Modern Beamer Presentations with the METROPOLIS package

Matthias Vogelgesang
matthias.vogelgesang@gmail.com

v1.1 — 2016/03/14

Contents

1   Introduction

Beamer is an awesome way to make presentations with LaTeX, but its theme selection is surprisingly sparse. The stock themes share an aesthetic that can be a little cluttered, while the few distinctive custom themes available are often specialized for a particular corporate or institutional brand.

The goal of **METROPOLIS** 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.

By default, **METROPOLIS** uses Fira Sans, a gorgeous typeface commissioned by Mozilla and designed by Carrois. For best results, you will need the Fira typeface installed and use XɘL�ATEX to typeset your slides. However, **METROPOLIS** can also be used with other typefaces and LATEX build systems.

METROPOLIS's codebase is maintained on GitHub. If you have issues, find mistakes in the manual or want to help make the theme even better, please get in touch there. The full list of contributors already contains over a dozen names!

2 Getting Started

2.1 Installing from CTAN

For most users, we recommend installing METROPOLIS from CTAN. If you keep your TEX distribution up-to-date, chances are good that METROPOLIS 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 any other distribution please refer to its documentation on how to update your packages.

To get the most out of the theme you should also install the `Fira` fonts. However, this is not mandatory; METROPOLIS also works with the standard fonts.

2.2 Installing from GitHub

If you want to use the cutting-edge development version of METROPOLIS, you can install it manually. Like any LATEX package, this involves four easy steps:

Download the source with a `git clone` of the METROPOLIS repository or as a zip archive of the latest development version.

Compile the style files by running `make sty` inside the downloaded directory. (Or run LATEX directly on `source/metropolistheme.ins`.)

Move the resulting `*.sty` files to the folder containing your presentation. To use METROPOLIS with many presentations, run `make install` or move the `*.sty` files to a folder in your TEX path instead.

Use the theme for your presentation by declaring `\usetheme{metropolis}` in the preamble of your Beamer document.

METROPOLIS uses the Make build system to offer the following installation options for advanced users:

`make sty` builds the theme style files.

`make doc` builds this documentation manual.

`make demo` builds a demo presentation to test the features of METROPOLIS.

`make all` builds the theme and manual.

`make clean` removes the files generated by `make all`.

`make install` installs the theme into your local texmf folder.

`make uninstall` removes the theme from your local texmf folder.

2.3 A Minimal Example

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

```latex
\documentclass{beamer}
\usetheme{metropolis}        % Use metropolis theme
\title{A minimal example}
\date{\today}
\author{Matthias Vogelgesang}
\institute{Centre for Modern Beamer Themes}
\begin{document}
  \maketitle
  \section{First Section}
  \begin{frame}{First Frame}
    Hello, world!
  \end{frame}
\end{document}
```

2.4 Dependencies

METROPOLIS depends on the **beamer** class and the following standard packages:

- `tikz`
- `pgfopts`
- `etoolbox`
- `calc`
- `ifxetex`
- `ifluatex`

For best results, we recommend installing the fonts Fira Sans and `Fira Mono` and compiling with METROPOLIS using X∃LATEX or LuaTEX. These are optional dependencies; METROPOLIS is compatible with (e.g.) pdfLATEX and will fall back to standard fonts if `Fira Sans` or `Fira Mono` is not installed.

The packaged name of `Fira Sans` is `Fira Sans OT` in some Linux distributions; this case is automatically handled by METROPOLIS.

2.5 Pandoc

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

```
$ pandoc -t beamer --latex-engine=xelatex -V theme:
    metropolis -o output.pdf input.md
```

3 Customization

3.1 Package options

The theme provides a number of options, which can be set using a key=value interface. The primary way to set options is to provide a comma-separated list of option-value pairs when loading METROPOLIS in the preamble:

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

Options can be changed at any time — even mid-presentation! — with the `\metroset` macro.

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

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

option key *list of possible values* . . . . default
A short description of the option.

3.1.1 Main theme

titleformat *regular, smallcaps, allsmallcaps, allcaps* . . . . regular
Changes the format of titles, subtitles, section titles, frame titles, and the text on
"standout" frames. The available options produce Regular, SMALLCAPS, ALLSMALL-
CAPS, or ALLCAPS titles. Please refer to Section 6.1 for known issues with these
options.

titleformat plain *regular, smallcaps, allsmallcaps, allcaps* . . . . regular
Changes the format of "standout" frames (see `titleformat`, above).

3.1.2 Inner theme

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

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

3.1.3 Outer theme

numbering *none, counter, fraction* . . . . counter

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

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

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

3.1.4 Color theme

block *transparent, fill* . . . . transparent

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

background *dark, light* . . . . light

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

3.1.5 Font theme

titleformat title *regular, smallcaps, allsmallcaps, allcaps* . . . . regular
titleformat subtitle
titleformat section   Individually controls the format of titles, subtitles, section titles, and frame titles
titleformat frame   (see `titleformat`, above).

3.2 Color Customization

The included METROPOLIS color theme is used by default, but its colors can be
easily changed to suit your tastes. All of the theme's styles are defined in terms
of three beamer colors:

· `normal text` (dark fg, light bg)

8

- · `alerted text` (colored fg, should be visible against dark or light)
- · `example text` (colored fg, should be visible against dark or light)

An easy way to customize the theme is to redefine these colors using

`\setbeamercolor{ ... }{ fg= ... , bg= ... }`

in your preamble. For greater customization, you can redefine any of the other stock beamer colors. In addition to the stock colors the theme defines a number of METROPOLIS specific colors, which can also be redefined to your liking.

`\setbeamercolor{progress bar}{ ... }`
`\setbeamercolor{title separator}{ ... }`
`\setbeamercolor{progress bar in head/foot}{ ... }`
`\setbeamercolor{progress bar in section page}{ ... }`

3.3   Font Customization

The default font for METROPOLIS is `Fira`. This can be easily changed using the standard font selection commands of the fontspec package. So if you prefer, for example, the Ubuntu font family, just add the following two commands after loading the METROPOLIS theme.

`\setsansfont{Ubuntu}`
`\setmonofont{Ubuntu Mono}`

If you are expecting to present in a large room or with an underpowered projector, you may want to change the font to a heavier weight of Fira to maximize readability.

`\setsansfont[BoldFont={Fira Sans SemiBold}]{Fira Sans Book}`

3.3.1   Old style figures

The regular fontspec mechanism for changing glyph appearance applies also to this theme. If you want to have old style figures in the text but regular lined

figures for math, you could add the following to your preamble:

```
\usefonttheme{professionalfonts}   % required for mathspec
\usepackage{mathspec}
\setsansfont[BoldFont={Fira Sans},
            Numbers={OldStyle}]{Fira Sans Light}
\setmathsfont(Digits)[Numbers={Lining, Proportional}]{Fira
    Sans Light}
```

3.4   Commands

3.4.1   Standout frames

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

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

4   pgfplots integration

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

4.1   Styles

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

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

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

`horizontal mbarplot`	Plot horizontal bar charts.
`disable thousands separator`	Helper style to remove thousands separator.

