

Modern Beamer Presentations with the **neo** package

v1.0 — 2017/10/01

Contents

1	Introduction	3
2	Getting Started	4
2.1	Installing from GitLab	4
2.2	A Minimal Example	4
2.3	Dependencies	5
2.4	Pandoc	5
3	Customization	6
3.1	Package options	6
3.1.1	Main theme	6
3.1.2	Inner theme	6
3.1.3	Outer theme	7
3.1.4	Color theme	7
3.1.5	Font theme	7
3.2	Color Customization	8
3.3	Font Customization	8
3.3.1	Old style figures	9
3.4	Commands	9
3.4.1	Standout frames	9
4	pgfplots integration	10
4.1	Styles	10

4.2	Paul Tol colors	10
5	Tips & Tricks	10
5.1	Backup Slides	10
6	Known Issues	11
6.1	Title formats	11
6.2	Interactions with other color themes	11
6.3	Notes on second screen	12
6.4	Standout frames with labels	13
6.5	Standout frames with Pandoc	13
7	License	14
8	Implementation	14
8.1	neo parent theme	14
8.1.1	Package dependencies	14
8.1.2	Options	14
8.1.3	Component sub-packages	16
8.1.4	Custom commands	17
8.1.5	Process package options	17
8.2	neo inner theme	17
8.2.1	Package dependencies	18
8.2.2	Options	18
8.2.3	Title page	19
8.2.4	Section page	22
8.2.5	Block environments	24
8.2.6	Lists and floats	27
8.2.7	Footnotes	27
8.2.8	Text and spacing settings	27
8.2.9	Standout frames	28
8.2.10	Process package options	29
8.3	neo outer theme	29
8.3.1	Package dependencies	29
8.3.2	Options	30
8.3.3	Head and footline	32
8.3.4	Frametitle	35
8.3.5	Process package options	38

8.4	neo font theme	38
8.4.1	Package dependencies	39
8.4.2	Load Fira fonts	39
8.4.3	General font definitions	41
8.4.4	Font style options	42
8.4.5	Title format options	42
8.4.6	Process package options	48
8.5	neo color theme	48
8.5.1	Package dependencies	48
8.5.2	Options	49
8.5.3	Base colors	49
8.5.4	Alias colors	50
8.5.5	Base styles	50
8.5.6	Derived colors	51
8.5.7	Process package options	55
8.6	Tol pgfplots theme	55

1 Introduction

Beamer is an awesome way to make presentations with LaTeX, but its theme selection is surprisingly sparse. The stock themes share an aesthetic that can be a little cluttered, while the few distinctive custom themes available are often specialized for a particular corporate or institutional brand.

The goal of **neo** is to provide a simple, modern Beamer theme suitable for anyone to use – it is based on the **metropolis** theme by **Matthias Vogelgesang** and **many other of contributors**. It tries to minimize noise and maximize space for content; the only visual flourish it offers is an (optional) progress bar added to each slide or to the section slides.

By default, **neo** uses **Fira Sans**, a gorgeous typeface commissioned by Mozilla and designed by **Carrois**. For best results, you should use the Fira typeface distributed by this package and use Xe_{La}TeX to typeset your slides. However, **neo** can also be used with other typefaces and L^AT_EX build systems.

2 Getting Started

2.1 Installing from GitLab

If you want to use the cutting-edge development version of **neo**, you can install it manually. Like any L^AT_EX package, this involves four easy steps:

Download the source with a `git clone` of the **neo repository** or as a **zip archive** of the latest development version.

Compile the style files by running `make sty` inside the downloaded directory. (Or run L^AT_EX directly on `source/neotheme.ins.`)

Move the resulting *.sty files to the folder containing your presentation. To use **neo** with many presentations, run `make install` or move the *.sty files to a folder in your T_EX path instead.

Use the theme for your presentation by declaring `\usetheme{neo}` in the preamble of your Beamer document.

neo uses the Make build system to offer the following installation options for advanced users:

`make sty` builds the theme style files.

`make doc` builds this documentation manual.

`make demo` builds a demo presentation to test the features of **neo**.

`make all` builds the theme and manual.

`make clean` removes the files generated by `make all`.

`make install` installs the theme into your local texmf folder.

`make uninstall` removes the theme from your local texmf folder.

2.2 A Minimal Example

The following code shows a minimal example of a Beamer presentation using **neo**.

```
\documentclass{beamer}
\usetheme{neo}          % Use neo theme
\title{A minimal example}
```

```

\date{\today}
\author{Matthias Vogelgesang}
\institute{Centre for Modern Beamer Themes}
\begin{document}
  \maketitle
  \section{First Section}
  \begin{frame}{First Frame}
    Hello, world!
  \end{frame}
\end{document}

```

2.3 Dependencies

neo depends on the **beamer** class and the following standard packages:

- **tikz**
- **etoolbox**
- **ifxetex**
- **pgfopts**
- **calc**
- **ifluatex**

For best results, we recommend installing the fonts **Fira Sans** and **Fira Mono** and compiling with **neo** using X_YL^AT_EX or Lua_TE_X. These are optional dependencies; **neo** is compatible with (e.g.) pdf_LA_TE_X and will fall back to standard fonts if **Fira Sans** or **Fira Mono** is not installed.

The packaged name of **Fira Sans** is **Fira Sans OT** in some Linux distributions; this case is automatically handled by **neo**.

2.4 Pandoc

To use this theme with **Pandoc**-based presentations, you can run the following command

```

$ pandoc -t beamer --latex-engine=xelatex -V theme:neo -
  o output.pdf input.md

```

3 Customization

3.1 Package options

The theme provides a number of options, which can be set using a key=value interface. The primary way to set options is to provide a comma-separated list of option-value pairs when loading **neo** in the preamble:

```
\usetheme[option1=value1, option2=value2, ...]{neo}
```

Options can be changed at any time — even mid-presentation! — with the `\neoset` macro.

```
\neoset{option1=newvalue1, option2=newvalue2, ...}
```

The list of options is structured as shown in the following example.

option key	<i>list of possible values</i> default
	A short description of the option.

3.1.1 Main theme

titleformat	<i>regular, smallcaps, allsmallcaps, allcaps</i> regular
	Changes the format of titles, subtitles, section titles, frame titles, and the text on “standout” frames. The available options produce Regular, SMALLCAPS, ALLSMALLCAPS, or ALLCAPS titles. Please refer to Section 6.1 for known issues with these options.

titleformat plain	<i>regular, smallcaps, allsmallcaps, allcaps</i> regular
	Changes the format of “standout” frames (see <code>titleformat</code> , above).

3.1.2 Inner theme

sectionpage	<i>none, simple, progressbar</i> progressbar
	Adds a slide at the start of each section (simple) with an optional thin progress bar below the section title (progressbar). The none option disables the section page.

subsectionpage *none, simple, progressbar* none

Optionally adds a slide at the start of each subsection. If enabled with the **simple** or **progressbar** options, the style of the **section page** will be updated to match the style of the **subsection page**. Note that section slides and subsection slides can appear consecutively if both are enabled; you may want to use this option together with **sectionpage=none** depending on the section structure of your presentation.

3.1.3 Outer theme

numbering *none, counter, fraction* counter

Controls whether the frame number at the bottom right of each slide is omitted (**none**), shown (**counter**) or displayed as a fraction of the total number of frames (**fraction**).

progressbar *none, head, frametitle, foot* none

Optionally adds a progress bar to the top of each frame (**head**), the bottom of each frame (**foot**), or directly below each frame title (**frametitle**).

3.1.4 Color theme

block *transparent, fill* transparent

Optionally adds a light grey background to block environments like **theorem** and **example**.

background *dark, light* light

Provides the option to have a dark background and light foreground instead of the reverse.

3.1.5 Font theme

titleformat title *regular, smallcaps, allsmallcaps, allcaps* regular

titleformat subtitle Individually controls the format of titles, subtitles, section titles, and frame titles (see **titleformat**, above).

titleformat section

titleformat frame

3.2 Color Customization

The included **neo** color theme is used by default, but its colors can be easily changed to suit your tastes. All of the theme's styles are defined in terms of three beamer colors:

- **normal text** (dark fg, light bg)
- **alerted text** (colored fg, should be visible against dark or light)
- **example text** (colored fg, should be visible against dark or light)

An easy way to customize the theme is to redefine these colors using

```
\setbeamercolor{ ... }{ fg= ... , bg= ... }
```

in your preamble. For greater customization, you can redefine any of the other stock beamer colors. In addition to the stock colors the theme defines a number of **neo** specific colors, which can also be redefined to your liking.

```
\setbeamercolor{progress bar}{ ... }  
\setbeamercolor{title separator}{ ... }  
\setbeamercolor{progress bar in head/foot}{ ... }  
\setbeamercolor{progress bar in section page}{ ... }
```

3.3 Font Customization

The default font for **neo** is **Fira**. This can be easily changed using the standard font selection commands of the **fontspec** package. So if you prefer, for example, the **Ubuntu** font family, just add the following two commands after loading the **neo** theme.

```
\setsansfont{Ubuntu}  
\setmonofont{Ubuntu Mono}
```

If you are expecting to present in a large room or with an underpowered projector, you may want to change the font to a heavier weight of **Fira** to maximize

readability.

```
\setsansfont[BoldFont={Fira Sans SemiBold}]{Fira Sans  
Book}
```

3.3.1 Old style figures

The regular `fontspec` mechanism for changing glyph appearance applies also to this theme. If you want to have old style figures in the text but regular lined figures for math, you could add the following to your preamble:

```
\usefonttheme{professionalfonts} % required for  
mathspec  
\usepackage{mathspec}  
\setsansfont[BoldFont={Fira Sans},  
Numbers={OldStyle}]{Fira Sans Light}  
\setmathsfon(Digits)[Numbers={Lining, Proportional}]{  
Fira Sans Light}
```

3.4 Commands

3.4.1 Standout frames

The **neo** inner theme offers a custom frame format with large, centered text and an inverted background — perfect for focusing attention on single sentence or image. To use it, add the key `standout` to the frame:

```
\begin{frame}[standout]  
Thank you!  
\end{frame}
```

4 pgfplots integration

neo comes with a set of pre-defined pgfplots styles and a color theme based on Paul Tol's color scheme.

