

Modern Beamer Presentations with the **NEO** package

v1.0 — 2017/10/01

Contents

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

5.1	Backup Slides	9
6	Known Issues	10
6.1	Title formats	10
6.2	Interactions with other color themes	10
6.3	Notes on second screen	11
6.4	Standout frames with labels	11
6.5	Standout frames with Pandoc	12
7	License	12
8	Implementation	13
8.1	NEO parent theme	13
8.1.1	Package dependencies	13
8.1.2	Options	13
8.1.3	Component sub-packages	15
8.1.4	Custom commands	16
8.1.5	Process package options	17
8.2	NEO inner theme	17
8.2.1	Package dependencies	17
8.2.2	Options	17
8.2.3	Title page	18
8.2.4	Section page	21
8.2.5	Block environments	24
8.2.6	Lists and floats	26
8.2.7	Footnotes	26
8.2.8	Text and spacing settings	27
8.2.9	Standout frames	27
8.2.10	Process package options	29
8.3	NEO outer theme	29
8.3.1	Package dependencies	29
8.3.2	Options	29
8.3.3	Head and footline	32
8.3.4	Frametitle	34
8.3.5	Process package options	37
8.4	NEO font theme	37
8.4.1	Package dependencies	37
8.4.2	Load Fira fonts	38
8.4.3	General font definitions	40
8.4.4	Font style options	41
8.4.5	Title format options	41

8.4.6	Process package options	46
8.5	NEO color theme	47
8.5.1	Package dependencies	47
8.5.2	Options	47
8.5.3	Base colors	47
8.5.4	Alias colors	48
8.5.5	Base styles	48
8.5.6	Derived colors	49
8.5.7	Process package options	52
8.6	Tol pgfplots theme	53

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 Xe₂LaTeX 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 TeX 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
- pgfplots
- calc
- ifluatex

For best results, we recommend installing the fonts **Fira Sans** and Fira Mono and compiling with **NEO** using Xe_{La}TeX or Lua_{La}TeX. These are optional dependencies; **NEO** is compatible with (e.g.) pdf_{La}TeX 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:

```
\usepackage[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` *regular, smallcaps, allsmallcaps, allcaps* 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` *regular, smallcaps, allsmallcaps, allcaps* regular

Changes the format of “standout” frames (see `titleformat`, above).

3.1.2 Inner theme

`sectionpage` *none, simple, progressbar* 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.

`startsection` *hide, show* show

Controls whether a section page as the very first slide is shown (*show*) or hidden (*hide*).

`sectionaftertoc` *hide, show* show

Controls whether a section page directly after a table of contents frame is shown (*show*) or hidden (*hide*).

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, white* light

Provides the option to have a dark background and light foreground instead of the reverse. White is similar to light but uses a pure white background.

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.

`isable 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 `pgfplots-themetol` 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 pdf \TeX , 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** sub-packages 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 relation-

ship 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_YTeX, 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_YTeX itself. You can work around it either by compiling with LuaTeX 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   whitebg/.code=\pgfkeysalso{color/background=white},
53   blockbg/.code=\pgfkeysalso{color/block=fill},
54   light/.code=\pgfkeysalso{font/style=light},
55   book/.code=\pgfkeysalso{font/style=book},
56   regular/.code=\pgfkeysalso{font/style=regular},
57   nostartsection/.code=\pgfkeysalso{inner/startsection=hide},
58   nosectionaftertoc/.code=\pgfkeysalso{inner/sectionaftertoc=hide},
59 }

```

Set default values for options.

```

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

```

```

63 }
64 }

```

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.

```

65 \providecommand{\tikzexternalenable}{}
66 \providecommand{\tikzexternaldisable}{}

```

Neo is derived from metropolis, and does very, very nasty things, like using patchcmd all over the place. It also tries to patch \beamer@@frametitle, and is unhappy if it does not find an outdated implementation there. At the same time, the people developing beamer must be saints, because they included some workaround in their library, just to keep this stupid shit working. Otherwise, builds (from TexLive 2023 onwards) fail with:

```

! Package beamerfontthemeneo Error: Patching frame title failed.

```

(See also: <https://github.com/josephwright/beamer/issues/802>) However, their workaround specifically targets the filename ‘beamerfontthememetropolis.sty’, but someone(TM) thought it to be wise to rename the theme and all filenames and commands, so it does not trigger. And this is how we have come to this beauty:

```

67 \@ifclasslater{beamer}{2023/02/20}{%
68 % The hooks where first defined for v3.69
69 \AddToHook{file/beamerfontthemeneo.sty/before}{\UseHook{file/beamerfontthememetropolis.sty/before}}
70 \AddToHook{file/beamerfontthemeneo.sty/after}{%
71 \let\metropolis@frametitleformat\neo@frametitleformat
72 \UseHook{file/beamerfontthememetropolis.sty/after}}
73 }
74 }{}

```

8.1.3 Component sub-packages

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

```

75 \useinnertheme{neo}
76 \useoutertheme{neo}
77 \usecolortheme{neo}
78 \usefonttheme{neo}

```

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

```

79 \AtEndPreamble{%
80   \RequirePackage{pdfpc}
81   \newcommand<>{\pnote}[2][item]{%
82     \note[#1]#3{#2}%
83     \ifbool{neo@pdfpcnotes}{%
84       \only#3{{\escapechar='_ \xdef\tmpnote{\expandafter\detokenize\expandafter{#
85     }}}%
86   }
87   \@ifpackageloaded{pgfplots}{%
88     \RequirePackage{pgfplotsthemetol}
89   }{}
90 }

```

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.

```

91 \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.

```

92 \def\neo@plaintitleformat#1{#1}
93 \newcommand{\plain}[2][]{%
94   \PackageWarning{beamerthemeneo}{%
95     The syntax ‘\plain’ may be deprecated in a future version of neo.
96     Please use a frame with [standout] instead.
97   }
98   \begin{frame}[standout]{#1}
99     \neo@plaintitleformat{#2}
100   \end{frame}
101 }

```

`\mreducelistspacing`

```

102 \newcommand{\mreducelistspacing}{\vspace{-\topsep}}

```


8.1.5 Process package options

```
103 \neo@setdefaults
104 \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

```
105 \RequirePackage{etoolbox}
106 \RequirePackage{keyval}
107 \RequirePackage{calc}
108 \RequirePackage{pgfopts}
109 \RequirePackage{pgfpages}
110 \RequirePackage{tikz}
```

8.2.2 Options

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

```
111 \pgfkeys{
112   /neo/inner/sectionpage/.cd,
113   .is choice,
114   none/.code=\neo@disablesectionpage,
115   simple/.code={\neo@enablesectionpage
116                 \setbeamertemplate{section page}[simple]},
117   progressbar/.code={\neo@enablesectionpage
118                     \setbeamertemplate{section page}[progressbar]},
119 }
```

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

```
120 \pgfkeys{
121   /neo/inner/subsectionpage/.cd,
122   .is choice,
123   none/.code=\neo@disablesubsectionpage,
124   simple/.code={\neo@enablesubsectionpage
```

```

125         \setbeamertemplate{section page}[simple]],
126     progressbar/.code={\neo@enablesubsectionpage
127         \setbeamertemplate{section page}[progressbar]],
128 }

```

hide first section Optionally hide the first section slide right at the beginning of the talk.

```

129 \pgfkeys{
130   /neo/inner/startsection/.cd,
131   .is choice,
132   hide/.code={\global\booltrue{neo@hidestartsection}},
133   show/.code={\global\boolfalse{neo@hidestartsection}},
134 }

```

hide section after toc Optionally hide the first section slide after a table of contents.

```

135 \pgfkeys{
136   /neo/inner/sectionaftertoc/.cd,
137   .is choice,
138   hide/.code={\global\booltrue{neo@hidesectionaftertoc}},
139   show/.code={\global\boolfalse{neo@hidesectionaftertoc}},
140 }

```

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

```

141 \newcommand{\neo@inner@setdefaults}{
142   \pgfkeys{/neo/inner/.cd,
143     sectionpage=progressbar,
144     subsectionpage=none,
145     startsection=show,
146     sectionaftertoc=show,
147   }
148 }

```

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.

```

149 \setbeamertemplate{title page}{
150   \begin{minipage}[b][0.95\paperheight]{\textwidth}
151     \vfill%
152     \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
153     \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi

```

```
154 \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](#).

