Modern Beamer Presentations with the **NEO** package

v1.0 — 2017/10/01

Contents

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 XƎLaTeX to typeset your slides. However, **NEO** can also be used with other typefaces and LaTeX 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`
- `pgfopts`
- `etoolbox`
- `calc`
- `ifxetex`
- `ifluatex`

For best results, we recommend installing the fonts `Fira Sans` and `Fira Mono` and compiling with **NEO** using XƎLaTeX or LuaTeX. These are optional dependencies; **NEO** is compatible with (e.g.) pdfLaTeX 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` ***list of possible values*** .... default

A short description of the option.

3.1.1 Main theme

`titleformat` *regular, smallcaps, allsmallcaps, allcaps* regular

Changes the format of titles, subtitles, section titles, frame titles, and the text on "standout" frames. The available options produce Regular, SmallCaps, allsmallcaps, or ALLCAPS titles. Please refer to Section 6.1 for known issues with these options.

`titleformat plain` *regular, smallcaps, allsmallcaps, allcaps* regular

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

3.1.2 Inner theme

`sectionpage` *none, simple, progressbar* progressbar

Adds a slide at the start of each section (`simple`) with an optional thin progress bar below the section title (`progressbar`). The none option disables the section page.

`subsectionpage` *none, simple, progressbar* none

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

`startsection` *hide, show* show

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

`sectionaftertoc` *hide, show* show

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

3.1.3 Outer theme

`numbering` *none, counter, fraction* counter

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

`progressbar` *none, head, frametitle, foot* .... none

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

3.1.4 Color theme

`block` *transparent, fill* .... transparent

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

`background` *dark, light, white* .... light

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

3.1.5 Font theme

`titleformat title`
`titleformat subtitle`
`titleformat section`
`titleformat frame` *regular, smallcaps, allsmallcaps, allcaps* .... regular

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

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}
\setmathsfont(Digits)[Numbers={Lining, Proportional}]{Fira
    Sans Light}
```

3.4 Commands

3.4.1 Standout frames

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

```
\begin{frame}[standout]
```

```
    Thank you!
\end{frame}
```

4 pgfplots integration

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

4.1 Styles

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

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

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

`horizontal mbarplot` Plot horizontal bar charts.

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

4.2 Paul Tol colors

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

In a technical note for SRON, Paul Tol proposed a palette of colors satisfying these constraints. The sub-package `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 pdfLaTeX, 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 seperately:

```
\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 XƎLATEX, 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 XƎLATEX 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

```
\RequirePackage{etoolbox}
\RequirePackage{pgfpages}
\RequirePackage{pgfopts}
```

8.1.2 Options

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

```
\pgfkeys{/neo/.cd,
  .search also={
    /neo/inner,
    /neo/outer,
    /neo/color,
    /neo/font,
  }
}
```

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

```
\pgfkeys{
  /neo/titleformat plain/.cd,
    .is choice,
    regular/.code={%
      \let\neo@plaintitleformat\@empty%
      \setbeamerfont{standout}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\neo@plaintitleformat\@empty%
      \setbeamerfont{standout}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\neo@plaintitleformat\MakeLowercase%
      \setbeamerfont{standout}{shape=\scshape}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat plain=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\neo@plaintitleformat\MakeUppercase%
      \setbeamerfont{standout}{shape=\normalfont}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat plain=allcaps can lead to problems%
      }
    },
}
```

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

```
\pgfkeys{
  /neo/titleformat/.code=\pgfkeysalso{
      font/titleformat title=#1,
      font/titleformat subtitle=#1,
      font/titleformat section=#1,
      font/titleformat frame=#1,
      titleformat plain=#1,
    }
}
```

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

```
\pgfkeys{/neo/.cd,
  noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
  usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
  nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
  darkcolors/.code=\pgfkeysalso{color/background=dark},
  whitebg/.code=\pgfkeysalso{color/background=white},
  blockbg/.code=\pgfkeysalso{color/block=fill},
  light/.code=\pgfkeysalso{font/style=light},
  book/.code=\pgfkeysalso{font/style=book},
  regular/.code=\pgfkeysalso{font/style=regular},
  nostartsection/.code=\pgfkeysalso{inner/startsection=hide},
  nosectionaftertoc/.code=\pgfkeysalso{inner/sectionaftertoc=hide},
}
```

Set default values for options.

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

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.

```
\providecommand{\tikzexternalenable}{}
\providecommand{\tikzexternaldisable}{}
```

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

```
! Package beamerfontthemeneo Error: Patching frame title failed.
```

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

```
\@ifclasslater{beamer}{2023/02/20}{%
 % The hooks where first defined for v3.69
 \AddToHook{file/beamerfontthemeneo.sty/before}{\UseHook{file/beamerfontthememetr
 \AddToHook{file/beamerfontthemeneo.sty/after}{%
  \let\metropolis@frametitleformat\neo@frametitleformat
  \UseHook{file/beamerfontthememetropolis.sty/after}
 }
}{}
```

8.1.3 Component sub-packages

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

```
\useinnertheme{neo}
\useoutertheme{neo}
\usecolortheme{neo}
\usefonttheme{neo}
```

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

```
\AtEndPreamble{%
  \RequirePackage{pdfpc}
  \newcommand<>{\pnote}[2][item]{%
    \note[#1]#3{#2}%
    \ifbool{neo@pdfpcnotes}{%
      \only#3{{\escapechar=‘_ \xdef\tmpnote{\expandafter\detokenize\expandafter{#
    }{}%
  }
  \@ifpackageloaded{pgfplots}{%
    \RequirePackage{pgfplotsthemetol}
  }{}
}
```

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.

```
\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.

```
\def\neo@plaintitleformat#1{#1}
\newcommand{\plain}[2][]{%
  \PackageWarning{beamerthemeneo}{%
    The syntax ‘\plain’ may be deprecated in a future version of neo.
    Please use a frame with [standout] instead.
  }
  \begin{frame}[standout]{#1}
    \neo@plaintitleformat{#2}
  \end{frame}
}
```

\mreducelistspacing

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

8.1.5 Process package options

```
\neo@setdefaults
\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

```
\RequirePackage{etoolbox}
\RequirePackage{keyval}
\RequirePackage{calc}
\RequirePackage{pgfopts}
\RequirePackage{pgfpages}
\RequirePackage{tikz}
```

8.2.2 Options

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

```
\pgfkeys{
  /neo/inner/sectionpage/.cd,
    .is choice,
    none/.code=\neo@disablesectionpage,
    simple/.code={\neo@enablesectionpage
                  \setbeamertemplate{section page}[simple]},
    progressbar/.code={\neo@enablesectionpage
                       \setbeamertemplate{section page}[progressbar]},
}
```

