

Modern Beamer Presentations with the MTHEME package

Matthias Vogelgesang
matthias.vogelgesang@gmail.com

v0.x.x

1 Introduction

Beamer is an awesome way to make presentations with LaTeX. But the stock themes do not necessarily look particularly nice and the custom themes often scream “Beamer” at first sight. The goal of MTHEME is to provide a modern Beamer theme with minimal visual noise. It provides section slides with a neat progress bar and it is intended to be used with [Fira Sans](#), a gorgeous typeface commissioned by Mozilla and designed by [Carrois](#). Hence to get the best results you should have installed the Fira typeface and use XeTeX to typeset your slides. Nevertheless this is no hard dependency. The theme also works fine with pdfTeX and the Computer Modern typeface.

The codebase is maintained on [GitHub](#). So if you have issues, find mistakes in the manual or want to contribute – to make the theme even better – get in touch there.

2 Getting Started

2.1 Installation

The `MTHHEME` uses Make as build system. Hence the installation is very straight forward. Simply type

```
$ make
$ make install
```

in the top directory and all the files will be created and installed on your computer. The complete list of make rules is as follows:

all

Build the theme, the manual and the demo presentation.

install

Install the theme into your local texmf folder.

uninstall

Remove the theme from your local texmf folder.

sty

Create the package files.

doc

Build the documentation.

demo

Build the demo presentation.

demo-min

Build the minimal demo presentation.

ctan

Create a package for CTAN distribution.

2.2 Dependencies

- XeLaTeX
- [Fira Sans](#) and Mono font

- TikZ

Depending on the Linux distribution, the packaged name of **Fira Sans** might be **Fira Sans OT** instead of **Fira Sans**. In that case, you may have to edit `beamerfontthememetropolis.dtx`. You may also need to install Fira Sans; see the `contrib/` directory for more. 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 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
```

2.4 A Minimal Example

To get started with the theme is very simple. The following code shows a minimal example of a Beamer presentation using the `MTHEME`.

```
\documentclass[10pt]{beamer}
\usetheme{m}                    % load mtheme
\title{A modern beamer theme}  % define title
\date{\today}                  % define date
\author{Matthias Vogelgesang}  % define author
\institute{Institute}         % define institute
\begin{document}
\maketitle                     % create titlepage
\section{First Section}        % create section
\begin{frame}{First Frame}     % first frame
  Lorem ipsum dolor sit amet, ...
\end{frame}
\begin{frame}{Second Frame}    % second frame
  Lorem ipsum dolor sit amet, ...
\end{frame}
```

`\end{document}`

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 `MTHHEME` in the preamble of the presentation.

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

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

`\metropolisset[<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.

`\metropolisset[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.

`plainformat` *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, progressbar* progressbar
Adds a thin progress bar similar to the section progress bar underneath each frame 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 framenumbers can be added additionally.

`progressbar` *none, frametitle* none
Setting this option to `frametitle` adds a progress bar underneath each frame title similar to the section progress bar.

`frametitleformat` *regular, lowercase, uppercase* lowercase
Control the case style of the frame title.

`frametitleoffset` *<dimension>* 2em
`noframetitleoffset` The frametitle offset is an additional vertical space after the frame title to center the content vertically on the frame. To remove this space entirely the short option `noframetitleoffset` is defined.

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 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.

3.4 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

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 Contributors

For a full list of contributors please visit the [GitHub Repository](#).

7 Implementation

7.1 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}
3 \RequirePackage{ifxetex}
4 \RequirePackage{ifluatex}
```

7.1.1 Options

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

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

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

```
6 \pgfkeys{/metropolis/.cd,
7   .search also={
8     /metropolis/inner,
9     /metropolis/outer,
10    /metropolis/color,
11  },
```

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

```
12  block/.code=\pgfkeysalso{
13    inner/block=#1,
14    color/block=#1,
15  },
16 }
```

