

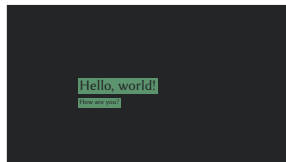
# ppt-slides: L<sup>A</sup>T<sub>E</sub>X Package for Slide Decks à la PowerPoint\*

Yegor Bugayenko  
yegor256@gmail.com

2025/04/17, 0.5.0

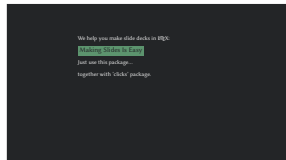
## 1 Introduction

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 \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}
```

---

\*The sources are in GitHub at [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, 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}

```

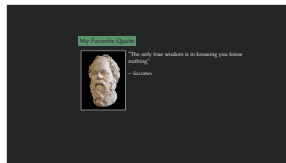
`\pptHeader` You can add a piece of code to the slide (we recommend using [ffcode](#)):



```
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}
```

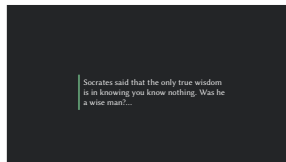
Don't use `\plick` or `\plush`, they won't work with code snippets. The `\pptHeader` command prints a header similar to what the `\pptSection` command prints, but doesn't start a new section.

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



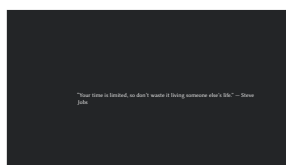
```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptBanner{My Favorite Quote}
5 \pptQuote{socrates.jpg}{The only true wisdom
6 is in knowing you know nothing}{Socrates}
7 \end{document}
```

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



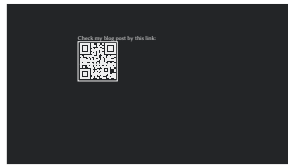
```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptThought{Socrates said that the only true
5 wisdom is in knowing you know nothing. Was he
6 a wise man?...}
7 \end{document}
```

`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}
```

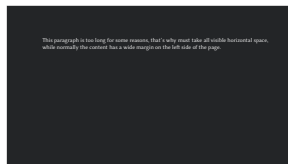
`\pptQR` Sometimes it's convenient to show a QR code to your audience instead of a URL, since it's easier to use — they can scan it:



```
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}
```

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

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}
```

`nocrumbs` By default, the top left corner of each slide has so called crumbs. You can get rid of them with the help of the `nocrumbs` package option:

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

## 5 Implementation

First, we parse the package options:

```
1 \RequirePackage{pgfopts}
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  nomminutes/.estore in = \ppt@nomminutes,
12  nocrumbs/.estore in = \ppt@nocrumbs,
13  directory
14 }
15 \ProcessPgfPackageOptions{/ppt-slides}
16 \makeatother
```

Then, we include the template:

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

Then, we include the scheme:

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

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

```
30 \RequirePackage{ifthen}
```

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

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

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

```
32 \RequirePackage{pagecolor}
```

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

```
33 \RequirePackage{varwidth}
```

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

```
34 \RequirePackage{qrcode}
```

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

```
35 \RequirePackage{tikz}
```

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

```
36 \RequirePackage{tikzpagenodes}
```

Then, we include [enumitem](#) in order to render inline lists in crumbs:

```
37 \PassOptionsToPackage{inline}{enumitem}
```

```
38 \RequirePackage{enumitem}
```

Then, we include [crumbs](#) in order to show top-page menus:

```
39 \RequirePackage{crumbs}
```

Then, we include [calc](#) in order to calculate positions of elements more precisely:

```
40 \RequirePackage{calc}
```

Then, we include [tabularx](#) in order to render `\pptQuote`:

```
41 \RequirePackage{tabularx}
```

Then, we include [seqsplit](#) in order to split long links:

```
42 \RequirePackage{seqsplit}
```

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

```
43 \newenvironment{pptMiddle}
```

```
44   {\vspace*{\fill}}
```

```
45   {\vspace*{\fill}}
```

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

```
46 \newcommand\pptBanner[2][green]{%
```

```
47   \par%
```

```
48   {\setlength{\fboxsep}{6pt}}%
```

```
49   \colorbox{#1}{\color{\thepagecolor}\large #2}}%
```

```
50   \par%
```

```
51 }
```

`\pptChapter` Then, we define the `\pptChapter` and `\pptChapterLabel` commands:

```
52 \newcommand\pptChapterLabel{Chapter}
```

```
53 \newcommand\pptChapter[2] []{%
```

```
54   \crumblection{#1}{#2}}%
```

```
55   \begin{pptMiddle}}%
```

```
56     {\large\ttfamily \pptChapterLabel{} \#\the\value{section}:}}%
```

```
57     \newline%
```

```
58     \rotatebox{1}{\pptBanner[blue]{\LARGE#2}}}}%
```

```
59   \end{pptMiddle}}%
```

```
60 }
```

