

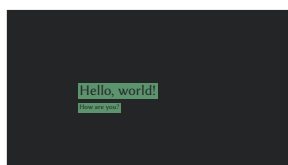
ppt-slides: L^AT_EX Package for Slide Decks à la Power Point*

Yegor Bugayenko
yegor256@gmail.com

?, ?

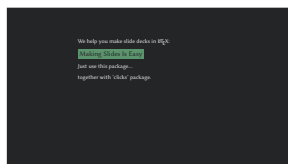
1 Introduction

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



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptMiddle{\pptTitle{Hello, world!}}{How are you?}
5 \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 \plick{\pptBanner{Making Slides Is Easy}}
7 \plick{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}
\pptToc
\plush{\pptChapter{About Me}}
```

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

```

...
\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](#)):



```

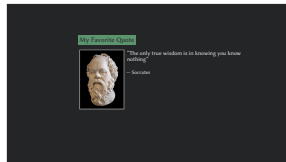
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \usepackage{ffcode}
4 \begin{document}
5 \pptHeader{This is How You Print to Console:}
6 \begin{ffcode}
7 void main(char** args) {
8     printf("Hello, world!");
9 }
10 \end{ffcode}
11 \end{document}

```

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`
`\pptBanner`

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



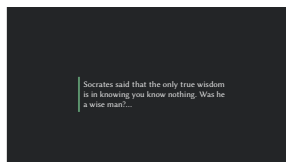
```

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`:



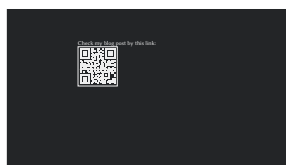
```

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}

```

`\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`

The code will look like this, thanks to [qr](#) package that we use behind the scene. Sometimes you need your slide content to take all visible horizontal space:



```

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{pgfkeys}
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 nominutes/.estore in = \ppt@nominutes,
12 directory
13 }

```

```

14 \ProcessPgfpPackageOptions{/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 a few packages:

```

29 \RequirePackage{ifthen} % for if/then/else in LaTeX
30 \RequirePackage{href-ul} % for nice hyperlinks
31 \RequirePackage{pagecolor} % to change page color
32 \RequirePackage{varwidth} % for pptWide
33 \RequirePackage{qrcode} % to print QR codes
34 \RequirePackage{tikz} % for drawings in TOC
35 \usetikzlibrary{calc} % for positioning of pinned boxes
36 \RequirePackage{tikzpagenodes} % for pptPin
37 \RequirePackage[inline]{enumitem} % for lists in crumbs
38 \RequirePackage{crumbs} % for crumbs

```

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

```

39 \newcommand\pptMiddle[1]{%
40   \vspace*{\fill}#1\vspace*{\fill}%
41 }

```

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

```

42 \newcommand\pptBanner[2][green]{%
43   \par%
44   {\setlength{\fboxsep}{6pt}%
45     \colorbox{#1}{\color{\thePageColor}\large #2}}
46   \par%
47 }

```

\pptChapter Then, we define the \pptChapter command:

```

48 \newcommand\pptChapter[2] [] {%
49   \crumbection[#1]{#2}%
50   \pptMiddle{%
51     {\large\ttfamily Chapter \#\the\value{section}:}%
52     \newline%
53     \rotatebox{1}{\pptBanner[blue]{\LARGE#2}}%
54   }%
55 }

```

\pptSection Then, we define the \pptSection command:

```

56 \newcommand\pptSection[2] [] {%
57   \subcrumbection[#1]{#2}%
58   \pptHeader{#2}%
59 }

```

\pptHeader Then, we define the \pptHeader command:

```

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

```

\pptTitle Then, we define the \pptTitle command:

```

65 \newcommand\pptTitle[2] {%
66   {\pptBanner{\Huge #1}}%
67   {\pptBanner{\large #2}}%
68 }