`plaintitleformat` Control the case style of the plain title


```

17 \pgfkeys{
18   /metropolis/plaintitleformat/.cd,
19   .is choice,
20   regular/.code=\renewcommand{\@metropolis@plaintitleformat}{#1},
21   lowercase/.code={%
22     \renewcommand{\@metropolis@plaintitleformat}{\MakeLowercase{#1}}
23   },
24   uppercase/.code={%
25     \renewcommand{\@metropolis@plaintitleformat}{\MakeUppercase{#1}}
26   },
27 }

```

`everytitleformat` Control the case style of the every title

```

28 \pgfkeys{
29   /metropolis/everytitleformat/.code=\pgfkeysalso{
30     inner/titleformat=#1,
31     inner/sectiontitleformat=#1,
32     outer/frametitleformat=#1,
33     plaintitleformat=#1,
34   }
35 }

```

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

```

36 \pgfkeys{/metropolis/.cd,
37   usetitleprogressbar/.code=\pgfkeysalso{outer/progressbar=frametitle},
38   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
39   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
40   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
41   darkcolors/.code=\pgfkeysalso{color/background=dark},
42   blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
43 }

```

Set default values for options.

```

44 \newcommand{\@metropolis@setdefaults}{
45   \pgfkeys{/metropolis/.cd,
46     plaintitleformat=lowercase,
47   }

```

```
48 }
```

7.1.2 Component sub-packages

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

```
49 \useinnertheme{metropolis}
50 \useoutertheme{metropolis}
51 \usecolortheme{metropolis}
```

The `fira` font theme, which depends on `fontspec`, is only loaded if the document is being processed by Xe_{La}TeX or Lua_{La}TeX.

```
52 \ifboolexpr{bool {xetex} or bool {luatex}}{
53   \usefonttheme{metropolis}
54 }{
55   \PackageWarning{beamerthemem}{%
56     You need to compile with XeLaTeX or LuaLaTeX to use the Fira fonts.
57   }
58 }
```

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

```
59 \AtEndPreamble{%
60   \@ifpackageloaded{pgfplots}{%
61     \RequirePackage{pgfplotsthemetol}
62   }}
63 }
```

7.1.3 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.

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

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

```
65 \newcommand{\plain}[2][\]{%
66   \begin{group
67     \setbeamercolor{background canvas}{
68       use=palette primary,
69       parent=palette primary
70     }
71     \begin{frame}{#1}
72       \centering
73       \vfill
74       \vspace{1em}
75       \usebeamercolor[fg]{palette primary}
76       \usebeamerfont{section title}
77       \@metropolis@plaintitleformat{#2}
78       \vfill
79     \end{frame}
80   \endgroup
81 }
```

`\mreducelistspacing`

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

Process package options

```
83 \@metropolis@setdefaults
84 \ProcessPgfOptions{/metropolis}
```

7.2 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.

Load required packages.

```
85 \RequirePackage{etoolbox}
86 \RequirePackage{calc}
87 \RequirePackage{pgfopts}
88 \RequirePackage{tikz}
```

7.2.1 Options

block This option controls the block style.

```
89 \pgfkeys{
90   /metropolis/inner/block/.cd,
91   .is choice,
92   transparent/.code=\setlength{\@metropolis@blockskip}{0ex},
93   fill/.code=\setlength{\@metropolis@blockskip}{1ex},
94 }
```

titleformat Control the case style of the title

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

sectiontitleformat Control the case style of the section title

```
106 \pgfkeys{
107   /metropolis/inner/sectiontitleformat/.cd,
108   .is choice,
109   regular/.code=\renewcommand{\@metropolis@sectiontitleformat}{},
110   lowercase/.code={%
111     \renewcommand{\@metropolis@sectiontitleformat}{\MakeLowercase}
```

```

112   },
113   uppercase/.code={%
114     \renewcommand{\@metropolis@sectiontitleformat}{\MakeUppercase}
115   },
116 }

```

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

```

117 \pgfkeys{
118   /metropolis/inner/sectionpage/.cd,
119   .is choice,
120   none/.code=\@metropolis@sectionpage@none,
121   progressbar/.code=\@metropolis@sectionpage@progressbar,
122 }

```

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

```

123 \newcommand{\@metropolis@inner@setdefaults}{
124   \pgfkeys{/metropolis/inner/.cd,
125     sectionpage=progressbar,
126     block=transparent,
127     titleformat=lowercase,
128     sectiontitleformat=lowercase,
129   }
130 }

```

7.2.2 Title page

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

```

131 \def\@metropolis@titleformat#1{#1}
132 \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](#).

