

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 | 39 |

| | | |
|-------|-------------------------------------|----|
| 8.4 | neo font theme | 39 |
| 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 | 43 |
| 8.4.6 | Process package options | 48 |
| 8.5 | neo color theme | 49 |
| 8.5.1 | Package dependencies | 49 |
| 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 X_YLaTeX 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 \LaTeX 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 \LaTeX 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**
- **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_YLaTeX, 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_YLaTeX 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 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   author section/.code=\setbeamertemplate{frame footer}[author section],
466   title/.code=\setbeamertemplate{frame footer}[title],
467   title section/.code=\setbeamertemplate{frame footer}[title section],
468 }

```

footer style Footer background color

```

469 \providebool{neo@standoutfooter}
470 \pgfkeys{
471   /neo/outer/footer style/.cd,
472   .is choice,
473   plain/.code={\boolfalse{neo@standoutfooter}\setbeamertemplate{footline}[plain]},
474   standout/.code={\booltrue{neo@standoutfooter}\setbeamertemplate{footline}[standout]},
475 }

```

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).

```

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

```

```

483     head/.code={\pgfkeys{/neo/outer/progressbar=none}
484       \addtobeamertemplate{headline}{}{%
485         \usebeamertemplate*{progress bar in head/foot}
486       }
487   },
488   frametitle/.code={\pgfkeys{/neo/outer/progressbar=none}
489     \addtobeamertemplate{frametitle}{}{%
490       \usebeamertemplate*{progress bar in head/foot}
491     }
492   },
493   foot/.code={\pgfkeys{/neo/outer/progressbar=none}
494     \ifbool{neo@standoutfooter}{}%
495       \addtobeamertemplate{footline}{\usebeamertemplate*{progress bar in head/foot}}{} %
496     }{%
497       \addtobeamertemplate{footline}{}{\usebeamertemplate*{progress bar in head/foot}} %
498     }
499   },
500 }

```

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

```

501 \newcommand{\neo@outer@setdefaults}{
502   \pgfkeys{/neo/outer/.cd,
503     frametitle icon=none,
504     footer=none,
505     footer style=plain,
506     numbering=counter,
507     progressbar=none,
508   }
509 }

```

8.3.3 Head and footline

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

```

510 \setbeamertemplate{navigation symbols}{}

```

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

```

511 \defbeamertemplate{frametitle icon}{none}{}

```



```

512 \defbeamertemplate{frametitle icon}{i4}{ \hfill\raisebox{-.25\height}{\includegraphics[height=1
513 \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.

```

514 \defbeamertemplate{frame footer}{none}{}
515 \defbeamertemplate{frame footer}{author}{
516 \node[item1] (footeritem1) at (current page.south west) {\insertshortauthor};%
517 }
518 \defbeamertemplate{frame footer}{author title}{
519 \node[item1] (footeritem1) at (current page.south west) {\insertshortauthor};%
520 \node[itemm] (footeritemm) at (current page.south) {\insertshorttitle};%
521 }
522 \defbeamertemplate{frame footer}{title}{
523 \node[item1] (footeritem1) at (current page.south west) {\insertshorttitle};%
524 }
525 \defbeamertemplate{frame footer}{title section}{
526 \node[item1] (footeritem1) at (current page.south west) {\insertshorttitle};%
527 \node[itemm] (footeritemm) at (current page.south) {\insertshortsection};%
528 }
529 \defbeamertemplate{frame footer}{author section}{
530 \node[item1] (footeritem1) at (current page.south west) {\insertshortauthor};%
531 \node[itemm] (footeritemm) at (current page.south) {\insertshortsection};%
532 }
533 \defbeamertemplate{frame footer}{custom}[1]{ #1 }

```

Add strut to ensure that frame numbers don't jump

```

534 \newcommand{\neo@framenumberingstrut}{\vphantom{0123456789}}
535 \defbeamertemplate{frame numbering}{none}{}
536 \defbeamertemplate{frame numbering}{counter}{\neo@framenumberingstrut\insertframenumber}
537 \defbeamertemplate{frame numbering}{fraction}{
538 \neo@framenumberingstrut\insertframenumber/\inserttotalframenumber
539 }