4.2 Paul Tol colors

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

In a technical note for SRON, Paul Tol proposed a palette of colors satisfying these constraints. The sub-package `pgfplotsthemetol` defines palettes for `pgfplots` charts based on Tol's work.

5 Tips & Tricks

5.1 Backup Slides

Speakers will often include extra slides at the end of their presentation to refer to during audience questions. One easy way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

METROPOLIS will automatically turn off slide numbering and progress bars for slides in the appendix.

6 Known Issues

6.1 Title formats

Be aware that not every font supports small caps, so the `smallcaps` or `allsmallcaps` options may not work if you use a font other than `Fira Sans`. In particular, the Computer Modern sans-serif typeface, which is used when METROPOLIS is compiled with pdfLaTeX, does not have a small-caps variant.

The title format options `allsmallcaps` and `allcaps` are quite nice from an aesthetic point of view, but their use of `\MakeLowercase` and `\MakeUppercase` can cause unexpected problems. For example:

- Some commands, like `\\`, do not work inside `\MakeLowercase` and `\MakeUppercase`. (See #125)
- Only alphabetic characters are affected by `\MakeLowercase`, so numerals and punctuation remain at full height. This can spoil some of the aesthetic benefits of `allsmallcaps`. (See #33)
- `\MakeLowercase` and `\MakeUppercase` apply to math mode and `\scshape` does not. This can easily introduce mathematical errors that are hard to catch.
- It is impossible to typeset symbols which are encoded as uppercase letters in a different font. In particular, `\mathbb` and `\mathcal` letters will be replaced by other math glyphs. (See #153)

The `allsmallcaps` and `allcaps` options are safe to use if your titles contain only alphabetic characters and do not require the expansion of any macros.

6.2 Interactions with other color themes

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

For example, overriding the color theme as follows may not work as expected because `\usetheme{metropolis}` loads the METROPOLIS 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{metropolis}
\usecolortheme{seahorse}
```

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

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

Please note that **METROPOLIS** may not use all the colors defined in your favourite Beamer color theme. In particular, **METROPOLIS** 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 XLATEX, 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 XLATEX itself. You can work around it either by compiling with LuaTEX or by adding the following code to your preamble to reset the text color on each slide.

```
\makeatletter
\def\beamer@framenotesbegin{% at beginning of slide
    \usebeamercolor[fg]{normal text}
     \gdef\beamer@noteitems{}%
     \gdef\beamer@notes{}%
}
\makeatother
```

6.4   Standout frames with labels

Because the `standout` frame option creates a group to restrict the colour change to a single slide, labels defined after calling `standout` will stay local to the group. In other words, the following may result in a "label undefined" error.

```
\begin{frame}[standout, label=conclusion]{Conclusion}
  Awesome slide
```

13

```
\end{frame}
```

To fix this problem, change the order of the keys in the frame.

```
\begin{frame}[label=conclusion, standout]{Conclusion}
    Awesome slide
\end{frame}
```

This error can be unwittingly triggered if you export your slides from Emacs Org mode, which automatically adds labels after frame options. Alex Branham offers the following solution for Org mode users, using `org-set-property`.

```
* Start of a frame
  :PROPERTIES:
  :BEAMER_opt: label=conclusion,standout
  :END:
```

6.5   Standout frames with Pandoc

With Pandoc versions prior 1.17.2 it was not possible to create standout frames because Pandoc only supported a specific list of frame attributes thus ignoring additional attributes such as `{.standout}`.

7   License

METROPOLIS is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. This means that if you change the theme and re-distribute it, you must retain the copyright notice header and license it under the same CC-BY-SA license. This does not affect any presentations that you create with the theme.

8 Implementation

8.1 METROPOLIS parent theme

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

8.1.1 Package dependencies

```
1 \RequirePackage{etoolbox}
2 \RequirePackage{pgfopts}
```

8.1.2 Options

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

```
3 \pgfkeys{/metropolis/.cd,
4   .search also={
5     /metropolis/inner,
6     /metropolis/outer,
7     /metropolis/color,
8     /metropolis/font,
9   }
10 }
```

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

```
11 \pgfkeys{
12   /metropolis/titleformat plain/.cd,
13     .is choice,
14     regular/.code={%
15       \let\metropolis@plaintitleformat\@empty%
16       \setbeamerfont{standout}{shape=\normalfont}%
17     },
18     smallcaps/.code={%
19       \let\metropolis@plaintitleformat\@empty%
20       \setbeamerfont{standout}{shape=\scshape}%
21     },
```

```
22    allsmallcaps/.code={%
23      \let\metropolis@plaintitleformat\MakeLowercase%
24      \setbeamerfont{standout}{shape=\scshape}%
25      \PackageWarning{beamerthememetropolis}{%
26        Be aware that titleformat plain=allsmallcaps can lead to prob-
   lems%
27        }
28      },
29    allcaps/.code={%
30      \let\metropolis@plaintitleformat\MakeUppercase%
31      \setbeamerfont{standout}{shape=\normalfont}%
32      \PackageWarning{beamerthememetropolis}{%
33        Be aware that titleformat plain=allcaps can lead to prob-
   lems%
34        }
35      },
36 }
```

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

```
37 \pgfkeys{
38   /metropolis/titleformat/.code=\pgfkeysalso{
39      font/titleformat title=#1,
40      font/titleformat subtitle=#1,
41      font/titleformat section=#1,
42      font/titleformat frame=#1,
43      titleformat plain=#1,
44      }
45 }
```

For backwards compatibility with earlier betas of the theme, we implement dep-
recated option names as aliases to the corresponding `key=value` options.

```
46 \pgfkeys{/metropolis/.cd,
47   usetitleprogressbar/.code=\pgfkeysalso{outer/progressbar=frametitle},
48   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
49   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
50   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
51   darkcolors/.code=\pgfkeysalso{color/background=dark},
52   blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
```

```
53 }
```

Set default values for options.

```
54 \newcommand{\metropolis@setdefaults}{
55   \pgfkeys{/metropolis/.cd,
56     titleformat plain=regular,
57   }
58 }
```

8.1.3 Component sub-packages

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

```
59 \useinnertheme{metropolis}
60 \useoutertheme{metropolis}
61 \usecolortheme{metropolis}
62 \usefonttheme{metropolis}
```

The `tol` theme for `pgfplots` is only loaded if `pgfplots` is used.

```
63 \AtEndPreamble{%
64   \@ifpackageloaded{pgfplots}{%
65     \RequirePackage{pgfplotsthemetol}
66   }{}
67 }
```

8.1.4 Custom commands

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

`\metroset` Allows the user to change options midway through a presentation.