```

133 \patchcmd{\sectionentry}
134   {\def\insertsectionhead{#2}}

```

```

135 {\def\insertsectionhead{\@metropolis@sectiontitleformat{#2}}}
136 {}
137 {\PackageError{beamerinnerthememetropolis}{Patching section ti-
title failed.}}
138 \patchcmd{\beamer@section}
139 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
140 {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{\@metropolis@sectiontit
141 }}
142 {\PackageError{beamerinnerthememetropolis}{Patching section ti-
title failed.}}

```

`title page` Template for the title page.

```

143 \setbeamertemplate{title page}{
144 \begin{minipage}[b][\paperheight]{\textwidth}

```

If the user has set a `titlegraphic`, we set it in a zero-height box so it doesn't change the position of other elements.

```

145 \ifx\inserttitlegraphic\@empty\else{%
146 \vbox to 0pt {
147 \vspace*{2em}
148 \usebeamercolor[fg]{titlegraphic}%
149 \inserttitlegraphic%
150 }%
151 \nointerlineskip%
152 }
153 \fi
154 \vfill%

```

We set the title and subtitle, but only if they are defined by the user. If `\subtitle` is empty, for example, it won't leave a blank space on the title slide.

```

155 \ifx\inserttitle\@empty\else{%
156 \raggedright%
157 \linespread{1.0}%
158 \usebeamerfont{title}%
159 \usebeamercolor[fg]{title}%
160 \@metropolis@titleformat{\inserttitle}%
161 \par%
162 \vspace*{0.5em}

```

```

163   }}
164   \fi
165   \ifx\insertsubtitle\@empty\else{%
166     \usebeamerfont{subtitle}%
167     \usebeamercolor[fg]{subtitle}%
168     \insertsubtitle%
169     \par%
170     \vspace*{0.5em}
171   }}
172   \fi

```

A horizontal rule (drawn in TikZ) separates the title and subtitle from the author, date, and institution.

```

173   \begin{tikzpicture}
174     \usebeamercolor{title separator}
175     \draw[fg] (0, 0) -- (\textwidth, 0);
176   \end{tikzpicture}%
177   \par%
178   \vspace*{1em}%

```

Like the title and subtitle, we display the author only when it is defined. But 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](#).

```

179   \ifx\beamer@shortauthor\@empty\else{%
180     \usebeamerfont{author}%
181     \usebeamercolor[fg]{author}%
182     \insertauthor%
183     \par%
184     \vspace*{0.25em}
185   }}
186   \fi

```

The date and institute are set after the author, again provided they are nonempty. Note that the default date in \LaTeX is `\today`, not `\empty`.

```

187   \ifx\insertdate\@empty\else{%
188     \usebeamerfont{date}%

```

```

189     \usebeamercolor[fg]{date}%
190     \insertdate%
191     \par%
192   }}
193   \fi
194   \ifx\insertinstitute\@empty\else{%
195     \vspace*{3mm}
196     \usebeamerfont{institute}%
197     \usebeamercolor[fg]{institute}%
198     \insertinstitute%
199     \par%
200   }}
201   \fi
202   \vfill
203   \vspace*{1mm}
204   \end{minipage}
205 }

```

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` template.
`\titlepage` plate.

