

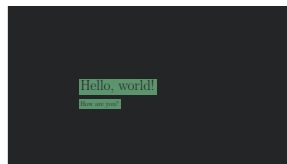
ppt-slides: L^AT_EX Package for Slide Decks à la PowerPoint*

Yegor Bugayenko
yegor256@gmail.com

2026-05-19, 0.7.0

1 Introduction

This package renders slide decks in the 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 that this package be employed in conjunction with [clicks](#):



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \usepackage[static]{clicks}
4 \begin{document}
5 Slide decks may be composed in \LaTeX:
6 \click{\pptBanner{Making Slides Is Easy}}
7 \click{Simply employ this package\ldots}
8 \plush{together with the 'clicks' package.}
9 The result resembles what PowerPoint
10 produces, yet with the precision of \LaTeX.
11 Use in concert with ‘‘clicks’’ is recommended.
12 \end{document}
```

2 Layout

`\pptToc` First, the narrative is divided 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` renders the table of contents in an interactive “clickable” format. Through the use of [crumbs](#), a navigation appears at the upper left corner.

`\pptSection` Sections are then placed within 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 employed in tandem render an elegant two-level navigation menu at the upper left corner.

`\pptLeft` The title of the presentation and the author's name may be placed at the foot
`\pptRight` of each slide, on the left and on the right respectively:

```

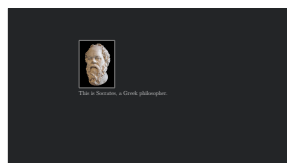
\documentclass{article}
\usepackage{clicks}
\usepackage[template,scheme=light]{ppt-slides}
\pptLeft{How Did I Spend Last Summer}
\pptRight{Yegor Bugayenko}
\begin{document}
...
\end{document}

```

Should neither the `\pptLeft` nor the `\pptRight` command be used, nothing is printed at the foot.

3 Commands

`\pptPic` An image may be added to a slide (the first argument denotes the width of the image relative to `\textwidth`, while the second supplies its path):



```

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` A code fragment may be added to a slide (the use of [ffcode](#) is recommended):



```
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}
```

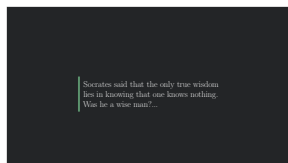
`\plick` or `\plush` should not be used here, as they do not function with code snippets. The `\pptHeader` command prints a header similar to that produced by the `\pptSection` command, yet without commencing a new section.

`\pptQuote` A quotation may be placed beside an image by means of the `\pptQuote` `\pptBanner` command:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptBanner{A Cherished Quote}
5 \pptQuote{socrates.jpg}{The only true wisdom
6 lies in knowing that one knows nothing.}{Socrates}
7 \end{document}
```

`\pptThought` For a noteworthy thought to be placed at the centre of the slide, `\pptThought` is employed:



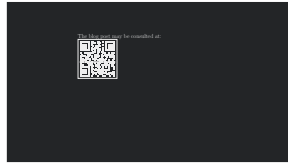
```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptThought{Socrates said that the only true
5 wisdom lies in knowing that one knows nothing.
6 Was he a wise man?...}
7 \end{document}
```

`pptMiddle` Content may be positioned vertically at the centre of the page by means of the `pptMiddle` environment:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptMiddle}
5 ‘‘One’s time is limited; it ought not be
6 wasted living someone else’s life.’’ --- Steve Jobs
7 \end{pptMiddle}
8 \end{document}
```

`\pptQR` On occasion, it is more convenient to present a QR code to the audience in lieu of a URL, since the former is more readily consulted—it may simply be scanned:



```

1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 The blog post may be consulted at: \\
5 \pptQR{https://www.yegor256.com}
6 \end{document}

```

`pptWide`
`pptWideOne`

The code is rendered thus by virtue of the `qrcode` package employed internally. Slide content occasionally needs to occupy the entire visible horizontal width:



```

1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptWideOne}
5 This paragraph happens to be unusually long,
6 and must therefore occupy the entire visible
7 horizontal width, although the content
8 ordinarily bears a generous left margin.
9 \end{pptWideOne}
10 \end{document}

```

The form `\begin{pptWide}{X}` is likewise admissible, where `X` denotes the number of columns to be rendered.