```
68 \newcommand{\metroset}[1]{\pgfkeys{/metropolis/.cd,#1}}
```

`\plain` Creates a plain frame with dark background, suitable for displaying images or a few words. The format of the text can be set with the `titleformat plain` option.

17

```
69 \def\metropolis@plaintitleformat#1{#1}
70 \newcommand{\plain}[2][]{%
71   \PackageWarning{beamerthememetropolis}{%
72     The syntax '\plain' may be deprecated in a future version of Metropo-
   lis.
73     Please use a frame with [standout] instead.
74   }
75   \begin{frame}[standout]{#1}
76     \metropolis@plaintitleformat{#2}
77   \end{frame}
78 }
```

\mreducelistspacing

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

8.1.5   Process package options

```
80 \metropolis@setdefaults
81 \ProcessPgfOptions{/metropolis}
```

8.2   METROPOLIS inner theme

A **beamer** inner theme dictates the style of the frame elements traditionally set
in the "body" of each slide. These include:

- · title, part, and section pages;
- · itemize, enumerate, and description environments;
- · block environments including theorems and proofs;
- · figures and tables; and
- · footnotes and plain text.

8.2.1   Package dependencies

```
82 \RequirePackage{etoolbox}
83 \RequirePackage{keyval}
84 \RequirePackage{calc}
85 \RequirePackage{pgfopts}
86 \RequirePackage{tikz}
```

8.2.2 Options

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

```
87 \pgfkeys{
88   /metropolis/inner/sectionpage/.cd,
89     .is choice,
90     none/.code=\metropolis@disablesectionpage,
91     seiscope/.code=\metropolis@customsectionpage,
92     simple/.code={\metropolis@enablesectionpage
93                   \setbeamertemplate{section page}[simple]},
94     progressbar/.code={\metropolis@enablesectionpage
95                        \setbeamertemplate{section page}[progressbar]},
96 }
```

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

```
97 \pgfkeys{
98   /metropolis/inner/subsectionpage/.cd,
99     .is choice,
100    none/.code=\metropolis@disablesubsectionpage,
101    simple/.code={\metropolis@enablesubsectionpage
102                  \setbeamertemplate{section page}[simple]},
103    progressbar/.code={\metropolis@enablesubsectionpage
104                       \setbeamertemplate{section page}[progressbar]},
105 }
```

etropolis@inner@setdefaults Set default values for inner theme options.

```
106 \newcommand{\metropolis@inner@setdefaults}{
107   \pgfkeys{/metropolis/inner/.cd,
108     sectionpage=progressbar,
109     subsectionpage=none
110   }
111 }
```

8.2.3 Title page

title page Template for the title page. Each element is only typset if it is defined by the user. If \subtitle is empty, for example, it won't leave a blank space on the

title slide.

```
112 \setbeamertemplate{title page}{
113   \begin{minipage}[b][\paperheight]{\textwidth}
114     \ifx\inserttitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
115     \vfill%
116     \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
117     \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
118     \usebeamertemplate*{title separator}
```

Beamer's definition of `\insertauthor` is always nonempty, so we have to test another macro initialized by `\author{...}` to see if the user has defined an author. This solution was suggested by Enrico Gregorio in an answer to .

```
119     \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
120     \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
121     \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
122     \vfill
123     \vspace*{1mm}
124   \end{minipage}
125 }
```

Normal people should use `\maketitle` or `\titlepage` instead of using the `title page` beamer template directly. Beamer already defines these macros, but we patch them here to make the title page `[plain]` by default, remove `\@thanks`, and ensure the title frame number doesn't count.

`\maketitle` Inserts the title frame, or causes the current frame to use the `title page` tem-
`\titlepage` plate.

```
126 \def\maketitle{%
127   \ifbeamer@inframe
128     \titlepage
129   \else
130     \frame[plain,noframenumbering]{\titlepage}
131   \fi
132 }
133 \def\titlepage{%
134   \usebeamertemplate{title page}
135 }
```

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

```
136 \setbeamertemplate{title graphic}{
137   \vbox to 0pt {
138     \vspace*{2em}
139     \inserttitlegraphic%
140   }%
141   \nointerlineskip%
142 }
```

title — Set the title on the title page.

```
143 \setbeamertemplate{title}{
144   \raggedright%
145   \linespread{1.0}%
146   \inserttitle%
147   \par%
148   \vspace*{0.5em}
149 }
```

subtitle — Set the subtitle on the title page.

```
150 \setbeamertemplate{subtitle}{
151   \raggedright%
152   \insertsubtitle%
153   \par%
154   \vspace*{0.5em}
155 }
```

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

```
156 \setbeamertemplate{title separator}{
157   \begin{tikzpicture}
158     \fill[fg] (0,0) rectangle (\textwidth, 1.5pt);
159   \end{tikzpicture}%
160   \par%
161 }
```

author — Set the author on the title page.

```
162 \setbeamertemplate{author}{
163   \vspace*{2em}
164   \insertauthor%
165   \par%
166   \vspace*{0.25em}
167 }
```

date   Set the date on the title page.

```
168 \setbeamertemplate{date}{
169   \insertdate%
170   \par%
171 }
```

institute   Set the institute on the title page.

```
172 \setbeamertemplate{institute}{
173   \vspace*{3mm}
174   \insertinstitute%
175   \par%
176 }
```

8.2.4   Section page

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

```
177 \defbeamertemplate{section page}{simple}{
178   % \begin{center}
179     \usebeamercolor[fg]{section title}
180     \usebeamerfont{section title}
181     \frametitle{Outline}
182     % \insertsectionhead\par
183     % \tableofcontents[currentsection]
184     \tableofcontents[currentsection,subsectionstyle=show/show/hide]
185     \ifx\insertsubsectionhead\@empty\else
186       \usebeamercolor[fg]{subsection title}
187       \usebeamerfont{subsection title}
188       \tableofcontents[currentsection,subsectionstyle=show/shaded/hide]
189     \fi
190   % \end{center}
```

22

```latex
191 }
192 \defbeamertemplate{section page}{progressbar}{
193   \centering
194   \begin{minipage}{22em}
195     \raggedright
196     \usebeamercolor[fg]{section title}
197     \usebeamerfont{section title}
198     % \insertsectionhead\\[-1ex]
199     \tableofcontents[currentsection]
200     \usebeamertemplate*{progress bar in section page}
201     \par
202     \ifx\insertsubsectionhead\@empty\else%
203       \usebeamercolor[fg]{subsection title}%
204       \usebeamerfont{subsection title}%
205       \insertsubsectionhead
206     \fi
207   \end{minipage}
208   \par
209   \vspace{\baselineskip}
210 }


213 \newcommand{\metropolis@customsectionpage}{
214   \AtBeginSection[ ]
215   {
216   \begin{frame}<beamer>[noframenumbering]
217   \frametitle{Outline}
218   \small{
219   \tableofcontents[currentsection,subsectionstyle=show/show/hide]
220   }
221   \end{frame}
222   }
223
224   \AtBeginSubsection[ ]
225   {
226   \begin{frame}<beamer>[noframenumbering]
227   \frametitle{Outline}
228   \small{
229   \tableofcontents[currentsection,subsectionstyle=show/shaded/hide]
230   }
```

```
231   \end{frame}
232   }
233 }
234
235
236
237
238 \newcommand{\metropolis@disablesectionpage}{
239   \AtBeginSection{
240     % intentionally empty
241   }
242 }
243 \newcommand{\metropolis@enablesectionpage}{
244   \AtBeginSection{
245     \ifbeamer@inframe
246       \sectionpage
247     \else
248       % \frame[plain,c,noframenumbering]{\sectionpage}
249       {
250       \metroset{numbering=none}
251       \frame[noframenumbering]{\sectionpage}
252       }
253     \fi
254   }
255 }
```

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

```
256 \setbeamertemplate{subsection page}{%
257   \usebeamertemplate*{section page}
258 }
259 \newcommand{\metropolis@disablesubsectionpage}{
260   \AtBeginSubsection{
261     % intentionally empty
262   }
263 }
264 \newcommand{\metropolis@enablesubsectionpage}{
265   \AtBeginSubsection{
266     \ifbeamer@inframe
```

```
267        \subsectionpage
268      \else
269      % \frame[plain,c,noframenumbering]{\subsectionpage}
270      {
271      \metroset{numbering=none}
272      \frame[noframenumbering]{\subsectionpage}
273      }
274      \fi
275    }
276 }
```

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`.

