Modern Beamer Presentations with the METROPOLIS package

Matthias Vogelgesang

matthias.vogelgesang@gmail.com

v0.x.x

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 is now overused and can be a little cluttered, and 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 XeLaTeX 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 GitHub

Installing METROPOLIS, like any Beamer theme, 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{m}` 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, manual, and demo presentation.

`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.

`make ctan` creates a package for CTAN distribution.

2.2 Installing the Debian Package

As an alternative users of Debian or Ubuntu can also install this .deb package containing the theme files as well as the Fira Sans font files.

2.3 A Minimal Example

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

```
\documentclass{beamer}
\usetheme{m}                       % 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

- XeLaTeX

- Fira Sans and Mono font
- TikZ

The `Fira Sans` font is not a hard dependency. METROPOLIS will try to load the font and use it if it is installed, but if not it will just use the standard font. Depending on the Linux distribution, the packaged name of `Fira Sans` might be `Fira Sans OT` instead of `Fira Sans`. METROPOLIS will check for this name too.

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:m -o
    output.pdf input.md
```

3   Customization

3.1   Package options

The theme provides a number of options. The options use a key=value interface. So every option is controlled by a key its value. To use an option you can either provide a comma separated list of options when invoking MTHEME in the preamble of the presentation.

```
\usetheme[<key=value list>]{m}
```

Or you can set them at any time with the `\metroset` macro.

```
\metroset[<key=value list>]
```

To set an option on a specific sub-package only you have to add the corresponding prefix (inner, outer, color), e.g.

```
\metroset[inner/block=fill]
```

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

key   *list of possible values* . . . . default value

A short description of the option.

Although the options are grouped into the corresponding packages every option can and in most cases should be set on the main theme directly. If an option is listed in multiple sub-packages, setting it on the main theme will set the option on every sub-package accordingly.

3.1.1   Main theme

everytitleformat   *regular, lowercase, uppercase* . . . . lowercase

Shortcut option to change the case style of all titles together.

plaintitleformat   *regular, lowercase, uppercase* . . . . lowercase

Control the case style of the plain title.

3.1.2   Inner theme

block   *transparent, fill* . . . . transparent

This option controls the block background. It can either be filled with a light grey or be transparent.

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

Disable section pages at all, typeset centered section title or add a thin progress bar below the centered section title.

titleformat   *regular, lowercase, uppercase* . . . . lowercase

Control the case style of the title.

sectiontitleformat   *regular, lowercase, uppercase* . . . . lowercase

Control the case style of the section title.

3.1.3 Outer theme

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

In the bottom right corner of each frame the current frame number is displayed. This can be disabled or the total framenumber can be added additionally.

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

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

frametitleformat *regular, lowercase, uppercase* . . . . lowercase

Control the case style of the frame title.

3.1.4 Color theme

block *transparent, fill* . . . . transparent

This option controls the block background. It can either be filled with a light grey or be transparent.

background *dark, light* . . . . light

This option defines whether the background shall be dark and the foreground be light or vice versa.

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)
- `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. Yet this can be easily changed using the standard font selection commands of the fontspec package. So if you for example prefer the Ubuntu font family just add the following two commands after loading the METROPOLIS theme.

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

3.4   Commands

The \plain{title=[]}{<body>} command sets a slide in plain dark colors which can be useful to focus attention on a single sentence or image.

3.5   Paul Tol's colors: a pgfplots theme

A good presentation uses colors that are

- distinct from each other as much as possible, and
- distinct from black and white,
- under many different lighting and display environments, and
- to color-blind viewers,
- all 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. Use the `mlineplot` key to plot line data and `mbarplot` or horizontal `mbarplot` to plot bar charts.

4 Known Issues

The `\plain` command does not work if you override the METROPOLIS color theme with the default beamer color theme `fly`.

5 License

The theme itself 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 the presentation that you create with the theme.

6 Implementation

6.1 METROPOLIS main 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.

Load the required packages.

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

9

6.1.1 Options

`\metroset`   First of all we define a macro for the user to set options.

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

Then we need to pass the unknown options to the sub-packages.