4.1 Styles

Pass the following style keys to the axis environment to get the appropriate effect:

`mlineplot` Plot regular line charts with reduced axis frames, less intrusive legend and subdued grid.

`mbarplot` Plot vertical bar charts in a similar way as `mlineplot` but reduce grid usage.

`horizontal mbarplot` Plot horizontal bar charts.

`disable thousands separator` Helper style to remove thousands separator.

4.2 Paul Tol colors

A good presentation uses colors that are distinct from each other as much as possible as well as from black and white, can be discerned item under different lighting and display environments and by color-blind viewers, while matching well together.

In a [technical note](#) for SRON, Paul Tol proposed a palette of colors satisfying these constraints. The sub-package `pgfplotsthemetol` defines palettes for `pgfplots` charts based on Tol's work.

5 Tips & Tricks

5.1 Backup Slides

Speakers will often include extra slides at the end of their presentation to refer to during audience questions. One easy way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

neo will automatically turn off slide numbering and progress bars for slides in the appendix.

6 Known Issues

6.1 Title formats

Be aware that not every font supports small caps, so the `smallcaps` or `allsmallcaps` options may not work if you use a font other than **Fira Sans**. In particular, the Computer Modern sans-serif typeface, which is used when **neo** is compiled with pdfL^AT_EX, does not have a small-caps variant.

The title format options `allsmallcaps` and `allcaps` are quite nice from an aesthetic point of view, but their use of `\MakeLowercase` and `\MakeUppercase` can cause unexpected problems. For example:

- Some commands, like `\`, do not work inside `\MakeLowercase` and `\MakeUppercase`. (See [#125](#))
- Only alphabetic characters are affected by `\MakeLowercase`, so numerals and punctuation remain at full height. This can spoil some of the aesthetic benefits of `allsmallcaps`. (See [#33](#))
- `\MakeLowercase` and `\MakeUppercase` apply to math mode and `\scshape` does not. This can easily introduce mathematical errors that are hard to catch.
- It is impossible to typeset symbols which are encoded as uppercase letters in a different font. In particular, `\mathbb` and `\mathcal` letters will be replaced by other math glyphs. (See [#153](#))

The `allsmallcaps` and `allcaps` options are safe to use if your titles contain only alphabetic characters and do not require the expansion of any macros.

6.2 Interactions with other color themes

neo can be used along with any other Beamer color theme, such as **crane** or **seahorse**. If you wish to do this, it is usually best to include the **neo** subpackages individually so the **neo** color theme is never loaded. This will prevent conflicts between the **neo** color theme and your preferred theme.

For example, overriding the color theme as follows may not work as expected because `\usetheme{neo}` loads the **neo** color theme, which defines a relationship between the frametitle background and the primary palette of the theme. Since **seahorse** assumes a different relationship between its palettes, the result is a grey, rather than periwinkle, frametitle background.

```
\usetheme{neo}
\usecolortheme{seahorse}
```

The correct colors are chosen if the **neo** outer, inner, and font themes are loaded separately:

```
\useoutertheme{neo}
\useinnertheme{neo}
\usefonttheme{neo}
\usecolortheme{seahorse}    % or your preferred color
                             theme
```

Please note that **neo** may not use all the colors defined in your favourite Beamer color theme. In particular, **neo** does not set a background color for the title; this will cause issues when using color themes like **whale** which set a white foreground for the title.

6.3 Notes on second screen

If you use the [show notes on second screen] option built in to Beamer and compile with X_YL^AT_EX, text on slides following the first section slide may be rendered in white instead of the regular colour. This is due to a bug in Beamer or X_YL^AT_EX itself. You can work around it either by compiling with Lua_TE_X or by adding the following code to your preamble to reset the text color on each slide.

```
\makeatletter
\def\beamer@framenotesbegin{% at beginning of slide
    \usebeamercolor[fg]{normal text}
    \gdef\beamer@noteitems{}%
    \gdef\beamer@notes{}%
}
\makeatother
```

6.4 Standout frames with labels

Because the `standout` frame option creates a group to restrict the colour change to a single slide, labels defined after calling `standout` will stay local to the group. In other words, the following may result in a “label undefined” error.

```
\begin{frame}[standout, label=conclusion]{Conclusion}
  Awesome slide
\end{frame}
```

To fix this problem, change the order of the keys in the frame.

```
\begin{frame}[label=conclusion, standout]{Conclusion}
  Awesome slide
\end{frame}
```

This error can be unwittingly triggered if you export your slides from Emacs Org mode, which automatically adds labels after frame options. Alex Branham [offers](#) the following solution for Org mode users, using `org-set-property`.

```
* Start of a frame
:PROPERTIES:
:BEAMER_opt: label=conclusion,standout
:END:
```

6.5 Standout frames with Pandoc

With Pandoc versions prior 1.17.2 it was not possible to create standout frames because Pandoc only supported a specific list of frame attributes thus ignoring additional attributes such as `{.standout}`.

7 License

neo is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#). This means that if you change the theme and re-distribute it, you must retain the copyright notice header and license it under the same CC-BY-SA license. This does not affect any presentations that you create with the theme.

8 Implementation

8.1 neo parent theme

The primary job of this package is to load the component sub-packages of the **neo** theme and route the theme options accordingly. It also provides some custom commands and environments for the user.

8.1.1 Package dependencies

```
1 \RequirePackage{etoolbox}
2 \RequirePackage{pgfpages}
3 \RequirePackage{pgfopts}
```

8.1.2 Options

Most options are passed off to the component sub-packages.

```
4 \pgfkeys{/neo/.cd,
5   .search also={
6     /neo/inner,
7     /neo/outer,
8     /neo/color,
9     /neo/font,
10  }
11 }
```

`titleformat plain` Controls the formatting of the text on standout “plain” frames.

```
12 \pgfkeys{
13   /neo/titleformat plain/.cd,
14   .is choice,
15   regular/.code={%
16     \let\neo@plaintitleformat\@empty%
```

```

17     \setbeamerfont{standout}{shape=\normalfont}%
18 },
19 smallcaps/.code={%
20     \let\neo@plaintitleformat\@empty%
21     \setbeamerfont{standout}{shape=\scshape}%
22 },
23 allsmallcaps/.code={%
24     \let\neo@plaintitleformat\MakeLowercase%
25     \setbeamerfont{standout}{shape=\scshape}%
26     \PackageWarning{beamerthemeneo}{%
27         Be aware that titleformat plain=allsmallcaps can lead to problems%
28     }
29 },
30 allcaps/.code={%
31     \let\neo@plaintitleformat\MakeUppercase%
32     \setbeamerfont{standout}{shape=\normalfont}%
33     \PackageWarning{beamerthemeneo}{%
34         Be aware that titleformat plain=allcaps can lead to problems%
35     }
36 },
37 }

```

titleformat Sets a standard format for titles, subtitles, section titles, frame titles, and the text on standout “plain” frames.

```

38 \pgfkeys{
39   /neo/titleformat/.code=\pgfkeysalso{
40     font/titleformat title=#1,
41     font/titleformat subtitle=#1,
42     font/titleformat section=#1,
43     font/titleformat frame=#1,
44     titleformat plain=#1,
45   }
46 }

```

Shortcut option names as aliases to the corresponding key=value options.

```

47 \pgfkeys{/neo/.cd,
48   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
49   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
50   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},

```

```

51 darkcolors/.code=\pgfkeysalso{color/background=dark},
52 blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
53 light/.code=\pgfkeysalso{font/style=light},
54 book/.code=\pgfkeysalso{font/style=book},
55 regular/.code=\pgfkeysalso{font/style=regular},
56 }

```

Set default values for options.

```

57 \newcommand{\neo@setdefaults}{
58   \pgfkeys{/neo/.cd,
59     titleformat plain=regular,
60   }
61 }

```

To avoid generating externalized figures of the progressbar we have to disable them with “tikzexternalenable” and “tikzexternaldisable”. However, if the “external” library is not loaded we would get undefined control sequence problems, hence we define them as no-ops if they are not defined yet.

```

62 \providecommand{\tikzexternalenable}{}
63 \providecommand{\tikzexternaldisable}{}

```

8.1.3 Component sub-packages

Having processed the options, we can now load the component sub-packages of the theme.

```

64 \useinnertheme{neo}
65 \useoutertheme{neo}
66 \usecolortheme{neo}
67 \usefonttheme{neo}

```

The `tol` theme for `pgfplots` is only loaded if `pgfplots` is used, `pdfpcnotes` always

```

68 \AtEndPreamble{%
69   \RequirePackage{pdfpcnotes}
70   \@ifpackageloaded{pgfplots}{%
71     \RequirePackage{pgfplotsthemetol}
72   }{}
73 }

```


8.1.4 Custom commands

The parent theme defines custom commands as their proper usage may depend on multiple sub-packages.

`\neoset` Allows the user to change options midway through a presentation.

```
74 \newcommand{\neoset}[1]{\pgfkeys{/neo/.cd,#1}}
```

`\plain` Creates a plain frame with dark background, suitable for displaying images or a few words. The format of the text can be set with the `titleformat plain` option.

```
75 \def\neo@plaintitleformat#1{#1}
76 \newcommand{\plain}[2][]{%
77   \PackageWarning{beamerthemeneo}{%
78     The syntax ‘\plain’ may be deprecated in a future version of neo.
79     Please use a frame with [standout] instead.
80   }
81   \begin{frame}[standout]{#1}
82     \neo@plaintitleformat{#2}
83   \end{frame}
84 }
```

`\mreducelistspacing`

```
85 \newcommand{\mreducelistspacing}{\vspace{-\topsep}}
```

8.1.5 Process package options

```
86 \neo@setdefaults
87 \ProcessPgfOptions{/neo}
```

8.2 neo inner theme

A `beamer` inner theme dictates the style of the frame elements traditionally set in the “body” of each slide. These include:

- title, part, and section pages;
- itemize, enumerate, and description environments;
- block environments including theorems and proofs;
- figures and tables; and
- footnotes and plain text.