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

```
61 \newcommand\pptSection[2] []{%
```

```
62   \subcrumblection{#1}{#2}}%
```

```
63   \pptHeader{#2}}%
```

```
64 }
```

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

```
65 \newcommand\pptHeader[1]{%
66   \vspace*{-0.75in}\hspace*{-0.5in}%
67   \rotatebox{1}{\pptBanner[orange]{\Large #1}}\vspace*{0.15in}%
68   \par%
69 }
```

`\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:

```
70 \newcommand\pptTitle[2]{%
71   {\pptBanner{\Huge #1}}%
72   \def\param{#2}%
73   \ifx\param\empty\else%
74     {\pptBanner{\large #2}}%
75   \fi%
76 }
```

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

```
77 \newcommand*\pptToc{}
78 \newcommand\pptToc[1][ ]{%
79   \renewcommand*\pptToc{}%
80   \print{%
81     \def\param{#1}%
82     \ifx\param\empty\else%
83       \pptBanner{#1}%
84     \fi%
85     \tikz{%
86       \node (z) {%
87         \begin{varwidth}{\textwidth}%
88           \ifx\pptToc\empty\else%
89             \begin{itemize}[label={}]
90               \pptToc
91             \end{itemize}
92           \fi%
93         \end{varwidth}%
94       };%
95       \path [draw=blue,line width=4pt] (z.north west) -- (z.south west);%
96     }%
97   }%
98   \begin{@empty}%
99     \renewcommand\crumb[2]{%
100       \ifx\pptToc\empty\else\ifdefined\click\click[0]\fi\fi
101       \gappto\pptToc{\item ##2}
102     }
103     \crumbs
104   \end{@empty}%
105   \ifdefined\flush\flush[1]\fi%
106 }
```

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

```
107 \newcommand\pptQuote[4][ ]{%
```

```

108 \begin{tabularx}{\columnwidth}{c}{\raggedright\arraybackslash}X}%
109 \raisebox{\dimexpr-\height+\ht\strutbox}{
110 \parbox{0.25\textwidth+2pt}{%
111 \raggedright%
112 \pptPic{0.25}{#2}%
113 \def\person{#1}%
114 \ifx\person\empty\else%
115 \par%
116 \small%
117 \person%
118 \fi%
119 }%
120 }%
121 &%
122 ‘‘#3{’’%
123 \def\param{#4}%
124 \ifx\param\empty\else%
125 \par\vspace*{1em}%
126 \small%
127 --- \param%
128 \par%
129 \fi%
130 \\%
131 \end{tabularx}}%
132 }

```

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

```

133 \newcommand\pptPic[2]{%
134 {%
135 \setlength{\fboxsep}{0pt}%
136 \setlength{\fboxrule}{1pt}%
137 \fcolorbox{gray}{white}{%
138 \includegraphics[width=#1\columnwidth]{#2}%
139 }%
140 }%
141 }

```

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

```

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

```



```

158     };%
159   }%
160 \end{tikzpicture}%
161 }

```

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

```

162 \newcommand\pptThought [1]{%
163   \begin{pptMiddle}%
164     \tikz{%
165       \node [inner sep=18pt] (z) {%
166         \begin{varwidth}{0.8\textwidth}%
167           \raggedright\Large #1%
168         \end{varwidth}%
169       };
170       \path [draw=green,line width=8pt] (z.north west) -- (z.south west);
171     }%
172   \end{pptMiddle}%
173 }

```

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

```

174 \newcommand\pptSnippet [2] [\small]{%
175   \begin{samepage}%
176     #1\verbatiminput{#2}%
177   \end{samepage}%
178 }

```

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

```

179 \newcommand\pptQR [2] [2in]{%
180   \tikz{%
181     \node [draw=white]{%
182       \href{#2}{%
183         \qrcode [height=#1]{#2}%
184       }%
185     }%
186   }%
187 }

```

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

```

188 \newcommand\pptPinQR [2] [right]{%
189   \pptPin[#1]{%
190     \begin{minipage}{1.4in}%
191       \raggedleft%
192       \pptQR[1in]{#2}%
193       \\[3pt]
194       \ttfamily\fontsize{8pt}{10pt}\selectfont%
195       \seqsplit{#2} \href{#2}{$\rightarrow$}%
196     \\
197   \end{minipage}%
198   }%
199 }

```

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

```

200 \newenvironment{pptWideOne}