```

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

footline

```

540 \defbeamertemplate{headline}{plain}{}
541 \defbeamertemplate{footline}{plain}{%
542 \begin{tikzpicture}[overlay]

```

```

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

```

```

583 }{%
584 \node[fit=(footeriteml)(footeritemr)] (footer) {};
585 }
586 }%
587 \phantom{\drawfooter}%
588 \path[fill={block title.fg}] (current page.south west) rectangle (footer.north -| current page.
589 \drawfooter%
590 \end{tikzpicture}
591 }

```

8.3.4 Frametitle

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

```

592 \newlength{\neo@frametitle@padding}
593 \setlength{\neo@frametitle@padding}{2.2ex}
594 \newcommand{\neo@frametitlestrut@start}{
595   \rule{0pt}{\neo@frametitle@padding +%
596     \totalheightof{
597       \ifcsdef{neo@frametitleformat}{\neo@frametitleformat X}{X}%
598     }}%
599 }%
600 }
601 \newcommand{\neo@frametitlestrut@end}{
602   \rule[-\neo@frametitle@padding]{0pt}{\neo@frametitle@padding}
603 }
604 \defbeamertemplate{frametitle}{plain}{%
605   \nointerlineskip%
606   \begin{beamercolorbox}[%
607     wd=\paperwidth,%
608     sep=0pt,%
609     leftskip=\neo@frametitle@padding,%
610     rightskip=\neo@frametitle@padding,%
611   ]{frametitle}%
612   \neo@frametitlestrut@start%
613   \insertframetitle%
614   \usebeamertemplate*{frametitle icon}%
615   \nolinebreak%
616   \neo@frametitlestrut@end%
617   \end{beamercolorbox}%
618 }

```

```

619 \setbeamertemplate{frametitle continuation}{%
620   \usebeamerfont{frametitle}
621   {\normalfont (\insertcontinuationcount)}
622 }

```

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**.

```

623 \newlength{\neo@progressinheadfoot}
624 \newlength{\neo@progressinheadfoot@linewidth}
625 \setlength{\neo@progressinheadfoot@linewidth}{0.8pt}
626 \setbeamertemplate{progress bar in head/foot}{
627   \nointerlineskip
628   \setlength{\neo@progressinheadfoot}{%
629     \paperwidth * \ratio{\insertframenum pt}{\inserttotalframenum pt}}%
630   }%
631   \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
632     \tikzexternaldisable%
633     \begin{tikzpicture}
634       \fill[bg] (0,0) rectangle (\paperwidth, \neo@progressinheadfoot@linewidth);
635       \fill[fg] (0,0) rectangle (\neo@progressinheadfoot, \neo@progressinheadfoot@linewidth);
636     \end{tikzpicture}%
637     \tikzexternalenable%
638   \end{beamercolorbox}
639 }

```

custom notes Templates for note pages

```

640 \defbeamertemplate{note page}{preview-big}
641 {%
642   {%
643     \scriptsize
644     \usebeamerfont{note title}\usebeamercolor[fg]{note title}%
645     \ifbeamercoloreempty[bg]{note title}{}{%
646       \insertvrule{.45\paperheight}{note title.bg}%
647       \vskip-.45\paperheight%
648       \nointerlineskip%
649     }%
650     \vbox{
651       \hfill\insertslideintonotes{0.45}\hskip-\Gm@rmargin\hskip0pt%

