

Modern Beamer Presentations with the METROPOLIS 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 METROPOLIS 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 METROPOLIS theme 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

Creat 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 METROPOLIS.

```
\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	<i>list of possible values</i>	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	<i>regular, lowercase, uppercase</i>	lowercase
	Shortcut option to change the case style of all titles together.	
plainformat	<i>regular, lowercase, uppercase</i>	lowercase
	Control the case style of the plain title.	

3.1.2 Inner theme

block	<i>transparent, fill</i>	transparent
	This option controls the block background. It can either be filled with a light grey or be transparent.	
sectionpage	<i>none, progressbar</i>	progressbar
	Adds a thin progress bar similar to the section progress bar underneath each frame title.	
titleformat	<i>regular, lowercase, uppercase</i>	lowercase
	Control the case style of the title.	
sectiontitleformat	<i>regular, lowercase, uppercase</i>	lowercase
	Control the case style of the section title.	

3.1.3 Outer theme

numbering	<i>none, counter, fraction</i>	counter
	In the bottom right corner of each frame the current frame number is displayed. This can be disabled or the total framenummer can be added additionally.	
progressbar	<i>none, head, frametitle, foot</i>	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.

`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

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 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{pgfplots}
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}[c]{#1}
72 \begin{center}
73 \usebeamerfont{section title}
74 \usebeamerfont{section title}
75 \@metropolis@plaintitleformat{#2}
76 \end{center}
77 \end{frame}
78 \end{group}
79 }

```

`\mreducelistspacing`

```

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

```

Process package options

```

81 \@metropolis@setdefaults
82 \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.

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

7.2.1 Options

block This option controls the block style.

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

titleformat Control the case style of the title

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

```

102     },
103 }

```

sectiontitleformat Control the case style of the section title

```

104 \pgfkeys{
105   /metropolis/inner/sectiontitleformat/.cd,
106   .is choice,
107   regular/.code=\renewcommand{\@metropolis@sectiontitleformat}{},
108   lowercase/.code={%
109     \renewcommand{\@metropolis@sectiontitleformat}{\MakeLowercase}
110   },
111   uppercase/.code={%
112     \renewcommand{\@metropolis@sectiontitleformat}{\MakeUppercase}
113   },
114 }

```

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

```

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

```

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

```

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

```

7.2.2 Title page

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

```

129 \def\@metropolis@titleformat#1{#1}
130 \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](#).

```

131 \patchcmd{\sectionentry}
132   {\def\insertsectionhead{#2}}
133   {\def\insertsectionhead{\@metropolis@sectiontitleformat{#2}}}
134   {}
135   {\PackageError{beamerinnerthememetropolis}{Patching section ti-
136     tle failed.}}
136 \patchcmd{\beamer@section}
137   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
138   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{\@metropolis@sectiontit
139     }}
140   {\PackageError{beamerinnerthememetropolis}{Patching section ti-
141     tle failed.}}

```

`title page` Template for the title page.

```

141 \setbeamertemplate{title page}{
142   \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.

```

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

```

153   \ifx\inserttitle\@empty\else{%
154     \raggedright%
155     \linespread{1.0}%
156     \usebeamerfont{title}%
157     \usebeamercolor[fg]{title}%
158     \@metropolis@titleformat{\inserttitle}%
159     \par%
160     \vspace*{0.5em}
161   }}
162   \fi
163   \ifx\insertsubtitle\@empty\else{%
164     \usebeamerfont{subtitle}%
165     \usebeamercolor[fg]{subtitle}%
166     \insertsubtitle%
167     \par%
168     \vspace*{0.5em}
169   }}
170   \fi

```

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

```

171   \begin{tikzpicture}
172     \usebeamercolor{title separator}
173     \draw[fg] (0, 0) -- (\textwidth, 0);
174   \end{tikzpicture}%
175   \par%
176   \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](#).

```

177   \ifx\beamer@shortauthor\@empty\else{%
178     \usebeamerfont{author}%
179     \usebeamercolor[fg]{author}%

```

```

180     \insertauthor%
181     \par%
182     \vspace*{0.25em}
183   }}
184   \fi

```

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

```

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

```

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`

```

204 \def\maketitle{%
205   \ifbeamer@inframe
206     \titlepage

```



```

207 \else
208   \frame[plain]{\titlepage}
209 \fi
210 }
211 \def\titlepage{%
212   \usebeamertemplate{title page}
213 }

```

7.2.3 Section page

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

```

214 \newcommand{\@metropolis@sectionpage@none}{
215   \AtBeginSection{
216     % intenionally empty
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 }

```

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`.