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

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

```
                  \setbeamertemplate{section page}[simple]},
    progressbar/.code={\neo@enablesubsectionpage
                       \setbeamertemplate{section page}[progressbar]},
}
```

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

```
\pgfkeys{
  /neo/inner/startsection/.cd,
    .is choice,
    hide/.code={\global\booltrue{neo@hidestartsection}},
    show/.code={\global\boolfalse{neo@hidestartsection}},
}
```

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

```
\pgfkeys{
  /neo/inner/sectionaftertoc/.cd,
    .is choice,
    hide/.code={\global\booltrue{neo@hidesectionaftertoc}},
    show/.code={\global\boolfalse{neo@hidesectionaftertoc}},
}
```

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

```
\newcommand{\neo@inner@setdefaults}{
  \pgfkeys{/neo/inner/.cd,
    sectionpage=progressbar,
    subsectionpage=none,
    startsection=show,
    sectionaftertoc=show,
  }
}
```

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.

```
\setbeamertemplate{title page}{
  \begin{minipage}[b][0.95\paperheight]{\textwidth}
    \vfill%
    \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
    \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
    \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.

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

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

```
\renewcommand{\maketitle}[1][extern]{%
  \ifbeamer@inframe
    \titlepage
  \else
    {
      \usebackgroundtemplate{
        \tikzexternaldisable%
        \begin{tikzpicture}
          \node[anchor=north west,inner sep=0,outer sep=0] at (0, \paperheight) {
#1}};
          \fill[nWhite] (0,0) rectangle (\paperwidth, 0.3\paperheight);
        \end{tikzpicture}%
        \tikzexternalenable%
      }
      \frame[plain,noframenumbering]{
        \neo@colors@dark
        \setbeamercolor{title separator}{
          fg=black!20,
          bg=normal text.fg
        }
        \titlepage
      }
    }
  \fi
}
\def\titlepage{%
  \usebeamertemplate{title page}
}
```

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

```
\setbeamertemplate{title graphic}{
  \vbox to 0pt {
    \vspace*{2em}
    \inserttitlegraphic%
  }%
  \nointerlineskip%
}
```

title Set the title on the title page.

```
\setbeamertemplate{title}{
  \raggedright%
  \linespread{1.0}%
  \inserttitle%
  \par%
  \vspace*{0.5em}
}
```

subtitle Set the subtitle on the title page.

```
\setbeamertemplate{subtitle}{
  \raggedright%
  \insertsubtitle%
  \par%
  \vspace*{0.5em}
}
```

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

```
\newlength{\neo@titleseparator@linewidth}
\setlength{\neo@titleseparator@linewidth}{0.4pt}
\setbeamertemplate{title separator}{
  \tikzexternaldisable%
  \begin{tikzpicture}
    \fill[fg] (0,0) rectangle (\textwidth, \neo@titleseparator@linewidth);
  \end{tikzpicture}%
  \tikzexternalenable%
  \par%
}
```

author Set the author on the title page.

```
\setbeamertemplate{author}{
  \vspace*{2em}
  \insertauthor%
  \par%
  \vspace*{0.25em}
}
```

date Set the date on the title page.

```
\setbeamertemplate{date}{
  \insertdate%
  \par%
}
```

institute Set the institute on the title page.

```
\setbeamertemplate{institute}{
  \vspace*{3mm}
  \insertinstitute%
  \par%
}
```

8.2.4 Section page

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

```
\defbeamertemplate{section page}{simple}{
  \begin{center}
    \usebeamercolor[fg]{section title}
    \usebeamerfont{section title}
    \insertsectionhead\par
    \ifx\insertsubsectionhead\@empty\else
      \usebeamercolor[fg]{subsection title}
      \usebeamerfont{subsection title}
      \insertsubsectionhead
    \fi
  \end{center}
}
\defbeamertemplate{section page}{progressbar}{
  \centering
  \begin{minipage}{22em}
    \raggedright
    \usebeamercolor[fg]{section title}
    \usebeamerfont{section title}
    \insertsectionhead\\[-1ex]
    \usebeamertemplate*{progress bar in section page}
    \par
    \ifx\insertsubsectionhead\@empty\else%
      \usebeamercolor[fg]{subsection title}%
      \usebeamerfont{subsection title}%
      \insertsubsectionhead
    \fi
  \end{minipage}
  \par
  \vspace{\baselineskip}
}

\global\providebool{neo@hidestartsection}
\global\providebool{neo@hidesectionaftertoc}
\global\providebool{neo@tocprinted}
\global\boolfalse{neo@tocprinted}
\let\neo@oldtableofcontents\tableofcontents
\renewcommand{\tableofcontents}{
  \global\booltrue{neo@tocprinted}
  \neo@oldtableofcontents
}

\newcommand{\neo@disablesectionpage}{
  \AtBeginSection{
    % intentionally empty
  }
}
\newcommand{\neo@enablesectionpage}{
  \AtBeginSection{
    \ifbeamer@inframe
      \sectionpage
    \else
      \providebool{neo@printsection}
      \booltrue{neo@printsection}
      \ifbool{neo@hidestartsection}{
        \ifnum\theframenumber=0
          \boolfalse{neo@printsection}
        \fi
      }{}
      \ifbool{neo@hidesectionaftertoc}{
        \ifbool{neo@tocprinted}{%
          \global\boolfalse{neo@tocprinted}
          \boolfalse{neo@printsection}
        }{}
      }{}
      \ifbool{neo@printsection}{
        \frame[plain,c,noframenumbering]{\sectionpage}
      }{}
    \fi
  }
}
```

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

```
\setbeamertemplate{subsection page}{%
  \usebeamertemplate*{section page}
}
\newcommand{\neo@disablesubsectionpage}{
  \AtBeginSubsection{
    % intentionally empty
  }
}
\newcommand{\neo@enablesubsectionpage}{
  \AtBeginSubsection{
    \ifbeamer@inframe
      \subsectionpage
    \else
```

```
      \frame[plain,c,noframenumbering]{\subsectionpage}
    \fi
  }
}
```

ogress 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`.