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

We have to forwarded keys that affect multiple sub-packages manually.

```
10   block/.code=\pgfkeysalso{
11     inner/block=#1,
12     color/block=#1,
13   },
14 }
```

`plaintitleformat`   Control the case style of the plain title

```
15 \pgfkeys{
16   /metropolis/plaintitleformat/.cd,
17     .is choice,
18     regular/.code=\renewcommand{\@metropolis@plaintitleformat}{#1},
19     lowercase/.code={%
20       \renewcommand{\@metropolis@plaintitleformat}{\MakeLowercase{#1}}
21     },
22     uppercase/.code={%
23       \renewcommand{\@metropolis@plaintitleformat}{\MakeUppercase{#1}}
24     },
25 }
```

`everytitleformat`   Control the case style of the every title

```
26 \pgfkeys{
27   /metropolis/everytitleformat/.code=\pgfkeysalso{
28       inner/titleformat=#1,
```

```
29      inner/sectiontitleformat=#1,
30      outer/frametitleformat=#1,
31      plaintitleformat=#1,
32    }
33 }
```

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

```
34 \pgfkeys{/metropolis/.cd,
35   usetitleprogressbar/.code=\pgfkeysalso{outer/progressbar=frametitle},
36   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
37   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
38   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
39   darkcolors/.code=\pgfkeysalso{color/background=dark},
40   blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
41 }
```

Set default values for options.

```
42 \newcommand{\@metropolis@setdefaults}{
43   \pgfkeys{/metropolis/.cd,
44     plaintitleformat=lowercase,
45   }
46 }
```

6.1.2   Component sub-packages

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

```
47 \useinnertheme{metropolis}
48 \useoutertheme{metropolis}
49 \usecolortheme{metropolis}
50 \usefonttheme{metropolis}
```

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

```
51 \AtEndPreamble{%
52   \@ifpackageloaded{pgfplots}{%
53     \RequirePackage{pgfplotsthemetol}
```

```
54    }{}
55 }
```

6.1.3   Custom commands

We define custom commands in this package as their proper usage may depend on multiple sub-packages.

metropolis@plaintitleformat   Define a hook to change the case format of the plain title.

```
56 \def\@metropolis@plaintitleformat#1{#1}
```

\plain   Creates a plain frame with dark background, suitable for displaying images or a few words.

```
57 \newcommand{\plain}[2][]{%
58   \begingroup
59     \setbeamercolor{background canvas}{
60       use=palette primary,
61       parent=palette primary
62     }
63     \begin{frame}[c]{#1}
64       \begin{center}
65         \usebeamercolor[fg]{palette primary}
66         \usebeamerfont{section title}
67         \@metropolis@plaintitleformat{#2}
68       \end{center}
69     \end{frame}
70   \endgroup
71 }
```

\mreducelistspacing

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

Process package options

```
73 \@metropolis@setdefaults
74 \ProcessPgfOptions{/metropolis}
```

6.2   SMALL CAPS 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.

Load required packages.

```
75 \RequirePackage{etoolbox}
76 \RequirePackage{calc}
77 \RequirePackage{pgfopts}
78 \RequirePackage{tikz}
```

6.2.1   Options

`block`   This option controls the block style.

```
79 \pgfkeys{
80   /metropolis/inner/block/.cd,
81     .is choice,
82     transparent/.code=\setlength{\@metropolis@blockskip}{0ex},
83     fill/.code=\setlength{\@metropolis@blockskip}{1ex},
84 }
```

`titleformat`   Control the case style of the title

```
85 \pgfkeys{
86   /metropolis/inner/titleformat/.cd,
87     .is choice,
88     regular/.code=\renewcommand{\@metropolis@titleformat}{},
89     lowercase/.code={%
90       \renewcommand{\@metropolis@titleformat}{\MakeLowercase}
91     },
92     uppercase/.code={%
93       \renewcommand{\@metropolis@titleformat}{\MakeUppercase}
```

```
94       },
95 }
```

sectiontitleformat — Control the case style of the section title

```
96 \pgfkeys{
97   /metropolis/inner/sectiontitleformat/.cd,
98     .is choice,
99     regular/.code=\renewcommand{\@metropolis@sectiontitleformat}{},
100    lowercase/.code={%
101      \renewcommand{\@metropolis@sectiontitleformat}{\MakeLowercase}
102    },
103    uppercase/.code={%
104      \renewcommand{\@metropolis@sectiontitleformat}{\MakeUppercase}
105    },
106 }
```

sectionpage — The `sectionpage` option defines the behaviour of the sectionpage.

```
107 \pgfkeys{
108   /metropolis/inner/sectionpage/.cd,
109     .is choice,
110     none/.code=\@metropolis@sectionpage@none,
111     simple/.code=\@metropolis@sectionpage@simple,
112     progressbar/.code=\@metropolis@sectionpage@progressbar,
113 }
```

