Moloch

A Minimalist, Feature-Rich Beamer Theme

Johan Larsson Matthias Vogelgesang

19 October 2026

Table of contents

1 Moloch **9**
Introduction . . . 9
A First Example . . . 10

I User Guide **11**

2 Getting Started **12**
2.1 Installing from CTAN . . . 12
2.2 Installing from Source . . . 12
2.3 A Minimal Example . . . 12
2.4 Dependencies . . . 14
2.5 Pandoc . . . 14

3 Customization **16**
3.1 Main Theme . . . 16
3.2 Inner Theme . . . 17
3.3 Outer Theme . . . 22
3.4 Color Theme . . . 25
3.4.1 Color Options . . . 25
3.4.2 Color Themes . . . 28
3.4.3 Color Customization . . . 38
3.4.4 Granular Color Customization with `\molochcolors` . . . 39
3.5 Font Theme . . . 43

4 Commands **44**
4.1 Standout Frames . . . 44

5 ltx-talk **46**
5.1 Requirements . . . 46
5.2 Usage . . . 46
5.3 Options . . . 48
5.4 Customization beyond these options . . . 49
5.4.1 Footer . . . 49
5.5 Differences from the Beamer theme . . . 49

6 Known Issues **51**
6.1 Title Formats . . . 51
6.2 Interactions with Other Color Themes . . . 51
6.3 Notes on Second Screen . . . 52
6.4 Standout Frames with Labels . . . 52
6.5 Standout Frames with Pandoc . . . 53

7 License **54**

8 Changelog **55**
8.1 2.3.1 (2026-10-19) . . . 55
8.1.1 Bug Fixes . . . 55
8.2 2.3.0 (2026-10-15) . . . 55
8.2.1 Highlights . . . 55
8.2.2 Features . . . 55
8.2.3 Bug Fixes . . . 55
8.3 2.2.0 (2026-02-17) . . . 56
8.3.1 Features . . . 56
8.3.2 Bug Fixes . . . 56
8.4 2.1.0 (2025-12-18) . . . 56
8.4.1 Features . . . 56
8.4.2 Bug Fixes . . . 56
8.5 2.0.0 (2025-12-05) . . . 56
8.5.1 BREAKING CHANGES . . . 56
8.5.2 Features . . . 56
8.5.3 Bug Fixes . . . 57
8.6 1.1.0 (2025-09-25) . . . 57
8.6.1 Features . . . 57
8.6.2 Bug Fixes . . . 57
8.7 1.0.1 (2025-04-01) . . . 57
8.7.1 Bug Fixes . . . 57
8.8 1.0.0 (2025-04-01) . . . 57
8.8.1 Features . . . 57
8.9 0.6.0 (2025-01-17) . . . 58
8.9.1 Features . . . 58
8.9.2 Reverts . . . 58
8.10 0.5.0 (2024-10-30) . . . 58
8.10.1 Features . . . 58
8.10.2 Bug Fixes . . . 58
8.11 0.4.0 (2024-07-09) . . . 58
8.11.1 Features . . . 58
8.11.2 Bug Fixes . . . 59

8.12 0.3.0 (2024-03-06) . . . 59
8.12.1 Features . . . 59
8.12.2 Bug Fixes . . . 59
8.13 0.2.0 (2024-01-23) . . . 59
8.13.1 BREAKING CHANGES . . . 59
8.13.2 Features . . . 59
8.14 0.1.0 (2024-01-15) . . . 60
8.14.1 BREAKING CHANGES . . . 60
8.14.2 Features . . . 60
8.14.3 Bug Fixes . . . 60
8.14.4 Reverts . . . 60
8.14.5 Miscellaneous Chores . . . 61

Appendices **62**

II Implementation **62**

A Main Theme **63**
A.1 Moloch Parent Theme . . . 63
A.2 Package Dependencies . . . 63
A.3 Options . . . 63
A.3.1 `titleformat plain` . . . 64
A.3.2 `titleformat` . . . 64
A.4 Component Sub-Packages . . . 65
A.5 Custom Commands . . . 65
A.5.1 `\molochset` . . . 65
A.5.2 `\mreducelistspacing` . . . 65
A.6 Process Package Options . . . 66

B Inner Theme **67**
B.1 Package Dependencies . . . 67
B.2 Memoization and Tikz Externalization . . . 67
B.3 Options . . . 67
B.3.1 `sectionpage` . . . 67
B.3.2 `subsectionpage` . . . 68
B.3.3 `standoutnumbering` . . . 68
B.3.4 `sectionpagenumbering` . . . 69
B.3.5 `titleseparator linewidth` . . . 69
B.3.6 `titleseparator aliases` . . . 70
B.3.7 `titlepage` . . . 70
B.3.8 `\moloch@inner@setdefaults` . . . 71

B.4 Title Page . . . 71
B.4.1 title page . . . 71
B.4.2 Moloch Variant . . . 71
B.4.3 Plain (Beamer-like) Variant . . . 72
B.4.4 Split (Two-column) Variant . . . 72
B.4.5 \maketitle . . . 73
B.4.6 \titlepage . . . 74
B.4.7 title graphic . . . 74
B.4.8 title . . . 74
B.4.9 subtitle . . . 75
B.4.10 title separator . . . 75
B.4.11 title separator vertical . . . 76
B.4.12 author . . . 76
B.4.13 institute . . . 77
B.4.14 date . . . 77
B.5 Section Pages . . . 78
B.5.1 section Page . . . 78
B.5.2 subsection page . . . 79
B.5.3 progress bar in section page . . . 79
B.6 Lists and Floats . . . 80
B.7 Footnotes . . . 82
B.8 Text and Spacing Settings . . . 82
B.9 Standout Frames . . . 82
B.9.1 standout . . . 82
B.10 Process Package Options . . . 84

C Outer Theme **85**
C.1 Moloch Outer Theme . . . 85
C.2 Package Dependencies . . . 85
C.3 Memoization and Tikz Externalization . . . 85
C.4 Options . . . 86
C.4.1 progressbar . . . 86
C.4.2 progressbar linewidth . . . 86
C.4.3 progressbar aliases . . . 87
C.4.4 frametitle margin . . . 87
C.4.5 frametitle margin aliases . . . 87
C.4.6 \moloch@outer@setdefaults . . . 87
C.5 Deprecated Options . . . 88
C.5.1 numbering . . . 88
C.6 Slide Numbering . . . 88
C.7 Head and Footline . . . 89
C.7.1 headline . . . 89
C.7.2 footline . . . 89

C.8 Frametitle . . . 89
C.8.1 `frametitle` . . . 89
C.8.2 `progress bar in head/foot` . . . 91
C.9 Process Package Options . . . 91

D Font Theme **92**
D.1 Moloch Font Theme . . . 92
D.2 Package Dependencies . . . 92
D.3 General Font Definitions . . . 92
D.4 Title Format Options . . . 93
D.4.1 `titleformat title` . . . 93
D.4.2 `titleformat subtitle` . . . 93
D.4.3 `titleformat section` . . . 94
D.4.4 `frametitleformat` . . . 95
D.4.5 `titleformat aliases` . . . 96
D.4.6 `\moloch@font@setdefaults` . . . 96
D.5 Process Package Options . . . 96

E Color Theme **97**
E.1 Moloch Color Theme . . . 97
E.1.1 Package Dependencies . . . 97
E.1.2 Options . . . 97
E.1.3 `block` . . . 97
E.1.4 `Variant tracking` . . . 97
E.1.5 `colors` . . . 98
E.1.6 `\moloch@color@setdefaults` . . . 98
E.1.7 Base Colors . . . 98
E.1.8 Hooks for Color Themes . . . 99
E.1.9 `\moloch@define@light@colors` . . . 99
E.1.10 `\moloch@define@dark@colors` . . . 99
E.1.11 `\moloch@register@light@colors` . . . 99
E.1.12 `\moloch@register@dark@colors` . . . 99
E.1.13 `\moloch@setup@block@colors` . . . 99
E.1.14 `\moloch@setup@text@colors` . . . 100
E.1.15 `\moloch@setup@progressbar@parents` . . . 100
E.1.16 Variant Switching Commands . . . 101
E.1.17 `\moloch@colors@dark` . . . 101
E.1.18 `\moloch@colors@light` . . . 101
E.1.19 `\moloch@apply@current@variant` . . . 102
E.1.20 Color Customization Storage System . . . 102
E.1.21 `\moloch@apply@store@light@colors` . . . 103
E.1.22 `\moloch@apply@store@dark@colors` . . . 104
E.1.23 Semantic Color Helper . . . 106

E.1.24 \moloch@apply@semantic@colors . . . 106
E.1.25 Color Theme Presets . . . 107
E.1.26 \moloch@colortheme@default . . . 107
E.1.27 \moloch@colortheme@tomorrow . . . 108
E.1.28 \moloch@colortheme@paper . . . 109
E.1.29 \moloch@colortheme@highcontrast . . . 110
E.1.30 \moloch@colortheme@catppuccin . . . 112
E.1.31 New Color System . . . 113
E.1.32 Color theme and variant options . . . 113
E.1.33 \molochcolors . . . 123
E.1.34 Process Package Options . . . 124

F Color Theme: Tomorrow **125**
F.1 Moloch Tomorrow Color Theme . . . 125

G Color Theme: High Contrast **126**

H ltx-talk Theme **127**
H.1 Moloch for ltx-talk . . . 127
H.2 Class Guard . . . 127
H.3 Colors . . . 128
H.3.1 __moloch_colors_light: . . . 128
H.3.2 __moloch_colors_dark: . . . 128
H.3.3 __moloch_colors_apply: . . . 129
H.4 Options . . . 129
H.4.1 \molochset . . . 130
H.5 Frame Title Band . . . 132
H.6 Rules and the Progress Bar . . . 133
H.6.1 __moloch_rule:nnn . . . 133
H.6.2 __moloch_progress_frac: . . . 134
H.6.3 __moloch_progress_width:n . . . 134
H.6.4 __moloch_progressbar:nn . . . 135
H.6.5 __moloch_progressbar_head: . . . 135
H.6.6 __moloch_progressbar_frametitle: . . . 135
H.6.7 __moloch_progressbar_foot: . . . 135
H.6.8 __moloch_progress_rule:n . . . 136
H.7 Page Style . . . 137
H.8 Lists . . . 139
H.9 Captions . . . 141
H.10 Title Page . . . 141
H.10.1 __moloch_titlepage:n . . . 142
H.10.2 \maketitle . . . 142
H.11 Section Pages . . . 143

H.12 Standout Frames 144
H.12.1 `standout` 145
H.13 Page Background and Text Color 145

1 Moloch

> **Note**
>
> You are reading the **local** documentation channel. Use the version switcher in the sidebar to jump between `latest`, `dev`, and TeX Live-compatible snapshots.

Introduction

Beamer is a great 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 Moloch is to provide a simple, modern Beamer theme suitable for anyone to use. 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.

Moloch is maintained at https://github.com/jolars/moloch. If you have any issues, find mistakes in the manual or want to help make the theme even better, please get in touch there.

Moloch is a fork of the excellent Metropolis theme by Matthias Vogelgesang. The motivation for the fork was to fix some longstanding bugs in Metropolis and simplify the codebase to make it easier to maintain and less fragile to changes in the underlying Beamer code. Since then, Moloch has grown to include several new features, such as new color themes, as well as additional customization options.

A First Example

Here's a small demo of what Moloch looks like:

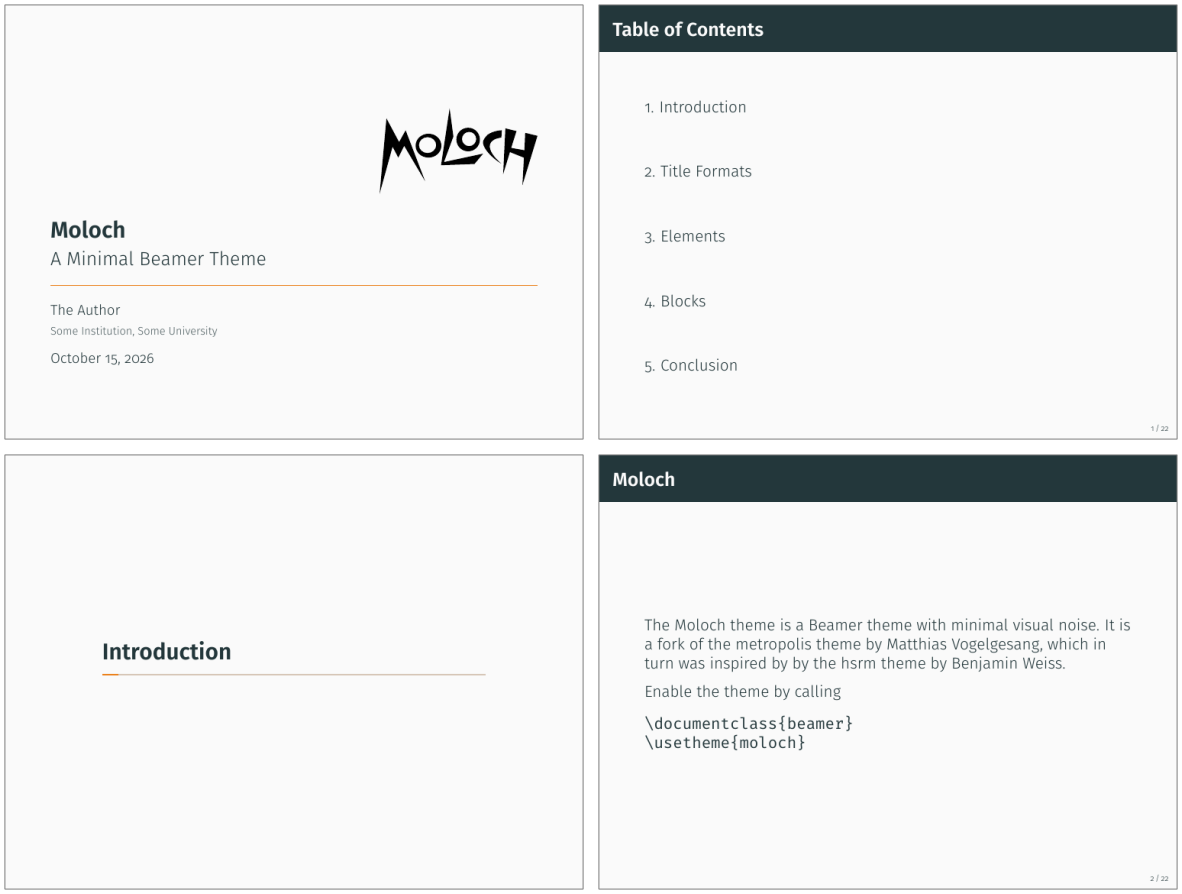

Figure 1.1: An example of the Moloch theme.

Part I

User Guide

2 Getting Started

2.1 Installing from CTAN

For most users, we recommend installing Moloch from CTAN. If you keep your TeX distribution up-to-date, chances are good that Moloch is already installed. If it is not, you need to update your packages. If your distribution is TeX Live (or MacTeX on OS X), the following command updates all packages.

```
tlmgr update --all
```

If this results in an error, you may need to run it with administrative privileges:

```
sudo tlmgr update --all
```

MacTeX on OS X also provides a graphical interface for `tlmgr` called TeX Live Utility.

For other distributions, please refer to its documentation on how to update your packages.

2.2 Installing from Source

If you want to use the development version of Moloch, you can install it manually. You need a recent LaTeX distribution which includes **l3build**. Then simply follow the steps below.

Download the source with a `git clone` of https://github.com/jolars/moloch:

```
git clone https://github.com/jolars/moloch.git
```

Then navigate into the downloaded directory and install the package by running:

```
cd moloch
l3build install
```

2.3 A Minimal Example

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

```latex
\documentclass{beamer}
\usetheme{moloch}

\title{A Minimal Example}

\date{\today}
\author{Johan Larsson}
\institute{Some University}

\begin{document}
\maketitle

\section{First Section}

\begin{frame}
  \frametitle{First Frame}

  Hello, world!

\end{frame}
\end{document}
```

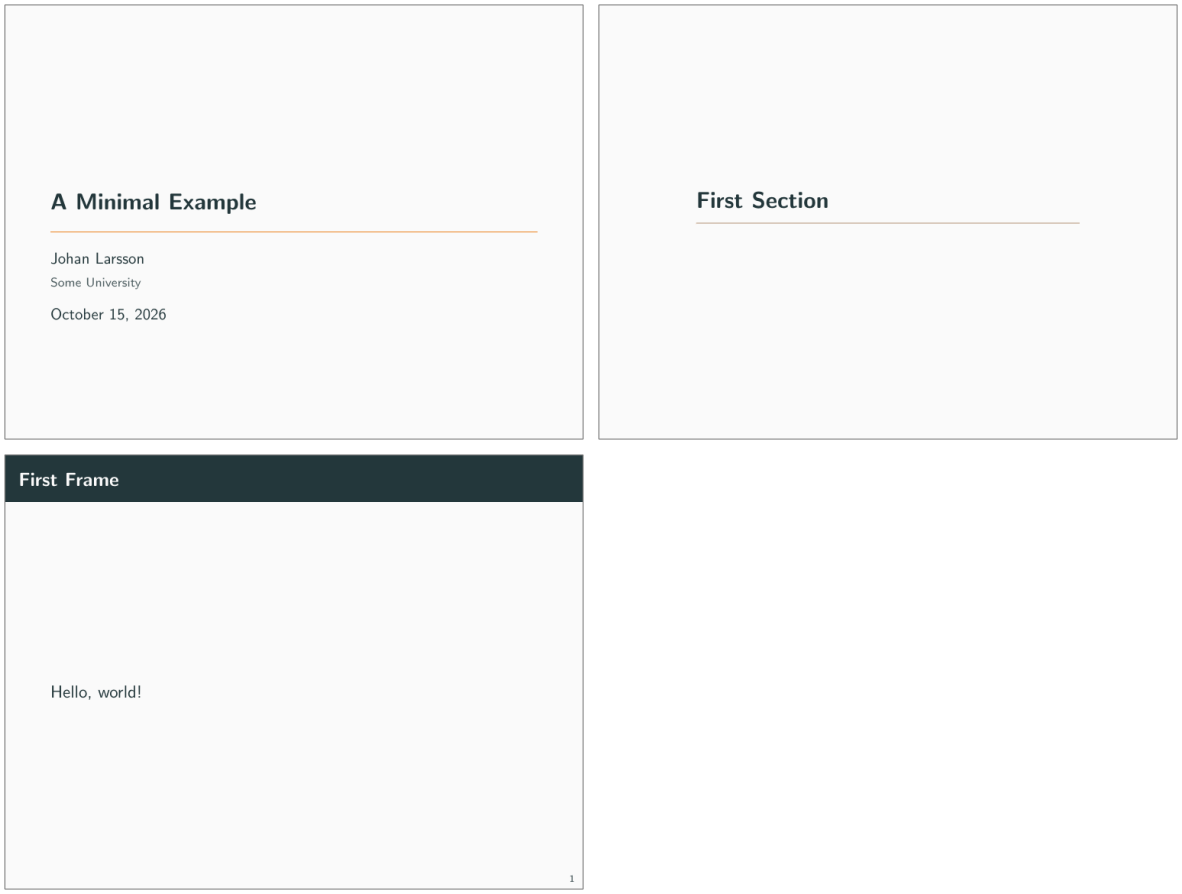

Figure 2.1: A minimal example of a Beamer presentation using Moloch.

2.4 Dependencies

Moloch depends on the `beamer` class and the following standard packages:

- tikz
- pgfopts
- etoolbox
- calc

2.5 Pandoc

To use this theme with Pandoc-based presentations, you can run the following command in your terminal:

```
pandoc -t beamer -V theme:moloch -o output.pdf input.md
```

3 Customization

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 Moloch in the preamble:

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

Options can be changed at any time—even mid-presentation—with the `molochset()` macro.

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

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

`some option` ***default value, another value, yet another value*** ..... default value

A short description of the option.

3.1 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. Note that these commands do not affect math and numbers, so may not work as you expect if your titles contain these.

`standoutnumberformat` ***regular, smallcaps, allsmallcaps, allcaps*** ..... regular

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

3.2 Inner Theme

These options control the inner theme, which is responsible for elements like section and subsection pages, as well as the title page layout.

`titlepage` ***moloch, plain, split*** ..... moloch

Controls the layout of the title page:

- `moloch`: Moloch's custom layout with golden ratio vertical spacing and a title separator line
- `plain`: Beamer-like layout with standard vertical spacing and titlegraphic at the bottom
- `split`: Two-column asymmetric layout with title/subtitle on the left and author/institute/date on the right

```latex
\documentclass{beamer}
\usetheme{moloch}

\title{Moloch Theme}
\subtitle{A Minimalist Beamer Theme}
\author{John Doe}
\institute{University of Example}
\date{\today}
\titlegraphic{\includegraphics[width=2cm]{assets/logo.pdf}}

\begin{document}

\maketitle

\end{document}
```

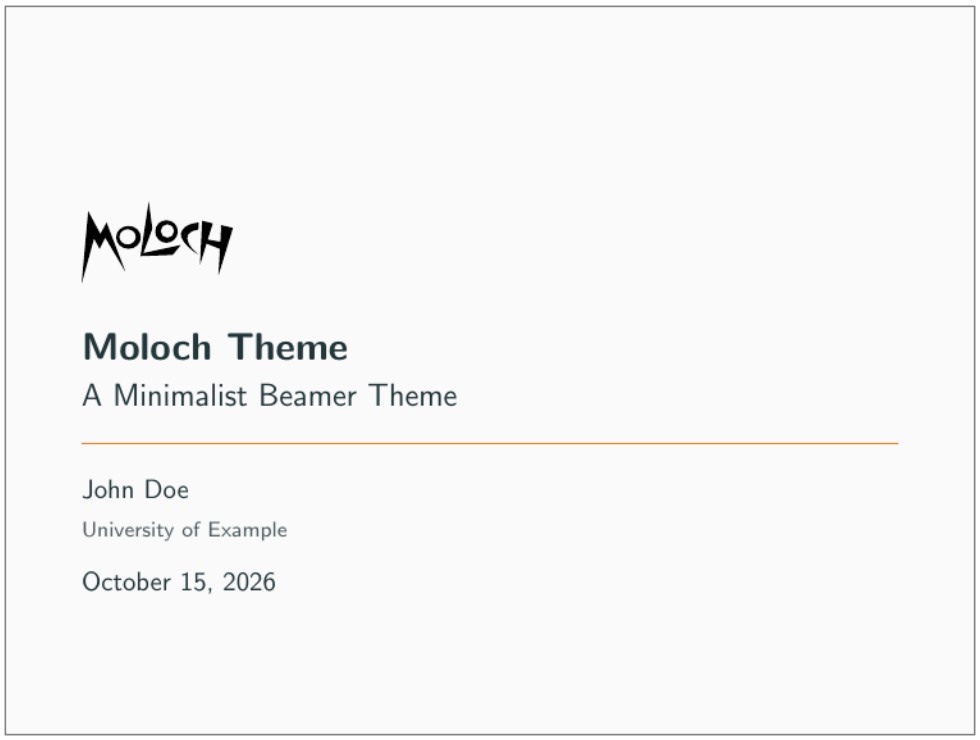

Figure 3.1: Moloch titlepage layout (default).

Plain Beamer-like layout:

```
\usetheme[titlepage=plain]{moloch}
```

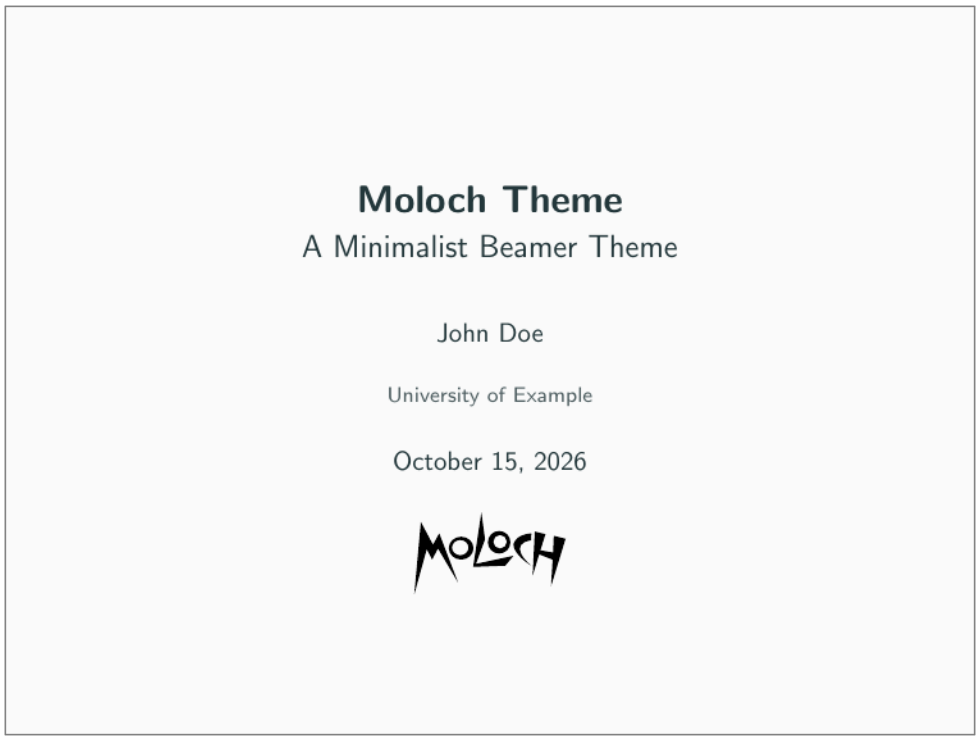

Figure 3.2: Plain titlepage layout.

