

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	7
3.1.3	Outer theme	7
3.1.4	Color theme	7
3.1.5	Font theme	8
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	11
5.1	Backup Slides	11
6	Known Issues	11
6.1	Title formats	11
6.2	Interactions with other color themes	12
6.3	Notes on second screen	12
6.4	Standout frames with labels	13
6.5	Standout frames with Pandoc	14
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	18
8.2	NEO inner theme	18
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	25
8.2.6	Lists and floats	27
8.2.7	Footnotes	27
8.2.8	Text and spacing settings	28
8.2.9	Standout frames	28
8.2.10	Process package options	29
8.3	NEO outer theme	30
8.3.1	Package dependencies	30
8.3.2	Options	30
8.3.3	Head and footline	31
8.3.4	Frametitle	32
8.3.5	Process package options	34

8.4	NEO font theme	34
8.4.1	Package dependencies	34
8.4.2	Load Fira fonts	35
8.4.3	General font definitions	37
8.4.4	Font style options	38
8.4.5	Title format options	38
8.4.6	Process package options	44
8.5	NEO color theme	44
8.5.1	Package dependencies	44
8.5.2	Options	44
8.5.3	Base colors	45
8.5.4	Alias colors	46
8.5.5	Base styles	46
8.5.6	Derived colors	47
8.5.7	Process package options	50
8.6	Tol pgfplots theme	51

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_YTeX to typeset your slides. However, **NEO** can also be used with other typefaces and T_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 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:

```
\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 titleformat, 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	<i>none, simple, progressbar</i>	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	<i>none, counter, fraction</i>	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	<i>none, head, frametitle, foot</i>	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	<i>transparent, fill</i>	transparent
	Optionally adds a light grey background to block environments like theorem and example.	
background	<i>dark, light</i>	light
	Provides the option to have a dark background and light foreground instead of the reverse.	

3.1.5 Font theme

<code>titleformat title</code>	<i>regular, smallcaps, allsmallcaps, allcaps</i> regular
<code>titleformat subtitle</code>	Individually controls the format of titles, subtitles, section titles, and frame titles (see <code>titleformat</code> , above).
<code>titleformat section</code>	
<code>titleformat frame</code>	

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 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 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 \LaTeX , 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 Xe_{La}TeX itself. You can work around it either by compiling with Lua_{TeX} 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{pgfopts}
```

8.1.2 Options

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

```
3 \pgfkeys{/neo/.cd,
4   .search also={
5     /neo/inner,
6     /neo/outer,
```

```

7    /neo/color,
8    /neo/font,
9  }
10 }

```

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

```

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

```

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

```

37 \pgfkeys{
38   /neo/titleformat/.code=\pgfkeysalso{
39     font/titleformat title=#1,
40     font/titleformat subtitle=#1,

```

```

41     font/titleformat section=#1,
42     font/titleformat frame=#1,
43     titleformat plain=#1,
44   }
45 }

```

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

```

46 \pgfkeys{/neo/.cd,
47   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
48   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
49   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
50   darkcolors/.code=\pgfkeysalso{color/background=dark},
51   blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
52   light/.code=\pgfkeysalso{font/style=light},
53   book/.code=\pgfkeysalso{font/style=book},
54   regular/.code=\pgfkeysalso{font/style=regular},
55 }

```

Set default values for options.

```

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

```

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

```

61 \providecommand{\tikzexternalenable}{}
62 \providecommand{\tikzexternaldisable}{}

```

8.1.3 Component sub-packages

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

```

63 \useinnertheme{neo}

```



```

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

```

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

```

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

```

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.

```

73 \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.

```

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

```

`\mreducelistspacing`

```

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

```

8.1.5 Process package options

```
85 \neo@setdefaults
86 \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

```
87 \RequirePackage{etoolbox}
88 \RequirePackage{keyval}
89 \RequirePackage{calc}
90 \RequirePackage{pgfopts}
91 \RequirePackage{tikz}
```

8.2.2 Options

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

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

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

```
101 \pgfkeys{
```

```

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

```

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

```

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

```

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.

```

116 \setbeamertemplate{title page}{
117   \begin{minipage}[b][0.95\paperheight]{\textwidth}
118     \vfill%
119     \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
120     \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
121     \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](#).

```

122   \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
123   \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
124   \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
125   \vfill

```

```

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

```

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` template.

```

135 \renewcommand{\maketitle}[1][extern]{%
136   \ifbeamer@inframe
137     \titlepage
138   \else
139     {
140       \usebackgroundtemplate{\includegraphics[width=\paperwidth]{images/titlepage-#
141       \frame[plain,noframenumbering]{
142         \neo@colors@dark
143         \setbeamercolor{title separator}{
144         fg=black!20,
145         bg=normal text.fg
146     }
147       \titlepage
148     }
149   }
150 \fi
151 }
152 \def\titlepage{%
153   \usebeamertemplate{title page}

```

154 }

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

```
155 \setbeamertemplate{title graphic}{
156   \vbox to 0pt {
157     \vspace*{2em}
158     \inserttitlegraphic%
159   }%
160   \nointerlineskip%
161 }
```

title Set the title on the title page.

```
162 \setbeamertemplate{title}{
163   \raggedright%
164   \linespread{1.0}%
165   \inserttitle%
166   \par%
167   \vspace*{0.5em}
168 }
```

subtitle Set the subtitle on the title page.

```
169 \setbeamertemplate{subtitle}{
170   \raggedright%
171   \insertsubtitle%
172   \par%
173   \vspace*{0.5em}
174 }
```

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

```
175 \newlength{\neo@titleseparator@linewidth}
176 \setlength{\neo@titleseparator@linewidth}{0.4pt}
177 \setbeamertemplate{title separator}{
178   \tikzexternaldisable%
179   \begin{tikzpicture}
180     \fill[fg] (0,0) rectangle (\textwidth, \neo@titleseparator@linewidth);
```

```

181 \end{tikzpicture}%
182 \tikzexternalenable%
183 \par%
184 }

```

author Set the author on the title page.