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

```
114 \newcommand{\@metropolis@inner@setdefaults}{
115   \pgfkeys{/metropolis/inner/.cd,
116     sectionpage=progressbar,
117     block=transparent,
118     titleformat=lowercase,
119     sectiontitleformat=lowercase,
120   }
121 }
```

6.2.2 Title page

`\@metropolis@titleformat` Define hooks to change the case format of the titles.

```
122 \def\@metropolis@titleformat#1{#1}
123 \def\@metropolis@sectiontitleformat#1{#1}
```

To make the `\MakeLowercase` and `\MakeUppercase` macros work in the sectiontitle we have to patch `\sectionentry` and `\beamer@section`. This solution was suggested by Enrico Gregorio in an answer to this StackExchange question.

```
124 \patchcmd{\sectionentry}
125   {\def\insertsectionhead{#2}}
126   {\def\insertsectionhead{\@metropolis@sectiontitleformat{#2}}}
127   {}
128   {\PackageError{beamerinnerthememetropolis}{Patching section ti-
      tle failed.}}
129 \patchcmd{\beamer@section}
130   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
131   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{\@metropolis@sectiontit
132   {}
133   {\PackageError{beamerinnerthememetropolis}{Patching section ti-
      tle failed.}}
```

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.

```
134 \setbeamertemplate{title page}{
135   \begin{minipage}[b][\paperheight]{\textwidth}
136     \ifx\inserttitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
137     \vfill%
138     \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
139     \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
140     \usebeamertemplate*{title separator}
```

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

author. This solution was suggested by Enrico Gregorio in an answer to this Stack Exchange question.

```
141     \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
142     \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
143     \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
144     \vfill
145     \vspace*{1mm}
146   \end{minipage}
147 }
```

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.

```
148 \def\maketitle{%
149   \ifbeamer@inframe
150     \titlepage
151   \else
152     \frame[plain]{\titlepage}
153   \fi
154 }
155 \def\titlepage{%
156   \usebeamertemplate{title page}
157 }
```

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

```
158 \setbeamertemplate{title graphic}{
159   \vbox to 0pt {
160     \vspace*{2em}
161     \inserttitlegraphic%
162   }%
163   \nointerlineskip%
164 }
```

16

title Set the title on the title page.

```
165 \setbeamertemplate{title}{
166   \raggedright%
167   \linespread{1.0}%
168   \@metropolis@titleformat{\inserttitle}%
169   \par%
170   \vspace*{0.5em}
171 }
```

subtitle Set the subtitle on the title page.

```
172 \setbeamertemplate{subtitle}{
173   \insertsubtitle%
174   \par%
175   \vspace*{0.5em}
176 }
```

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

```
177 \setbeamertemplate{title separator}{
178   \begin{tikzpicture}
179     \draw[fg] (0, 0) -- (\textwidth, 0);
180   \end{tikzpicture}%
181   \par%
182 }
```

author Set the author on the title page.

```
183 \setbeamertemplate{author}{
184   \vspace*{2em}
185   \insertauthor%
186   \par%
187   \vspace*{0.25em}
188 }
```

date Set the date on the title page.

```
189 \setbeamertemplate{date}{
190   \insertdate%
191   \par%
```

17

```
192 }
```

institute   Set the institute on the title page.

```
193 \setbeamertemplate{institute}{
194   \vspace*{3mm}
195   \insertinstitute%
196   \par%
197 }
```

6.2.3 Section page

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

```
198 \newcommand{\@metropolis@sectionpage@none}{
199   \AtBeginSection{
200     % intenionally empty
201   }
202 }
203 \defbeamertemplate{section page}{simple}{
204   \centering
205   \usebeamercolor[fg]{section title}
206   \usebeamerfont{section title}
207   \insertsectionhead\\
208 }
209 \newcommand{\@metropolis@sectionpage@simple}{
210   \setbeamertemplate{section page}[simple]
211   \AtBeginSection{
212     \ifbeamer@inframe
213       \sectionpage
214     \else
215       \frame[plain,c]{\sectionpage}
216     \fi
217   }
218 }
219 \defbeamertemplate{section page}{progressbar}{
220   \centering
221   \begin{minipage}{22em}
222     \usebeamercolor[fg]{section title}
223     \usebeamerfont{section title}
```

```
224     \insertsectionhead\\[-1ex]
225     \usebeamertemplate*{progress bar in section page}
226   \end{minipage}
227   \par
228 }
229 \newcommand{\@metropolis@sectionpage@progressbar}{
230   \setbeamertemplate{section page}[progressbar]
231   \AtBeginSection{
232     \ifbeamer@inframe
233       \sectionpage
234     \else
235       \frame[plain,c]{\sectionpage}
236     \fi
237   }
238 }
```