Split two-column layout:

```
\usetheme[titlepage=split]{moloch}
```

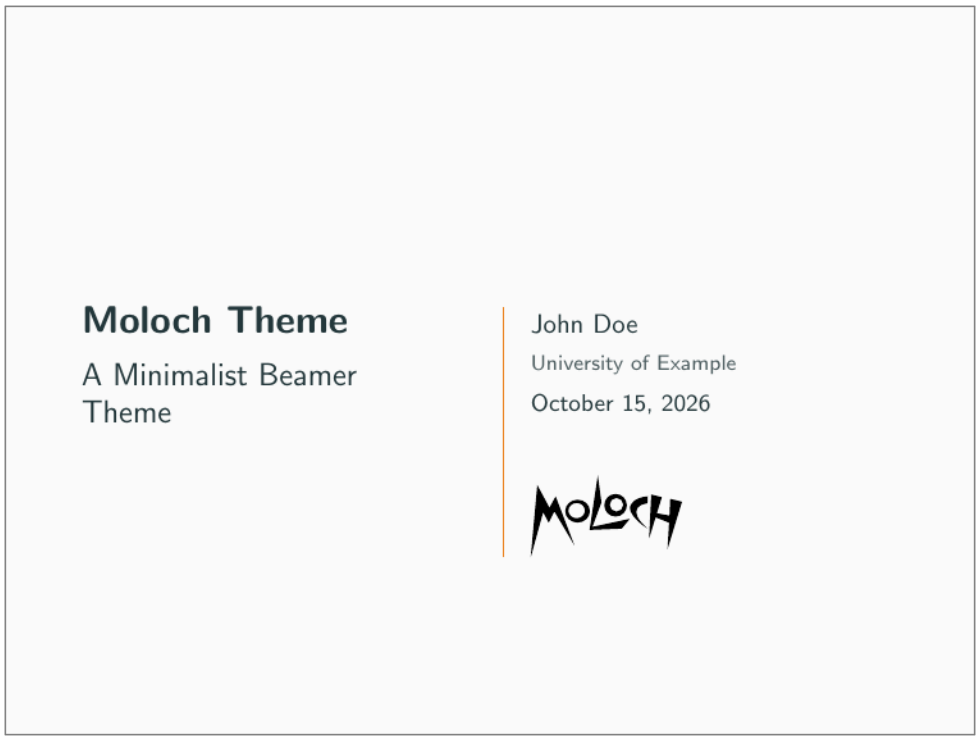

Figure 3.3: Split two-column titlepage layout.

`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. Section pages display the full title even when you provide a short title with `\section[Short title]{Full title}`.

```
\documentclass{beamer}
\usetheme{moloch}

\begin{document}

\molochset{sectionpage=progressbar} % the default

\section{Section One}

\molochset{sectionpage=simple}

\section{Section Two}
```

```latex
\molochset{sectionpage=none}

\section{Section Three} % Will not have a section page

\end{document}
```

Figure 3.4: Example of the `sectionpage` options.

`subsectionpage` ***none, simple, progressbar*** ..... none

Optionally adds a slide at the start of each subsection, displaying the full section and subsection titles. 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.

`standoutnumbering` ***none, hide, show*** ..... none

This option decides whether or not to count standout pages as frames if frame counting. Option `none` (the default) means that the standout frames are not counted. `hide` means that they are counted but that there won't be any footer showing a frame number. `show` means that they are counted and that the frame number count is shown in the same fashion as for regular frames.

`titleseparator linewidth` ***<dimension>*** ..... default: 0.5pt

Controls the width of the horizontal line that separates the title from the author information on the title page (when using `titlepage=moloch`). Set to `0pt` to hide the separator line entirely.

```latex
\usetheme[titleseparator linewidth=2pt]{moloch}
```

3.3 Outer Theme

These options control the outer theme, which is responsible for elements like the frame numbering and progress bar.

`numbering` ***none, counter, fraction*** ..... (none specified)

> ⚠ Deprecated
>
> This option is deprecated and will be removed in a future version. Please use Beamer's `page number in head/foot` template instead.

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`).

```latex
\documentclass{beamer}

\usetheme{moloch}

\begin{document}

\molochset{progressbar=none} % the default

\begin{frame}
  \frametitle{Progress Bar Example}

  The default is to have no progress bar.
\end{frame}

\molochset{progressbar=head}

\begin{frame}
  \frametitle{Progress Bar Example}

  The progress bar is now displayed at the top of the frame.
\end{frame}

\molochset{progressbar=foot}
```

```latex
\begin{frame}
  \frametitle{Progress Bar Example}

  The progress bar is now displayed at the bottom of the frame.
\end{frame}

\molochset{progressbar=frametitle}

\begin{frame}
  \frametitle{Progress Bar Example}

  The progress bar is now displayed in the frametitle area.
\end{frame}

\end{document}
```

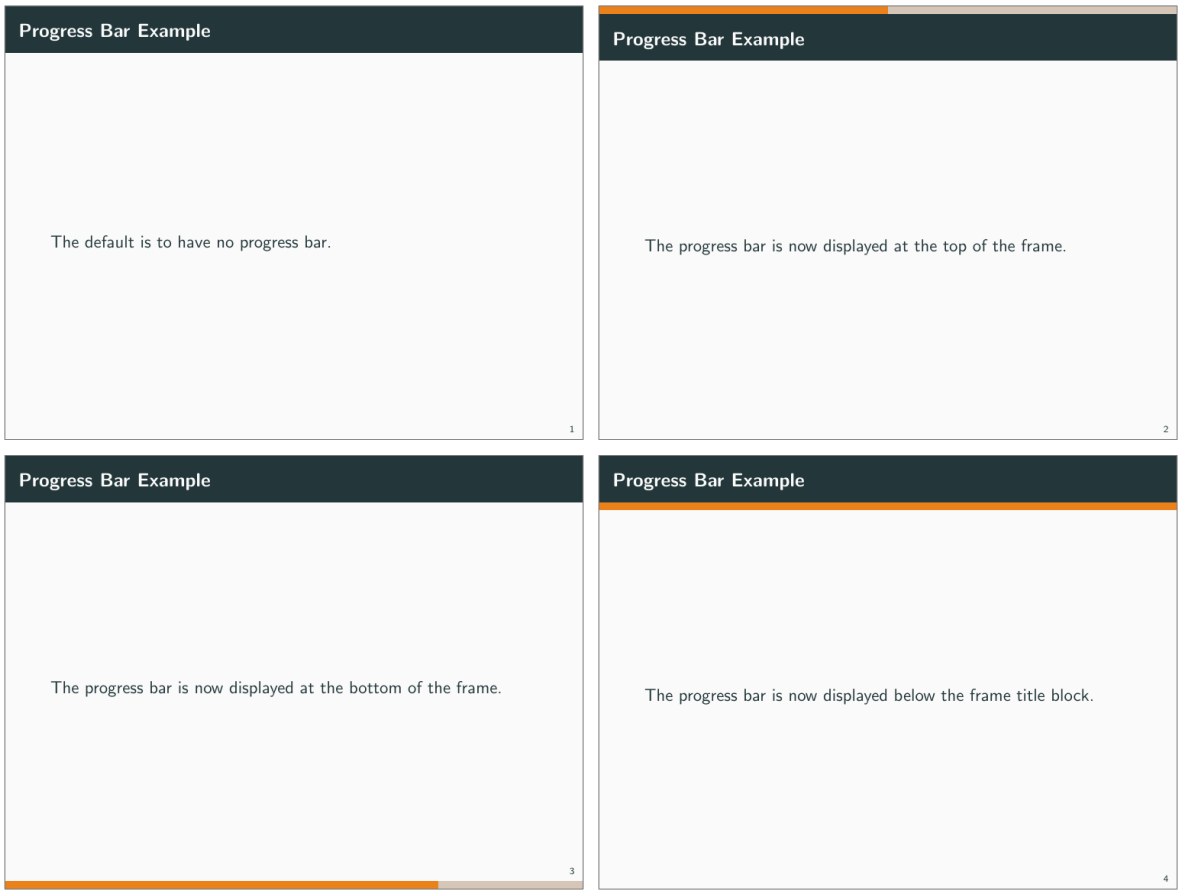

Figure 3.5: Example of the `progressbar` options.

`frametitle margin top/bottom/left/right` ***numeric*** 1.4ex/1.4ex/1.6ex/1.6ex

Controls the margins around the frame title. The default is `1.4ex` for the top and bottom margins, and `1.6ex` for the left and right margins.

```
\documentclass{beamer}
\usetheme{moloch}

\begin{document}

\begin{frame}
  \frametitle{This is a frame title with default margins}

\end{frame}

\molochset{frametitle margin top=1cm}

\begin{frame}
  \frametitle{This is a frame title with increased top margin}
\end{frame}

\molochset{frametitle margin left=0pt}

\begin{frame}
  \frametitle{And now the left margin is removed}
\end{frame}

\end{document}
```

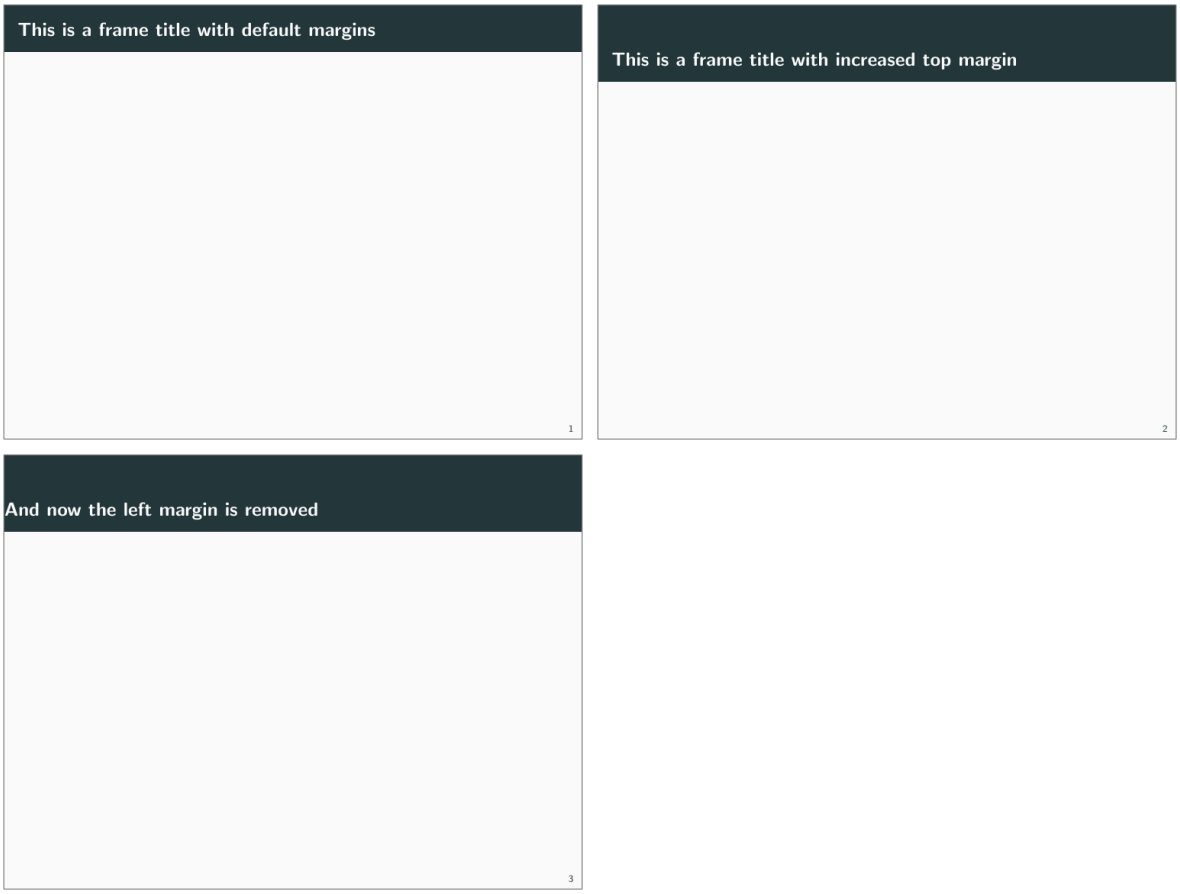

Figure 3.6: Example of the `frametitle margin` options.

3.4 Color Theme

Moloch provides extensive color customization options, from choosing preset color themes to fine-tuning individual colors for light and dark variants.

3.4.1 Color Options

These basic color options control visual aspects of the theme:

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

Optionally adds a light grey background to block environments like `theorem` and `example`. The selected block style is preserved when changing the color theme or switching between light and dark variants.

```latex
\documentclass{beamer}
\usetheme{moloch}

\begin{document}

\begin{frame}
  \begin{block}{Default Block}
    This is a standard block.
  \end{block}

  \molochset{block=fill}

  \begin{block}{Filled Block}
    This is a filled block.
  \end{block}

\end{frame}
\end{document}
```

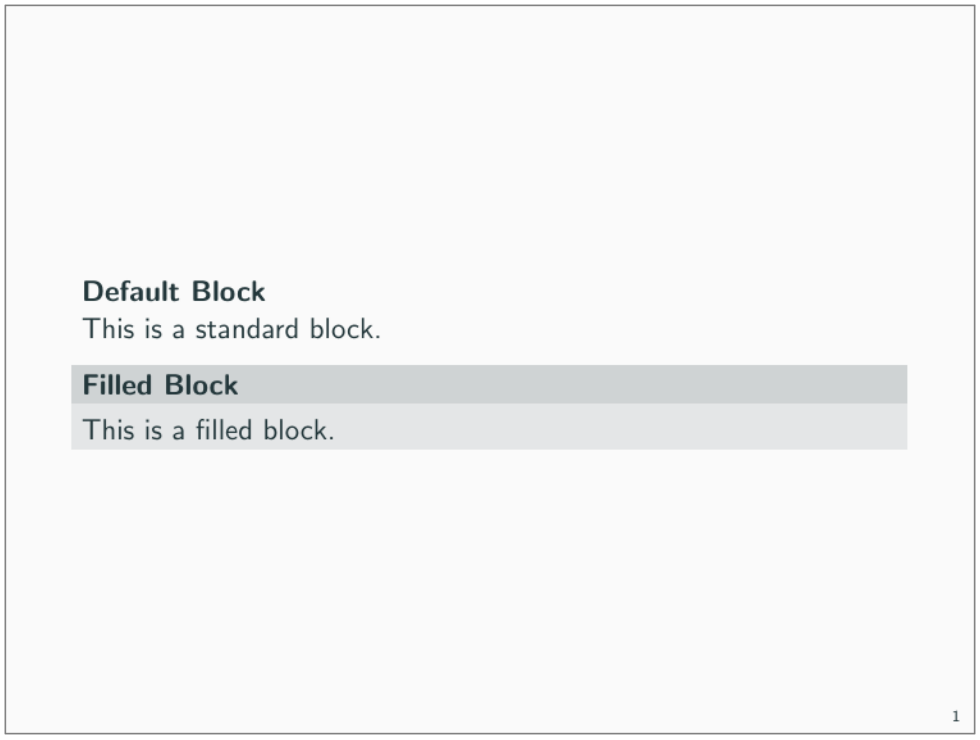

Figure 3.7: Example of the “block” option set to “fill”, adding a light grey background to the second block.

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

Deprecated

Provides the option to have a dark background and light foreground instead of the reverse. This option is deprecated. Use `\molochcolors{variant=dark}` instead.

```
\documentclass{beamer}
\usetheme[background=dark]{moloch}

\begin{document}

\begin{frame}
  \frametitle{Background Color}

  This slide uses the \texttt{background=dark} option, swaps the
  foreground and background colors defined by the moloch theme.
\end{frame}
\end{document}
```

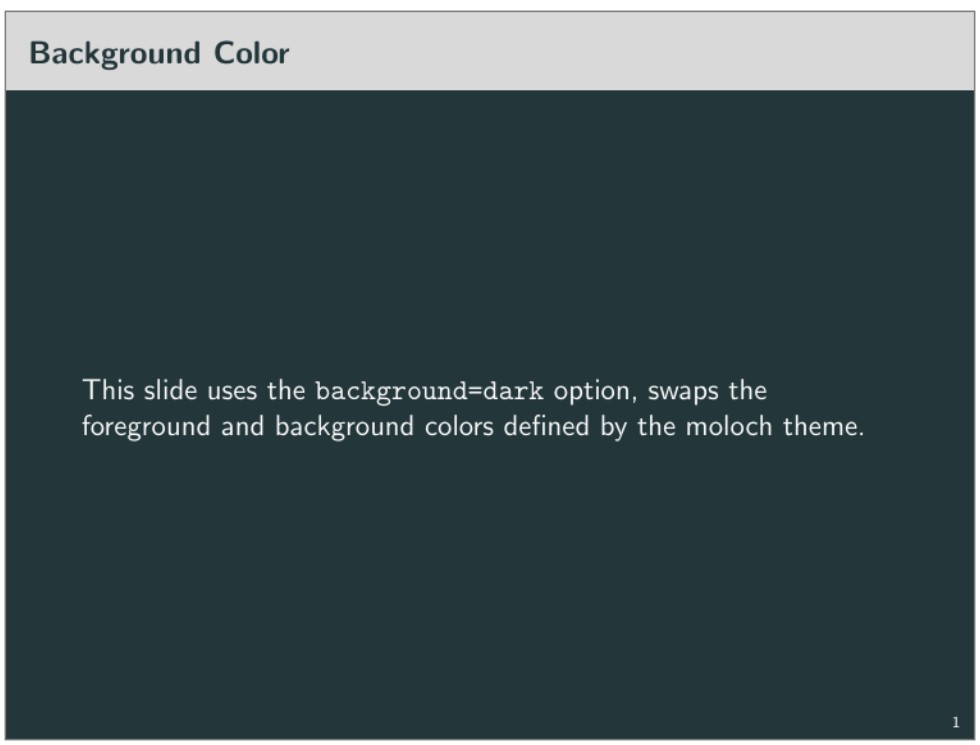

Figure 3.8: Example of the "background" option set to "dark", swapping the foreground and background colors.

3.4.2 Color Themes

Moloch includes several preset color themes. You can select a theme using the `\usecolortheme` command or the newer `\molochcolors{theme=...}` syntax.

3.4.2.1 Default Theme

The default color theme is based on the original Metropolis theme, except we have modified the green color for the example blocks to be a bit less saturated.

```
\usetheme[colortheme=default]{moloch}
% Or equivalently:
% \molochcolors{theme=default}
```

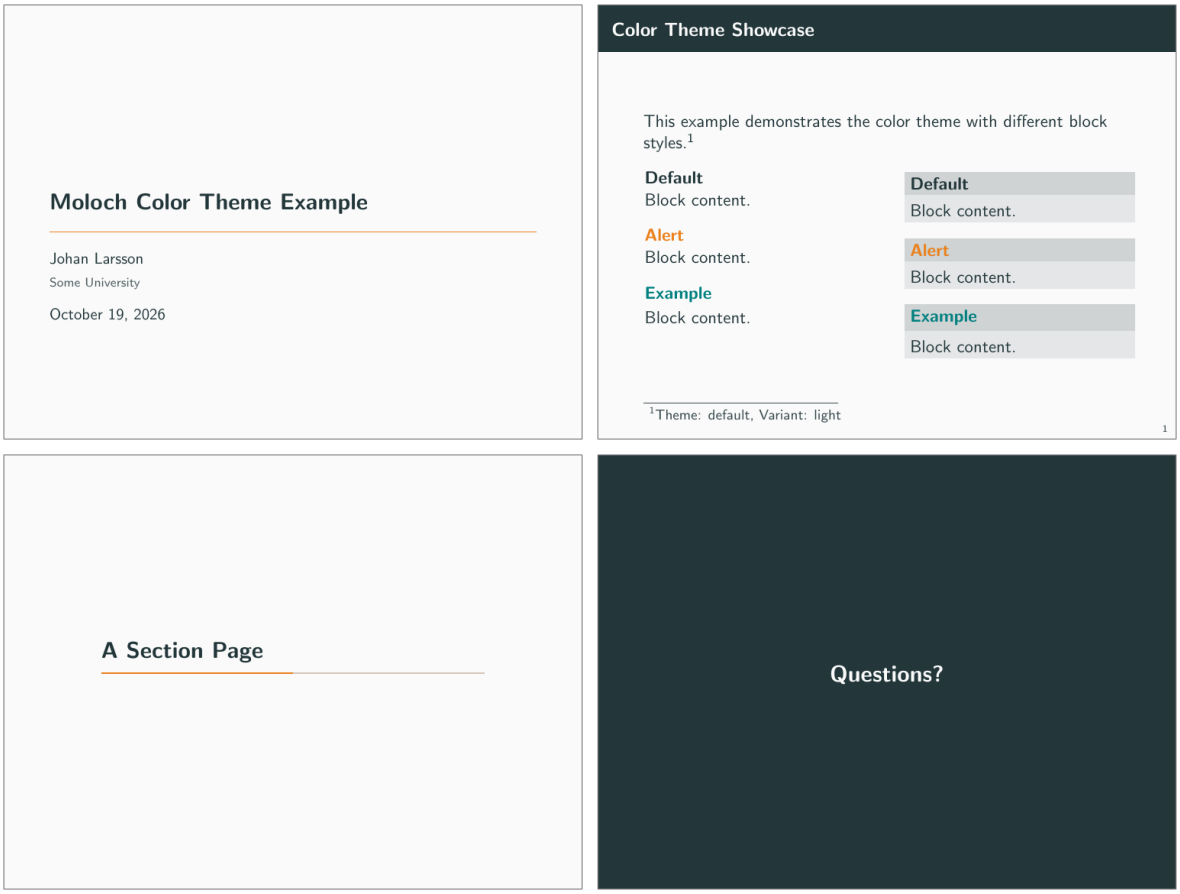

Figure 3.9: Example of the default Moloch color theme.

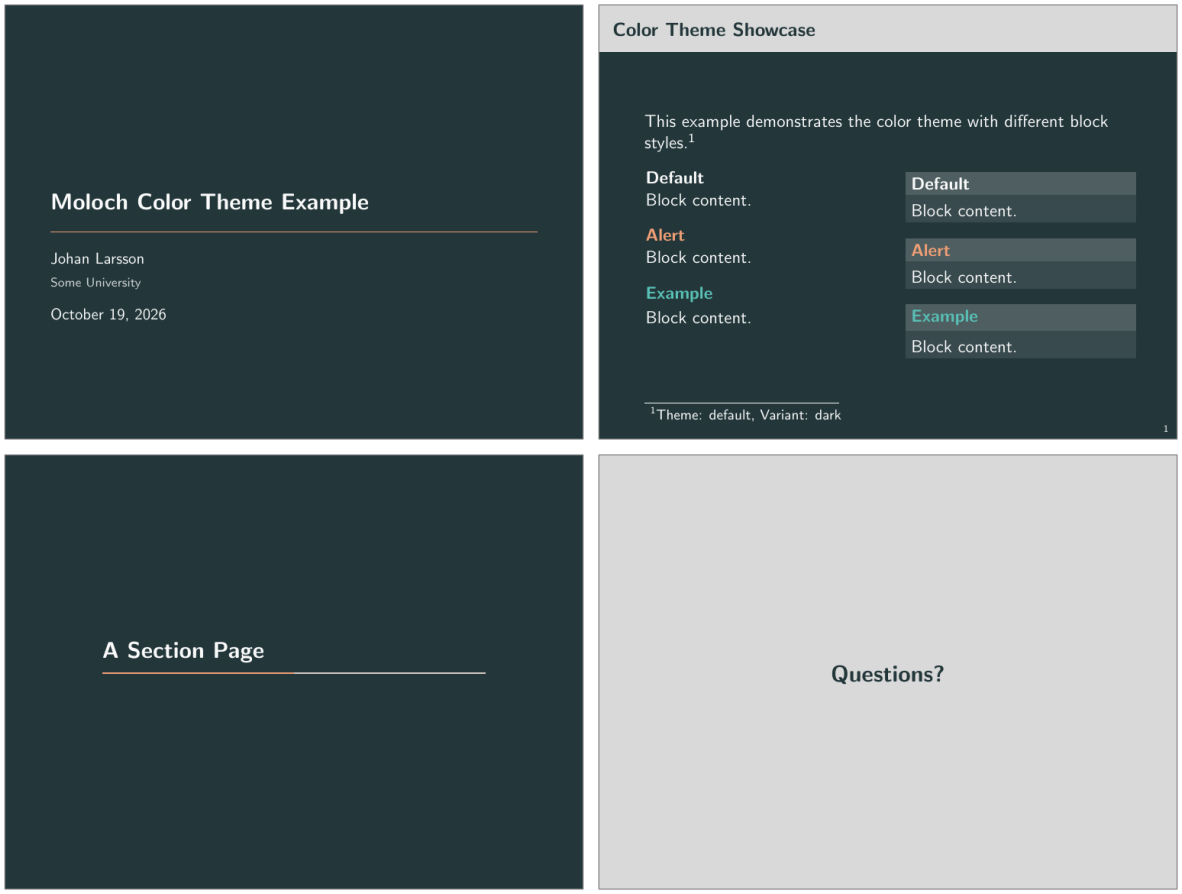

Figure 3.10: Example of the dark version of the default Moloch color theme.

3.4.2.2 The High Contrast Theme

For low-light situations it might be helpful to use the `moloch-highcontrast` color theme. It uses high-contrast colors adapted from the Okabe-Ito color palette, which is designed to be easily distinguishable by individuals with various types of color blindness. Otherwise, the colors used are mostly black and white.

```
\usetheme[colortheme=highcontrast]{moloch}
% Or equivalently:
% \molochcolors{theme=highcontrast}
```

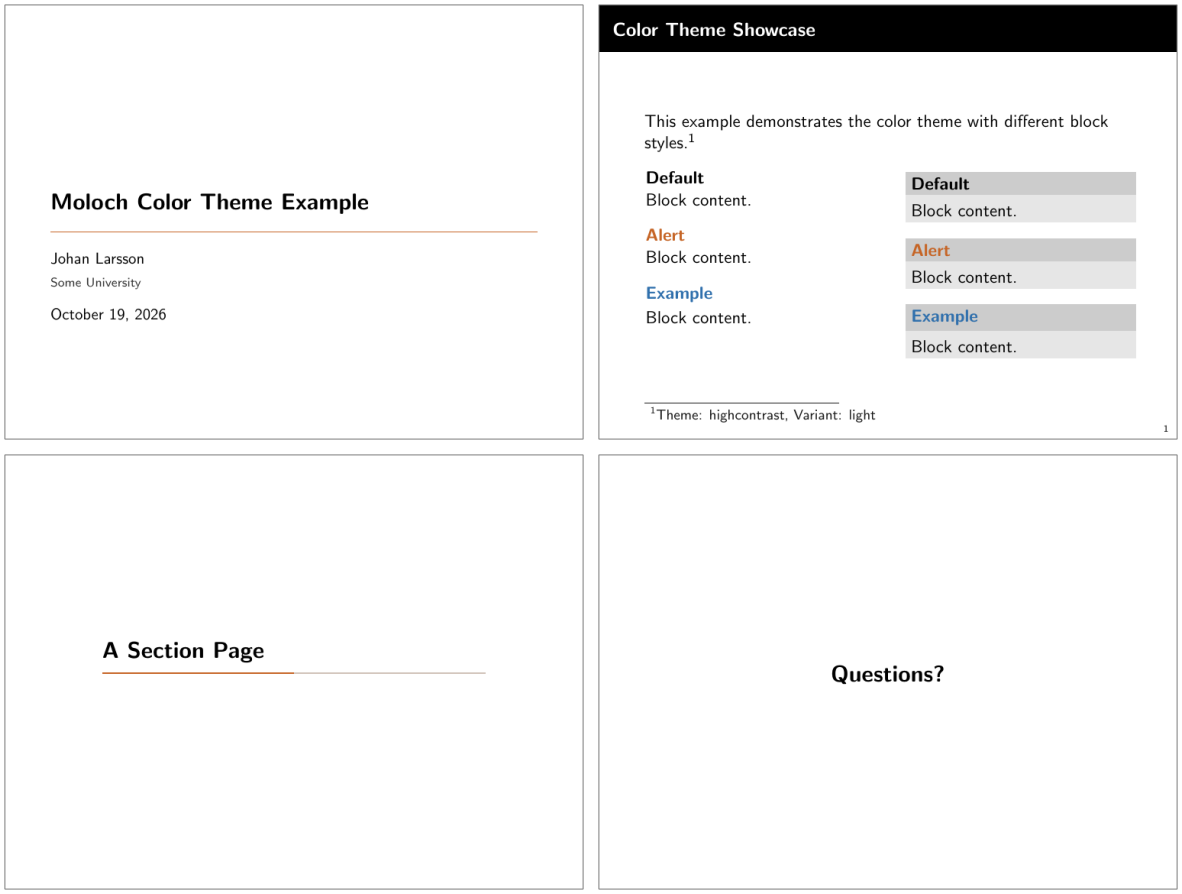

Figure 3.11: Example of the light version of the high contrast color theme.