```

185 \setbeamertemplate{author}{
186   \vspace*{2em}
187   \insertauthor%
188   \par%
189   \vspace*{0.25em}
190 }

```

date Set the date on the title page.

```

191 \setbeamertemplate{date}{
192   \insertdate%
193   \par%
194 }

```

institute Set the institute on the title page.

```

195 \setbeamertemplate{institute}{
196   \vspace*{3mm}
197   \insertinstitute%
198   \par%
199 }

```

8.2.4 Section page

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

```

200 \defbeamertemplate{section page}{simple}{
201   \begin{center}
202     \usebeamercolor[fg]{section title}
203     \usebeamerfont{section title}
204     \insertsectionhead\par
205     \ifx\insertsubsectionhead\@empty\else
206       \usebeamercolor[fg]{subsection title}

```

```

207     \usebeamerfont{subsection title}
208     \insertsubsectionhead
209     \fi
210 \end{center}
211 }
212 \defbeamertemplate{section page}{progressbar}{
213   \centering
214   \begin{minipage}{22em}
215     \raggedright
216     \usebeamercolor[fg]{section title}
217     \usebeamerfont{section title}
218     \insertsectionhead\[-1ex]
219     \usebeamertemplate*{progress bar in section page}
220     \par
221     \ifx\insertsubsectionhead\@empty\else%
222       \usebeamercolor[fg]{subsection title}%
223       \usebeamerfont{subsection title}%
224       \insertsubsectionhead
225     \fi
226   \end{minipage}
227   \par
228   \vspace{\baselineskip}
229 }
230 \newcommand{\neo@disablesectionpage}{
231   \AtBeginSection{
232     % intentionally empty
233   }
234 }
235 \newcommand{\neo@enablesectionpage}{
236   \AtBeginSection{
237     \ifbeamer@inframe
238       \sectionpage
239     \else
240       \frame[plain,c,noframenumbering]{\sectionpage}
241     \fi
242   }
243 }

```

subsection page Template for the subsection title slide that can optionally be added to at the

beginning of each subsection.

```

244 \setbeamertemplate{subsection page}{%
245   \usebeamertemplate*{section page}
246 }
247 \newcommand{\neo@disablesubsectionpage}{
248   \AtBeginSubsection{
249     % intentionally empty
250   }
251 }
252 \newcommand{\neo@enablesubsectionpage}{
253   \AtBeginSubsection{
254     \ifbeamer@inframe
255       \subsectionpage
256     \else
257       \frame[plain,c,noframenumbering]{\subsectionpage}
258     \fi
259   }
260 }

```

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.

```

261 \newlength{\neo@progressonsectionpage}
262 \newlength{\neo@progressonsectionpage@linewidth}
263 \setlength{\neo@progressonsectionpage@linewidth}{0.4pt}
264 \setbeamertemplate{progress bar in section page}{
265   \setlength{\neo@progressonsectionpage}{%
266     \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}}%
267   }%
268   \tikzexternaldisable%
269   \begin{tikzpicture}
270     \fill[bg] (0,0) rectangle (\textwidth, \neo@progressonsectionpage@linewidth);
271     \fill[fg] (0,0) rectangle (\neo@progressonsectionpage, \neo@progressonsectionpage@linewidth);
272   \end{tikzpicture}%
273   \tikzexternalenable%
274 }

```

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 `\inserttotalframenumbers`; presentations with over 4000 slides will still break on first compile, but users in that situation likely have deeper problems to solve.

```
275 \def\inserttotalframenumbers{100}
```

8.2.5 Block environments

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

```
276 \newlength{\neo@blocksep}
277 \newlength{\neo@blockadjust}
278 \setlength{\neo@blocksep}{0.75ex}
279 \setlength{\neo@blockadjust}{0.25ex}
280 \providecommand{\neo@strut}{%
281   \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}}%
282 }
283 \newcommand{\neo@block}[1]{
284   \par\vskip\medskipamount%
285   \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.

```

286 \ifbeamercoloreempty[bg]{block title#1}{%
287   \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}}{%
288   \ifbeamercoloreempty[bg]{block title}{%
289     \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}%
290   }%
291 %   \end{macrocode}
292 %
293 %   Otherwise, if the |block title| has a background, we set the padding based
294 %   on |\neo@blockskip|. However, we have to visually compensate for
295 %   the |\neo@strut| added to the block title (see below) by
296 %   subtracting |\neo@blockadjust| from the top and bottom padding.
297 %
298 %   \begin{macrocode}
299   {%
300     \begin{beamercolorbox}[
301       sep=\dimexpr\neo@blocksep-\neo@blockadjust\relax,
302       leftskip=\neo@blockadjust,
303       rightskip=\dimexpr\neo@blockadjust plus 4em\relax
304     ]{block title#1}%
305   }%
306 %   \end{macrocode}
307 %
308 %   We can now set the contents of the |block title|. The zero-width but
309 %   positive-height box |\neo@strut| ensures that the block title box
310 %   has a consistent height, even if it lacks punctuation, ascenders, or
311 %   descenders.
312 %
313 %   \begin{macrocode}
314     \usebeamerfont*{block title#1}%
315     \neo@strut%
316     \insertblocktitle%
317     \neo@strut%
318 \end{beamercolorbox}%
319 %   \end{macrocode}
320 %
321 %   Next, we typeset the |block body|. This the code is similar to, but simpler
322 %   than, the |block title| code since we don't need to adjust for any struts.

