

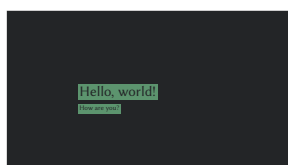
ppt-slides: L^AT_EX Package for Slide Decks à la Power Point*

Yegor Bugayenko
yegor256@gmail.com

2024-01-16, 0.3.1

1 Introduction

This package helps you render slide decks in PowerPoint™ style:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptMiddle}
5 \pptTitle{Hello, world!}{How are you?}
6 \end{pptMiddle}
7 \end{document}
```

It is recommended to use this package together with [clicks](#):



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \usepackage[static]{clicks}
4 \begin{document}
5 We help you make slide decks in \LaTeX:
6 \click{\pptBanner{Making Slides Is Easy}}
7 \click{Just use this package...}
8 \plush{together with 'clicks' package.}
9 They will look similar to what PowerPoint can
10 make, but with the precision of \LaTeX. We
11 recommend using us together with ‘‘clicks.’’
12 \end{document}
```

2 Layout

`\pptToc` First, split your story into chapters:
`\pptChapter`

```
\documentclass{article}
\usepackage{clicks}
\usepackage[template,scheme=light]{ppt-slides}
\begin{document}
```

*The sources are in GitHub at [yegor256/ppt-slides](https://github.com/yegor256/ppt-slides)

```

\pptToc
\plush{\pptChapter{About Me}}
...
\plush{\pptChapter[Idea]{My Idea Is Novel}}
...
\plush{\pptChapter[FAQ]{Discussion \& Questions}}
...
\end{document}

```

\pptToc will render the table of contents in an interactive “clickable” format. Thanks to the use of **crumbs**, there will be a navigation at the top left corner.

\pptSection Then, put sections inside chapters:

```

\begin{document}
\pptToc
\plush{\pptChapter{About Me}}
\plush{\pptSection[Student]{I’m a Student}}
...
\plush{\pptSection[Athlete]{Also, I’m an Athlete}}
...
\plush{\pptChapter[Idea]{My Idea Is Novel}}
\plush{\pptSection{Novelty}}
\plush{\pptSection{Impact}}
\end{document}

```

The \pptChapter and \pptSection commands used together will render nice two-level navigation menu at the top left corner.

\pptLeft You can place the title of the presentation and your name at the bottom of each slide,
 \pptRight on the left and on the right respectively:

```

\documentclass{article}
\usepackage{clicks}
\usepackage[template,scheme=light]{ppt-slides}
\pptLeft{How Did I Spend Last Summer}
\pptRight{Yegor Bugayenko}
\begin{document}
...
\end{document}

```

If you don’t use the \pptLeft or \pptRight commands, nothing will be printed at the bottom.

3 Commands

\pptPic You can add an image to the slide (the first argument is the width of the image in relation to \textwidth, while the second one is the path of it):



```

1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptPic{0.2}{socrates.jpg} \\
5 This is Socrates, a Greek philosopher.
6 \end{document}

```

`\pptHeader` You can add a piece of code to the slide (we recommend using [ffcode](#)):



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \usepackage{ffcode}
4 \begin{document}
5 \pptHeader{This is How You Print to Console:}
6 \begin{ffcode}
7 void main(char** args) {
8     printf("Hello, world!");
9 }
10 \end{ffcode}
11 \end{document}
```

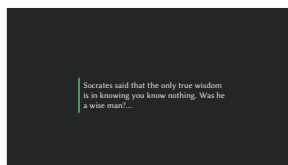
Don't use `\plick` or `\plush`, they won't work with code snippets. The `\pptHeader` command prints a header similar to what the `\pptSection` command prints, but doesn't start a new section.

`\pptQuote` You can put a quote next to the image, by using the `\pptQuote` command:
`\pptBanner`



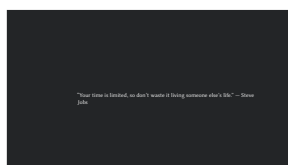
```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptBanner{My Favorite Quote}
5 \pptQuote{socrates.jpg}{The only true wisdom
6 is in knowing you know nothing.}{Socrates}
7 \end{document}
```

`\pptThought` For an interesting thought in the middle of the slide, use `\pptThought`:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptThought{Socrates said that the only true
5 wisdom is in knowing you know nothing. Was he
6 a wise man?...}
7 \end{document}
```