4 Package Options

`template` Each slide deck begins with a template:

```

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

```

Only one template is supplied and is used by default: `9x6`. Should the name be omitted, this template is selected. Should the `template` option itself be omitted entirely, the default `article` is rendered, which is rarely the desired outcome.

`scheme` A colour scheme for the slides may be selected by means of the `scheme` option of the package:

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

Several schemes are available out of the box: `light`, `dark`, `light-mono`, and `dark-mono`. A bespoke scheme may also be devised, taking the `ppt-light.tex` file as a model:

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

`directory` The location of the templates and schemes may be altered by means of the `directory` package option:

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

`nominutes` By default, when the slide deck is rendered in non-static mode (the `static` option of the “clicks” package being absent), a minute tracker appears at the upper right corner of each slide. This may be disabled by means of the `nominutes` option of the package:

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

`nocrumbs` By default, the upper left corner of each slide bears so-called crumbs. These may be removed by means 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  nominutes/.estore in = \ppt@nominutes,
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 `csquotes` in order to use double quotes:

```
34 \RequirePackage{csquotes}
```

Then, we include `qrcode` in order to render QR codes:

```
35 \RequirePackage{qrcode}
```

Then, we include `tikz` in order to position elements on the page, especially in the Table of Contents:

```
36 \RequirePackage{tikz}
```

Then, we include `tikzpagenodes` in order to pin a block in `\pptPin`:

```
37 \RequirePackage{tikzpagenodes}
```

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

```
38 \PassOptionsToPackage{inline}{enumitem}
```

```
39 \RequirePackage{enumitem}
```

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

```
40 \RequirePackage{crumbs}
```

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

```
41 \RequirePackage{calc}
```

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

```
42 \RequirePackage{tabularx}
```

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

```
43 \RequirePackage{seqsplit}
```

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

```
44 \newenvironment{pptMiddle}
```

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

```
46 {\vspace*{\fill}}
```

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

```
47 \newcommand{\pptBanner}[2][green]{%
```

```
48 \par%
```

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

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

```
51 \par%
```

```
52 }
```

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

```
53 \newcommand{\pptChapterLabel}{Chapter}
```

```
54 \newcommand{\pptChapter}[2][ ]{%
```

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

```
56 \begin{pptMiddle}}%
```

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

```

58     \newline%
59     \rotatebox{1}{\pptBanner[blue]{\LARGE#2}}%
60     \end{pptMiddle}%
61 }

```

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

```

62 \newcommand\pptSection[2] [] {%
63     \subcrumblection[#1]{#2}%
64     \pptHeader{#2}%
65 }

```

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

```

66 \newcommand\pptHeader[1] {%
67     \vspace*{-0.75in}\hspace*{-0.5in}%
68     \rotatebox{1}{\pptBanner[orange]{\Large #1}}\vspace*{0.15in}%
69     \par%
70 }

```

`\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:

```

71 \newcommand\pptTitle[2] {%
72     {\pptBanner{\Huge #1}}%
73     \def\param{#2}%
74     \ifx\param\empty\else%
75         {\pptBanner{\large #2}}%
76     \fi%
77 }

```

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

```

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

```

```

101     \ifx\pptTOC\empty\else\ifdefined\click\click[0]\fi\fi
102     \gappto\pptTOC{\item ##2}
103   }
104   \crumbs
105   \end{@empty}}%
106   \ifdefined\flush\flush[1]\fi%
107 }

```

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

```

108 \newcommand\pptQuote[4][]{%
109   \begin{tabularx}{\columnwidth}{c>\raggedright\arraybackslash}X}%
110   \raisebox{\dimexpr-\height+\ht\strutbox}{
111     \parbox{0.25\textwidth+2pt}{%
112       \raggedright%
113       \pptPic{0.25}{#2}%
114       \def\person{#1}%
115       \ifx\person\empty\else%
116         \par%
117         \small%
118         \person%
119       \fi%
120     }%
121   }%
122   &%
123   \enquote{#3{}}%
124   \def\param{#4}%
125   \ifx\param\empty\else%
126     \par\vspace*{1em}%
127     \small%
128     \textendash{} \param%
129     \par%
130   \fi%
131   \\%
132 \end{tabularx}%
133 }

```

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

```

134 \newcommand\pptPic[2]{%
135   {%
136     \setlength{\fboxsep}{0pt}%
137     \setlength{\fboxrule}{1pt}%
138     \fcolorbox{gray}{white}{%
139       \includegraphics[width=#1\columnwidth]{#2}%
140     }%
141   }%
142 }