```

```

323 %
324 %   \begin{macrocode}
325 \nointerlineskip%
326 \ifbeamercoloreempty[bg]{block body#1}{%
327   \begin{beamercolorbox}[vmode]{block body#1}}{
328   \ifbeamercoloreempty[bg]{block body}{%
329     \begin{beamercolorbox}[vmode]{block body#1}%
330   }{%
331     \begin{beamercolorbox}[sep=\neo@blocksep, vmode]{block body#1}%
332     \vspace{-\neo@parskip}
333   }}%
334   \usebeamerfont{block body#1}%
335   \setlength{\parskip}{\neo@parskip}%
336 }

```

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

```

337 \setbeamertemplate{block begin}{\neo@block{}}
338 \setbeamertemplate{block alerted begin}{\neo@block{ alerted}}
339 \setbeamertemplate{block example begin}{\neo@block{ example}}
340 \setbeamertemplate{block end}{\end{beamercolorbox}\vspace*{0.2ex}}
341 \setbeamertemplate{block alerted end}{\end{beamercolorbox}\vspace*{0.2ex}}
342 \setbeamertemplate{block example end}{\end{beamercolorbox}\vspace*{0.2ex}}

```

8.2.6 Lists and floats

```

343 \setbeamertemplate{itemize items}{\raise1pt\hbox{\vrule width 0.8ex height 0.8ex}}
344 \setbeamertemplate{itemize subitem}{\raise1pt\hbox{\vrule width 0.5ex height 0.5ex}}
345 \setbeamertemplate{itemize subsubitem}{\raise.5ex\hbox{\vrule width 1ex height 0.2ex}}
346 \defbeamertemplate{description item}{align left}{\insertdescriptionitem\hfill}
347 \setbeamertemplate{caption label separator}{: }
348 \setbeamertemplate{caption}[numbered]

```

8.2.7 Footnotes

```

349 \setbeamertemplate{footnote}{%
350   \parindent 0em\noindent%
351   \raggedright
352   \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\hangindent=0.8em
353 }

```

8.2.8 Text and spacing settings

```
354 \newlength{\neo@parskip}
355 \setlength{\neo@parskip}{0.5em}
356 \setlength{\parskip}{\neo@parskip}
357 \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](#).

```
358 \define@key{beamerframe}{c}[true]{% centered
359   \beamer@frametopskip=0pt plus 1fill\relax%
360   \beamer@framebottomskip=0pt plus 1fill\relax%
361   \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
362   \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
363   \def\beamer@initfirstlineunskip{}}%
364 }
```

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.

```
365 \providebool{neo@standout}
366 \define@key{beamerframe}{standout}[true]{%
367   \booltrue{neo@standout}
368   \begingroup
369     \setkeys{beamerframe}{c,plain}
370     \ifbeamercolorempy[bg]{palette primary}{
371       \setbeamercolor{background canvas}{
372         use=palette primary,
373         bg=-palette primary.fg
374       }
375     }
```

```

375   }{
376     \setbeamercolor{background canvas}{
377       use=palette primary,
378       bg=palette primary.bg
379     }
380   }
381   \setbeamercolor{local structure}{
382     fg=palette primary.fg
383   }
384   \usebeamercolor[fg]{palette primary}
385 }

```

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.

```

386 \pretocmd{\beamer@reseteecodes}{%
387   \ifbool{neo@standout}{
388     \endgroup
389     \boolfalse{neo@standout}
390   }{}
391 }{}{}

```

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.

```

392 \AtBeginEnvironment{beamer@frameslide}{
393   \ifbool{neo@standout}{
394     \centering
395     \usebeamerfont{standout}
396   }{}
397 }

```

8.2.10 Process package options

```

398 \neo@inner@setdefaults
399 \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

```
400 \RequirePackage{etoolbox}
401 \RequirePackage{calc}
402 \RequirePackage{pgfopts}
```

8.3.2 Options

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

```
403 \pgfkeys{
404   /neo/outer/frametitle icon/.cd,
405   .is choice,
406   none/.code=\setbeamertemplate{frametitle icon}[none],
407   i4/.code=\setbeamertemplate{frametitle icon}[i4],
408   fau/.code=\setbeamertemplate{frametitle icon}[fau],
409 }
```

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

```
410 \pgfkeys{
411   /neo/outer/numbering/.cd,
412   .is choice,
413   none/.code=\setbeamertemplate{frame numbering}[none],
414   counter/.code=\setbeamertemplate{frame numbering}[counter],
415   fraction/.code=\setbeamertemplate{frame numbering}[fraction],
416 }
```

`progressbar` Adds a progress bar to the top, bottom, or frametitle of each slide.

```
417 \pgfkeys{
418   /neo/outer/progressbar/.cd,
419   .is choice,
420   none/.code={%
421     \setbeamertemplate{headline}[plain]
422     \setbeamertemplate{frametitle}[plain]
```

```

423     \setbeamertemplate{footline}[plain]
424 },
425 head/.code={\pgfkeys{/neo/outer/progressbar=none}
426     \addtobeamertemplate{headline}{}{}%
427     \usebeamertemplate*{progress bar in head/foot}
428 }
429 },
430 frametitle/.code={\pgfkeys{/neo/outer/progressbar=none}
431     \addtobeamertemplate{frametitle}{}{}%
432     \usebeamertemplate*{progress bar in head/foot}
433 }
434 },
435 foot/.code={\pgfkeys{/neo/outer/progressbar=none}
436     \addtobeamertemplate{footline}{}{}%
437     \usebeamertemplate*{progress bar in head/foot}%
438 }
439 },
440 }