```
\newlength{\neo@progressonsectionpage}
\newlength{\neo@progressonsectionpage@linewidth}
\setlength{\neo@progressonsectionpage@linewidth}{0.4pt}
\setbeamertemplate{progress bar in section page}{
  \pgfmathsetlength{\neo@progressonsectionpage}{\textwidth * min(1,\insertframenu
  \tikzexternaldisable%
  \begin{tikzpicture}
    \fill[bg] (0,0) rectangle (\textwidth, \neo@progressonsectionpage@linewidth);
    \fill[fg] (0,0) rectangle (\neo@progressonsectionpage, \neo@progressonsection
  \end{tikzpicture}%
  \tikzexternalenable%
}
```

8.2.5 Block environments

block
block alerted
block example

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

```
\newlength{\neo@blocksep}
\newlength{\neo@blockadjust}
\setlength{\neo@blocksep}{0.75ex}
\setlength{\neo@blockadjust}{0.25ex}
\providecommand{\neo@strut}{%
  \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz()}%
}
\newcommand{\neo@block}[1]{
  \par\vskip\medskipamount%
  \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.

```
  \ifbeamercolorempty[bg]{block title#1}{%
    \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}}{%
  \ifbeamercolorempty[bg]{block title}{%
    \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}%
  }%
%   \end{macrocode}
%
%   Otherwise, if the |block title| has a background, we set the padding based
%   on |\neo@blockskip|. However, we have to visually compensate for
%   the |\neo@strut| added to the block title (see below) by
%   subtracting |\neo@blockadjust| from the top and bottom padding.
%
%   \begin{macrocode}
  {%
    \begin{beamercolorbox}[
      sep=\dimexpr\neo@blocksep-\neo@blockadjust\relax,
      leftskip=\neo@blockadjust,
      rightskip=\dimexpr\neo@blockadjust plus 4em\relax
    ]{block title#1}%
  }}%
%   \end{macrocode}
%
%   We can now set the contents of the |block title|. The zero-width but
%   positive-height box |\neo@strut| ensures that the block title box
%   has a consistent height, even if it lacks punctuation, ascenders, or
%   descenders.
%
%   \begin{macrocode}
      \usebeamerfont*{block title#1}%
      \neo@strut%
      \insertblocktitle%
      \neo@strut%
  \end{beamercolorbox}%
%   \end{macrocode}
%
%   Next, we typeset the |block body|. This the code is similar to, but simpler
%   than, the |block title| code since we don’t need to adjust for any struts.
%
%   \begin{macrocode}
  \nointerlineskip%
  \ifbeamercolorempty[bg]{block body#1}{%
    \begin{beamercolorbox}[vmode]{block body#1}}{
  \ifbeamercolorempty[bg]{block body}{%
    \begin{beamercolorbox}[vmode]{block body#1}%
  }{%
    \begin{beamercolorbox}[sep=\neo@blocksep, vmode]{block body#1}%
    \vspace{-\neo@parskip}
  }}%
      \usebeamerfont{block body#1}%
      \setlength{\parskip}{\neo@parskip}%
}
```

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

```
\setbeamertemplate{block begin}{\neo@block{}}
\setbeamertemplate{block alerted begin}{\neo@block{ alerted}}
\setbeamertemplate{block example begin}{\neo@block{ example}}
\setbeamertemplate{block end}{\end{beamercolorbox}\vspace*{0.2ex}}
\setbeamertemplate{block alerted end}{\end{beamercolorbox}\vspace*{0.2ex}}
\setbeamertemplate{block example end}{\end{beamercolorbox}\vspace*{0.2ex}}
```

8.2.6 Lists and floats

```
\setbeamertemplate{itemize items}{\raise1pt\hbox{\vrule width 0.8ex height 0.8ex
\setbeamertemplate{itemize subitem}{\raise1pt\hbox{\vrule width 0.5ex height 0.5
\setbeamertemplate{itemize subsubitem}{\raise.5ex\hbox{\vrule width 1ex height 0.
\defbeamertemplate{description item}{align left}{\insertdescriptionitem\hfill}
\setbeamertemplate{caption label separator}{: }
\setbeamertemplate{caption}[numbered]
```

8.2.7 Footnotes

```
\setbeamertemplate{footnote}{%
  \parindent 0em\noindent%
  \raggedright
  \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\hangindent=0.
}
```

8.2.8 Text and spacing settings

```
\newlength{\neo@parskip}
\setlength{\neo@parskip}{0.5em}
\setlength{\parskip}{\neo@parskip}
\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.

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

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{fra`

`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.

```
\providebool{neo@standout}
\define@key{beamerframe}{standout}[true]{%
  \booltrue{neo@standout}
  \begingroup
    \setkeys{beamerframe}{c,plain}
    \ifbeamercolorempty[bg]{palette primary}{
      \setbeamercolor{background canvas}{
        use=palette primary,
        bg=-palette primary.fg
      }
    }{
      \setbeamercolor{background canvas}{
```

```
    use=palette primary,
    bg=palette primary.bg
   }
  }
  \setbeamercolor{local structure}{
   fg=palette primary.fg
  }
  \usebeamercolor[fg]{palette primary}
  \makeatletter
  \def\beamer@framenotesbegin{% at beginning of slide
   \usebeamercolor[fg]{palette primary}
   \gdef\beamer@noteitems{}%
   \gdef\beamer@notes{}%
  }
  \makeatother
}
```

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.

```
\pretocmd{\beamer@reseteecodes}{%
 \ifbool{neo@standout}{
  \endgroup
  \boolfalse{neo@standout}
 }{}
}{}{}
```

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.

```
\AtBeginEnvironment{beamer@frameslide}{
 \makeatletter
 \usebeamercolor[fg]{normal text}
 \gdef\beamer@noteitems{}%
 \gdef\beamer@notes{}%
 \makeatother
 \ifbool{neo@standout}{
  \centering
  \usebeamerfont{standout}
 }{}
```

```
  }
```

8.2.10 Process package options

```
\neo@inner@setdefaults
\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

```
\RequirePackage{etoolbox}
\RequirePackage{calc}
\RequirePackage{pgfpages}
\RequirePackage{pgfopts}
```

8.3.2 Options

icon Adds an icon to the frametitle on each slide.

```
\pgfkeys{
  /neo/outer/frametitle icon/.cd,
    .is choice,
    none/.code=\setbeamertemplate{frametitle icon}[none],
    i4/.code=\setbeamertemplate{frametitle icon}[i4],
    fau/.code=\setbeamertemplate{frametitle icon}[fau],
    fau-new/.code=\setbeamertemplate{frametitle icon}[fau-new],
}
```