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`.

```
239 \newlength{\metropolis@progressonsectionpage}
240 \setbeamertemplate{progress bar in section page}{
241   \setlength{\metropolis@progressonsectionpage}{%
242     \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
243   }%
244   \begin{tikzpicture}
245     \draw[bg, fill=bg] (0,0) rectangle (\textwidth, 0.4pt);
246     \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressonsectionpage, 0.4pt);
247   \end{tikzpicture}%
248 }
```

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.

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

6.2.4 Block environments

Regular block environment

```
250 \newlength{\@metropolis@blockskip}
251 \setbeamertemplate{block begin}{%
252   \vspace*{1ex}
253   \begin{beamercolorbox}[%
254     ht=2.4ex,
255     dp=1ex,
256     leftskip=\@metropolis@blockskip,
257     rightskip=\@metropolis@blockskip]{block title}
258       \usebeamerfont*{block title}\insertblocktitle%
259   \end{beamercolorbox}%
260   \vspace*{-1pt}
261   \usebeamerfont{block body}%
262   \begin{beamercolorbox}[%
263     dp=1ex,
264     leftskip=\@metropolis@blockskip,
265     rightskip=\@metropolis@blockskip,
266     vmode]{block body}%
267 }
268 \setbeamertemplate{block end}{%
269   \end{beamercolorbox}
270   \vspace*{0.2ex}
271 }
```

Alerted block environment

```
272 \setbeamertemplate{block alerted begin}{%
273   \vspace*{1ex}
274   \begin{beamercolorbox}[%
275     ht=2.4ex,
276     dp=1ex,
277     leftskip=\@metropolis@blockskip,
278     rightskip=\@metropolis@blockskip]{block title alerted}
279       \usebeamerfont*{block title alerted}\insertblocktitle%
280   \end{beamercolorbox}%
```

```
281  \vspace*{-1pt}
282  \usebeamerfont{block body alerted}%
283  \begin{beamercolorbox}[%
284    dp=1ex,
285    leftskip=\@metropolis@blockskip,
286    rightskip=\@metropolis@blockskip,
287    vmode]{block body}%
288 }
289 \setbeamertemplate{block alerted end}{%
290  \end{beamercolorbox}
291  \vspace*{0.2ex}
292 }
```

Example block environment

```
293 \setbeamertemplate{block example begin}{%
294  \vspace*{1ex}
295  \begin{beamercolorbox}[%
296    ht=2.4ex,
297    dp=1ex,
298    leftskip=\@metropolis@blockskip,
299    rightskip=\@metropolis@blockskip]{block title example}
300      \usebeamerfont*{block title example}\insertblocktitle%
301  \end{beamercolorbox}%
302  \vspace*{-1pt}
303  \usebeamerfont{block body example}%
304  \begin{beamercolorbox}[%
305    dp=1ex,
306    leftskip=\@metropolis@blockskip,
307    rightskip=\@metropolis@blockskip,
308    vmode]{block body}%
309 }
310 \setbeamertemplate{block example end}{%
311  \end{beamercolorbox}
312  \vspace*{0.2ex}
313 }
```

6.2.5 Lists and floats

```
314 \setbeamertemplate{itemize items}{\textbullet}
```

```
315 \setbeamertemplate{caption label separator}{: }
316 \setbeamertemplate{caption}[numbered]
```

6.2.6   Footnotes

```
317 \setbeamertemplate{footnote}{%
318   \parindent 0em\noindent%
319   \raggedright
320   \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotet
321 }
```

6.2.7   Text and spacing settings

```
322 \setlength{\parskip}{0.5em}
323 \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.

```
324 \define@key{beamerframe}{c}[true]{% centered
325   \beamer@frametopskip=0pt plus 1fill\relax%
326   \beamer@framebottomskip=0pt plus 1fill\relax%
327   \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
328   \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
329   \def\beamer@initfirstlineunskip{}%
330 }
```

Process package options

```
331 \@metropolis@inner@setdefaults
332 \ProcessPgfPackageOptions{/metropolis/inner}
```

6.3   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.

Load required packages.

```
333 \RequirePackage{etoolbox}
```

```
334 \RequirePackage{calc}
335 \RequirePackage{pgfopts}
```

6.3.1 Options

numbering   This option controls the page numbering.