```

206 \def\maketitle{%
207   \ifbeamer@inframe
208     \titlepage
209   \else
210     \frame[plain]{\titlepage}
211   \fi
212 }
213 \def\titlepage{%
214   \usebeamertemplate{title page}
215 }

```


7.2.3 Section page

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

```
216 \newcommand{\@metropolis@sectionpage@none}{
217   \AtBeginSection{
218     % intenionally empty
219   }
220 }
221 \defbeamertemplate{section page}{progressbar}{
222   \vspace{2em}
223   \centering
224   \begin{minipage}{22em}
225     \usebeamerfont{section title}
226     \usebeamerfont{section title}
227     \insertsectionhead\[-1ex]
228     \usebeamertemplate*{progress bar in section page}
229   \end{minipage}
230   \par
231 }
232 \newcommand{\@metropolis@sectionpage@progressbar}{
233   \setbeamertemplate{section page}[progressbar]
234   \AtBeginSection{
235     \ifbeamer@inframe
236       \sectionpage
237     \else
238       \frame[plain,c]{\sectionpage}
239     \fi
240   }
241 }
```

`progress bar in section page` Template for the progress bar displayed by default on the section page. This code is duplicated in large part in the outer theme's template `progress bar in head-
/foot`.

```
242 \newlength{\metropolis@progressonsectionpage}
243 \setbeamertemplate{progress bar in section page}{
244   \setlength{\metropolis@progressonsectionpage}{%
245     \textwidth * \ratio{\insertframenum pt}{\inserttotalframenum pt}}%
246   }%
```

```

247 \begin{tikzpicture}
248   \draw[bg, fill=bg] (0,0) rectangle (\textwidth, 0.4pt);
249   \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressonsectionpage, 0.4pt);
250 \end{tikzpicture}%
251 }

```

The above code assumes that `\insertframenum` is less than or equal to `\inserttotalframenum`. However, this is not true on the first compile; in the absence of an `.aux` file, `\inserttotalframenum` 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 `\inserttotalframenum`; presentations with over 4000 slides will still break on first compile, but users in that situation likely have deeper problems to solve.

```

252 \def\inserttotalframenum{100}

```

7.2.4 Block environments

Regular block environment

```

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

```

```

271 \setbeamertemplate{block end}{%
272   \end{beamercolorbox}
273   \vspace*{0.2ex}
274 }

```

Alerted block environment

```

275 \setbeamertemplate{block alerted begin}{%
276   \vspace*{1ex}
277   \begin{beamercolorbox}[%
278     ht=2.4ex,
279     dp=1ex,
280     leftskip=\@metropolis@blockskip,
281     rightskip=\@metropolis@blockskip]{block title alerted}
282     \usebeamerfont*{block title alerted}\insertblocktitle%
283   \end{beamercolorbox}%
284   \vspace*{-1pt}
285   \usebeamerfont{block body alerted}%
286   \begin{beamercolorbox}[%
287     dp=1ex,
288     leftskip=\@metropolis@blockskip,
289     rightskip=\@metropolis@blockskip,
290     vmode]{block body}%
291 }
292 \setbeamertemplate{block alerted end}{%
293   \end{beamercolorbox}
294   \vspace*{0.2ex}
295 }

```

Example block environment

```

296 \setbeamertemplate{block example begin}{%
297   \vspace*{1ex}
298   \begin{beamercolorbox}[%
299     ht=2.4ex,
300     dp=1ex,
301     leftskip=\@metropolis@blockskip,
302     rightskip=\@metropolis@blockskip]{block title example}
303     \usebeamerfont*{block title example}\insertblocktitle%
304   \end{beamercolorbox}%
305   \vspace*{-1pt}

```

```

306 \usebeamerfont{block body example}%
307 \begin{beamercolorbox}[%
308   dp=1ex,
309   leftskip=\@metropolis@blockskip,
310   rightskip=\@metropolis@blockskip,
311   vmode]{block body}%
312 }
313 \setbeamertemplate{block example end}{%
314   \end{beamercolorbox}
315   \vspace*{0.2ex}
316 }

```

7.2.5 Itemize/enumerate environments

```

317 \setlength{\leftmargini}{1em}
318 \setlength{\leftmarginii}{1em}
319 \setlength{\leftmarginiii}{1em}
320 \setbeamertemplate{itemize item}{\textbullet}
321 \setbeamertemplate{itemize subitem}{\textbullet}
322 \setbeamertemplate{itemize subsubitem}{\textbullet}

```

7.2.6 Figures and tables