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

```
\pgfkeys{
  /neo/outer/numbering/.cd,
    .is choice,
    none/.code=\setbeamertemplate{frame numbering}[none],
    counter/.code=\setbeamertemplate{frame numbering}[counter],
    fraction/.code=\setbeamertemplate{frame numbering}[fraction],
}
```

notes Show notes in presentation

```
\providebool{neo@pdfpcnotes}
\pgfkeys{
  /neo/outer/notes/.cd,
    .is choice,
    none/.code=\pgfkeysalso{notes=hide},
    hide/.code={\boolfalse{neo@pdfpcnotes}\setbeameroption{hide notes}},
    pdfpc/.code={\booltrue{neo@pdfpcnotes}\setbeameroption{hide notes}},
    show/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\se
    full/.code={\booltrue{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\set
    only/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\se
    preview-left/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[d
    preview-right/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[
    preview-top/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[de
    preview-bottom/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}
    preview-left-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note pag
big]\setbeameroption{show notes on second screen=left}},
    preview-right-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note pa
big]\setbeameroption{show notes on second screen=right}},
    preview-top-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page
big]\setbeameroption{show notes on second screen=top}},
    preview-bottom-big/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note p
big]\setbeameroption{show notes on second screen=bottom}},
    left/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\se
    right/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\s
    top/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\set
    bottom/.code={\boolfalse{neo@pdfpcnotes}\setbeamertemplate{note page}[print]\
}
```

footer Adds additional presentation information to the footer

```
\pgfkeys{
  /neo/outer/footer/.cd,
    .is choice,
    none/.code=\setbeamertemplate{frame footer}[none],
    author/.code=\setbeamertemplate{frame footer}[author],
    author title/.code=\setbeamertemplate{frame footer}[author title],
    author title section/.code=\setbeamertemplate{frame footer}[author title sect
    title/.code=\setbeamertemplate{frame footer}[title],
    title section/.code=\setbeamertemplate{frame footer}[title section],
    title section subsection/.code=\setbeamertemplate{frame footer}[title section
}
```

footer style Footer background color

```
\providebool{neo@standoutfooter}
\pgfkeys{
  /neo/outer/footer style/.cd,
    .is choice,
    plain/.code={\boolfalse{neo@standoutfooter}\setbeamertemplate{footline}[plair
    standout/.code={\booltrue{neo@standoutfooter}\setbeamertemplate{footline}[sta
}
```

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).

```
\pgfkeys{
  /neo/outer/progressbar/.cd,
    .is choice,
    none/.code={%
      \setbeamertemplate{headline}[plain]
      \setbeamertemplate{frametitle}[plain]
    },
    head/.code={\pgfkeys{/neo/outer/progressbar=none}
      \addtobeamertemplate{headline}{}{%
        \usebeamertemplate*{progress bar in head/foot}
      }
    },
    frametitle/.code={\pgfkeys{/neo/outer/progressbar=none}
      \addtobeamertemplate{frametitle}{}{%
        \usebeamertemplate*{progress bar in head/foot}
      }
    },
    foot/.code={\pgfkeys{/neo/outer/progressbar=none}
      \ifbool{neo@standoutfooter}{%
        \addtobeamertemplate{footline}{\usebeamertemplate*{progress bar in head/f
      }{%
        \addtobeamertemplate{footline}{}{\usebeamertemplate*{progress bar in heac
      }
    },
}
```

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

```
\newcommand{\neo@outer@setdefaults}{
  \pgfkeys{/neo/outer/.cd,
    frametitle icon=none,
    footer=none,
    footer style=plain,
    numbering=counter,
    progressbar=none,
  }
}
```

8.3.3 Head and footline

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

```
\setbeamertemplate{navigation symbols}{}
```

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

```
\defbeamertemplate{frametitle icon}{none}{}
\defbeamertemplate{frametitle icon}{i4}{ \hfill\raisebox{-.25\height}{\includegra
i4-white}}}
\defbeamertemplate{frametitle icon}{fau}{ \hfill\raisebox{-.25\height}{\includegr
fau-white}\kern-0.275em}}
```

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

```
\defbeamertemplate{frame footer}{none}{}
\defbeamertemplate{frame footer}{author}{
  \insertshortauthor%
}
\defbeamertemplate{frame footer}{author title}{
  \insertshortauthor%
  \hfill%
  \insertshorttitle%
  \hfill%
}
\defbeamertemplate{frame footer}{author title section}{
  \insertshortauthor%
  \hfill%
  \insertshorttitle%
  \hfill%
  \insertsection%
  \hfill%
}
\defbeamertemplate{frame footer}{title}{
  \insertshorttitle%
}
\defbeamertemplate{frame footer}{title section}{
  \insertshorttitle%
  \hfill%
  \insertsection%
  \hfill%
}
\defbeamertemplate{frame footer}{title section subsection}{
  \insertshorttitle%
  \hfill%
  \insertsection%
  \ifx\insertsubsection\@empty\else\ -- \insertsubsection\fi%%
  \hfill%
}
\defbeamertemplate{frame footer}{custom}[1]{ #1 }
```

Add strut to ensure that frame numbers don't jump

```
\newcommand{\neo@framenumberingstrut}{\vphantom{0123456789}}
\defbeamertemplate{frame numbering}{none}{}
\defbeamertemplate{frame numbering}{counter}{\neo@framenumberingstrut\insertframe
\defbeamertemplate{frame numbering}{fraction}{
  \neo@framenumberingstrut\insertframenumber/\inserttotalframenumber
}
```

headline
footline

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

```
\defbeamertemplate{headline}{plain}{}
\defbeamertemplate{footline}{plain}{%
  \begin{beamercolorbox}[wd=\textwidth, sep=1ex]{footline}%
    \usebeamerfont{page number in head/foot}%
    \usebeamertemplate*{frame footer}
    \hfill%
    \parbox{.1\framewidth}{\hfill\usebeamertemplate*{frame numbering}}
  \end{beamercolorbox}%
}
\defbeamertemplate{footline}{standout}{%
  \begin{beamercolorbox}[wd=\textwidth, sep=1ex]{palette primary}%
    \usebeamerfont{page number in head/foot}%
    \usebeamertemplate*{frame footer}
    \hfill%
    \parbox{.1\framewidth}{\hfill\usebeamertemplate*{frame numbering}}
  \end{beamercolorbox}%
}
```

8.3.4 Frametitle

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

```
\newlength{\neo@frametitle@padding}
\setlength{\neo@frametitle@padding}{2.2ex}
\newcommand{\neo@frametitlestrut@start}{
  \rule{0pt}{\neo@frametitle@padding +%
    \totalheightof{%
      \ifcsdef{neo@frametitleformat}{\neo@frametitleformat X}{X}%
    }%
  }%
}
\newcommand{\neo@frametitlestrut@end}{
  \rule[-\neo@frametitle@padding]{0pt}{\neo@frametitle@padding}
}
\defbeamertemplate{frametitle}{plain}{%
  \nointerlineskip%
  \begin{beamercolorbox}[%
      wd=\paperwidth,%
      sep=0pt,%
      leftskip=\neo@frametitle@padding,%
      rightskip=\neo@frametitle@padding,%
    ]{frametitle}%
  \neo@frametitlestrut@start%
  \insertframetitle%
  \ifx\insertframesubtitle\@empty
    \usebeamertemplate*{frametitle icon}%
  \else
    \hfill{\usebeamerfont{framesubtitle}\insertframesubtitle}%
  \fi
  \nolinebreak%
  \neo@frametitlestrut@end%
  \end{beamercolorbox}%
}
\setbeamertemplate{frametitle continuation}{%
  \usebeamerfont{frametitle}
   {\normalfont (\insertcontinuationcount)}
}
```