8.2.1 Package dependencies

```
88 \RequirePackage{etoolbox}
89 \RequirePackage{keyval}
90 \RequirePackage{calc}
91 \RequirePackage{pgfopts}
92 \RequirePackage{pgfpages}
93 \RequirePackage{tikz}
```

8.2.2 Options

sectionpage Optionally add a slide marking the beginning of each section.

```
94 \pgfkeys{
95   /neo/inner/sectionpage/.cd,
96   .is choice,
97   none/.code=\neo@disablesectionpage,
98   simple/.code={\neo@enablesectionpage
99                 \setbeamertemplate{section page}[simple]},
100  progressbar/.code={\neo@enablesectionpage
101                     \setbeamertemplate{section page}[progressbar]},
102 }
```

subsectionpage Optionally add a slide marking the beginning of each subsection.

```
103 \pgfkeys{
104   /neo/inner/subsectionpage/.cd,
105   .is choice,
106   none/.code=\neo@disablesubsectionpage,
107   simple/.code={\neo@enablesubsectionpage
108                 \setbeamertemplate{section page}[simple]},
109  progressbar/.code={\neo@enablesubsectionpage
110                     \setbeamertemplate{section page}[progressbar]},
111 }
```

\neo@inner@setdefaults Set default values for inner theme options.

```
112 \newcommand{\neo@inner@setdefaults}{
113   \pgfkeys{/neo/inner/.cd,
114     sectionpage=progressbar,
115     subsectionpage=none
116   }
117 }
```

8.2.3 Title page

`title page` Template for the title page. Each element is only typset if it is defined by the user. If `\subtitle` is empty, for example, it won't leave a blank space on the title slide.

```
118 \setbeamertemplate{title page}{
119   \begin{minipage}[b][0.95\paperheight]{\textwidth}
120     \vfill%
121     \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
122     \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
123     \usebeamertemplate*{title separator}
```

Beamer's definition of `\insertauthor` is always nonempty, so we have to test another macro initialized by `\author{...}` to see if the user has defined an author. This solution was suggested by Enrico Gregorio in an answer to [this Stack Exchange question](#).

```
124   \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
125   \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
126   \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
127   \vfill
128   \begin{minipage}[b][0.25\paperheight][t]{\textwidth}
129 % The lower part of the title page background contains a white area which
130 % covers this whole minipage. Thus switch the text color back to normal
131     \neo@colors@light%
132     \usebeamercolor[fg]{normal text}%
133     \ifx\inserttitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
134   \end{minipage}
135 \end{minipage}
136 }
```

Normal people should use `\maketitle` or `\titlepage` instead of using the `title page` beamer template directly. Beamer already defines these macros, but we patch them here to make the title page `[plain]` by default, remove `\@thanks`, and ensure the title frame number doesn't count. In addition, `\maketitle` used outside of a frame will load a predefined background image, which can be changed using the optional argument: `\maketitle{extern}` for `images/titlepage-extern.pdf`

`\maketitle` Inserts the title frame, or causes the current frame to use the `title page` template.

```
\titlepage
137 \renewcommand{\maketitle}[1][extern]{%
138   \ifbeamer@inframe
139     \titlepage
140   \else
141     {
142       \def\backgroundimg{images/titlepage-#1}
143       \ifdimcomp{\beamer@paperwidth}{=}{16.00cm}{\ifdimcomp{\beamer@paperheight}{=}{9.00cm}{%
144         % 16:9
145         \def\backgroundimg{images/titlepage-#1-highres-169}
146       }{}{}}
147       \usebackgroundtemplate{\includegraphics[width=\paperwidth]{\backgroundimg}}
148       \frame[plain,noframenumbering]{
149         \neo@colors@dark
150         \setbeamercolor{title separator}{
151           fg=black!20,
152           bg=normal text.fg
153         }
154         \titlepage
155       }
156     }
157   \fi
158 }
159 \def\titlepage{%
160   \usebeamertemplate{title page}
161 }
```

`title graphic` Set the title graphic in a zero-height box, so it doesn't change the position of other elements.

```
162 \setbeamertemplate{title graphic}{
163   \vbox to 0pt {
164     \vspace*{2em}
165     \inserttitlegraphic%
166   }%
167   \nointerlineskip%
168 }
```

`title` Set the title on the title page.

```

169 \setbeamertemplate{title}{
170   \raggedright%
171   \linespread{1.0}%
172   \inserttitle%
173   \par%
174   \vspace*{0.5em}
175 }

```

subtitle Set the subtitle on the title page.

```

176 \setbeamertemplate{subtitle}{
177   \raggedright%
178   \insertsubtitle%
179   \par%
180   \vspace*{0.5em}
181 }

```

title separator Template to set the title graphic in a zero-height box. (It won't change the position of other elements.)

```

182 \newlength{\neo@titleseparator@linewidth}
183 \setlength{\neo@titleseparator@linewidth}{0.4pt}
184 \setbeamertemplate{title separator}{
185   \tikzexternaldisable%
186   \begin{tikzpicture}
187     \fill[fg] (0,0) rectangle (\textwidth, \neo@titleseparator@linewidth);
188   \end{tikzpicture}%
189   \tikzexternalenable%
190   \par%
191 }

```

author Set the author on the title page.

```

192 \setbeamertemplate{author}{
193   \vspace*{2em}
194   \insertauthor%
195   \par%
196   \vspace*{0.25em}
197 }

```

date Set the date on the title page.

```
198 \setbeamertemplate{date}{
199   \insertdate%
200   \par%
201 }
```

institute Set the institute on the title page.

```
202 \setbeamertemplate{institute}{
203   \vspace*{3mm}
204   \insertinstitute%
205   \par%
206 }
```

8.2.4 Section page

section page Template for the section title slide at the beginning of each section.

```
207 \defbeamertemplate{section page}{simple}{
208   \begin{center}
209     \usebeamercolor[fg]{section title}
210     \usebeamerfont{section title}
211     \insertsectionhead\par
212     \ifx\insertsubsectionhead\@empty\else
213       \usebeamercolor[fg]{subsection title}
214       \usebeamerfont{subsection title}
215       \insertsubsectionhead
216     \fi
217   \end{center}
218 }
219 \defbeamertemplate{section page}{progressbar}{
220   \centering
221   \begin{minipage}{22em}
222     \raggedright
223     \usebeamercolor[fg]{section title}
224     \usebeamerfont{section title}
225     \insertsectionhead\[-1ex]
226     \usebeamertemplate*{progress bar in section page}
227     \par
228     \ifx\insertsubsectionhead\@empty\else%
```

```

229     \usebeamercolor[fg]{subsection title}%
230     \usebeamerfont{subsection title}%
231     \insertsubsectionhead
232   \fi
233 \end{minipage}
234 \par
235 \vspace{\baselineskip}
236 }
237 \newcommand{\neo@disablesectionpage}{
238   \AtBeginSection{
239     % intentionally empty
240   }
241 }
242 \newcommand{\neo@enablesectionpage}{
243   \AtBeginSection{
244     \ifbeamer@inframe
245       \sectionpage
246     \else
247       \frame[plain,c,noframenumbering]{\sectionpage}
248     \fi
249   }
250 }

```

subsection page Template for the subsection title slide that can optionally be added to at the beginning of each subsection.

```

251 \setbeamertemplate{subsection page}{%
252   \usebeamertemplate*{section page}
253 }
254 \newcommand{\neo@disablesubsectionpage}{
255   \AtBeginSubsection{
256     % intentionally empty
257   }
258 }
259 \newcommand{\neo@enablesubsectionpage}{
260   \AtBeginSubsection{
261     \ifbeamer@inframe
262       \subsectionpage
263     \else
264       \frame[plain,c,noframenumbering]{\subsectionpage}

```

```

265     \fi
266   }
267 }

```

`progress bar in section page` Template for the progress bar displayed by default on the section page. This code is duplicated in large part in the outer theme's template `progress bar in head/foot`.

```

268 \newlength{\neo@progressonsectionpage}
269 \newlength{\neo@progressonsectionpage@linewidth}
270 \setlength{\neo@progressonsectionpage@linewidth}{0.4pt}
271 \setbeamertemplate{progress bar in section page}{
272   \setlength{\neo@progressonsectionpage}{%
273     \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}}%
274   }%
275   \tikzexternaldisable%
276   \begin{tikzpicture}
277     \fill[bg] (0,0) rectangle (\textwidth, \neo@progressonsectionpage@linewidth);
278     \fill[fg] (0,0) rectangle (\neo@progressonsectionpage, \neo@progressonsectionpage@linewidth)
279   \end{tikzpicture}%
280   \tikzexternalenable%
281 }

```

The above code assumes that `\insertframenumber` is less than or equal to `\inserttotalframenumber`. However, this is not true on the first compile; in the absence of an `.aux` file, `\inserttotalframenumber` defaults to 1. This behaviour could cause fatal errors for long presentations, as `\neo@progressonsectionpage` would exceed TeX's maximum length (16383.99999pt, roughly 5.75 metres or 18.9 feet). To avoid this, we increase the default value for `\inserttotalframenumber`; presentations with over 4000 slides will still break on first compile, but users in that situation likely have deeper problems to solve.

```

282 \def\inserttotalframenumber{100}

```

8.2.5 Block environments

`block` The three different block environments differ only in their colours. Rather than
`block alerted` repeat the essentially the same template three times, we use the auxiliary macro
`block example` `\neo@block` to define all three templates.

```

283 \newlength{\neo@blocksep}
284 \newlength{\neo@blockadjust}

```



```

285 \setlength{\neo@blocksep}{0.75ex}
286 \setlength{\neo@blockadjust}{0.25ex}
287 \providecommand{\neo@strut}{%
288   \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}}%
289 }
290 \newcommand{\neo@block}[1]{
291   \par\vskip\medskipamount%
292   \setlength{\parskip}{0pt}

```

If a background color is defined for the block title or body, we need to add a little bit of padding to the corresponding box. Ideally, this would be accomplished by setting `colsep=0.75ex`, which is intended to add “color separation space” only when the box has a colored background. Unfortunately, `colsep` also adds this separation if the background color is inherited, even if the inherited color is actually empty. (The technical reason for this boils down to the fact that the `\ifx` directive does not expand macros.)

To achieve the correct spacing for `alertblocks` and `exampleblocks` as well as for normal blocks, we have to begin the `beamercolorbox` differently based on whether `block title` has an empty background.