```

```

652     \vskip-0.45\paperheight%
653     \nointerlineskip
654     \begin{pgfpicture}{0cm}{0cm}{0cm}{0cm}
655         \begin{pgflowlevelslope}{\pgftransformrotate{90}}
656             {\pgftransformshift{\pgfpoint{-2cm}{0.2cm}}}%
657             \pgftext[base,left]{\usebeamerfont{note date}\usebeamercolor[fg]{note date}\the\year-
658         \end{pgflowlevelslope}
659     \end{pgfpicture}}
660     \nointerlineskip
661     \vbox to .45\paperheight{\vskip0.5em
662         \hbox{\insertshorttitle[width=8cm]}%
663         \setbox\beamer@tempbox=\hbox{\insertsection}%
664         \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip4pt\raise3pt\hbox{\vrule
665             width0.4pt height7pt\vrule width 9pt
666             height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere{}\insertsection
667         }}%
668         \setbox\beamer@tempbox=\hbox{\insertsubsection}%
669         \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip17.4pt\raise3pt\hbox{\vrule
670             width0.4pt height7pt\vrule width 9pt
671             height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere{}\insertsubsec
672         }}%
673         \setbox\beamer@tempbox=\hbox{\insertshortframetitle}%
674         \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip30.8pt\raise3pt\hbox{\vrule
675             width0.4pt height7pt\vrule width 9pt
676             height0.4pt}}\hskip1pt\hbox{\insertshortframetitle[width=7cm]}\fi%
677         }}%
678     \vfil}%
679 }%
680 \ifbeamercolorempy[bg]{note page}{-}{%
681     \nointerlineskip%
682     \insertvrule{.55\paperheight}{note page.bg}%
683     \vskip-.55\paperheight%
684 }%
685 \vskip.25em
686 \nointerlineskip
687 \insertnote
688 }
689 \defbeamer-template{note page}{print}
690 {%
691     {%

```

```

692 \nointerlineskip%
693 \begin{beamercolorbox}[%
694     wd=\paperwidth,%
695     sep=0pt,%
696     leftskip=\neo@frametitle@padding,%
697     rightskip=\neo@frametitle@padding,%
698 ]{note title}%
699     \usebeamerfont{frametitle}%
700     \neo@frametitlestrut@start%
701     \insertframetitle%
702     \usebeamertemplate*{frametitle icon}%
703     \nolinebreak%
704     \neo@frametitlestrut@end%
705 \end{beamercolorbox}%
706 }%
707 \insertnote%
708 \vfill%
709 \begin{beamercolorbox}[wd=\paperwidth, sep=3ex]{footline}%
710     \usebeamerfont{page number in head/foot}%
711     \usebeamertemplate*{frame footer}
712     \hfill%
713     \usebeamertemplate*{frame numbering}
714 \end{beamercolorbox}%
715 \vskip4pt%
716 }

```

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`.

```

717 \AtBeginDocument{%
718     \apptocmd{\appendix}{%
719         \pgfkeys{%
720             /neo/outer/.cd,
721             numbering=none,
722             progressbar=none}
723     }{}{}
724 }

```

8.3.5 Process package options

```
725 \neo@outer@setdefaults
726 \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

```
727 \RequirePackage{etoolbox}
728 \RequirePackage{ifxetex}
729 \RequirePackage{ifluatex}
730 \RequirePackage{pgfopts}
```

8.4.2 Load Fira fonts

If the presentation is compiled with Xe^LA_TE_X or Lua^LA_TE_X, the fontspec package is loaded and we search for the Fira fonts.

```
731 \ifboolexpr{bool {xetex} or bool {luatex}}{
732   \@ifpackageloaded{fontspec}{
733     \PassOptionsToPackage{no-math}{fontspec}
734   }{
735     \RequirePackage[no-math]{fontspec}
736   }

737   \IfFileExists{FiraSans-Regular.otf}{
738     \defaultfontfeatures{
739       Scale      = 1.0,
740       Extension = .otf
741     }
742   }{
743     \PackageWarning{beamerthemeneo}{%
744       FiraSans font not found in path, therefore using system fonts. %
745       Make sure you have the fonts installed.%
746     }
747   }

748   \setmonofont
749     [ Numbers = {Monospaced,Lining},
750     UprightFont = *-Regular ,
751     ItalicFont  = *-Regular ,
```

```

752     BoldFont      = *-Medium ,
753     BoldItalicFont = *-Medium ,
754 ]
755 {FiraMono}
756 \newcommand{\neo@fontsave}{
757     \let\firaneofamily\sfddefault
758     \renewcommand*\familydefault{\firaneofamily}
759 }
760 \newcommand{\neo@fontlight}{
761     \setsansfont[
762         Numbers = {OldStyle, Monospaced},
763         UprightFont = *-Light ,
764         ItalicFont = *-LightItalic ,
765         BoldFont = *-Regular ,
766         BoldItalicFont = *-RegularItalic ,
767     ]{FiraSans}
768     \neo@fontsave
769 }
770 \newcommand{\neo@fontbook}{
771     \setsansfont[
772         Numbers = {OldStyle, Monospaced},
773         UprightFont = *-Book ,
774         ItalicFont = *-BookItalic ,
775         BoldFont = *-Medium ,
776         BoldItalicFont = *-MediumItalic ,
777     ]{FiraSans}
778     \neo@fontsave
779 }
780 \newcommand{\neo@fontregular}{
781     \setsansfont[
782         Numbers = {OldStyle, Monospaced},
783         UprightFont = *-Regular ,
784         ItalicFont = *-RegularItalic ,
785         BoldFont = *-SemiBold ,
786         BoldItalicFont = *-SemiBoldItalic ,
787     ]{FiraSans}
788     \neo@fontsave
789 }
790 \AtBeginEnvironment{tabular}{%
791     \addfontfeature{Numbers={Monospaced}}%

```