```

323 \setbeamertemplate{caption label separator}{: }
324 \setbeamertemplate{caption}[numbered]

```

7.2.7 Footnotes

```

325 \setbeamertemplate{footnote}{%
326   \parindent 0em\noindent%
327   \raggedright
328   \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotet
329 }

```

7.2.8 General text settings

```

330 \mode<all>
331 \setlength{\parskip}{0.5em}
332 \linespread{1.15}

```

Process package options

```
333 \@metropolis@inner@setdefaults
334 \ProcessPgfPackageOptions{/metropolis/inner}
```

7.3 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.

```
335 \RequirePackage{etoolbox}
336 \RequirePackage{calc}
337 \RequirePackage{pgfopts}
```

7.3.1 Options

numbering This option controls the page numbering.

```
338 \pgfkeys{
339   /metropolis/outer/numbering/.cd,
340   .is choice,
341   none/.code=\setbeamertemplate{frame numbering}[none],
342   counter/.code=\setbeamertemplate{frame numbering}[counter],
343   fraction/.code=\setbeamertemplate{frame numbering}[fraction],
344 }
```

progressbar This option controls the progressbar.

```
345 \pgfkeys{
346   /metropolis/outer/progressbar/.cd,
347   .is choice,
348   none/.code=\setbeamertemplate{frametitle}[plain],
349   frametitle/.code=\setbeamertemplate{frametitle}[progressbar],
350 }
```

frametitleformat Control the case style of the frame title

```
351 \pgfkeys{
352   /metropolis/outer/frametitleformat/.cd,
353   .is choice,
354   regular/.code=\renewcommand{\@metropolis@frametitleformat}{},
```

```

355 lowercase/.code={%
356   \renewcommand{\@metropolis@frametitleformat}{\MakeLowercase}
357 },
358 uppercase/.code={%
359   \renewcommand{\@metropolis@frametitleformat}{\MakeUppercase}
360 },
361 }

```

frametitleoffset This option controls the frame title offset.

```

362 \pgfkeys{
363   /metropolis/outer/.cd,
364   frametitleoffset/.code=\setlength{\@metropolis@voffset}{#1},
365   noframetitleoffset/.code=\setlength{\@metropolis@voffset}{0em},
366 }

```

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

```

367 \newcommand{\@metropolis@outer@setdefaults}{
368   \pgfkeys{/metropolis/outer/.cd,
369     numbering=counter,
370     progressbar=none,
371     frametitleformat=lowercase,
372     frametitleoffset=2em,
373   }
374 }

```

7.3.2 Head and footline

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

```

375 \setbeamertemplate{navigation symbols}{}

```

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

```

376 \defbeamertemplate{frame numbering}{none}{
377   % intentionally empty
378 }
379 \defbeamertemplate{frame numbering}{counter}{

```

```

380 \insertframenumbers
381 }
382 \defbeamertemplate{frame numbering}{fraction}{
383 \insertframenumbers/\inserttotalframenumbers
384 }

```

Define additional space between frame title and content. By default 2em.

```

385 \newlength{\@metropolis@voffset}

```

The only element in the footline by default is the frame number.

```

386 \setbeamertemplate{footline}{%
387 \begin{beamercolorbox}[%
388 wd=\textwidth,
389 ht=3ex,
390 dp=3ex,
391 leftskip=0.3cm,
392 rightskip=0.3cm
393 ]{footline}%
394 \hfill\usebeamerfont{page number in head/foot}%
395 \usebeamertemplate*{frame numbering}
396 \end{beamercolorbox}%
397 }

```

7.3.3 Frametitle

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

```

398 \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](#).

```

399 \patchcmd{\beamer@@frametitle}
400 {\beamer@ifempty{#2}}{%
401 \gdef\insertframetitle{#2\ifnum\beamer@autobreakcount>0\relax}\space\usebeamertemplate{frame numbering}
402 \gdef\beamer@frametitle{#2}%
403 \gdef\beamer@shortframetitle{#1}%