```

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

```

143 \newcommand\pptPin[2][right]{%
144   \begin{tikzpicture}[remember picture,overlay]%
145     \ifthenelse{\equal{#1}{left}}{%
146       {%
147         \node[anchor=north east] at

```



```

148      ($ (current page text area.north west) - (24pt, 18pt)$) {%
149      \begin{minipage}{0.25\textwidth}%
150      \raggedleft #2%
151      \end{minipage}%
152    };%
153  }%
154  {%
155    \node[anchor=north east] at (current page text area.north east) {%
156      \begin{minipage}{0.25\textwidth}%
157      \raggedright #2%
158      \end{minipage}%
159    };%
160  }%
161  \end{tikzpicture}%
162 }

```

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

```

163 \newcommand\pptThought[1]{%
164   \begin{pptMiddle}%
165     \tikz{%
166       \node [inner sep=18pt] (z) {%
167         \begin{varwidth}{0.8\textwidth}%
168         \raggedright\Large #1%
169         \end{varwidth}%
170       };
171       \path [draw=green,line width=8pt] (z.north west) -- (z.south west);
172     }%
173   \end{pptMiddle}%
174 }

```

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

```

175 \newcommand\pptPunch[1]{%
176   \begin{textblock}{16}[0.5,0.5](8,8)%
177     \centering #1%
178   \end{textblock}%
179 }

```

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

```

180 \newcommand\pptSnippet[2][\small]{%
181   \begin{samepage}%
182     #1\verbatiminput{#2}%
183   \end{samepage}%
184 }

```

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

```

185 \newcommand\pptQR[2][2in]{%
186   \tikz{%
187     \node[draw=white]{%
188       \href{#2}{%
189         \qrcode[height=#1]{#2}%
190       }%
191     }%
192   }%
193 }

```

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

```

194 \newcommand\pptPinQR[2][right]{%
195   \pptPin[#1]{%
196     \begin{minipage}{1.4in}%
197       \raggedleft%
198       \pptQR[1in]{#2}%
199       \\[3pt]
200       \ttfamily\fontsize{8pt}{10pt}\selectfont%
201       \seqsplit{#2} \href{#2}{$\rightarrow$}%
202       \\
203     \end{minipage}%
204   }%
205 }
```

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

```

206 \newenvironment{pptWideOne}
207   {\begin{adjustwidth}{-2in}{-1in}}
208   {\end{adjustwidth}}
209 \newenvironment{pptWide}[1]
210   {\begin{pptWideOne}\begin{multicols}{#1}}
211   {\end{multicols}\end{pptWideOne}}

212 \endinput
```

Change History

0.0.1		the person on the photo can be	
General: First draft.	5	specified.	8
0.1.4		<code>\pptThought</code> : The <code>\pptThought</code>	
General: The <code>nominutes</code> package		command aligns text to the left.	9
option added.	5		
0.1.5		0.3.1	
General: The default directory		General: The <code>calc</code> package added.	6
fixed.	5	0.4.0	
0.2.0		<code>\pptChapter</code> : The	
General: We migrated to the		<code>\pptChapterLabel</code> command	
<code>ltxdoc</code> format and <code>.dtx</code> file.	5	introduced.	6
0.2.1		0.5.0	
<code>\pptMiddle</code> : The <code>\pptMiddle</code>		General: The <code>nocrumbs</code> package	
command turned into		option was introduced.	5
<code>pptMiddle</code> environment.	6	0.7.0	
0.3.0		<code>\pptPunch</code> : The <code>\pptPunch</code>	
<code>\pptQuote</code> : The <code>\pptQuote</code>		command places content right	
command has got an optional		in the middle of the slide, both	
parameter, where the name of		vertically and horizontally.	9

