

The `fancytooltips` package^{*†}

Robert Mařík
marik@mendelu.cz

June 8, 2010

1 Introduction

The package `fancytooltips` is a package for L^AT_EX. The pdf can be created by `pdflatex` or by `latex + dvips + AdobeDistiller`¹ + Adobe Acrobat². It allows to create tooltips in a similar way like `cooltooltips` package, but the tooltip is a page from another PDF file. In this way you can use mathematics, pictures and animations in your tooltips. The resulting PDF file can be used also with free Adobe Reader.

The tooltips are activated by clicking the active area on the screen and deactivated after closing page or by moving mouse outside the link. You can try the links [here](#) (Einstein's formula) and also [here](#) (animation – numbers from 1 to 6). You have to use the free Adobe Reader or nonfree Adobe Acrobat to see the effect (`xpdf`, `evince` and others fail to work with JavaScripts). For more examples how the presentation may look like see the `examples` subdirectory.

Similar functionality can be achieved using `pdfcomment` (smaller size of the resulting PDF, but restricted to plain text), `ocgtools` (works with layers, only `pdflatex` is supported) and AcroT_EX (works with layers, only `latex + dvips + Adobe Distiller + Adobe Acrobat` are supported). All these “similar” packages allow to write the text for the tooltips and the text for the main document in one file. In contrast to this approach, `fancytooltips` package makes use of an external file. However, if you prefer to write both tooltips in one file, you can preprocess your file using `preview` package and extract tooltips automatically, as shown in the `examples/fancy-preview` directory.

The buttons are created using `eforms.sty` which is a part of AcroTeX bundle.

^{*}This document corresponds to `fancytooltips` v1.8, dated 2010/06/10.

[†]Supported by grants 18/2006, 99/2008 and 131/2010 of Higher Education Development Fund (FRVŠ)

¹not free ps2pdf

²not free Adobe Reader.

2 Usage

2.1 How the package works

We have to prepare external file with content of tooltips first. The pages of this external file are inserted as icons at hidden buttons into the presentation. Whenever we use a command to insert tooltip, the following actions are performed

- The current page is covered by invisible transparent button which spans across the whole page (the information about the pages with tooltips is stored in `aux` file and hence we have to run \LaTeX *three times*). The button has an associated JavaScript action to close all tooltips (i.e. make itself and other related buttons hidden).
- The button is created in text. This button is transparent, covers the text typeset by \TeX into `\hbox` of dimension 0 pt and allows to activate a JavaScript. This JavaScript takes the picture required, puts this picture as icon to the button which covers the corresponding page and makes this button visible.
- Each page has an associated action which closes all tooltips when the page is opened.

2.2 The file with tooltips

The file with tooltips is an ordinary pdf file, one tooltip per page, tooltips should be in the top right corner at the page, in a colored box and the rest of the page should be transparent. If you consider to use `movetips` option (see below), then every page should have the dimensions equal to the dimensions of the colored box with tooltip³. We also provide simple cross referencing mechanism to refer to the tooltips. If the pdf file is created by \LaTeX , you can define keywords to refer to the pages using `\keytip` command. Simply put `\usepackage[createtips]{fancytooltips}` into preamble and write `\keytip{<foo>}` in document. This writes information about keyword `<foo>` and the pagenumber into file `\jobname.tips`.

`\keytip`

2.3 The file with presentation – pdf \LaTeX users

In the file with presentation, the user is responsible

- input either `color` or `xcolor` package in the preamble
- \LaTeX the file two times (we write some macros into `aux` file).

This is not comfortable for the user, but everybody uses different set of packages and from this reason, this part is left to the user. (And among others, the `color`

³Look at the files `tooltipy.tex` and `tooltipy.pdf` from `examples` subdirectory for a simple example how to meet this condition under pdf \LaTeX

or `xcolor` package is probably inputted by the package which is used to build the presentation.)

filename To input the tooltips from file (*foo.pdf*) call the package with `filename` option: `\usepackage[filename=foo]{fancytooltips}`.

movetips By default, tooltip appears in the top right corner of the page (use View–PageLayout–Single Page in your Adobe Reader, please). If the option `movetips` is used, then tooltip appears close to the mouse pointer. More precisely, tooltip appears with left down corner at the mouse position, if there is enough place. If not, tooltip appears with right down corner at the mouse position. Finally, the tooltip is shifted down to fit the page, if necessary⁴.

mouseover If you use `mouseover` option, then tooltip appears if you move the mouse pointer to the active area (no clicking is necessary).

inactive This option makes the package inactive.

active This option forces the package active even if `inactive` option is loaded.

blur The rest of the page is blurred, when showing tooltip. Use either `\usepackage[blur]{fancytooltips}` or `\usepackage[blur=number]{fancytooltips}`, where `number` is a number between 0 and 1. (Note that we use transparent package and hence this could have an influence on the colors of the document and could make your PDF less portable.) This option is allowed in pdfL^AT_EX mode only. If this options brakes colors only on the page which include tooltips, you may want to use option `fixcolor` to fix it.

debug Prints two alerts reporting success or problems when opening PDF file in Adobe Acrobat (Adobe Reader). Use this option to find possible source of problems. See also the section Troubleshooting ... in this manual.

noextratext Do not append any mark at the end of the link which opens tooltips (see help for `\tooltip`).

fg The button for displaying pictures is placed on background and the buttons which activate tooltips are placed immediately in the text by default. With `fg` option all these buttons are placed into foreground after the page is completed. We use `\pdfsavepos` command and keep track of the position for buttons in aux file and hence this option works in pdfL^AT_EX only and the file needs more compilations. This option does not do anything in dvi mode now (despite the fact that pdfL^AT_EX provides the feature also for dvi mode), but this could be changed if someone requests this feature. Use this option for example, if you use buttons and forms in your PDF (like tests produced by AcroT_EX) and the tooltips should be on the top when displayed.