```

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

```

441 \newcommand{\neo@outer@setdefaults}{
442     \pgfkeys{/neo/outer/.cd,
443     frametitle icon=none,
444     numbering=counter,
445     progressbar=none,
446 }
447 }

```

8.3.3 Head and footline

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

```

448 \setbeamertemplate{navigation symbols}{}

```

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

```

449 \defbeamertemplate{frametitle icon}{none}{}
450 \defbeamertemplate{frametitle icon}{i4}{\hfill\raisebox{-.25\height}{\includegraphics[width=1cm]{i4.png}}}

```

```
451 \defbeamertemplate{frametitle icon}{fau}{ \hfill\raisebox{-.25\height}{\includegraphics[width=1cm]{fau.png}}}
```

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

```
452 \defbeamertemplate{frame footer}{none}{}
453 \defbeamertemplate{frame footer}{custom}[1]{ #1 }
```

Add strut to ensure that frame numbers don't jump

```
454 \newcommand{\neo@framenumberingstrut}{\vphantom{0123456789}}
455 \defbeamertemplate{frame numbering}{none}{}
456 \defbeamertemplate{frame numbering}{counter}{\neo@framenumberingstrut\insertframenum}
457 \defbeamertemplate{frame numbering}{fraction}{
458   \neo@framenumberingstrut\insertframenum/\inserttotalframenum
459 }
```

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

footline

```
460 \defbeamertemplate{headline}{plain}{}
461 \defbeamertemplate{footline}{plain}{%
462   \begin{beamercolorbox}[wd=\textwidth, sep=3ex]{footline}%
463     \usebeamerfont{page number in head/foot}%
464     \usebeamertemplate*{frame footer}
465     \hfill%
466     \usebeamertemplate*{frame numbering}
467   \end{beamercolorbox}%
468 }
```

8.3.4 Frametitle

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

```
469 \newlength{\neo@frametitle@padding}
470 \setlength{\neo@frametitle@padding}{2.2ex}
471 \newcommand{\neo@frametitlestrut@start}{
472   \rule{0pt}{\neo@frametitle@padding +%
473     \totalheightof{
474       \ifcsdef{neo@frametitleformat}{\neo@frametitleformat X}{X}%
475     }%
476 }
```



```

476 }%
477 }
478 \newcommand{\neo@frametitlestrut@end}{
479 \rule[-\neo@frametitle@padding]{0pt}{\neo@frametitle@padding}
480 }
481 \defbeamertemplate{frametitle}{plain}{%
482 \nointerlineskip
483 \begin{beamercolorbox}[%
484 wd=\paperwidth,%
485 sep=0pt,%
486 leftskip=\neo@frametitle@padding,%
487 rightskip=\neo@frametitle@padding,%
488 ]{frametitle}%
489 \neo@frametitlestrut@start%
490 \insertframetitle%
491 \usebeamertemplate*{frametitle icon}%
492 \nolinebreak%
493 \neo@frametitlestrut@end%
494 \end{beamercolorbox}%
495 }
496 \setbeamertemplate{frametitle continuation}{%
497 \usebeamerfont{frametitle}
498 \romannumeral \insertcontinuationcount
499 }

```

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.

```

500 \newlength{\neo@progressinheadfoot}
501 \newlength{\neo@progressinheadfoot@linewidth}
502 \setlength{\neo@progressinheadfoot@linewidth}{0.8pt}
503 \setbeamertemplate{progress bar in head/foot}{
504 \nointerlineskip
505 \setlength{\neo@progressinheadfoot}{%
506 \paperwidth * \ratio{\insertframenum pt}{\inserttotalframenum pt}}%
507 }%
508 \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
509 \tikzexternaldisable%
510 \begin{tikzpicture}

```

```

511     \fill[bg] (0,0) rectangle (\paperwidth, \neo@progressinheadfoot@linewidth);
512     \fill[fg] (0,0) rectangle (\neo@progressinheadfoot, \neo@progressinheadfoot@l
513 \end{tikzpicture}%
514 \tikzexternalenable%
515 \end{beamercolorbox}
516 }

```

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`.

```

517 \AtBeginDocument{%
518   \apptocmd{\appendix}{%
519     \pgfkeys{%
520       /neo/outer/.cd,
521       numbering=none,
522       progressbar=none}
523   }{}{}
524 }

```

8.3.5 Process package options

```

525 \neo@outer@setdefaults
526 \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