```
336 \pgfkeys{
337   /metropolis/outer/numbering/.cd,
338     .is choice,
339     none/.code=\setbeamertemplate{frame numbering}[none],
340     counter/.code=\setbeamertemplate{frame numbering}[counter],
341     fraction/.code=\setbeamertemplate{frame numbering}[fraction],
342 }
```

progressbar   This option controls the progressbar.

```
343 \pgfkeys{
344   /metropolis/outer/progressbar/.cd,
345     .is choice,
346     none/.code={%
347       \setbeamertemplate{headline}[plain]
348       \setbeamertemplate{frametitle}[plain]
349       \setbeamertemplate{footline}[plain]
350     },
351     head/.code={\pgfkeys{/metropolis/outer/progressbar=none}
352       \addtobeamertemplate{headline}{}{\usebeamertemplate*{progress bar in head-
  /foot}}
353     },
354     frametitle/.code={\pgfkeys{/metropolis/outer/progressbar=none}
355       \addtobeamertemplate{frametitle}{}{\usebeamertemplate*{progress bar in head-
  /foot}}
356     },
357     foot/.code={\pgfkeys{/metropolis/outer/progressbar=none}
358       \addtobeamertemplate{footline}{}{\usebeamertemplate*{progress bar in head-
  /foot}}
359     },
360 }
```

frametitleformat   Control the case style of the frame title

```
361 \pgfkeys{
362   /metropolis/outer/frametitleformat/.cd,
363     .is choice,
364     regular/.code={%
365       \renewcommand{\@metropolis@frametitleformat}{}%
366       \renewcommand{\@metropolis@frametitlestrut}{%
367         \rule{0pt}{\heightof{ABCDEFGHIJKLMNOPQRSTUVWXYZ}}
368       }
369     },
370     lowercase/.code={%
371       \renewcommand{\@metropolis@frametitleformat}{\MakeLowercase}%
372       \renewcommand{\@metropolis@frametitlestrut}{%
373         \rule{0pt}{\heightof{abcdefghijklmnopqrstuvwxyz}}
374       }
375     },
376     uppercase/.code={%
377       \renewcommand{\@metropolis@frametitleformat}{\MakeUppercase}%
378       \renewcommand{\@metropolis@frametitlestrut}{%
379         \rule{0pt}{\heightof{ABCDEFGHIJKLMNOPQRSTUVWXYZ}}
380       }
381     },
382 }
```

\@metropolis@outer@setdefaults   Set default values for outer theme options.

```
383 \newcommand{\@metropolis@outer@setdefaults}{
384   \pgfkeys{/metropolis/outer/.cd,
385     numbering=counter,
386     progressbar=none,
387     frametitleformat=lowercase,
388   }
389 }
```

6.3.2   Head and footline

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

```latex
390 \setbeamertemplate{navigation symbols}{}
```

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

```latex
391 \defbeamertemplate{frame numbering}{none}{}
392 \defbeamertemplate{frame numbering}{counter}{\insertframenumber}
393 \defbeamertemplate{frame numbering}{fraction}{
394   \insertframenumber/\inserttotalframenumber
395 }

396 \defbeamertemplate{headline}{plain}{}
397 \defbeamertemplate{footline}{plain}{%
398   \begin{beamercolorbox}[wd=\textwidth, sep=3ex]{footline}%
399     \hfill%
400     \usebeamerfont{page number in head/foot}%
401     \usebeamertemplate*{frame numbering}
402   \end{beamercolorbox}%
403 }
```

6.3.3   Frametitle

metropolis@frametitleformat   Define a hook to change the case format of the frame title.

```latex
404 \def\@metropolis@frametitleformat#1{#1}
```

To make the **\MakeLowercase** and **\MakeUppercase** macros work in the frame title we have to patch **\beamer@@frametitle**. This solution was suggested by Enrico Gregorio in an answer to this StackExchange question.