```
155 \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
156 \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
157 \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
158 \vfill
159 \begin{minipage}[b][0.25\paperheight][t]{\textwidth}
160 % The lower part of the title page background contains a white area which
161 % covers this whole minipage. Thus switch the text color back to normal
162 \neo@colors@light%
163 \usebeamercolor[fg]{normal text}%
164 \ifx\inserttitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
165 \end{minipage}
166 \end{minipage}
167 }
```

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` `\titlepage` template.

```
168 \renewcommand{\maketitle}[1][extern]{%
169 \ifbeamer@inframe
170 \titlepage
171 \else
172 {
173 \usebackgroundtemplate{
174 \tikzexternaldisable%
175 \begin{tikzpicture}
176 \node[anchor=north west,inner sep=0,outer sep=0] at (0, \paperheight) {
177 #1}};
177 \fill[nWhite] (0,0) rectangle (\paperwidth, 0.3\paperheight);
178 \end{tikzpicture}%
179 }
```

```

179     \tikzexternalenable%
180   }
181   \frame[plain,noframenumbering]{
182     \neo@colors@dark
183     \setbeamercolor{title separator}{
184       fg=black!20,
185       bg=normal text.fg
186     }
187     \titlepage
188   }
189 }
190 \fi
191 }
192 \def\titlepage{%
193   \usebeamertemplate{title page}
194 }

```

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

```

195 \setbeamertemplate{title graphic}{
196   \vbox to 0pt {
197     \vspace*{2em}
198     \inserttitlegraphic%
199   }%
200   \nointerlineskip%
201 }

```

title Set the title on the title page.

```

202 \setbeamertemplate{title}{
203   \raggedright%
204   \linespread{1.0}%
205   \inserttitle%
206   \par%
207   \vspace*{0.5em}
208 }

```

subtitle Set the subtitle on the title page.

```

209 \setbeamertemplate{subtitle}{
210   \raggedright%
211   \insertsubtitle%
212   \par%

```

```

213 \vspace*{0.5em}
214 }

```

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

```

215 \newlength{\neo@titleseparator@linewidth}
216 \setlength{\neo@titleseparator@linewidth}{0.4pt}
217 \setbeamertemplate{title separator}{
218   \tikzexternaldisable%
219   \begin{tikzpicture}
220     \fill[fg] (0,0) rectangle (\textwidth, \neo@titleseparator@linewidth);
221   \end{tikzpicture}%
222   \tikzexternalenable%
223   \par%
224 }

```

author Set the author on the title page.

```

225 \setbeamertemplate{author}{
226   \vspace*{2em}
227   \insertauthor%
228   \par%
229   \vspace*{0.25em}
230 }

```

date Set the date on the title page.

```

231 \setbeamertemplate{date}{
232   \insertdate%
233   \par%
234 }

```

institute Set the institute on the title page.

```

235 \setbeamertemplate{institute}{
236   \vspace*{3mm}
237   \insertinstitute%
238   \par%
239 }

```

8.2.4 Section page

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

```

240 \defbeamertemplate{section page}{simple}{
241   \begin{center}
242     \usebeamercolor[fg]{section title}
243     \usebeamerfont{section title}
244     \insertsectionhead\par
245     \ifx\insertsubsectionhead\@empty\else
246       \usebeamercolor[fg]{subsection title}
247       \usebeamerfont{subsection title}
248       \insertsubsectionhead
249     \fi
250   \end{center}
251 }
252 \defbeamertemplate{section page}{progressbar}{
253   \centering
254   \begin{minipage}{22em}
255     \raggedright
256     \usebeamercolor[fg]{section title}
257     \usebeamerfont{section title}
258     \insertsectionhead\[-1ex]
259     \usebeamertemplate*{progress bar in section page}
260     \par
261     \ifx\insertsubsectionhead\@empty\else%
262       \usebeamercolor[fg]{subsection title}%
263       \usebeamerfont{subsection title}%
264       \insertsubsectionhead
265     \fi
266   \end{minipage}
267   \par
268   \vspace{\baselineskip}
269 }
270
271 \global\providebool{neo@hidestartsection}
272 \global\providebool{neo@hidesectionaftertoc}
273 \global\providebool{neo@tocprinted}
274 \global\boolfalse{neo@tocprinted}
275 \let\neo@oldtableofcontents\tableofcontents
276 \renewcommand{\tableofcontents}{
277   \global\booltrue{neo@tocprinted}
278   \neo@oldtableofcontents
279 }
280
281 \newcommand{\neo@disablesectionpage}{
282   \AtBeginSection{

```

```

283 % intentionally empty
284 }
285 }
286 \newcommand{\neo@enablesectionpage}{
287 \AtBeginSection{
288 \ifbeamer@inframe
289 \sectionpage
290 \else
291 \providebool{neo@printsection}
292 \booltrue{neo@printsection}
293 \ifbool{neo@hidestartsection}{
294 \ifnum\theframenumber=0
295 \boolfalse{neo@printsection}
296 \fi
297 }{}
298 \ifbool{neo@hidesectionaftertoc}{
299 \ifbool{neo@tocprinted}{%
300 \global\boolfalse{neo@tocprinted}
301 \boolfalse{neo@printsection}
302 }{}
303 }{}
304 \ifbool{neo@printsection}{
305 \frame[plain,c,noframenumering]{\sectionpage}
306 }{}
307 \fi
308 }
309 }

```

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

```

310 \setbeamertemplate{subsection page}{%
311 \usebeamertemplate*{section page}
312 }
313 \newcommand{\neo@disablesubsectionpage}{
314 \AtBeginSubsection{
315 % intentionally empty
316 }
317 }
318 \newcommand{\neo@enablesubsectionpage}{
319 \AtBeginSubsection{
320 \ifbeamer@inframe
321 \subsectionpage
322 \else

```

```

323     \frame[plain,c,noframenumbering]{\subsectionpage}
324     \fi
325   }
326 }

```

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.

```

327 \newlength{\neo@progressonsectionpage}
328 \newlength{\neo@progressonsectionpage@linewidth}
329 \setlength{\neo@progressonsectionpage@linewidth}{0.4pt}
330 \setbeamertemplate{progress bar in section page}{
331   \pgfmathsetlength{\neo@progressonsectionpage}{\textwidth * min(1,\insertframenumber)}
332   \tikzexternaldisable%
333   \begin{tikzpicture}
334     \fill[bg] (0,0) rectangle (\textwidth, \neo@progressonsectionpage@linewidth);
335     \fill[fg] (0,0) rectangle (\neo@progressonsectionpage, \neo@progressonsectionpage@linewidth);
336   \end{tikzpicture}%
337   \tikzexternalenable%
338 }

```

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 block example macro `\neo@block` to define all three templates.