⁴This option works in this way if every page of the file with tooltips has dimensions of the box with tooltip. See the `examples` subdirectory.

<code>\tooltip</code>	The user can put the tooltip into her or his presentation using the command <code>\tooltip{<stuff>}{<keyword-or-pagenu-ber>}</code> where <code><stuff></code> is the printed text in <code><tooltipcolor></code> color and <code><keyword-or-pagenu-ber></code> is either the pagenu-ber of the tooltip in the external file or the keyword defined by <code>\keytip</code> command and stored in <code>\TooltipFilename.tips</code> file, where <code>\TooltipFilename</code> is set automatically from the <code>filename</code> option. The printed text <code><stuff></code> is fol-lowed by <code>\TooltipExtratext</code> command. The default value is small blue soap, as you have seen in the second paragraph of this documentation. There is a package <code>noextratext</code> which defines <code>\TooltipExtratext</code> to be empty. If <code>{<keyword-or-pagenu-ber>}</code> is not recognized as valid keyword for tooltips,
<code>\TooltipExtratext</code>	
<code>noextratext</code> option	
<code>\tooltip*</code>	The text <code>{<stuff>}</code> is inserted in <code>\hbox</code> by <code>\tooltip</code> . With starred version of the <code>\tooltip</code> macro the text <code>{<stuff>}</code> is not inserted into the box and the active button dos not cover the text <code>{<stuff>}</code> , but covers the mark produced by <code>\TooltipExtratext</code> .
<code>\tooltipanim</code>	The user can put a series (animation) of tooltips into the presentation by using
<code>\tooltipanim*</code>	<code>\tooltipanim{<stuff>}{<start>}{<end>}</code> command, where <code><start></code> and <code><end></code> are
<code>\delayinterval</code>	keywords defined by <code>\keytip</code> command or page numbers. The delay between two frames is <code>\delayinterval</code> milliseconds. The default value is 200, you can change it by command <code>\def\delayinterval{100}</code> . There is also starred version which works similarly like <code>\tooltip*</code> command.
<code>\FancyHook</code>	Redefines <code>\ref</code> and <code>\eqref</code> commands to work with preview, see the demo files in <code>example/fancy-preview</code> directory.
<code>\TooltipRefmark</code>	Extra text added to <code>\ref</code> and <code>\eqref</code> commands with previews, see the demo files in <code>example/fancy-preview</code> directory. The default value is the same as for <code>\TooltipExtratext</code> .

2.4 Changes for dvips users

Dvips users also have to prepare tooltips into PDF file, not eps as usual. But we have to insert these tooltips in Adobe Acrobat Pro program. If you use Acrobat Pro version 8.1 and later, install the file `aeb.js` from AcroTeX eDucation bundle as described in the documentation to AcroTeX.⁵

Since L^AT_EX is not capable to find the number of pages in external PDF file with tooltips, dvips users have to specify option `dvips` in `fancytooltips` package. You have to use also a `pages` option with the number of pages in the PDF file with tooltips, if you use dvips route. You have to call the package by something like this:

```
\usepackage[dvips,filename=tooltipy,pages=27]{fancytooltips}
```

You have to `latex` (two times) and `dvips` your file first. This produces `filename.ps` and `Tooltipsdljs.fdf` files. Distill the `filename.ps` file into `filename.pdf` and open this file by Adobe Acrobat Professional – this imports macros from `Tooltipsdljs.fdf` file. In Acrobat’s JavaScript console (open by `Crtl+J`) run (using `Ctrl+Enter`) the command `ImportTooltips()`; which is defined for the document. This command inserts invisible buttons on the first page,

⁵If you do not install `aeb.js` properly, you can still create your presentation , but you have to import `fdf` file manually, see the section Troubleshooting and known problems.

imports icons (the file with icons specified as *filename*) parameter when loading fancytooltips must be in working directory). You should see a message “**importing pictures**” and the command returns 1 when finished. Then **save the file** (you can use the same name). If the command `ImportTooltips();` fails, you either have not the PDF file with tooltips in current directory, or the PDF file does not contain JavaScripts. In the latter case insert document level JavaScripts manually as described in the next section.

3 Troubleshooting and known problems

The source code is in Mercurial repository at <http://bitbucket.org/robert.marik/fancytooltips/>. You can also report problems and issues in the forum at this site. The code on bitbucket.org is considered as development version and repository for older versions. The last stable version is always the version from CTAN.

- The package works with `eforms.sty` from version 2006/10/03 v1.0a. You can download this or newer version from www.arotex.net site.
- If the graphics included by `\TooltipExtratext` and `\TooltipRefmark` has colors with **custom opacity**, Adobe Acrobat Pro sometimes renders the pictures bad. No problems of this type have been observed with free Adobe Reader.
- For **dvips** users: In some cases the file `Tooltipsdljs.fdf` is **not imported automatically** (probably some setting in Adobe Acrobat or new versions of `eforms.sty` and `insdljs.sty`). In this case you do not see any message when using `debug` option. You have troubles of this type if you see in the Javascript console (Ctrl+J) error messages like “`aebTrustedFunctions is not defined 3:Page:Open CloseTooltips is not defined 1:Page:Open`”. In this case you have to import the file `Tooltipsdljs.fdf` **manually from “Form” menu** in Adobe Acrobat Pro. Then go to the JavaScript console and run `ImportTooltips();` command.

Follow the points below if you want to find the source of your problems.