If the `block title` background is empty, or the user has explicitly removed the background from (e.g.) `block title alerted`, we just need to set a `rightskip` for a nice ragged-right block title.

```

293 \ifbeamercoloreempty[bg]{block title#1}{%
294   \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}}{%
295   \ifbeamercoloreempty[bg]{block title}{%
296     \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}%
297   }%
298 %   \end{macrocode}
299 %
300 %   Otherwise, if the |block title| has a background, we set the padding based
301 %   on |\neo@blockskip|. However, we have to visually compensate for
302 %   the |\neo@strut| added to the block title (see below) by
303 %   subtracting |\neo@blockadjust| from the top and bottom padding.
304 %
305 %   \begin{macrocode}
306 {%
307   \begin{beamercolorbox}[
308     sep=\dimexpr\neo@blocksep-\neo@blockadjust\relax,

```

```

309     leftskip=\neo@blockadjust,
310     rightskip=\dimexpr\neo@blockadjust plus 4em\relax
311   ]{block title#1}%
312 }}%
313 %   \end{macrocode}
314 %
315 %   We can now set the contents of the |block title|. The zero-width but
316 %   positive-height box |\neo@strut| ensures that the block title box
317 %   has a consistent height, even if it lacks punctuation, ascenders, or
318 %   descenders.
319 %
320 %   \begin{macrocode}
321     \usebeamerfont*{block title#1}%
322     \neo@strut%
323     \insertblocktitle%
324     \neo@strut%
325   \end{beamercolorbox}%
326 %   \end{macrocode}
327 %
328 %   Next, we typeset the |block body|. This the code is similar to, but simpler
329 %   than, the |block title| code since we don't need to adjust for any struts.
330 %
331 %   \begin{macrocode}
332   \nointerlineskip%
333   \ifbeamercoloreempty[bg]{block body#1}{%
334     \begin{beamercolorbox}[vmode]{block body#1}{%
335       \ifbeamercoloreempty[bg]{block body}{%
336         \begin{beamercolorbox}[vmode]{block body#1}%
337       }{%
338         \begin{beamercolorbox}[sep=\neo@blocksep, vmode]{block body#1}%
339         \vspace{-\neo@parskip}
340       }%
341       \usebeamerfont{block body#1}%
342       \setlength{\parskip}{\neo@parskip}%
343     }

```

This concludes the auxiliary macro `\neo@block`. Finally, we define the block beamer templates using this macro.

```

344 \setbeamertemplate{block begin}{\neo@block{}}

```

```

345 \setbeamertemplate{block alerted begin}{\neo@block{ alerted}}
346 \setbeamertemplate{block example begin}{\neo@block{ example}}
347 \setbeamertemplate{block end}{\end{beamercolorbox}\vspace*{0.2ex}}
348 \setbeamertemplate{block alerted end}{\end{beamercolorbox}\vspace*{0.2ex}}
349 \setbeamertemplate{block example end}{\end{beamercolorbox}\vspace*{0.2ex}}

```

8.2.6 Lists and floats

```

350 \setbeamertemplate{itemize items}{\raise1pt\hbox{\vrule width 0.8ex height 0.8ex}}
351 \setbeamertemplate{itemize subitem}{\raise1pt\hbox{\vrule width 0.5ex height 0.5ex}}
352 \setbeamertemplate{itemize subsubitem}{\raise.5ex\hbox{\vrule width 1ex height 0.2ex}}
353 \defbeamertemplate{description item}{align left}{\insertdescriptionitem\hfill}
354 \setbeamertemplate{caption label separator}{: }
355 \setbeamertemplate{caption}[numbered]

```

8.2.7 Footnotes

```

356 \setbeamertemplate{footnote}{%
357   \parindent 0em\noindent%
358   \raggedright
359   \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\hangindent=0.8em\insertfoot.
360 }

```

8.2.8 Text and spacing settings

```

361 \newlength{\neo@parskip}
362 \setlength{\neo@parskip}{0.5em}
363 \setlength{\parskip}{\neo@parskip}
364 \linespread{1.15}

```

By default, Beamer frames offer the `c` option to *almost* vertically center the text, but the placement is a little too high. To fix this, we redefine the `c` option to equalize `\beamer@frametopskip` and `\beamer@framebottomskip`. This solution was suggested by Enrico Gregorio in an answer to [this Stack Exchange question](#).

```

365 \define@key{beamerframe}{c}[true]{% centered
366   \beamer@frametopskip=0pt plus 1fill\relax%
367   \beamer@framebottomskip=0pt plus 1fill\relax%
368   \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
369   \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
370   \def\beamer@initfirstlineunskip{}}%
371 }

```

8.2.9 Standout frames

neo offers a custom frame format with large, centered text and an inverted background. To use it, add the key **standout** to the frame: `\begin{frame}[standout] ... \end{frame}`.

standout Optional arguments to Beamer's frames are implemented using `\define@key` from the **keyval** package, which will execute code when the defined option is called. For the **standout** option, we begin a group, change the colors and fonts, use a plain slide, and set a alignment.

```
372 \providebool{neo@standout}
373 \define@key{beamerframe}{standout}[true]{%
374   \booltrue{neo@standout}
375   \begingroup
376     \setkeys{beamerframe}{c,plain}
377     \ifbeamercoloreempty{bg}{palette primary}{
378       \setbeamercolor{background canvas}{
379         use=palette primary,
380         bg=-palette primary.fg
381       }
382     }{
383       \setbeamercolor{background canvas}{
384         use=palette primary,
385         bg=palette primary.bg
386       }
387     }
388     \setbeamercolor{local structure}{
389       fg=palette primary.fg
390     }
391     \usebeamercolor[fg]{palette primary}
392     \makeatletter
393     \def\beamer@framenotesbegin{% at beginning of slide
394       \usebeamercolor[fg]{palette primary}
395       \gdef\beamer@noteitems{}%
396       \gdef\beamer@notes{}%
397     }
398     \makeatother
399 }
```

Then we just have to close the group after the standout slide is finished in order to restore the colours and fonts for the rest of the presentation. Unfortunately,

we cannot use or this (see <http://tex.stackexchange.com/questions/226319/>). Instead, we prepend the `\endgroup` to `\beamer@reseteecodes`, which is run exactly once at the end of each slide.

```
400 \pretocmd{\beamer@reseteecodes}{%
401   \ifbool{neo@standout}{
402     \endgroup
403     \boolfalse{neo@standout}
404   }{}
405 }{}{}
```

We set the fonts and the alignment on the inner content, in such a way that the speaker's note layout isn't affected by the custom formatting.

```
406 \AtBeginEnvironment{beamer@frameslide}{
407   \makeatletter
408   \usebeamercolor[fg]{normal text}
409   \gdef\beamer@noteitems{%
410     \gdef\beamer@notes{%
411       \makeatother
412       \ifbool{neo@standout}{
413         \centering
414         \usebeamerfont{standout}
415       }{}
416     }
417   }
```

8.2.10 Process package options

```
417 \neo@inner@setdefaults
418 \ProcessPgfPackageOptions{/neo/inner}
```

8.3 neo outer theme

A `beamer` outer theme dictates the style of the frame elements traditionally set outside the body of each slide: the head, footline, and frame title.

8.3.1 Package dependencies

```
419 \RequirePackage{etoolbox}
420 \RequirePackage{calc}
421 \RequirePackage{pgfpages}
```

```

422 \RequirePackage{pgfopts}
423 \RequirePackage{tikz}
424 \usetikzlibrary{fit}

```

8.3.2 Options

icon Adds an icon to the frametitle on each slide.

```

425 \pgfkeys{
426   /neo/outer/frametitle icon/.cd,
427   .is choice,
428   none/.code=\setbeamertemplate{frametitle icon}[none],
429   i4/.code=\setbeamertemplate{frametitle icon}[i4],
430   fau/.code=\setbeamertemplate{frametitle icon}[fau],
431 }

```

numbering Adds slide numbers to the bottom right of each slide.

```

432 \pgfkeys{
433   /neo/outer/numbering/.cd,
434   .is choice,
435   none/.code=\setbeamertemplate{frame numbering}[none],
436   counter/.code=\setbeamertemplate{frame numbering}[counter],
437   fraction/.code=\setbeamertemplate{frame numbering}[fraction],
438 }

```

notes Show notes in presentation

```

439 \pgfkeys{
440   /neo/outer/notes/.cd,
441   .is choice,
442   none/.code=\pgfkeysalso{notes=hide},
443   hide/.code=\setbeameroption{hide notes},
444   show/.code={\setbeamertemplate{note page}[print]\setbeameroption{show notes}},
445   only/.code={\setbeamertemplate{note page}[print]\setbeameroption{show only notes}},
446   preview-left/.code={\setbeamertemplate{note page}[default]\setbeameroption{show notes on se
447   preview-right/.code={\setbeamertemplate{note page}[default]\setbeameroption{show notes on s
448   preview-top/.code={\setbeamertemplate{note page}[default]\setbeameroption{show notes on sec
449   preview-bottom/.code={\setbeamertemplate{note page}[default]\setbeameroption{show notes on
450   preview-left-big/.code={\setbeamertemplate{note page}[preview-big]\setbeameroption{show not
451   preview-right-big/.code={\setbeamertemplate{note page}[preview-big]\setbeameroption{show no
452   preview-top-big/.code={\setbeamertemplate{note page}[preview-big]\setbeameroption{show note

```

```

453 preview-bottom-big/.code={\setbeamertemplate{note page}[preview-big]\setbeameroption{show n
454 left/.code={\setbeamertemplate{note page}[print]\setbeameroption{show notes on second scree
455 right/.code={\setbeamertemplate{note page}[print]\setbeameroption{show notes on second scree
456 top/.code={\setbeamertemplate{note page}[print]\setbeameroption{show notes on second screen
457 bottom/.code={\setbeamertemplate{note page}[print]\setbeameroption{show notes on second scr
458 }

```

footer Adds additional presentation information to the footer

```

459 \pgfkeys{
460   /neo/outer/footer/.cd,
461   .is choice,
462   none/.code=\setbeamertemplate{frame footer}[none],
463   author/.code=\setbeamertemplate{frame footer}[author],
464   author title/.code=\setbeamertemplate{frame footer}[author title],
465   title/.code=\setbeamertemplate{frame footer}[title],
466   title section/.code=\setbeamertemplate{frame footer}[title section],
467 }