```

339 \newlength{\neo@blocksep}
340 \newlength{\neo@blockadjust}
341 \setlength{\neo@blocksep}{0.75ex}
342 \setlength{\neo@blockadjust}{0.25ex}
343 \providecommand{\neo@strut}{%
344   \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}}%
345 }
346 \newcommand{\neo@block}[1]{
347   \par\vskip\medskipamount%
348   \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 right-skip for a nice ragged-right block title.

```

349 \ifbeamercoloreempty[bg]{block title#1}{%
350   \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}}{%
351 \ifbeamercoloreempty[bg]{block title}{%
352   \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}%
353 }%
354 % \end{macrocode}
355 %
356 % Otherwise, if the |block title| has a background, we set the padding based
357 % on |\neo@blockskip|. However, we have to visually compensate for
358 % the |\neo@strut| added to the block title (see below) by
359 % subtracting |\neo@blockadjust| from the top and bottom padding.
360 %
361 % \begin{macrocode}
362 {%
363   \begin{beamercolorbox}[
364     sep=\dimexpr\neo@blocksep-\neo@blockadjust\relax,
365     leftskip=\neo@blockadjust,
366     rightskip=\dimexpr\neo@blockadjust plus 4em\relax
367   ]{block title#1}%
368 }}%
369 % \end{macrocode}
370 %
371 % We can now set the contents of the |block title|. The zero-
    width but
372 % positive-height box |\neo@strut| ensures that the block title box
373 % has a consistent height, even if it lacks punctuation, ascenders, or
374 % descenders.
375 %
376 % \begin{macrocode}
377   \usebeamerfont*{block title#1}%
378   \neo@strut%
379   \insertblocktitle%
380   \neo@strut%
```

```

381 \end{beamercolorbox}%
382 % \end{macrocode}
383 %
384 % Next, we typeset the |block body|. This the code is similar to, but simpler
385 % than, the |block title| code since we don't need to adjust for any struts.
386 %
387 % \begin{macrocode}
388 \nointerlineskip%
389 \ifbeamercoloreempty[bg]{block body#1}{%
390 \begin{beamercolorbox}[vmode]{block body#1}}{
391 \ifbeamercoloreempty[bg]{block body}{%
392 \begin{beamercolorbox}[vmode]{block body#1}%
393 }}{%
394 \begin{beamercolorbox}[sep=\neo@blocksep, vmode]{block body#1}%
395 \vspace{-\neo@parskip}
396 }}%
397 \usebeamerfont{block body#1}%
398 \setlength{\parskip}{\neo@parskip}%
399 }

```

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

```

400 \setbeamertemplate{block begin}{\neo@block{}}
401 \setbeamertemplate{block alerted begin}{\neo@block{ alerted}}
402 \setbeamertemplate{block example begin}{\neo@block{ example}}
403 \setbeamertemplate{block end}{\end{beamercolorbox}\vspace*{0.2ex}}
404 \setbeamertemplate{block alerted end}{\end{beamercolorbox}\vspace*{0.2ex}}
405 \setbeamertemplate{block example end}{\end{beamercolorbox}\vspace*{0.2ex}}

```

8.2.6 Lists and floats

```

406 \setbeamertemplate{itemize items}{\raise1pt\hbox{\vrule width 0.8ex height 0.8ex}}
407 \setbeamertemplate{itemize subitem}{\raise1pt\hbox{\vrule width 0.5ex height 0.5ex}}
408 \setbeamertemplate{itemize subsubitem}{\raise.5ex\hbox{\vrule width 1ex height 0.5ex}}
409 \defbeamertemplate{description item}{align left}{\insertdescriptionitem\hfill}
410 \setbeamertemplate{caption label separator}{: }
411 \setbeamertemplate{caption}[numbered]

```

8.2.7 Footnotes

```

412 \setbeamertemplate{footnote}{%
413 \parindent 0em\noindent%
414 \raggedright
415 \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\hangindent=0.8em}

```

```
416 }
```

8.2.8 Text and spacing settings

```
417 \newlength{\neo@parskip}
418 \setlength{\neo@parskip}{0.5em}
419 \setlength{\parskip}{\neo@parskip}
420 \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](#).

```
421 \define@key{beamerframe}{c}[true]{% centered
422   \beamer@frametopskip=0pt plus 1fill\relax%
423   \beamer@framebottomskip=0pt plus 1fill\relax%
424   \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
425   \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
426   \def\beamer@initfirstlineunskip{}}%
427 }
```

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.

```
428 \providebool{neo@standout}
429 \define@key{beamerframe}{standout}[true]{%
430   \booltrue{neo@standout}
431   \begingroup
432     \setkeys{beamerframe}{c,plain}
433     \ifbeamercolorempy[bg]{palette primary}{
434       \setbeamercolor{background canvas}{
435         use=palette primary,
436         bg=-palette primary.fg
437       }
438     }{
439       \setbeamercolor{background canvas}{
```

```

440         use=palette primary,
441         bg=palette primary.bg
442     }
443 }
444 \setbeamercolor{local structure}{
445     fg=palette primary.fg
446 }
447 \usebeamercolor[fg]{palette primary}
448 \makeatletter
449 \def\beamer@framesnotesbegin{% at beginning of slide
450     \usebeamercolor[fg]{palette primary}
451     \gdef\beamer@noteitems{}%
452     \gdef\beamer@notes{}%
453 }
454 \makeatother
455 }

```

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.

```

456 \pretocmd{\beamer@reseteecodes}{%
457     \ifbool{neo@standout}{
458         \endgroup
459         \boolfalse{neo@standout}
460     }{}
461 }{}{}

```

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.

```

462 \AtBeginEnvironment{beamer@frameslide}{
463     \makeatletter
464     \usebeamercolor[fg]{normal text}
465     \gdef\beamer@noteitems{}%
466     \gdef\beamer@notes{}%
467     \makeatother
468     \ifbool{neo@standout}{
469         \centering
470         \usebeamerfont{standout}
471     }{}

```

```
472 }
```

8.2.10 Process package options

```
473 \neo@inner@setdefaults
474 \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

```
475 \RequirePackage{etoolbox}
476 \RequirePackage{calc}
477 \RequirePackage{pgfpages}
478 \RequirePackage{pgfopts}
```

8.3.2 Options

`icon` Adds an icon to the frametitle on each slide.

```
479 \pgfkeys{
480   /neo/outer/frametitle icon/.cd,
481   .is choice,
482   none/.code=\setbeamertemplate{frametitle icon}[none],
483   i4/.code=\setbeamertemplate{frametitle icon}[i4],
484   fau/.code=\setbeamertemplate{frametitle icon}[fau],
485   fau-new/.code=\setbeamertemplate{frametitle icon}[fau-new],
486 }
```

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

```
487 \pgfkeys{
488   /neo/outer/numbering/.cd,
489   .is choice,
490   none/.code=\setbeamertemplate{frame numbering}[none],
491   counter/.code=\setbeamertemplate{frame numbering}[counter],
492   fraction/.code=\setbeamertemplate{frame numbering}[fraction],
493 }
```