```
277 \newlength{\metropolis@progressonsectionpage}
278 \setbeamertemplate{progress bar in section page}{
279    \setlength{\metropolis@progressonsectionpage}{%
280      \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
281    }%
282    \begin{tikzpicture}
283      \fill[bg] (0,0) rectangle (\textwidth, 1.2pt);
284      \fill[fg] (0,0) rectangle (\metropolis@progressonsectionpage, 1.2pt);
285    \end{tikzpicture}%
286 }
```

The above code assumes that `\insertframenumber` is less than or equal to `\inserttotalframenumber`. However, this is not true on the first compile; in the absence of an `.aux` file, `\inserttotalframenumber` defaults to 1. This behaviour could cause fatal errors for long presentations, as `\metropolis@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.

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

8.2.5 Block environments

block

block alerted

block example

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

```
288 \newlength{\metropolis@blocksep}
289 \newlength{\metropolis@blockadjust}
290 \setlength{\metropolis@blocksep}{0.75ex}
291 \setlength{\metropolis@blockadjust}{0.25ex}
292 \providecommand{\metropolis@strut}{%
293   \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz()}%
294 }
295 \newcommand{\metropolis@block}[1]{
296   \par\vskip\medskipamount%
297   \setlength{\parskip}{0pt}
```

If a background color is defined for the block title or body, we need to add a little bit of padding to the corresponding box. Ideally, this would be accomplished by setting `colsep=0.75ex`, which is intended to add "color separation space" only when the box has a colored background. Unfortunately, `colsep` also adds this separation if the background color is inherited, even if the inherited color is actually empty. (The technical reason for this boils down to the fact that the `\ifx` directive does not expand macros.)

To achieve the correct spacing for `alertblock`s and `exampleblock`s as well as for normal blocks, we have to begin the `beamercolorbox` differently based on whether `block title` has an empty background.

If the `block title` background is empty, or the user has explicitly removed the background from (e.g.) `block title alerted`, we just need to set a rightskip for a nice ragged-right block title.

```
298   \ifbeamercolorempty[bg]{block title#1}{%
299     \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}}{%
300   \ifbeamercolorempty[bg]{block title}{%
301     \begin{beamercolorbox}[rightskip=0pt plus 4em]{block title#1}%
302   }%
303 %   \end{macrocode}
304 %
305 %   Otherwise, if the |block title| has a background, we set the padding based
```

26

```
306 %   on |\metropolis@blockskip|. However, we have to visually com-
    pensate for
307 %   the |\metropolis@strut| added to the block title (see below) by
308 %   subtracting |\metropolis@blockadjust| from the top and bottom padding.
309 %
310 %   \begin{macrocode}
311   {%
312     \begin{beamercolorbox}[
313       sep=\dimexpr\metropolis@blocksep-\metropolis@blockadjust\relax,
314       leftskip=\metropolis@blockadjust,
315       rightskip=\dimexpr\metropolis@blockadjust plus 4em\relax
316     ]{block title#1}%
317   }}%
318 %   \end{macrocode}
319 %
320 %   We can now set the contents of the |block title|. The zero-width but
321 %   positive-height box |\metropolis@strut| ensures that the block ti-
    tle box
322 %   has a consistent height, even if it lacks punctuation, ascen-
    ders, or
323 %   descenders.
324 %
325 %   \begin{macrocode}
326     \usebeamerfont*{block title#1}%
327     \metropolis@strut%
328     \insertblocktitle%
329     \metropolis@strut%
330   \end{beamercolorbox}%
331 %   \end{macrocode}
332 %
333 %   Next, we typeset the |block body|. This the code is similar to, but sim-
    pler
334 %   than, the |block title| code since we don't need to adjust for any struts.
335 %
336 %   \begin{macrocode}
337   \nointerlineskip%
338   \ifbeamercolorempty[bg]{block body#1}{%
339     \begin{beamercolorbox}[vmode]{block body#1}}{
340   \ifbeamercolorempty[bg]{block body}{%
341     \begin{beamercolorbox}[vmode]{block body#1}%
```

```
342  }{%
343    \begin{beamercolorbox}[sep=\metropolis@blocksep, vmode]{block body#1}%
344    \vspace{-\metropolis@parskip}
345  }}%
346      \usebeamerfont{block body#1}%
347      \setlength{\parskip}{\metropolis@parskip}%
348  }
```

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

```
349  \setbeamertemplate{block begin}{\metropolis@block{}}
350  \setbeamertemplate{block alerted begin}{\metropolis@block{ alerted}}
351  \setbeamertemplate{block example begin}{\metropolis@block{ exam-
       ple}}
352  \setbeamertemplate{block end}{\end{beamercolorbox}\vspace*{0.2ex}}
353  \setbeamertemplate{block alerted end}{\end{beamercolorbox}\vspace*{0.2ex}}
354  \setbeamertemplate{block example end}{\end{beamercolorbox}\vspace*{0.2ex}}
```

8.2.6   Lists and floats

```
355  \setbeamertemplate{itemize items}{\textbullet}
356  \setbeamertemplate{caption label separator}{: }
357  \setbeamertemplate{caption}[numbered]
```

8.2.7   Footnotes

```
358  \setbeamertemplate{footnote}{%
359    \parindent 0em\noindent%
360    \raggedright
361    \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotete
362  }
```

8.2.8   Text and spacing settings

```
363  \newlength{\metropolis@parskip}
364  \setlength{\metropolis@parskip}{0.5em}
365  \setlength{\parskip}{\metropolis@parskip}
366  \linespread{1.15}
```

By default, Beamer frames offer the **c** option to *almost* vertically center the text, but the placement is a little too high. To fix this, we redefine the **c** option

to equalize `\beamer@frametopskip` and `\beamer@framebottomskip`. This solution was suggested by Enrico Gregorio in an answer to this Stack Exchange question.

```
367 \define@key{beamerframe}{c}[true]{% centered
368   \beamer@frametopskip=0pt plus 1fill\relax%
369   \beamer@framebottomskip=0pt plus 1fill\relax%
370   \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
371   \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
372   \def\beamer@initfirstlineunskip{}%
373 }
```

8.2.9   Standout frames

METROPOLIS 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] ... \`

standout Optional arguments to Beamer's frames are implemented using `\define@key` from the `keyval` package, which will execute code when the defined option is called. For the `standout` option, we begin a group, change the colors and fonts, and set a alignment.