```latex
405 \patchcmd{\beamer@@frametitle}
406   {\beamer@ifempty{#2}{}{%
407     \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}\space\usebea
  tinuation}\fi}}%
408     \gdef\beamer@frametitle{#2}%
409     \gdef\beamer@shortframetitle{#1}%
410     }}
411   {\beamer@ifempty{#2}{}{%
412     \gdef\insertframetitle{{\@metropolis@frametitleformat{#2}\ifnum\beamer@autobr
  tinuation}\fi}}%
```

```
413     \gdef\beamer@frametitle{#2}%
414     \gdef\beamer@shortframetitle{#1}%
415     }}
416   {}
417   {\PackageError{beameroutertheme metropolis}{Patching frame title failed.}}
```

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

```
418 \newlength{\@metropolis@frametitlestrut}
419 \defbeamertemplate{frametitle}{plain}{%
420   \nointerlineskip%
421   \begin{beamercolorbox}[%
422       wd=\paperwidth,%
423       sep=1.5ex,%
424     ]{frametitle}%
425   \@metropolis@frametitlestrut\insertframetitle\@metropolis@frametitlestrut%
426   \end{beamercolorbox}%
427 }
```

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`.

```
428 \newlength{\metropolis@progressinheadfoot}
429 \setbeamertemplate{progress bar in head/foot}{
430   \nointerlineskip
431   \setlength{\metropolis@progressinheadfoot}{%
432     \paperwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
433   }%
434   \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}
435     \begin{tikzpicture}
436       \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
437       \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
438     \end{tikzpicture}%
439   \end{beamercolorbox}
440 }
```

Process package options

```
441 \@metropolis@outer@setdefaults
442 \ProcessPgfPackageOptions{/metropolis/outer}
```

6.4 METROPOLIS font theme

Load required packages.

```
443 \RequirePackage{etoolbox}
444 \RequirePackage{ifxetex}
445 \RequirePackage{ifluatex}
```

6.4.1 Load Fira font

If the presentation is compiled with XeLaTeX or LuaLaTeX the fontspec package will be loaded.

```
446 \ifboolexpr{bool {xetex} or bool {luatex}}{
447   \RequirePackage[no-math]{fontspec}
448   \defaultfontfeatures{Mapping=tex-text}
```

To simplify the check whether the `Fira` fonts are installed, a set macros is defined.

\checkfont Checks if a font is installed and increases `fontsnotfound` counter if not.

```
449   \newcounter{fontsnotfound}
450   \newcommand{\checkfont}[1]{%
451     \suppressfontnotfounderror=1%
452     \font\x = "#1" at 10pt
453     \ifx\x\nullfont%
454       \stepcounter{fontsnotfound}%
455     \fi%
456     \suppressfontnotfounderror=0%
457   }
458
```

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

```
459   \newcommand{\iffontsexist}[3]{%
460     \setcounter{fontsnotfound}{0}%
461     \expandafter\forcsvlist\expandafter%
462     \checkfont\expandafter{#1}%
```

```
463    \ifnum\value{fontsnotfound}=0%
464        #2%
465    \else%
466        #3%
467    \fi%
468  }
```

Using the previously defined macros it is tried to load the **Fira** fonts. First the default **Fira** name will be tried. Second the **Fira** fonts with 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.

```
469  \iffontsexist{Fira Sans Light,%
470                Fira Sans Light Italic,%
471                Fira Sans,%
472                Fira Sans Italic}{%
473    \setsansfont[BoldFont={Fira Sans}]{Fira Sans Light}%
474  }{%
475    \iffontsexist{Fira Sans Light OT,%
476                  Fira Sans Light Italic OT,%
477                  Fira Sans OT,%
478                  Fira Sans Italic OT}{%
479      \setsansfont[BoldFont={Fira Sans OT}]{Fira Sans Light OT}%
480    }{%
481      \PackageWarning{beamerthemem}{%
482        Could not find Fira Sans fonts.%
483      }
484    }
485  }
486  \iffontsexist{Fira Mono, Fira Mono Bold}{%
487    \setmonofont{Fira Mono}%
488  }{%
489    \iffontsexist{Fira Mono OT, Fira Mono Bold OT}{%
490      \setmonofont{Fira Mono OT}%
491    }{%
492      \PackageWarning{beamerthemem}{%
493        Could not find Fira Mono fonts.%
494      }
495    }
496  }
```

```
497  \AtBeginEnvironment{tabular}{%
498    \addfontfeature{Numbers={Monospaced}}}%
499  }
500 }{%
501  \PackageWarning{beamerthemem}{%
502    You need to compile with XeLaTeX or LuaLaTeX to use the Fira fonts.%
503  }
504 }
```

6.4.2   General font definitions