- For dvips users it is a good idea to check that AcroTeX is properly installed. Do the demo files from AcroTeX work for you?
- Try to use `debug` option, prepare the PDF file and open it in Adobe Acrobat or Adobe Reader.
 - You should see two messages when opening the file. If not, the Javascript do not work in your document (are not inserted properly).
 - Both messages should report success for pdflatex users. For dvips users one of the message should report error and if you run `ImportTooltips();` command in Javascript console, you should see a message in Javascript

console which confirms that the pictures from external PDF file have been imported. If you save the PDF file and open again, both messages should report success.

4 Implementation

```
1 (*package)
2 \RequirePackage{graphicx}
3 \RequirePackage{xkeyval}
4 \RequirePackage{atbegshi}
5
6 \newif\if@fancytooltips@createtips\@fancytooltips@createtipsfalse
7 \DeclareOptionX{createtips}{\@fancytooltips@createtipstrue}
8
9 \newif\if@fancytooltips@usepdftex\@fancytooltips@usepdftexttrue
10 \DeclareOptionX{dvips}{\@fancytooltips@usepdftextfalse}
11
12 \newif\if@fancytooltips@extratext\@fancytooltips@extratexttrue
13 \DeclareOptionX{noextratext}{\@fancytooltips@extratextfalse}
14
15 \newif\if@fancytooltips@movetips\@fancytooltips@movetipsfalse
16 \DeclareOptionX{movetips}{\@fancytooltips@movetipstrue}
17
18 \newif\if@fancytooltips@mouseover\@fancytooltips@mouseoverfalse
19 \DeclareOptionX{mouseover}{\@fancytooltips@mouseovertrue}
20
21 \newif\if@fancytooltips@inactive\@fancytooltips@inactivetrue
22 \DeclareOptionX{inactive}{\@fancytooltips@inactivetrue}
23
24 \newif\if@fancytooltips@active\@fancytooltips@activetrue
25 \DeclareOptionX{active}{\@fancytooltips@activetrue}
26
27 \newif\if@fancytooltips@fg\@fancytooltips@fgfalse
28 \DeclareOptionX{fg}{\@fancytooltips@fgtrue}
29
30 \DeclareOptionX{filename}{\xdef\TooltipFilename{#1}}
31 \DeclareOptionX{pages}{\xdef\TooltipPages{#1}}
32
33 \newif\if@fancytooltips@blur\@fancytooltips@blurfalse
34 \DeclareOptionX{blur}[0.4]{\@fancytooltips@blurtrue
35 \xdef\fancytooltips@transparency{#1}}
36
37 \newif\if@fancytooltips@fixcolor\@fancytooltips@fixcolorfalse
38 \DeclareOptionX{fixcolor}{\@fancytooltips@fixcolortrue}
39
40 \def\fancytooltipsdebugmsg{}
41 \DeclareOptionX{debug}{\def \fancytooltipsdebugmsg
42 {
43 if (this.getField("ikona") == null)
```

```

44 {app.alert("No buttons for placing tooltips are available. Contact the author. The file may
45 else
46 {app.alert("Buttons for placing tooltips are available. Congratulations! Hope everything will
47 if (this.getField("animtip") == null)
48 {app.alert("No hidden buttons containing tooltips available. The interactive features will no
49 else
50 {app.alert("Hidden buttons containing tooltips are available. Congratulations! Hope everythin
51 }}
52
53 \ProcessOptionsX
54
55 \if@fancytooltips@blur
56 \ifTooltip@usepdftex\else
57 \@fancytooltips@blurfalse
58 \AtEndDocument{\PackageWarning{fancytooltips}
59 {Blur option is incompatible with dvips. ^^J The option blur is turned off }}
60 \fi
61 \fi
62
63 \ifTooltip@usepdftex\else
64 \@fancytooltips@fgfalse
65 \fi
66
67 \newdimen\buttontipwidth
68 \newdimen\buttontipheight
69 \newdimen\fancy@a
70 \newdimen\fancy@b
71 \newdimen\fancy@layerHshift\fancy@layerHshift=0pt
72 \newdimen\fancy@layerVshift\fancy@layerVshift=0pt
73 \newdimen\fancy@button@Vshift \fancy@button@Vshift=0pt
74 \newdimen\fancy@button@Hshift \fancy@button@Hshift=0pt
75 \newtoks\pos@fancy@toks
76
77 \if@fancytooltips@active\@fancytooltips@inactivefalse\fi
78
79 \if@fancytooltips@inactive
80 \newcommand{\tooltip}{\@ifstar\tooltip@Star\tooltip@NoStar}%
81 \newcommand{\tooltip@Star}[2]{#1}%
82 \newcommand{\tooltip@NoStar}[2]{#1}%
83
84 \newcommand{\tooltipananim}{\@ifstar\tooltipananim@Star\tooltipananim@NoStar}%
85 \newcommand{\tooltipananim@Star}[2]{#1}%
86 \newcommand{\tooltipananim@NoStar}[2]{#1}%
87 \def\keytip#1{}%
88 \def\FancyHook{}%
89 \def\TooltipPage#1#2{}%
90 \let\TooltipExtratext\relax
91 \let\TooltipRefmark\relax
92 \ifx\@cog@makeknown\undefined
93 \def\@cog@makeknown#1#2#3{\fi

```

```

94 \def\fancy@pushButton#1#2#3#4#5#6#7#8{}
95 \def\fancy@anim@pushButton#1#2#3#4#5#6#7#8#9{}
96
97 \PackageWarning{fancytooltips}{Fancytooltips inactive}%
98 \expandafter\endinput\fi
99