```

\pptToc Then, we define the \pptToc command:

```

69 \newcommand*\pptTOC{}
70 \newcommand\pptToc[1] [] {%
71   \renewcommand*\pptTOC{}%
72   \print{%
73     \def\param{#1}%
74     \ifx\param\empty\else%
75       \pptBanner{#1}%
76     \fi%
77     \tikz{%
78       \node (z) {%
79         \begin{varwidth}{\textwidth}%
80           \ifx\pptTOC\empty\else%
81             \begin{itemize}[label={}]
82               \pptTOC
83             \end{itemize}
84           \fi%
85         \end{varwidth}%
86       };%
87       \path [draw=blue,line width=4pt] (z.north west) -- (z.south west);%
88     }%
89   }%
90   \begin{@empty}%
91     \renewcommand\crumb[2] {%

```

```

92     \ifx\pptTOC\empty\else\ifdefined\click\click[0]\fi\fi
93     \gappto\pptTOC{\item ##2}
94   }
95   \crumbs
96   \end{@empty}%
97   \ifdefined\flush\flush[1]\fi%
98 }

```

\pptQuote Then, we define the \pptQuote command:

```

99 \RequirePackage{tabularx}
100 \newcommand\pptQuote[3]{%
101   \begin{tabularx}{\columnwidth}{c>{\raggedright\arraybackslash}X}%
102     \raisebox{\dimexpr-\height+\ht\strutbox}{\pptPic{0.25}{#1}}%
103     &%
104     ‘‘#2’’%
105   \def\param{#3}%
106   \ifx\param\empty\else%
107     \newline\newline%
108     \small%
109     --- #3%
110   \fi%
111   \\%
112   \end{tabularx}%
113 }

```

\pptPic Then, we define the \pptPic command:

```

114 \newcommand\pptPic[2]{%
115   {%
116     \setlength{\fboxsep}{0pt}%
117     \setlength{\fboxrule}{1pt}%
118     \fcolorbox{gray}{white}{%
119       \includegraphics[width=#1\columnwidth]{#2}%
120     }%
121   }%
122 }

```

\pptPin Then, we define the \pptPin command:

```

123 \newcommand\pptPin[2][right]{%
124   \begin{tikzpicture}[remember picture,overlay]%
125     \ifthenelse{\equal{#1}{left}}{%
126       {%
127         \node[anchor=north east] at
128           ($(\current page text area.north west) - (24pt, 18pt)$) {%
129           \begin{minipage}{0.25\textwidth}%
130             \raggedleft #2%
131           \end{minipage}%
132         };%
133       }%
134     }{%
135       \node[anchor=north east] at (\current page text area.north east) {%
136       \begin{minipage}{0.25\textwidth}%
137         \raggedright #2%
138       \end{minipage}%
139     };%
140     }%
141   }%
142 }

```

```

140 }%
141 \end{tikzpicture}%
142 }

```

\pptThought Then, we define the \pptThought command:

```

143 \newcommand\pptThought[1]{%
144   \pptMiddle{%
145     \tikz{%
146       \node [inner sep=18pt] (z) {%
147         \begin{varwidth}{0.8\textwidth}
148           \Large
149           #1
150         \end{varwidth}
151       };
152       \path [draw=green,line width=8pt] (z.north west) -- (z.south west);
153     }%
154   }%
155 }

```

\pptSnippet Then, we define the \pptSnippet command:

```

156 \newcommand\pptSnippet[2][\small]{%
157   \begin{samepage}%
158     #1\verbatiminput{#2}%
159   \end{samepage}%
160 }

```

\pptQR Then, we define the \pptQR command:

```

161 \newcommand\pptQR[2][2in]{%
162   \tikz{%
163     \node[draw=white]{%
164       \href{#2}{%
165         \qrcode[height=#1]{#2}%
166       }%
167     }%
168   }%
169 }

```

\pptPinQR Then, we define the \pptPinQR command:

```

170 \RequirePackage{seqsplit}
171 \newcommand\pptPinQR[2][right]{%
172   \pptPin[#1]{%
173     \begin{minipage}{1.4in}%
174       \raggedleft%
175       \pptQR[1in]{#2}%
176       \\[3pt]
177       \ttfamily\fontsize{8pt}{10pt}\selectfont%
178       \seqsplit{#2} \href{#2}{\rightarrow}%
179     \\
180   \end{minipage}%
181   }%
182 }