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.

```
\newlength{\neo@progressinheadfoot}
\newlength{\neo@progressinheadfoot@linewidth}
\setlength{\neo@progressinheadfoot@linewidth}{0.8pt}
\setbeamertemplate{progress bar in head/foot}{
  \nointerlineskip
  \pgfmathsetlength{\neo@progressinheadfoot}{\paperwidth * min(1,\insertframenumb
  \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
    \tikzexternaldisable%
    \begin{tikzpicture}
      \fill[bg] (0,0) rectangle (\paperwidth, \neo@progressinheadfoot@linewidth);
      \fill[fg] (0,0) rectangle (\neo@progressinheadfoot, \neo@progressinheadfoot
    \end{tikzpicture}%
    \tikzexternalenable%
  \end{beamercolorbox}
}
```

custom notes Templates for note pages

```
\defbeamertemplate{note page}{preview-big}
{%
  {%
    \scriptsize
    \usebeamerfont{note title}\usebeamercolor[fg]{note title}%
    \ifbeamercolorempty[bg]{note title}{}{%
      \insertvrule{.45\paperheight}{note title.bg}%
      \vskip-.45\paperheight%
      \nointerlineskip%
    }%
    \vbox{
      \hfill\insertslideintonotes{0.45}\hskip-\Gm@rmargin\hskip0pt%
      \vskip-0.45\paperheight%
      \nointerlineskip
      \begin{pgfpicture}{0cm}{0cm}{0cm}{0cm}
        \begin{pgflowlevelscope}{\pgftransformrotate{90}}
          {\pgftransformshift{\pgfpoint{-2cm}{0.2cm}}%
          \pgftext[base,left]{\usebeamerfont{note date}\usebeamercolor[fg]{note c
\ifnum\month<10\relax0\fi\the\month-\ifnum\day<10\relax0\fi\the\day}}
        \end{pgflowlevelscope}
      \end{pgfpicture}}
    \nointerlineskip
    \vbox to .45\paperheight{\vskip0.5em
      \hbox{\insertshorttitle[width=8cm]}%
      \setbox\beamer@tempbox=\hbox{\insertsection}%
      \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip4pt\raise3pt\hbox{\vrule
            width0.4pt height7pt\vrule width 9pt
            height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere{
      }%
      \setbox\beamer@tempbox=\hbox{\insertsubsection}%
      \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip17.4pt\raise3pt\hbox{\vrule
            width0.4pt height7pt\vrule width 9pt
            height0.4pt}}\hskip1pt\hbox{\begin{minipage}[t]{7.5cm}\def\breakhere{
      }%
      \setbox\beamer@tempbox=\hbox{\insertshortframetitle}%
      \hbox{\ifdim\wd\beamer@tempbox>1pt{\hskip30.8pt\raise3pt\hbox{\vrule
            width0.4pt height7pt\vrule width 9pt
            height0.4pt}}\hskip1pt\hbox{\insertshortframetitle[width=7cm]}\fi%
      }%
      \vfil}%
  }%
  \ifbeamercolorempty[bg]{note page}{}{%
    \nointerlineskip%
    \insertvrule{.55\paperheight}{note page.bg}%
    \vskip-.55\paperheight%
  }%
  \vskip.25em
  \nointerlineskip
  \insertnote
}
\defbeamertemplate{note page}{print}
{%
  {%
  \nointerlineskip%
  \begin{beamercolorbox}[%
     wd=\paperwidth,%
     sep=0pt,%
     leftskip=\neo@frametitle@padding,%
     rightskip=\neo@frametitle@padding,%
  ]{note title}%
    \usebeamerfont{frametitle}%
    \neo@frametitlestrut@start%
    \insertframetitle%
    \ifx\insertframesubtitle\@empty
      \usebeamertemplate*{frametitle icon}%
    \else
      \hfill{\usebeamerfont{framesubtitle}\insertframesubtitle}%
    \fi
    \nolinebreak%
    \neo@frametitlestrut@end%
  \end{beamercolorbox}%
  }%
  \insertnote%
  \vfill%
  \begin{beamercolorbox}[wd=\paperwidth, sep=3ex]{footline}%
    \usebeamerfont{page number in head/foot}%
    \usebeamertemplate*{frame footer}
    \hfill%
    \usebeamertemplate*{frame numbering}
  \end{beamercolorbox}%
  \vskip4pt%
}
```

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.

```
\AtBeginDocument{%
  \apptocmd{\appendix}{%
    \pgfkeys{%
      /neo/outer/.cd,
      numbering=none,
      progressbar=none}
    }{}{}
}
```

8.3.5 Process package options

```
\neo@outer@setdefaults
\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

```
\RequirePackage{etoolbox}
\RequirePackage{ifxetex}
\RequirePackage{ifluatex}
\RequirePackage{pgfopts}
```

8.4.2 Load Fira fonts

If the presentation is compiled with XeLaTeX or LuaLaTeX, the fontspec package is loaded and we search for the Fira fonts.

```
\ifboolexpr{bool {xetex} or bool {luatex}}{
  \@ifpackageloaded{fontspec}{
    \PassOptionsToPackage{no-math}{fontspec}
  }{
    \RequirePackage[no-math]{fontspec}
  }