```

792 }
793 }{%
794 \RequirePackage[utf8]{inputenc}
795 \IfFileExists{FiraSans.sty}{
796 \RequirePackage[T1]{fontenc}
797 \RequirePackage[sfdefault]{FiraSans}
798 \RequirePackage[nomap,lining]{FiraMono}
799 \def\bfseries@tt{mb}
800 \newcommand{\neo@fontsave}{
801 \edef\familydefault{\sfdefault}
802 \edef\seriesdefault{\mdseries@sf}
803 }
804 \newcommand{\neo@fontlight}{
805 \def\mdseries@sf{l}
806 \def\bfseries@sf{m}
807 \neo@fontsave
808 }
809 \newcommand{\neo@fontbook}{
810 \def\bfseries@sf{mb}
811 \neo@fontsave
812 }
813 \newcommand{\neo@fontregular}{
814 \def\mdseries@sf{m}
815 \def\bfseries@sf{sb}
816 \neo@fontsave
817 }
818 }{
819 \PackageWarning{beamerthemeneo}{%
820 You need to install the Fira Fonts package or compile with XeLaTeX or %
821 LuaLaTeX to use the included Fira fonts%
822 }
823 }
824 }

```

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

8.4.3 General font definitions

```

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

```

8.4.4 Font style options

`titleformat title` Controls the overall font style.

```

852 \pgfkeys{
853   /neo/font/style/.cd,
854   .is choice,
855   light/.code={\neo@fontlight},
856   book/.code={\neo@fontbook},
857   regular/.code={\neo@fontregular},
858 }

```

8.4.5 Title format options

`titleformat title` Controls the format of the title.

```
859 \pgfkeys{
860   /neo/font/titleformat title/.cd,
861   .is choice,
862   regular/.code={%
863     \let\neo@titleformat\@empty%
864     \setbeamerfont{title}{shape=\normalfont}%
865   },
866   smallcaps/.code={%
867     \let\neo@titleformat\@empty%
868     \setbeamerfont{title}{shape=\scshape}%
869   },
870   allsmallcaps/.code={%
871     \let\neo@titleformat\lowercase%
872     \setbeamerfont{title}{shape=\scshape}%
873     \PackageWarning{beamerthemeneo}{%
874       Be aware that titleformat title=allsmallcaps can lead to problems%
875     }
876   },
877   allcaps/.code={%
878     \let\neo@titleformat\uppercase%
879     \setbeamerfont{title}{shape=\normalfont}%
880     \PackageWarning{beamerthemeneo}{%
881       Be aware that titleformat title=allcaps can lead to problems%
882     }
883   },
884 }
```

`titleformat subtitle` Control the format of the subtitle.

```
885 \pgfkeys{
886   /neo/font/titleformat subtitle/.cd,
887   .is choice,
888   regular/.code={%
889     \let\neo@subtitleformat\@empty%
890     \setbeamerfont{subtitle}{shape=\normalfont}%
891   },
892   smallcaps/.code={%
```

```

893     \let\neo@subtitleformat\@empty%
894     \setbeamerfont{subtitle}{shape=\scshape}%
895 },
896 allsmallcaps/.code={%
897     \let\neo@subtitleformat\lowercase%
898     \setbeamerfont{subtitle}{shape=\scshape}%
899     \PackageWarning{beamerthemeneo}{%
900         Be aware that titleformat subtitle=allsmallcaps can lead to problems%
901     }
902 },
903 allcaps/.code={%
904     \let\neo@subtitleformat\uppercase%
905     \setbeamerfont{subtitle}{shape=\normalfont}%
906     \PackageWarning{beamerthemeneo}{%
907         Be aware that titleformat subtitle=allcaps can lead to problems%
908     }
909 },
910 }

```

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

```

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

```

```

930     \let\neo@sectiontitleformat\MakeUppercase%
931     \setbeamerfont{section title}{shape=\normalfont}%
932     \PackageWarning{beamerthemeneo}{%
933       Be aware that titleformat section=allcaps can lead to problems%
934     }
935   },
936 }

```

`frametitleformat` Control the format of the frame title.