```

239 \newlength{\metropolis@progressonsectionpage}
240 \setbeamertemplate{progress bar in section page}{
241   \setlength{\metropolis@progressonsectionpage}{%
242     \textwidth * \ratio{\insertframenum pt}{\inserttotalframenum 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 `\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.

```

249 \def\inserttotalframenum{100}

```

7.2.4 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 \setbeamerfont{block example end}{%
311 \end{beamercolorbox}
312 \vspace*{0.2ex}
313 }

```

7.2.5 Lists and floats

```

314 \setbeamerfont{itemize items}{\textbullet}
315 \setbeamerfont{caption label separator}{: }
316 \setbeamerfont{caption}[numbered]

```

7.2.6 Footnotes

```

317 \setbeamerfont{footnote}{%
318 \parindent 0em\noindent%
319 \raggedright
320 \usebeamerfont{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotetext
321 }

```

7.2.7 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}
```

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.

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

7.3.1 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=\renewcommand{\@metropolis@frametitleformat}{},
365 lowercase/.code={%
366   \renewcommand{\@metropolis@frametitleformat}{\MakeLowercase}
367 },
368 uppercase/.code={%
369   \renewcommand{\@metropolis@frametitleformat}{\MakeUppercase}
370 },
371 }

```

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

```

372 \newcommand{\@metropolis@outer@setdefaults}{
373   \pgfkeys{/metropolis/outer/.cd,
374     numbering=counter,
375     progressbar=none,

```

```

376     frametitleformat=lowercase,
377   }
378 }

```

7.3.2 Head and footline

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

```

379 \setbeamertemplate{navigation symbols}{}

```

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

```

380 \defbeamertemplate{frame numbering}{none}{}
381 \defbeamertemplate{frame numbering}{counter}{\insertframenumbers}
382 \defbeamertemplate{frame numbering}{fraction}{
383   \insertframenumbers/\inserttotalframenumbers
384 }

385 \defbeamertemplate{headline}{plain}{}
386 \defbeamertemplate{footline}{plain}{%
387   \begin{beamercolorbox}[wd=\textwidth, sep=3ex]{footline}%
388     \hfill%
389     \usebeamerfont{page number in head/foot}%
390     \usebeamertemplate*{frame numbering}
391   \end{beamercolorbox}%
392 }

```

7.3.3 Frametitle

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

```

393 \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](#).

```

394 \patchcmd{\beamer@@frametitle}

```

```

395 {\beamer@ifempty{#2}}{%
396     \gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax{}}\space\usebea
tinuation}\fi}}%
397     \gdef\beamer@frametitle{#2}%
398     \gdef\beamer@shortframetitle{#1}%
399     }}
400 {\beamer@ifempty{#2}}{%
401     \gdef\insertframetitle{{\@metropolis@frametitleformat{#2}\ifnum\beamer@autobr
tinuation}\fi}}%
402     \gdef\beamer@frametitle{#2}%
403     \gdef\beamer@shortframetitle{#1}%
404     }}
405 {}
406 {\PackageError{beamerouterthememetropolis}{Patching frame title failed.}}

```

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

```

407 \defbeamertemplate{frametitle}{plain}{%
408     \nointerlineskip
409     \begin{beamercolorbox}[%
410         wd=\paperwidth,
411         leftskip=0.3cm,
412         rightskip=0.3cm,
413         ht=2.5ex,
414         dp=1.5ex
415     ]{frametitle}
416     \insertframetitle%
417     \end{beamercolorbox}%
418 }

```

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**.

```

419 \newlength{\metropolis@progressinheadfoot}
420 \setbeamertemplate{progress bar in head/foot}{
421     \nointerlineskip
422     \setlength{\metropolis@progressinheadfoot}{%
423         \paperwidth * \ratio{\insertframenum pt}{\inserttotalframenum pt}%
424     }%
425     \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/foot}

```



```

426 \begin{tikzpicture}
427   \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
428   \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
429 \end{tikzpicture}%
430 \end{beamercolorbox}
431 }

```

Process package options

```

432 \@metropolis@outer@setdefaults
433 \ProcessPgfPackageOptions{/metropolis/outer}