The dark version of the High Contrast theme uses white text on a black.

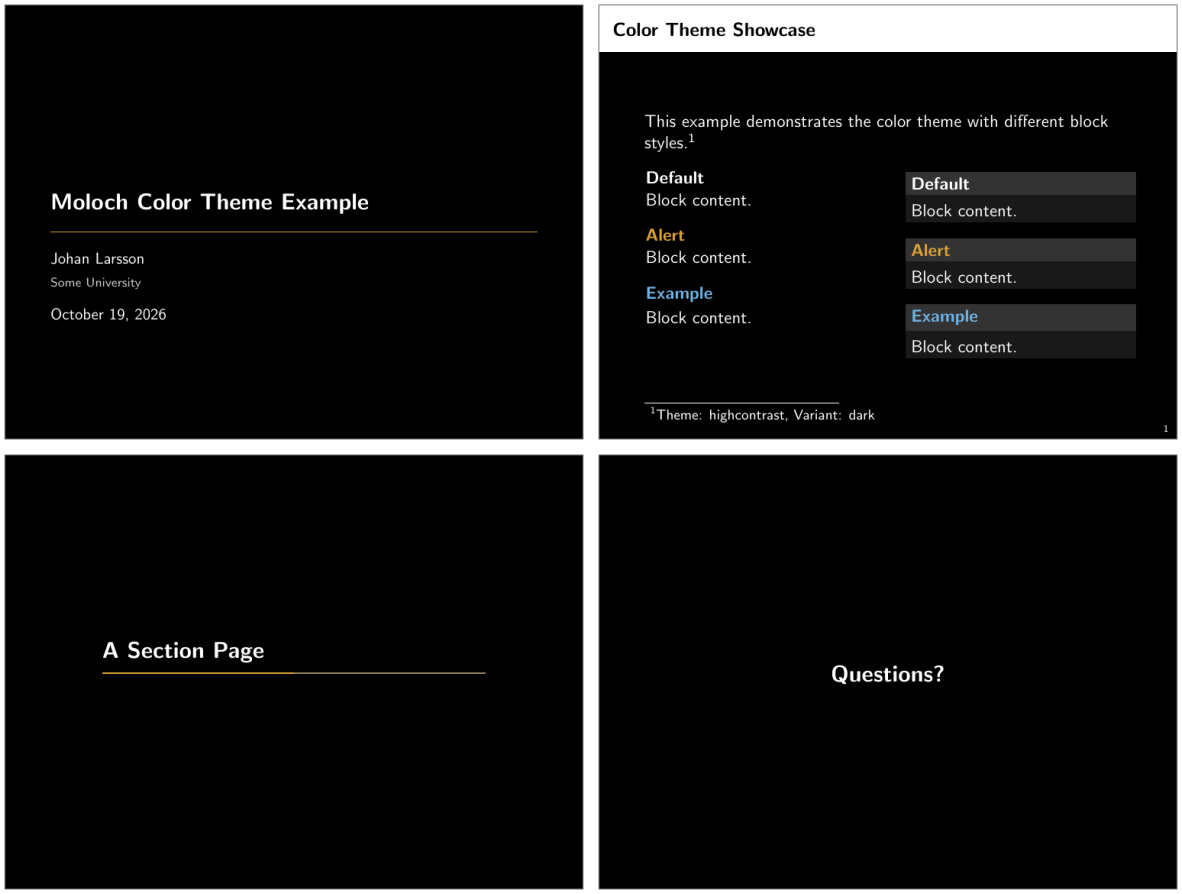

Figure 3.12: Example of the dark version of the high contrast color theme.

3.4.2.3 The Tomorrow Theme

There is also a theme based on the Tomorrow color theme:

```
\usetheme[colortheme=tomorrow]{moloch}
% Or equivalently:
% \molochcolors{theme=tomorrow}
```

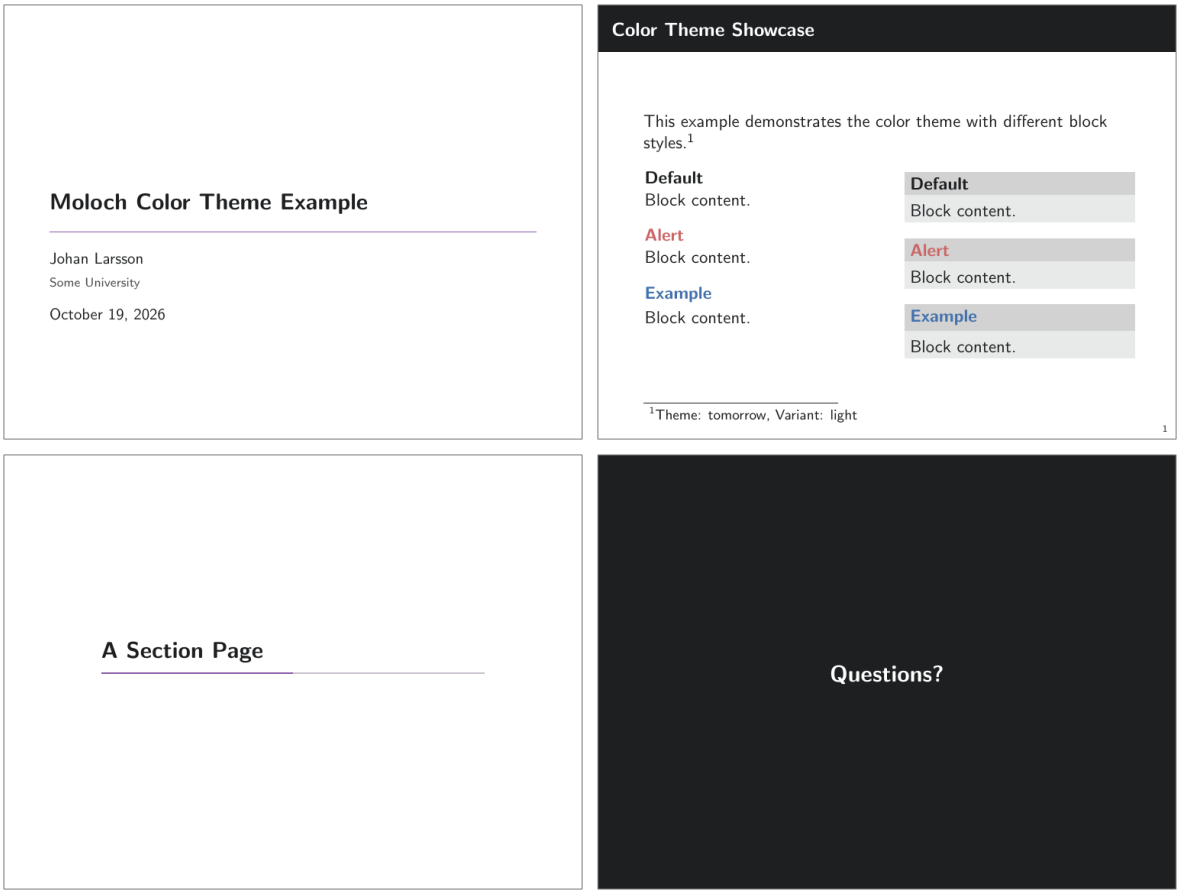

Figure 3.13: Example of the light version of the Tomorrow color theme.

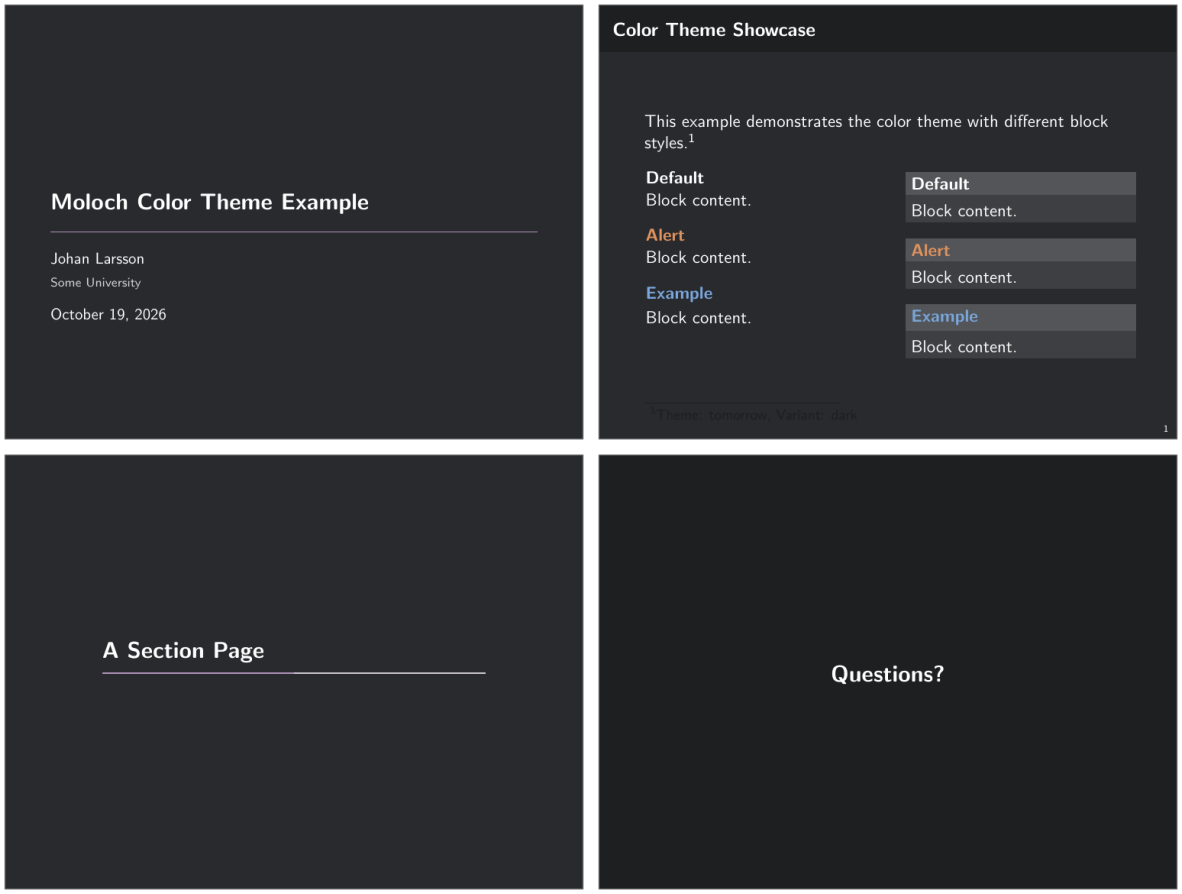

Figure 3.14: Example of the dark version of the Tomorrow color theme.

3.4.2.4 The Paper Theme

A minimalistic theme with black text on white background, no frame title background. The colors uses are based on Stephen Few's *Show Me the Numbers* book.

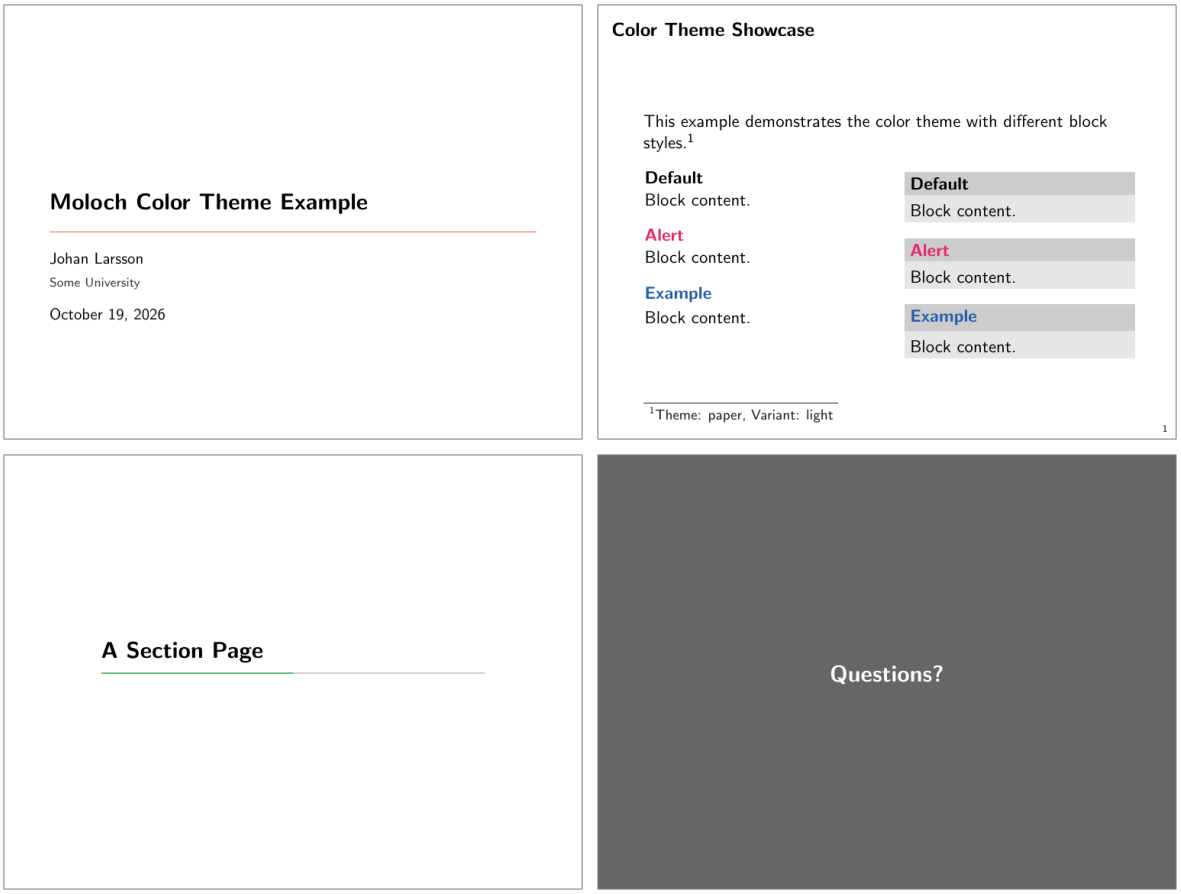

Figure 3.15: Example of the Paper color theme.

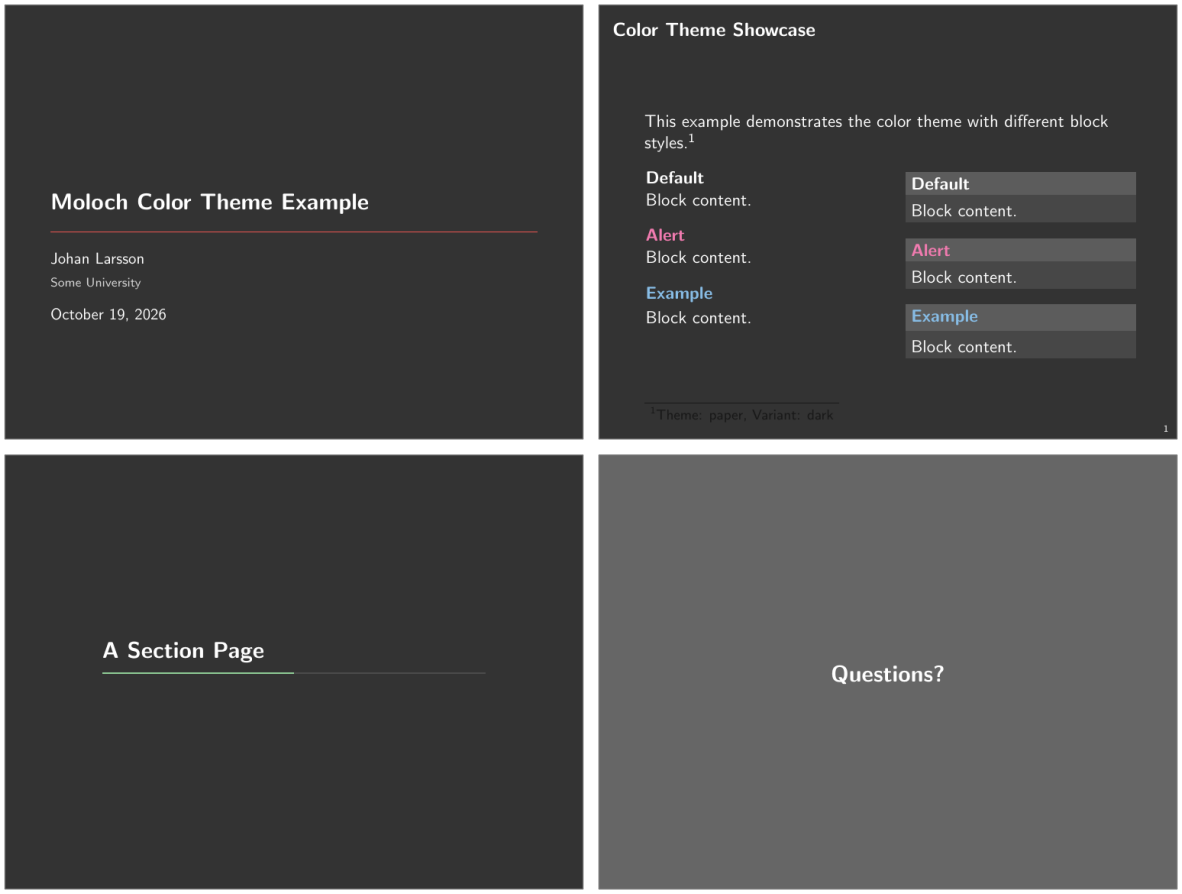

Figure 3.16: Example of the Paper color theme.

3.4.2.5 The Catppuccin Theme

A theme based on the popular Catppuccin color palette. The themes are based on the Latte (light) and Macchiato (dark) variants. The Cattpuccin theme provides a soft color theme with pastel colors and not a lot of contrast, and might therefore work best for online presentations unless you have a good projector and favorable conditions.

```
\usetheme[colortheme=catppuccin]{moloch}
% Or equivalently:
% \molochcolors{theme=catppuccin}
```

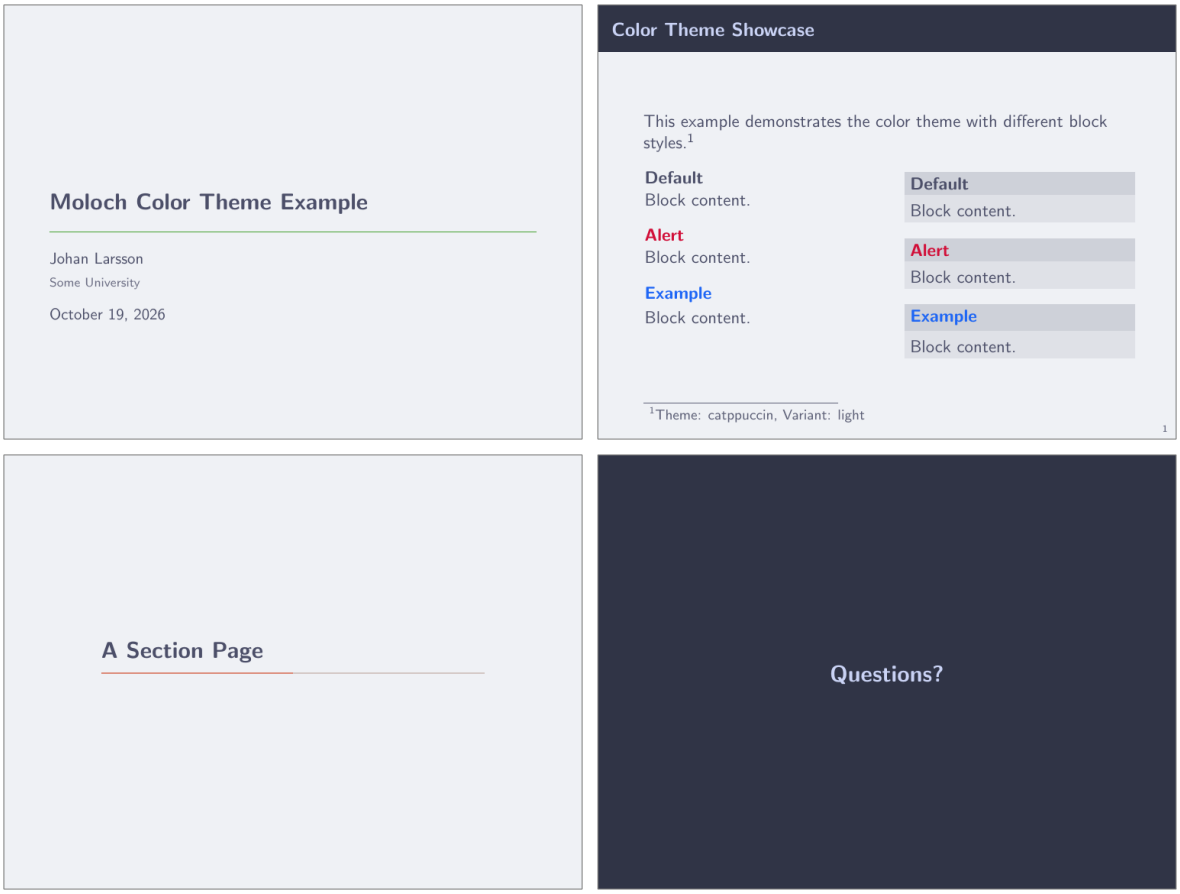

Figure 3.17: Example of the light version of the Catppuccin color theme.

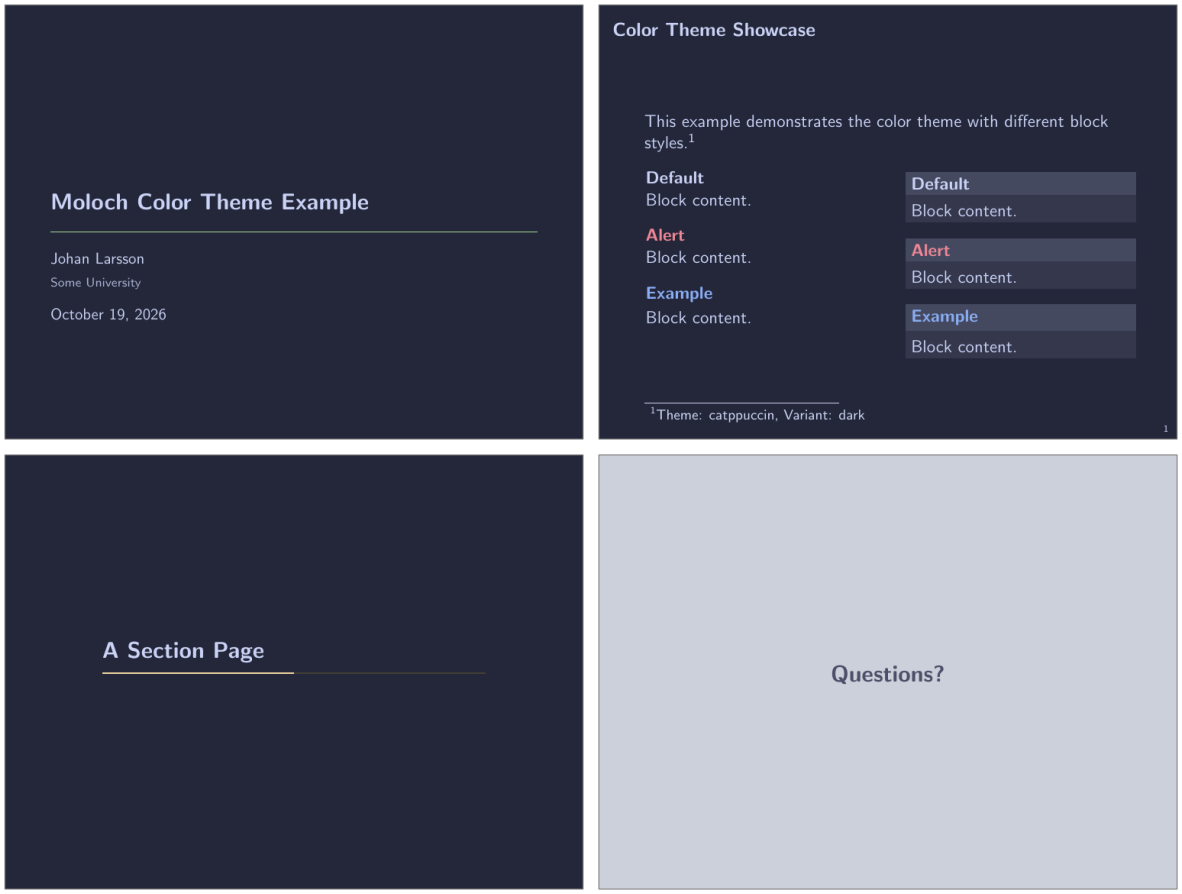

Figure 3.18: Example of the dark version of the Catppuccin color theme.

3.4.3 Color Customization

The included Moloch color theme is used by default, but its colors can be easily changed to suit your tastes. Moloch provides two ways to customize colors:

1. **Using `\molochcolors`** (recommended): A convenient key-value interface for customizing individual colors, especially useful for setting up different colors for light and dark variants. See the detailed documentation below.
2. **Using `\setbeamercolor`** (advanced): For low-level customization or when you need to set colors not exposed through `\molochcolors`, you can use Beamer's standard color setting command:

```
\setbeamercolor{<beamer color name>}{fg=..., bg=...}
```

For example, to customize Moloch-specific colors:

```
\setbeamercolor{progress bar}{fg=orange, bg=orange!50!black!30}
\setbeamercolor{title separator}{fg=teal}
\setbeamercolor{progress bar in head/foot}{parent=progress bar}
\setbeamercolor{progress bar in section page}{parent=progress bar}
```

3.4.4 Granular Color Customization with `\molochcolors`

For more convenient color customization, Moloch provides the `\molochcolors` command, which allows you to customize individual colors using a key-value interface. This is especially useful when you want to customize specific colors for both light and dark variants of a theme.