```

937 \pgfkeys{
938   /neo/font/titleformat frame/.cd,
939   .is choice,
940   regular/.code={%
941     \let\neo@frametitleformat\@empty%
942     \setbeamerfont{frametitle}{shape=\normalfont}%
943   },
944   smallcaps/.code={%
945     \let\neo@frametitleformat\@empty%
946     \setbeamerfont{frametitle}{shape=\scshape}%
947   },
948   allsmallcaps/.code={%
949     \let\neo@frametitleformat\MakeLowercase%
950     \setbeamerfont{frametitle}{shape=\scshape}%
951     \PackageWarning{beamerthemeneo}{%
952       Be aware that titleformat frame=allsmallcaps can lead to problems%
953     }
954   },
955   allcaps/.code={%
956     \let\neo@frametitleformat\MakeUppercase%
957     \setbeamerfont{frametitle}{shape=\normalfont}
958     \PackageWarning{beamerthemeneo}{%
959       Be aware that titleformat frame=allcaps can lead to problems%
960     }
961   },
962 }

```

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

```

963 \pgfkeys{

```

```

964 /neo/font/.cd,
965 titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
966 titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
967 titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
968 titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
969 }

```

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

```

970 \newcommand{\neo@font@setdefaults}{
971   \pgfkeys{/neo/font/.cd,
972     style=book,
973     titleformat title=regular,
974     titleformat subtitle=regular,
975     titleformat section=regular,
976     titleformat frame=regular,
977   }
978 }

```

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

```

979 \def\neo@titleformat#1{#1}
980 \def\neo@subtitleformat#1{#1}
981 \def\neo@sectiontitleformat#1{#1}
982 \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](#).

```

983 \patchcmd{\beamer@title}%
984   {\def\inserttitle{#2}}%
985   {\def\inserttitle{\neo@titleformat{#2}}}%
986   {}%
987   {\PackageError{beamerfontthemeneo}{Patching title failed}\@ehc}
988 \patchcmd{\beamer@subtitle}%
989   {\def\insertsubtitle{#2}}%
990   {\def\insertsubtitle{\neo@subtitleformat{#2}}}%
991   {}%
992   {\PackageError{beamerfontthemeneo}{Patching subtitle failed}\@ehc}

```

```

993 \patchcmd{\sectionentry}
994   {\def\insertsectionhead{#2}}
995   {\def\insertsectionhead{\neo@sectiontitleformat{#2}}}
996   {}
997   {\PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc}
998 \@tempswafalse
999 \patchcmd{\beamer@section}
1000   {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#1}}}}
1001   {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1002     \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}
1003   {\@tempswatrue}
1004   {}
1005 \patchcmd{\beamer@section}
1006   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
1007   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
1008     \neo@sectiontitleformat{#1}}}
1009   {\@tempswatrue}
1010   {}
1011 \patchcmd{\beamer@section}
1012   {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#1}}}}
1013   {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1014     \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}
1015   {\@tempswatrue}
1016   {}
1017 \patchcmd{\beamer@section}
1018   {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#1}}}
1019   {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1020     \noexpand\neo@sectiontitleformat{#1}}}
1021   {\@tempswatrue}
1022   {}
1023 \if@tempswa\else
1024   \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
1025 \fi
1026 \@tempswafalse
1027 \patchcmd{\beamer@subsection}
1028   {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#1}}}}
1029   {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1030     \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}
1031   {\@tempswatrue}
1032   {}