`pptMiddle` You can put your content vertically in the middle of the page, using the `pptMiddle` environment:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptMiddle}
5 ‘‘Your time is limited, so don't waste it
6 living someone else's life.’’ --- Steve Jobs
7 \end{pptMiddle}
8 \end{document}
```

`\pptQR` Sometimes it's convenient to show a QR code to your audience instead of a URL, since it's easier to use — they can scan it:

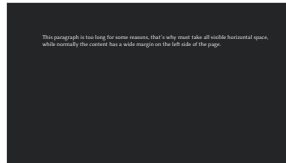


```

1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 Check my blog post by this link: \\
5 \pptQR{https://www.yegor256.com}
6 \end{document}

```

pptWide
pptWideOne



```

1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptWideOne}
5 This paragraph is too long for some reasons,
6 that's why must take all visible horizontal
7 space, while normally the content has a wide
8 margin on the left side of the page.
9 \end{pptWideOne}
10 \end{document}

```

You can also use `\begin{pptWide}{X}`, where `X` is the number of columns to render.

4 Package Options

template You start with a template for your slide deck:

```

\documentclass{article}
\usepackage[template=9x6]{ppt-slides}
\begin{document}
...
\end{document}

```

There is only one template, which is used by default: `9x6`. If you don't specify the name, it will be used. If you don't use the `template` option at all, the default `article` will be rendered, which is not what you want.

scheme You can choose a color scheme for your slides, using the `scheme` option of the package:

```
\usepackage[template,scheme=light]{ppt-slides}
```

There are a few out-of-the-box schemes available: `light`, `dark`, `light-mono`, and `dark-mono`. You can design your own, using `ppt-light.tex` file as an example:

```
\usepackage[template,scheme=/usr/local/my-colors.tex]{ppt-slides}
```

directory You can change the location of the templates and schemes by means of the `directory` package option:

```
\usepackage[directory=/tmp,template=foo]{ppt-slides}
```

nominutes By default, if you render your slide deck in non-static mode (the `static` option for the “clicks” package), there will be minutes tracking in the right top corner of each slide. You can turn this off by using the `nominutes` option of the package:

```
\usepackage[nominutes]{ppt-slides}
```

5 Implementation

First, we parse the package options:

```
1 \RequirePackage{pgfplots}
2 \makeatletter
3 \pgfkeys{
4   /ppt-slides/.is family,
5   /ppt-slides,
6   template/.estore in = \ppt@template,
7   template/.default = 9x6,
8   scheme/.estore in = \ppt@scheme,
9   directory/.estore in = \ppt@directory,
10  directory/.default = ,
11  nominales/.estore in = \ppt@nominales,
12  directory
13 }
14 \ProcessPgfPackageOptions{/ppt-slides}
15 \makeatother
```

Then, we include the template:

```
16 \makeatletter\ifdefined\ppt@template
17   \input{\ppt@directory ppt-templates/ppt-\ppt@template.tex}
18 \else
19   \message{^^Jppt: No template is loaded, because
20     the 'template' option is not specified}
21 \fi\makeatother
```

Then, we include the scheme:

```
22 \makeatletter\ifdefined\ppt@scheme
23   \RequirePackage{xcolor}
24   \input{\ppt@directory ppt-schemes/ppt-\ppt@scheme.tex}
25 \else
26   \message{^^Jppt: No color scheme is loaded, because
27     the 'scheme' option is not specified}
28 \fi\makeatother
```

Then, we include [ifthen](#) in order to enable if/then/else commands:

```
29 \RequirePackage{ifthen}
```

Then, we include [href-ul](#) in order to enable underlined hyperlinks:

```
30 \RequirePackage{href-ul}
```

Then, we include [pagecolor](#) in order to make it possible to change the background color of a page:

```
31 \RequirePackage{pagecolor}
```

Then, we include [varwidth](#) in order to change the width of the page, inside pptWide:

```
32 \RequirePackage{varwidth}
```

Then, we include [qrcode](#) in order to render QR codes:

```
33 \RequirePackage{qrcode}
```

Then, we include [tikz](#) in order to position elements on the page, especially in the Table of Contents:

```
34 \RequirePackage{tikz}
```

Then, we include [tikzpagenodes](#) in order to pin a block in \pptPin:

```
35 \RequirePackage{tikzpagenodes}
```

Then, we include `enumitem` in order to render inline lists in crumbs:

```
36 \RequirePackage[inline]{enumitem}
```

Then, we include `crumbs` in order to show top-page menus:

```
37 \RequirePackage{crumbs}
```

Then, we include `calc` in order to calculate positions of elements more precisely:

```
38 \RequirePackage{calc}
```

Then, we include `tabularx` in order to render `\pptQuote`:

```
39 \RequirePackage{tabularx}
```

Then, we include `seqsplit` in order to split long links:

```
40 \RequirePackage{seqsplit}
```

`\pptMiddle` Then, we define the `\pptMiddle` environment:

```
41 \newenvironment{pptMiddle}
42 {\vspace*{\fill}}
43 {\vspace*{\fill}}
```

`\pptBanner` Then, we define the `\pptBanner` command:

```
44 \newcommand\pptBanner[2][green]{%
45 \par%
46 {\setlength{\fboxsep}{6pt}%
47 \colorbox{#1}{\color{\thePAGEcolor}\large #2}}%
48 \par%
49 }
```

`\pptChapter` Then, we define the `\pptChapter` command:

```
50 \newcommand\pptChapter[2][ ]{%
51 \crumbsection{#1}{#2}%
52 \begin{pptMiddle}%
53 {\large\ttfamily Chapter \#\the\value{section}:}%
54 \newline%
55 \rotatebox{1}{\pptBanner[blue]{\LARGE#2}}%
56 \end{pptMiddle}%
57 }
```

`\pptSection` Then, we define the `\pptSection` command:

```
58 \newcommand\pptSection[2][ ]{%
59 \subcrumbsection{#1}{#2}%
60 \pptHeader{#2}%
61 }
```

`\pptHeader` Then, we define the `\pptHeader` command:

```
62 \newcommand\pptHeader[1]{%
63 \vspace*{-0.75in}\hspace*{-0.5in}%
64 \rotatebox{1}{\pptBanner[orange]{\Large #1}}\vspace*{0.15in}%
65 \par%
66 }
```

`\pptTitle` Then, we define the `\pptTitle` command, which prints the title and subtitle on the first page. The first argument is the title, the second is the subtitle. If the second argument is empty, it won't be printed:

```

67 \newcommand\pptTitle[2]{%
68   {\pptBanner{\Huge #1}}%
69   \def\param{#2}%
70   \ifx\param\empty\else%
71     {\pptBanner{\large #2}}%
72   \fi%
73 }

```

\pptToc Then, we define the \pptToc command:

```

74 \newcommand*\pptToc{}
75 \newcommand\pptToc[1] []{%
76   \renewcommand*\pptToc{}%
77   \print{%
78     \def\param{#1}%
79     \ifx\param\empty\else%
80       \pptBanner{#1}%
81     \fi%
82     \tikz{%
83       \node (z) {%
84         \begin{varwidth}{\textwidth}%
85           \ifx\pptToc\empty\else%
86             \begin{itemize}[label={}]
87               \pptToc
88             \end{itemize}
89           \fi%
90         \end{varwidth}%
91       };%
92       \path [draw=blue,line width=4pt] (z.north west) -- (z.south west);%
93     }%
94   }%
95   \begin{@empty}%
96     \renewcommand\crumb[2]{%
97       \ifx\pptToc\empty\else\ifdefined\click\click[0]\fi\fi
98       \gappto\pptToc{\item ##2}
99     }
100     \crumbs
101   \end{@empty}%
102   \ifdefined\flush\flush[1]\fi%
103 }

```

\pptQuote Then, we define the \pptQuote command, with one optional and three mandatory arguments:

```

104 \newcommand\pptQuote[4] []{%
105   \begin{tabularx}{\columnwidth}{c>{\raggedright\arraybackslash}X}%
106     \raisebox{\dimexpr-\height+\ht\strutbox}{
107       \parbox{0.25\textwidth+2pt}{%
108         \raggedright%
109         \pptPic{0.25}{#2}%
110         \def\person{#1}%
111         \ifx\person\empty\else%
112           \par%
113           \small%
114           \person%
115         \fi%

```

```

116     }%
117 }%
118 &%
119 ‘‘#3’’%
120 \def\param{#4}%
121 \ifx\param\empty\else%
122     \newline\newline%
123     \small%
124     --- \param%
125 \fi%
126 \\\%
127 \end{tabularx}%
128 }

```

\pptPic Then, we define the \pptPic command:

```

129 \newcommand\pptPic[2]{%
130     {%
131         \setlength{\fboxsep}{0pt}%
132         \setlength{\fboxrule}{1pt}%
133         \fcolorbox{gray}{white}{%
134             \includegraphics[width=#1\columnwidth]{#2}%
135         }%
136     }%
137 }

```

\pptPin Then, we define the \pptPin command:

```

138 \newcommand\pptPin[2][right]{%
139     \begin{tikzpicture}[remember picture,overlay]%
140         \ifthenelse{\equal{#1}{left}}{%
141             {%
142                 \node[anchor=north east] at
143                     ($ (current page text area.north west) - (24pt, 18pt)$) {%
144                     \begin{minipage}{0.25\textwidth}%
145                         \raggedleft #2%
146                     \end{minipage}%
147                 };%
148             }%
149         }{%
150             \node[anchor=north east] at (current page text area.north east) {%
151                 \begin{minipage}{0.25\textwidth}%
152                     \raggedright #2%
153                 \end{minipage}%
154             };%
155         }%
156     \end{tikzpicture}%
157 }

```

\pptThought Then, we define the \pptThought command:

```

158 \newcommand\pptThought[1]{%
159     \begin{pptMiddle}%
160         \tikz{%
161             \node [inner sep=18pt] (z) {%
162                 \begin{varwidth}{0.8\textwidth}%
163                     \raggedright\Large #1%

```



```

164         \end{varwidth}%
165     };
166     \path [draw=green,line width=8pt] (z.north west) -- (z.south west);
167 }%
168 \end{pptMiddle}%
169 }

```

`\pptSnippet` Then, we define the `\pptSnippet` command:

```

170 \newcommand\pptSnippet[2][\small]{%
171     \begin{samepage}%
172         #1\verbatiminput{#2}%
173     \end{samepage}%
174 }

```

`\pptQR` Then, we define the `\pptQR` command:

```

175 \newcommand\pptQR[2][2in]{%
176     \tikz{%
177         \node[draw=white]{%
178             \href{#2}{%
179                 \qrcode[height=#1]{#2}%
180             }%
181         }%
182     }%
183 }

```

`\pptPinQR` Then, we define the `\pptPinQR` command:

```

184 \newcommand\pptPinQR[2][right]{%
185     \pptPin[#1]{%
186         \begin{minipage}{1.4in}%
187             \raggedleft%
188             \pptQR[1in]{#2}%
189             \\\[3pt]
190             \ttfamily\fontsize{8pt}{10pt}\selectfont%
191             \seqsplit{#2} \href{#2}{$\rightarrow$}%
192             \\
193         \end{minipage}%
194     }%
195 }

```

`pptWide` Finally, we define the `pptWide` and `pptWideOne` environments:

```

196 \newenvironment{pptWideOne}
197 { \begin{adjustwidth}{-2in}{-1in}}
198 { \end{adjustwidth}}
199 \newenvironment{pptWide}[1]
200 { \begin{pptWideOne}\begin{multicols}{#1}}
201 { \end{multicols}\end{pptWideOne}}
202 \endinput

```

Change History

0.0.1		command turned into pptMiddle environment.	6
	General: First draft.	5	
0.1.4		0.3.0	
	General: The <code>nominutes</code> package option added.	<code>\pptQuote</code> : The <code>\pptQuote</code> command has got an optional parameter, where the name of the person on the photo can be specified.	7
0.1.5			
	General: The default directory fixed. . .		5
0.2.0		<code>\pptThought</code> : The <code>\pptThought</code> command aligns text to the left. . . .	8
	General: We migrated to the <code>ltxdoc</code> format and <code>.dtx</code> file.	5	
0.2.1		0.3.1	
	<code>\pptMiddle</code> : The <code>\pptMiddle</code>	General: The <code>calc</code> package added. . . .	6

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		I	\pptQuote <u>104</u>	
\# 53		\ifdefined 16, 22, 97, 102	\pptSection <u>58</u>	
A		\ifthenelse 140	\pptSnippet <u>170</u>	
\arraybackslash ... 105		\ifx . 70, 79, 85, 97, 111, 121	\pptThought <u>158</u>	
B		\includegraphics .. 134	\pptTitle <u>67</u>	
\begin 52, 84, 86, 95, 105, 139, 144, 151, 159, 162, 171, 186, 197, 200		\input 17, 24	\pptTOC 74, 76, 85, 87, 97, 98	
C		\item 98	\pptToc <u>74</u>	
		L		
		\LARGE 55	\pptWide <u>196</u>	
		\Large 64, 163	\print 77	
		\large 47, 53, 71	\ProcessPgfPackageOptions 14	
D		M		
\click 97		\makeatletter .. 2, 16, 22	\qrcode 179	
\color 47		\makeatother ... 15, 21, 28	R	
\colorbox 47		\message 19, 26	\raggedleft 145, 187	
\columnwidth ... 105, 134		N		\raggedright 105, 108, 152, 163
\crumb 96		\newcommand 44, 50, 58, 62, 67, 74, 75, 104, 129, 138, 158, 170, 175, 184	\raisebox 106	
\crumbsection 51		\newenvironment 41, 196, 199	\renewcommand 76, 96	
\crumbs 100		\newline 54, 122	\RequirePackage 1, 23, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40	
E		\node . 83, 142, 150, 161, 177	\rightarrow 191	
\def 69, 78, 110, 120		P		\rotatebox 55, 64
\dimexpr 106		\par 45, 48, 65, 112	S	
F		\param 69, 70, 78, 79, 120, 121, 124	\selectfont 190	
\empty 70, 79, 85, 97, 111, 121		\parbox 107	\seqsplit 191	
\end 56, 88, 90, 101, 127, 146, 153, 156, 164, 168, 173, 193, 198, 201		\path 92, 166	\setlength .. 46, 131, 132	
\endinput 202		\person 110, 111, 114	\small 113, 123, 170	
\equal 140		\pgfkeys 3	\strutbox 106	
G		\ppt@directory . 9, 17, 24	\subcrumbsection 59	
\fboxrule 132		\ppt@nominutes 11	T	
\fboxsep 46, 131		\ppt@scheme 8, 22, 24	\textwidth 84, 107, 144, 151, 162	
\fcolorbox 133		\ppt@template .. 6, 16, 17	\the 53	
\fill 42, 43		\pptBanner 44, 55, 64, 68, 71, 80	\thepagecolor 47	
\flush 102		\pptChapter 50	\tikz 82, 160, 176	
\fontsize 190		\pptHeader 60, 62	\ttfamily 53, 190	
H		\pptMiddle 41	V	
\height 106		\pptPic 109, 129	\value 53	
\href 178, 191		\pptPin 138, 185	\verbatiminput 172	
\hspace 63		\pptPinQR 184	\vspace 42, 43, 63, 64	
\ht 106		\pptQR 175, 188		
\Huge 68				