```

footer style Footer background color

```

468 \providebool{neo@standoutfooter}
469 \pgfkeys{
470   /neo/outer/footer style/.cd,
471   .is choice,
472   plain/.code={\boolfalse{neo@standoutfooter}\setbeamertemplate{footline}[plain]},
473   standout/.code={\booltrue{neo@standoutfooter}\setbeamertemplate{footline}[standout]},
474 }

```

progressbar Adds a progress bar to the top, bottom, or frametitle of each slide. In case this is used in combination with footer style, it must be set afterwards (e.g. to have the progressbar on top of the footer styled with standout).

```

475 \pgfkeys{
476   /neo/outer/progressbar/.cd,
477   .is choice,
478   none/.code={%
479     \setbeamertemplate{headline}[plain]
480     \setbeamertemplate{frametitle}[plain]
481   },
482   head/.code={\pgfkeys{/neo/outer/progressbar=none}

```

```

483     \addtobeamertemplate{headline}{}{%
484         \usebeamertemplate*{progress bar in head/foot}
485     }
486 },
487 frametitle/.code={\pgfkeys{/neo/outer/progressbar=none}
488     \addtobeamertemplate{frametitle}{}{%
489         \usebeamertemplate*{progress bar in head/foot}
490     }
491 },
492 foot/.code={\pgfkeys{/neo/outer/progressbar=none}
493     \ifbool{neo@standoutfooter}{%
494         \addtobeamertemplate{footline}{\usebeamertemplate*{progress bar in head/foot}}{} %
495     }{%
496         \addtobeamertemplate{footline}{}{\usebeamertemplate*{progress bar in head/foot}} %
497     }
498 },
499 }

```

`\neo@outer@setdefaults` Sets default values for outer theme options.

```

500 \newcommand{\neo@outer@setdefaults}{
501     \pgfkeys{/neo/outer/.cd,
502         frametitle icon=none,
503         footer=none,
504         footer style=plain,
505         numbering=counter,
506         progressbar=none,
507     }
508 }

```

8.3.3 Head and footline

All good **beamer** presentations should already remove the navigation symbols, but **neo** removes them automatically (just in case).

```

509 \setbeamertemplate{navigation symbols}{}

```

`frametitle icon` Templates for the icon on the right of the frame title.

```

510 \defbeamertemplate{frametitle icon}{none}{}
511 \defbeamertemplate{frametitle icon}{i4}{\hfill\raisebox{-.25\height}{\includegraphics[height=1

```



```
512 \defbeamertemplate{frametitle icon}{fau}{ \hfill\raisebox{-.25\height}{\includegraphics[height=
```

frame numbering Templates for the frame number. Can be omitted, shown or displayed as a fraction of the total frames.

```
513 \defbeamertemplate{frame footer}{none}{}
514 \defbeamertemplate{frame footer}{author}{
515 \node[itml] (footeriteml) at (current page.south west) {\insertshortauthor};%
516 }
517 \defbeamertemplate{frame footer}{author title}{
518 \node[itml] (footeriteml) at (current page.south west) {\insertshortauthor};%
519 \node[itemm] (footeritemm) at (current page.south) {\insertshorttitle};%
520 }
521 \defbeamertemplate{frame footer}{title}{
522 \node[itml] (footeriteml) at (current page.south west) {\insertshorttitle};%
523 }
524 \defbeamertemplate{frame footer}{title section}{
525 \node[itml] (footeriteml) at (current page.south west) {\insertshorttitle};%
526 \node[itemm] (footeritemm) at (current page.south) {\insertshortsection};%
527 }
528 \defbeamertemplate{frame footer}{custom}[1]{ #1 }
```

Add strut to ensure that frame numbers don't jump

```
529 \newcommand{\neo@framenumberingstrut}{\vphantom{0123456789}}
530 \defbeamertemplate{frame numbering}{none}{}
531 \defbeamertemplate{frame numbering}{counter}{\neo@framenumberingstrut\insertframenumber}
532 \defbeamertemplate{frame numbering}{fraction}{
533 \neo@framenumberingstrut\insertframenumber/\inserttotalframenumber
534 }
```

headline Templates for the head- and footline at the top and bottom of each frame.

footline

```
535 \defbeamertemplate{headline}{plain}{}
536 \defbeamertemplate{footline}{plain}{%
537 \begin{tikzpicture}[overlay]
538 \node[anchor=south] (footerbaseline) at (current page.south) {\vphantom{Ag}};
539 \tikzset{%
540 every node/.style={font=\usebeamerfont{page number in head/foot}\usebeamercolor[fg]{block title}
541 every node/.append style={baseline=(footerbaseline.base)},
542 iteml/.style={anchor=base west, xshift=1ex},
```

```

543 itemm/.style={anchor=base},
544 itemr/.style={anchor=base east, xshift=-1ex},
545 };
546 \def\drawfooter{%
547 \begin{scope}[every node/.append style={yshift=0.9em}]
548 \usebeamertemplate{frame footer}
549 \node[itemr] (footeritemr) at (current page.south east) {\usebeamertemplate{frame numbering}};
550 \end{scope}
551 }%
552 \drawfooter%
553 \end{tikzpicture}
554 \end{beamercolorbox}%
555 }
556 \long\def\ifnodedefined#1#2#3{%
557 \ifundefined{pgf@sh@ns@#1}{#3}{#2}%
558 }
559 \defbeamertemplate{footline}{standout}{%
560 \begin{tikzpicture}[overlay]
561 \node[anchor=south] (footerbaseline) at (current page.south) {\vphantom{Ag}};
562 \tikzset{%
563 every node/.style={font=\usebeamerfont{page number in head/foot}\usebeamercolor[bg]{block title}
564 every node/.append style={baseline=(footerbaseline.base)},
565 iteml/.style={anchor=base west, xshift=1ex},
566 itemm/.style={anchor=base},
567 itemr/.style={anchor=base east, xshift=-1ex},
568 };
569 \def\drawfooter{%
570 \begin{scope}[every node/.append style={yshift=0.9em}]
571 \usebeamertemplate{frame footer}
572 \node[itemr] (footeritemr) at (current page.south east) {\usebeamertemplate{frame numbering}};
573 \end{scope}
574 \ifnodedefined{footeritemm}{%
575 \node[fit=(footeriteml)(footeritemm)(footeritemr)] (footer) {};
576 }{%
577 \node[fit=(footeriteml)(footeritemr)] (footer) {};
578 }
579 }%
580 \phantom{\drawfooter}%
581 \path[fill={block title.fg}] (current page.south west) rectangle (footer.north -| current page.
582 \drawfooter%

```

```

583 \end{tikzpicture}
584 }

```

8.3.4 Frametitle

frametitle Templates for the frame title, which is optionally underlined with a progress bar.

```

585 \newlength{\neo@frametitle@padding}
586 \setlength{\neo@frametitle@padding}{2.2ex}
587 \newcommand{\neo@frametitlestrut@start}{
588   \rule{0pt}{\neo@frametitle@padding +%
589     \totalheightof{
590       \ifcsdef{neo@frametitleformat}{\neo@frametitleformat X}{X}%
591     }%
592   }%
593 }
594 \newcommand{\neo@frametitlestrut@end}{
595   \rule[-\neo@frametitle@padding]{0pt}{\neo@frametitle@padding}
596 }
597 \defbeamertemplate{frametitle}{plain}{%
598   \nointerlineskip%
599   \begin{beamercolorbox}[%
600     wd=\paperwidth,%
601     sep=0pt,%
602     leftskip=\neo@frametitle@padding,%
603     rightskip=\neo@frametitle@padding,%
604   ]{frametitle}%
605   \neo@frametitlestrut@start%
606   \insertframetitle%
607   \usebeamertemplate*{frametitle icon}%
608   \nolinebreak%
609   \neo@frametitlestrut@end%
610   \end{beamercolorbox}%
611 }
612 \setbeamertemplate{frametitle continuation}{%
613   \usebeamerfont{frametitle}
614   {\normalfont (\insertcontinuationcount)}
615 }

```

progress bar in head/foot Template for the progress bar optionally displayed below the frame title on each page. Much of this code is duplicated in the inner theme's template

progress bar in section page.

```

616 \newlength{\neo@progressinheadfoot}
617 \newlength{\neo@progressinheadfoot@linewidth}
618 \setlength{\neo@progressinheadfoot@linewidth}{0.8pt}
619 \setbeamertemplate{progress bar in head/footer}{
620   \nointerlineskip
621   \setlength{\neo@progressinheadfoot}{%
622     \paperwidth * \ratio{\insertframenum pt}{\inserttotalframenum pt}}%
623   }%
624   \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/footer}
625     \tikzexternaldisable%
626     \begin{tikzpicture}
627       \fill[bg] (0,0) rectangle (\paperwidth, \neo@progressinheadfoot@linewidth);
628       \fill[fg] (0,0) rectangle (\neo@progressinheadfoot, \neo@progressinheadfoot@linewidth);
629     \end{tikzpicture}%
630     \tikzexternalenable%
631   \end{beamercolorbox}
632 }

```

custom notes Templates for note pages

```

633 \defbeamertemplate{note page}{preview-big}
634 {%
635   {%
636     \scriptsize
637     \usebeamerfont{note title}\usebeamercolor[fg]{note title}%
638     \ifbeamercolorempy[bg]{note title}{}{%
639       \insertvrule{.45\paperheight}{note title.bg}%
640       \vskip-.45\paperheight%
641       \nointerlineskip%
642     }%
643     \vbox{
644       \hfill\insertslideintonotes{0.45}\hskip-\Gm@rmargin\hskip0pt%
645       \vskip-0.45\paperheight%
646       \nointerlineskip
647       \begin{pgfpicture}{0cm}{0cm}{0cm}{0cm}
648         \begin{pgfflowlevelslope}{\pgftransformrotate{90}}
649           {\pgftransformshift{\pgfpoint{-2cm}{0.2cm}}}%
650           \pgftext[base,left]{\usebeamerfont{note date}\usebeamercolor[fg]{note date}\the\year-
651         \end{pgfflowlevelslope}