`notes` Show notes in presentation

```
494 \providebool{neo@pdfpcnotes}
495 \pgfkeys{
```

```

496 /neo/outer/notes/.cd,
497 .is choice,
498 none/.code=\pgfkeysalso{notes=hide},
499 hide/.code={\boolfalse{neo@pdfpcnotes}\setbeameroption{hide notes}},
500 pdfpc/.code={\booltrue{neo@pdfpcnotes}\setbeameroption{hide notes}},
501 show/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\se
502 full/.code={\booltrue{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\set
503 only/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\se
504 preview-left/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[c
505 preview-right/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[
506 preview-top/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[de
507 preview-bottom/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}
508 preview-left-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note pag
big]\setbeameroption{show notes on second screen=left}},
509 preview-right-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note pa
big]\setbeameroption{show notes on second screen=right}},
510 preview-top-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page
big]\setbeameroption{show notes on second screen=top}},
511 preview-bottom-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note p
big]\setbeameroption{show notes on second screen=bottom}},
512 left/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\se
513 right/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\s
514 top/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\set
515 bottom/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\
516 }

```

footer Adds additional presentation information to the footer

```

517 \pgfkeys{
518 /neo/outer/footer/.cd,
519 .is choice,
520 none/.code=\setbeamertemplate{frame footer}[none],
521 author/.code=\setbeamertemplate{frame footer}[author],
522 author title/.code=\setbeamertemplate{frame footer}[author title],
523 author title section/.code=\setbeamertemplate{frame footer}[author title sect
524 title/.code=\setbeamertemplate{frame footer}[title],
525 title section/.code=\setbeamertemplate{frame footer}[title section],
526 title section subsection/.code=\setbeamertemplate{frame footer}[title section
527 }

```

footer style Footer background color

```

528 \providebool{neo@standoutfooter}
529 \pgfkeys{

```

```

530 /neo/outer/footer style/.cd,
531 .is choice,
532 plain/.code={\boolfalse{neo@standoutfooter}\setbeamertemplate{footline}[plain]}
533 standout/.code={\booltrue{neo@standoutfooter}\setbeamertemplate{footline}[standout]}
534 }

```

`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).

```

535 \pgfkeys{
536 /neo/outer/progressbar/.cd,
537 .is choice,
538 none/.code={%
539   \setbeamertemplate{headline}[plain]
540   \setbeamertemplate{frametitle}[plain]
541 },
542 head/.code={\pgfkeys{/neo/outer/progressbar=none}
543   \addtobeamertemplate{headline}{}{%
544     \usebeamertemplate*{progress bar in head/footer}
545   }
546 },
547 frametitle/.code={\pgfkeys{/neo/outer/progressbar=none}
548   \addtobeamertemplate{frametitle}{}{%
549     \usebeamertemplate*{progress bar in head/footer}
550   }
551 },
552 foot/.code={\pgfkeys{/neo/outer/progressbar=none}
553   \ifbool{neo@standoutfooter}{%
554     \addtobeamertemplate{footline}{\usebeamertemplate*{progress bar in head/footer}}{}
555   }{%
556     \addtobeamertemplate{footline}{}{\usebeamertemplate*{progress bar in head/footer}}
557   }
558 },
559 }

```

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

```

560 \newcommand{\neo@outer@setdefaults}{
561   \pgfkeys{/neo/outer/.cd,
562     frametitle icon=none,
563     footer=none,
564     footer style=plain,
565     numbering=counter,

```

```

566     progressbar=none,
567   }
568 }

```

8.3.3 Head and footline

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

```

569 \setbeamertemplate{navigation symbols}{}

```

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

```

570 \defbeamertemplate{frametitle icon}{none}{}
571 \defbeamertemplate{frametitle icon}{i4}{ \hfill\raisebox{-.25\height}{\includegraphics[width=1cm]{i4-white}}}
572 \defbeamertemplate{frametitle icon}{fau}{ \hfill\raisebox{-.25\height}{\includegraphics[width=1cm]{fau-white}}\kern-0.275em}

```

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

```

573 \defbeamertemplate{frame footer}{none}{}
574 \defbeamertemplate{frame footer}{author}{
575   \insertshortauthor%
576 }
577 \defbeamertemplate{frame footer}{author title}{
578   \insertshortauthor%
579   \hfill%
580   \insertshorttitle%
581   \hfill%
582 }
583 \defbeamertemplate{frame footer}{author title section}{
584   \insertshortauthor%
585   \hfill%
586   \insertshorttitle%
587   \hfill%
588   \insertsection%
589   \hfill%
590 }
591 \defbeamertemplate{frame footer}{title}{
592   \insertshorttitle%
593 }
594 \defbeamertemplate{frame footer}{title section}{

```



```

595 \insertshorttitle%
596 \hfill%
597 \insertsection%
598 \hfill%
599 }
600 \defbeamertemplate{frame footer}{title section subsection}{
601 \insertshorttitle%
602 \hfill%
603 \insertsection%
604 \ifx\insertsubsection\@empty\else\ -- \insertsubsection\fi%
605 \hfill%
606 }
607 \defbeamertemplate{frame footer}{custom}[1]{ #1 }

```

Add strut to ensure that frame numbers don't jump

```

608 \newcommand{\neo@framenumberingstrut}{\vphantom{0123456789}}
609 \defbeamertemplate{frame numbering}{none}{}
610 \defbeamertemplate{frame numbering}{counter}{\neo@framenumberingstrut\insertframe
611 \defbeamertemplate{frame numbering}{fraction}{
612 \neo@framenumberingstrut\insertframenumber/\inserttotalframenumber
613 }

```

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

footline

```

614 \defbeamertemplate{headline}{plain}{}
615 \defbeamertemplate{footline}{plain}{%
616 \begin{beamercolorbox}[wd=\textwidth, sep=1ex]{footline}%
617 \usebeamerfont{page number in head/foot}%
618 \usebeamertemplate*{frame footer}
619 \hfill%
620 \parbox{.1\framewidth}{\hfill\usebeamertemplate*{frame numbering}}
621 \end{beamercolorbox}%
622 }
623 \defbeamertemplate{footline}{standout}{%
624 \begin{beamercolorbox}[wd=\textwidth, sep=1ex]{palette primary}%
625 \usebeamerfont{page number in head/foot}%
626 \usebeamertemplate*{frame footer}
627 \hfill%
628 \parbox{.1\framewidth}{\hfill\usebeamertemplate*{frame numbering}}
629 \end{beamercolorbox}%
630 }

```

8.3.4 Frametitle

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

```

631 \newlength{\neo@frametitle@padding}
632 \setlength{\neo@frametitle@padding}{2.2ex}
633 \newcommand{\neo@frametitlestrut@start}{
634   \rule{0pt}{\neo@frametitle@padding +%
635     \totalheightof{%
636       \ifcsdef{neo@frametitleformat}{\neo@frametitleformat X}{X}%
637     }%
638   }%
639 }
640 \newcommand{\neo@frametitlestrut@end}{
641   \rule[-\neo@frametitle@padding]{0pt}{\neo@frametitle@padding}
642 }
643 \defbeamertemplate{frametitle}{plain}{%
644   \nointerlineskip%
645   \begin{beamercolorbox}[%
646     wd=\paperwidth,%
647     sep=0pt,%
648     leftskip=\neo@frametitle@padding,%
649     rightskip=\neo@frametitle@padding,%
650   ]{frametitle}%
651   \neo@frametitlestrut@start%
652   \insertframetitle%
653   \ifx\insertframesubtitle\@empty
654     \usebeamertemplate*{frametitle icon}%
655   \else
656     \hfill{\usebeamerfont{framesubtitle}\insertframesubtitle}%
657   \fi
658   \nolinebreak%
659   \neo@frametitlestrut@end%
660   \end{beamercolorbox}%
661 }
662 \setbeamertemplate{frametitle continuation}{%
663   \usebeamerfont{frametitle}
664   {\normalfont (\insertcontinuationcount)}
665 }

