             \tex_hss:D
93         }
94         ##1
95         \bool_if:nT {\str_if_eq_p:nn {#1} {c} || \str_if_eq_p:nn {#1} {l}} {
96             \tex_hss:D
97         }
98     }
99 }

```

100 }