```

```

652     \end{pgfpicture}}
653     \nointerlineskip
654     \vbox to .45\paperheight{\vskip0.5em
655         \hbox{\insertshorttitle[width=8cm]}}%
656         \setbox\beamer@tempbox=\hbox{\insertsection}%
657         \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip4pt\raise3pt\hbox{\vrule
658             width0.4pt height7pt\vrule width 9pt
659             height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere{}\insertsection
660             }%
661             \setbox\beamer@tempbox=\hbox{\insertsubsection}%
662             \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip17.4pt\raise3pt\hbox{\vrule
663                 width0.4pt height7pt\vrule width 9pt
664                 height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere{}\insertsubsec
665                 }%
666                 \setbox\beamer@tempbox=\hbox{\insertshortframetitle}%
667                 \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip30.8pt\raise3pt\hbox{\vrule
668                     width0.4pt height7pt\vrule width 9pt
669                     height0.4pt}}\hskip1pt\hbox{\insertshortframetitle[width=7cm]}}\fi%
670                 }%
671                 \vfil}}%
672             }%
673             \ifbeamercolorempy[bg]{note page}{-}{-%
674                 \nointerlineskip%
675                 \insertvrule{.55\paperheight}{note page.bg}%
676                 \vskip-.55\paperheight%
677             }%
678             \vskip.25em
679             \nointerlineskip
680             \insertnote
681         }
682     \defbeamertemplate{note page}{\print}
683     {%
684     {%
685         \nointerlineskip%
686         \begin{beamercolorbox}[%
687             wd=\paperwidth,%
688             sep=0pt,%
689             leftskip=\neo@frametitle@padding,%
690             rightskip=\neo@frametitle@padding,%
691             ]{note title}%

```

```

692 \usebeamerfont{frametitle}%
693 \neo@frametitlestrut@start%
694 \insertframetitle%
695 \usebeamertemplate*{frametitle icon}%
696 \nolinebreak%
697 \neo@frametitlestrut@end%
698 \end{beamercolorbox}%
699 }%
700 \insertnote%
701 \vfill%
702 \begin{beamercolorbox}[wd=\paperwidth, sep=3ex]{footline}%
703 \usebeamerfont{page number in head/foot}%
704 \usebeamertemplate*{frame footer}
705 \hfill%
706 \usebeamertemplate*{frame numbering}
707 \end{beamercolorbox}%
708 \vskip4pt%
709 }

```

appendix Removes page numbering and per-slide progress bars when `\appendix` is called. This makes it easier to include additional “backup slides” at the end of the presentation, especially in conjunction with the package `appendixnumberbeamer`.

```

710 \AtBeginDocument{%
711 \apptocmd{\appendix}{%
712 \pgfkeys{%
713 /neo/outer/.cd,
714 numbering=none,
715 progressbar=none}
716 }{}{}
717 }

```

8.3.5 Process package options

```

718 \neo@outer@setdefaults
719 \ProcessPgfPackageOptions{/neo/outer}

```

8.4 neo font theme

A `beamer` font theme sets the style of the font used in the document.

8.4.1 Package dependencies

```
720 \RequirePackage{etoolbox}
721 \RequirePackage{ifxetex}
722 \RequirePackage{ifluatex}
723 \RequirePackage{pgfopts}
```

8.4.2 Load Fira fonts

If the presentation is compiled with Xe_{La}T_EX or Lua_{La}T_EX, the fontspec package is loaded and we search for the Fira fonts.

```
724 \ifboolexpr{bool {xetex} or bool {luatex}}{
725   \ifpackageloaded{fontspec}{
726     \PassOptionsToPackage{no-math}{fontspec}
727   }{
728     \RequirePackage[no-math]{fontspec}
729   }

730   \IfFileExists{FiraSans-Regular.otf}{
731     \defaultfontfeatures{
732       Scale      = 1.0,
733       Extension = .otf
734     }
735   }{
736     \PackageWarning{beamerthemeneo}{%
737       FiraSans font not found in path, therefore using system fonts. %
738       Make sure you have the fonts installed.%
739     }
740   }

741   \setmonofont
742     [ Numbers = {Monospaced,Lining},
743       UprightFont   = *-Regular ,
744       ItalicFont    = *-Regular ,
745       BoldFont      = *-Medium ,
746       BoldItalicFont = *-Medium ,
747     ]
748     {FiraMono}

749   \newcommand{\neo@fontsave}{
750     \let\firaneofamily\sfdefault
751     \renewcommand*{familydefault}{\firaneofamily}
752   }
```

```

753 \newcommand{\neo@fontlight}{
754   \setsansfont[
755     Numbers = {OldStyle, Monospaced},
756     UprightFont    = *-Light ,
757     ItalicFont     = *-LightItalic ,
758     BoldFont       = *-Regular ,
759     BoldItalicFont = *-RegularItalic ,
760   ]{FiraSans}
761   \neo@fontsave
762 }
763 \newcommand{\neo@fontbook}{
764   \setsansfont[
765     Numbers = {OldStyle, Monospaced},
766     UprightFont    = *-Book ,
767     ItalicFont     = *-BookItalic ,
768     BoldFont       = *-Medium ,
769     BoldItalicFont = *-MediumItalic ,
770   ]{FiraSans}
771   \neo@fontsave
772 }
773 \newcommand{\neo@fontregular}{
774   \setsansfont[
775     Numbers = {OldStyle, Monospaced},
776     UprightFont    = *-Regular ,
777     ItalicFont     = *-RegularItalic ,
778     BoldFont       = *-SemiBold ,
779     BoldItalicFont = *-SemiBoldItalic ,
780   ]{FiraSans}
781   \neo@fontsave
782 }
783 \AtBeginEnvironment{tabular}{%
784   \addfontfeature{Numbers={Monospaced}}}%
785 }
786 }{%
787   \RequirePackage[utf8]{inputenc}
788   \IfFileExists{FiraSans.sty}{
789     \RequirePackage[T1]{fontenc}
790     \RequirePackage[sfdefault]{FiraSans}
791     \RequirePackage[nomap,lining]{FiraMono}
792     \def\bfseries@tt{mb}

```



```

793 \newcommand{\neo@fontsave}{
794   \edef\familydefault{\sfdefault}
795   \edef\seriesdefault{\mdseries@sf}
796 }
797 \newcommand{\neo@fontlight}{
798   \def\mdseries@sf{l}
799   \def\bfseries@sf{m}
800   \neo@fontsave
801 }
802 \newcommand{\neo@fontbook}{
803   \def\bfseries@sf{mb}
804   \neo@fontsave
805 }
806 \newcommand{\neo@fontregular}{
807   \def\mdseries@sf{m}
808   \def\bfseries@sf{sb}
809   \neo@fontsave
810 }
811 }{
812   \PackageWarning{beamerthemeneo}{%
813     You need to install the Fira Fonts package or compile with XeLaTeX or %
814     LuaLaTeX to use the included Fira fonts%
815   }
816 }
817 }

```

This concludes the portion of the code which is only run when compiled with Xe^LA^TE^X or Lua^LA^TE^X. The remainder of this package applies regardless of the compiling engine.

8.4.3 General font definitions

```

818 \setbeamerfont{title}{size=\Large,%
819               series=\bfseries}
820 \setbeamerfont{author}{size=\small}
821 \setbeamerfont{date}{size=\small}
822 \setbeamerfont{section title}{size=\Large,%
823               series=\bfseries}
824 \setbeamerfont{block title}{size=\normalsize,%
825               series=\bfseries}

```

```

826 \setbeamerfont{block title alerted}{size=\normalsize,%
827                               series=\bfseries}
828 \setbeamerfont*{subtitle}{size=\large}
829 \setbeamerfont{frametitle}{size=\large,%
830                               series=\bfseries}
831 \setbeamerfont{caption}{size=\small}
832 \setbeamerfont{caption name}{series=\bfseries}
833 \setbeamerfont{description item}{series=\bfseries}
834 \setbeamerfont{page number in head/foot}{size=\scriptsize}
835 \setbeamerfont{bibliography entry author}{size=\normalsize,%
836                               series=\normalfont}
837 \setbeamerfont{bibliography entry title}{size=\normalsize,%
838                               series=\bfseries}
839 \setbeamerfont{bibliography entry location}{size=\normalsize,%
840                               series=\normalfont}
841 \setbeamerfont{bibliography entry note}{size=\small,%
842                               series=\normalfont}
843 \setbeamerfont{standout}{size=\Large,%
844                               series=\bfseries}

```

8.4.4 Font style options

`titleformat title` Controls the overall font style.

```

845 \pgfkeys{
846   /neo/font/style/.cd,
847   .is choice,
848   light/.code={\neo@fontlight},
849   book/.code={\neo@fontbook},
850   regular/.code={\neo@fontregular},
851 }

```

8.4.5 Title format options

`titleformat title` Controls the format of the title.

```

852 \pgfkeys{
853   /neo/font/titleformat title/.cd,
854   .is choice,
855   regular/.code={%
856     \let\neo@titleformat\@empty%
857     \setbeamerfont{title}{shape=\normalfont}%

```

```

858 },
859 smallcaps/.code={%
860     \let\neo@titleformat\@empty%
861     \setbeamerfont{title}{shape=\scshape}%
862 },
863 allsmallcaps/.code={%
864     \let\neo@titleformat\lowercase%
865     \setbeamerfont{title}{shape=\scshape}%
866     \PackageWarning{beamerthemeneo}{%
867         Be aware that titleformat title=allsmallcaps can lead to problems%
868     }
869 },
870 allcaps/.code={%
871     \let\neo@titleformat\uppercase%
872     \setbeamerfont{title}{shape=\normalfont}%
873     \PackageWarning{beamerthemeneo}{%
874         Be aware that titleformat title=allcaps can lead to problems%
875     }
876 },
877 }

```

`titleformat subtitle` Control the format of the subtitle.

```

878 \pgfkeys{
879 /neo/font/titleformat subtitle/.cd,
880 .is choice,
881 regular/.code={%
882     \let\neo@subtitleformat\@empty%
883     \setbeamerfont{subtitle}{shape=\normalfont}%
884 },
885 smallcaps/.code={%
886     \let\neo@subtitleformat\@empty%
887     \setbeamerfont{subtitle}{shape=\scshape}%
888 },
889 allsmallcaps/.code={%
890     \let\neo@subtitleformat\lowercase%
891     \setbeamerfont{subtitle}{shape=\scshape}%
892     \PackageWarning{beamerthemeneo}{%
893         Be aware that titleformat subtitle=allsmallcaps can lead to problems%
894     }