```

Each `\ref` command has associated a number and writes new label into aux file. If the target page for the ref is different from the page with this `\ref` command, the corresponding tooltip is inserted.

```

100 \newcount\fancycheckcount\fancycheckcount=0
101 \def\fancy@second#1#2#3#4{#2}
102
103 \def\FancyHook{
104 \global\let\oldref\ref
105 \gdef\ref##1{\oldref{##1}\global\advance\fancycheckcount by 1\relax
106 \edef\templabel{\fancyanchorref:\the\fancycheckcount}%
107 \expandafter\label\expandafter{\templabel}%
108 \expandafter\ifx\csname FancyToolTip@##1\endcsname\relax\else\hbox to 0 pt{%
109 \expandafter\ifx\csname r@##1\endcsname\relax\else
110 \edef\temp{\csname r@##1\endcsname}%
111 \expandafter\edef\expandafter\fan@tempa\expandafter{\expandafter\fancy@second\temp}%
112 \fi
113 \expandafter\ifx\csname r@fancyanchorref:\the\fancycheckcount\endcsname\relax\else
114 \edef\fan@temp{r@fancyanchorref:\the\fancycheckcount}%
115 \expandafter\let\expandafter\fan@temp@w\csname \fan@temp\endcsname
116 \edef\fan@temp@ww{\expandafter\fancy@second\fan@temp@w}%
117 \fi
118 \ifx\fan@temp@ww\undefined\else
119 \fi
120 \ifx\fan@tempa\fan@temp@ww\else\smash{%
121 \let\TooltipExtratext\relax\tooltip{\strut\TooltipRefmark}{##1}}%
122 \fi
123 \hss}%
124 \fi}
125 \ifTooltip@usepdftex
126 \def\TooltipRefmark{\hbox {\smash{\raisebox{0.4em}{\includegraphics[width=0.7em]{
127 fancytipmark.pdf}}}}}%
128 \else
129 \def\TooltipRefmark{\hbox {\smash{\raisebox{0.4em}{\includegraphics[width=0.7em]{
130 fancytipmark.eps}}}}}%
131 \fi%\ifTooltip@usepdftex
132 }
133
134 \if@fancytooltips@createtips

```

This part (three lines) is processed if the option `createtips` is used. In the opposite case we process the second part, up to the end of the package.

```

135 \newwrite\tipfile
136 \immediate\openout\tipfile \jobname.tips
137 \def\keytip#1{\write\tipfile{\string\tooltipname{#1}{\arabic{page}}}}

```



```
138 \expandafter\endinput\fi
```

This part is processed if the option `createtips` is not used. We define macros which put the hidden button with the name `ikona.n` in the background of the page `n`, if one of the commands `\tooltip` or `\tooltipanim` has been used on this page. Javascripts defined by `\tooltip` and `\tooltipanim` commands then unhide this button and show the corresponding picture.

```
139
```

```
140 \ifx\TooltipFilename\undefined
```

```
141 \PackageWarning{fancytooltips}{** The filename with tooltips is not given. **}
```

```
142 \fi
```

```
143
```

```
144 \ifTooltip@usepdftex
```

```
145 \RequirePackage[pdftex]{eforms}
```

```
146 \def\TooltipExtratext{\hbox{\smash{\raisebox{0.4em}{\includegraphics[width=0.7em]{
```

```
147   fancytipmark.pdf}}}}}
```

```
148 \else
```

```
149 \RequirePackage[dvips]{eforms}
```

```
150 \def\TooltipExtratext{\hbox{\smash{\raisebox{0.4em}{\includegraphics[width=0.7em]{
```

```
151   fancytipmark.eps}}}}}
```

```
152 \fi%\ifTooltip@usepdftex
```

```
153 \if@fancytooltips@extratext\else\let\TooltipExtratext\relax\fi
```

```
154
```

```
155 \AtBeginDocument{
```

```
156 \global\buttontipwidth=\paperwidth
```

```
157 \global\buttontipheight=\paperheight
```

```
158 }
```

```
159
```

```
160 \ifTooltip@usepdftex
```

```
161 \def\frametip@{%
```

```
162   \pdfstartlink user{%
```

```
163     /Subtype /Widget
```

```
164     /F 6
```

```
165     /T (ikona.\thepage)
```

```
166     /FT /Btn
```

```
167     /Ff 65536
```

```
168     /H /N
```

```
169     /BS << /W 1 /S /S >>
```

```
170     /MK << /TP 1 /IF <</A[1.0 1.0]/SW /B>> >>
```

```
171   }%
```

```
172   \vbox to \buttontipheight {\vss\hbox to \buttontipwidth{\hss}\pdfendlink}%
```

```
173 \else
```

For dvips users we use the macros from `eqxerquiz.sty` package.

```
174 \def\everyeqIcon#1{\def\every@eqIcon{#1}}
```

```
175 \def\every@eqIcon{}
```

```
176 \newcommand\eqIconFTT[4] []
```

```
177 {%
```

```
178   \push@Button{#1}{#2}{#3}{#4}{\eq@setButtonProps\eq@Button@driver}%
```

```
179   {\eqIconDefaults\every@ButtonField\every@eqIcon}%
```

```
180 }
```

```

181 \def\eqIconDefaults
182 {%
183   \rawPDFf{S}{mkIns{/TP 1 /IF<</A[1.0 1.0]/SW/B>>}\R{0}
184   \CA{}\RC{}\AC{}\BC{}\BG{}\H{B}
185   \textColor{0 g}\Ff{\FfReadOnly}
186 }
187 \def\frametip@{\eqIconFTT[\BC{}\BG{}\F{\FHIDDEN}]}%
188 {ikona.\thepage}{\paperwidth}{\paperheight}%
189 }%
190 \fi%\ifTooltip@usepdftex
191
192 \def\fancytooltips@one{1}
193 \if@fancytooltips@blur
194   \RequirePackage{ocg}
195   \ifx\fancytooltips@one\fancytooltips@transparency
196     \def\transparent#1{}
197   \else
198     \RequirePackage{transparent}
199   \fi
200 \else
201   \ifx@ocg@makeknown\undefined
202     \def@ocg@makeknown#1#2#3{}\fi
203 \fi
204
205 \if@fancytooltips@fg\else
206 \RequirePackage{eso-pic}
207 \def\frametip{%
208   \expandafter\ifx \csname TooltipPage\thepage\endcsname\relax
209   \else
210     \setbox0=\hbox{\frametip@}%
211     \hbox{\raise \dp0 \box0}
212   \fi}%
213 \AddToShipoutPicture{\hbox to 0 pt{\frametip\hss}}
214 \fi
215
216 \def\fancytooltips@save@position{\pdfsavepos%
217   \write@auxout{\string\global\string \fancy@layerVshift \the\pdfastypos sp\string\relax}%
218   \write@auxout{\string\global\string \fancy@layerHshift \the\pdfastxpos sp\string\relax}%
219 }
220
221 \def\fancy@beginshipout@hook{}
222 \AtBeginShipout{%
223   \TooltipPageopenCloseJS
224   \setbox\AtBeginShipoutBox=\hbox{%
225     \hbox to 0 pt{\TooltipHidden}\global\def\TooltipHidden{}%
226     \fancy@beginshipout@hook\if@fancytooltips@fixcolor\hbox to 0 pt{\resizebox{1pt}{!}{\TooltipEx
227     \hbox{\box\AtBeginShipoutBox}}%
228   \ifTooltip@usepdftex
229     \fancytooltips@save@position
230   \if@fancytooltips@blur