```

527 \RequirePackage{etoolbox}
528 \RequirePackage{ifxetex}
529 \RequirePackage{ifluatex}
530 \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.

```
531 \ifboolexpr{bool {xetex} or bool {luatex}}{
532   \@ifpackageloaded{fontspec}{
533     \PassOptionsToPackage{no-math}{fontspec}
534   }{
535     \RequirePackage[no-math]{fontspec}
536   }

537   \IfFileExists{FiraSans-Regular.otf}{
538     \defaultfontfeatures{
539       Scale      = 1.0,
540       Extension = .otf
541     }
542   }{
543     \PackageWarning{beamerthemeneo}{%
544       FiraSans font not found in path, therefore using system fonts. %
545       Make sure you have the fonts installed.%
546     }
547   }
548   \setmonofont
549     [ Numbers = {Monospaced,Lining},
550       UprightFont   = *-Regular ,
551       ItalicFont    = *-Regular ,
552       BoldFont      = *-Medium ,
553       BoldItalicFont = *-Medium ,
554     ]
555     {FiraMono}
556   \newcommand{\neo@fontsave}{
557     \let\firaneofamily\sfdefault
558     \renewcommand*\familydefault{\firaneofamily}
559   }
560   \newcommand{\neo@fontlight}{
561     \setsansfont[
562       Numbers = {OldStyle, Monospaced},
563       UprightFont   = *-Light ,
564       ItalicFont    = *-LightItalic ,
```

```

565         BoldFont      = *-Regular ,
566         BoldItalicFont = *-RegularItalic ,
567     ]{FiraSans}
568     \neo@fontsave
569 }
570 \newcommand{\neo@fontbook}{
571     \setsansfont[
572         Numbers = {OldStyle, Monospaced},
573         UprightFont      = *-Book ,
574         ItalicFont       = *-BookItalic ,
575         BoldFont         = *-Medium ,
576         BoldItalicFont   = *-MediumItalic ,
577     ]{FiraSans}
578     \neo@fontsave
579 }
580 \newcommand{\neo@fontregular}{
581     \setsansfont[
582         Numbers = {OldStyle, Monospaced},
583         UprightFont      = *-Regular ,
584         ItalicFont       = *-RegularItalic ,
585         BoldFont         = *-SemiBold ,
586         BoldItalicFont   = *-SemiBoldItalic ,
587     ]{FiraSans}
588     \neo@fontsave
589 }
590 \AtBeginEnvironment{tabular}{%
591     \addfontfeature{Numbers={Monospaced}}%
592 }
593 }{%
594     \RequirePackage[utf8]{inputenc}
595     \IfFileExists{FiraSans.sty}{
596         \RequirePackage[T1]{fontenc}
597         \RequirePackage[sfdefault]{FiraSans}
598         \RequirePackage[nomap,lining]{FiraMono}
599         \def\bfseries@tt{mb}
600         \newcommand{\neo@fontsave}{
601             \edef\familydefault{\sfdefault}
602             \edef\seriesdefault{\mdseries@sf}
603         }
604         \newcommand{\neo@fontlight}{

```

```

605     \def\mdseries@sf{l}
606     \def\bfseries@sf{m}
607     \neo@fontsave
608   }
609   \newcommand{\neo@fontbook}{
610     \def\bfseries@sf{mb}
611     \neo@fontsave
612   }
613   \newcommand{\neo@fontregular}{
614     \def\mdseries@sf{m}
615     \def\bfseries@sf{sb}
616     \neo@fontsave
617   }
618   }{
619     \PackageWarning{beamerthemeneo}{%
620       You need to install the Fira Fonts package or compile with XeLaTeX or %
621       LuaLaTeX to use the included Fira fonts%
622     }
623   }
624 }

```

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

```

625 \setbeamerfont{title}{size=\Large,%
626           series=\bfseries}
627 \setbeamerfont{author}{size=\small}
628 \setbeamerfont{date}{size=\small}
629 \setbeamerfont{section title}{size=\Large,%
630           series=\bfseries}
631 \setbeamerfont{block title}{size=\normalsize,%
632           series=\bfseries}
633 \setbeamerfont{block title alerted}{size=\normalsize,%
634           series=\bfseries}
635 \setbeamerfont*{subtitle}{size=\large}
636 \setbeamerfont{frametitle}{size=\large,%
637           series=\bfseries}

```

```

638 \setbeamerfont{caption}{size=\small}
639 \setbeamerfont{caption name}{series=\bfseries}
640 \setbeamerfont{description item}{series=\bfseries}
641 \setbeamerfont{page number in head/foot}{size=\scriptsize}
642 \setbeamerfont{bibliography entry author}{size=\normalsize,%
643                                     series=\normalfont}
644 \setbeamerfont{bibliography entry title}{size=\normalsize,%
645                                     series=\bfseries}
646 \setbeamerfont{bibliography entry location}{size=\normalsize,%
647                                     series=\normalfont}
648 \setbeamerfont{bibliography entry note}{size=\small,%
649                                     series=\normalfont}
650 \setbeamerfont{standout}{size=\Large,%
651                                     series=\bfseries}

```

8.4.4 Font style options

`titleformat title` Controls the overall font style.

```

652 \pgfkeys{
653   /neo/font/style/.cd,
654   .is choice,
655   light/.code={\neo@fontlight},
656   book/.code={\neo@fontbook},
657   regular/.code={\neo@fontregular},
658 }

```

8.4.5 Title format options

`titleformat title` Controls the format of the title.

```

659 \pgfkeys{
660   /neo/font/titleformat title/.cd,
661   .is choice,
662   regular/.code={%
663     \let\neo@titleformat\@empty%
664     \setbeamerfont{title}{shape=\normalfont}%
665   },
666   smallcaps/.code={%
667     \let\neo@titleformat\@empty%

```

```

668     \setbeamerfont{title}{shape=\scshape}%
669   },
670   allsmallcaps/.code={%
671     \let\neo@titleformat\lowercase%
672     \setbeamerfont{title}{shape=\scshape}%
673     \PackageWarning{beamerthemeneo}{%
674       Be aware that titleformat title=allsmallcaps can lead to problems%
675     }
676   },
677   allcaps/.code={%
678     \let\neo@titleformat\uppercase%
679     \setbeamerfont{title}{shape=\normalfont}%
680     \PackageWarning{beamerthemeneo}{%
681       Be aware that titleformat title=allcaps can lead to problems%
682     }
683   },
684 }

```

`titleformat subtitle` Control the format of the subtitle.

```

685 \pgfkeys{
686   /neo/font/titleformat subtitle/.cd,
687   .is choice,
688   regular/.code={%
689     \let\neo@subtitleformat\@empty%
690     \setbeamerfont{subtitle}{shape=\normalfont}%
691   },
692   smallcaps/.code={%
693     \let\neo@subtitleformat\@empty%
694     \setbeamerfont{subtitle}{shape=\scshape}%
695   },
696   allsmallcaps/.code={%
697     \let\neo@subtitleformat\lowercase%
698     \setbeamerfont{subtitle}{shape=\scshape}%
699     \PackageWarning{beamerthemeneo}{%
700       Be aware that titleformat subtitle=allsmallcaps can lead to problems%
701     }
702   },
703   allcaps/.code={%
704     \let\neo@subtitleformat\uppercase%