The basic syntax is:

```
\molochcolors{
  option1=color1,
  option2=color2,
  ...
}
```

3.4.4.1 Theme and Variant Selection

Before customizing individual colors, you can select a theme preset and variant:

`theme` ***default, tomorrow, highcontrast*** ..... default

Selects a color theme preset. Each preset defines complete color schemes for both light and dark variants.

`variant` ***light, dark*** ..... light

Switches between light and dark variants of the current theme. Color customizations are preserved when switching variants, so you can set up colors once and freely toggle between light and dark modes.

```
\documentclass{beamer}
\usetheme{moloch}

\begin{document}

% Start with tomorrow theme in dark mode
\molochcolors{theme=tomorrow, variant=dark}

\begin{frame}
  \frametitle{Dark Tomorrow Theme}
  This uses the Tomorrow theme in dark mode.
```

```latex
\end{frame}

% Switch to light variant
\molochcolors{variant=light}

\begin{frame}
  \frametitle{Light Tomorrow Theme}
  Same theme, but now in light mode.
\end{frame}

\end{document}
```

3.4.4.2 Customizable Colors

You can customize individual colors using the following keys. Each key has three variants:

- **Variant-agnostic** (e.g., `normal text fg`): Sets the color for the current variant and stores it for that variant
- **Light-specific** (e.g., `light/normal text fg`): Sets the color for the light variant (applies immediately if in light mode, otherwise stored for when you switch to light)
- **Dark-specific** (e.g., `dark/normal text fg`): Sets the color for the dark variant (applies immediately if in dark mode, otherwise stored for when you switch to dark)

`alerted text` *<color>* ..... default: theme-dependent

Sets the foreground color for alerted text (e.g., `\alert{text}`) and alerted block titles. This affects both inline alerted text and `alertblock` titles.

`example text` *<color>* ..... default: theme-dependent

Sets the foreground color for example text and example block titles. This affects both inline example text and `exampleblock` titles.

`normal text fg` *<color>* ..... default: theme-dependent

Sets the foreground color for normal text. Changing this automatically updates title elements, author, date, and other text colors that derive from normal text.

`normal text bg` *<color>* ..... default: theme-dependent

Sets the background color for normal text.

`frametitle fg` *<color>* ..... default: theme-dependent

Sets the foreground color for frame titles.

`frametitle bg` *<color>* ..... default: theme-dependent

Sets the background color for frame titles.

`progressbar fg` *<color>* ..... default: theme-dependent

Sets the foreground color for progress bars. This affects progress bars in all locations (head, foot, frame title, section pages).

`progressbar bg` *<color>* ..... default: theme-dependent

Sets the background color for progress bars.

`title separator` *<color>* ..... default: theme-dependent

Sets the foreground color for the title separator (horizontal rule on the title page).

`standout fg` *<color>* ..... default: theme-dependent

Sets the foreground color for standout frames (created with `\begin{frame}[standout]`).

`standout bg` *<color>* ..... default: theme-dependent

Sets the background color for standout frames.

`standout alerted fg` *<color>* ..... default: theme-dependent

Sets the foreground color for alerted standout frames (created with `\begin{frame}[standout=alerted]`).

`standout alerted bg` *<color>* ..... default: theme-dependent

Sets the background color for alerted standout frames.

`standout example fg` *<color>* ..... default: theme-dependent

Sets the foreground color for example standout frames (created with `\begin{frame}[standout=example]`).

`standout example bg` *<color>* ..... default: theme-dependent

Sets the background color for example standout frames.

3.4.4.3 Example: Customizing Colors Per Variant

You can set different colors for light and dark variants:

```
\documentclass{beamer}
\usetheme{moloch}
```

```latex
\begin{document}

% Customize colors for both variants
\molochcolors{
  light/alerted text=red!80!black,
  dark/alerted text=orange,
  light/progressbar fg=blue,
  dark/progressbar fg=cyan
}

% Start in light mode
\molochcolors{variant=light}

\begin{frame}
  \frametitle{Light Mode}
  \alert{Alerted text} is red-ish, progress bar is blue.
\end{frame}

% Switch to dark mode - your customizations are preserved!
\molochcolors{variant=dark}

\begin{frame}
  \frametitle{Dark Mode}
  \alert{Alerted text} is orange, progress bar is cyan.
\end{frame}

\end{document}
```

3.4.4.4 Example: Quick Color Tweaks

For simple adjustments to the current variant:

```latex
\molochcolors{
  alerted text=purple,
  progressbar fg=teal
}
```

This sets the colors for whichever variant is currently active (light or dark).

3.5 Font Theme

`titleformat plain`, `titleformat frametitle`, `titleformat section` ***regular, smallcaps, allsmallcaps, allcaps*** ..... regular

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

4 Commands

4.1 Standout Frames

The Moloch 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:

```latex
\documentclass{beamer}
\usetheme{moloch}

\begin{document}

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

\begin{frame}[standout=alerted]
  Important warning!
\end{frame}

\begin{frame}[standout=example]
  Positive takeaway.
\end{frame}

\end{document}
```

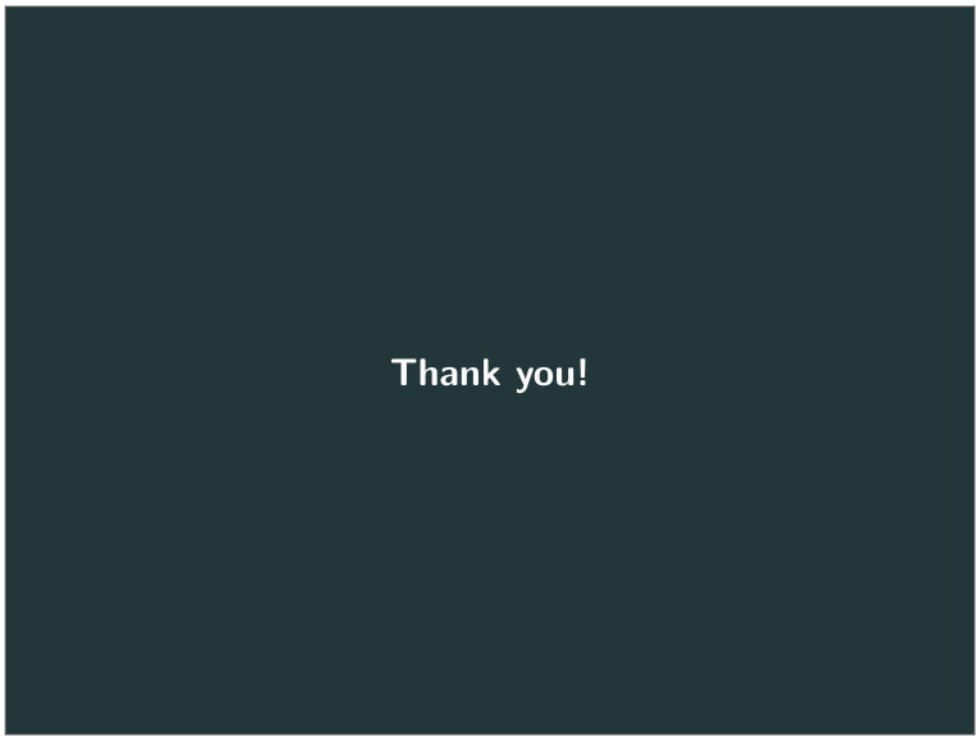

Figure 4.1: Example of a standout frame with large, centered text and an inverted background.

5 ltx-talk

> **Warning**
>
> This part of Moloch is experimental. It targets ltx-talk, which is itself an experimental 0.x class whose author warns that "changes may occur to almost all interfaces". Expect changes to both the class and this theme.

`ltx-talk` is a presentation class by Joseph Wright, built around tagged PDF. That is the main reason to prefer it over Beamer: Beamer cannot currently produce properly tagged, accessible output, and `ltx-talk` is designed to.

Moloch provides its design for this class through the `moloch-talk` package.

5.1 Requirements

Moloch is tested with `ltx-talk` 0.6.10, which needs a LaTeX kernel from 2026-06-01 or later. With an older kernel, you will see:

```
! LaTeX Error: Class ltx-talk Error: LaTeX kernel too old.
```

The class supports only `pdflatex` and `lualatex`. Use `lualatex` if you want tagged output.

5.2 Usage

Unlike Beamer, `ltx-talk` has no theme mechanism, and its author has said that themes are deliberately not part of its design. So there is no `\usetheme` here: Moloch is loaded as an ordinary package, after the class.

```latex
\DocumentMetadata{lang=en-US,tagging=on}
\documentclass{ltx-talk}
\usepackage{moloch-talk}

\title{A Minimal Example}
\subtitle{Moloch for ltx-talk}
\author{Johan Larsson}
```

```latex
\institute{Some University}
\date{\today}

\begin{document}

\maketitle

\section{First Section}

\begin{frame}
  \frametitle{First Frame}

  Hello, world!
\end{frame}

\begin{standout}
  Thank you!
\end{standout}

\end{document}
```

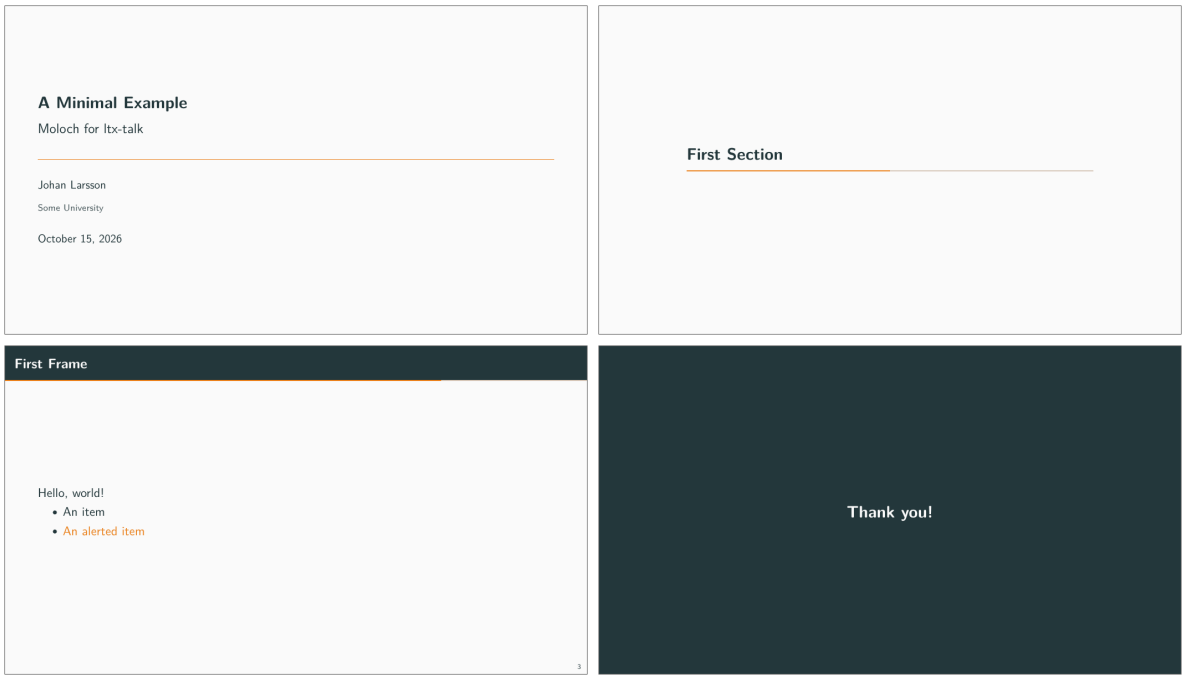

Figure 5.1: An example of Moloch for `ltx-talk`.

The `\DocumentMetadata` line is required and must come before `\documentclass`. Set `lang` to the presentation's language and use `tagging=on` to enable tagged output.

The class defaults to a 16:9 aspect ratio, while Beamer defaults to 4:3. To match the Beamer demo, use `\documentclass[10pt,aspect-ratio=4:3]{ltx-talk}`. The slides still have different physical dimensions: `ltx-talk` fixes the slide height at 100mm, while Beamer's default 4:3 slides are 96mm high.

5.3 Options

You can set options when loading the package or later with `\molochset`. Changes to the palette, progress bar, and frame number settings inside a frame apply to that frame. The previous settings are restored afterward.

`variant` ***light, dark*** ..... light

Selects the light or dark version of the color palette. Unlike in the Beamer theme, you can also change the palette in the middle of a document. A change inside a frame applies to its text, header, footer, and background. The previous palette is restored after the frame.

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

Adds a progress bar at the top of the slide, directly below the frame title band, or at the bottom. The bar is not drawn on the title page, on section pages, or on standout frames.

`sectionpage` ***none, simple, progressbar*** ..... progressbar

Controls whether each `\section` produces a section page and whether that page carries a progress bar. The class ignores `\section` entirely when it appears inside a frame, so place it outside a frame to produce a section page.

`framenumber` ***true, false*** ..... true

Controls whether the frame number appears in the bottom right corner.

`progressbar-linewidth` ***<length>*** ..... default: 1pt

Sets the thickness of the progress bar.

`titleseparator-linewidth` ***<length>*** ..... default: 0.5pt

Sets the thickness of the rule between the title block and the author block on the title page.

5.4 Customization beyond these options

The option list above is deliberately short. Use the LaTeX template layer to adjust aspects of the appearance that these options do not cover. For example, to change the height of the frame title band and the font it uses:

```
\geometry{tmargin = 15mm, headheight = 15mm}
\EditInstance{frametitle}{header}{font = \Large\bfseries}
```

The palette uses ordinary color definitions. To change a color:

```
\definecolor{mFrametitleBg}{HTML}{4C566A}
```

See the `ltx-talk` manual for the available templates and keys.

5.4.1 Footer

Moloch uses its own `footer` instance, named `moloch`. Edit this instance to add content while keeping the theme's margins and progress bar:

```
\EditInstance{footer}{moloch}{element-order={author,framenumber}}
```

This places the author on the left and the frame number on the right. The footer uses short metadata when available, such as the optional argument to `\author`. Empty elements are omitted, and `framenumber=false` hides the frame number without hiding other content. You can also adjust `font`, `color`, `left-hspace`, `right-hspace`, and `separator`, or customize each element through the class's `footer-element` instances.

Captions use smaller text and bold labels while retaining the class's unnumbered format.

Itemize lists use the same drawn markers as the Beamer theme: a filled circle, square, and triangle for the first three nesting levels. The fourth level retains the class's default marker. Customize the markers with `\renewcommand{\labelitemi}{...}` (or `\labelitemii` and `\labelitemiii`), or use `\item[<label>]` for an individual item.

5.5 Differences from the Beamer theme

Moloch for `ltx-talk` is a starting point rather than a complete port of the Beamer theme. Some differences reflect features not yet implemented; others follow from the class's design.

The title block is centered vertically. In Beamer, it sits slightly above center in golden-ratio proportion.

Moloch for `ltx-talk` does not yet support the following features:

- The alternative color themes (`tomorrow`, `highcontrast`, `catppuccin`, `paper`). Only the default palette is available, in light and dark.
- The `titleformat` options for small caps and capitalization.
- `standout=alerted` and `standout=example`.

The class imposes the following limitations and differences from Beamer:

- **The frame title band's height is fixed.** Long titles wrap, but the band cannot grow to fit. Increase `tmargin` and `headheight` with `\geometry` when titles need more space, and increase `headsep` to keep the extra lines inside the band. For example, for a two-line title:

  ```
  \geometry{tmargin=20mm,headheight=20mm,headsep=8mm}
  ```

- **`\framesubtitle` produces no output.** The class stores the subtitle but does not currently typeset it.
- **Section pages and standout frames count as frames**, so they advance the frame number. The Beamer theme does not count them by default.
- **No predefined `alertblock`, `exampleblock`, or theorem environments.** Define theorem environments with `\newtheorem`. Blocks cannot be filled with a background color: `block` is a fixed environment in the class rather than a template.
- **No `\titlegraphic`.** The class has no such command.
- With `lualatex`, the class selects New Computer Modern Sans by file name, leaving no bold face registered. Bold text therefore silently appears in the regular weight. Either compile with `pdflatex` or set up the fonts yourself with `fontspec`.

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 for all fonts. In particular, the Computer Modern sans-serif typeface, which is used by default when Moloch is compiled with pdfLaTeX, does not have a small-caps variant.

Note that title format options `allsmallcaps` and `allcaps` do not affect the sizes of numerals, punctuation, and math symbol, and are probably best avoided if your titles contain these characters.

6.2 Interactions with Other Color Themes

Moloch 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 Moloch subpackages individually so the Moloch color theme is never loaded. This will prevent conflicts between the Moloch color theme and your preferred theme.

For example, overriding the color theme as follows may not work as expected because '`loads the Moloch 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{moloch}
\usecolortheme{seahorse}
```

The correct colors are chosen if the Moloch outer, inner, and font themes are loaded seperately:

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

Please note that Moloch may not use all the colors defined in your favourite Beamer color theme. In particular, Moloch 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 XeLaTeX, 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 itself. You can work around it either by compiling with 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 to 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

Moloch 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.

You can read the full license text above or in the license file at https://github.com/jolars/moloch/blob/main/LICENSE.

8 Changelog

8.1 2.3.1 (2026-10-19)

8.1.1 Bug Fixes

- support `ltx-talk` 0.6.10 (`dc10638`), fixes #89

8.2 2.3.0 (2026-10-15)

8.2.1 Highlights

Moloch is now also available as a ltx-talk theme! The theme is still experimental but please feel free to try it out and report any issues you encounter.

8.2.2 Features

- add an ltx-talk version of the theme (#86) (`862938a`), closes #76
- enable alerted and example standout styles (#80) (`dfdc31e`)
- add `sectionpagenumbering` for section-page frame numbering (#83) (`bc20862`)

8.2.3 Bug Fixes

- use full titles on divider pages (`291ff48`), fixes #87
- **ci:** use a single TeX Live mirror (`9507cad`)
- **ci:** bypass broken TeX Live mirror list (`5451656`)
- preserve block fills across color changes (`11c841d`), fixes #84
- **readme:** minor typo correction (`5d59da6`)

8.3 2.2.0 (2026-02-17)

8.3.1 Features

- increase margin between frame title and subtitle (f8e8725)
- increase top-bottom margins for frametitle, use strut (f537931)

8.3.2 Bug Fixes

- add missing color option for title separator (1221aed), closes #70

8.4 2.1.0 (2025-12-18)

8.4.1 Features

- add `titlepage` option with moloch, plain, slit styles (505426d)

8.4.2 Bug Fixes

- alias `colortheme variant` to `colorthemevariant` (70a2bf2)
- don't expose all colortheme settings as package options (4827f60)
- don't wrap titlepage frame in group (b6be996)

8.5 2.0.0 (2025-12-05)

8.5.1 BREAKING CHANGES

- refactor color theme

8.5.2 Features

- add `frametitle margin` options (bc02e8a)
- add catppuccin theme (3452624)
- add new "Paper" color theme (05de59a)
- change default progressbar linewidth to 1pt (3b71fc9)
- draw all itemize symbols manually (a82527c)
- refactor color theme (37d3a37)

- tune color themes (a11710f)
- use brighter aler and example colors in dark hcontr (8babd1e)

8.5.3 Bug Fixes

- enable background macro for alternative color themes (59b4d79)
- remove and fix examples (35523ee)
- remove example text bg in highcontrast theme (528ea8d)
- work-around division by zero in progress bar arithmetic (28fc1a2)

8.6 1.1.0 (2025-09-25)

8.6.1 Features

- redesign itemize environment (#50) (8594369)
- use beamer-like itemize item and subitem templates (81037c6), closes #49

8.6.2 Bug Fixes

- move command substitutions into inner and outer themes (fd86a30), closes #47
- use `\ht\strutbox` instead of X height in title padding (93792f2), closes #54

8.7 1.0.1 (2025-04-01)

8.7.1 Bug Fixes

- add missing files for updating version tags (ba01d33)
- use proper package names (0c4d38e)

8.8 1.0.0 (2025-04-01)

8.8.1 Features

- add customization of bar line width (#42) (be0d8f2)
- support `\thanks` in title page (#45) (de3db5e)

8.9 0.6.0 (2025-01-17)

8.9.1 Features

- add back highcontrast theme from metropolis (3039994)
- add new color theme based on the tomorrow color theme (13e7300)
- add option to customize frame numbering of standout pages (d6d0e3c), closes #33
- deprecate `\mreducelistspacing` (3d87ec2)
- redesign section and subsection pages (#30) (5a98c69)
- use golden rule in top-bottom spacing of title page (85eabc7)

8.9.2 Reverts

- "feat: deprecate `\mreducelistspacing`" (eb4bf54)
- "test: remove standoutnumbering test" (ce7540d)

8.10 0.5.0 (2024-10-30)

8.10.1 Features

- replace `\vfill` with `\vfil` in title frame (8f2a59e)
- switch from fill to fil (#21) (873ae15)

8.10.2 Bug Fixes

- remove duplicated footnote color specifications (#26) (1306af5)
- remove extra space below progressbar section pages (#27) (e5be480)
- set use=block title in block body for fill (e8921a6), closes #28
- tag files with date (28003e7)

8.11 0.4.0 (2024-07-09)

8.11.1 Features

- add a small amount of vertical space before subtitle (b224a9b)
- support memoize package (closes #16) (1ca26e8)
- support title format modifications (#18) (46a4f49)

8.11.2 Bug Fixes

- add % at the end of some commands (33b9833)
- convert raw command in manual to text (ac305e9)

8.12 0.3.0 (2024-03-06)

8.12.1 Features

- bring back the `numbering` option and deprecate it (9a013e4)
- make `framenumber` slide numbering the default (54e0fce)
- redesign title page (9e6506e)
- reinstate support for `block` option (bf63dcd)
- remove custom linespread setting (6a85ef4), closes #8
- remove parskip setting (5477051), closes #9
- support subtitles (1c49957)

8.12.2 Bug Fixes

- apply upstream beamer fix for empty shortauthor (b8b2d10)
- improve vertical centering of title slide (6a14981)
- remove extraneous line space after minipage (c0b9826)
- remove extraneous space in frametitle continuation template (66f7d83)
- remove font configuration for bibliography; closes #5 (ccd19fe)

8.13 0.2.0 (2024-01-23)

8.13.1 BREAKING CHANGES

- remove titleformat configuration
- use beamer's own frame numbering template

8.13.2 Features

- remove titleformat configuration (e19f93c)
- use beamer's own frame numbering template (01b698c)

8.14 0.1.0 (2024-01-15)

8.14.1 BREAKING CHANGES

- remove `\plain` macro
- drop deprecated options
- remove appendix redefinition
- remove pgfplots theme
- remove high-contrast theme
- uncouple fira fonts from theme
- simplify blocks and remove blocks with background

8.14.2 Features

- disable hyphenation in frame titles (253234e)
- drop deprecated options (548ede8)
- group institute with author and subdue color (0f3f6da)
- increase minipage width for section pages (160fe58)
- modify bullet symbols for itemize lists (0042dcb)
- modify example color to teal instead of light green (1a2f157)
- remove `\plain` macro (95f24ae)
- remove appendix redefinition (17edfad)
- remove high-contrast theme (9932e55)
- remove pgfplots theme (c2f1bef)
- simplify blocks and remove blocks with background (d5317d4)
- uncouple fira fonts from theme (f960aba)

8.14.3 Bug Fixes

- avoid overflow warnings from titlepage definition (c1c5b85)
- fix frametitle patchmd (a302893)
- fix progress bar in inner theme too (47dcf8d)
- fix reference to Knuth in demo (fbfcc0b)
- insert missing parenthesis (e72eaa9)
- metropolis -> moloch (24cbc04)
- set maxium progressbar to 1 (b401410)

8.14.4 Reverts

- "build: use alternative texlive action" (5d32f73)

- “fix: avoid overflow warnings from titlepage definition” (24c5409)

8.14.5 Miscellaneous Chores

- remove newline (cd97b2d)

Part II

Implementation

A Main Theme

> **Note**
>
> **Source file:** `src/beamerthememoloch.dtx`

A.1 Moloch Parent Theme

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

A.2 Package Dependencies

```
\RequirePackage{pgfopts}
```

A.3 Options

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

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

Route color theme options to the new color system.

```
\pgfkeys{/moloch/.cd,
  colortheme/.code=\pgfkeysalso{/moloch/colors/theme=#1},
  colorthemevariant/.code=\pgfkeysalso{/moloch/colors/variant=#1},
```

```
  colortheme variant/.code=\pgfkeysalso{/moloch/colors/variant=#1},
}
```

A.3.1 titleformat plain

Controls the formatting of the text on standout "plain" frames.

```
\pgfkeys{
  /moloch/titleformat plain/.cd,
  .is choice,
  regular/.code={%
      \let\moloch@plaintitleformat\@empty%
      \setbeamerfont{standout}{shape=\normalfont}%
    },
  smallcaps/.code={%
      \let\moloch@plaintitleformat\@empty%
      \setbeamerfont{standout}{shape=\scshape}%
    },
  allsmallcaps/.code={%
      \let\moloch@plaintitleformat\MakeLowercase%
      \setbeamerfont{standout}{shape=\scshape}%
    },
  allcaps/.code={%
      \let\moloch@plaintitleformat\MakeUppercase%
      \setbeamerfont{standout}{shape=\normalfont}%
    },
}
```

A.3.2 titleformat

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

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

Set default values for options.

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

A.4 Component Sub-Packages

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

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

A.5 Custom Commands

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

A.5.1 \molochset

Allows the user to change options midway through a presentation.

```
\newcommand{\molochset}[1]{\pgfkeys{/moloch/.cd,#1}}
```

DEPRECATED: The following command is deprecated and will be removed in a future version.

A.5.2 \mreducelistspacing

```
\newcommand{\mreducelistspacing}{%
  \PackageWarning{beamerthememoloch}{%
    \string\mreducelistspacing\space is deprecated and will be removed in a future version}%
  \vspace{-\topsep}%
}
```

A.6 Process Package Options

```
\moloch@setdefaults
\ProcessPgfOptions{/moloch}
```

B Inner Theme

> **Note**
>
> **Source file:** src/beamerinnerthememoloch.dtx

The `beamer` inner theme dictates the style of the frame elements traditionally set in the "body" of each slide.

B.1 Package Dependencies

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

B.2 Memoization and Tikz Externalization

See the documentation for the correspondign section under the outer theme for more information on the following lines.

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

B.3 Options

B.3.1 `sectionpage`

Optionally add a slide marking the beginning of each section.