```

```

231 \expandafter\ifx \csname TooltipPage\thepage\endcsname\relax \else
232 \lower\fancy@layerVshift\hbox to 0 pt{\kern-\fancy@layerHshift\relax
233 \begin{ocg}{fancyOCG\thepage}{fancyOCG\thepage}{0}%
234 \expandafter\transparent\expandafter{\fancytooltips@transparency}%
235 \color{black}%
236 \vbox to 0 pt{\vss\hbox{\vrule width \paperwidth height \paperheight}}%
237 \end{ocg}\hss}%
238 \fi
239 \fi
240 \if@fancytooltips@fg
241 \expandafter\ifx \csname TooltipPage\thepage\endcsname\relax
242 \else
243 \lower\fancy@layerVshift\vbox to 0 pt{\vss\hbox to 0 pt{\kern-\fancy@layerHshift\relax\hbox
244 \fi
245 \lower\fancy@layerVshift\vbox to 0 pt{\vss\hbox to 0 pt{\kern-\fancy@layerHshift\relax\hbox
246 \fi
247 \fi
248 }%
249 }%
250

```

In the macros `\tooltip` and `\tooltipanim` we print the text into box with zero dimensions and then we build a button which covers this text and has an associated JavaScript action. The important part is the `\PushButton` macro. You can adjust these macros or write similar macros which do what you need.

```

251 \definecolor{tooltipcolor}{rgb}{0,0,1}
252
253 \newcount\tooltip@count
254 \newtoks\tooltip@toks
255 \newtoks\tooltip@pagetoks
256 \tooltip@pagetoks={\thepage}
257 \def\tooltippage{}
258
259 \def\TooltipPage#1#2{%
260 \expandafter\gdef\csname TooltipPage#2\endcsname{#2}%
261 \expandafter\gdef\csname Tooltipcount2page#1\endcsname{#2}%
262 }

```

We use the hack for Adobe Acrobat suggested by DPS and Jorg at <http://www.acrotex.net/forum/showthread.php?p=10>

```

263
264 \def\fancy@JS#1{\JS{var DirtyBeforeTooltip=this.dirty; #1
265 this.dirty=DirtyBeforeTooltip;}}
266
267 \newcommand{\tooltip}{\@ifstar
268 \tooltip@Star%
269 \tooltip@NoStar%
270 }
271
272 \newcommand{\tooltip@Star}[2]{\color{tooltipcolor}#1}%
273 {\let\SaveTooltipExtratext\TooltipExtratext
274 \let\TooltipExtratext\relax

```

```

275 \hbox to 0 pt{\tooltip@NoStar{\SaveTooltipExtratext
276 \vrule height 10pt depth 0 pt width 0 pt}{#2}\hss}}
277
278 \newcommand{\tooltip@NoStar}[2]{%
279 \global\advance\tooltip@count by 1%
280 \edef\act{\write\@auxout{\noexpand\string\noexpand\TooltipPage{\the\tooltip@count}{\the\toolt
281 \expandafter\ifx\csname Tooltipcount2page\the\tooltip@count \endcsname\relax
282 \global\edef\tooltippage{
283 \else
284 \global\edef\tooltippage{\csname Tooltipcount2page\the\tooltip@count \endcsname}%
285 \fi
286 \checkTipNumber{#2}\edef\TipNumber{\FindTipNumber{#2}}%
287 \leavevmode
288 \setbox0=\hbox{\color{tooltipcolor}{#1}}\copy0\fancy@a=\dp0\advance\fancy@a by \ht0\relax\h
289
290 \def\fancy@tooltip@options#1#2{\BC{} \BG{}
291 \S{} \AA{\AAMouseExit{\fancy@JS{CloseTooltips()}}}%
292 \if@fancytooltips@mouseover
293 \AAMouseEnter{\fancy@JS{this.getField("ikona."+(#2)).hidden=false;
294 try {app.clearInterval(animace);}catch (e) {}
295 \if@fancytooltips@movetips nastav(#1,#2);\fi
296 \if@fancytooltips@blur
297 try{
298 for(var i=0; fancytooltipsOCGs && i<fancytooltipsOCGs.length;i++)
299 {
300 if(fancytooltipsOCGs[i].name == "fancyOCG"+(#2))
301 fancytooltipsOCGs[i].state = true;
302 else
303 fancytooltipsOCGs[i].state = false;
304 }} catch (e) {};
305 \fi
306 zobraz(#1,#2);
307 }}
308 \fi}
309 \A{\fancy@JS{this.getField("ikona."+(#2)).hidden=false;
310 try {app.clearInterval(animace);}catch (e) {}
311 \if@fancytooltips@movetips nastav(#1,#2);\fi
312 \if@fancytooltips@blur
313 try {
314 for(var i=0; fancytooltipsOCGs && i<fancytooltipsOCGs.length;i++)
315 {
316 if(fancytooltipsOCGs[i].name == "fancyOCG"+(#2))
317 fancytooltipsOCGs[i].state = true;
318 else
319 fancytooltipsOCGs[i].state = false;
320 }} catch (e) {};
321 \fi
322 zobraz(#1,#2);
323 }}}
324