```

```

705     \setbeamerfont{subtitle}{shape=\normalfont}%
706     \PackageWarning{beamerthemeneo}{%
707       Be aware that titleformat subtitle=allcaps can lead to problems%
708     }
709   },
710 }

```

titleformat section Controls the format of the section title.

```

711 \pgfkeys{
712   /neo/font/titleformat section/.cd,
713   .is choice,
714   regular/.code={%
715     \let\neo@sectiontitleformat\@empty%
716     \setbeamerfont{section title}{shape=\normalfont}%
717   },
718   smallcaps/.code={%
719     \let\neo@sectiontitleformat\@empty%
720     \setbeamerfont{section title}{shape=\scshape}%
721   },
722   allsmallcaps/.code={%
723     \let\neo@sectiontitleformat\MakeLowercase%
724     \setbeamerfont{section title}{shape=\scshape}%
725     \PackageWarning{beamerthemeneo}{%
726       Be aware that titleformat section=allsmallcaps can lead to problems%
727     }
728   },
729   allcaps/.code={%
730     \let\neo@sectiontitleformat\MakeUppercase%
731     \setbeamerfont{section title}{shape=\normalfont}%
732     \PackageWarning{beamerthemeneo}{%
733       Be aware that titleformat section=allcaps can lead to problems%
734     }
735   },
736 }

```

frametitleformat Control the format of the frame title.

```

737 \pgfkeys{
738   /neo/font/titleformat frame/.cd,
739   .is choice,

```



```

740 regular/.code={%
741   \let\neo@frametitleformat\@empty%
742   \setbeamerfont{frametitle}{shape=\normalfont}%
743 },
744 smallcaps/.code={%
745   \let\neo@frametitleformat\@empty%
746   \setbeamerfont{frametitle}{shape=\scshape}%
747 },
748 allsmallcaps/.code={%
749   \let\neo@frametitleformat\MakeLowercase%
750   \setbeamerfont{frametitle}{shape=\scshape}%
751   \PackageWarning{beamerthemeneo}{%
752     Be aware that titleformat frame=allsmallcaps can lead to problems%
753   }
754 },
755 allcaps/.code={%
756   \let\neo@frametitleformat\MakeUppercase%
757   \setbeamerfont{frametitle}{shape=\normalfont}%
758   \PackageWarning{beamerthemeneo}{%
759     Be aware that titleformat frame=allcaps can lead to problems%
760   }
761 },
762 }

```

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

```

763 \pgfkeys{
764   /neo/font/.cd,
765   titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
766   titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
767   titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
768   titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
769 }

```

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

```

770 \newcommand{\neo@font@setdefaults}{
771   \pgfkeys{/neo/font/.cd,
772     style=book,
773     titleformat title=regular,

```

```

774 titleformat subtitle=regular,
775 titleformat section=regular,
776 titleformat frame=regular,
777 }
778 }

```

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

```

779 \def\neo@titleformat#1{#1}
780 \def\neo@subtitleformat#1{#1}
781 \def\neo@sectiontitleformat#1{#1}
782 \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](#).

```

783 \patchcmd{\beamer@title}%
784   {\def\inserttitle{#2}}%
785   {\def\inserttitle{\neo@titleformat{#2}}}%
786   {}%
787   {\PackageError{beamerfontthemeneo}{Patching title failed}\@ehc}
788 \patchcmd{\beamer@subtitle}%
789   {\def\insertsubtitle{#2}}%
790   {\def\insertsubtitle{\neo@subtitleformat{#2}}}%
791   {}%
792   {\PackageError{beamerfontthemeneo}{Patching subtitle failed}\@ehc}
793 \patchcmd{\sectionentry}
794   {\def\insertsectionhead{#2}}
795   {\def\insertsectionhead{\neo@sectiontitleformat{#2}}}
796   {}
797   {\PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc}
798 \@tempwafalse
799 \patchcmd{\beamer@section}
800   {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#
801   {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
802     \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
803   {\@tempwattrue}
804   {}
805 \patchcmd{\beamer@section}

```

```

806 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
807 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
808   \neo@sectiontitleformat{#1}}}}
809 {\@tempswattrue}
810 {}
811 \patchcmd{\beamer@section}
812 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#
813 {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
814   \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
815 {\@tempswattrue}
816 {}
817 \patchcmd{\beamer@section}
818 {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#1}
819 {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
820   \noexpand\neo@sectiontitleformat{#1}}}}}
821 {\@tempswattrue}
822 {}
823 \if@tempswa\else
824   \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
825 \fi
826 \@tempswafalse
827 \patchcmd{\beamer@subsection}
828 {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded{#
829 {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
830   \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
831 {\@tempswattrue}
832 {}
833 \patchcmd{\beamer@subsection}
834 {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
835 {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{%
836   \neo@sectiontitleformat{#1}}}}
837 {\@tempswattrue}
838 {}
839 \patchcmd{\beamer@subsection}
840 {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#
841 {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
842   \noexpand\neo@sectiontitleformat{#1}}}}}
843 {\@tempswattrue}
844 {}
845 \if@tempswa\else

```

```

846 \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
847 \fi

```

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