```
\pgfkeys{
  /moloch/inner/sectionpage/.cd,
  .is choice,
```

```
  none/.code=\moloch@disablesectionpage,
  simple/.code={%
      \moloch@enablesectionpage%
      \setbeamertemplate{section page}[simple]%
    },
  progressbar/.code={%
      \moloch@enablesectionpage%
      \setbeamertemplate{section page}[progressbar]%
    },
}
```

B.3.2 subsectionpage

Optionally add a slide marking the beginning of each subsection.

```
\pgfkeys{
  /moloch/inner/subsectionpage/.cd,
  .is choice,
  none/.code=\moloch@disablesubsectionpage,
  simple/.code={%
      \moloch@enablesubsectionpage%
      \setbeamertemplate{section page}[simple]%
    },
  progressbar/.code={%
      \moloch@enablesubsectionpage%
      \setbeamertemplate{section page}[progressbar]%
    },
}
```

B.3.3 standoutnumbering

Whether or not to number standout pages. Option `none` means that standout pages are not numbered (do not count as frames). `hide` means that they do count as frames, but that the footer with the number is not shown. Option `show` means that they both count as frames and that the footer with a frame count is shown.

```
\providebool{moloch@enableStandoutFooter}
\providebool{moloch@enableStandoutNumbering}
\pgfkeys{
  /moloch/inner/standoutnumbering/.cd,
  .is choice,
  none/.code={
```

```
      \boolfalse{moloch@enableStandoutNumbering}
      \boolfalse{moloch@enableStandoutFooter}
    },
  show/.code={
      \booltrue{moloch@enableStandoutNumbering}
      \booltrue{moloch@enableStandoutFooter}
    },
  hide/.code={
      \booltrue{moloch@enableStandoutNumbering}
      \boolfalse{moloch@enableStandoutFooter}
    }
}
```

B.3.4 `sectionpagenumbering`

Whether or not to number section pages. Option `none` (the default) means that section pages are not numbered (do not count as frames), which matches the historical behaviour. Option `show` means that they count as frames, so that the frame number is contiguous across section pages. This mirrors the `standoutnumbering` option above. (Section pages are `plain`, so the number is not printed on the section page itself; the effect is that subsequent frames' numbers account for the section pages.)

```
\providebool{moloch@enableSectionPageNumbering}
\pgfkeys{
  /moloch/inner/sectionpagenumbering/.cd,
  .is choice,
  none/.code={\boolfalse{moloch@enableSectionPageNumbering}},
  show/.code={\booltrue{moloch@enableSectionPageNumbering}},
}
```

B.3.5 `titleseparator linewidth`

Set the width of the line separating the title from the author.

```
\newlength{\moloch@titleseparator@linewidth}
\pgfkeys{
  /moloch/inner/.cd,
  titleseparatorlinewidth/.code={\setlength{\moloch@titleseparator@linewidth}{#1}},
}
```

B.3.6 `titleseparator aliases`

Allows `titleseparator linewidth` to be used in `\usetheme`.

```
\pgfkeys{
  /moloch/inner/.cd,
  titleseparator linewidth/.code=\pgfkeysalso{titleseparatorlinewidth=#1},
}
```

B.3.7 `titlepage`

Select which title page template to use.

```
\pgfkeys{
  /moloch/inner/titlepage/.cd,
  .is choice,
  moloch/.code={%
      \setbeamertemplate{title page}[moloch]%
      \setbeamertemplate{title}[moloch]%
      \setbeamertemplate{subtitle}[moloch]%
      \setbeamertemplate{author}[moloch]%
      \setbeamertemplate{institute}[moloch]%
      \setbeamertemplate{date}[moloch]%
      \setbeamertemplate{title graphic}[moloch]%
    },
  plain/.code={%
      \setbeamertemplate{title page}[plain]%
      \setbeamertemplate{title}[plain]%
      \setbeamertemplate{subtitle}[plain]%
      \setbeamertemplate{author}[plain]%
      \setbeamertemplate{institute}[plain]%
      \setbeamertemplate{date}[plain]%
      \setbeamertemplate{title graphic}[plain]%
    },
  split/.code={%
      \setbeamerfont{date}{size=\footnotesize}
      \setbeamertemplate{title page}[split]%
      \setbeamertemplate{title}[split]%
      \setbeamertemplate{subtitle}[split]%
      \setbeamertemplate{author}[split]%
      \setbeamertemplate{institute}[split]%
      \setbeamertemplate{date}[split]%
      \setbeamertemplate{title graphic}[split]%
```

```
    },
}
```

B.3.8 \moloch@inner@setdefaults

Set default values for inner theme options.

```
\newcommand{\moloch@inner@setdefaults}{
  \pgfkeys{/moloch/inner/.cd,
    sectionpage=progressbar,
    subsectionpage=none,
    standoutnumbering=none,
    sectionpagenumbering=none,
    titleseparator linewidth=0.5pt,
    titlepage=moloch,
  }
}
```

B.4 Title Page

B.4.1 title page

Template for the title page. The moloch variant uses Moloch's custom layout with golden ratio spacing and a title separator. The plain variant mimics Beamer's default title page layout.

B.4.2 Moloch Variant

```
\defbeamertemplate*{title page}{moloch}{
  \null%
  \vspace{0pt plus 1.618fil}%
  \vfil%
  \ifx\insertitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
  \ifx\insertitle\@empty\else\usebeamertemplate*{title}\fi
  \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
  \usebeamertemplate*{title separator}
  \expandafter\ifblank\expandafter{\beamer@andstripped}{}{%
    \usebeamertemplate*{author}%
  }
  \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
  \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
```

```
  \vspace{0pt plus 1fil}%
  \null
}
```

B.4.3 Plain (Beamer-like) Variant

```
\defbeamertemplate{title page}{plain}{
  \null%
  \vspace{0pt plus 1fil}%
  \vfil%
  \begingroup
  \centering
  \usebeamertemplate*{title}
  \expandafter\ifblank\expandafter{\beamer@andstripped}{}{%
    \usebeamertemplate*{author}%
  }
  \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
  \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
  \ifx\insertitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
  \endgroup
  \vspace{0pt plus 1fil}%
  \null
}
```

B.4.4 Split (Two-column) Variant

```
\defbeamertemplate{title page}{split}{
  \null%
  \vspace{0pt plus 3fil}%
  \vfil%
  \begin{columns}[c,onlytextwidth]
    \begin{column}[t]{0.45\textwidth}
      \raggedright
      \ifx\insertitle\@empty\else\usebeamertemplate*{title}\fi
      \ifx\insertsubtitle\@empty\else
        \usebeamertemplate*{subtitle}
      \fi%
    \end{column}%
    \hfill%
    \usebeamertemplate*{title separator vertical}%
    \hfill%
```

```
    \begin{column}[t]{0.45\textwidth}
      \expandafter\ifblank\expandafter{\beamer@andstripped}{}{%
        \usebeamertemplate*{author}%
      }
      \ifx\insertinstitute\@empty\else
        \usebeamertemplate*{institute}
      \fi
      \ifx\insertdate\@empty\else
        \usebeamertemplate*{date}
      \fi
      \ifx\insertttitlegraphic\@empty\else
        \usebeamertemplate*{title graphic}%
      \fi%
    \end{column}
  \end{columns}
  \vfil%
  \vspace{0pt plus 1fil}%
  \null%
}
```

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 and ensure the title frame number doesn't count.

B.4.5 `\maketitle`

Inserts the title frame, or causes the current frame to use the `title page` template. Since we want to remove the footnote rule from the title page, we have to store and restore it around the call to `\frame`.

```
\def\maketitle{%
  \ifbeamer@inframe
    \titlepage
  \else
    \let\beamer@saved@footnoterule\footnoterule%
    \renewcommand\footnoterule{}%
    \frame[plain,noframenumbering]{\titlepage}%
    \let\footnoterule\beamer@saved@footnoterule%
  \fi
}
```

Also patch `\titlepage` to ensure that footnotes on the title page are formatted correctly.

B.4.6 `\titlepage`

Inserts the title page using the `title page` beamer template.

```
\def\titlepage{%
  % Apply title-page specific footnote settings
  \renewcommand{\@makefntext}[1]{%
    {\par\usebeamercolor[fg]{thanks}\usebeamerfont{thanks}$^{\@thefnmark}$##1\medskip}%
  }

  % Process the title page
  \usebeamertemplate{title page}\@thanks
}
```

B.4.7 `title graphic`

Set the title graphic. Default variant for moloch titlepage.

```
\defbeamertemplate{title graphic}{moloch}{
  \inserttitlegraphic%
  \par%
  \vspace*{1em}
}
\defbeamertemplate{title graphic}{plain}{
\vskip0.5em%
{\usebeamercolor[fg]{titlegraphic}\inserttitlegraphic\par}
}
\defbeamertemplate{title graphic}{split}{
  \vskip2em%
  \inserttitlegraphic%
  \par%
}
\setbeamertemplate{title graphic}[moloch]
```

B.4.8 `title`

Set the title on the title page. Default variant for moloch titlepage.

```
\defbeamertemplate{title}{moloch}{
  \raggedright%
  \moloch@titleformat{\inserttitle}%
  \par%
}
```

```
\defbeamertemplate{title}{plain}{
  \begin{beamercolorbox}[sep=8pt,center]{title}
    \usebeamerfont{title}\inserttitle\par%
    \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi%
  \end{beamercolorbox}%
  \vskip0.5em%
}
\defbeamertemplate{title}{split}{
  \raggedright%
  \moloch@titleformat{\inserttitle}%
  \par%
}
\setbeamertemplate{title}[moloch]
```

B.4.9 subtitle

Set the subtitle on the title page. Default variant for moloch titlepage.

```
\defbeamertemplate{subtitle}{moloch}{
  \vspace*{0.3em}
  \raggedright%
  \moloch@subtitleformat{\insertsubtitle}%
  \par%
}
\defbeamertemplate{subtitle}{split}{
  \vspace*{0.5em}
  \raggedright%
  \moloch@subtitleformat{\insertsubtitle}%
  \par%
}
\defbeamertemplate{subtitle}{plain}{%
\vskip0.25em%
{\usebeamerfont{subtitle}\usebeamercolor[fg]{subtitle}\insertsubtitle\par}%
}
\setbeamertemplate{subtitle}[moloch]
```

B.4.10 title separator

Template to set the title separator (horizontal).

```
\setbeamertemplate{title separator}{
  \tikzexternaldisable%
```

```
  \begin{tikzpicture}[baseline=(current bounding box.north)]
    \mmzUnmemoizable%
    \fill[fg] (0,0) rectangle (\textwidth, \moloch@titleseparator@linewidth);
    \useasboundingbox (0,0) rectangle (\textwidth,-\moloch@titleseparator@linewidth);
  \end{tikzpicture}%
  \tikzexternalenable%
  \par%
  \vspace*{0.8em}
}
```

B.4.11 title separator vertical

Template for a vertical title separator.

```
\setbeamertemplate{title separator vertical}{%
  {\usebeamercolor[fg]{title separator}%
      \vrule width \moloch@titleseparator@linewidth}%
}
```

B.4.12 author

Set the author on the title page. Default variant for moloch titlepage.

```
\defbeamertemplate{author}{moloch}{
  \raggedright%
  \insertauthor%
  \par%
  \vspace*{0.5em}
}
\defbeamertemplate{author}{plain}{
  \begin{beamercolorbox}[sep=8pt,center]{author}
    \usebeamerfont{author}\insertauthor
  \end{beamercolorbox}
}
\defbeamertemplate{author}{split}{
  \raggedright%
  \insertauthor%
  \par%
}
\setbeamertemplate{author}[moloch]
```

B.4.13 `institute`

Set the institute on the title page. Default variant for moloch titlepage.

```
\defbeamertemplate{institute}{moloch}{
  \insertinstitute%
  \par%
  \vspace*{1em}
}
\defbeamertemplate{institute}{plain}{
  \begin{beamercolorbox}[sep=8pt,center]{institute}
    \usebeamerfont{institute}\insertinstitute
  \end{beamercolorbox}
}
\defbeamertemplate{institute}{split}{
  \vskip0.5em%
  \raggedright%
  \insertinstitute%
  \par%
}
\setbeamertemplate{institute}[moloch]
```

B.4.14 `date`

Set the date on the title page. Default variant for moloch titlepage.

```
\defbeamertemplate{date}{moloch}{
  \insertdate%
  \par%
}
\defbeamertemplate{date}{plain}{
  \begin{beamercolorbox}[sep=8pt,center]{date}
    \usebeamerfont{date}\insertdate
  \end{beamercolorbox}
}
\defbeamertemplate{date}{split}{
  \vskip0.5em%
  \raggedright%
  \insertdate%
  \par%
}
\setbeamertemplate{date}[moloch]
```

B.5 Section Pages

B.5.1 section Page

Template for the section title slide at the beginning of each section. Divider pages use full titles, leaving short titles for navigation.

```
\defbeamertemplate{section page}{simple}{
  \begin{center}
    \usebeamercolor[fg]{section title}
    \usebeamerfont{section title}
    \moloch@sectiontitleformat{\insertsection}\par
    \usebeamercolor[fg]{subsection title}%
    \usebeamerfont{subsection title}%
    \strut%
    \ifx\insertsubsection\@empty\else%
      \insertsubsection%
    \fi
  \end{center}
  \vspace{\baselineskip - 1ex + \moloch@titleseparator@linewidth}
}
\defbeamertemplate{section page}{progressbar}{
  \centering
  \begin{minipage}{0.7875\linewidth}
    \raggedright
    \usebeamercolor[fg]{section title}
    \usebeamerfont{section title}
    \moloch@sectiontitleformat{\insertsection}\\[-0.5\baselineskip]
    \usebeamertemplate*{progress bar in section page}
    \par
    \usebeamercolor[fg]{subsection title}%
    \usebeamerfont{subsection title}%
    \strut%
    \ifx\insertsubsection\@empty\else%
      \insertsubsection%
    \fi
  \end{minipage}
  \par
}
\newcommand{\moloch@disablesectionpage}{
  \AtBeginSection{
    % intentionally empty
  }
```

```
}
\newcommand{\moloch@enablesectionpage}{
  \AtBeginSection{
    \ifbeamer@inframe
      \sectionpage
    \else
      \ifbool{moloch@enableSectionPageNumbering}
        {\frame[plain,c]{\sectionpage}}
        {\frame[plain,c,noframenumbering]{\sectionpage}}
    \fi
  }
}
```

B.5.2 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{\moloch@disablesubsectionpage}{
  \AtBeginSubsection{
    % intentionally empty
  }
}
\newcommand{\moloch@enablesubsectionpage}{
  \AtBeginSubsection{
    \ifbeamer@inframe
      \subsectionpage
    \else
      \frame[plain,c,noframenumbering]{\subsectionpage}
    \fi
  }
}
```

B.5.3 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`.

```
\setbeamertemplate{progress bar in section page}{
  \pgfmathsetlength{\moloch@progressonsectionpage}{
    \textwidth * min(1,\insertframenumber/max(1,\inserttotalframenumber))
  }%
  \tikzexternaldisable%
  \begin{tikzpicture}[baseline=(current bounding box.north)]
    \mmzUnmemoizable%
    \fill[bg]
    (0,0)
    rectangle
    (\textwidth, \moloch@progressonsectionpage@linewidth);
    \fill[fg]
    (0,0)
    rectangle
    (\moloch@progressonsectionpage,
    \moloch@progressonsectionpage@linewidth);
    \useasboundingbox (0,0) rectangle (\textwidth,-\moloch@progressonsectionpage@linewidth);
  \end{tikzpicture}%
  \tikzexternalenable%
}
```

The code above 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 `\moloch@progressonsectionpage` would exceed TeX's maximum length (16383.99999pt, roughly 5.75 metres or 18.9 feet). To avoid this, we increase the default value for `\inserttotalframenumber`; presentations with over 4000 slides will still break on first compile, but users in that situation likely have deeper problems to solve.

```
\def\inserttotalframenumber{100}
```

B.6 Lists and Floats

Moloch uses custom symbols for the `itemize` environment. The symbols are defined as below, using `pgf` commands to draw the shapes. This is slightly different than what beamer does by default, which is to use font glyphs from the math font. But we want to avoid this reliance on the math font, which may have somewhat surprising side effects.

By default, we use a filled circle for the first-level `itemize` items, a filled square for the second level, and a filled triangle for the third level. Since the triangle tapers to a point, we add a slight overhang to it so that it visually aligns better with the other symbols. We do the same for the circle, but to a lower extent.

```latex
\newcommand{\mitemover}[2]{\makebox[0pt][r]{#1}\kern-#2}
\newcommand{\mitem}[1]{\mitemover{#1}{0pt}}

\newcommand{\overhangSquare}{0pt}
\newcommand{\overhangCircle}{0.05ex}
\newcommand{\overhangTriangle}{0.25ex}

\newcommand{\moloch@circle}{
  \begin{pgfpicture}
    \pgfsetbaseline{-0.7ex}
    \pgfpathcircle{\pgfpoint{0}{0}}{0.16em}
    \pgfusepath{fill}
  \end{pgfpicture}%
}
%
\newcommand{\moloch@square}{
  \begin{pgfpicture}
    \pgfsetbaseline{-0.7ex}
    \pgfpathrectangle{\pgfpoint{-0.165em}{-0.165em}}{\pgfpoint{0.31em}{0.31em}}
    \pgfusepath{fill}
  \end{pgfpicture}%
}
%
\newcommand{\moloch@triangle}{
  \begin{pgfpicture}
    \pgfsetbaseline{-0.7ex}
    \pgfpathmoveto{\pgfpoint{0.21em}{0}}       % right vertex (tip)
    \pgfpathlineto{\pgfpoint{-0.21em}{0.21em}}  % top left
    \pgfpathlineto{\pgfpoint{-0.21em}{-0.21em}} % bottom left
    \pgfpathclose
    \pgfusepath{fill}
  \end{pgfpicture}%
}
```

Next, we set the itemize templates to use these symbols.

```latex
% \setbeamertemplate{itemize item}[circle]
\setbeamertemplate{itemize item}{\mitemover{\moloch@circle}{\overhangCircle}}
\setbeamertemplate{itemize subitem}{\mitemover{\moloch@square}{\overhangSquare}}
\setbeamertemplate{itemize subsubitem}{\mitemover{\moloch@triangle}{\overhangTriangle}}
\setbeamertemplate{caption label separator}{: }
\setbeamertemplate{caption}[numbered]
```

B.7 Footnotes

```
\setbeamertemplate{footnote}{%
  \parindent 0em\noindent\raggedright \usebeamercolor{footnote}\hbox to
  0.8em{\hfil\insertfootnotemark}\insertfootnotetext%%%
  \par%
}
```

B.8 Text and Spacing Settings

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 1fil\relax%
  \beamer@framebottomskip=0pt plus 1fil\relax%
  \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
  \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
  \def\beamer@initfirstlineunskip{}%
}
```

B.9 Standout Frames

Moloch 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}`.

B.9.1 `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 set frame options.

```
\providebool{moloch@standout}
\def\moloch@standout@style@default{default}
\def\moloch@standout@style@true{true}
\def\moloch@standout@style@alerted{alerted}
```

```
\def\moloch@standout@style@example{example}
\newcommand{\moloch@set@standout@color}[1]{%
  \def\moloch@standout@selected@style{#1}%
  \def\moloch@standout@color{standout}%
  \ifx\moloch@standout@selected@style\moloch@standout@style@default
  \else\ifx\moloch@standout@selected@style\moloch@standout@style@true
  \else\ifx\moloch@standout@selected@style\moloch@standout@style@alerted
    \def\moloch@standout@color{standout alerted}%
  \else\ifx\moloch@standout@selected@style\moloch@standout@style@example
    \def\moloch@standout@color{standout example}%
  \else
    \PackageError{beamerinnerthememoloch}{Unknown standout style '#1'}{%
      Use standout, standout=default, standout=alerted, or standout=example.%
    }%
  \fi\fi\fi\fi
}
\define@key{beamerframe}{standout}[true]{%
  \booltrue{moloch@standout}
  \begingroup
  \moloch@set@standout@color{#1}%
  \setkeys{beamerframe}{c}
  \ifbool{moloch@enableStandoutNumbering}{}{%
    \setkeys{beamerframe}{noframenumbering}}
  \ifbeamercolorempty[bg]{\moloch@standout@color}{
    \setbeamercolor{background canvas}{
      use=\moloch@standout@color,
      bg=-\moloch@standout@color.fg
    }
  }{
    \setbeamercolor{background canvas}{
      use=\moloch@standout@color,
      bg=\moloch@standout@color.bg
    }
  }
  \setbeamercolor{local structure}{
    fg=\moloch@standout@color.fg
  }
  \usebeamercolor[fg]{\moloch@standout@color}
  \setbeamercolor{page number in head/foot}{
    use=\moloch@standout@color,
    fg=\moloch@standout@color.fg
  }
  \ifbool{moloch@enableStandoutFooter}{}{\setbeamertemplate{footline}{}}
```

```
}
```

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 `\AfterEndEnvironment{frame}` for 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{moloch@standout}{
    \endgroup
    \boolfalse{moloch@standout}
  }{}
}{}{}
```

We set the fonts and the `\centering` 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}{
  \ifbool{moloch@standout}{
    \centering
    \usebeamerfont{standout}
  }{}
}
```

B.10 Process Package Options

```
\moloch@inner@setdefaults
\ProcessPgfPackageOptions{/moloch/inner}
```

C Outer Theme

> **Note**
>
> **Source file:** `src/beameroutertheme moloch.dtx`

C.1 Moloch 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.

C.2 Package Dependencies

```
\RequirePackage{calc}
\RequirePackage{pgfopts}
```

C.3 Memoization and Tikz Externalization

To avoid generating externalized figures of the progressbar we have to disable them with "tikzexternalenable" and "tikzexternaldisable". However, if the "external" library is not loaded we would get undefined control sequence problems, hence we define them as no-ops if they are not defined yet. We do the same for the "mmzUnmemoizable" command from the memoize package, in order to avoid memoization of the progress bars.

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

C.4 Options

C.4.1 progressbar

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

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

C.4.2 progressbar linewidth

Sets the linewidth of the progress bar for sectionpages and frames.

```
\newlength{\moloch@progressonsectionpage}
\newlength{\moloch@progressonsectionpage@linewidth}
\newlength{\moloch@progressinheadfoot}
\newlength{\moloch@progressinheadfoot@linewidth}
\pgfkeys{
  /moloch/outer/.cd,
  progressbarlinewidth/.code={
      \setlength{\moloch@progressonsectionpage@linewidth}{#1}
```

```
      \setlength{\moloch@progressinheadfoot@linewidth}{#1}
    },
}
```

C.4.3 progressbar aliases

Allows `progressbar linewidth` to be used in `\molochset`.

```
\pgfkeys{
  /moloch/outer/.cd,
  progressbar linewidth/.code=\pgfkeysalso{progressbarlinewidth=#1},
}
```

C.4.4 frametitle margin

Sets the margins of the frame title.

```
\pgfkeys{
  /moloch/outer/.cd,
  frametitlemarginleft/.code=\renewcommand{\moloch@frametitle@margin@left}{#1},
  frametitlemarginright/.code=\renewcommand{\moloch@frametitle@margin@right}{#1},
  frametitlemargintop/.code=\renewcommand{\moloch@frametitle@margin@top}{#1},
  frametitlemarginbottom/.code=\renewcommand{\moloch@frametitle@margin@bottom}{#1},
}
```

C.4.5 frametitle margin aliases

Allows `frametitle margin x` to be used in `\molochset`.

```
\pgfkeys{
  /moloch/outer/.cd,
  frametitle margin left/.code=\pgfkeysalso{frametitlemarginleft=#1},
  frametitle margin right/.code=\pgfkeysalso{frametitlemarginright=#1},
  frametitle margin top/.code=\pgfkeysalso{frametitlemargintop=#1},
  frametitle margin bottom/.code=\pgfkeysalso{frametitlemarginbottom=#1},
}
```

C.4.6 \moloch@outer@setdefaults

Sets default values for outer theme options.

```
\newcommand{\moloch@outer@setdefaults}{
  \pgfkeys{/moloch/outer/.cd,
    progressbar=none,
    progressbar linewidth=1pt,
  }
}
```

C.5 Deprecated Options

These options are deprecated and will be removed in a future version.

C.5.1 numbering

Adds slide numbers to the bottom right of each slide.

```
\pgfkeys{
  /moloch/outer/numbering/.cd,
  .is choice,
  none/.code={%
      \PackageWarning{moloch}{The ``numbering'' option is deprecated.
        Use beamer's ``page number in head/foot'' template instead}%
      \setbeamertemplate{page number in head/foot}[default]
    },
  counter/.code={%
      \PackageWarning{moloch}{The ``numbering'' option is deprecated.
        Use beamer's ``page number in head/foot'' template instead}%
      \setbeamertemplate{page number in head/foot}[framenumber]
    },
  fraction/.code={%
      \PackageWarning{moloch}{The ``numbering'' option is deprecated.
        Use beamer's ``page number in head/foot'' template instead}%
      \setbeamertemplate{page number in head/foot}[totalframenumber]
    },
}
```

C.6 Slide Numbering

Moloch defaults to numbering frames. To modify this, simply copy this line to your preamble and replace `framenumber`.

```
\setbeamertemplate{page number in head/foot}[framenumber]
```

C.7 Head and Footline

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

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

C.7.1 `headline`

Template for the headline at the top of each frame, which is empty by default.

```
\defbeamertemplate{headline}{plain}{}
```

C.7.2 `footline`

Template for the footline at the bottom of each frame, which includes the slide number on the right.