```

```

404   }}
405   {\beamer@ifempty{#2}}{%
406     \gdef\insertframetitle{{\@metropolis@frametitleformat{#2}\ifnum\beamer@autobr
tinuation}\fi}}%
407     \gdef\beamer@frametitle{#2}%
408     \gdef\beamer@shortframetitle{#1}%
409   }}
410 {}
411 {\PackageError{beamerouterthememetropolis}{Patching frame title failed.}}

```

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

```

412 \defbeamertemplate{frametitle}{plain}{%
413   \nointerlineskip
414   \begin{beamercolorbox}[%
415     wd=\paperwidth,
416     leftskip=0.3cm,
417     rightskip=0.3cm,
418     ht=2.5ex,
419     dp=1.5ex
420   ]{frametitle}
421   \insertframetitle%
422   \end{beamercolorbox}%
423   \vspace{\@metropolis@voffset}
424 }
425 \defbeamertemplate{frametitle}{progressbar}{%
426   \nointerlineskip
427   \begin{beamercolorbox}[%
428     wd=\paperwidth,
429     leftskip=0.3cm,
430     rightskip=0.3cm,
431     ht=2.5ex,
432     dp=1.5ex
433   ]{frametitle}
434   \insertframetitle%
435   \end{beamercolorbox}%
436   \usebeamertemplate*{progress bar in head/foot}
437   \vspace{\@metropolis@voffset}
438 }

```


`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`.

```
439 \newlength{\metropolis@progressinheadfoot}
440 \setbeamertemplate{progress bar in head/foot}{
441   \nointerlineskip
442   \setlength{\metropolis@progressinheadfoot}{%
443     \paperwidth * \ratio{\insertframenumbers pt}{\inserttotalframenumbers pt}%
444   }%
445   \begin{beamercolorbox}[
446     wd=\paperwidth,
447     ht=0.4pt,
448     dp=0pt]{progress bar in head/foot}
449     \begin{tikzpicture}
450       \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
451       \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
452     \end{tikzpicture}%
453   \end{beamercolorbox}
454 }
```

Process package options

```
455 \@metropolis@outer@setdefaults
456 \ProcessPgfPackageOptions{/metropolis/outer}
```

7.4 Fira font theme

Font Definitions

```
457 \RequirePackage[no-math]{fontspec}
458 \defaultfontfeatures{Mapping=tex-text}
459 \setsansfont[BoldItalicFont={Fira Sans Italic},%
460             ItalicFont={Fira Sans Light Italic},%
461             BoldFont={Fira Sans}]{Fira Sans Light}
462 \setmonofont{Fira Mono}
463 \newfontfamily\ExtraLight{Fira Sans ExtraLight}
464 \newfontfamily\Light{Fira Sans Light}
465 \newfontfamily\Book{Fira Sans}
466 \newfontfamily\Medium{Fira Sans Medium}
```

```

467 \AtBeginEnvironment{tabular}{%
468   \setsansfont[BoldFont={Fira Sans},%
469     Numbers={Monospaced}]{Fira Sans Light}%
470 }

```

Font Assignment

```

471 \setbeamerfont{title}{family=\Book, size=\Large, shape=\scshape}
472 \setbeamerfont{author}{family=\ExtraLight, size=\small}
473 \setbeamerfont{date}{family=\ExtraLight, size=\small}
474 \setbeamerfont{section title}{family=\Book, size=\Large, shape=\scshape}
475 \setbeamerfont{block title}{family=\Book, size=\normalsize}
476 \setbeamerfont{block title alerted}{family=\Book, size=\normalsize}
477 \setbeamerfont{subtitle}{family=\Light, size=\fontsize{12}{14}}
478 \setbeamerfont{frametitle}{family=\Book, size=\large, shape=\scshape}
479 \setbeamerfont{caption}{size=\small}
480 \setbeamerfont{caption name}{family=\Book}
481 \setbeamerfont{description item}{family=\Book}
482 \setbeamerfont{page number in head/foot}{size=\scriptsize}