```
505 \setbeamerfont{title}{size=\Large,%
506                      series=\bfseries,%
507                      shape=\scshape}
508 \setbeamerfont{author}{size=\small}
509 \setbeamerfont{date}{size=\small}
510 \setbeamerfont{section title}{size=\Large,%
511                            series=\bfseries,%
512                            shape=\scshape}
513 \setbeamerfont{block title}{size=\normalsize,%
514                          series=\bfseries}
515 \setbeamerfont{block title alerted}{size=\normalsize,%
516                                  series=\bfseries}
517 \setbeamerfont*{subtitle}{size=\large,%
518                        shape=\scshape}
519 \setbeamerfont{frametitle}{size=\large,%
520                         series=\bfseries,%
521                         shape=\scshape}
522 \setbeamerfont{caption}{size=\small}
523 \setbeamerfont{caption name}{series=\bfseries}
524 \setbeamerfont{description item}{series=\bfseries}
525 \setbeamerfont{page number in head/foot}{size=\scriptsize}
526 \setbeamerfont{bibliography entry author}{size=\normalsize,%
527                                        series=\normalfont}
528 \setbeamerfont{bibliography entry title}{size=\normalsize,%
529                                       series=\bfseries}
530 \setbeamerfont{bibliography entry location}{size=\normalsize,%
531                                          series=\normalfont}
532 \setbeamerfont{bibliography entry note}{size=\small,%
533                                      series=\normalfont}
```

```
534 \linespread{1.15}
```

6.5 METROPOLIS color theme

Load required packages.

```
535 \RequirePackage{pgfopts}
```

6.5.1 Options

block This option controls whether the blocks are filled or transparent.

```
536 \pgfkeys{
537   /metropolis/color/block/.cd,
538     .is choice,
539     transparent/.code=\@metropolis@block@transparent,
540     fill/.code=\@metropolis@block@fill,
541 }
```

colors Defines whether the background shall be dark and the foreground be light or
        vice versa

```
542 \pgfkeys{
543   /metropolis/color/background/.cd,
544     .is choice,
545     dark/.code=\@metropolis@colors@dark,
546     light/.code=\@metropolis@colors@light,
547 }
```

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

```
548 \newcommand{\@metropolis@color@setdefaults}{
549   \pgfkeys{/metropolis/color/.cd,
550     background=light,
551     block=transparent,
552   }
553 }
```

30

6.5.2 Base colors

```
554 \definecolor{mDarkBrown}{HTML}{604c38}
555 \definecolor{mDarkTeal}{HTML}{23373b}
556 \definecolor{mLightBrown}{HTML}{EB811B}
557 \definecolor{mLightGreen}{HTML}{14B03D}
```

6.5.3 Base styles

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

```
558 \newcommand{\@metropolis@colors@dark}{
559   \setbeamercolor{normal text}{%
560     fg=black!2,
561     bg=mDarkTeal
562   }
563 }
564 \newcommand{\@metropolis@colors@light}{
565   \setbeamercolor{normal text}{%
566     fg=mDarkTeal,
567     bg=black!2
568   }
569 }
570 \setbeamercolor{alerted text}{%
571   fg=mLightBrown
572 }
573 \setbeamercolor{example text}{%
574   fg=mLightGreen
575 }
```

6.5.4 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`.

```
576 \setbeamercolor{titlelike}{use=normal text, parent=normal text}
577 \setbeamercolor{author}{use=normal text, parent=normal text}
```

```
578 \setbeamercolor{date}{use=normal text, parent=normal text}
579 \setbeamercolor{institute}{use=normal text, parent=normal text}
580 \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. The METROPOLIS theme uses it for frame titles and slides.

```
581 \setbeamercolor{palette primary}{%
582   use=normal text,
583   fg=normal text.bg,
584   bg=normal text.fg
585 }
586 \setbeamercolor{frametitle}{%
587   use=palette primary,
588   parent=palette primary
589 }
```

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`.

```
590 \setbeamercolor{progress bar}{%
591   use=alerted text,
592   fg=alerted text.fg,
593   bg=normal text.bg!50!normal text.fg
594 }
595 \setbeamercolor{title separator}{
596   use=progress bar,
597   parent=progress bar
598 }
599 \setbeamercolor{progress bar in head/foot}{%
600   use=progress bar,
601   parent=progress bar
602 }
603 \setbeamercolor{progress bar in section page}{
604   use=progress bar,
605   parent=progress bar
606 }
```