```
\defbeamertemplate{footline}{plain}{%
  \begin{beamercolorbox}[
      leftskip=4pt,%
      rightskip=5pt,%
      wd=\textwidth,%
    ]{footline}%
    \usebeamercolor[fg]{page number in head/foot}%
    \usebeamerfont{page number in head/foot}%
    \usebeamertemplate*{frame footer}%
    \hfill%
    \usebeamertemplate*{page number in head/foot}\vskip4pt%
  \end{beamercolorbox}%
}
```

C.8 Frametitle

C.8.1 `frametitle`

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

```
\newcommand{\moloch@frametitlestrut@start}{%
  \rule{0pt}{\moloch@frametitle@margin@top + \ht\strutbox}%
}%

\newcommand{\moloch@frametitlestrut@end}{%
  \rule[-\moloch@frametitle@margin@bottom]{0pt}{\moloch@frametitle@margin@bottom}%
}

\newcommand{\moloch@frametitle@margin@left}{\dimexpr0.85\ht\strutbox\relax}
\newcommand{\moloch@frametitle@margin@right}{\dimexpr0.85\ht\strutbox\relax}
\newcommand{\moloch@frametitle@margin@top}{\dimexpr\ht\strutbox\relax}
\newcommand{\moloch@frametitle@margin@bottom}{\dimexpr\ht\strutbox\relax}

\defbeamertemplate{frametitle}{plain}{%
  \nointerlineskip%
  \begin{beamercolorbox}[%
      wd=\paperwidth,%
      leftskip=\moloch@frametitle@margin@left,%
      rightskip=\the\glueexpr \moloch@frametitle@margin@right plus 1fil\relax,%
    ]{frametitle}%
    \usebeamerfont{frametitle}%
    \moloch@frametitlestrut@start%
    \moloch@frametitleformat{\insertframetitle}%
    {%
      \ifx\insertframesubtitle\@empty%
        \else%
        {%
          \par%
          \usebeamerfont{framesubtitle}%
          \usebeamercolor[fg]{framesubtitle}%
          \insertframesubtitle%
        }%
      \fi
    }%
    \moloch@frametitlestrut@end%
  \end{beamercolorbox}%
}
\setbeamertemplate{frametitle continuation}{%
  \romannumeral\insertcontinuationcount}
```

C.8.2 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`.

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

C.9 Process Package Options

```
\moloch@outer@setdefaults
\ProcessPgfPackageOptions{/moloch/outer}
```

D Font Theme

> **Note**
>
> **Source file:** src/beamerfontthememoloch.dtx

D.1 Moloch Font Theme

The Moloch font theme defines the fonts used throughout the presentation.

D.2 Package Dependencies

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

D.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}{size=\small}
\setbeamerfont{caption}{size=\small}
\setbeamerfont{caption name}{series=\bfseries}
\setbeamerfont{description item}{series=\bfseries}
\setbeamerfont{standout}{size=\Large, series=\bfseries}
```

D.4 Title Format Options

D.4.1 titleformat title

Controls the format of the title.

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

D.4.2 titleformat subtitle

Control the format of the subtitle.

```
\pgfkeys{
  /moloch/font/titleformat subtitle/.cd,
  .is choice,
  regular/.code={%
      \let\moloch@subtitleformat\@empty%
      \setbeamerfont{subtitle}{shape=\normalfont}%
```

```
    },
  smallcaps/.code={%
      \let\moloch@subtitleformat\@empty%
      \setbeamerfont{subtitle}{shape=\scshape}%
    },
  allsmallcaps/.code={%
      \let\moloch@subtitleformat\MakeLowercase%
      \setbeamerfont{subtitle}{shape=\scshape}%
      \PackageNote{beamerthememoloch}{%
        Be aware that titleformat subtitle=allsmallcaps
        can lead to problems}
    },
  allcaps/.code={%
      \let\moloch@subtitleformat\MakeUppercase%
      \setbeamerfont{subtitle}{shape=\normalfont}%
      \PackageNote{beamerthememoloch}{%
        Be aware that titleformat subtitle=allcaps can
        lead to problems}
    },
}
```

D.4.3 `titleformat section`

Controls the format of the section title.

```
\pgfkeys{
  /moloch/font/titleformat section/.cd,
  .is choice,
  regular/.code={%
      \let\moloch@sectiontitleformat\@empty%
      \setbeamerfont{section title}{shape=\normalfont}%
    },
  smallcaps/.code={%
      \let\moloch@sectiontitleformat\@empty%
      \setbeamerfont{section title}{shape=\scshape}%
    },
  allsmallcaps/.code={%
      \let\moloch@sectiontitleformat\MakeLowercase%
      \setbeamerfont{section title}{shape=\scshape}%
      \PackageNote{beamerthememoloch}{%
        Be aware that titleformat section=allsmallcaps
        can lead to problems}
```

```
    },
  allcaps/.code={%
      \let\moloch@sectiontitleformat\MakeUppercase%
      \setbeamerfont{section title}{shape=\normalfont}%
      \PackageNote{beamerthememoloch}{%
        Be aware that titleformat section=allcaps
        can lead to problems}
    },
}
```

D.4.4 frametitleformat

Control the format of the frame title.

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

D.4.5 `titleformat aliases`

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

```
\pgfkeys{
  /moloch/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},
}
```

D.4.6 `\moloch@font@setdefaults`

Sets default values for font theme options.

```
\newcommand{\moloch@font@setdefaults}{
  \pgfkeys{/moloch/font/.cd,
    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\moloch@titleformat#1{#1}
\def\moloch@subtitleformat#1{#1}
\def\moloch@sectiontitleformat#1{#1}
\def\moloch@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.

D.5 Process Package Options

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

E Color Theme

> **Note**
>
> **Source file:** src/beamercolorthememoloch.dtx

E.1 Moloch Color Theme

E.1.1 Package Dependencies

```
\RequirePackage{pgfopts}
```

E.1.2 Options

E.1.3 `block`

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

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

E.1.4 `Variant tracking`

Boolean to track whether we're in dark or light variant.

```
\newif\ifmoloch@variant@dark
\moloch@variant@darkfalse  % Default to light variant
```

E.1.5 colors

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

```
\pgfkeys{
  /moloch/color/background/.cd,
  .is choice,
  dark/.code={%
      \PackageWarning{beamercolorthememoloch}{%
        Option 'background' is deprecated.\MessageBreak
        Use 'colortheme variant=light/dark' instead
      }%
      \moloch@colors@dark
    },
  light/.code={%
      \moloch@colors@light
    },
}
```

E.1.6 \moloch@color@setdefaults

Sets default values for color theme options.

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

E.1.7 Base Colors

These are the base colors used in the default Moloch color theme. They are kept here mostly for backwards compatibility, and colors for the themes are now defined in the preset commands below.

```
\definecolor{mDarkBrown}{HTML}{604c38}
\definecolor{mDarkTeal}{HTML}{23373b}
\definecolor{mLightBrown}{HTML}{EB811B}
\definecolor{mLightGreen}{RGB}{0,128,128}
```

E.1.8 Hooks for Color Themes

Moloch color themes can register light and dark color schemes using the commands below. The registered colors will be stored in the macros `\moloch@define@light@colors` and `\moloch@define@dark@colors` respectively. These macros are invoked when the `variant=light` or `variant=dark` options are selected, allowing theme presets to define complete color schemes for both variants.

E.1.9 `\moloch@define@light@colors`

E.1.10 `\moloch@define@dark@colors`

Storage macros for the light and dark color definitions. Initialized empty and filled by color theme presets via the registration commands.

```
\newcommand{\moloch@define@light@colors}{}
\newcommand{\moloch@define@dark@colors}{}
```

E.1.11 `\moloch@register@light@colors`

Allows color theme presets to register their light variant color scheme. The argument should contain `\setbeamercolor` commands that define all necessary colors for the light variant.

```
\newcommand{\moloch@register@light@colors}[1]{%
  \renewcommand{\moloch@define@light@colors}{#1}%
}
```

E.1.12 `\moloch@register@dark@colors`

Allows color theme presets to register their dark variant color scheme. The argument should contain `\setbeamercolor` commands that define all necessary colors for the dark variant.

```
\newcommand{\moloch@register@dark@colors}[1]{%
  \renewcommand{\moloch@define@dark@colors}{#1}%
}
```

E.1.13 `\moloch@setup@block@colors`

Configures the colors for block environments to derive from normal text and alerted/example text colors. This is called after switching variants to ensure block colors stay consistent with the current color scheme. Block titles inherit from normal text foreground, while alerted and

example block titles additionally use their respective text colors. Backgrounds are initialized separately so color changes preserve the selected block style and any backgrounds set with `\setbeamercolor`.

```
\newcommand{\moloch@setup@block@colors}{%
  \setbeamercolor{block title}{%
    use=normal text,
    fg=normal text.fg
  }%
  \setbeamercolor{block title alerted}{%
    use={block title, alerted text}
  }%
  \setbeamercolor{block title example}{%
    use={block title, example text}
  }%
  \setbeamercolor{block body alerted}{use=block body, parent=block body}%
  \setbeamercolor{block body example}{use=block body, parent=block body}%
}
```

E.1.14 `\moloch@setup@text@colors`

Configures title-related colors to derive from normal text. This ensures that when normal text color changes (e.g., during variant switching), all title elements automatically adapt. Institute and thanks use a slightly muted version (80% mix) for visual hierarchy.

```
\newcommand{\moloch@setup@text@colors}{%
  \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, fg=normal text.fg!80!normal text.bg}%
  \setbeamercolor{structure}{use=normal text, fg=normal text.fg}%
  \setbeamercolor{thanks}{%
    use=normal text,fg=normal text.fg!80!normal text.bg}%
}
```

E.1.15 `\moloch@setup@progressbar@parents`

Updates colors that derive from the progress bar color. This includes the title separator (horizontal rule on title page) and progress bars in various locations (head, foot, section pages). Called whenever the progress bar foreground or background color is customized.

```
\newcommand{\moloch@setup@progressbar@parents}{%
  \setbeamercolor{progress bar in head/foot}{%
    use=progress bar,
    parent=progress bar
  }%
  \setbeamercolor{progress bar in section page}{%
    use=progress bar,
    parent=progress bar
  }%
}
```

E.1.16 Variant Switching Commands

E.1.17 `\moloch@colors@dark`

Switches to the dark variant color scheme. This command:

1. Sets the dark variant flag
2. Applies the registered dark colors from the current theme preset
3. Ensures normal text color is active
4. Updates dependent colors (blocks, text) to match
5. Applies any stored user customizations for the dark variant

```
\newcommand{\moloch@colors@dark}{
  \moloch@variant@darktrue
  \moloch@define@dark@colors
  \usebeamercolor[fg]{normal text}

  \moloch@setup@block@colors
  \moloch@setup@text@colors

  % Apply stored dark variant colors
  \moloch@apply@store@dark@colors
}
```

E.1.18 `\moloch@colors@light`

Switches to the light variant color scheme. Mirrors the dark variant command but applies light colors instead. The sequence ensures that user customizations override theme defaults.

```
\newcommand{\moloch@colors@light}{
  \moloch@variant@darkfalse
  \moloch@define@light@colors
  \usebeamercolor[fg]{normal text}

  \moloch@setup@block@colors
  \moloch@setup@text@colors

  \setbeamercolor{title separator}{
    use=progress bar,
    parent=progress bar
  }
  \moloch@setup@progressbar@parents

  % Apply stored light variant colors AFTER block definitions
  \moloch@apply@store@light@colors
}
```

E.1.19 `\moloch@apply@current@variant`

Helper command that applies the appropriate variant based on the `\ifmoloch@variant@dark` boolean flag. Used during initialization and when switching color theme presets to ensure the correct variant is applied.

```
\newcommand{\moloch@apply@current@variant}{%
  \ifmoloch@variant@dark
    \moloch@colors@dark
  \else
    \moloch@colors@light
  \fi
}
```

E.1.20 Color Customization Storage System

The storage system allows users to customize individual colors while preserving those customizations across variant switches. Each color option has separate storage for light and dark variants. When a user sets a color, it's both applied immediately and stored. When switching variants, stored customizations are reapplied on top of the theme defaults.

E.1.21 \moloch@apply@store@light@colors

Applies stored user customizations for the light variant. Checks each storage macro and applies it if non-empty. This is called after applying the theme's light colors to override with user preferences. For colors that affect multiple elements (like `alerted text` affecting both inline text and block titles), both are updated. Setup functions are called when needed to propagate changes to dependent colors.

```
\newcommand{\moloch@apply@store@light@colors}{%
  \ifx\moloch@store@light@alerted@text\empty\else
    \setbeamercolor{alerted text}{fg=\moloch@store@light@alerted@text}%
    \setbeamercolor{block title alerted}{fg=\moloch@store@light@alerted@text}%
  \fi
  \ifx\moloch@store@light@example@text\empty\else
    \setbeamercolor{example text}{fg=\moloch@store@light@example@text}%
    \setbeamercolor{block title example}{fg=\moloch@store@light@example@text}%
  \fi
  \ifx\moloch@store@light@frametitle@fg\empty\else
    \setbeamercolor{frametitle}{fg=\moloch@store@light@frametitle@fg}%
  \fi
  \ifx\moloch@store@light@frametitle@bg\empty\else
    \setbeamercolor{frametitle}{bg=\moloch@store@light@frametitle@bg}%
  \fi
  \ifx\moloch@store@light@progressbar@fg\empty\else
    \setbeamercolor{progress bar}{fg=\moloch@store@light@progressbar@fg}%
    \moloch@setup@progressbar@parents
  \fi
  \ifx\moloch@store@light@progressbar@bg\empty\else
    \setbeamercolor{progress bar}{bg=\moloch@store@light@progressbar@bg}%
    \moloch@setup@progressbar@parents
  \fi
  \ifx\moloch@store@light@standout@fg\empty\else
    \setbeamercolor{standout}{fg=\moloch@store@light@standout@fg}%
  \fi
  \ifx\moloch@store@light@standout@bg\empty\else
    \setbeamercolor{standout}{bg=\moloch@store@light@standout@bg}%
  \fi
  \ifx\moloch@store@light@standout@alerted@fg\empty\else
    \setbeamercolor{standout alerted}{fg=\moloch@store@light@standout@alerted@fg}%
  \fi
  \ifx\moloch@store@light@standout@alerted@bg\empty\else
    \setbeamercolor{standout alerted}{bg=\moloch@store@light@standout@alerted@bg}%
  \fi
```

```
  \ifx\moloch@store@light@standout@example@fg\empty\else
    \setbeamercolor{standout example}{fg=\moloch@store@light@standout@example@fg}%
  \fi
  \ifx\moloch@store@light@standout@example@bg\empty\else
    \setbeamercolor{standout example}{bg=\moloch@store@light@standout@example@bg}%
  \fi
  \ifx\moloch@store@light@normal@text@fg\empty\else
    \setbeamercolor{normal text}{fg=\moloch@store@light@normal@text@fg}%
    \moloch@setup@text@colors
    \moloch@setup@block@colors
  \fi
  \ifx\moloch@store@light@normal@text@bg\empty\else
    \setbeamercolor{normal text}{bg=\moloch@store@light@normal@text@bg}%
  \fi
  \ifx\moloch@store@light@title@separator\empty\else
    \setbeamercolor{title separator}{fg=\moloch@store@light@title@separator}%
  \fi
}
```

E.1.22 `\moloch@apply@store@dark@colors`

Applies stored user customizations for the dark variant. Mirrors the light variant application but for dark colors. The logic is identical: check each storage macro, apply if non-empty, and call setup functions for colors that affect dependent elements.

```
\newcommand{\moloch@apply@store@dark@colors}{%
  \ifx\moloch@store@dark@alerted@text\empty\else
    \setbeamercolor{alerted text}{fg=\moloch@store@dark@alerted@text}%
    \setbeamercolor{block title alerted}{fg=\moloch@store@dark@alerted@text}%
  \fi
  \ifx\moloch@store@dark@example@text\empty\else
    \setbeamercolor{example text}{fg=\moloch@store@dark@example@text}%
    \setbeamercolor{block title example}{fg=\moloch@store@dark@example@text}%
  \fi
  \ifx\moloch@store@dark@frametitle@fg\empty\else
    \setbeamercolor{frametitle}{fg=\moloch@store@dark@frametitle@fg}%
  \fi
  \ifx\moloch@store@dark@frametitle@bg\empty\else
    \setbeamercolor{frametitle}{bg=\moloch@store@dark@frametitle@bg}%
  \fi
  \ifx\moloch@store@dark@progressbar@fg\empty\else
    \setbeamercolor{progress bar}{fg=\moloch@store@dark@progressbar@fg}%
```

```
    \moloch@setup@progressbar@parents
  \fi
  \ifx\moloch@store@dark@progressbar@bg\empty\else
    \setbeamercolor{progress bar}{bg=\moloch@store@dark@progressbar@bg}%
    \moloch@setup@progressbar@parents
  \fi
  \ifx\moloch@store@dark@standout@fg\empty\else
    \setbeamercolor{standout}{fg=\moloch@store@dark@standout@fg}%
  \fi
  \ifx\moloch@store@dark@standout@bg\empty\else
    \setbeamercolor{standout}{bg=\moloch@store@dark@standout@bg}%
  \fi
  \ifx\moloch@store@dark@standout@alerted@fg\empty\else
    \setbeamercolor{standout alerted}{fg=\moloch@store@dark@standout@alerted@fg}%
  \fi
  \ifx\moloch@store@dark@standout@alerted@bg\empty\else
    \setbeamercolor{standout alerted}{bg=\moloch@store@dark@standout@alerted@bg}%
  \fi
  \ifx\moloch@store@dark@standout@example@fg\empty\else
    \setbeamercolor{standout example}{fg=\moloch@store@dark@standout@example@fg}%
  \fi
  \ifx\moloch@store@dark@standout@example@bg\empty\else
    \setbeamercolor{standout example}{bg=\moloch@store@dark@standout@example@bg}%
  \fi
  \ifx\moloch@store@dark@normal@text@fg\empty\else
    \setbeamercolor{normal text}{fg=\moloch@store@dark@normal@text@fg}%
    \moloch@setup@text@colors
    \moloch@setup@block@colors
  \fi
  \ifx\moloch@store@dark@normal@text@bg\empty\else
    \setbeamercolor{normal text}{bg=\moloch@store@dark@normal@text@bg}%
  \fi
  \ifx\moloch@store@dark@title@separator\empty\else
    \setbeamercolor{title separator}{fg=\moloch@store@dark@title@separator}%
  \fi
}
```

E.1.23 Semantic Color Helper

E.1.24 `\moloch@apply@semantic@colors`

Maps semantic color names to Beamer color definitions. This helper expects the following colors to be defined (via `\definecolor` or `\colorlet`) before being called:

- `mNormaltextFg` - Normal text foreground
- `mNormaltextBg` - Normal text background
- `mFrametitleFg` - Frame title foreground
- `mFrametitleBg` - Frame title background
- `mAlertFg` - Alerted text foreground
- `mExampleFg` - Example text foreground
- `mProgressbarFg` - Progress bar foreground
- `mProgressbarBg` - Progress bar background
- `mStandoutFg` - Standout frame foreground
- `mStandoutBg` - Standout frame background
- `mStandoutAlertedFg` - Alerted standout foreground
- `mStandoutAlertedBg` - Alerted standout background
- `mStandoutExampleFg` - Example standout foreground
- `mStandoutExampleBg` - Example standout background
- `mFootnoteFg` - Footnote foreground

Color themes should define all these colors (using any color specification method: HTML, RGB, color names, or color expressions), then call this helper to apply them to the appropriate Beamer color elements.

```
\newcommand{\moloch@apply@semantic@colors}{%
  % Apply semantic colors to Beamer color definitions
  \setbeamercolor{normal text}{fg=mNormaltextFg, bg=mNormaltextBg}%
  \setbeamercolor{frametitle}{fg=mFrametitleFg, bg=mFrametitleBg}%
  \setbeamercolor{palette primary}{use=frametitle, parent=frametitle}%
  \setbeamercolor{alerted text}{fg=mAlertFg}%
  \setbeamercolor{example text}{fg=mExampleFg}%
  \setbeamercolor{progress bar}{fg=mProgressbarFg, bg=mProgressbarBg}%
  \setbeamercolor{title separator}{fg=mTitleseparatorFg}%
  \setbeamercolor{footnote}{fg=mFootnoteFg}%
```

```
  \setbeamercolor{footnote mark}{fg=.}%
  \setbeamercolor{standout}{fg=mStandoutFg, bg=mStandoutBg}%
  \setbeamercolor{standout alerted}{fg=mStandoutAlertedFg, bg=mStandoutAlertedBg}%
  \setbeamercolor{standout example}{fg=mStandoutExampleFg, bg=mStandoutExampleBg}%
}
```

E.1.25 Color Theme Presets

These commands define complete color schemes that can be switched between. Each preset registers both light and dark variants.

E.1.26 `\moloch@colortheme@default`

The default moloch color scheme.

```
\newcommand{\moloch@colortheme@default}{%
  \moloch@register@light@colors{%
    \definecolor{mNormaltextFg}{HTML}{23373b}
    \colorlet{mNormaltextBg}{black!2}
    \definecolor{mAlertFg}{HTML}{EB811B}
    \definecolor{mExampleFg}{RGB}{0,128,128}
    \colorlet{mFrametitleFg}{mNormaltextBg}
    \colorlet{mFrametitleBg}{mNormaltextFg}
    \colorlet{mProgressbarFg}{mAlertFg}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!30}
    \colorlet{mTitleseparatorFg}{mProgressbarFg}
    \colorlet{mStandoutFg}{mNormaltextBg}
    \colorlet{mStandoutBg}{mNormaltextFg}
    \definecolor{mStandoutAlertedFg}{HTML}{F7F0EA}
    \definecolor{mStandoutAlertedBg}{HTML}{9E3D12}
    \definecolor{mStandoutExampleFg}{HTML}{EAF4F3}
    \definecolor{mStandoutExampleBg}{HTML}{0E4D4F}
    \colorlet{mFootnoteFg}{mNormaltextFg!90}

    \moloch@apply@semantic@colors
  }%

  % Register dark variant - use brighter colors for better contrast
  \moloch@register@dark@colors{%
    \colorlet{mNormaltextFg}{black!2}
    \definecolor{mNormaltextBg}{HTML}{23373b}
    \definecolor{mAlertFg}{HTML}{EF9F76}
```

```
    \definecolor{mExampleFg}{HTML}{5ABFB5}
    \colorlet{mFrametitleFg}{mNormaltextBg}
    \colorlet{mFrametitleBg}{black!15}
    \colorlet{mProgressbarFg}{mAlertFg}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!30}
    \colorlet{mTitleseparatorFg}{mProgressbarFg}
    \colorlet{mStandoutFg}{mNormaltextBg}
    \colorlet{mStandoutBg}{black!15}
    \definecolor{mStandoutAlertedFg}{HTML}{23373B}
    \definecolor{mStandoutAlertedBg}{HTML}{F3C4A0}
    \definecolor{mStandoutExampleFg}{HTML}{23373B}
    \definecolor{mStandoutExampleBg}{HTML}{BFE6E2}
    \colorlet{mFootnoteFg}{mNormaltextFg!90}

    \moloch@apply@semantic@colors
  }%

  % Apply current variant after registering both
  \moloch@apply@current@variant
}
```

E.1.27 \moloch@colortheme@tomorrow

The Tomorrow color scheme by Chris Kempson.

```
\newcommand{\moloch@colortheme@tomorrow}{%
  \moloch@register@light@colors{%
    \definecolor{mNormaltextFg}{HTML}{1d1f21}
    \colorlet{mNormaltextBg}{white}
    \definecolor{mAlertFg}{HTML}{cc6666}
    \definecolor{mExampleFg}{HTML}{4271ae}
    \colorlet{mFrametitleFg}{white}
    \definecolor{mFrametitleBg}{HTML}{1d1f21}
    \definecolor{mProgressbarFg}{HTML}{8959a8}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!30}
    \definecolor{mTitleseparatorFg}{HTML}{8959a8}
    \colorlet{mStandoutFg}{white}
    \definecolor{mStandoutBg}{HTML}{1d1f21}
    \definecolor{mStandoutAlertedFg}{HTML}{FFFFFF}
    \definecolor{mStandoutAlertedBg}{HTML}{A63D3D}
    \definecolor{mStandoutExampleFg}{HTML}{FFFFFF}
    \definecolor{mStandoutExampleBg}{HTML}{2E5578}
```

```
    \definecolor{mFootnoteFg}{HTML}{1d1f21}

    \moloch@apply@semantic@colors
  }

  \moloch@register@dark@colors{%
    \colorlet{mNormaltextFg}{white}
    \definecolor{mNormaltextBg}{HTML}{282a2e}
    \definecolor{mAlertFg}{HTML}{de935f}
    \definecolor{mExampleFg}{HTML}{7aa6da}
    \colorlet{mFrametitleFg}{white}
    \definecolor{mFrametitleBg}{HTML}{1d1f21}
    \definecolor{mProgressbarFg}{HTML}{b294bb}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!30}
    \definecolor{mTitleseparatorFg}{HTML}{b294bb}
    \colorlet{mStandoutFg}{white}
    \definecolor{mStandoutBg}{HTML}{1d1f21}
    \definecolor{mStandoutAlertedFg}{HTML}{1D1F21}
    \definecolor{mStandoutAlertedBg}{HTML}{CC6666}
    \definecolor{mStandoutExampleFg}{HTML}{1D1F21}
    \definecolor{mStandoutExampleBg}{HTML}{81A2BE}
    \definecolor{mFootnoteFg}{HTML}{1d1f21}

    \moloch@apply@semantic@colors
  }%

  % Apply current variant after registering both
  \moloch@apply@current@variant
}
```

E.1.28 \moloch@colortheme@paper

A crisp and clean color scheme with white frametitle background and high contrast.

```
\newcommand{\moloch@colortheme@paper}{%
  \moloch@register@light@colors{%
    \colorlet{mNormaltextFg}{black}
    \colorlet{mNormaltextBg}{white}
    \definecolor{mAlertFg}{HTML}{E5276F}
    \definecolor{mExampleFg}{HTML}{265DAB}
    \colorlet{mFrametitleFg}{black}
    \colorlet{mFrametitleBg}{white}
```

```
    \definecolor{mProgressbarFg}{HTML}{60BD68}
    \colorlet{mProgressbarBg}{black!20}
    \definecolor{mTitleseparatorFg}{HTML}{F15854}
    \colorlet{mStandoutFg}{white}
    \colorlet{mStandoutBg}{black!60}
    \definecolor{mStandoutAlertedFg}{HTML}{FFFFFF}
    \definecolor{mStandoutAlertedBg}{HTML}{B01E58}
    \definecolor{mStandoutExampleFg}{HTML}{FFFFFF}
    \definecolor{mStandoutExampleBg}{HTML}{1E4C8A}
    \colorlet{mFootnoteFg}{black!90}

    \moloch@apply@semantic@colors
  }%

  \moloch@register@dark@colors{%
    \colorlet{mNormaltextFg}{white}
    \colorlet{mNormaltextBg}{black!80}
    \definecolor{mAlertFg}{HTML}{F17CB0}
    \definecolor{mExampleFg}{HTML}{88BDE6}
    \colorlet{mFrametitleFg}{white}
    \colorlet{mFrametitleBg}{black!80}
    \definecolor{mProgressbarFg}{HTML}{90CD97}
    \colorlet{mProgressbarBg}{black!70}
    \definecolor{mTitleseparatorFg}{HTML}{F15854}
    \colorlet{mStandoutFg}{white}
    \colorlet{mStandoutBg}{black!60}
    \definecolor{mStandoutAlertedFg}{HTML}{FFFFFF}
    \definecolor{mStandoutAlertedBg}{HTML}{E5276F}
    \definecolor{mStandoutExampleFg}{HTML}{FFFFFF}
    \definecolor{mStandoutExampleBg}{HTML}{265DAB}
    \colorlet{mFootnoteFg}{black!90}

    \moloch@apply@semantic@colors
  }%

  % Apply current variant after registering both
  \moloch@apply@current@variant
}
```

E.1.29 `\moloch@colortheme@highcontrast`

High contrast color scheme for better accessibility.