```

Bibliography

```

483 \setbeamerfont{bibliography entry author}{family=\Light, size=\normalsize}
484 \setbeamerfont{bibliography entry title}{family=\Book, size=\normalsize}
485 \setbeamerfont{bibliography entry location}{family=\Light, size=\normalsize}
486 \setbeamerfont{bibliography entry note}{family=\Light, size=\small}
487 \linespread{1.15}

```

7.5 METROPOLIS color theme

Load required packages.

```

488 \RequirePackage{pgfopts}

```

7.5.1 Options

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

```

489 \pgfkeys{
490   /metropolis/color/block/.cd,

```

```

491     .is choice,
492     transparent/.code=\@metropolis@block@transparent,
493     fill/.code=\@metropolis@block@fill,
494 }

```

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

```

495 \pgfkeys{
496   /metropolis/color/background/.cd,
497   .is choice,
498   dark/.code=\@metropolis@colors@dark,
499   light/.code=\@metropolis@colors@light,
500 }

```

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

```

501 \newcommand{\@metropolis@color@setdefaults}{
502   \pgfkeys{/metropolis/color/.cd,
503     background=light,
504     block=transparent,
505   }
506 }

```

7.5.2 Base colors

```

507 \definecolor{mDarkBrown}{HTML}{604c38}
508 \definecolor{mDarkTeal}{HTML}{23373b}
509 \definecolor{mLightBrown}{HTML}{EB811B}
510 \definecolor{mLightGreen}{HTML}{14B03D}

```

7.5.3 Base styles

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

```

511 \newcommand{\@metropolis@colors@dark}{
512   \setbeamercolor{normal text}{%
513     fg=black!2,
514     bg=mDarkTeal
515   }

```

```

516 }
517 \newcommand{\@metropolis@colors@light}{
518   \setbeamercolor{normal text}{%
519     fg=mDarkTeal,
520     bg=black!2
521   }
522 }
523 \setbeamercolor{alerted text}{%
524   fg=mLightBrown
525 }
526 \setbeamercolor{example text}{%
527   fg=mLightGreen
528 }

```

7.5.4 Derived colors

The titles and structural elements (e.g. `itemize` bullets) are set in the same color as `normal text`. This would ideally be 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`.

```

529 \setbeamercolor{titlelike}{%
530   use=normal text,
531   parent=normal text
532 }
533 \setbeamercolor{structure}{%
534   fg=normal text.fg
535 }

```

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.

```

536 \setbeamercolor{palette primary}{%
537   use=normal text,
538   fg=normal text.bg,
539   bg=normal text.fg
540 }
541 \setbeamercolor{frametitle}{%
542   use=palette primary,

```

```

543 parent=palette primary
544 }

```

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`.

```

545 \setbeamercolor{progress bar}{%
546 use=alerted text,
547 fg=alerted text.fg,
548 bg=normal text.bg!50!normal text.fg
549 }
550 \setbeamercolor{title separator}{
551 use=progress bar,
552 parent=progress bar
553 }
554 \setbeamercolor{progress bar in head/foot}{%
555 use=progress bar,
556 parent=progress bar
557 }
558 \setbeamercolor{progress bar in section page}{
559 use=progress bar,
560 parent=progress bar
561 }

```

Blocks

```

562 \newcommand{\@metropolis@block@transparent}{
563   \setbeamercolor{block title}{use=normal text, parent=normal text}
564 }
565 \newcommand{\@metropolis@block@fill}{
566   \setbeamercolor{block title}{%
567     use=normal text,
568     fg=normal text.fg,
569     bg=normal text.bg!80!fg
570   }
571 }
572 \setbeamercolor{block title alerted}{%
573   use={block title, alerted text},
574   bg=block title.bg,

```

```

575     fg=alerted text.fg
576 }
577 \setbeamercolor{block title example}{%
578     use={block title, example text},
579     bg=block title.bg,
580     fg=example text.fg
581 }
582 \setbeamercolor{block body alerted}{use=block body, parent=block body}
583 \setbeamercolor{block body example}{use=block body, parent=block body}
584 \setbeamercolor{block body}{
585     use={block title, normal text},
586     bg=block title.bg!50!normal text.bg
587 }