```

```

201 {\begin{adjustwidth}{-2in}{-1in}}
202 {\end{adjustwidth}}
203 \newenvironment{pptWide}[1]
204 {\begin{pptWideOne}\begin{multicols}{#1}}
205 {\end{multicols}\end{pptWideOne}}

206 \endinput

```

## Change History

0.0.1		has got an optional parameter, where the name of the person on the photo can be specified. . . . . 7
0.1.4	General: First draft. . . . . 5	
	General: The <code>nominutes</code> package option added. . . . . 5	<code>\pptThought</code> : The <code>\pptThought</code> command aligns text to the left. . . . 9
0.1.5	General: The default directory fixed. . . 5	0.3.1
0.2.0	General: We migrated to the <code>ltxdoc</code> format and <code>.dtx</code> file. . . . . 5	General: The <code>calc</code> package added. . . . 6
0.2.1	<code>\pptMiddle</code> : The <code>\pptMiddle</code> command turned into <code>pptMiddle</code> environment. . . . . 6	0.4.0
		<code>\pptChapter</code> : The <code>\pptChapterLabel</code> command introduced. . . . . 6
0.3.0	<code>\pptQuote</code> : The <code>\pptQuote</code> command	0.5.0
		General: The <code>nocrumbs</code> package option was introduced. . . . . 5

## 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.

	<b>Symbols</b>		
\#	56	\ifthenelse	144
	<b>A</b>	\ifx	73, 82, 88, 100, 114, 124
\arraybackslash	108	\includegraphics	138
	<b>B</b>	\input	18, 25
\begin	55, 87, 89, 98, 108, 143, 148, 155, 163, 166, 175, 190, 201, 204	\item	101
	<b>C</b>		<b>L</b>
\click	100	\LARGE	58
\color	49	\Large	67, 167
\colorbox	49	\large	49, 56, 74
\columnwidth	108, 138		<b>M</b>
\crumb	99	\makeatletter	2, 17, 23
\crumblection	54	\makeatother	16, 22, 29
\crumbs	103	\message	20, 27
	<b>D</b>		<b>N</b>
\def	72, 81, 113, 123	\newcommand	46, 52, 53, 61, 65, 70, 77, 78, 107, 133, 142, 162, 174, 179, 188
\dimeq	109	\newenvironment	43, 200, 203
	<b>E</b>	\newline	57
\empty	73, 82, 88, 100, 114, 124	\node	86, 146, 154, 165, 181
\end	59, 91, 93, 104, 131, 150, 157, 160, 168, 172, 177, 197, 202, 205		<b>P</b>
\endinput	206	\par	47, 50, 68, 115, 125, 128
\equal	144	\param	72, 73, 81, 82, 123, 124, 127
	<b>F</b>	\parbox	110
\fboxrule	136	\PassOptionsToPackage	37
\fboxsep	48, 135	\path	95, 170
\fcolorbox	137	\person	113, 114, 117
\fill	44, 45	\pgfkeys	3
\flush	105	\ppt@directory	9, 18, 25
\fontsize	194	\ppt@nocrumbs	12
	<b>G</b>	\ppt@nominutes	11
\gappto	101	\ppt@scheme	8, 23, 25
	<b>H</b>	\ppt@template	6, 17, 18
\height	109	\pptBanner	46, 58, 67, 71, 74, 83
\href	182, 195	\pptChapter	52
\hspace	66	\pptChapterLabel	52, 56
\ht	109	\pptHeader	63, 65
\Huge	71	\pptMiddle	43
	<b>I</b>	\pptPic	112, 133
\ifdefined	17, 23, 100, 105	\pptPin	142, 189
		\pptPinQR	188
		\pptQR	179, 192
		\pptQuote	107
		\pptSection	61
		\pptSnippet	174
		\pptThought	162
		\pptTitle	70
		\pptTOC	77, 79, 88, 90, 100, 101
		\pptToc	77
		\pptWide	200
		\print	80
		\ProcessPgfPackageOptions	15
			<b>Q</b>
		\qrcode	183
			<b>R</b>
		\raggedleft	149, 191
		\raggedright	108, 111, 156, 167
		\raisebox	109
		\renewcommand	79, 99
		\RequirePackage	1, 24, 30, 31, 32, 33, 34, 35, 36, 38, 39, 40, 41, 42
		\rightarrow	195
		\rotatebox	58, 67
			<b>S</b>
		\selectfont	194
		\seqsplit	195
		\setlength	48, 135, 136
		\small	116, 126, 174
		\strutbox	109
		\subcrumblection	62
			<b>T</b>
		\textwidth	87, 110, 148, 155, 166
		\the	56
		\thepagecolor	49
		\tikz	85, 164, 180
		\ttfamily	56, 194
			<b>V</b>
		\value	56
		\verbatiminput	176
		\vspace	44, 45, 66, 67, 125