```

```

1033 \patchcmd{\beamer@subsection}
1034   {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
1035   {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{%
1036     \neo@sectiontitleformat{#1}}}}
1037   {\@tempswatrue}
1038   {}
1039 \patchcmd{\beamer@subsection}
1040   {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#1}}}
1041   {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
1042     \noexpand\neo@sectiontitleformat{#1}}}}
1043   {\@tempswatrue}
1044   {}
1045 \if@tempswa\else
1046   \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
1047 \fi

```

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

```

1048 \patchcmd{\beamer@@frametitle}
1049   {{%
1050     \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
1051       \usebeamertemplate*{frametitle continuation}\fi}}%
1052     \gdef\beamer@frametitle{#2}%
1053     \gdef\beamer@shortframetitle{#1}%
1054   }}
1055   {{%
1056     \gdef\insertframetitle{{\neo@frametitleformat{#2}\ifnum%
1057       \beamer@autobreakcount>0\relax{}\space%
1058       \usebeamertemplate*{frametitle continuation}\fi}}%
1059     \gdef\beamer@frametitle{#2}%
1060     \gdef\beamer@shortframetitle{#1}%
1061   }}
1062   {}
1063   {\PackageError{beamerfontthemeneo}{Patching frame title failed}\@ehc}

```

8.4.6 Process package options

```

1064 \neo@font@setdefaults
1065 \ProcessPgfpPackageOptions{/neo/font}

```


8.5 neo color theme

8.5.1 Package dependencies

```
1066 \RequirePackage{pgfopts}
```

8.5.2 Options

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

```
1067 \pgfkeys{
1068   /neo/color/block/.cd,
1069   .is choice,
1070   transparent/.code=\neo@block@transparent,
1071   fill/.code=\neo@block@fill,
1072 }
```

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

```
1073 \pgfkeys{
1074   /neo/color/background/.cd,
1075   .is choice,
1076   dark/.code=\neo@colors@dark,
1077   light/.code=\neo@colors@light,
1078 }
```

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

```
1079 \newcommand{\neo@color@setdefaults}{
1080   \pgfkeys{/neo/color/.cd,
1081     background=light,
1082     block=transparent,
1083   }
1084 }
```

8.5.3 Base colors

```
1085
1086 \definecolor{nDarkGrey}{RGB}{152,164,174}
1087 \definecolor{nGrey}{RGB}{210,213,215}
1088 \definecolor{nLightGrey}{RGB}{235,236,238}
```

```

1089
1090 \definecolor{nDarkRed}{RGB}{141,20,41}
1091 \definecolor{nRed}{RGB}{201,169,147}
1092 \definecolor{nLightRed}{RGB}{237,231,222}
1093
1094 \definecolor{nDarkGreen}{RGB}{0,155,119}
1095 \definecolor{nGreen}{RGB}{170,207,189}
1096 \definecolor{nLightGreen}{RGB}{229,239,234}
1097
1098 \definecolor{nDarkBlue}{RGB}{0,56,101}
1099 \definecolor{nBlue}{RGB}{144,167,198}
1100 \definecolor{nLightBlue}{RGB}{221,229,240}
1101
1102 \definecolor{nDarkCyan}{RGB}{0,177,235}
1103 \definecolor{nCyan}{RGB}{180,214,245}
1104 \definecolor{nLightCyan}{RGB}{234,243,252}
1105
1106 \definecolor{nDarkYellow}{RGB}{201,147,19}
1107 \definecolor{nYellow}{RGB}{217,198,137}
1108 \definecolor{nLightYellow}{RGB}{243,238,223}
1109
1110 \definecolor{nBlack}{HTML}{011F32}
1111 \definecolor{nWhite}{RGB}{250,250,250}

```

8.5.4 Alias colors

Support the colors provided by the old i4 beamer theme.

```

1112 \colorlet{i4red}{nDarkRed}
1113 \colorlet{i4green}{nDarkGreen}
1114 \colorlet{i4blue}{nDarkBlue}
1115 \colorlet{i4cyan}{nDarkCyan}
1116 \colorlet{i4yellow}{nDarkYellow}
1117 \colorlet{i4grey}{nDarkGrey}
1118 \definecolor{darkred}{rgb}{0.8,0,0}
1119 \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`.