```

```

895 },
896 allcaps/.code={%
897     \let\neo@subtitleformat\uppercase%
898     \setbeamerfont{subtitle}{shape=\normalfont}%
899     \PackageWarning{beamerthemeneo}{%
900         Be aware that titleformat subtitle=allcaps can lead to problems%
901     }
902 },
903 }

```

`titleformat section` Controls the format of the section title.

```

904 \pgfkeys{
905     /neo/font/titleformat section/.cd,
906     .is choice,
907     regular/.code={%
908         \let\neo@sectiontitleformat\@empty%
909         \setbeamerfont{section title}{shape=\normalfont}%
910     },
911     smallcaps/.code={%
912         \let\neo@sectiontitleformat\@empty%
913         \setbeamerfont{section title}{shape=\scshape}%
914     },
915     allsmallcaps/.code={%
916         \let\neo@sectiontitleformat\MakeLowercase%
917         \setbeamerfont{section title}{shape=\scshape}%
918         \PackageWarning{beamerthemeneo}{%
919             Be aware that titleformat section=allsmallcaps can lead to problems%
920         }
921     },
922     allcaps/.code={%
923         \let\neo@sectiontitleformat\MakeUppercase%
924         \setbeamerfont{section title}{shape=\normalfont}%
925         \PackageWarning{beamerthemeneo}{%
926             Be aware that titleformat section=allcaps can lead to problems%
927         }
928     },
929 }

```

`frametitleformat` Control the format of the frame title.

```

930 \pgfkeys{
931   /neo/font/titleformat frame/.cd,
932   .is choice,
933   regular/.code={%
934     \let\neo@frametitleformat\@empty%
935     \setbeamerfont{frametitle}{shape=\normalfont}%
936   },
937   smallcaps/.code={%
938     \let\neo@frametitleformat\@empty%
939     \setbeamerfont{frametitle}{shape=\scshape}%
940   },
941   allsmallcaps/.code={%
942     \let\neo@frametitleformat\MakeLowercase%
943     \setbeamerfont{frametitle}{shape=\scshape}%
944     \PackageWarning{beamerthemeneo}{%
945       Be aware that titleformat frame=allsmallcaps can lead to problems%
946     }
947   },
948   allcaps/.code={%
949     \let\neo@frametitleformat\MakeUppercase%
950     \setbeamerfont{frametitle}{shape=\normalfont}%
951     \PackageWarning{beamerthemeneo}{%
952       Be aware that titleformat frame=allcaps can lead to problems%
953     }
954   },
955 }
```

`titleformat aliases` Allows `titleformat title` et al. to be used in the `\usetheme` declaration, where L^AT_EX automatically removes all spaces.

```

956 \pgfkeys{
957   /neo/font/.cd,
958   titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
959   titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
960   titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
961   titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
962 }
```

`\neo@font@setdefaults` Sets default values for font theme options.

```
963 \newcommand{\neo@font@setdefaults}{  
964   \pgfkeys{/neo/font/.cd,  
965     style=book,  
966     titleformat title=regular,  
967     titleformat subtitle=regular,  
968     titleformat section=regular,  
969     titleformat frame=regular,  
970   }  
971 }
```

We first define hooks to change the case format of the titles.

```
972 \def\neo@titleformat#1{#1}  
973 \def\neo@subtitleformat#1{#1}  
974 \def\neo@sectiontitleformat#1{#1}  
975 \def\neo@frametitleformat#1{#1}
```

To make the uppercase and lowercase macros work in the title, subtitle, etc., we have to patch the appropriate `beamer` commands that set their values. This solution was suggested by Enrico Gregorio in an answer to [this StackExchange question](#).

```
976 \patchcmd{\beamer@title}%  
977   {\def\inserttitle{#2}}%  
978   {\def\inserttitle{\neo@titleformat{#2}}}%  
979   {}%  
980   {\PackageError{beamerfontthemeneo}{Patching title failed}\@ehc}  
981 \patchcmd{\beamer@subtitle}%  
982   {\def\insertsubtitle{#2}}%  
983   {\def\insertsubtitle{\neo@subtitleformat{#2}}}%  
984   {}%  
985   {\PackageError{beamerfontthemeneo}{Patching subtitle failed}\@ehc}  
986 \patchcmd{\sectionentry}  
987   {\def\insertsectionhead{#2}}  
988   {\def\insertsectionhead{\neo@sectiontitleformat{#2}}}  
989   {}  
990   {\PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc}  
991 \@tempswafalse  
992 \patchcmd{\beamer@section}
```

```

993 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#1}}}}
994 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
995   \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
996 {\@tempswatrue}
997 {}
998 \patchcmd{\beamer@section}
999 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
1000 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
1001   \neo@sectiontitleformat{#1}}}}
1002 {\@tempswatrue}
1003 {}
1004 \patchcmd{\beamer@section}
1005 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#1}}}}
1006 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1007   \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
1008 {\@tempswatrue}
1009 {}
1010 \patchcmd{\beamer@section}
1011 {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#1}}}
1012 {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1013   \noexpand\neo@sectiontitleformat{#1}}}}
1014 {\@tempswatrue}
1015 {}
1016 \if@tempswa\else
1017   \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
1018 \fi
1019 \@tempswafalse
1020 \patchcmd{\beamer@subsection}
1021 {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#1}}}}
1022 {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1023   \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
1024 {\@tempswatrue}
1025 {}
1026 \patchcmd{\beamer@subsection}
1027 {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
1028 {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{%
1029   \neo@sectiontitleformat{#1}}}}
1030 {\@tempswatrue}
1031 {}
1032 \patchcmd{\beamer@subsection}

```

```

1033 {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#1}}}
1034 {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1035   \noexpand\neo@sectiontitleformat{#1}}}}
1036 {\@tempswatrue}
1037 {}
1038 \if@tempswa\else
1039   \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
1040 \fi

```

Similarly, to make the `\MakeLowercase` and `\MakeUppercase` macros work in the frame title we have to patch `\beamer@@frametitle`.

```

1041 \patchcmd{\beamer@@frametitle}
1042   {%
1043     \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
1044     \usebeamertemplate*{frametitle continuation}\fi}}%
1045     \gdef\beamer@frametitle{#2}%
1046     \gdef\beamer@shortframetitle{#1}%
1047   }}
1048   {%
1049     \gdef\insertframetitle{{\neo@frametitleformat{#2}\ifnum%
1050     \beamer@autobreakcount>0\relax{}\space%
1051     \usebeamertemplate*{frametitle continuation}\fi}}%
1052     \gdef\beamer@frametitle{#2}%
1053     \gdef\beamer@shortframetitle{#1}%
1054   }}
1055   {}
1056   {\PackageError{beamerfontthemeneo}{Patching frame title failed}\@ehc}

```

8.4.6 Process package options

```

1057 \neo@font@setdefaults
1058 \ProcessPgfPackageOptions{/neo/font}

```

8.5 neo color theme

8.5.1 Package dependencies

```

1059 \RequirePackage{pgfopts}

```


8.5.2 Options

block Optionally adds a light grey background to block environments like **theorem** and **example**.

```
1060 \pgfkeys{
1061   /neo/color/block/.cd,
1062   .is choice,
1063   transparent/.code=\neo@block@transparent,
1064   fill/.code=\neo@block@fill,
1065 }
```

colors Provides the option to have a dark background and light foreground instead of the reverse.

```
1066 \pgfkeys{
1067   /neo/color/background/.cd,
1068   .is choice,
1069   dark/.code=\neo@colors@dark,
1070   light/.code=\neo@colors@light,
1071 }
```

\neo@color@setdefaults Sets default values for color theme options.

```
1072 \newcommand{\neo@color@setdefaults}{
1073   \pgfkeys{/neo/color/.cd,
1074     background=light,
1075     block=transparent,
1076   }
1077 }
```

8.5.3 Base colors

```
1078
1079 \definecolor{nDarkGrey}{RGB}{152,164,174}
1080 \definecolor{nGrey}{RGB}{210,213,215}
1081 \definecolor{nLightGrey}{RGB}{235,236,238}
1082
1083 \definecolor{nDarkRed}{RGB}{141,20,41}
1084 \definecolor{nRed}{RGB}{201,169,147}
1085 \definecolor{nLightRed}{RGB}{237,231,222}
1086
```

```

1087 \definecolor{nDarkGreen}{RGB}{0,155,119}
1088 \definecolor{nGreen}{RGB}{170,207,189}
1089 \definecolor{nLightGreen}{RGB}{229,239,234}
1090
1091 \definecolor{nDarkBlue}{RGB}{0,56,101}
1092 \definecolor{nBlue}{RGB}{144,167,198}
1093 \definecolor{nLightBlue}{RGB}{221,229,240}
1094
1095 \definecolor{nDarkCyan}{RGB}{0,177,235}
1096 \definecolor{nCyan}{RGB}{180,214,245}
1097 \definecolor{nLightCyan}{RGB}{234,243,252}
1098
1099 \definecolor{nDarkYellow}{RGB}{201,147,19}
1100 \definecolor{nYellow}{RGB}{217,198,137}
1101 \definecolor{nLightYellow}{RGB}{243,238,223}
1102
1103 \definecolor{nBlack}{HTML}{011F32}
1104 \definecolor{nWhite}{RGB}{250,250,250}

```

8.5.4 Alias colors

Support the colors provided by the old i4 beamer theme.

```

1105 \colorlet{i4red}{nDarkRed}
1106 \colorlet{i4green}{nDarkGreen}
1107 \colorlet{i4blue}{nDarkBlue}
1108 \colorlet{i4cyan}{nDarkCyan}
1109 \colorlet{i4yellow}{nDarkYellow}
1110 \colorlet{i4grey}{nDarkGrey}
1111 \definecolor{darkred}{rgb}{0.8,0,0}
1112 \colorlet{beamergreen}{green!50!black}

```

8.5.5 Base styles

All colors in **neo** are derived from the definitions of `normal text`, `alerted text`, and `example text`.

```

1113 \newcommand{\neo@colors@dark}{
1114   \setbeamercolor{normal text}{%
1115     fg=nWhite,
1116     bg=nBlack
1117   }