```

Footnotes

```

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

Process package options

```

590 \@metropolis@color@setdefaults
591 \ProcessPgfPackageOptions{/metropolis/color}
592 \mode<all>

```

7.6 Tol pgfplots theme

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

```

593 \definecolor{TolDarkPurple}{HTML}{332288}
594 \definecolor{TolDarkBlue}{HTML}{6699CC}
595 \definecolor{TolLightBlue}{HTML}{88CCEE}
596 \definecolor{TolLightGreen}{HTML}{44AA99}
597 \definecolor{TolDarkGreen}{HTML}{117733}
598 \definecolor{TolDarkBrown}{HTML}{999933}
599 \definecolor{TolLightBrown}{HTML}{DDCC77}
600 \definecolor{TolDarkRed}{HTML}{661100}
601 \definecolor{TolLightRed}{HTML}{CC6677}

```

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

```

602 \definecolor{TolLightPink}{HTML}{AA4466}
603 \definecolor{TolDarkPink}{HTML}{882255}
604 \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.

```

605 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
606   {draw=TolDarkBlue,   fill=TolDarkBlue!70},
607   {draw=TolLightBrown, fill=TolLightBrown!70},
608   {draw=TolLightGreen, fill=TolLightGreen!70},
609   {draw=TolDarkPink,   fill=TolDarkPink!70},
610   {draw=TolDarkPurple, fill=TolDarkPurple!70},
611   {draw=TolDarkRed,    fill=TolDarkRed!70},
612   {draw=TolDarkBrown,  fill=TolDarkBrown!70},
613   {draw=TolLightRed,   fill=TolLightRed!70},
614   {draw=TolLightPink,  fill=TolLightPink!70},
615   {draw=TolLightPurple,fill=TolLightPurple!70},
616   {draw=TolLightBlue,  fill=TolLightBlue!70},
617   {draw=TolDarkGreen,  fill=TolDarkGreen!70},
618 }

```

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

```

619 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
620   {TolDarkBlue, mark=*, mark size=1.5pt},
621   {TolLightBrown, mark=square*, mark size=1.3pt},
622   {TolLightGreen, mark=triangle*, mark size=1.5pt},
623   {TolDarkBrown, mark=diamond*, mark size=1.5pt},
624 }

```

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.

```

625 \pgfplotsset{
626   compat=1.9,

```

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

```
627 mlineplot/.style={
628     mbaseplot,
629     xmajorgrids=true,
630     ymajorgrids=true,
631     major grid style={dotted},
632     axis x line=bottom,
633     axis y line=left,
634     legend style={
635         cells={anchor=west},
636         draw=none
637     },
638     cycle list name=mlineplot cycle,
639 },
```

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

```
640 mbarplot base/.style={
641     mbaseplot,
642     bar width=6pt,
643     axis y line*=none,
644 },
645 mbarplot/.style={
646     mbarplot base,
647     ybar,
648     xmajorgrids=false,
649     ymajorgrids=true,
650     area legend,
651     legend image code/.code={%
652         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
653     },
654     cycle list name=mbarplot cycle,
655 },
656 horizontal mbarplot/.style={
657     mbarplot base,
658     xmajorgrids=true,
659     ymajorgrids=false,
```



```

660     xbar stacked,
661     area legend,
662     legend image code/.code={%
663         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
664     },
665     cycle list name=mbarplot cycle,
666 },

```

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

```

667 mbaseplot/.style={
668     legend style={
669         draw=none,
670         fill=none,
671         cells={anchor=west},
672     },
673     x tick label style={
674         font=\footnotesize
675     },
676     y tick label style={
677         font=\footnotesize
678     },
679     legend style={
680         font=\footnotesize
681     },
682     major grid style={
683         dotted,
684     },
685     axis x line*=bottom,
686 },
687 disable thousands separator/.style={
688     /pgf/number format/.cd,
689     1000 sep={}
690 },
691 }

```