Blocks

```
607 \newcommand{\@metropolis@block@transparent}{
608   \setbeamercolor{block title}{use=normal text, parent=normal text}
609 }
610 \newcommand{\@metropolis@block@fill}{
611   \setbeamercolor{block title}{%
612     use=normal text,
613     fg=normal text.fg,
614     bg=normal text.bg!80!fg
615   }
616 }
617 \setbeamercolor{block title alerted}{%
618     use={block title, alerted text},
619     bg=block title.bg,
620     fg=alerted text.fg
621 }
622 \setbeamercolor{block title example}{%
623     use={block title, example text},
624     bg=block title.bg,
625     fg=example text.fg
626 }
627 \setbeamercolor{block body alerted}{use=block body, parent=block body}
628 \setbeamercolor{block body example}{use=block body, parent=block body}
629 \setbeamercolor{block body}{
630   use={block title, normal text},
631   bg=block title.bg!50!normal text.bg
632 }
```

Footnotes

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

Process package options

```
635 \@metropolis@color@setdefaults
636 \ProcessPgfPackageOptions{/metropolis/color}

637 \mode<all>
```

6.6 Tol `pgfplots` theme

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

```
638 \definecolor{TolDarkPurple}{HTML}{332288}
639 \definecolor{TolDarkBlue}{HTML}{6699CC}
640 \definecolor{TolLightBlue}{HTML}{88CCEE}
641 \definecolor{TolLightGreen}{HTML}{44AA99}
642 \definecolor{TolDarkGreen}{HTML}{117733}
643 \definecolor{TolDarkBrown}{HTML}{999933}
644 \definecolor{TolLightBrown}{HTML}{DDCC77}
645 \definecolor{TolDarkRed}{HTML}{661100}
646 \definecolor{TolLightRed}{HTML}{CC6677}
647 \definecolor{TolLightPink}{HTML}{AA4466}
648 \definecolor{TolDarkPink}{HTML}{882255}
649 \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.

```
650 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
651   {draw=TolDarkBlue,    fill=TolDarkBlue!70},
652   {draw=TolLightBrown,  fill=TolLightBrown!70},
653   {draw=TolLightGreen,  fill=TolLightGreen!70},
654   {draw=TolDarkPink,    fill=TolDarkPink!70},
655   {draw=TolDarkPurple,  fill=TolDarkPurple!70},
656   {draw=TolDarkRed,     fill=TolDarkRed!70},
657   {draw=TolDarkBrown,   fill=TolDarkBrown!70},
658   {draw=TolLightRed,    fill=TolLightRed!70},
659   {draw=TolLightPink,   fill=TolLightPink!70},
660   {draw=TolLightPurple, fill=TolLightPurple!70},
661   {draw=TolLightBlue,   fill=TolLightBlue!70},
662   {draw=TolDarkGreen,   fill=TolDarkGreen!70},
663 }
```

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

---

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

```
664 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
665   {TolDarkBlue, mark=*, mark size=1.5pt},
666   {TolLightBrown, mark=square*, mark size=1.3pt},
667   {TolLightGreen, mark=triangle*, mark size=1.5pt},
668   {TolDarkBrown, mark=diamond*, mark size=1.5pt},
669 }
```

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.

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

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

```
672   mlineplot/.style={
673     mbaseplot,
674     xmajorgrids=true,
675     ymajorgrids=true,
676     major grid style={dotted},
677     axis x line=bottom,
678     axis y line=left,
679     legend style={
680       cells={anchor=west},
681       draw=none
682     },
683     cycle list name=mlineplot cycle,
684   },
```

`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`.

```
685   mbarplot base/.style={
686     mbaseplot,
687     bar width=6pt,
688     axis y line*=none,
689   },
690   mbarplot/.style={
```

35

```
691    mbarplot base,
692    ybar,
693    xmajorgrids=false,
694    ymajorgrids=true,
695    area legend,
696    legend image code/.code={%
697      \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
698    },
699    cycle list name=mbarplot cycle,
700  },
701  horizontal mbarplot/.style={
702    mbarplot base,
703    xmajorgrids=true,
704    ymajorgrids=false,
705    xbar stacked,
706    area legend,
707    legend image code/.code={%
708      \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
709    },
710    cycle list name=mbarplot cycle,
711  },
```

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

```
712  mbaseplot/.style={
713    legend style={
714      draw=none,
715      fill=none,
716      cells={anchor=west},
717    },
718    x tick label style={
719      font=\footnotesize
720    },
721    y tick label style={
722      font=\footnotesize
723    },
724    legend style={
725      font=\footnotesize
726    },
727    major grid style={
```

```
728        dotted,
729      },
730      axis x line*=bottom,
731    },
732    disable thousands separator/.style={
733      /pgf/number format/.cd,
734        1000 sep={}
735    },
736 }
```