```
374 \providebool{metropolis@standout}
375 \define@key{beamerframe}{standout}[true]{%
376   \booltrue{metropolis@standout}
377   \begingroup
378     \setkeys{beamerframe}{c}
379     \setkeys{beamerframe}{noframenumbering}
380     \ifbeamercolorempty[bg]{palette primary}{
381       \setbeamercolor{background canvas}{
382         use=palette primary,
383         bg=-palette primary.fg
384       }
385     }{
386       \setbeamercolor{background canvas}{
387         use=palette primary,
388         bg=palette primary.bg
389       }
390     }
391   \centering
```

```
392    \usebeamercolor[fg]{palette primary}
393    \usebeamerfont{standout}
394 }
```

Then we just have to close the group after the standout slide is finished in order to restore the colours and fonts for the rest of the presentation. Unfortunately, we cannot use or this (see). Instead, we add the `\endgroup` to `\beamer@reseteecodes`, which is run exactly once at the end of each slide.

```
395    \apptocmd{\beamer@reseteecodes}{%
396      \ifbool{metropolis@standout}{
397        \endgroup
398        \boolfalse{metropolis@standout}
399      }{}
400    }{}{}
```

8.2.10   Process package options

```
401 \metropolis@inner@setdefaults
402 \ProcessPgfPackageOptions{/metropolis/inner}
```

8.3   METROPOLIS outer theme

A `beamer` outer theme dictates the style of the frame elements traditionally set outside the body of each slide: the head, footline, and frame title.

8.3.1   Package dependencies

```
403 \RequirePackage{etoolbox}
404 \RequirePackage{calc}
405 \RequirePackage{pgfopts}
```

8.3.2   Options

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

```
406 \pgfkeys{
407   /metropolis/outer/numbering/.cd,
```

```
408     .is choice,
409     none/.code=\setbeamertemplate{frame numbering}[none],
410     counter/.code=\setbeamertemplate{frame numbering}[counter],
411     fraction/.code=\setbeamertemplate{frame numbering}[fraction],
412 }
```

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

```
413 \pgfkeys{
414   /metropolis/outer/progressbar/.cd,
415     .is choice,
416     none/.code={%
417       \setbeamertemplate{headline}[plain]
418       \setbeamertemplate{frametitle}[plain]
419       \setbeamertemplate{footline}[plain]
420     },
421     head/.code={\pgfkeys{/metropolis/outer/progressbar=none}
422       \addtobeamertemplate{headline}{}{%
423         \usebeamertemplate*{progress bar in head/foot}
424       }
425     },
426     frametitle/.code={\pgfkeys{/metropolis/outer/progressbar=none}
427       \addtobeamertemplate{frametitle}{}{%
428         \usebeamertemplate*{progress bar in head/foot}
429       }
430     },
431     foot/.code={\pgfkeys{/metropolis/outer/progressbar=none}
432       \addtobeamertemplate{footline}{}{%
433         \usebeamertemplate*{progress bar in head/foot}%
434       }
435     },
436 }
```

etropolis@outer@setdefaults Sets default values for outer theme options.

```
437 \newcommand{\metropolis@outer@setdefaults}{
438   \pgfkeys{/metropolis/outer/.cd,
439     numbering=counter,
440     progressbar=none,
441   }
442 }
```

8.3.3 Head and footline

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

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

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

```
444 \defbeamertemplate{frame footer}{none}{}
445 \defbeamertemplate{frame footer}{custom}[1]{ #1 }

446 \defbeamertemplate{frame numbering}{none}{}
447 \defbeamertemplate{frame numbering}{counter}{\insertframenumber}
448 \defbeamertemplate{frame numbering}{fraction}{
449   \insertframenumber/\inserttotalframenumber
450 }
```

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

footline

```
451 \defbeamertemplate{headline}{plain}{}
452 \defbeamertemplate{footline}{plain}{%
453   \begin{beamercolorbox}[wd=\textwidth, sep=1ex]{footline}%
454     \usebeamerfont{page number in head/foot}%
455     \usebeamertemplate*{frame footer}
456     \hfill%
457     \usebeamertemplate*{frame numbering}
458   \end{beamercolorbox}%
459 }
```

8.3.4 Frametitle

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

```
460 \newlength{\metropolis@frametitle@padding}
461 \setlength{\metropolis@frametitle@padding}{1.5ex}
462 \newcommand{\metropolis@frametitlestrut@start}{
463   \rule{0pt}{\metropolis@frametitle@padding +%
464     \totalheightof{%
465       \ifcsdef{metropolis@frametitleformat}{\metropolis@frametitleformat X}{X}%
```

```
466    }%
467   }%
468 }
469 \newcommand{\metropolis@frametitlestrut@end}{
470   \rule[-\metropolis@frametitle@padding]{0pt}{\metropolis@frametitle@padding}
471 }
472 \defbeamertemplate{frametitle}{plain}{%
473   \nointerlineskip%
474   \begin{beamercolorbox}[%
475       wd=\paperwidth,%
476       sep=0pt,%
477       leftskip=\metropolis@frametitle@padding,%
478       rightskip=\metropolis@frametitle@padding,%
479     ]{frametitle}%
480   \metropolis@frametitlestrut@start%
481   \insertframetitle%
482   \nolinebreak%
483   \metropolis@frametitlestrut@end%
484   \end{beamercolorbox}%
485 }
486 \setbeamertemplate{frametitle continuation}{%
487   \usebeamerfont{frametitle}
488   \romannumeral \insertcontinuationcount
489 }
```

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`.

```
490 \newlength{\metropolis@progressinheadfoot}
491 \setbeamertemplate{progress bar in head/foot}{
492   \nointerlineskip
493   \setlength{\metropolis@progressinheadfoot}{%
494     \paperwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
495   }%
496   \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
497     \begin{tikzpicture}
498       \fill[bg] (0,0) rectangle (\paperwidth, 1.2pt);
499       \fill[fg] (0,0) rectangle (\metropolis@progressinheadfoot, 2pt);
500     \end{tikzpicture}%
```

```
501    \end{beamercolorbox}
502 }
```

appendix   Removes page numbering and per-slide progress bars when `\appendix` is called. This makes it easier to include additional "backup slides" at the end of the presentation, especially in conjunction with the package `appendixnumberbeamer`.

```
503 \AtBeginDocument{%
504   \apptocmd{\appendix}{%
505     \pgfkeys{%
506       /metropolis/outer/.cd,
507       numbering=none,
508       progressbar=none}
509     }{}{}
510 }
```

8.3.5   Process package options

```
511 \metropolis@outer@setdefaults
512 \ProcessPgfPackageOptions{/metropolis/outer}
```

8.4   METROPOLIS font theme

A `beamer` font theme sets the style of the font used in the document.

8.4.1   Package dependencies

```
513 \RequirePackage{etoolbox}
514 \RequirePackage{ifxetex}
515 \RequirePackage{ifluatex}
516 \RequirePackage{pgfopts}
```

8.4.2   Load Fira fonts

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

```
517 \ifboolexpr{bool {xetex} or bool {luatex}}{
518   \RequirePackage[no-math]{fontspec}
```

\checkfont   Checks if a font is installed; if not, `fontsnotfound` is increased.