```
\newcommand{\moloch@colortheme@highcontrast}{%
  % Define high contrast colors
  \definecolor{mAlert}{HTML}{AD003D}%
  \definecolor{mExample}{HTML}{005580}%

  % Register light variant
  \moloch@register@light@colors{%
    \colorlet{mNormaltextFg}{black}
    \colorlet{mNormaltextBg}{white}
    \definecolor{mAlertFg}{HTML}{C66526}
    \definecolor{mExampleFg}{HTML}{3070AD}
    \colorlet{mFrametitleFg}{white}
    \colorlet{mFrametitleBg}{black}
    \colorlet{mProgressbarFg}{mAlertFg}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!30}
    \colorlet{mTitleseparatorFg}{mAlertFg}
    \colorlet{mStandoutFg}{black}
    \colorlet{mStandoutBg}{white}
    \definecolor{mStandoutAlertedFg}{HTML}{FFFFFF}
    \definecolor{mStandoutAlertedBg}{HTML}{AD003D}
    \definecolor{mStandoutExampleFg}{HTML}{FFFFFF}
    \definecolor{mStandoutExampleBg}{HTML}{005580}
    \colorlet{mFootnoteFg}{mNormaltextFg!90}

    \moloch@apply@semantic@colors
  }%
  \definecolor{mAlertDark}{HTML}{F7768E}%
  \definecolor{mExampleDark}{HTML}{7AA2F7}%

  % Register dark variant
  \moloch@register@dark@colors{%
    \colorlet{mNormaltextFg}{white}
    \colorlet{mNormaltextBg}{black}
    \definecolor{mAlertFg}{HTML}{DCA237}
    \definecolor{mExampleFg}{HTML}{6FB2E4}
    \colorlet{mFrametitleFg}{black}
    \colorlet{mFrametitleBg}{white}
    \colorlet{mProgressbarFg}{mAlertFg}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!70}
    \colorlet{mTitleseparatorFg}{mAlertFg}
    \colorlet{mStandoutFg}{white}
    \colorlet{mStandoutBg}{black}
    \definecolor{mStandoutAlertedFg}{HTML}{FFFFFF}
```

```
    \definecolor{mStandoutAlertedBg}{HTML}{AD003D}
    \definecolor{mStandoutExampleFg}{HTML}{FFFFFF}
    \definecolor{mStandoutExampleBg}{HTML}{005580}
    \colorlet{mFootnoteFg}{mNormaltextFg!90}

    \moloch@apply@semantic@colors
  }%

  % Apply current variant after registering both
  \moloch@apply@current@variant
}
```

E.1.30 \moloch@colortheme@catppuccin

A color theme with soothing pastels and light contrasts.

```
\newcommand{\moloch@colortheme@catppuccin}{%
  \moloch@register@light@colors{%
    \definecolor{mNormaltextFg}{HTML}{4C4F69}
    \definecolor{mNormaltextBg}{HTML}{EFF1F5}
    \definecolor{mAlertFg}{HTML}{D20F39}
    \definecolor{mExampleFg}{HTML}{1E66F5}
    \definecolor{mFrametitleFg}{HTML}{CAD3F5}
    \definecolor{mFrametitleBg}{HTML}{303446}
    \definecolor{mProgressbarFg}{HTML}{DC8A78}
    \colorlet{mProgressbarBg}{mProgressbarFg!50!black!30}
    \definecolor{mTitleseparatorFg}{HTML}{40A02B}
    \definecolor{mStandoutFg}{HTML}{CAD3F5}
    \definecolor{mStandoutBg}{HTML}{303446}
    \definecolor{mStandoutAlertedFg}{HTML}{EFF1F5}
    \definecolor{mStandoutAlertedBg}{HTML}{D20F39}
    \definecolor{mStandoutExampleFg}{HTML}{EFF1F5}
    \definecolor{mStandoutExampleBg}{HTML}{1E66F5}
    \colorlet{mFootnoteFg}{mNormaltextFg!90}

    \moloch@apply@semantic@colors
  }%

  \moloch@register@dark@colors{%
    \definecolor{mNormaltextFg}{HTML}{CAD3F5}
    \definecolor{mNormaltextBg}{HTML}{24273A}
    \definecolor{mAlertFg}{HTML}{ED8796}
```

```
    \definecolor{mExampleFg}{HTML}{8AADF4}
    \definecolor{mFrametitleFg}{HTML}{CAD3F5}
    \definecolor{mFrametitleBg}{HTML}{24273A}
    \definecolor{mProgressbarFg}{HTML}{EED49F}
    \colorlet{mProgressbarBg}{mProgressbarFg!20!black!90}
    \definecolor{mTitleseparatorFg}{HTML}{A6DA95}
    \definecolor{mStandoutFg}{HTML}{4C4F69}
    \definecolor{mStandoutBg}{HTML}{CCD0DA}
    \definecolor{mStandoutAlertedFg}{HTML}{24273A}
    \definecolor{mStandoutAlertedBg}{HTML}{ED8796}
    \definecolor{mStandoutExampleFg}{HTML}{24273A}
    \definecolor{mStandoutExampleBg}{HTML}{8AADF4}
    \colorlet{mFootnoteFg}{mNormaltextFg!90}

    \moloch@apply@semantic@colors
  }%

  % Apply current variant after registering both
  \moloch@apply@current@variant
}
```

E.1.31 New Color System

The color system provides a unified interface for theme selection, variant switching, and granular color customization. It uses pgfkeys to organize options into a `/moloch/colors/` namespace.

The system supports three layers of customization:

1. Theme presets (default, tomorrow, highcontrast)
2. Variant switching (light/dark)
3. Per-variant color overrides for individual elements

All customizations are preserved when switching between variants, allowing users to set up colors once and then freely toggle between light and dark modes without losing their customizations.

E.1.32 `Color theme and variant options`

The main pgfkeys tree defines theme selection, variant switching, and the storage system for user customizations. Storage keys are defined first to declare the macros, then initialized to empty, and finally the actual color setting commands reference these storage macros.

The storage system uses a convention: for each customizable color, there are separate storage macros for light and dark variants named `\moloch@store@<variant>@<color>@<property>`. For example: `\moloch@store@light@progressbar@fg` stores the light variant's progress bar foreground color.

```
\pgfkeys{
  /moloch/colors/.cd,
  theme/.is choice,
  theme/default/.code=\moloch@colortheme@default,
  theme/tomorrow/.code=\moloch@colortheme@tomorrow,
  theme/highcontrast/.code=\moloch@colortheme@highcontrast,
  theme/paper/.code=\moloch@colortheme@paper,
  theme/catppuccin/.code=\moloch@colortheme@catppuccin,
  variant/.is choice,
  variant/light/.code={%
      \moloch@variant@darkfalse
      \moloch@colors@light
    },
  variant/dark/.code={%
      \moloch@variant@darktrue
      \moloch@colors@dark
    },
  % Storage keys - store in macros
  % These define the storage macros that will hold user customizations
  store/light/alerted text/.store in=\moloch@store@light@alerted@text,
  store/dark/alerted text/.store in=\moloch@store@dark@alerted@text,
  store/light/frametitle fg/.store in=\moloch@store@light@frametitle@fg,
  store/dark/frametitle fg/.store in=\moloch@store@dark@frametitle@fg,
  store/light/frametitle bg/.store in=\moloch@store@light@frametitle@bg,
  store/dark/frametitle bg/.store in=\moloch@store@dark@frametitle@bg,
  store/light/example text/.store in=\moloch@store@light@example@text,
  store/dark/example text/.store in=\moloch@store@dark@example@text,
  store/light/progressbar fg/.store in=\moloch@store@light@progressbar@fg,
  store/dark/progressbar fg/.store in=\moloch@store@dark@progressbar@fg,
  store/light/progressbar bg/.store in=\moloch@store@light@progressbar@bg,
  store/dark/progressbar bg/.store in=\moloch@store@dark@progressbar@bg,
  store/light/standout fg/.store in=\moloch@store@light@standout@fg,
  store/dark/standout fg/.store in=\moloch@store@dark@standout@fg,
  store/light/standout bg/.store in=\moloch@store@light@standout@bg,
  store/dark/standout bg/.store in=\moloch@store@dark@standout@bg,
  store/light/standout alerted fg/.store in=\moloch@store@light@standout@alerted@fg,
  store/dark/standout alerted fg/.store in=\moloch@store@dark@standout@alerted@fg,
  store/light/standout alerted bg/.store in=\moloch@store@light@standout@alerted@bg,
```

```
store/dark/standout alerted bg/.store in=\moloch@store@dark@standout@alerted@bg,
store/light/standout example fg/.store in=\moloch@store@light@standout@example@fg,
store/dark/standout example fg/.store in=\moloch@store@dark@standout@example@fg,
store/light/standout example bg/.store in=\moloch@store@light@standout@example@bg,
store/dark/standout example bg/.store in=\moloch@store@dark@standout@example@bg,
store/light/normal text fg/.store in=\moloch@store@light@normal@text@fg,
store/dark/normal text fg/.store in=\moloch@store@dark@normal@text@fg,
store/light/normal text bg/.store in=\moloch@store@light@normal@text@bg,
store/dark/normal text bg/.store in=\moloch@store@dark@normal@text@bg,
store/light/title separator/.store in=\moloch@store@light@title@separator,
store/dark/title separator/.store in=\moloch@store@dark@title@separator,
% Initialize storage macros as empty
% This prevents errors when checking if macros are set
store/light/alerted text=,
store/dark/alerted text=,
store/light/frametitle fg=,
store/dark/frametitle fg=,
store/light/frametitle bg=,
store/dark/frametitle bg=,
store/light/example text=,
store/dark/example text=,
store/light/progressbar fg=,
store/dark/progressbar fg=,
store/light/progressbar bg=,
store/dark/progressbar bg=,
store/light/standout fg=,
store/dark/standout fg=,
store/light/standout bg=,
store/dark/standout bg=,
store/light/standout alerted fg=,
store/dark/standout alerted fg=,
store/light/standout alerted bg=,
store/dark/standout alerted bg=,
store/light/standout example fg=,
store/dark/standout example fg=,
store/light/standout example bg=,
store/dark/standout example bg=,
store/light/normal text fg=,
store/dark/normal text fg=,
store/light/normal text bg=,
store/dark/normal text bg=,
store/light/title separator=,
store/dark/title separator=,
```

```
% Color setting commands
% Each color has three variants:
% 1. Variant-agnostic: sets current variant + stores for current
% 2. light/<color>: sets if in light mode, always stores for light
% 3. dark/<color>: sets if in dark mode, always stores for dark
progressbar fg/.code={%
    \setbeamercolor{progress bar}{fg=#1}%
    \moloch@setup@progressbar@parents
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/progressbar fg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/progressbar fg=#1}%
    \fi
  },
light/progressbar fg/.code={%
    \pgfkeys{/moloch/colors/store/light/progressbar fg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{progress bar}{fg=#1}%
      \moloch@setup@progressbar@parents
    \fi
  },
dark/progressbar fg/.code={%
    \pgfkeys{/moloch/colors/store/dark/progressbar fg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{progress bar}{fg=#1}%
      \moloch@setup@progressbar@parents
    \fi
  },
progressbar bg/.code={%
    \setbeamercolor{progress bar}{bg=#1}%
    \moloch@setup@progressbar@parents
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/progressbar bg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/progressbar bg=#1}%
    \fi
  },
light/progressbar bg/.code={%
    \pgfkeys{/moloch/colors/store/light/progressbar bg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{progress bar}{bg=#1}%
      \moloch@setup@progressbar@parents
    \fi
```

```
  },
dark/progressbar bg/.code={%
    \pgfkeys{/moloch/colors/store/dark/progressbar bg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{progress bar}{bg=#1}%
      \moloch@setup@progressbar@parents
    \fi
  },
normal text fg/.code={%
    \setbeamercolor{normal text}{fg=#1}%
    \moloch@setup@text@colors
    \moloch@setup@block@colors
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/normal text fg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/normal text fg=#1}%
    \fi
  },
light/normal text fg/.code={%
    \pgfkeys{/moloch/colors/store/light/normal text fg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{normal text}{fg=#1}%
      \moloch@setup@text@colors
      \moloch@setup@block@colors
    \fi
  },
dark/normal text fg/.code={%
    \pgfkeys{/moloch/colors/store/dark/normal text fg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{normal text}{fg=#1}%
      \moloch@setup@text@colors
      \moloch@setup@block@colors
    \fi
  },
normal text bg/.code={%
    \setbeamercolor{normal text}{bg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/normal text bg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/normal text bg=#1}%
    \fi
  },
light/normal text bg/.code={%
```

```
    \pgfkeys{/moloch/colors/store/light/normal text bg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{normal text}{bg=#1}%
    \fi
  },
dark/normal text bg/.code={%
    \pgfkeys{/moloch/colors/store/dark/normal text bg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{normal text}{bg=#1}%
    \fi
  },
alerted text/.code={%
    \setbeamercolor{alerted text}{fg=#1}%
    \setbeamercolor{block title alerted}{fg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/alerted text=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/alerted text=#1}%
    \fi
  },
light/alerted text/.code={%
    \pgfkeys{/moloch/colors/store/light/alerted text=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{alerted text}{fg=#1}%
      \setbeamercolor{block title alerted}{fg=#1}%
    \fi
  },
dark/alerted text/.code={%
    \pgfkeys{/moloch/colors/store/dark/alerted text=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{alerted text}{fg=#1}%
      \setbeamercolor{block title alerted}{fg=#1}%
    \fi
  },
example text/.code={%
    \setbeamercolor{example text}{fg=#1}%
    \setbeamercolor{block title example}{fg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/example text=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/example text=#1}%
    \fi
  },
```

```
light/example text/.code={%
    \pgfkeys{/moloch/colors/store/light/example text=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{example text}{fg=#1}%
      \setbeamercolor{block title example}{fg=#1}%
    \fi
  },
dark/example text/.code={%
    \pgfkeys{/moloch/colors/store/dark/example text=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{example text}{fg=#1}%
      \setbeamercolor{block title example}{fg=#1}%
    \fi
  },
frametitle fg/.code={%
    \setbeamercolor{frametitle}{fg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/frametitle fg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/frametitle fg=#1}%
    \fi
  },
frametitle bg/.code={%
    \setbeamercolor{frametitle}{bg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/frametitle bg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/frametitle bg=#1}%
    \fi
  },
light/frametitle fg/.code={%
    \pgfkeys{/moloch/colors/store/light/frametitle fg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{frametitle}{fg=#1}%
    \fi
  },
dark/frametitle fg/.code={%
    \pgfkeys{/moloch/colors/store/dark/frametitle fg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{frametitle}{fg=#1}%
    \fi
  },
light/frametitle bg/.code={%
```

```
    \pgfkeys{/moloch/colors/store/light/frametitle bg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{frametitle}{bg=#1}%
    \fi
  },
dark/frametitle bg/.code={%
    \pgfkeys{/moloch/colors/store/dark/frametitle bg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{frametitle}{bg=#1}%
    \fi
  },
standout fg/.code={%
    \setbeamercolor{standout}{fg=#1}%
  },
light/standout fg/.code={%
    \pgfkeys{/moloch/colors/store/light/standout fg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{standout}{fg=#1}%
    \fi
  },
dark/standout fg/.code={%
    \pgfkeys{/moloch/colors/store/dark/standout fg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{standout}{fg=#1}%
    \fi
  },
standout bg/.code={%
    \setbeamercolor{standout}{bg=#1}%
  },
light/standout bg/.code={%
    \pgfkeys{/moloch/colors/store/light/standout bg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{standout}{bg=#1}%
    \fi
  },
dark/standout bg/.code={%
    \pgfkeys{/moloch/colors/store/dark/standout bg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{standout}{bg=#1}%
    \fi
  },
standout alerted fg/.code={%
    \setbeamercolor{standout alerted}{fg=#1}%
```

```
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/standout alerted fg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/standout alerted fg=#1}%
    \fi
  },
light/standout alerted fg/.code={%
    \pgfkeys{/moloch/colors/store/light/standout alerted fg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{standout alerted}{fg=#1}%
    \fi
  },
dark/standout alerted fg/.code={%
    \pgfkeys{/moloch/colors/store/dark/standout alerted fg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{standout alerted}{fg=#1}%
    \fi
  },
standout alerted bg/.code={%
    \setbeamercolor{standout alerted}{bg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/standout alerted bg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/standout alerted bg=#1}%
    \fi
  },
light/standout alerted bg/.code={%
    \pgfkeys{/moloch/colors/store/light/standout alerted bg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{standout alerted}{bg=#1}%
    \fi
  },
dark/standout alerted bg/.code={%
    \pgfkeys{/moloch/colors/store/dark/standout alerted bg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{standout alerted}{bg=#1}%
    \fi
  },
standout example fg/.code={%
    \setbeamercolor{standout example}{fg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/standout example fg=#1}%
    \else
```

```
      \pgfkeys{/moloch/colors/store/light/standout example fg=#1}%
    \fi
  },
light/standout example fg/.code={%
    \pgfkeys{/moloch/colors/store/light/standout example fg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{standout example}{fg=#1}%
    \fi
  },
dark/standout example fg/.code={%
    \pgfkeys{/moloch/colors/store/dark/standout example fg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{standout example}{fg=#1}%
    \fi
  },
standout example bg/.code={%
    \setbeamercolor{standout example}{bg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/standout example bg=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/standout example bg=#1}%
    \fi
  },
light/standout example bg/.code={%
    \pgfkeys{/moloch/colors/store/light/standout example bg=#1}%
    \ifmoloch@variant@dark\else
      \setbeamercolor{standout example}{bg=#1}%
    \fi
  },
dark/standout example bg/.code={%
    \pgfkeys{/moloch/colors/store/dark/standout example bg=#1}%
    \ifmoloch@variant@dark
      \setbeamercolor{standout example}{bg=#1}%
    \fi
  },
title separator/.code={%
    \setbeamercolor{title separator}{fg=#1}%
    \ifmoloch@variant@dark
      \pgfkeys{/moloch/colors/store/dark/title separator=#1}%
    \else
      \pgfkeys{/moloch/colors/store/light/title separator=#1}%
    \fi
  },
```

```
  light/title separator/.code={%
      \pgfkeys{/moloch/colors/store/light/title separator=#1}%
      \ifmoloch@variant@dark\else
        \setbeamercolor{title separator}{fg=#1}%
      \fi
    },
  dark/title separator/.code={%
      \pgfkeys{/moloch/colors/store/dark/title separator=#1}%
      \ifmoloch@variant@dark
        \setbeamercolor{title separator}{fg=#1}%
      \fi
    },
}
```

E.1.33 \molochcolors

User-facing command for color customization.

```
\newcommand{\molochcolors}[1]{\pgfkeys{/moloch/colors/.cd,#1}}
```

The Moloch 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=alerted text,
  fg=alerted text.fg,
  bg=alerted text.fg!50!black!30
}
\setbeamercolor{title separator}{
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar in head/foot}{%
  use=progress bar,
  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.

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

E.1.34 Process Package Options

```
\moloch@colortheme@default
\moloch@color@setdefaults
\ProcessPgfPackageOptions{/moloch/color}

\mode<all>
```

F Color Theme: Tomorrow

> **Note**
>
> **Source file:** src/beamercolorthememoloch-tomorrow.dtx

F.1 Moloch Tomorrow Color Theme

DEPRECATED: This color theme is deprecated and will be removed in a future version. Please use |verb| with |verb| instead.

The Moloch Tomorrow color theme provides a color scheme based on the "Tomorrow" color scheme created by Chris Kempson.

This file now simply loads the base color theme and applies the tomorrow preset.

```
\PackageWarning{beamercolorthememoloch-tomorrow}{%
  This color theme is deprecated and will be removed in a future version.\MessageBreak
  Please use \string\usecolortheme{moloch} with \string\molochset{colortheme=tomorrow} instea
}

\usecolortheme{moloch}

% Apply tomorrow theme with light variant (default behavior)
\moloch@colortheme@tomorrow
\moloch@colors@light

\mode<all>
```

G Color Theme: High Contrast

> **Note**
>
> **Source file:** src/beamercolorthememoloch-highcontrast.dtx

DEPRECATED: This color theme is deprecated and will be removed in a future version. Please use `\usecolortheme{moloch}` with `\molochset{colortheme=highcontrast}` instead.

This is a high-contrast variant of the Moloch color theme. It provides better readability for users with visual impairments by using stark color contrasts.

This file now simply loads the base color theme and applies the highcontrast preset.

```
\PackageWarning{beamercolorthememoloch-highcontrast}{%
  This color theme is deprecated and will be removed in a future version.\MessageBreak
  Please use \string\usecolortheme{moloch} with \string\molochset{colortheme=highcontrast} in
}

\usecolortheme{moloch}

% Apply highcontrast theme with light variant (default behavior)
\moloch@colortheme@highcontrast
\moloch@colors@light

\mode<all>
```

H ltx-talk Theme

> **Note**
>
> **Source file:** src/moloch-talk.dtx

H.1 Moloch for ltx-talk

This package brings the Moloch design to ltx-talk, the experimental presentation class by Joseph Wright. Unlike Beamer, ltx-talk has no theme mechanism: its author states explicitly that themes are not part of the ltx-talk approach, and that appearance is instead adjusted through the LaTeX3 template layer. This package is therefore an ordinary package that a document loads after the class:

```
\DocumentMetadata{lang=en-US,tagging=on}
\documentclass{ltx-talk}
\usepackage{moloch-talk}
```

Both the class and this package are experimental. ltx-talk is a 0.x release whose author warns that "changes may occur to almost all interfaces", so this package may need to follow those changes.

H.2 Class Guard

There is no sensible way to recover if the class is missing, and the errors that would follow (undefined templates and instances) are unhelpful, so fail early with a clear message.

```
\msg_new:nnnn { moloch-talk } { wrong-class }
  { The~'moloch-talk'~package~requires~the~'ltx-talk'~class. }
  {
    Load~the~class~first:~\iow_char:N\\documentclass{ltx-talk}.~
    For~Beamer~presentations~use~\iow_char:N\\usetheme{moloch}~instead.
  }
\cs_if_exist:NF \ps@talk
  { \msg_fatal:nn { moloch-talk } { wrong-class } }
```

H.3 Colors

The color names are the same as in the Beamer theme, so that recipes carry over between the two and the two ports stay conceptually in sync. Only the default color theme is provided; the alternative themes of the Beamer version (`tomorrow`, `catppuccin`, and so on) are not ported yet.

H.3.1 `\__moloch_colors_light:`

The light variant, transcribed from the Beamer color theme.

```
\cs_new_protected:Npn \__moloch_colors_light:
  {
    \definecolor { mNormaltextFg } { HTML } { 23373B }
    \colorlet    { mNormaltextBg } { black!2 }
    \definecolor { mAlertFg }      { HTML } { EB811B }
    \definecolor { mExampleFg }    { RGB }  { 0 , 128 , 128 }
    \colorlet    { mFrametitleFg } { mNormaltextBg }
    \colorlet    { mFrametitleBg } { mNormaltextFg }
    \colorlet    { mProgressbarFg } { mAlertFg }
    \colorlet    { mProgressbarBg } { mProgressbarFg!50!black!30 }
    \colorlet    { mTitleseparatorFg } { mProgressbarFg }
    \colorlet    { mStandoutFg }   { mNormaltextBg }
    \colorlet    { mStandoutBg }   { mNormaltextFg }
    \colorlet    { mFootnoteFg }   { mNormaltextFg!90 }
  }
```

H.3.2 `\__moloch_colors_dark:`

The dark variant. Note that it is not simply the light variant inverted: the alert and example colors are lightened for contrast against the dark background, exactly as in the Beamer theme.

```
\cs_new_protected:Npn \__moloch_colors_dark:
  {
    \colorlet    { mNormaltextFg } { black!2 }
    \definecolor { mNormaltextBg } { HTML } { 23373B }
    \definecolor { mAlertFg }      { HTML } { EF9F76 }
    \definecolor { mExampleFg }    { HTML } { 5ABFB5 }
    \colorlet    { mFrametitleFg } { mNormaltextBg }
    \colorlet    { mFrametitleBg } { black!15 }
    \colorlet    { mProgressbarFg } { mAlertFg }
    \colorlet    { mProgressbarBg } { mProgressbarFg!50!black!30 }
```

```
    \colorlet    { mTitleseparatorFg } { mProgressbarFg }
    \colorlet    { mStandoutFg }   { mNormaltextBg }
    \colorlet    { mStandoutBg }   { black!15 }
    \colorlet    { mFootnoteFg }   { mNormaltextFg!90 }
  }
```

H.3.3 `\__moloch_colors_apply:`

The class uses three semantic colors. Mapping `structure` to the normal text color reproduces Moloch's convention that block titles, list labels and table-of-contents entries are simply the body color in bold, rather than a separate accent color.

Selecting the body color again also updates the current color after a variant change. Redefining its name alone leaves the previously selected color active, which would make dark text disappear on a dark background. Inside a standout, both the current color and the structure color must instead follow the standout foreground to retain contrast.

```
\bool_new:N \l__moloch_standout_bool
\cs_new_protected:Npn \__moloch_colors_apply:
  {
    \bool_if:NTF \l__moloch_standout_bool
      {
        \DeclareColor { structure } { mStandoutFg }
        \color_select:n { mStandoutFg }
      }
      {
        \DeclareColor { structure } { mNormaltextFg }
        \color_select:n { mNormaltextFg }
      }
    \DeclareColor { alert }     { mAlertFg }
    \DeclareColor { example }   { mExampleFg }
  }
```

H.4 Options

The option surface is deliberately much smaller than the Beamer theme's. Anything not covered here is adjusted with the class's own interfaces, and .