```

`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.

```
666 \newlength{\neo@progressinheadfoot}
667 \newlength{\neo@progressinheadfoot@linewidth}
668 \setlength{\neo@progressinheadfoot@linewidth}{0.8pt}
669 \setbeamertemplate{progress bar in head/foot}{
670   \nointerlineskip
671   \pgfmathsetlength{\neo@progressinheadfoot}{\paperwidth * min(1,\insertframenum)}
672   \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
673     \tikzexternaldisable%
674     \begin{tikzpicture}
675       \fill[bg] (0,0) rectangle (\paperwidth, \neo@progressinheadfoot@linewidth);
676       \fill[fg] (0,0) rectangle (\neo@progressinheadfoot, \neo@progressinheadfoot);
677     \end{tikzpicture}%
678     \tikzexternalenable%
679   \end{beamercolorbox}
680 }
```

custom notes Templates for note pages

```
681 \defbeamertemplate{note page}{preview-big}
682 {%
683   {%
684     \scriptsize
685     \usebeamerfont{note title}\usebeamercolor[fg]{note title}%
686     \ifbeamercoloreempty[bg]{note title}{}{%
687       \insertvrule{.45\paperheight}{note title.bg}%
688       \vskip-.45\paperheight%
689       \nointerlineskip%
690     }%
691     \vbox{
692       \hfill\insertslideintonotes{0.45}\hskip-\Gm@rmargin\hskip0pt%
693       \vskip-0.45\paperheight%
694       \nointerlineskip
695       \begin{pgfpicture}{0cm}{0cm}{0cm}{0cm}
696         \begin{pgflowlevelscope}{\pgftransformrotate{90}}
697           {\pgftransformshift{\pgfpoint{-2cm}{0.2cm}}}%
698           \pgftext[base,left]{\usebeamerfont{note date}\usebeamercolor[fg]{note date}}
699           \ifnum\month<10\relax0\fi\the\month-\ifnum\day<10\relax0\fi\the\day}}
700       \end{pgflowlevelscope}
701     \end{pgfpicture}}
702     \nointerlineskip
703     \vbox to .45\paperheight{\vskip0.5em
704       \hbox{\insertshorttitle[width=8cm]}}%
```

```

704     \setbox\beamer@tempbox=\hbox{\insertsection}%
705     \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip4pt\raise3pt\hbox{\vrule
706         width0.4pt height7pt\vrule width 9pt
707         height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere
708     }%
709     \setbox\beamer@tempbox=\hbox{\insertsubsection}%
710     \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip17.4pt\raise3pt\hbox{\vrule
711         width0.4pt height7pt\vrule width 9pt
712         height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere
713     }%
714     \setbox\beamer@tempbox=\hbox{\insertshortframetitle}%
715     \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip30.8pt\raise3pt\hbox{\vrule
716         width0.4pt height7pt\vrule width 9pt
717         height0.4pt}}\hskip1pt\hbox{\insertshortframetitle[width=7cm]}\fi
718     }%
719     \vfil}%
720 }%
721 \ifbeamercoloreempty[bg]{note page}{}{%
722     \nointerlineskip%
723     \insertvrule{.55\paperheight}{note page.bg}%
724     \vskip-.55\paperheight%
725 }%
726 \vskip.25em
727 \nointerlineskip
728 \insertnote
729 }
730 \defbeamertemplate{note page}{print}
731 {%
732     {%
733         \nointerlineskip%
734         \begin{beamercolorbox}[%
735             wd=\paperwidth,%
736             sep=0pt,%
737             leftskip=\neo@frametitle@padding,%
738             rightskip=\neo@frametitle@padding,%
739         ]{note title}%
740         \usebeamerfont{frametitle}%
741         \neo@frametitlestrut@start%
742         \insertframetitle%
743         \ifx\insertframesubtitle\@empty
744             \usebeamertemplate*{frametitle icon}%
745         \else
746             \hfill{\usebeamerfont{framesubtitle}\insertframesubtitle}%

```

```

747 \fi
748 \nolinebreak%
749 \neo@frametitlestrut@end%
750 \end{beamercolorbox}%
751 }%
752 \insertnote%
753 \vfill%
754 \begin{beamercolorbox}[wd=\paperwidth, sep=3ex]{footline}%
755 \usebeamerfont{page number in head/foot}%
756 \usebeamertemplate*{frame footer}
757 \hfill%
758 \usebeamertemplate*{frame numbering}
759 \end{beamercolorbox}%
760 \vskip4pt%
761 }

```

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`.

```

762 \AtBeginDocument{%
763 \apptocmd{\appendix}{%
764 \pgfkeys{%
765 /neo/outer/.cd,
766 numbering=none,
767 progressbar=none}
768 }{}{}
769 }

```

8.3.5 Process package options

```

770 \neo@outer@setdefaults
771 \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

```

772 \RequirePackage{etoolbox}
773 \RequirePackage{ifxetex}
774 \RequirePackage{ifluatex}
775 \RequirePackage{pgfopts}

```

8.4.2 Load Fira fonts

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

```
776 \ifboolexpr{bool {xetex} or bool {luatex}}{
777   \@ifpackageloaded{fontspec}{
778     \PassOptionsToPackage{no-math}{fontspec}
779   }{
780     \RequirePackage[no-math]{fontspec}
781   }

782   \IfFontExistsTF{FiraSans-Regular.otf}{
783     \defaultfontfeatures{
784       Scale      = 1.0,
785       Extension = .otf
786     }
787   }{
788     \PackageWarning{beamerthemeneo}{%
789       FiraSans font not found in path, therefore using system fonts. %
790       Make sure you have the fonts installed.%
791     }
792   }
793   \setmonofont
794     [ Numbers = {Monospaced,Lining},
795       UprightFont    = *-Regular ,
796       ItalicFont     = *-Regular ,
797       BoldFont       = *-Medium ,
798       BoldItalicFont = *-Medium ,
799     ]
800     {FiraMono}
801   \newcommand{\neo@fontsave}{
802     \let\firaneofamily\sfdefault
803     \renewcommand*\familydefault{\firaneofamily}
804   }
805   \newcommand{\neo@fontlight}{
806     \setsansfont[
807       Numbers = {OldStyle, Monospaced},
808       UprightFont    = *-Light ,
809       ItalicFont     = *-LightItalic ,
810       BoldFont       = *-Regular ,
811       BoldItalicFont = *-Italic ,
812     ]{FiraSans}
813     \neo@fontsave
```

```

814 }
815 \newcommand{\neo@fontbook}{
816   \setsansfont[
817     Numbers = {OldStyle, Monospaced},
818     UprightFont = *-Book ,
819     ItalicFont = *-BookItalic ,
820     BoldFont = *-Medium ,
821     BoldItalicFont = *-MediumItalic ,
822   ]{FiraSans}
823   \neo@fontsave
824 }
825 \newcommand{\neo@fontregular}{
826   \setsansfont[
827     Numbers = {OldStyle, Monospaced},
828     UprightFont = *-Regular ,
829     ItalicFont = *-Italic ,
830     BoldFont = *-SemiBold ,
831     BoldItalicFont = *-SemiBoldItalic ,
832   ]{FiraSans}
833   \neo@fontsave
834 }
835 \AtBeginEnvironment{tabular}{%
836   \addfontfeature{Numbers={Monospaced}}%
837 }
838 }{%
839   \RequirePackage[utf8]{inputenc}
840   \IfFileExists{FiraSans.sty}{
841     \RequirePackage[T1]{fontenc}
842     \RequirePackage[sfdefault]{FiraSans}
843     \RequirePackage[nomap,lining]{FiraMono}
844     \def\bfseries@tt{mb}
845     \newcommand{\neo@fontsave}{
846       \edef\familydefault{\sfdefault}
847       \edef\seriesdefault{\mdseries@sf}
848     }
849     \newcommand{\neo@fontlight}{
850       \def\mdseries@sf{l}
851       \def\bfseries@sf{m}
852       \neo@fontsave
853     }
854     \newcommand{\neo@fontbook}{
855       \def\mdseries@sf{sl}
856       \def\bfseries@sf{medium}

```