```
519  \newcounter{fontsnotfound}
520  \newcommand{\checkfont}[1]{%
521    \suppressfontnotfounderror=1%
522    \font\x = "#1" at 10pt
523    \selectfont
524    \ifx\x\nullfont%
525      \stepcounter{fontsnotfound}%
526    \fi%
527    \suppressfontnotfounderror=0%
528  }
529
```

\iffontsavailable   Resets the `fontsnotfound` counter and calls `\checkfont` for each font in the comma separated list in the first argument.

```
530  \newcommand{\iffontsavailable}[3]{%
531    \setcounter{fontsnotfound}{0}%
532    \expandafter\forcsvlist\expandafter%
533    \checkfont\expandafter{#1}%
534    \ifnum\value{fontsnotfound}=0%
535      #2%
536    \else%
537      #3%
538    \fi%
539  }
```

We search for regular, italic, light, light italic, mono, and mono bold fonts under the default Fira Sans and Fira Mono names. If this fails, the suffix OT — used by some Linux distributions — will be tried. If this also fails, a warning will be displayed and the standard fonts will be used.

```
540  \iffontsavailable{Fira Sans Light,%
541                    Fira Sans Light Italic,%
542                    Fira Sans,%
543                    Fira Sans Italic}%
544  {%
545    \setsansfont[ItalicFont={Fira Sans Light Italic},%
546              BoldFont={Fira Sans},%
547              BoldItalicFont={Fira Sans Italic}]%
```

```
548            {Fira Sans Light}%
549    }{%
550      \iffontsavailable{Fira Sans Light OT,%
551                        Fira Sans Light Italic OT,%
552                        Fira Sans OT,%
553                        Fira Sans Italic OT}%
554      {%
555        \setsansfont[ItalicFont={Fira Sans Light Italic OT},%
556                    BoldFont={Fira Sans OT},%
557                    BoldItalicFont={Fira Sans Italic OT}]%
558                   {Fira Sans Light OT}%
559      }{%
560        \PackageWarning{beamerthememetropolis}{%
561          Could not find Fira Sans fonts%
562        }
563      }
564    }
565    \iffontsavailable{Fira Mono, Fira Mono Bold}{%
566      \setmonofont[BoldFont={Fira Mono Medium}]{Fira Mono}%
567    }{%
568      \iffontsavailable{Fira Mono OT, Fira Mono Bold OT}{%
569        \setmonofont[BoldFont={Fira Mono Medium OT}]{Fira Mono OT}%
570      }{%
571        \PackageWarning{beamerthememetropolis}{%
572          Could not find Fira Mono fonts%
573        }
574      }
575    }
576    \AtBeginEnvironment{tabular}{%
577      \addfontfeature{Numbers={Monospaced}}%
578    }
579  }{%
580    \PackageWarning{beamerthememetropolis}{%
581      You need to compile with XeLaTeX or LuaLaTeX to use the Fira fonts%
582    }
583  }
```

This concludes the portion of the code which is only run when compiled with
XeLATeX or LuaLATeX. The remainder of this package applies regardless of the com-
piling engine.

8.4.3 General font definitions

```
584 \setbeamerfont{title}{size=\Large,%
585                      series=\bfseries}
586 \setbeamerfont{author}{size=\small}
587 \setbeamerfont{date}{size=\small}
588 \setbeamerfont{section title}{size=\Large,%
589                               series=\bfseries}
590 \setbeamerfont{block title}{size=\normalsize,%
591                             series=\bfseries}
592 \setbeamerfont{block title alerted}{size=\normalsize,%
593                                     series=\bfseries}
594 \setbeamerfont*{subtitle}{size=\large}
595 \setbeamerfont{frametitle}{size=\large,%
596                            series=\bfseries}
597 \setbeamerfont{caption}{size=\small}
598 \setbeamerfont{caption name}{series=\bfseries}
599 \setbeamerfont{description item}{series=\bfseries}
600 \setbeamerfont{page number in head/foot}{size=\scriptsize}
601 \setbeamerfont{bibliography entry author}{size=\normalsize,%
602                                           series=\normalfont}
603 \setbeamerfont{bibliography entry title}{size=\normalsize,%
604                                          series=\bfseries}
605 \setbeamerfont{bibliography entry location}{size=\normalsize,%
606                                             series=\normalfont}
607 \setbeamerfont{bibliography entry note}{size=\small,%
608                                         series=\normalfont}
609 \setbeamerfont{standout}{size=\Large,%
610                          series=\bfseries}
```

8.4.4 Title format options

`titleformat title` Controls the format of the title.

```
611 \pgfkeys{
612   /metropolis/font/titleformat title/.cd,
613     .is choice,
614     regular/.code={%
615       \let\metropolis@titleformat\@empty%
616       \setbeamerfont{title}{shape=\normalfont}%
617     },
```

```
618    smallcaps/.code={%
619      \let\metropolis@titleformat\@empty%
620      \setbeamerfont{title}{shape=\scshape}%
621    },
622    allsmallcaps/.code={%
623      \let\metropolis@titleformat\lowercase%
624      \setbeamerfont{title}{shape=\scshape}%
625      \PackageWarning{beamerthememetropolis}{%
626        Be aware that titleformat title=allsmallcaps can lead to prob-
   lems%
627      }
628    },
629    allcaps/.code={%
630      \let\metropolis@titleformat\uppercase%
631      \setbeamerfont{title}{shape=\normalfont}
632      \PackageWarning{beamerthememetropolis}{%
633        Be aware that titleformat title=allcaps can lead to prob-
   lems%
634      }
635    },
636 }
```

titleformat subtitle   Control the format of the subtitle.

```
637 \pgfkeys{
638   /metropolis/font/titleformat subtitle/.cd,
639     .is choice,
640     regular/.code={%
641       \let\metropolis@subtitleformat\@empty%
642       \setbeamerfont{subtitle}{shape=\normalfont}%
643     },
644     smallcaps/.code={%
645       \let\metropolis@subtitleformat\@empty%
646       \setbeamerfont{subtitle}{shape=\scshape}%
647     },
648     allsmallcaps/.code={%
649       \let\metropolis@subtitleformat\lowercase%
650       \setbeamerfont{subtitle}{shape=\scshape}%
651       \PackageWarning{beamerthememetropolis}{%
652         Be aware that titleformat subtitle=allsmallcaps can lead to prob-
```

```
      lems%
653        }
654      },
655      allcaps/.code={%
656        \let\metropolis@subtitleformat\uppercase%
657        \setbeamerfont{subtitle}{shape=\normalfont}%
658        \PackageWarning{beamerthememetropolis}{%
659          Be aware that titleformat subtitle=allcaps can lead to prob-
      lems%
660        }
661      },
662 }
```

titleformat section   Controls the format of the section title.