```

```

325 \newtoks\@fxtoks\@fxtoks={\the\pdflastxpos}
326 \newtoks\@fytoks\@fytoks={\the\pdflastypos}
327
328 \def\fancy@pushButton#1#2#3#4#5#6{%
329   \lower #6\hbox to 0 pt{\hss\expandafter\pushButton\expandafter[\fancy@tooltip@options{#1}{#2}
330 \def\fancy@@pushButton#1#2#3#4#5#6#7#8{
331
332 \ifTooltip@usepdfTeX
333 \if@fancy@tooltips@fg
334 \def\fancy@pushButton#1#2#3#4#5#6{\pdfsavepos%
335   \edef\act{\write\@auxout{\string\fancy@@pushButton{#1}{#2}{#3}{\the#4}{\the#5}{\the\@fxtoks}{
336 }
337 \def\fancy@@pushButton#1#2#3#4#5#6#7#8{%
338   \expandafter\global\expandafter\pos@fancy@toks\expandafter{\the\pos@fancy@toks\fancy@onlypage
339 \fi
340 \fi
341
342
343 \def\fancy@onlypage#1#2{\def\ft@a{#1}\edef\ft@b{\the#2}}%
344 \ifx\ft@a\ft@b#2\fi
345
346
347 \def\delayinterval{200}
348
349 \newcommand{\tooltipanim}{\@ifstar
350   \tooltipanim@Star%
351   \tooltipanim@NoStar%
352 }
353
354 \newcommand{\tooltipanim@Star}[3]{\color{tooltipcolor}#1}%
355   {\let\SaveTooltipExtratext\TooltipExtratext
356   \let\TooltipExtratext\relax
357   \hbox to 0 pt{\tooltipanim@NoStar{\SaveTooltipExtratext
358     \vrule height 10pt depth 0 pt width 0 pt}{#2}{#3}\hss}}
359
360 \newcommand{\tooltipanim@NoStar}[3]{%
361   \global\advance\tooltip@count by 1%
362   \edef\act{\write\@auxout{\noexpand\string\noexpand\TooltipPage{\the\tooltip@count}{\the\toolt
363   \expandafter\ifx\csname Tooltipcount2page\the\tooltip@count \endcsname\relax
364   \global\edef\tooltippage{
365   \else
366   \global\edef\tooltippage{\csname Tooltipcount2page\the\tooltip@count \endcsname}%
367   \fi
368   \checkTipNumber{#2}\edef\TipNumberA{\FindTipNumber{#2}}%
369   \checkTipNumber{#3}\edef\TipNumberB{\FindTipNumber{#3}}%
370   \leavevmode
371   \setbox0=\hbox{\color{tooltipcolor}{#1}}\copy0\fancy@a=\dp0\advance\fancy@a by \ht0\relax\h
372
373
374 \def\fancy@tooltipanim@options#1#2#3{

```

```

375 \BC{} \BG{} \S{} \AA{ \AAMouseExit{ \fancy@JS{ CloseTooltips(); } }
376 \if@fancytooltips@mouseover
377 \AAMouseEnter{ \fancy@JS{
378     try { app.clearInterval(animace); } catch (e) {}
379     var cislo=#1;
380     \if@fancytooltips@movetips nastav(#1,#2); \fi
381     \if@fancytooltips@blur
382     try{
383     for(var i=0; fancytooltipsOCGs && i<fancytooltipsOCGs.length;i++)
384     {
385         if(fancytooltipsOCGs[i].name == "fancyOCG"+(#2))
386         fancytooltipsOCGs[i].state = true;
387         else
388         fancytooltipsOCGs[i].state = false;
389     } catch (e) {};
390     \fi
391     function animuj()
392     {
393         var DirtyBeforeTooltipananim=this.dirty;
394         if (cislo<#3) cislo=cislo+1;
395         this.getField('ikona.'+#2).buttonSetIcon(this.getField("animtiph."+cislo).buttonGetIcon());
396         this.dirty=DirtyBeforeTooltipananim;
397     };
398     this.getField('ikona.'+#2).buttonSetIcon(this.getField("animtiph."+#1).buttonGetIcon());
399     this.getField("ikona."+#2).hidden=false;
400     animace=app.setInterval('animuj();', \delayinterval);
401     }}
402 \fi}
403 \A{ \fancy@JS{
404     try { app.clearInterval(animace); } catch (e) {}
405     var cislo=#1;
406     \if@fancytooltips@movetips nastav(#1,#2); \fi
407     \if@fancytooltips@blur
408     try{
409     for(var i=0; fancytooltipsOCGs && i<fancytooltipsOCGs.length;i++)
410     {
411         if(fancytooltipsOCGs[i].name == "fancyOCG"+(#3))
412         fancytooltipsOCGs[i].state = true;
413         else
414         fancytooltipsOCGs[i].state = false;
415     } catch (e) {};
416     \fi
417     function animuj()
418     {
419         var DirtyBeforeTooltipananim=this.dirty;
420         if (cislo<#3) cislo=cislo+1;
421         this.getField('ikona.'+#2).buttonSetIcon(this.getField("animtiph."+cislo).buttonGetIcon());
422         this.dirty=DirtyBeforeTooltipananim;
423     };
424     this.getField('ikona.'+#2).buttonSetIcon(this.getField("animtiph."+#1).buttonGetIcon());