```

857     \neo@fontsave
858   }
859   \newcommand{\neo@fontregular}{
860     \def\mdseries@sf{sl}
861     \def\bfseries@sf{sb}
862     \neo@fontsave
863   }
864 }{
865   \PackageWarning{beamerthemeneo}{%
866     You need to install the Fira Fonts package or compile with XeLaTeX or %
867     LuaLaTeX to use the included Fira fonts%
868   }
869 }
870 }

```

This concludes the portion of the code which is only run when compiled with Xe_ΛTeX or Lua_ΛTeX. The remainder of this package applies regardless of the compiling engine.

8.4.3 General font definitions

```

871 \setbeamerfont{title}{size=\Large,%
872               series=\bfseries}
873 \setbeamerfont{author}{size=\small}
874 \setbeamerfont{date}{size=\small}
875 \setbeamerfont{section title}{size=\Large,%
876               series=\bfseries}
877 \setbeamerfont{block title}{size=\normalsize,%
878               series=\bfseries}
879 \setbeamerfont{block title alerted}{size=\normalsize,%
880               series=\bfseries}
881 \setbeamerfont*{subtitle}{size=\large}
882 \setbeamerfont{frametitle}{size=\large,%
883               series=\bfseries}
884 \setbeamerfont{framesubtitle}{parent=frametitle,%
885               size=\footnotesize}
886 \setbeamerfont{caption}{size=\small}
887 \setbeamerfont{caption name}{series=\bfseries}
888 \setbeamerfont{description item}{series=\bfseries}
889 \setbeamerfont{page number in head/foot}{size=\scriptsize}
890 \setbeamerfont{bibliography entry author}{size=\normalsize,%
891               series=\normalfont}
892 \setbeamerfont{bibliography entry title}{size=\normalsize,%
893               series=\bfseries}

```



```

894 \setbeamerfont{bibliography entry location}{size=\normalsize,%
895                                     series=\normalfont}
896 \setbeamerfont{bibliography entry note}{size=\small,%
897                                     series=\normalfont}
898 \setbeamerfont{standout}{size=\Large,%
899                                     series=\bfseries}

```

8.4.4 Font style options

`titleformat title` Controls the overall font style.

```

900 \pgfkeys{
901   /neo/font/style/.cd,
902   .is choice,
903   light/.code={\neo@fontlight},
904   book/.code={\neo@fontbook},
905   regular/.code={\neo@fontregular},
906 }

```

8.4.5 Title format options

`titleformat title` Controls the format of the title.

```

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

```

```

929         Be aware that titleformat title=allcaps can lead to problems%
930     }
931 },
932 }

```

`titleformat subtitle` Control the format of the subtitle.

```

933 \pgfkeys{
934   /neo/font/titleformat subtitle/.cd,
935   .is choice,
936   regular/.code={%
937     \let\neo@subtitleformat\@empty%
938     \setbeamerfont{subtitle}{shape=\normalfont}%
939   },
940   smallcaps/.code={%
941     \let\neo@subtitleformat\@empty%
942     \setbeamerfont{subtitle}{shape=\scshape}%
943   },
944   allsmallcaps/.code={%
945     \let\neo@subtitleformat\lowercase%
946     \setbeamerfont{subtitle}{shape=\scshape}%
947     \PackageWarning{beamerthemeneo}{%
948       Be aware that titleformat subtitle=allsmallcaps can lead to problems%
949     }
950   },
951   allcaps/.code={%
952     \let\neo@subtitleformat\uppercase%
953     \setbeamerfont{subtitle}{shape=\normalfont}%
954     \PackageWarning{beamerthemeneo}{%
955       Be aware that titleformat subtitle=allcaps can lead to problems%
956     }
957   },
958 }

```

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

```

959 \pgfkeys{
960   /neo/font/titleformat section/.cd,
961   .is choice,
962   regular/.code={%
963     \let\neo@sectiontitleformat\@empty%
964     \setbeamerfont{section title}{shape=\normalfont}%
965   },
966   smallcaps/.code={%

```

```

967     \let\neo@sectiontitleformat\@empty%
968     \setbeamerfont{section title}{shape=\scshape}%
969 },
970 allsmallcaps/.code={%
971     \let\neo@sectiontitleformat\MakeLowercase%
972     \setbeamerfont{section title}{shape=\scshape}%
973     \PackageWarning{beamerthemeneo}{%
974         Be aware that titleformat section=allsmallcaps can lead to problems%
975     }
976 },
977 allcaps/.code={%
978     \let\neo@sectiontitleformat\MakeUppercase%
979     \setbeamerfont{section title}{shape=\normalfont}%
980     \PackageWarning{beamerthemeneo}{%
981         Be aware that titleformat section=allcaps can lead to problems%
982     }
983 },
984 }

```

frametitleformat Control the format of the frame title.

```

985 \pgfkeys{
986   /neo/font/titleformat frame/.cd,
987   .is choice,
988   regular/.code={%
989     \let\neo@frametitleformat\@empty%
990     \setbeamerfont{frametitle}{shape=\normalfont}%
991   },
992   smallcaps/.code={%
993     \let\neo@frametitleformat\@empty%
994     \setbeamerfont{frametitle}{shape=\scshape}%
995   },
996   allsmallcaps/.code={%
997     \let\neo@frametitleformat\MakeLowercase%
998     \setbeamerfont{frametitle}{shape=\scshape}%
999     \PackageWarning{beamerthemeneo}{%
1000         Be aware that titleformat frame=allsmallcaps can lead to problems%
1001     }
1002   },
1003   allcaps/.code={%
1004     \let\neo@frametitleformat\MakeUppercase%
1005     \setbeamerfont{frametitle}{shape=\normalfont}%
1006     \PackageWarning{beamerthemeneo}{%
1007         Be aware that titleformat frame=allcaps can lead to problems%

```

```

1008     }
1009   },
1010 }

```

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

```

1011 \pgfkeys{
1012   /neo/font/.cd,
1013   titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
1014   titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
1015   titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
1016   titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
1017 }

```

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

```

1018 \newcommand{\neo@font@setdefaults}{
1019   \pgfkeys{/neo/font/.cd,
1020     style=book,
1021     titleformat title=regular,
1022     titleformat subtitle=regular,
1023     titleformat section=regular,
1024     titleformat frame=regular,
1025   }
1026 }

```

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

```

1027 \def\neo@titleformat#1{#1}
1028 \def\neo@subtitleformat#1{#1}
1029 \def\neo@sectiontitleformat#1{#1}
1030 \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](#).

```

1031 \patchcmd{\beamer@title}%
1032   {\def\inserttitle{#2}}%
1033   {\def\inserttitle{\neo@titleformat{#2}}}%
1034   {}%
1035   {\PackageError{beamerfontthemeneo}{Patching title failed}\@ehc}
1036 \patchcmd{\beamer@subtitle}%