```
663 \pgfkeys{
664   /metropolis/font/titleformat section/.cd,
665     .is choice,
666     regular/.code={%
667       \let\metropolis@sectiontitleformat\@empty%
668       \setbeamerfont{section title}{shape=\normalfont}%
669     },
670     smallcaps/.code={%
671       \let\metropolis@sectiontitleformat\@empty%
672       \setbeamerfont{section title}{shape=\scshape}%
673     },
674     allsmallcaps/.code={%
675       \let\metropolis@sectiontitleformat\MakeLowercase%
676       \setbeamerfont{section title}{shape=\scshape}%
677       \PackageWarning{beamerthememetropolis}{%
678         Be aware that titleformat section=allsmallcaps can lead to prob-
      lems%
679       }
680     },
681     allcaps/.code={%
682       \let\metropolis@sectiontitleformat\MakeUppercase%
683       \setbeamerfont{section title}{shape=\normalfont}%
684       \PackageWarning{beamerthememetropolis}{%
685         Be aware that titleformat section=allcaps can lead to prob-
      lems%
```

```
686           }
687         },
688 }
```

frametitleformat Control the format of the frame title.

```
689 \pgfkeys{
690   /metropolis/font/titleformat frame/.cd,
691     .is choice,
692     regular/.code={%
693       \let\metropolis@frametitleformat\@empty%
694       \setbeamerfont{frametitle}{shape=\normalfont}%
695     },
696     smallcaps/.code={%
697       \let\metropolis@frametitleformat\@empty%
698       \setbeamerfont{frametitle}{shape=\scshape}%
699     },
700     allsmallcaps/.code={%
701       \let\metropolis@frametitleformat\MakeLowercase%
702       \setbeamerfont{frametitle}{shape=\scshape}%
703       \PackageWarning{beamerthememetropolis}{%
704         Be aware that titleformat frame=allsmallcaps can lead to prob-
   lems%
705       }
706     },
707     allcaps/.code={%
708       \let\metropolis@frametitleformat\MakeUppercase%
709       \setbeamerfont{frametitle}{shape=\normalfont}
710       \PackageWarning{beamerthememetropolis}{%
711         Be aware that titleformat frame=allcaps can lead to prob-
   lems%
712       }
713     },
714 }
```

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

```
715 \pgfkeys{
716   /metropolis/font/.cd,
717     titleformattitle/.code=\pgfkeysalso{titleformat title=#1},
```

```
718   titleformatsubtitle/.code=\pgfkeysalso{titleformat subtitle=#1},
719   titleformatsection/.code=\pgfkeysalso{titleformat section=#1},
720   titleformatframe/.code=\pgfkeysalso{titleformat frame=#1},
721 }
```

metropolis@font@setdefaults   Sets default values for font theme options.

```
722 \newcommand{\metropolis@font@setdefaults}{
723   \pgfkeys{/metropolis/font/.cd,
724     titleformat title=regular,
725     titleformat subtitle=regular,
726     titleformat section=regular,
727     titleformat frame=regular,
728   }
729 }
```

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

```
730 \def\metropolis@titleformat#1{#1}
731 \def\metropolis@subtitleformat#1{#1}
732 \def\metropolis@sectiontitleformat#1{#1}
733 \def\metropolis@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.

```
734 \patchcmd{\beamer@title}%
735   {\def\inserttitle{#2}}%
736   {\def\inserttitle{\metropolis@titleformat{#2}}}%
737   {}%
738   {\PackageError{beamerfontthememetropolis}{Patching title failed}}
739 \patchcmd{\beamer@subtitle}%
740   {\def\insertsubtitle{#2}}%
741   {\def\insertsubtitle{\metropolis@subtitleformat{#2}}}%
742   {}%
743   {\PackageError{beamerfontthememetropolis}{Patching subtitle failed}}
744 \patchcmd{\sectionentry}
745   {\def\insertsectionhead{#2}}
746   {\def\insertsectionhead{\metropolis@sectiontitleformat{#2}}}
```

```
747    {}
748    {\PackageError{beamerfontthememetropolis}{Patching section ti-
       tle failed}}
749 \patchcmd{\beamer@section}
750    {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
751    {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
752       \metropolis@sectiontitleformat{#1}}}}
753    {}
754    {\PackageError{beamerfontthememetropolis}{Patching section ti-
       tle failed}}
755 \patchcmd{\beamer@subsection}
756    {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
757    {\def\insertsubsectionhead{\hyperlink{Navigation\the\c@page}{%
758       \metropolis@sectiontitleformat{#1}}}}
759    {}
760    {\PackageError{beamerfontthememetropolis}{Patching section ti-
       tle failed}}
```

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

```
761 \patchcmd{\beamer@@frametitle}
762    {\beamer@ifempty{#2}{}{%
763       \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
764       \usebeamertemplate*{frametitle continuation}\fi}}%
765    \gdef\beamer@frametitle{#2}%
766    \gdef\beamer@shortframetitle{#1}%
767    }}
768    {\beamer@ifempty{#2}{}{%
769       \gdef\insertframetitle{{\metropolis@frametitleformat{#2}\ifnum%
770       \beamer@autobreakcount>0\relax{}\space%
771       \usebeamertemplate*{frametitle continuation}\fi}}%
772    \gdef\beamer@frametitle{#2}%
773    \gdef\beamer@shortframetitle{#1}%
774    }}
775    {}
776    {\PackageError{beamerfontthememetropolis}{Patching frame title failed}}
```

8.4.5   Process package options

```
777 \metropolis@font@setdefaults
778 \ProcessPgfPackageOptions{/metropolis/font}
```

8.5 METROPOLIS color theme

8.5.1 Package dependencies

```
779 \RequirePackage{pgfopts}
```

8.5.2 Options

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

```
780 \pgfkeys{
781   /metropolis/color/block/.cd,
782     .is choice,
783     transparent/.code=\metropolis@block@transparent,
784     fill/.code=\metropolis@block@fill,
785 }
```

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

```
786 \pgfkeys{
787   /metropolis/color/background/.cd,
788     .is choice,
789     dark/.code=\metropolis@colors@dark,
790     light/.code=\metropolis@colors@light,
791 }
```

etropolis@color@setdefaults   Sets default values for color theme options.

```
792 \newcommand{\metropolis@color@setdefaults}{
793   \pgfkeys{/metropolis/color/.cd,
794     background=light,
795     block=transparent,
796   }
797 }
```

8.5.3 Base colors

```
798 \definecolor{mDarkBrown}{HTML}{604c38}
799 \definecolor{mDarkTeal}{HTML}{575757}
800 \definecolor{mLightBrown}{HTML}{de9533}
801 \definecolor{mLightGreen}{HTML}{14B03D}
```

8.5.4 Base styles

All colors in METROPOLIS are derived from the definitions of `normal text`, `alerted text`, and `example text`.

```
802 \newcommand{\metropolis@colors@dark}{
803   \setbeamercolor{normal text}{%
804     fg=black!2,
805     bg=mDarkTeal
806   }
807   \usebeamercolor[fg]{normal text}
808 }
809 \newcommand{\metropolis@colors@light}{
810   \setbeamercolor{normal text}{%
811     fg=mDarkTeal,
812     bg=black!2
813   }
814 }
815 \setbeamercolor{alerted text}{%
816   fg=mLightBrown
817 }
818 \setbeamercolor{example text}{%
819   fg=mLightGreen
820 }
```