  \IfFontExistsTF{FiraSans-Regular.otf}{
    \defaultfontfeatures{
          Scale     = 1.0,
          Extension = .otf
    }
  }{
    \PackageWarning{beamerthemeneo}{%
      FiraSans font not found in path, therefore using system fonts. %
      Make sure you have the fonts installed.%
    }
  }
  \setmonofont
    [ Numbers = {Monospaced,Lining},
      UprightFont    = *-Regular ,
      ItalicFont     = *-Regular ,
      BoldFont       = *-Medium ,
      BoldItalicFont = *-Medium ,
    ]
    {FiraMono}
  \newcommand{\neo@fontsave}{
    \let\firaneofamily\sfdefault
    \renewcommand*\familydefault{\firaneofamily}
  }
  \newcommand{\neo@fontlight}{
    \setsansfont[
        Numbers = {OldStyle, Monospaced},
        UprightFont    = *-Light ,
        ItalicFont     = *-LightItalic ,
        BoldFont       = *-Regular ,
        BoldItalicFont = *-Italic ,
      ]{FiraSans}
    \neo@fontsave
  }
  \newcommand{\neo@fontbook}{
    \setsansfont[
        Numbers = {OldStyle, Monospaced},
        UprightFont    = *-Book ,
        ItalicFont     = *-BookItalic ,
        BoldFont       = *-Medium ,
        BoldItalicFont = *-MediumItalic ,
      ]{FiraSans}
    \neo@fontsave
  }
  \newcommand{\neo@fontregular}{
    \setsansfont[
        Numbers = {OldStyle, Monospaced},
        UprightFont    = *-Regular ,
        ItalicFont     = *-Italic ,
        BoldFont       = *-SemiBold ,
        BoldItalicFont = *-SemiBoldItalic ,
      ]{FiraSans}
    \neo@fontsave
  }
  \AtBeginEnvironment{tabular}{%
    \addfontfeature{Numbers={Monospaced}}%
  }
}{%
  \RequirePackage[utf8]{inputenc}
  \IfFileExists{FiraSans.sty}{
    \RequirePackage[T1]{fontenc}
    \RequirePackage[sfdefault]{FiraSans}
    \RequirePackage[nomap,lining]{FiraMono}
    \def\bfseries@tt{mb}
    \newcommand{\neo@fontsave}{
      \edef\familydefault{\sfdefault}
      \edef\seriesdefault{\mdseries@sf}
    }
    \newcommand{\neo@fontlight}{
      \def\mdseries@sf{l}
      \def\bfseries@sf{m}
      \neo@fontsave
    }
    \newcommand{\neo@fontbook}{
      \def\mdseries@sf{sl}
      \def\bfseries@sf{medium}
      \neo@fontsave
    }
    \newcommand{\neo@fontregular}{
      \def\mdseries@sf{sl}
      \def\bfseries@sf{sb}
      \neo@fontsave
    }
  }{
    \PackageWarning{beamerthemeneo}{%
      You need to install the Fira Fonts package or compile with XeLaTeX or %
      LuaLaTeX to use the included Fira fonts%
    }
  }
}
```

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

8.4.3 General font definitions

```
\setbeamerfont{title}{size=\Large,%
                      series=\bfseries}
\setbeamerfont{author}{size=\small}
\setbeamerfont{date}{size=\small}
\setbeamerfont{section title}{size=\Large,%
                              series=\bfseries}
\setbeamerfont{block title}{size=\normalsize,%
                            series=\bfseries}
\setbeamerfont{block title alerted}{size=\normalsize,%
                                    series=\bfseries}
\setbeamerfont*{subtitle}{size=\large}
\setbeamerfont{frametitle}{size=\large,%
                           series=\bfseries}
\setbeamerfont{framesubtitle}{parent=frametitle,%
                              size=\footnotesize}
\setbeamerfont{caption}{size=\small}
\setbeamerfont{caption name}{series=\bfseries}
\setbeamerfont{description item}{series=\bfseries}
\setbeamerfont{page number in head/foot}{size=\scriptsize}
\setbeamerfont{bibliography entry author}{size=\normalsize,%
                                          series=\normalfont}
\setbeamerfont{bibliography entry title}{size=\normalsize,%
                                         series=\bfseries}
```

```
\setbeamerfont{bibliography entry location}{size=\normalsize,%
                                            series=\normalfont}
\setbeamerfont{bibliography entry note}{size=\small,%
                                        series=\normalfont}
\setbeamerfont{standout}{size=\Large,%
                         series=\bfseries}
```

8.4.4 Font style options

titleformat title Controls the overall font style.

```
\pgfkeys{
  /neo/font/style/.cd,
    .is choice,
    light/.code={\neo@fontlight},
    book/.code={\neo@fontbook},
    regular/.code={\neo@fontregular},
}
```

8.4.5 Title format options

titleformat title Controls the format of the title.

```
\pgfkeys{
  /neo/font/titleformat title/.cd,
    .is choice,
    regular/.code={%
      \let\neo@titleformat\@empty%
      \setbeamerfont{title}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\neo@titleformat\@empty%
      \setbeamerfont{title}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\neo@titleformat\lowercase%
      \setbeamerfont{title}{shape=\scshape}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat title=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\neo@titleformat\uppercase%
      \setbeamerfont{title}{shape=\normalfont}
      \PackageWarning{beamerthemeneo}{%
```

```
        Be aware that titleformat title=allcaps can lead to problems%
      }
    },
}
```

titleformat subtitle Control the format of the subtitle.

```
\pgfkeys{
  /neo/font/titleformat subtitle/.cd,
    .is choice,
    regular/.code={%
      \let\neo@subtitleformat\@empty%
      \setbeamerfont{subtitle}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\neo@subtitleformat\@empty%
      \setbeamerfont{subtitle}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\neo@subtitleformat\lowercase%
      \setbeamerfont{subtitle}{shape=\scshape}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat subtitle=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\neo@subtitleformat\uppercase%
      \setbeamerfont{subtitle}{shape=\normalfont}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat subtitle=allcaps can lead to problems%
      }
    },
}
```

titleformat section Controls the format of the section title.

```
\pgfkeys{
  /neo/font/titleformat section/.cd,
    .is choice,
    regular/.code={%
      \let\neo@sectiontitleformat\@empty%
      \setbeamerfont{section title}{shape=\normalfont}%
    },
    smallcaps/.code={%
```

```
      \let\neo@sectiontitleformat\@empty%
      \setbeamerfont{section title}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\neo@sectiontitleformat\MakeLowercase%
      \setbeamerfont{section title}{shape=\scshape}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat section=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\neo@sectiontitleformat\MakeUppercase%
      \setbeamerfont{section title}{shape=\normalfont}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat section=allcaps can lead to problems%
      }
    },
}
```

`frametitleformat` Control the format of the frame title.