```

```

425     this.getField("ikona."+(#2)).hidden=false;
426     animace=app.setInterval('animuj();', \delayinterval);
427   }}
428 }
429
430
431 \def\fancy@anim@pushButton#1#2#3#4#5#6#7{%
432   \lower #6 \hbox to 0 pt{\hss\expandafter\pushButton\expandafter[\fancy@tooltipanim@options{#1
433   \def\fancy@@anim@pushButton#1#2#3#4#5#6#7#8#9{}}
434
435   \ifTooltip@usepdfTeX
436   \if@fancytooltips@fg
437   \def\fancy@anim@pushButton#1#2#3#4#5#6#7{\pdfsavepos%
438     \edef\act{\write\auxout{\string\fancy@anim@pushButton{#1}{#2}{#3}{\the#4}{\the#5}{\the\@fxt
439   }}
440   \def\fancy@@anim@pushButton#1#2#3#4#5#6#7#8#9{%
441     \expandafter\global\expandafter\pos@fancy@toks\expandafter{\the\pos@fancy@toks\fancy@onlypage
442   \fi
443   \fi
444
445
446   This code closes tooltip if the page is closed.
447   \edef\fancytooltips@pdfpageattrJS{%
448     var DirtyBeforeCloseTooltip=this.dirty;
449     \if@fancytooltips@blur
450     try{
451       var temp = fancytooltipsOCGs.length;
452       for(var i=0; fancytooltipsOCGs && i<fancytooltipsOCGs.length;i++)
453       {
454         fancytooltipsOCGs[i].state = false;
455       }
456     }
457     catch (e){}
458     \fi
459     CloseTooltips();
460     this.dirty=DirtyBeforeCloseTooltip;
461   }
462   \ifTooltip@usepdfTeX
463   \edef\fancy@temp{/AA << /O << /S /JavaScript /JS (
464     \fancytooltips@pdfpageattrJS) >> >>}
465   \expandafter\def\expandafter\TooltipPageopencloseJS\expandafter{
466   \expandafter\global\expandafter\pdfpageattr\expandafter{\fancy@temp}%
467   }
468   \pdfximage{\TooltipFilename.pdf}%
469   \edef\TooltipPages{\the\pdfLASTximagepages}%
470   \else
471   \edef\fancy@temp{ [ {ThisPage} << /AA <<
472     /O << /S /JavaScript /JS (var DirtyBeforeCloseTooltip=this.dirty; CloseTooltips(); this.dir
473     >> >> /PUT pdfmark}

```

```

473 \expandafter\def\expandafter\TooltipPageopencloseJS\expandafter{
474 \expandafter\literalps@out\expandafter{\fancy@temp}}
475 \edef\fancy@temp{/S /JavaScript /JS ( \fancytooltips@pdfpageattrJS)}
476 \expandafter\OpenAction\expandafter{\fancy@temp}
477 \fi%\ifTooltip@usepdftex
478
479 \ifTooltip@usepdftex
480 \def\fancytempA{}
481 \ifx\FancytooltipsAfterClose\undefined\else\edef\fancytempA{\FancytooltipsAfterClose}\fi
482 \def\fancytempAA{}
483 \ifx\FancytooltipsAfterShow\undefined\else\edef\fancytempAA{\FancytooltipsAfterShow}\fi
484
485 \if@fancytooltips@blur
486 \def\fancytempAAA{
    OCG's have not been initialized yet, do it now.
487   var inifancytooltipsOCGs = this.getOCGs();
488   var fancytooltipsOCGs = [];
489   for(var i=0; inifancytooltipsOCGs && i<inifancytooltipsOCGs.length;i++)
490   {
491     if(inifancytooltipsOCGs[i].name.substr(0,5) == "fancy")
492     {
493       fancytooltipsOCGs.push(inifancytooltipsOCGs[i]);
494       inifancytooltipsOCGs[i].state=false;
495     }
496   }
497   inifancytooltipsOCGs[0].state=true;
498   inifancytooltipsOCGs[0].state=false;
499 }
500 \else
501 \def\fancytempAAA{}
502 \fi
503
504 \if@fancytooltips@blur
505 \def\fancytempB{
506   try {for(var i=0; fancytooltipsOCGs && i<fancytooltipsOCGs.length;i++)
507     {fancytooltipsOCGs[i].state = false;} } catch (e) {}}
508 \else
509 \def\fancytempB{}
510 \fi
511
512
513 \begin{insDLJS}[fancyTooltipsLoaded]{Tooltipsdljs}{DLJS for Tooltips}
514   \fancytooltipsdebugmsg
515
516   var animace;
517   var fancyTooltipsLoaded = true;
518
519   \fancytempAAA
520