8.5.5 Derived colors

The titles and structural elements (e.g. `itemize` bullets) are set in the same color as `normal text`. This would ideally done by setting `normal text` as a parent style, which we do to set `titlelike`, but this doesn't work for `structure` as its foreground is set explicitly in `beamercolorthemedefault.sty`.

```
821 \setbeamercolor{titlelike}{use=normal text, parent=normal text}
```

```
822 \setbeamercolor{author}{use=normal text, parent=normal text}
823 \setbeamercolor{date}{use=normal text, parent=normal text}
824 \setbeamercolor{institute}{use=normal text, parent=normal text}
825 \setbeamercolor{structure}{use=normal text, fg=normal text.fg}
```

The "primary" palette should be used for the most important navigational elements, and possibly of other elements. METROPOLIS uses it for frame titles and slides.

```
826 \setbeamercolor{palette primary}{%
827   use=normal text,
828   fg=normal text.bg,
829   bg=normal text.fg
830 }
831 \setbeamercolor{frametitle}{%
832   use=palette primary,
833   parent=palette primary
834 }
```

The METROPOLIS 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`.

```
835 \setbeamercolor{progress bar}{%
836   use=alerted text,
837   fg=alerted text.fg,
838   bg=alerted text.fg!50!black!30
839 }
840 \setbeamercolor{title separator}{
841   use=progress bar,
842   parent=progress bar
843 }
844 \setbeamercolor{progress bar in head/foot}{%
845   use=progress bar,
846   parent=progress bar
847 }
848 \setbeamercolor{progress bar in section page}{
849   use=progress bar,
850   parent=progress bar
851 }
```

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.

```
852 \newcommand{\metropolis@block@transparent}{
853   \setbeamercolor{block title}{%
854     use=normal text,
855     fg=normal text.fg,
856     bg=
857   }
858   \setbeamercolor{block body}{
859     bg=
860   }
861 }
862 \newcommand{\metropolis@block@fill}{
863   \setbeamercolor{block title}{%
864     use=normal text,
865     fg=normal text.fg,
866     bg=normal text.bg!80!fg
867   }
868   \setbeamercolor{block body}{
869     use={block title, normal text},
870     bg=block title.bg!50!normal text.bg
871   }
872 }
873 \setbeamercolor{block title alerted}{%
874     use={block title, alerted text},
875     bg=block title.bg,
876     fg=alerted text.fg
877 }
878 \setbeamercolor{block title example}{%
879     use={block title, example text},
880     bg=block title.bg,
881     fg=example text.fg
882 }
883 \setbeamercolor{block body alerted}{use=block body, parent=block body}
884 \setbeamercolor{block body example}{use=block body, parent=block body}
```

46

Footnotes

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

8.5.6   Process package options

```
887 \metropolis@color@setdefaults
888 \ProcessPgfPackageOptions{/metropolis/color}
```

```
889 \mode<all>
```

8.6   Tol `pgfplots` theme

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

```
890 \definecolor{TolDarkPurple}{HTML}{332288}
891 \definecolor{TolDarkBlue}{HTML}{6699CC}
892 \definecolor{TolLightBlue}{HTML}{88CCEE}
893 \definecolor{TolLightGreen}{HTML}{44AA99}
894 \definecolor{TolDarkGreen}{HTML}{117733}
895 \definecolor{TolDarkBrown}{HTML}{999933}
896 \definecolor{TolLightBrown}{HTML}{DDCC77}
897 \definecolor{TolDarkRed}{HTML}{661100}
898 \definecolor{TolLightRed}{HTML}{CC6677}
899 \definecolor{TolLightPink}{HTML}{AA4466}
900 \definecolor{TolDarkPink}{HTML}{882255}
901 \definecolor{TolLightPurple}{HTML}{AA4499}
```

To use these colors, we describe "cycle lists" from which PGF chooses styles for the different series in a chart.

mbarplot cycle   Colors and styles intended for bar charts with up to 12 series.

```
902 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
903   {draw=TolDarkBlue,    fill=TolDarkBlue!70},
904   {draw=TolLightBrown,  fill=TolLightBrown!70},
905   {draw=TolLightGreen,  fill=TolLightGreen!70},
906   {draw=TolDarkPink,    fill=TolDarkPink!70},
```

---

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

```
907    {draw=TolDarkPurple,  fill=TolDarkPurple!70},
908    {draw=TolDarkRed,     fill=TolDarkRed!70},
909    {draw=TolDarkBrown,   fill=TolDarkBrown!70},
910    {draw=TolLightRed,    fill=TolLightRed!70},
911    {draw=TolLightPink,   fill=TolLightPink!70},
912    {draw=TolLightPurple, fill=TolLightPurple!70},
913    {draw=TolLightBlue,   fill=TolLightBlue!70},
914    {draw=TolDarkGreen,   fill=TolDarkGreen!70},
915 }
```

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

```
916 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
917   {TolDarkBlue, mark=*, mark size=1.5pt},
918   {TolLightBrown, mark=square*, mark size=1.3pt},
919   {TolLightGreen, mark=triangle*, mark size=1.5pt},
920   {TolDarkBrown, mark=diamond*, mark size=1.5pt},
921 }
```

However, the above cycle lists are not applied automatically. We still need to define styles — `mlineplot` and `mbarplot` — that the user can apply to the axis of a `pgfplots` chart to use the colors. We'll also take the opportunity to adjust the display of chart axes when these styles are used.

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

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

```
924   mlineplot/.style={
925     mbaseplot,
926     xmajorgrids=true,
927     ymajorgrids=true,
928     major grid style={dotted},
929     axis x line=bottom,
930     axis y line=left,
931     legend style={
932       cells={anchor=west},
933       draw=none
934     },
935     cycle list name=mlineplot cycle,
```

```
936    },
```

mbarplot   A style to apply to the axis of a PGF bar chart. `mbarplot` uses vertical bars by
horizontal mbarplot   default, while `horizontal mbarplot` has horizontal bars as the name implies.
Their shared properties are factored out into the internal style `mbarplot base`.

```
937    mbarplot base/.style={
938      mbaseplot,
939      bar width=6pt,
940      axis y line*=none,
941    },
942    mbarplot/.style={
943      mbarplot base,
944      ybar,
945      xmajorgrids=false,
946      ymajorgrids=true,
947      area legend,
948      legend image code/.code={%
949        \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
950      },
951      cycle list name=mbarplot cycle,
952    },
953    horizontal mbarplot/.style={
954      mbarplot base,
955      xmajorgrids=true,
956      ymajorgrids=false,
957      xbar stacked,
958      area legend,
959      legend image code/.code={%
960        \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
961      },
962      cycle list name=mbarplot cycle,
963    },
```

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

```
964    mbaseplot/.style={
965      legend style={
966        draw=none,
967        fill=none,
968        cells={anchor=west},
```

```
969      },
970      x tick label style={
971        font=\footnotesize
972      },
973      y tick label style={
974        font=\footnotesize
975      },
976      legend style={
977        font=\footnotesize
978      },
979      major grid style={
980        dotted,
981      },
982      axis x line*=bottom,
983    },
984    disable thousands separator/.style={
985      /pgf/number format/.cd,
986        1000 sep={}
987    },
988 }
```