```

1120 \newcommand{\neo@colors@dark}{
1121   \setbeamercolor{normal text}{%
1122     fg=nWhite,
1123     bg=nBlack
1124   }
1125   \setbeamercolor{normal item}{%
1126     fg=nWhite,
1127     bg=nDarkBlue
1128   }
1129   \usebeamercolor[fg]{normal text}
1130 }
1131 \newcommand{\neo@colors@light}{
1132   \setbeamercolor{normal text}{%
1133     fg=nBlack,
1134     bg=nWhite
1135   }
1136   \setbeamercolor{normal item}{%
1137     fg=nDarkBlue,
1138     bg=nWhite
1139   }
1140 }
1141 \setbeamercolor{alerted text}{%
1142   fg=nDarkRed
1143 }
1144 \setbeamercolor{example text}{%
1145   fg=nDarkYellow
1146 }
1147 \setbeamercolor{note title}{%
1148   fg=nDarkBlue,
1149   bg=nGrey
1150 }
1151 \setbeamercolor{note page}{%
1152   fg=nBlack,
1153   bg=nLightGrey
1154 }

```

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`.

```
1155 \setbeamercolor{titlelike}{use=normal text, parent=normal text}
1156 \setbeamercolor{author}{use=normal text, parent=normal text}
1157 \setbeamercolor{date}{use=normal text, parent=normal text}
1158 \setbeamercolor{institute}{use=normal text, parent=normal text}
1159 \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.

```
1160 \setbeamercolor{palette primary}{%
1161   use=normal text,
1162   fg=normal text.bg,
1163   bg=nDarkBlue
1164 }
1165 \setbeamercolor{frametitle}{%
1166   use=palette primary,
1167   parent=palette primary
1168 }
```

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`.

```
1169 \setbeamercolor{progress bar}{%
1170   use=normal text,
1171   fg=nDarkBlue,
1172   bg=nLightBlue
1173 }
1174 \setbeamercolor{title separator}{
1175   use=progress bar,
1176   parent=progress bar
1177 }
1178 \setbeamercolor{progress bar in head/foot}{%
1179   use=normal text.fg,
1180   fg=nBlack,
1181   parent=progress bar
```

```

1182 }
1183 \setbeamercolor{progress bar in section page}{
1184   use=progress bar,
1185   parent=progress bar
1186 }

```

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.

```

1187 \newcommand{\neo@block@transparent}{
1188   \setbeamercolor{block title}{%
1189     use=normal text,
1190     fg=nDarkBlue,
1191     bg=
1192   }
1193   \setbeamercolor{block title alerted}{%
1194     use={block title, alerted text},
1195     bg=block title.bg,
1196     fg=alerted text.fg
1197   }
1198   \setbeamercolor{block title example}{%
1199     use={block title, example text},
1200     bg=block title.bg,
1201     fg=example text.fg
1202   }
1203   \setbeamercolor{block body}{
1204     bg=
1205   }
1206   \setbeamercolor{block body alerted}{
1207     use=block body,
1208     parent=block body
1209   }
1210   \setbeamercolor{block body example}{
1211     use=block body,
1212     parent=block body
1213   }
1214 }

```

```

1215 \newcommand{\neo@block@fill}{
1216   \setbeamercolor{block title}{%
1217     use=normal text,
1218     fg=nDarkBlue,
1219     bg=nGrey
1220   }
1221   \setbeamercolor{block title alerted}{%
1222     use={block title, alerted text},
1223     bg=alerted text.fg,
1224     fg=alerted text.bg
1225   }
1226   \setbeamercolor{block title example}{%
1227     use={block title, example text},
1228     bg=example text.fg,
1229     fg=example text.bg
1230   }
1231   \setbeamercolor{block body}{
1232     use={block title, normal text},
1233     bg=nLightGrey
1234   }
1235   \setbeamercolor{block body alerted}{
1236     use=block body,
1237     parent=block body,
1238     bg=nRed!50,
1239   }
1240   \setbeamercolor{block body example}{
1241     use=block body,
1242     parent=block body,
1243     bg=nYellow!50
1244   }
1245 }
1246

```

Footnotes

```

1247 \setbeamercolor{footnote}{fg=normal text.fg!90}
1248 \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.