```

848 \patchcmd{\beamer@@frametitle}
849   {%
850     \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
851     \usebeamertemplate*{frametitle continuation}\fi}}%
852   \gdef\beamer@frametitle{#2}%
853   \gdef\beamer@shortframetitle{#1}%
854   }}
855   {%
856     \gdef\insertframetitle{{\neo@frametitleformat{#2}\ifnum%
857     \beamer@autobreakcount>0\relax{}\space%
858     \usebeamertemplate*{frametitle continuation}\fi}}%
859   \gdef\beamer@frametitle{#2}%
860   \gdef\beamer@shortframetitle{#1}%
861   }}
862 {}
863 {\PackageError{beamerfontthemeneo}{Patching frame title failed}\@ehc}

```

8.4.6 Process package options

```

864 \neo@font@setdefaults
865 \ProcessPgfPackageOptions{/neo/font}

```

8.5 NEO color theme

8.5.1 Package dependencies

```

866 \RequirePackage{pgfopts}

```

8.5.2 Options

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

```

867 \pgfkeys{
868   /neo/color/block/.cd,

```

```

869     .is choice,
870     transparent/.code=\neo@block@transparent,
871     fill/.code=\neo@block@fill,
872 }

```

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

```

873 \pgfkeys{
874   /neo/color/background/.cd,
875   .is choice,
876   dark/.code=\neo@colors@dark,
877   light/.code=\neo@colors@light,
878 }

```

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

```

879 \newcommand{\neo@color@setdefaults}{
880   \pgfkeys{/neo/color/.cd,
881     background=light,
882     block=transparent,
883   }
884 }

```

8.5.3 Base colors

```

885
886 \definecolor{nDarkGrey}{RGB}{152,164,174}
887 \definecolor{nGrey}{RGB}{210,213,215}
888 \definecolor{nLightGrey}{RGB}{235,236,238}
889
890 \definecolor{nDarkRed}{RGB}{141,20,41}
891 \definecolor{nRed}{RGB}{201,169,147}
892 \definecolor{nLightRed}{RGB}{237,231,222}
893
894 \definecolor{nDarkGreen}{RGB}{0,155,119}
895 \definecolor{nGreen}{RGB}{170,207,189}
896 \definecolor{nLightGreen}{RGB}{229,239,234}
897
898 \definecolor{nDarkBlue}{RGB}{0,56,101}

```

```

899 \definecolor{nBlue}{RGB}{144,167,198}
900 \definecolor{nLightBlue}{RGB}{221,229,240}
901
902 \definecolor{nDarkCyan}{RGB}{0,177,235}
903 \definecolor{nCyan}{RGB}{180,214,245}
904 \definecolor{nLightCyan}{RGB}{234,243,252}
905
906 \definecolor{nDarkYellow}{RGB}{201,147,19}
907 \definecolor{nYellow}{RGB}{217,198,137}
908 \definecolor{nLightYellow}{RGB}{243,238,223}
909
910 \definecolor{nBlack}{HTML}{011F32}
911 \definecolor{nWhite}{RGB}{250,250,250}

```

8.5.4 Alias colors

Support the colors provided by the old i4 beamer theme.

```

912 \colorlet{i4red}{nDarkRed}
913 \colorlet{i4green}{nDarkGreen}
914 \colorlet{i4blue}{nDarkBlue}
915 \colorlet{i4cyan}{nDarkCyan}
916 \colorlet{i4yellow}{nDarkYellow}
917 \colorlet{i4grey}{nDarkGrey}
918 \definecolor{darkred}{rgb}{0.8,0,0}
919 \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`.

```

920 \newcommand{\neo@colors@dark}{
921   \setbeamercolor{normal text}{%
922     fg=nWhite,
923     bg=nBlack
924   }
925   \setbeamercolor{normal item}{%
926     fg=nWhite,
927     bg=nDarkBlue

```

```

928 }
929 \usebeamercolor[fg]{normal text}
930 }
931 \newcommand{\neo@colors@light}{
932   \setbeamercolor{normal text}{%
933     fg=nBlack,
934     bg=nWhite
935   }
936   \setbeamercolor{normal item}{%
937     fg=nDarkBlue,
938     bg=nWhite
939   }
940 }
941 \setbeamercolor{alerted text}{%
942   fg=nDarkRed
943 }
944 \setbeamercolor{example text}{%
945   fg=nDarkYellow
946 }

```

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`.

```

947 \setbeamercolor{titlelike}{use=normal text, parent=normal text}
948 \setbeamercolor{author}{use=normal text, parent=normal text}
949 \setbeamercolor{date}{use=normal text, parent=normal text}
950 \setbeamercolor{institute}{use=normal text, parent=normal text}
951 \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.

```

952 \setbeamercolor{palette primary}{%
953   use=normal text,
954   fg=normal text.bg,

```

```

955   bg=nDarkBlue
956 }
957 \setbeamercolor{frametitle}{%
958   use=palette primary,
959   parent=palette primary
960 }

```

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`.

```

961 \setbeamercolor{progress bar}{%
962   use=normal text,
963   fg=nDarkBlue,
964   bg=nLightBlue
965 }
966 \setbeamercolor{title separator}{
967   use=progress bar,
968   parent=progress bar
969 }
970 \setbeamercolor{progress bar in head/foot}{%
971   use=normal text.fg,
972   fg=nBlack,
973   parent=progress bar
974 }
975 \setbeamercolor{progress bar in section page}{
976   use=progress bar,
977   parent=progress bar
978 }

```

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.