```
\pgfkeys{
  /neo/font/titleformat frame/.cd,
    .is choice,
    regular/.code={%
      \let\neo@frametitleformat\@empty%
      \setbeamerfont{frametitle}{shape=\normalfont}%
    },
    smallcaps/.code={%
      \let\neo@frametitleformat\@empty%
      \setbeamerfont{frametitle}{shape=\scshape}%
    },
    allsmallcaps/.code={%
      \let\neo@frametitleformat\MakeLowercase%
      \setbeamerfont{frametitle}{shape=\scshape}%
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat frame=allsmallcaps can lead to problems%
      }
    },
    allcaps/.code={%
      \let\neo@frametitleformat\MakeUppercase%
      \setbeamerfont{frametitle}{shape=\normalfont}
      \PackageWarning{beamerthemeneo}{%
        Be aware that titleformat frame=allcaps can lead to problems%
      }
    },
}
```

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

```
\pgfkeys{
  /neo/font/.cd,
  titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
  titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
  titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
  titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
}
```

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

```
\newcommand{\neo@font@setdefaults}{
  \pgfkeys{/neo/font/.cd,
    style=book,
    titleformat title=regular,
    titleformat subtitle=regular,
    titleformat section=regular,
    titleformat frame=regular,
  }
}
```

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

```
\def\neo@titleformat#1{#1}
\def\neo@subtitleformat#1{#1}
\def\neo@sectiontitleformat#1{#1}
\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.

```
\patchcmd{\beamer@title}%
  {\def\inserttitle{#2}}%
  {\def\inserttitle{\neo@titleformat{#2}}}%
  {}%
  {\PackageError{beamerfontthemeneo}{Patching title failed}\@ehc}
\patchcmd{\beamer@subtitle}%
```

```
  {\def\insertsubtitle{#2}}%
  {\def\insertsubtitle{\neo@subtitleformat{#2}}}%
  {}%
  {\PackageError{beamerfontthemeneo}{Patching subtitle failed}\@ehc}
\patchcmd{\sectionentry}
  {\def\insertsectionhead{#2}}
  {\def\insertsectionhead{\neo@sectiontitleformat{#2}}}
  {}
  {\PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc}
\@tempswafalse
\patchcmd{\beamer@section}
  {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded
  {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
    \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
  {\@tempswatrue}
  {}
\patchcmd{\beamer@section}
  {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
  {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
    \neo@sectiontitleformat{#1}}}}
  {\@tempswatrue}
  {}
\patchcmd{\beamer@section}
  {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpanded
  {\edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
    \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
  {\@tempswatrue}
  {}
\patchcmd{\beamer@section}
  {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{#
  {\protected@edef\insertsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
    \noexpand\neo@sectiontitleformat{#1}}}}
  {\@tempswatrue}
  {}
\if@tempswa\else
  \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
\fi
\@tempswafalse
\patchcmd{\beamer@subsection}
  {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{\unexpan
  {\edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page}{%
    \noexpand\neo@sectiontitleformat{\unexpanded{#1}}}}}
  {\@tempswatrue}
  {}
\patchcmd{\beamer@subsection}
  {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
  {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{%
    \neo@sectiontitleformat{#1}}}}
  {\@tempswatrue}
  {}
\patchcmd{\beamer@subsection}
  {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page
  {\protected@edef\insertsubsectionhead{\noexpand\hyperlink{Navigation\the\c@page
    \noexpand\neo@sectiontitleformat{#1}}}}
  {\@tempswatrue}
  {}
\if@tempswa\else
  \PackageError{beamerfontthemeneo}{Patching section title failed}\@ehc
\fi
```

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

```
\patchcmd{\beamer@@frametitle}
  {{%
      \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
      \usebeamertemplate*{frametitle continuation}\fi}}%
    \gdef\beamer@frametitle{#2}%
    \gdef\beamer@shortframetitle{#1}%
    }}
  {{%
      \gdef\insertframetitle{{\neo@frametitleformat{#2}\ifnum%
      \beamer@autobreakcount>0\relax{}\space%
      \usebeamertemplate*{frametitle continuation}\fi}}%
    \gdef\beamer@frametitle{#2}%
    \gdef\beamer@shortframetitle{#1}%
    }}
  {}
  {\PackageError{beamerfontthemeneo}{Patching frame title failed}\@ehc}
```

8.4.6 Process package options

```
\neo@font@setdefaults
\ProcessPgfPackageOptions{/neo/font}
```

8.5 NEO color theme

8.5.1 Package dependencies

```
\RequirePackage{pgfopts}
```

8.5.2 Options

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

```
\pgfkeys{
  /neo/color/block/.cd,
    .is choice,
    transparent/.code=\neo@block@transparent,
    fill/.code=\neo@block@fill,
}
```

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

```
\pgfkeys{
  /neo/color/background/.cd,
    .is choice,
    dark/.code=\neo@colors@dark,
    light/.code=\neo@colors@light,
    white/.code=\neo@colors@white,
}
```

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

```
\newcommand{\neo@color@setdefaults}{
  \pgfkeys{/neo/color/.cd,
    background=light,
    block=transparent,
  }
}
```

8.5.3 Base colors

```

\definecolor{nDarkGrey}{RGB}{152,164,174}
\definecolor{nGrey}{RGB}{210,213,215}
\definecolor{nLightGrey}{RGB}{235,236,238}

\definecolor{nDarkRed}{RGB}{141,20,41}
```

```
\definecolor{nRed}{RGB}{201,169,147}
\definecolor{nLightRed}{RGB}{237,231,222}

\definecolor{nDarkGreen}{RGB}{0,155,119}
\definecolor{nGreen}{RGB}{170,207,189}
\definecolor{nLightGreen}{RGB}{229,239,234}

\definecolor{nDarkBlue}{RGB}{0,56,101}
\definecolor{nBlue}{RGB}{144,167,198}
\definecolor{nLightBlue}{RGB}{221,229,240}

\definecolor{nDarkCyan}{RGB}{0,177,235}
\definecolor{nCyan}{RGB}{180,214,245}
\definecolor{nLightCyan}{RGB}{234,243,252}

\definecolor{nDarkYellow}{RGB}{201,147,19}
\definecolor{nYellow}{RGB}{217,198,137}
\definecolor{nLightYellow}{RGB}{243,238,223}

\definecolor{nBlack}{HTML}{011F32}
\definecolor{nWhite}{RGB}{250,250,250}
```

8.5.4 Alias colors

Support the colors provided by the old i4 beamer theme.

```
\colorlet{i4red}{nDarkRed}
\colorlet{i4green}{nDarkGreen}
\colorlet{i4blue}{nDarkBlue}
\colorlet{i4cyan}{nDarkCyan}
\colorlet{i4yellow}{nDarkYellow}
\colorlet{i4grey}{nDarkGrey}
\definecolor{darkred}{rgb}{0.8,0,0}
\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`.