```

```

1118 \setbeamercolor{normal item}{%
1119     fg=nWhite,
1120     bg=nDarkBlue
1121 }
1122 \usebeamercolor[fg]{normal text}
1123 }
1124 \newcommand{\neo@colors@light}{
1125     \setbeamercolor{normal text}{%
1126         fg=nBlack,
1127         bg=nWhite
1128     }
1129     \setbeamercolor{normal item}{%
1130         fg=nDarkBlue,
1131         bg=nWhite
1132     }
1133 }
1134 \setbeamercolor{alerted text}{%
1135     fg=nDarkRed
1136 }
1137 \setbeamercolor{example text}{%
1138     fg=nDarkYellow
1139 }
1140 \setbeamercolor{note title}{%
1141     fg=nDarkBlue,
1142     bg=nGrey
1143 }
1144 \setbeamercolor{note page}{%
1145     fg=nBlack,
1146     bg=nLightGrey
1147 }

```

8.5.6 Derived colors

The titles and structural elements (e.g. `itemize` bullets) are set in the same color as `normal text` and `normal item`. This would ideally be done by setting `normal text` and `normal item` as a parent style, which we do to set `titlelike`, but this doesn't work for `structure` as its foreground is set explicitly in `beamercolorthemedefault.sty`.

```

1148 \setbeamercolor{titlelike}{use=normal text, parent=normal text}

```

```

1149 \setbeamercolor{author}{use=normal text, parent=normal text}
1150 \setbeamercolor{date}{use=normal text, parent=normal text}
1151 \setbeamercolor{institute}{use=normal text, parent=normal text}
1152 \setbeamercolor{structure}{use=normal item, fg=normal item.fg}

```

The “primary” palette should be used for the most important navigational elements, and possibly of other elements. **neo** uses it for frame titles and slides.

```

1153 \setbeamercolor{palette primary}{%
1154   use=normal text,
1155   fg=normal text.bg,
1156   bg=nDarkBlue
1157 }
1158 \setbeamercolor{frametitle}{%
1159   use=palette primary,
1160   parent=palette primary
1161 }

```

The **neo** inner or outer themes optionally display progress bars in various locations. Their color is set by **progress bar** but the two different kinds can be customized separately. The horizontal rule on the title page is also set based on the progress bar color and can be customized with **title separator**.

```

1162 \setbeamercolor{progress bar}{%
1163   use=normal text,
1164   fg=nDarkBlue,
1165   bg=nLightBlue
1166 }
1167 \setbeamercolor{title separator}{
1168   use=progress bar,
1169   parent=progress bar
1170 }
1171 \setbeamercolor{progress bar in head/foot}{%
1172   use=normal text.fg,
1173   fg=nBlack,
1174   parent=progress bar
1175 }
1176 \setbeamercolor{progress bar in section page}{
1177   use=progress bar,
1178   parent=progress bar
1179 }

```

Block environments such as `theorem` and `example` have no background color by default. The option `block=fill` sets a background color based on the background and foreground of `normal text`. The option `block=transparent` reverts the block environments to an empty background, which can be useful if changing colors mid-presentation.

```

1180 \newcommand{\neo@block@transparent}{
1181   \setbeamercolor{block title}{%
1182     use=normal text,
1183     fg=nDarkBlue,
1184     bg=
1185   }
1186   \setbeamercolor{block title alerted}{%
1187     use={block title, alerted text},
1188     bg=block title.bg,
1189     fg=alerted text.fg
1190   }
1191   \setbeamercolor{block title example}{%
1192     use={block title, example text},
1193     bg=block title.bg,
1194     fg=example text.fg
1195   }
1196   \setbeamercolor{block body}{
1197     bg=
1198   }
1199   \setbeamercolor{block body alerted}{
1200     use=block body,
1201     parent=block body
1202   }
1203   \setbeamercolor{block body example}{
1204     use=block body,
1205     parent=block body
1206   }
1207 }
1208 \newcommand{\neo@block@fill}{
1209   \setbeamercolor{block title}{%
1210     use=normal text,
1211     fg=nDarkBlue,
1212     bg=nGrey
1213   }

```

```

1214 \setbeamercolor{block title alerted}{%
1215     use={block title, alerted text},
1216     bg=alerted text.fg,
1217     fg=alerted text.bg
1218 }
1219 \setbeamercolor{block title example}{%
1220     use={block title, example text},
1221     bg=example text.fg,
1222     fg=example text.bg
1223 }
1224 \setbeamercolor{block body}{
1225     use={block title, normal text},
1226     bg=nLightGrey
1227 }
1228 \setbeamercolor{block body alerted}{
1229     use=block body,
1230     parent=block body,
1231     bg=nRed!50,
1232 }
1233 \setbeamercolor{block body example}{
1234     use=block body,
1235     parent=block body,
1236     bg=nYellow!50
1237 }
1238 }
1239

```

Footnotes

```

1240 \setbeamercolor{footnote}{fg=normal text.fg!90}
1241 \setbeamercolor{footnote mark}{fg=.
```

We also reset the bibliography colors in order to pick up the surrounding colors at the time of use. This prevents us having to set the correct color in normal and standout mode.

```

1242 \setbeamercolor{bibliography entry author}{fg=, bg=}
1243 \setbeamercolor{bibliography entry title}{fg=, bg=}
1244 \setbeamercolor{bibliography entry location}{fg=, bg=}
1245 \setbeamercolor{bibliography entry note}{fg=, bg=}

```

8.5.7 Process package options

```
1246 \neo@color@setdefaults
1247 \ProcessPgfPackageOptions{/neo/color}
1248 \mode<all>
```

8.6 Tol pgfplots theme

Paul Tol's 12-color palette¹ is as follows:

```
1249 \definecolor{TolDarkPurple}{HTML}{332288}
1250 \definecolor{TolDarkBlue}{HTML}{6699CC}
1251 \definecolor{TolLightBlue}{HTML}{88CCFF}
1252 \definecolor{TolLightGreen}{HTML}{44AA99}
1253 \definecolor{TolDarkGreen}{HTML}{117733}
1254 \definecolor{TolDarkBrown}{HTML}{999933}
1255 \definecolor{TolLightBrown}{HTML}{DDCC77}
1256 \definecolor{TolDarkRed}{HTML}{661100}
1257 \definecolor{TolLightRed}{HTML}{CC6677}
1258 \definecolor{TolLightPink}{HTML}{AA4466}
1259 \definecolor{TolDarkPink}{HTML}{882255}
1260 \definecolor{TolLightPurple}{HTML}{AA4499}
```

To use these colors, we describe “cycle lists” from which PGF chooses styles for the different series in a chart.

`mbarplot cycle` Colors and styles intended for bar charts with up to 12 series.

```
1261 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
1262   {draw=TolDarkBlue,    fill=TolDarkBlue!70},
1263   {draw=TolLightBrown,  fill=TolLightBrown!70},
1264   {draw=TolLightGreen,  fill=TolLightGreen!70},
1265   {draw=TolDarkPink,    fill=TolDarkPink!70},
1266   {draw=TolDarkPurple,  fill=TolDarkPurple!70},
1267   {draw=TolDarkRed,     fill=TolDarkRed!70},
1268   {draw=TolDarkBrown,   fill=TolDarkBrown!70},
1269   {draw=TolLightRed,    fill=TolLightRed!70},
1270   {draw=TolLightPink,   fill=TolLightPink!70},
1271   {draw=TolLightPurple, fill=TolLightPurple!70},
```

¹Tol actually describes several palettes; these colours are taken from the bottom row of Figure 3 in his technical note.

```

1272 {draw=TolLightBlue, fill=TolLightBlue!70},
1273 {draw=TolDarkGreen, fill=TolDarkGreen!70},
1274 }

```

mlineplot cycle Colors and styles intended for line charts with up to 4 series.

```

1275 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
1276 {TolDarkBlue, mark=*, mark size=1.5pt},
1277 {TolLightBrown, mark=square*, mark size=1.3pt},
1278 {TolLightGreen, mark=triangle*, mark size=1.5pt},
1279 {TolDarkBrown, mark=diamond*, mark size=1.5pt},
1280 }

```

However, the above cycle lists are not applied automatically. We still need to define styles — **mlineplot** and **mbarplot** — that the user can apply to the axis of a **pgfplots** chart to use the colors. We'll also take the opportunity to adjust the display of chart axes when these styles are used.

```

1281 \pgfplotsset{
1282 compat=1.9,

```

mlineplot A style to apply to the axis of a PGF line plot.

```

1283 mlineplot/.style={
1284 mbaseplot,
1285 xmajorgrids=true,
1286 ymajorgrids=true,
1287 major grid style={dotted},
1288 axis x line=bottom,
1289 axis y line=left,
1290 legend style={
1291 cells={anchor=west},
1292 draw=none
1293 },
1294 cycle list name=mlineplot cycle,
1295 },

```

mbarplot A style to apply to the axis of a PGF bar chart. **mbarplot** uses vertical bars by default, while **horizontal mbarplot** has horizontal bars as the name implies. Their shared properties are factored out into the internal style **mbarplot base**.


```

1296 mbarplot base/.style={
1297     mbaseplot,
1298     bar width=6pt,
1299     axis y line*=none,
1300 },
1301 mbarplot/.style={
1302     mbarplot base,
1303     ybar,
1304     xmajorgrids=false,
1305     ymajorgrids=true,
1306     area legend,
1307     legend image code/.code={%
1308         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1309     },
1310     cycle list name=mbarplot cycle,
1311 },
1312 horizontal mbarplot/.style={
1313     mbarplot base,
1314     xmajorgrids=true,
1315     ymajorgrids=false,
1316     xbar stacked,
1317     area legend,
1318     legend image code/.code={%
1319         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1320     },
1321     cycle list name=mbarplot cycle,
1322 },

```

mbaseplot Adjusts the appearance of the axes in a PGF chart.

```

1323 mbaseplot/.style={
1324     legend style={
1325         draw=none,
1326         fill=none,
1327         cells={anchor=west},
1328     },
1329     x tick label style={
1330         font=\footnotesize
1331     },
1332     y tick label style={

```

```

1333     font=\footnotesize
1334 },
1335 legend style={
1336     font=\footnotesize
1337 },
1338 major grid style={
1339     dotted,
1340 },
1341 axis x line*=bottom,
1342 },
1343 disable thousands separator/.style={
1344     /pgf/number format/.cd,
1345     1000 sep={}
1346 },
1347 }

```