```

pptWide Finally, we define the pptWide and pptWideOne environments:


```

183 \newenvironment{pptWideOne}
184   {\begin{adjustwidth}{-2in}{-1in}}
185   {\end{adjustwidth}}
186 \newenvironment{pptWide}[1]
187   {\begin{pptWideOne}\begin{multicols}{#1}}
188   {\end{multicols}\end{pptWideOne}}

```

Change History

0.0.1		0.1.5	
General: First draft.	4	General: The default directory fixed. . .	4
0.1.4		0.2.0	
General: The <code>nominutes</code> package		General: We migrated to the <code>ltxdoc</code>	
option added.	4	format and <code>.dtx</code> file.	4

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	<code>\equal</code>	125	L
<code>\#</code>	51		<code>\LARGE</code>
A	F		53
<code>\arraybackslash</code> . . .	<code>\fboxrule</code>	117	<code>\Large</code>
	<code>\fboxsep</code>	44, 116	62, 148
B	<code>\fcolorbox</code>	118	<code>\large</code>
<code>\begin</code>	<code>\fill</code>	40	45, 51, 67
79, 81, 90,	<code>\flush</code>	97	M
101, 124, 129, 136,	<code>\fontsize</code>	177	<code>\makeatletter</code> . .
147, 157, 173, 184, 187			2, 16, 22
C	G		<code>\makeatother</code> . . .
<code>\click</code>	<code>\gappto</code>	93	15, 21, 28
<code>\color</code>			<code>\message</code>
<code>\colorbox</code>	H		19, 26
<code>\columnwidth</code> . . .	<code>\height</code>	102	N
<code>\crumb</code>	<code>\href</code>	164, 178	<code>\newcommand</code>
<code>\crumblection</code>	<code>\hspace</code>	61	39, 42, 48, 56, 60,
<code>\crumbs</code>	<code>\ht</code>	102	65, 69, 70, 100, 114,
D	<code>\Huge</code>	66	123, 143, 156, 161, 171
<code>\def</code>			<code>\newenvironment</code> 183, 186
<code>\dimexpr</code>	I		<code>\newline</code>
E	<code>\ifdefined</code> . .	16, 22, 92, 97	52, 107
<code>\empty</code>	<code>\ifthenelse</code>	125	<code>\node</code>
<code>\end</code>	<code>\ifx</code>	74, 80, 92, 106	78, 127, 135, 146, 163
83, 85, 96,	<code>\includegraphics</code> . .	119	P
112, 131, 138, 141,	<code>\input</code>	17, 24	<code>\par</code>
150, 159, 180, 185, 188	<code>\item</code>	93	43, 46, 63
			<code>\param</code>
			73, 74, 105, 106
			<code>\path</code>
			87, 152
			<code>\pgfkeys</code>
			3
			<code>\ppt@directory</code> . .
			9, 17, 24
			<code>\ppt@nominutes</code>
			11
			<code>\ppt@scheme</code>
			8, 22, 24
			<code>\ppt@template</code> . .
			6, 16, 17

\pptBanner	\ProcessPgfPackageOptions	\setlength ..	44, 116, 117
. 42 , 53 , 62 , 66 , 67 , 75	\small	108, 156
\pptChapter		\strutbox	102
\pptHeader	Q	\subcrumbection	57
\pptMiddle ...	\qrcode		
	165		
\pptPic	R	T	
		\textwidth	79, 129, 136, 147
\pptPin	\raggedleft	\the	51
	123, 172	\thepagecolor	45
\pptPinQR	\raggedright ...	\tikz	77, 145, 162
	170	\ttfamily	51, 177
\pptQR	\raisebox		
	161, 175		
\pptQuote	\renewcommand		
	99		
\pptSection	\RequirePackage		
	56		
\pptSnippet	1, 23,		
	29, 30, 31, 32, 33,	U	
\pptThought	34, 36, 37, 38, 99, 170	\usetikzlibrary	35
	143		
\pptTitle	\rightarrow		
	65		
\pptTOC	\rotatebox	V	
69, 71, 80, 82, 92, 93	53, 62	\value	51
\pptToc		\verbatiminput	158
	S	\vspace	40, 61, 62
\pptWide	\selectfont		
	183		
\print	\seqsplit		
72	178		