```

979 \newcommand{\neo@block@transparent}{
980   \setbeamercolor{block title}{%
981     use=normal text,
982     fg=nDarkBlue,

```



```

983     bg=
984 }
985 \setbeamercolor{block title alerted}{%
986     use={block title, alerted text},
987     bg=block title.bg,
988     fg=alerted text.fg
989 }
990 \setbeamercolor{block title example}{%
991     use={block title, example text},
992     bg=block title.bg,
993     fg=example text.fg
994 }
995 \setbeamercolor{block body}{
996     bg=
997 }
998 \setbeamercolor{block body alerted}{
999     use=block body,
1000     parent=block body
1001 }
1002 \setbeamercolor{block body example}{
1003     use=block body,
1004     parent=block body
1005 }
1006 }
1007 \newcommand{\neo@block@fill}{
1008     \setbeamercolor{block title}{%
1009         use=normal text,
1010         fg=nDarkBlue,
1011         bg=nGrey
1012     }
1013     \setbeamercolor{block title alerted}{%
1014         use={block title, alerted text},
1015         bg=alerted text.fg,
1016         fg=alerted text.bg
1017     }
1018     \setbeamercolor{block title example}{%
1019         use={block title, example text},
1020         bg=example text.fg,
1021         fg=example text.bg
1022     }

```

```

1023 \setbeamercolor{block body}{
1024   use={block title, normal text},
1025   bg=nLightGrey
1026 }
1027 \setbeamercolor{block body alerted}{
1028   use=block body,
1029   parent=block body,
1030   bg=nRed!50,
1031 }
1032 \setbeamercolor{block body example}{
1033   use=block body,
1034   parent=block body,
1035   bg=nYellow!50
1036 }
1037 }
1038

```

Footnotes

```

1039 \setbeamercolor{footnote}{fg=normal text.fg!90}
1040 \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.

```

1041 \setbeamercolor{bibliography entry author}{fg=, bg=}
1042 \setbeamercolor{bibliography entry title}{fg=, bg=}
1043 \setbeamercolor{bibliography entry location}{fg=, bg=}
1044 \setbeamercolor{bibliography entry note}{fg=, bg=}

```

8.5.7 Process package options

```

1045 \neo@color@setdefaults
1046 \ProcessPgfPackageOptions{/neo/color}
1047 \mode<all>

```

8.6 Tol pgfplots theme

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

```
1048 \definecolor{TolDarkPurple}{HTML}{332288}
1049 \definecolor{TolDarkBlue}{HTML}{6699CC}
1050 \definecolor{TolLightBlue}{HTML}{88CCFF}
1051 \definecolor{TolLightGreen}{HTML}{44AA99}
1052 \definecolor{TolDarkGreen}{HTML}{117733}
1053 \definecolor{TolDarkBrown}{HTML}{999933}
1054 \definecolor{TolLightBrown}{HTML}{DDCC77}
1055 \definecolor{TolDarkRed}{HTML}{661100}
1056 \definecolor{TolLightRed}{HTML}{CC6677}
1057 \definecolor{TolLightPink}{HTML}{AA4466}
1058 \definecolor{TolDarkPink}{HTML}{882255}
1059 \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.

```
1060 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
1061   {draw=TolDarkBlue, fill=TolDarkBlue!70},
1062   {draw=TolLightBrown, fill=TolLightBrown!70},
1063   {draw=TolLightGreen, fill=TolLightGreen!70},
1064   {draw=TolDarkPink, fill=TolDarkPink!70},
1065   {draw=TolDarkPurple, fill=TolDarkPurple!70},
1066   {draw=TolDarkRed, fill=TolDarkRed!70},
1067   {draw=TolDarkBrown, fill=TolDarkBrown!70},
1068   {draw=TolLightRed, fill=TolLightRed!70},
1069   {draw=TolLightPink, fill=TolLightPink!70},
1070   {draw=TolLightPurple, fill=TolLightPurple!70},
1071   {draw=TolLightBlue, fill=TolLightBlue!70},
1072   {draw=TolDarkGreen, fill=TolDarkGreen!70},
1073 }
```

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

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

```

1074 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
1075   {TolDarkBlue, mark=*, mark size=1.5pt},
1076   {TolLightBrown, mark=square*, mark size=1.3pt},
1077   {TolLightGreen, mark=triangle*, mark size=1.5pt},
1078   {TolDarkBrown, mark=diamond*, mark size=1.5pt},
1079 }

```

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.

```

1080 \pgfplotsset{
1081   compat=1.9,

```

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

```

1082   mlineplot/.style={
1083     mbaseplot,
1084     xmajorgrids=true,
1085     ymajorgrids=true,
1086     major grid style={dotted},
1087     axis x line=bottom,
1088     axis y line=left,
1089     legend style={
1090       cells={anchor=west},
1091       draw=none
1092     },
1093     cycle list name=mlineplot cycle,
1094   },

```

`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`.

```

1095   mbarplot base/.style={
1096     mbaseplot,
1097     bar width=6pt,
1098     axis y line*=none,

```

```

1099 },
1100 mbarplot/.style={
1101     mbarplot base,
1102     ybar,
1103     xmajorgrids=false,
1104     ymajorgrids=true,
1105     area legend,
1106     legend image code/.code={%
1107         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1108     },
1109     cycle list name=mbarplot cycle,
1110 },
1111 horizontal mbarplot/.style={
1112     mbarplot base,
1113     xmajorgrids=true,
1114     ymajorgrids=false,
1115     xbar stacked,
1116     area legend,
1117     legend image code/.code={%
1118         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
1119     },
1120     cycle list name=mbarplot cycle,
1121 },

```

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

```

1122 mbaseplot/.style={
1123     legend style={
1124         draw=none,
1125         fill=none,
1126         cells={anchor=west},
1127     },
1128     x tick label style={
1129         font=\footnotesize
1130     },
1131     y tick label style={
1132         font=\footnotesize
1133     },
1134     legend style={
1135         font=\footnotesize

```

```

1136     },
1137     major grid style={
1138         dotted,
1139     },
1140     axis x line*=bottom,
1141 },
1142 disable thousands separator/.style={
1143     /pgf/number format/.cd,
1144     1000 sep={}
1145 },
1146 }

```