```

```

1037 {\def\insertsubtitle{#2}}%
1038 {\def\insertsubtitle{\neo@subtitleformat{#2}}}%
1039 {}%
1040 {\PackageError{beamerfontthemeneo}{Patching subtitle failed}\@ehc}
1041 \patchcmd{\sectionentry}
1042 {\def\insertsectionhead{#2}}
1043 {\def\insertsectionhead{\neo@sectiontitleformat{#2}}}
1044 {}
1045 {\PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc}
1046 \@tempswafalse
1047 \patchcmd{\beamer@section}
1048 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded
1049 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1050 \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
1051 {\@tempswatrue}
1052 {}
1053 \patchcmd{\beamer@section}
1054 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
1055 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
1056 \neo@sectiontitleformat{#1}}}}
1057 {\@tempswatrue}
1058 {}
1059 \patchcmd{\beamer@section}
1060 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded
1061 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1062 \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
1063 {\@tempswatrue}
1064 {}
1065 \patchcmd{\beamer@section}
1066 {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#
1067 {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1068 \noexpand\neo@sectiontitleformat{#1}}}}}
1069 {\@tempswatrue}
1070 {}
1071 \if@tempswa\else
1072 \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
1073 \fi
1074 \@tempswafalse
1075 \patchcmd{\beamer@subsection}
1076 {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded
1077 {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1078 \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
1079 {\@tempswatrue}

```

```

1080 {}
1081 \patchcmd{\beamer@section}
1082 {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
1083 {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{%
1084   \neo@sectiontitleformat{#1}}}}
1085 {\@tempswatrue}
1086 {}
1087 \patchcmd{\beamer@section}
1088 {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}
1089 {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}
1090   \noexpand\neo@sectiontitleformat{#1}}}}
1091 {\@tempswatrue}
1092 {}
1093 \if@tempswa\else
1094   \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
1095 \fi

```

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

```

1096 \patchcmd{\beamer@@frametitle}
1097 {{%
1098   \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}}\space%
1099   \usebeamertemplate*{frametitle continuation}\fi}}%
1100 \gdef\beamer@frametitle{#2}%
1101 \gdef\beamer@shortframetitle{#1}%
1102 }}
1103 {{%
1104   \gdef\insertframetitle{{\neo@frametitleformat{#2}\ifnum%
1105     \beamer@autobreakcount>0\relax{}}\space%
1106   \usebeamertemplate*{frametitle continuation}\fi}}%
1107 \gdef\beamer@frametitle{#2}%
1108 \gdef\beamer@shortframetitle{#1}%
1109 }}
1110 {}
1111 {\PackageError{beamerfontthemeneo}{Patching frame title failed}\@ehc}

```

8.4.6 Process package options

```

1112 \neo@font@setdefaults
1113 \ProcessPgfpPackageOptions{/neo/font}

```

8.5 NEO color theme

8.5.1 Package dependencies

```
1114 \RequirePackage{pgfopts}
```

8.5.2 Options

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

```
1115 \pgfkeys{
1116   /neo/color/block/.cd,
1117   .is choice,
1118   transparent/.code=\neo@block@transparent,
1119   fill/.code=\neo@block@fill,
1120 }
```

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

```
1121 \pgfkeys{
1122   /neo/color/background/.cd,
1123   .is choice,
1124   dark/.code=\neo@colors@dark,
1125   light/.code=\neo@colors@light,
1126   white/.code=\neo@colors@white,
1127 }
```

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

```
1128 \newcommand{\neo@color@setdefaults}{
1129   \pgfkeys{/neo/color/.cd,
1130     background=light,
1131     block=transparent,
1132   }
1133 }
```

8.5.3 Base colors

```
1134
1135 \definecolor{nDarkGrey}{RGB}{152,164,174}
1136 \definecolor{nGrey}{RGB}{210,213,215}
1137 \definecolor{nLightGrey}{RGB}{235,236,238}
1138
1139 \definecolor{nDarkRed}{RGB}{141,20,41}
```

```

1140 \definecolor{nRed}{RGB}{201,169,147}
1141 \definecolor{nLightRed}{RGB}{237,231,222}
1142
1143 \definecolor{nDarkGreen}{RGB}{0,155,119}
1144 \definecolor{nGreen}{RGB}{170,207,189}
1145 \definecolor{nLightGreen}{RGB}{229,239,234}
1146
1147 \definecolor{nDarkBlue}{RGB}{0,56,101}
1148 \definecolor{nBlue}{RGB}{144,167,198}
1149 \definecolor{nLightBlue}{RGB}{221,229,240}
1150
1151 \definecolor{nDarkCyan}{RGB}{0,177,235}
1152 \definecolor{nCyan}{RGB}{180,214,245}
1153 \definecolor{nLightCyan}{RGB}{234,243,252}
1154
1155 \definecolor{nDarkYellow}{RGB}{201,147,19}
1156 \definecolor{nYellow}{RGB}{217,198,137}
1157 \definecolor{nLightYellow}{RGB}{243,238,223}
1158
1159 \definecolor{nBlack}{HTML}{011F32}
1160 \definecolor{nWhite}{RGB}{250,250,250}

```

8.5.4 Alias colors

Support the colors provided by the old i4 beamer theme.

```

1161 \colorlet{i4red}{nDarkRed}
1162 \colorlet{i4green}{nDarkGreen}
1163 \colorlet{i4blue}{nDarkBlue}
1164 \colorlet{i4cyan}{nDarkCyan}
1165 \colorlet{i4yellow}{nDarkYellow}
1166 \colorlet{i4grey}{nDarkGrey}
1167 \definecolor{darkred}{rgb}{0.8,0,0}
1168 \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`.

```

1169 \newcommand{\neo@colors@dark}{
1170   \setbeamercolor{normal text}{%
1171     fg=nWhite,
1172     bg=nBlack
1173   }

```



```

1174 \setbeamercolor{normal item}{%
1175     fg=nWhite,
1176     bg=nDarkBlue
1177 }
1178 \usebeamercolor[fg]{normal text}
1179 }
1180 \newcommand{\neo@colors@light}{
1181     \setbeamercolor{normal text}{%
1182         fg=nBlack,
1183         bg=nWhite
1184     }
1185     \setbeamercolor{normal item}{%
1186         fg=nDarkBlue,
1187         bg=nWhite
1188     }
1189 }
1190 \newcommand{\neo@colors@white}{
1191     \definecolor{nWhite}{RGB}{255,255,255}
1192     \neo@colors@light
1193 }
1194 \setbeamercolor{alerted text}{%
1195     fg=nDarkRed
1196 }
1197 \setbeamercolor{example text}{%
1198     fg=nDarkYellow
1199 }
1200 \setbeamercolor{note title}{%
1201     fg=nDarkBlue,
1202     bg=nGrey
1203 }
1204 \setbeamercolor{note page}{%
1205     fg=nBlack,
1206     bg=nLightGrey
1207 }

```

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`.