```
\tl_new:N   \l__moloch_progressbar_tl
\tl_new:N   \l__moloch_sectionpage_tl
\dim_new:N  \l__moloch_progressbar_dim
\dim_new:N  \l__moloch_titlerule_dim
```

```
\bool_new:N \l__moloch_framenumber_bool

\keys_define:nn { moloch }
  {
    variant .choice: ,
    variant / light .code:n =
      { \__moloch_colors_light: \__moloch_colors_apply: } ,
    variant / dark .code:n =
      { \__moloch_colors_dark: \__moloch_colors_apply: } ,
    progressbar .choices:nn =
      { none , head , frametitle , foot }
      { \tl_set_eq:NN \l__moloch_progressbar_tl \l_keys_choice_tl } ,
    progressbar-linewidth .dim_set:N = \l__moloch_progressbar_dim ,
    sectionpage .choices:nn =
      { none , simple , progressbar }
      { \tl_set_eq:NN \l__moloch_sectionpage_tl \l_keys_choice_tl } ,
    titleseparator-linewidth .dim_set:N = \l__moloch_titlerule_dim ,
    framenumber .bool_set:N = \l__moloch_framenumber_bool
  }
```

The defaults match the Beamer theme: no progress bar, section pages with a progress bar, a 1pt bar and a 0.5pt title separator.

```
\keys_set:nn { moloch }
  {
    variant = light ,
    progressbar = none ,
    progressbar-linewidth = 1pt ,
    sectionpage = progressbar ,
    titleseparator-linewidth = 0.5pt ,
    framenumber = true
  }
\ProcessKeyOptions [ moloch ]
```

H.4.1 \molochset

Options can also be set after loading. Changing variants selects the new body color. Inside a frame, the palette and page options are saved globally because the class closes the body's typesetting groups before drawing the header, footer, and background. The frame and slide numbers limit those saved settings to the pages produced by the current body.

Nested groups must still restore their enclosing settings. The class runs its tagging setup in the box surrounding the body, immediately before opening the body's box and color groups.

Recording the depth of the inner group lets us refresh the saved settings as nested groups close, while keeping the final body settings available after the body itself closes.

```
\int_new:N \l__moloch_body_group_int
\cs_new_eq:NN \__moloch_slide_tag_on_begin: \__talk_slide_tag_on_begin:
\cs_set_protected:Npn \__talk_slide_tag_on_begin:
  {
    \__moloch_slide_tag_on_begin:
    \int_set:Nn \l__moloch_body_group_int
      { \tex_currentgrouplevel:D + 2 }
  }
\cs_new_eq:NN \__moloch_slide_tag_off_begin: \__talk_slide_tag_off_begin:
\cs_set_protected:Npn \__talk_slide_tag_off_begin:
  {
    \__moloch_slide_tag_off_begin:
    \int_set:Nn \l__moloch_body_group_int
      { \tex_currentgrouplevel:D + 2 }
  }
\int_new:N \g__moloch_settings_frame_int
\int_new:N \g__moloch_settings_slide_int
\int_gset:Nn \g__moloch_settings_frame_int { -1 }
\tl_new:N \g__moloch_frame_settings_tl
\tl_new:N \l__moloch_color_spec_tl
\cs_new_protected:Npn \__moloch_frame_settings_save:
  {
    \int_gset:Nn \g__moloch_settings_frame_int { \value { frame } }
    \int_gset:Nn \g__moloch_settings_slide_int { \value { slide } }
    \tl_gset:Ne \g__moloch_frame_settings_tl
      {
        \exp_not:N \keys_set:nn { moloch }
          {
            progressbar = \exp_not:V \l__moloch_progressbar_tl ,
            progressbar-linewidth = \dim_use:N \l__moloch_progressbar_dim ,
            framenumber = \bool_if:NTF \l__moloch_framenumber_bool
              { true } { false }
          }
      }
    \clist_map_inline:nn
      {
        mNormaltextFg , mNormaltextBg , mAlertFg , mExampleFg ,
        mFrametitleFg , mFrametitleBg , mProgressbarFg , mProgressbarBg ,
        mTitleseparatorFg , mStandoutFg , mStandoutBg , mFootnoteFg ,
        structure , alert , example
```

```
      }
      {
        \extractcolorspec {##1} \l__moloch_color_spec_tl
        \tl_gput_right:Ne \g__moloch_frame_settings_tl
          { \exp_not:N \definecolor {##1} \exp_not:V \l__moloch_color_spec_tl }
      }
    \int_compare:nNnT
      { \tex_currentgrouplevel:D } > \l__moloch_body_group_int
      { \group_insert_after:N \__moloch_frame_settings_save: }
  }
\NewDocumentCommand \molochset { m }
  {
    \keys_set:nn { moloch } {#1}
    \bool_if:NT \l__talk_frame_bool { \__moloch_frame_settings_save: }
  }
```

H.5 Frame Title Band

Moloch's frame title sits in a band that spans the full paper width, with the title one strut height above the band's lower edge and two above its baseline — three strut heights in total, measured in the frame title font.

The class already draws a full-width band for the running header, sized from the `geometry` package's top margin and header separation. So rather than rebuilding the header template, the proportions are obtained by setting those two lengths: `geometry`'s `tmargin` measures from the top of the paper to the top of the body text, which is exactly the band's visible height.

```
\dim_new:N \g__moloch_band_dim
\dim_new:N \g__moloch_headsep_dim
\group_begin:
  \large
  \dim_gset:Nn \g__moloch_band_dim { 3 \ht \strutbox }
  \dim_gset:Nn \g__moloch_headsep_dim { \ht \strutbox }
\group_end:
```

The frame title also hangs closer to the edge of the paper than the body text does. This is one of the places where the class's default differs from Moloch: it aligns the frame title with the body text instead. The measure is taken in the body font, as in the Beamer theme.

```
\dim_new:N \g__moloch_hmargin_dim
\group_begin:
  \normalsize
  \dim_gset:Nn \g__moloch_hmargin_dim { 0.85 \ht \strutbox }
```

```
\group_end:
```

The footer sits lower than the class puts it, so that the frame number ends up on the same line as in the Beamer theme, whose footer leaves 4pt below the number. The distance from the bottom of the body text to the footer baseline is what `geometry` calls the foot skip.

```
\dim_const:Nn \c__moloch_footer_baseline_dim { 4pt }
\dim_new:N \g__moloch_footskip_dim
\dim_gset:Nn \g__moloch_footskip_dim
  { \Gm@bmargin - \c__moloch_footer_baseline_dim }
```

The header box also has to be tall enough to hold the band, or every frame reports an overfull box while the output routine is active.

```
\use:e
  {
    \exp_not:N \geometry
      {
        tmargin = \dim_use:N \g__moloch_band_dim ,
        headheight = \dim_use:N \g__moloch_band_dim ,
        headsep = \dim_use:N \g__moloch_headsep_dim ,
        footskip = \dim_use:N \g__moloch_footskip_dim
      }
  }
```

With the geometry in place only the colors and the font need changing. The header's `height` key is left at its default, which is written in terms of the `geometry` lengths and so follows the change above.

```
\EditInstance { header } { std }
  {
    background-color = mFrametitleBg ,
    color = mFrametitleFg ,
    left-hspace = \g__moloch_hmargin_dim
  }
\EditInstance { frametitle } { header }
  { font = \large \bfseries }
```

H.6 Rules and the Progress Bar

H.6.1 `\__moloch_rule:nnn`

A rule given as width, height and depth. The class has an equivalent helper, but it is internal to the class, so this package provides its own.

```
\dim_new:N \l__moloch_rule_wd_dim
\dim_new:N \l__moloch_rule_ht_dim
\dim_new:N \l__moloch_rule_dp_dim
\cs_new_protected:Npn \__moloch_rule:nnn #1#2#3
  {
    \group_begin:
      \dim_set:Nn \l__moloch_rule_wd_dim { #1 }
      \dim_set:Nn \l__moloch_rule_dp_dim { 0pt - ( #3 ) }
      \dim_set:Nn \l__moloch_rule_ht_dim { ( #2 ) + ( #3 ) }
      \rule
        [ \l__moloch_rule_dp_dim ]
        { \l__moloch_rule_wd_dim }
        { \l__moloch_rule_ht_dim }
    \group_end:
  }
```

H.6.2 __moloch_progress_frac:

How far through the presentation we are, as a number between 0 and 1. The fallback of 100 total frames keeps the first compilation, when the total is not yet known, from producing an over-long bar; the Beamer theme uses the same guard.

```
\cs_new:Npn \__moloch_progress_frac:
  {
    \fp_eval:n
      {
        min
          (
            1 ,
            \int_use:N \c@frame /
            max ( 1 , \property_ref:nnn { lastpage } { totalframes } { 100 } )
          )
      }
  }
```

H.6.3 __moloch_progress_width:n

The filled portion of a bar of the given width. The fraction cannot simply be placed in front of the width, because that is only valid when the width is a length register rather than an expression.

```
\cs_new:Npn \__moloch_progress_width:n #1
  {
    \fp_to_dim:n
      { \__moloch_progress_frac: * \dim_to_fp:n { #1 } }
  }
```

H.6.4 __moloch_progressbar:nn

The bar itself, given a height and a depth relative to the current baseline. It is drawn as an overlapping box of zero width, so that it can be placed at the start of the header or footer without disturbing what follows.

Drawing the bar here, rather than in a `shipout/background` hook, is what makes it obey the page style: a background hook cannot tell a plain page from an ordinary one, and the bar would then also appear on the title page, on section pages and on standout frames.

```
\cs_new_protected:Npn \__moloch_progressbar:nn #1#2
  {
    \hbox_overlap_right:n
      {
        \skip_horizontal:n { -\Gm@lmargin }
        \group_begin:
          \color_select:n { mProgressbarBg }
          \__moloch_rule:nnn { \paperwidth } {#1} {#2}
        \group_end:
        \skip_horizontal:n { -\paperwidth }
        \group_begin:
          \color_select:n { mProgressbarFg }
          \__moloch_rule:nnn
            { \__moloch_progress_width:n { \paperwidth } } {#1} {#2}
        \group_end:
      }
  }
```

H.6.5 __moloch_progressbar_head:

H.6.6 __moloch_progressbar_frametitle:

H.6.7 __moloch_progressbar_foot:

The three positions. The header baseline sits one header separation above the lower edge of the band, so a negative height places the bar exactly at that edge for the `frametitle` position. The

top of the page is therefore one header separation closer to the baseline than the top margin.

```
\cs_new_protected:Npn \__moloch_progressbar_head:
  {
    \__moloch_progressbar:nn
      { \Gm@tmargin - \headsep }
      { \l__moloch_progressbar_dim - \Gm@tmargin + \headsep }
  }
\cs_new_protected:Npn \__moloch_progressbar_frametitle:
  {
    \__moloch_progressbar:nn
      { -\headsep }
      { \headsep + \l__moloch_progressbar_dim }
  }
\cs_new_protected:Npn \__moloch_progressbar_foot:
  {
    \__moloch_progressbar:nn
      { \l__moloch_progressbar_dim - \Gm@bmargin + \footskip }
      { \Gm@bmargin - \footskip }
  }
```

H.6.8 __moloch_progress_rule:n

A horizontal progress bar of a given width, for use on section pages, where it is part of the text rather than part of the page furniture. It hangs below the baseline rather than sitting on it, as the Beamer theme's does.

```
\cs_new_protected:Npn \__moloch_progress_rule:n #1
  {
    \leavevmode
    \hbox_overlap_right:n
      {
        \group_begin:
          \color_select:n { mProgressbarBg }
          \__moloch_rule:nnn {#1} { 0pt } { \l__moloch_progressbar_dim }
        \group_end:
      }
    \hbox_overlap_right:n
      {
        \group_begin:
          \color_select:n { mProgressbarFg }
          \__moloch_rule:nnn
            { \__moloch_progress_width:n {#1} }
```

```
            { 0pt }
            { \l__moloch_progressbar_dim }
        \group_end:
      }
    \rule {#1} { 0pt }
  }
```

H.7 Page Style

A separate footer instance keeps the class's own footer available through {talk}. It uses the class's footer elements and short metadata, but places a final frame number at the right edge without the trailing glue that the class's footer template adds.

Both edges sit closer to the paper edge than the body text does, as in the Beamer theme. The skips outside the box keep its total width equal to the text width, so the output routine can place it normally.

```
\tl_new:N \l__moloch_footer_color_tl
\tl_new:N \l__moloch_footer_font_tl
\tl_new:N \l__moloch_footer_separator_tl
\tl_new:N \l__moloch_footer_metadata_tl
\clist_new:N \l__moloch_footer_order_clist
\seq_new:N \l__moloch_footer_elements_seq
\dim_new:N \l__moloch_footer_left_dim
\dim_new:N \l__moloch_footer_right_dim
\DeclareTemplateInterface { footer } { moloch } { 0 }
  {
    color         : tokenlist = mFootnoteFg ,
    font          : tokenlist = \tiny ,
    element-order : commalist = { framenumber } ,
    left-hspace   : length = 4pt ,
    right-hspace  : length = 5pt ,
    separator     : tokenlist = \hfil
  }
\cs_new:Npn \__moloch_footer_metadata:n #1
  {
    \cs_if_exist:cTF { @ short #1 }
      { @ short #1 }
      { @ #1 }
  }
\cs_new_protected:Npn \__moloch_footer_element:n #1
  {
```

```
    \tl_set:Ne \l__moloch_footer_metadata_tl
      { \__moloch_footer_metadata:n {#1} }
    \ExpandArgs { nnv } \UseInstance { footer-element } {#1}
      { \l__moloch_footer_metadata_tl }
  }
\DeclareTemplateCode { footer } { moloch } { 0 }
  {
    color         = \l__moloch_footer_color_tl ,
    font          = \l__moloch_footer_font_tl ,
    element-order = \l__moloch_footer_order_clist ,
    left-hspace   = \l__moloch_footer_left_dim ,
    right-hspace  = \l__moloch_footer_right_dim ,
    separator     = \l__moloch_footer_separator_tl
  }
  {
    \seq_clear:N \l__moloch_footer_elements_seq
    \clist_map_inline:Nn \l__moloch_footer_order_clist
      {
        \bool_lazy_or:nnT
          { ! \str_if_eq_p:nn {##1} { framenumber } }
          { \l__moloch_framenumber_bool }
          {
            \tl_if_empty:cF { \__moloch_footer_metadata:n {##1} }
              { \seq_put_right:Nn \l__moloch_footer_elements_seq {##1} }
          }
      }
    \skip_horizontal:n { \l__moloch_footer_left_dim - \Gm@lmargin }
    \hbox_to_wd:nn
      { \paperwidth - \l__moloch_footer_left_dim - \l__moloch_footer_right_dim }
      {
        \tl_if_empty:NF \l__moloch_footer_color_tl
          { \color_select:V \l__moloch_footer_color_tl }
        \l__moloch_footer_font_tl
        \seq_map_indexed_inline:Nn \l__moloch_footer_elements_seq
          {
            \int_compare:nNnTF {##1} > { 1 }
              { \l__moloch_footer_separator_tl }
              { \str_if_eq:nnT {##2} { framenumber } { \hfil } }
            \__moloch_footer_element:n {##2}
          }
        \str_if_eq:eeF
          { \seq_item:Nn \l__moloch_footer_elements_seq { -1 } }
          { framenumber }
```

```
          { \hfil }
      }
    \skip_horizontal:n { \l__moloch_footer_right_dim - \Gm@rmargin }
  }
\DeclareInstance { footer } { moloch } { moloch } { }
```

The header is drawn in an overlapping box so that the progress bar can follow it at the same horizontal position. This keeps the header's background from covering a bar placed at the top of the page.

```
\cs_new_nopar:Npn \ps@moloch
  {
    \cs_set_nopar:Npn \@oddhead
      {
        \hbox_overlap_right:n { \UseInstance { header } { std } }
        \str_case:Vn \l__moloch_progressbar_tl
          {
            { head }       { \__moloch_progressbar_head: }
            { frametitle } { \__moloch_progressbar_frametitle: }
          }
        \hfil
      }
    \cs_set_nopar:Npn \@oddfoot
      {
        \str_if_eq:VnT \l__moloch_progressbar_tl { foot }
          { \__moloch_progressbar_foot: }
        \UseInstance { footer } { moloch }
      }
    \cs_set_eq:NN \@evenhead \@oddhead
    \cs_set_eq:NN \@evenfoot \@oddfoot
  }
\pagestyle { moloch }
```

H.8 Lists

Draw the same circle, square, and triangle as the Beamer theme so that the markers do not depend on font glyphs. The class's list instances use the standard label commands, which keeps explicit item labels and the class's tagging and overlay handling intact. The fourth level retains its class default, since Beamer only supports three levels.

The circle and triangle overhang slightly to align their visible edges with the square, just as they do in the Beamer theme.

The class selects colors through the LaTeX3 color layer, while PGF reads xcolor's current color. Export the current label color explicitly so the drawn markers also follow dark variants and custom label colors.

```
\ExplSyntaxOff
\RequirePackage{pgf}
\ExplSyntaxOn

\tl_new:N \l__moloch_itemize_color_tl
\cs_new_protected:Npn \__moloch_itemize_marker:nn #1#2
  {
    \hbox_overlap_left:n
      {
        \color_export:nnN { . } { comma-sep-rgb } \l__moloch_itemize_color_tl
        \exp_args:NnnV \definecolor { molochItemizeColor } { rgb }
          \l__moloch_itemize_color_tl
        #1
      }
    \kern -#2 \scan_stop:
  }
\cs_new_protected:Npn \__moloch_itemize_circle:
  {
    \begin{pgfpicture}
      \pgfsetfillcolor { molochItemizeColor }
      \pgfsetbaseline { -0.7ex }
      \pgfpathcircle { \pgfpoint { 0 } { 0 } } { 0.16em }
      \pgfusepath { fill }
    \end{pgfpicture}
  }
\cs_new_protected:Npn \__moloch_itemize_square:
  {
    \begin{pgfpicture}
      \pgfsetfillcolor { molochItemizeColor }
      \pgfsetbaseline { -0.7ex }
      \pgfpathrectangle
        { \pgfpoint { -0.165em } { -0.165em } }
        { \pgfpoint { 0.31em } { 0.31em } }
      \pgfusepath { fill }
    \end{pgfpicture}
  }
\cs_new_protected:Npn \__moloch_itemize_triangle:
  {
    \begin{pgfpicture}
```

```
      \pgfsetfillcolor { molochItemizeColor }
      \pgfsetbaseline { -0.7ex }
      \pgfpathmoveto { \pgfpoint { 0.21em } { 0 } }
      \pgfpathlineto { \pgfpoint { -0.21em } { 0.21em } }
      \pgfpathlineto { \pgfpoint { -0.21em } { -0.21em } }
      \pgfpathclose
      \pgfusepath { fill }
    \end{pgfpicture}
  }
\cs_set_protected:Npn \labelitemi
  { \__moloch_itemize_marker:nn { \__moloch_itemize_circle: } { 0.05ex } }
\cs_set_protected:Npn \labelitemii
  { \__moloch_itemize_marker:nn { \__moloch_itemize_square: } { 0pt } }
\cs_set_protected:Npn \labelitemiii
  { \__moloch_itemize_marker:nn { \__moloch_itemize_triangle: } { 0.25ex } }
```

H.9 Captions

The class keeps captions unnumbered. The kernel's caption hook and label socket let us adjust their typography without replacing the code that tags the caption and its label.

```
\AddToHook { cmd / @makecaption / before } { \small }
\NewSocketPlug { caption / label } { moloch }
  { \textbf {#1} :~ }
\AssignSocketPlug { caption / label } { moloch }
```

H.10 Title Page

The class builds its title page from a list of named elements, each of which is an instance of the `titlepage-element` template, and takes the content of an element named `x` from the macro . A separator rule is therefore added by defining the content and declaring a matching instance, then naming it in the element order.

```
\tl_new:c { @titleseparator }
\tl_set:cn { @titleseparator }
  { \noindent \rule { \textwidth } { \l__moloch_titlerule_dim } }
\DeclareInstance { titlepage-element } { titleseparator } { talk }
  {
    before-skip = 0.4em ,
    after-skip = 0.8em ,
```

```
    color = mTitleseparatorFg
  }
```

The remaining elements only need Moloch's fonts and spacing.

```
\EditInstance { titlepage-element } { title }
  { font = \Large \bfseries , color = mNormaltextFg }
\EditInstance { titlepage-element } { subtitle }
  { font = \large , color = mNormaltextFg , before-skip = 0.3em }
\EditInstance { titlepage-element } { author }
  { font = \small , before-skip = 0em , after-skip = 0.5em }
\EditInstance { titlepage-element } { institute }
  { font = \scriptsize , color = mNormaltextFg!80!mNormaltextBg }
\EditInstance { titlepage-element } { date }
  { font = \small , before-skip = 1em }
```

H.10.1 `\__moloch_titlepage:n`

The `titlepage` template has no instance, so its defaults cannot be edited and has to be redefined instead.

The Beamer theme places the title block slightly above the vertical center, in golden-ratio proportion. This theme uses the class's centered alignment instead.

```
\cs_new_protected:Npn \__moloch_titlepage:n #1
  {
    \UseTemplate { titlepage } { talk }
      {
        frame-style = plain ,
        horizontal-alignment = left ,
        vertical-alignment = center ,
        element-order =
          { title , subtitle , titleseparator , author , institute , date } ,
        #1
      }
  }
```

H.10.2 `\maketitle`

Whether the frame environment takes a mandatory title argument depends on the class option `frame-title-arg`, which is already fixed by the time this package loads. The empty group is therefore inserted at definition time, as the class itself does. Everything must sit in a single macro body, because the frame environment collects its content up to a literal {frame}.

```
\use:e
  {
    \exp_not:N \RenewDocumentCommand \exp_not:N \maketitle { O { } }
      {
        \exp_not:N \bool_if:NTF \exp_not:N \l__talk_frame_bool
          { \exp_not:N \__moloch_titlepage:n { \exp_not:N #1 } }
          {
            \exp_not:N \begin {_frame }
              \bool_if:NT \l__talk_frame_title_bool { { } }
              \exp_not:N \__moloch_titlepage:n { \exp_not:N #1 }
            \exp_not:N \end {_frame }
          }
      }
  }
```

H.11 Section Pages

The class emits nothing at a section heading, but does provide a hook. Note that, unlike in Beamer, a section heading inside a frame is ignored entirely by the class, so a section page can only be produced from outside a frame.

```
\cs_new_protected:Npn \__moloch_section_page_body:
  {
    \thispagestyle { plain }
    \centering
    \begin {_minipage } { 0.7875 \linewidth }
      \raggedright
      \color { mNormaltextFg }
      \Large \bfseries
      \insertsection
      \par
      \str_if_eq:VnT \l__moloch_sectionpage_tl { progressbar }
        {
          \skip_vertical:n { -0.5 \baselineskip }
          \__moloch_progress_rule:n { \linewidth }
        }
      % The Beamer theme always leaves a line below the bar for the
      % subsection, whether or not there is one. Keeping it means the block
      % is the same height, and so sits at the same place on the page.
      \par
      \large
      \strut
```

```
    \end {_minipage }
    \par
```

A section page carries neither the frame title band nor the footer, so the block is centered on the paper rather than in the text area, which is what the Beamer theme does. The frame centers its content between equal glue, so a skip at the end moves the block up by half its size.

```
    \skip_vertical:n { \Gm@tmargin - \Gm@bmargin }
  }
\use:e
  {
    \exp_not:N \cs_new_protected:Npn \exp_not:N \__moloch_section_page:
      {
        \exp_not:N \begin {_frame }
          [ vertical-alignment = center ]
          \bool_if:NT \l__talk_frame_title_bool { { } }
          \exp_not:N \__moloch_section_page_body:
        \exp_not:N \end {_frame }
      }
  }
\AddToHook { section / begin }
  {
    \str_if_eq:VnF \l__moloch_sectionpage_tl { none }
      { \__moloch_section_page: }
  }
```

H.12 Standout Frames

A standout frame inverts the colors and centers a short message. The class has no per-frame appearance option, so this is a separate environment. The semantic structure color follows the standout foreground so list labels and block titles remain visible against its background.

The background cannot be set with : the class redefines that command to install a permanent background hook, so it would tint every later page as well. Instead a single hook is installed once, guarded by the frame number recorded when the standout body is typeset. The class saves this body for reused and supplementary frames, so they also record their current frame number when rendered. All slides of that frame share the same background.

```
\int_new:N \g__moloch_standout_frame_int
\int_gset:Nn \g__moloch_standout_frame_int { -1 }
```

H.12.1 standout

```
\cs_new_protected:Npn \__moloch_standout_body:n #1
  {
    \int_gset:Nn \g__moloch_standout_frame_int { \value { frame } }
    \bool_set_true:N \l__moloch_standout_bool
    \DeclareColor { structure } { mStandoutFg }
    \color { mStandoutFg }
    \centering
    \Large \bfseries
    #1
    \par
  }
```

The frame number is recorded globally because the body is typeset in a box whose group ends before shipout. The next ordinary frame advances the counter, so the hook no longer applies to it.

```
\use:e
  {
    \exp_not:N \NewDocumentEnvironment { standout }
      { !D <> { all } !O { } +b }
      {
        \exp_not:N \begin {_frame } < \exp_not:N #1 >
          [ vertical-alignment = center , \exp_not:N #2 ]
          \bool_if:NT \l__talk_frame_title_bool { { } }
          \exp_not:N \__moloch_standout_body:n { \exp_not:N #3 }
        \exp_not:N \end {_frame }
      }
      { }
  }
```

H.13 Page Background and Text Color

The page-building hooks keep a frame's saved settings active through both the page furniture and the shipout background hooks. Their extra group restores the surrounding palette, page options, and page style afterward. A standout uses the plain style on every page, including automatic frame breaks.

```
\AddToHook { build / page / before }
  {
    \group_begin:
    \bool_lazy_and:nnT
```

```
      { \int_compare_p:nNn { \value { frame } } = \g__moloch_settings_frame_int }
      { \int_compare_p:nNn { \value { slide } } = \g__moloch_settings_slide_int }
      { \tl_use:N \g__moloch_frame_settings_tl }
    \int_compare:nNnT
      { \value { frame } } = \g__moloch_standout_frame_int
      { \pagestyle { plain } }
  }
\AddToHook { build / page / after } { \group_end: }
```

The page color and the default text color are set at the beginning of the document, so that a change of variant in the preamble is picked up.

Standard color resets, including those used by minipage footnotes, must retain the standout foreground while its body is being typeset.

The standout background hook is installed here too, and deliberately after the page color, so that it is applied on top of it.

```
\AddToHook { begindocument }
  {
    \pagecolor { mNormaltextBg }
    \color { mNormaltextFg }
    \renewcommand \normalcolor
      {
        \bool_if:NTF \l__moloch_standout_bool
          { \color { mStandoutFg } }
          { \color { mNormaltextFg } }
      }
    \AddToHook { shipout / background }
      {
        \int_compare:nNnT
          { \value { frame } } = \g__moloch_standout_frame_int
          {
            \put ( 0cm , -\paperheight )
              {
                \group_begin:
                  \color { mStandoutBg }
                  \rule { \paperwidth } { \paperheight }
                \group_end:
              }
          }
      }
  }
```