```
\newcommand{\neo@colors@dark}{
  \setbeamercolor{normal text}{%
    fg=nWhite,
    bg=nBlack
  }
  \setbeamercolor{normal item}{%
    fg=nWhite,
    bg=nDarkBlue
  }
  \usebeamercolor[fg]{normal text}
}
\newcommand{\neo@colors@light}{
  \setbeamercolor{normal text}{%
    fg=nBlack,
    bg=nWhite
  }
  \setbeamercolor{normal item}{%
    fg=nDarkBlue,
    bg=nWhite
  }
}
\newcommand{\neo@colors@white}{
  \definecolor{nWhite}{RGB}{255,255,255}
  \neo@colors@light
}
\setbeamercolor{alerted text}{%
  fg=nDarkRed
}
\setbeamercolor{example text}{%
  fg=nDarkYellow
}
\setbeamercolor{note title}{%
  fg=nDarkBlue,
  bg=nGrey
}
\setbeamercolor{note page}{%
  fg=nBlack,
  bg=nLightGrey
}
```

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 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`.

```
\setbeamercolor{titlelike}{use=normal text, parent=normal text}
\setbeamercolor{author}{use=normal text, parent=normal text}
\setbeamercolor{date}{use=normal text, parent=normal text}
\setbeamercolor{institute}{use=normal text, parent=normal text}
\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.

```
\setbeamercolor{palette primary}{%
  use=normal text,
  fg=normal text.bg,
  bg=nDarkBlue
}
\setbeamercolor{frametitle}{%
  use=palette primary,
  parent=palette primary
}
```

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`.

```
\setbeamercolor{progress bar}{%
  use=normal text,
  fg=nDarkBlue,
  bg=nLightBlue
}
\setbeamercolor{title separator}{
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar in head/foot}{%
  use=normal text.fg,
  fg=nBlack,
  parent=progress bar
}
\setbeamercolor{progress bar in section page}{
  use=progress bar,
  parent=progress bar
}
```

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.

```
\newcommand{\neo@block@transparent}{
  \setbeamercolor{block title}{%
    use=normal text,
    fg=nDarkBlue,
    bg=
  }
  \setbeamercolor{block title alerted}{%
    use={block title, alerted text},
    bg=block title.bg,
    fg=alerted text.fg
  }
  \setbeamercolor{block title example}{%
    use={block title, example text},
    bg=block title.bg,
    fg=example text.fg
  }
  \setbeamercolor{block body}{
    bg=
  }
  \setbeamercolor{block body alerted}{
    use=block body,
    parent=block body,
    bg=
  }
  \setbeamercolor{block body example}{
    use=block body,
    parent=block body,
    bg=
  }
}
\newcommand{\neo@block@fill}{
  \setbeamercolor{block title}{%
    use=normal text,
    fg=nDarkBlue,
    bg=nGrey
  }
  \setbeamercolor{block title alerted}{%
    use={block title, alerted text},
    bg=alerted text.fg,
    fg=alerted text.bg
  }
  \setbeamercolor{block title example}{%
    use={block title, example text},
    bg=example text.fg,
    fg=example text.bg
  }
  \setbeamercolor{block body}{
    use={block title, normal text},
    bg=nLightGrey
  }
  \setbeamercolor{block body alerted}{
    use=block body,
    parent=block body,
    bg=nRed!50,
  }
  \setbeamercolor{block body example}{
    use=block body,
    parent=block body,
    bg=nYellow!50
  }
}

```

Footnotes

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

```
\setbeamercolor{bibliography entry author}{fg=, bg=}
\setbeamercolor{bibliography entry title}{fg=, bg=}
\setbeamercolor{bibliography entry location}{fg=, bg=}
\setbeamercolor{bibliography entry note}{fg=, bg=}
```

8.5.7 Process package options

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

```
\mode<all>
```

8.6 Tol `pgfplots` theme

Paul Tol's 12-color palette[1] is as follows:

```
\definecolor{TolDarkPurple}{HTML}{332288}
\definecolor{TolDarkBlue}{HTML}{6699CC}
\definecolor{TolLightBlue}{HTML}{88CCEE}
\definecolor{TolLightGreen}{HTML}{44AA99}
\definecolor{TolDarkGreen}{HTML}{117733}
\definecolor{TolDarkBrown}{HTML}{999933}
\definecolor{TolLightBrown}{HTML}{DDCC77}
\definecolor{TolDarkRed}{HTML}{661100}
\definecolor{TolLightRed}{HTML}{CC6677}
\definecolor{TolLightPink}{HTML}{AA4466}
\definecolor{TolDarkPink}{HTML}{882255}
\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.

```
\pgfplotscreateplotcyclelist{mbarplot cycle}{%
  {draw=TolDarkBlue,    fill=TolDarkBlue!70},
  {draw=TolLightBrown,  fill=TolLightBrown!70},
  {draw=TolLightGreen,  fill=TolLightGreen!70},
  {draw=TolDarkPink,    fill=TolDarkPink!70},
  {draw=TolDarkPurple,  fill=TolDarkPurple!70},
  {draw=TolDarkRed,     fill=TolDarkRed!70},
  {draw=TolDarkBrown,   fill=TolDarkBrown!70},
  {draw=TolLightRed,    fill=TolLightRed!70},
  {draw=TolLightPink,   fill=TolLightPink!70},
  {draw=TolLightPurple, fill=TolLightPurple!70},
  {draw=TolLightBlue,   fill=TolLightBlue!70},
  {draw=TolDarkGreen,   fill=TolDarkGreen!70},
}
```

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

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

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

```
  {TolDarkBrown, mark=diamond*, mark size=1.5pt},
}
```

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.

```
\pgfplotsset{
  compat=1.9,
```

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

```
  mlineplot/.style={
    mbaseplot,
    xmajorgrids=true,
    ymajorgrids=true,
    major grid style={dotted},
    axis x line=bottom,
    axis y line=left,
    legend style={
      cells={anchor=west},
      draw=none
    },
    cycle list name=mlineplot cycle,
  },
```

mbarplot
horizontal 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.

```
  mbarplot base/.style={
    mbaseplot,
    bar width=6pt,
    axis y line*=none,
  },
  mbarplot/.style={
    mbarplot base,
    ybar,
    xmajorgrids=false,
    ymajorgrids=true,
    area legend,
    legend image code/.code={%
```

```
      \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
    },
    cycle list name=mbarplot cycle,
  },
  horizontal mbarplot/.style={
    mbarplot base,
    xmajorgrids=true,
    ymajorgrids=false,
    xbar stacked,
    area legend,
    legend image code/.code={%
      \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
    },
    cycle list name=mbarplot cycle,
  },
```

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

```
  mbaseplot/.style={
    legend style={
      draw=none,
      fill=none,
      cells={anchor=west},
    },
    x tick label style={
      font=\footnotesize
    },
    y tick label style={
      font=\footnotesize
    },
    legend style={
      font=\footnotesize
    },
    major grid style={
      dotted,
    },
    axis x line*=bottom,
  },
  disable thousands separator/.style={
    /pgf/number format/.cd,
      1000 sep={}
  },
}
```