```
1249 \setbeamercolor{bibliography entry author}{fg=, bg=}
1250 \setbeamercolor{bibliography entry title}{fg=, bg=}
1251 \setbeamercolor{bibliography entry location}{fg=, bg=}
1252 \setbeamercolor{bibliography entry note}{fg=, bg=}
```

8.5.7 Process package options

```
1253 \neo@color@setdefaults
1254 \ProcessPgfPackageOptions{/neo/color}

1255 \mode<all>
```

8.6 Tol pgfplots theme

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

```
1256 \definecolor{TolDarkPurple}{HTML}{332288}
1257 \definecolor{TolDarkBlue}{HTML}{6699CC}
1258 \definecolor{TolLightBlue}{HTML}{88CCFF}
1259 \definecolor{TolLightGreen}{HTML}{44AA99}
1260 \definecolor{TolDarkGreen}{HTML}{117733}
1261 \definecolor{TolDarkBrown}{HTML}{999933}
1262 \definecolor{TolLightBrown}{HTML}{DDCC77}
1263 \definecolor{TolDarkRed}{HTML}{661100}
1264 \definecolor{TolLightRed}{HTML}{CC6677}
1265 \definecolor{TolLightPink}{HTML}{AA4466}
1266 \definecolor{TolDarkPink}{HTML}{882255}
1267 \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.

```
1268 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
1269 {draw=TolDarkBlue, fill=TolDarkBlue!70},
1270 {draw=TolLightBrown, fill=TolLightBrown!70},
1271 {draw=TolLightGreen, fill=TolLightGreen!70},
1272 {draw=TolDarkPink, fill=TolDarkPink!70},
```

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

```

1273 {draw=TolDarkPurple, fill=TolDarkPurple!70},
1274 {draw=TolDarkRed, fill=TolDarkRed!70},
1275 {draw=TolDarkBrown, fill=TolDarkBrown!70},
1276 {draw=TolLightRed, fill=TolLightRed!70},
1277 {draw=TolLightPink, fill=TolLightPink!70},
1278 {draw=TolLightPurple, fill=TolLightPurple!70},
1279 {draw=TolLightBlue, fill=TolLightBlue!70},
1280 {draw=TolDarkGreen, fill=TolDarkGreen!70},
1281 }

```

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

```

1282 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
1283 {TolDarkBlue, mark=*, mark size=1.5pt},
1284 {TolLightBrown, mark=square*, mark size=1.3pt},
1285 {TolLightGreen, mark=triangle*, mark size=1.5pt},
1286 {TolDarkBrown, mark=diamond*, mark size=1.5pt},
1287 }

```

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.

```

1288 \pgfplotsset{
1289 compat=1.9,

```

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

```

1290 mlineplot/.style={
1291 mbaseplot,
1292 xmajorgrids=true,
1293 ymajorgrids=true,
1294 major grid style={dotted},
1295 axis x line=bottom,
1296 axis y line=left,
1297 legend style={
1298 cells={anchor=west},
1299 draw=none
1300 },
1301 cycle list name=mlineplot cycle,

```



```
1302  },
```

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**.

```
1303  mbarplot base/.style={
1304      mbaseplot,
1305      bar width=6pt,
1306      axis y line*=none,
1307  },
1308  mbarplot/.style={
1309      mbarplot base,
1310      ybar,
1311      xmajorgrids=false,
1312      ymajorgrids=true,
1313      area legend,
1314      legend image code/.code={%
1315          \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1316      },
1317      cycle list name=mbarplot cycle,
1318  },
1319  horizontal mbarplot/.style={
1320      mbarplot base,
1321      xmajorgrids=true,
1322      ymajorgrids=false,
1323      xbar stacked,
1324      area legend,
1325      legend image code/.code={%
1326          \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1327      },
1328      cycle list name=mbarplot cycle,
1329  },
```

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

```
1330  mbaseplot/.style={
1331      legend style={
1332          draw=none,
1333          fill=none,
1334          cells={anchor=west},
```

```

1335     },
1336     x tick label style={
1337         font=\footnotesize
1338     },
1339     y tick label style={
1340         font=\footnotesize
1341     },
1342     legend style={
1343         font=\footnotesize
1344     },
1345     major grid style={
1346         dotted,
1347     },
1348     axis x line*=bottom,
1349 },
1350 disable thousands separator/.style={
1351     /pgf/number format/.cd,
1352     1000 sep={}
1353 },
1354 }

```