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	H	<code>\ppt@directory</code> 9, 18, 25
<code>\#</code> 57	<code>\height</code> 110	<code>\ppt@nocrumbs</code> 12
A	<code>\href</code> 188, 201	<code>\ppt@nominutes</code> 11
<code>\arraybackslash</code> ... 109	<code>\hspace</code> 67	<code>\ppt@scheme</code> ... 8, 23, 25
B	<code>\ht</code> 110	<code>\ppt@template</code> . 6, 17, 18
<code>\begin</code> 56,	<code>\Huge</code> 72	<code>\pptBanner</code> 47,
88, 90, 99, 109,	I	59, 68, 72, 75, 84
144, 149, 156,	<code>\ifdefined</code> 17, 23, 101, 106	<code>\pptChapter</code> 53
164, 167, 176,	<code>\ifthenelse</code> 145	<code>\pptChapterLabel</code> 53, 57
181, 196, 207, 210	<code>\ifx</code> 74,	<code>\pptHeader</code> 64, 66
C	83, 89, 101, 115, 125	<code>\pptMiddle</code> 44
<code>\centering</code> 177	<code>\includegraphics</code> .. 139	<code>\pptPic</code> 113, 134
<code>\click</code> 101	<code>\input</code> 18, 25	<code>\pptPin</code> 143, 195
<code>\color</code> 50	<code>\item</code> 102	<code>\pptPinQR</code> 194
<code>\colorbox</code> 50	L	<code>\pptPunch</code> 175
<code>\columnwidth</code> .. 109, 139	<code>\LARGE</code> 59	<code>\pptQR</code> 185, 198
<code>\crumb</code> 100	<code>\Large</code> 68, 168	<code>\pptQuote</code> 108
<code>\crumblection</code> 55	<code>\large</code> 50, 57, 75	<code>\pptSection</code> 62
<code>\crumbs</code> 104	M	<code>\pptSnippet</code> 180
D	<code>\makeatletter</code> . 2, 17, 23	<code>\pptThought</code> 163
<code>\def</code> 73, 82, 114, 124	<code>\makeatother</code> . 16, 22, 29	<code>\pptTitle</code> 71
<code>\dimexpr</code> 110	<code>\message</code> 20, 27	<code>\pptTOC</code> 78,
E	N	80, 89, 91, 101, 102
<code>\empty</code> 74,	<code>\newcommand</code> 47,	<code>\pptToc</code> 78
83, 89, 101, 115, 125	53, 54, 62, 66,	<code>\pptWide</code> 206
<code>\end</code> 60,	71, 78, 79, 108,	<code>\print</code> 81
92, 94, 105, 132,	134, 143, 163,	<code>\ProcessPgfpPackageOptions</code>
151, 158, 161,	175, 180, 185, 194 15
169, 173, 178,	<code>\newenvironment</code> ...	Q
183, 203, 208, 211 44, 206, 209	<code>\qrcode</code> 189
<code>\endinput</code> 212	<code>\newline</code> 58	R
<code>\enquote</code> 123	<code>\node</code> 87, 147, 155, 166, 187	<code>\raggedleft</code> ... 150, 197
<code>\equal</code> 145	P	<code>\raggedright</code>
F	<code>\par</code> 48,	. 109, 112, 157, 168
<code>\fboxrule</code> 137	51, 69, 116, 126, 129	<code>\raisebox</code> 110
<code>\fboxsep</code> 49, 136	<code>\param</code> 73, 74,	<code>\renewcommand</code> .. 80, 100
<code>\fcolorbox</code> 138	82, 83, 124, 125, 128	<code>\RequirePackage</code> ...
<code>\fill</code> 45, 46	<code>\parbox</code> 111	.. 1, 24, 30, 31,
<code>\flush</code> 106	<code>\PassOptionsToPackage</code>	32, 33, 34, 35, 36,
<code>\fontsize</code> 200 38	37, 39, 40, 41, 42, 43
G	<code>\path</code> 96, 171	<code>\rightarrow</code> 201
<code>\gappto</code> 102	<code>\person</code> ... 114, 115, 118	<code>\rotatebox</code> 59, 68
	<code>\pgfkeys</code> 3	S
		<code>\selectfont</code> 200
		<code>\seqsplit</code> 201

<code>\setlength</code> .	49, 136, 137	<code>\textwidth</code>	88,		V
<code>\small</code>	117, 127, 180		111, 149, 156, 167	<code>\value</code>	57
<code>\strutbox</code>	110	<code>\the</code>	57	<code>\verbatiminput</code>	182
<code>\subcrumbsection</code> . . .	63	<code>\thepagecolor</code>	50	<code>\vspace</code> 45, 46, 67, 68, 126	
T		<code>\tikz</code>	86, 165, 186		
<code>\textendash</code>	128	<code>\ttfamily</code>	57, 200		