```



```

521 function CloseTooltips()
522 {
523   try {this.getField("ikona").hidden=true;}catch (e) {}
524   try {app.clearInterval(animace);}catch (e) {}
525   \fancytempB
526   \fancytempA
527 }
528
529 function nastav(cislo,strana)
530 {
531   var f=this.getField("ikona."+strana);
532   var g=this.getField("animtiph."+cislo);
533   var sourf=f.rect;
534   var sourg=g.rect;
535   if ((mouseX+sourg[2]-sourf[0])<sourf[2])
536   var percX=100*(mouseX-sourf[0])/((sourf[2]-sourf[0])-(sourg[2]-sourf[0]));
537   else
538   var percX=100*(mouseX-sourf[0]-(sourg[2]-sourf[0]))/((sourf[2]-sourf[0])-(sourg[2]-sourf[0]));
539   var percY=100*(mouseY-sourf[3])/((sourf[1]-sourf[3])-(sourg[1]-sourf[3]));
540   if (percX>100) percX=100;
541   if (percY>100) percY=100;
542   if (percX<0) percX=0;
543   if (percY<0) percY=0;
544   f.buttonAlignX=percX;
545   f.buttonAlignY=percY;
546 }
547
548 function zobraz(cislo,strana)
549 {
550   var f=this.getField("ikona."+strana);
551   var g=this.getField("animtiph."+cislo);
552   f.hidden=false;
553   f.buttonSetIcon(g.buttonGetIcon());
554   \fancytempAA
555 }
556
557 this.dirty=false;
558 app.focusRect = false;
559 \end{insDLJS}
560 \else
561 \begin{insDLJS}[fancyTooltipsLoaded]{Tooltipsdljs}{DLJS for Tooltips}
562   var animace;
563   var fancyTooltipsLoaded = true;
564
565   \fancytooltipsdebugmsg
566
567   function CloseTooltips()
568   {
569     try {this.getField("ikona").hidden=true;}catch (e) {}
570     try {app.clearInterval(animace);}catch (e) {}

```

```

571 }
572
573 function ImportTooltips()
574 {
575     console.println("importing pictures");
576     for (var i=1;i<=\TooltipPages;i++)
577     {
578         this.insertPages(this.numPages-1,"\TooltipFilename.pdf",(i-1),(i-1));
579         var rozm=this.getPageBox("Crop",this.numPages-1);
580         this.deletePages(this.numPages-1);
581         var p=this.addField("animtiph."+i,"button",0,rozm);
582         p.buttonPosition=position.iconOnly;
583         p.hidden=true;
584         this.getField("animtiph."+i).buttonImportIcon("\TooltipFilename.pdf",(i-1));
585     }
586     console.println("Imported \TooltipPages pictures, save the PDF file.");
587     return(1);
588 }
589
590 function nastav(cislo,strana)
591 {
592     var f=this.getField("ikona."+strana);
593     var g=this.getField("animtiph."+cislo);
594     var sourf=f.rect;
595     var sourg=g.rect;
596     if ((mouseX+sourg[2]-sourf[0])<sourf[2])
597     var percX=100*(mouseX-sourf[0])/((sourf[2]-sourf[0])-(sourg[2]-sourf[0]));
598     else
599     var percX=100*(mouseX-sourf[0]-(sourg[2]-sourf[0]))/((sourf[2]-sourf[0])-(sourg[2]-sourf[0]));
600     var percY=100*(mouseY-sourf[3])/((sourf[1]-sourf[3])-(sourg[1]-sourf[3]));
601     if (percX>100) percX=100;
602     if (percY>100) percY=100;
603     if (percX<0) percX=0;
604     if (percY<0) percY=0;
605     f.buttonAlignX=percX;
606     f.buttonAlignY=percY;
607 }
608
609 function zobraz(cislo,strana)
610 {
611     var f=this.getField("ikona."+strana);
612     var g=this.getField("animtiph."+cislo);
613     f.hidden=false;
614     f.buttonSetIcon(g.buttonGetIcon());
615 }
616
617 this.dirty=false;
618 app.focusRect = false;
619 \end{insDLJS}
620 \fi

```

A cycle is used to create hidden buttons. Each button has associated a page from the file with tooltips as icon. These icons are invoked by JavaScripts defined in `\tooltip` and `\tooltipan` macros.

```

621 \newcount\tooltip@count
622 \ifTooltip@usepdfTeX
623 \newcommand*\TooltipHidden{%
624   \count@=0
625   \@whilenum\count@<\TooltipPages \do{%
626     \tooltip@count=\count@
627     \advance \tooltip@count by 1%
628     \bgroup
629     \immediate\pdfximage
630     page \the\tooltip@count{\TooltipFilename.pdf}%
631     \mbox{\leavevmode
632       \vbox to 0 pt{\vss\hbox to 0 pt{\pdfstartlink user{%
633         /Subtype /Widget
634         /F 6
635         /T (anintiph.\the\tooltip@count)
636         /FT /Btn
637         /Ff 65536
638         /H /N
639         /BS << /W 1 /S /S >>
640         /MK <<
641         /TP 1
642         /I \the\pdflastximage\space 0 R
643         /IF << /SW /A >>
644         >>
645       }%
646       \phantom{\pdfrefximage \pdflastximage}%
647       \pdfendlink\hss}}}%
648     \egroup
649     \advance\count@\@ne}%
650 }
651 \else
652 \let\TooltipHidden\relax
653 \fi

The keywords for the tooltips can be stored in the file \TooltipFilename.tips.
The topics in this file are created by \keytip macro (see the first part of the code).
654 \AtBeginDocument{\IfFileExists{\TooltipFilename.tips}%
655   {\input{\TooltipFilename.tips}
656   \PackageInfo{fancytooltips}{Inputting \TooltipFilename.tips.}}%
657   {\PackageWarning{fancytooltips}{No file \TooltipFilename.tips!
658     Your keywords for tooltips will not work!}}}
659
660 \def\tooltipname#1#2{\expandafter\xdef\csname FancyToolTip@#1\endcsname{#2}}
661
662 \def\FindTipNumber#1{\expandafter\ifx \csname FancyToolTip@#1\endcsname\relax
663   #1\else\csname FancyToolTip@#1\endcsname\fi}
664

```

```
665 \def\checkTipNumber#1{\expandafter\ifx
666   \csname FancyToolTip@#1\endcsname\relax \PackageInfo{fancytooltips}{No
667     framenumber is assigned to keyword #1. I assume that #1 is the
668     number of the frame.}%
669   \fi}
670
671 \end{package}
```