```

1208 \setbeamercolor{titlelike}{use=normal text, parent=normal text}
1209 \setbeamercolor{author}{use=normal text, parent=normal text}
1210 \setbeamercolor{date}{use=normal text, parent=normal text}
1211 \setbeamercolor{institute}{use=normal text, parent=normal text}
1212 \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.

```

1213 \setbeamercolor{palette primary}{%
1214   use=normal text,
1215   fg=normal text.bg,
1216   bg=nDarkBlue
1217 }
1218 \setbeamercolor{frametitle}{%
1219   use=palette primary,
1220   parent=palette primary
1221 }

```

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`.

```

1222 \setbeamercolor{progress bar}{%
1223   use=normal text,
1224   fg=nDarkBlue,
1225   bg=nLightBlue
1226 }
1227 \setbeamercolor{title separator}{
1228   use=progress bar,
1229   parent=progress bar
1230 }
1231 \setbeamercolor{progress bar in head/foot}{%
1232   use=normal text.fg,
1233   fg=nBlack,
1234   parent=progress bar
1235 }
1236 \setbeamercolor{progress bar in section page}{
1237   use=progress bar,
1238   parent=progress bar
1239 }

```

Block environments such as `theorem` and `example` have no background color by default. The option `block=fill` sets a background color based on the back-

ground 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.

```
1240 \newcommand{\neo@block@transparent}{%
1241   \setbeamercolor{block title}{%
1242     use=normal text,
1243     fg=nDarkBlue,
1244     bg=
1245   }
1246   \setbeamercolor{block title alerted}{%
1247     use={block title, alerted text},
1248     bg=block title.bg,
1249     fg=alerted text.fg
1250   }
1251   \setbeamercolor{block title example}{%
1252     use={block title, example text},
1253     bg=block title.bg,
1254     fg=example text.fg
1255   }
1256   \setbeamercolor{block body}{
1257     bg=
1258   }
1259   \setbeamercolor{block body alerted}{
1260     use=block body,
1261     parent=block body,
1262     bg=
1263   }
1264   \setbeamercolor{block body example}{
1265     use=block body,
1266     parent=block body,
1267     bg=
1268   }
1269 }
1270 \newcommand{\neo@block@fill}{%
1271   \setbeamercolor{block title}{%
1272     use=normal text,
1273     fg=nDarkBlue,
1274     bg=nGrey
1275   }
1276   \setbeamercolor{block title alerted}{%
1277     use={block title, alerted text},
1278     bg=alerted text.fg,
1279     fg=alerted text.bg
```

```

1280 }
1281 \setbeamercolor{block title example}{%
1282   use={block title, example text},
1283   bg=example text.fg,
1284   fg=example text.bg
1285 }
1286 \setbeamercolor{block body}{
1287   use={block title, normal text},
1288   bg=nLightGrey
1289 }
1290 \setbeamercolor{block body alerted}{
1291   use=block body,
1292   parent=block body,
1293   bg=nRed!50,
1294 }
1295 \setbeamercolor{block body example}{
1296   use=block body,
1297   parent=block body,
1298   bg=nYellow!50
1299 }
1300 }
1301

```

Footnotes

```

1302 \setbeamercolor{footnote}{fg=normal text.fg!90}
1303 \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.

```

1304 \setbeamercolor{bibliography entry author}{fg=, bg=}
1305 \setbeamercolor{bibliography entry title}{fg=, bg=}
1306 \setbeamercolor{bibliography entry location}{fg=, bg=}
1307 \setbeamercolor{bibliography entry note}{fg=, bg=}

```

8.5.7 Process package options

```

1308 \neo@color@setdefaults
1309 \ProcessPgfpPackageOptions{/neo/color}
1310 \mode<all>

```

8.6 Tol pgfplots theme

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

```
1311 \definecolor{TolDarkPurple}{HTML}{332288}  
1312 \definecolor{TolDarkBlue}{HTML}{6699CC}  
1313 \definecolor{TolLightBlue}{HTML}{88CCEE}  
1314 \definecolor{TolLightGreen}{HTML}{44AA99}  
1315 \definecolor{TolDarkGreen}{HTML}{117733}  
1316 \definecolor{TolDarkBrown}{HTML}{999933}  
1317 \definecolor{TolLightBrown}{HTML}{DDCC77}  
1318 \definecolor{TolDarkRed}{HTML}{661100}  
1319 \definecolor{TolLightRed}{HTML}{CC6677}  
1320 \definecolor{TolLightPink}{HTML}{AA4466}  
1321 \definecolor{TolDarkPink}{HTML}{882255}  
1322 \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.

```
1323 \pgfplotscreateplotcyclelist{mbarplot cycle}{%  
1324 {draw=TolDarkBlue, fill=TolDarkBlue!70},  
1325 {draw=TolLightBrown, fill=TolLightBrown!70},  
1326 {draw=TolLightGreen, fill=TolLightGreen!70},  
1327 {draw=TolDarkPink, fill=TolDarkPink!70},  
1328 {draw=TolDarkPurple, fill=TolDarkPurple!70},  
1329 {draw=TolDarkRed, fill=TolDarkRed!70},  
1330 {draw=TolDarkBrown, fill=TolDarkBrown!70},  
1331 {draw=TolLightRed, fill=TolLightRed!70},  
1332 {draw=TolLightPink, fill=TolLightPink!70},  
1333 {draw=TolLightPurple, fill=TolLightPurple!70},  
1334 {draw=TolLightBlue, fill=TolLightBlue!70},  
1335 {draw=TolDarkGreen, fill=TolDarkGreen!70},  
1336 }
```

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

```
1337 \pgfplotscreateplotcyclelist{mlineplot cycle}{%  
1338 {TolDarkBlue, mark=*, mark size=1.5pt},  
1339 {TolLightBrown, mark=square*, mark size=1.3pt},  
1340 {TolLightGreen, mark=triangle*, mark size=1.5pt},
```

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

```

1341 {TolDarkBrown, mark=diamond*, mark size=1.5pt},
1342 }

```

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.

```

1343 \pgfplotsset{
1344   compat=1.9,

```

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

```

1345   mlineplot/.style={
1346     mbaseplot,
1347     xmajorgrids=true,
1348     ymajorgrids=true,
1349     major grid style={dotted},
1350     axis x line=bottom,
1351     axis y line=left,
1352     legend style={
1353       cells={anchor=west},
1354       draw=none
1355     },
1356     cycle list name=mlineplot cycle,
1357   },

```

`mbarplot` A style to apply to the axis of a PGF bar chart. `mbarplot` uses vertical horizontal `mbarplot` 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`.

```

1358   mbarplot base/.style={
1359     mbaseplot,
1360     bar width=6pt,
1361     axis y line*=none,
1362   },
1363   mbarplot/.style={
1364     mbarplot base,
1365     ybar,
1366     xmajorgrids=false,
1367     ymajorgrids=true,
1368     area legend,
1369     legend image code/.code={%

```

```

1370     \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1371 },
1372 cycle list name=mbarplot cycle,
1373 },
1374 horizontal mbarplot/.style={
1375     mbarplot base,
1376     xmajorgrids=true,
1377     ymajorgrids=false,
1378     xbar stacked,
1379     area legend,
1380     legend image code/.code={%
1381         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1382     },
1383     cycle list name=mbarplot cycle,
1384 },

```

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

```

1385 mbaseplot/.style={
1386     legend style={
1387         draw=none,
1388         fill=none,
1389         cells={anchor=west},
1390     },
1391     x tick label style={
1392         font=\footnotesize
1393     },
1394     y tick label style={
1395         font=\footnotesize
1396     },
1397     legend style={
1398         font=\footnotesize
1399     },
1400     major grid style={
1401         dotted,
1402     },
1403     axis x line*=bottom,
1404 },
1405 disable thousands separator/.style={
1406     /pgf/number format/.cd,
1407     1000 sep={}
1408 },
1409 }

```