```

7.4 Fira font theme

Font Definitions

```

434 \RequirePackage[no-math]{fontspec}
435 \defaultfontfeatures{Mapping=tex-text}
436
437
438 \setsansfont[Path=fonts/,
439             BoldItalicFont={FiraSans-Italic.otf},
440             ItalicFont={FiraSans-LightItalic.otf},
441             BoldFont={FiraSans-Regular.otf}]{FiraSans-Light.otf}
442 \setmonofont[Path=fonts/]{FiraMono-Regular.otf}
443 \newfontfamily\ExtraLight[Path=fonts/,
444                           ItalicFont=FiraSans-ExtraLightItalic.otf]{FiraSans-
ExtraLight.otf}
445 \newfontfamily\Light[Path=fonts/,
446                      ItalicFont=FiraSans-LightItalic.otf]{FiraSans-
Light.otf}
447 \newfontfamily\Book[Path=fonts/,
448                     ItalicFont=FiraSans-Italic.otf,
449                     BoldFont=FiraSans-Bold.otf,
450                     BoldItalicFont=FiraSans-BoldItalic.otf]{FiraSans-
Regular.otf}
451 \newfontfamily\Medium[Path=fonts/,ItalicFont=FiraSans-MediumItalic.otf]{FiraSans-
Medium.otf}
452
453

```

```

454 % \setsansfont[BoldItalicFont={Fira Sans Italic},%
455 %           ItalicFont={Fira Sans Light Italic},%
456 %           BoldFont={Fira Sans}]{Fira Sans Light}
457 % \setmonofont{Fira Mono}
458 % \newfontfamily\ExtraLight{Fira Sans ExtraLight}
459 % \newfontfamily\Light{Fira Sans Light}
460 % \newfontfamily\Book{Fira Sans}
461 % \newfontfamily\Medium{Fira Sans Medium}
462
463 \AtBeginEnvironment{tabular}{%
464     \setsansfont[Path=fonts/,BoldFont={FiraSans-Regular.otf},%
465     Numbers={Monospaced}]{FiraSans-Light.otf}%
466 }

```

Font Assignment

```

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

```

Bibliography

```

479 \setbeamerfont{bibliography entry author}{family=\Light, size=\normalsize}
480 \setbeamerfont{bibliography entry title}{family=\Book, size=\normalsize}
481 \setbeamerfont{bibliography entry location}{family=\Light, size=\normalsize}
482 \setbeamerfont{bibliography entry note}{family=\Light, size=\small}
483 \linespread{1.15}

```

7.5 METROPOLIS color theme

Load required packages.

```
484 \RequirePackage{pgfopts}
```

7.5.1 Options

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

```
485 \pgfkeys{
486   /metropolis/color/block/.cd,
487   .is choice,
488   transparent/.code=\@metropolis@block@transparent,
489   fill/.code=\@metropolis@block@fill,
490 }
```

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

```
491 \pgfkeys{
492   /metropolis/color/background/.cd,
493   .is choice,
494   dark/.code=\@metropolis@colors@dark,
495   light/.code=\@metropolis@colors@light,
496 }
```

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

```
497 \newcommand{\@metropolis@color@setdefaults}{
498   \pgfkeys{/metropolis/color/.cd,
499     background=light,
500     block=transparent,
501   }
502 }
```

7.5.2 Base colors

```
503 \definecolor{mDarkBrown}{HTML}{604c38}
504 \definecolor{mDarkTeal}{HTML}{23373b}
505 \definecolor{mLightBrown}{HTML}{EB811B}
506 \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`.

```
507 \newcommand{\@metropolis@colors@dark}{
508   \setbeamercolor{normal text}{%
509     fg=black!2,
510     bg=mDarkTeal
511   }
512 }
513 \newcommand{\@metropolis@colors@light}{
514   \setbeamercolor{normal text}{%
515     fg=mDarkTeal,
516     bg=black!2
517   }
518 }
519 \setbeamercolor{alerted text}{%
520   fg=mLightBrown
521 }
522 \setbeamercolor{example text}{%
523   fg=mLightGreen
524 }
```

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`.

```
525 \setbeamercolor{titlelike}{%
526   use=normal text,
527   parent=normal text
528 }
529 \setbeamercolor{structure}{%
530   fg=normal text.fg
531 }
```

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.

```
532 \setbeamercolor{palette primary}{%
533   use=normal text,
534   fg=normal text.bg,
535   bg=normal text.fg
536 }
537 \setbeamercolor{frametitle}{%
538   use=palette primary,
539   parent=palette primary
540 }
```

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**.

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

Blocks

```
558 \newcommand{\@metropolis@block@transparent}{
559   \setbeamercolor{block title}{use=normal text, parent=normal text}
```

```

560 }
561 \newcommand{\@metropolis@block@fill}{
562   \setbeamercolor{block title}{%
563     use=normal text,
564     fg=normal text.fg,
565     bg=normal text.bg!80!fg
566   }
567 }
568 \setbeamercolor{block title alerted}{%
569   use={block title, alerted text},
570   bg=block title.bg,
571   fg=alerted text.fg
572 }
573 \setbeamercolor{block title example}{%
574   use={block title, example text},
575   bg=block title.bg,
576   fg=example text.fg
577 }
578 \setbeamercolor{block body alerted}{use=block body, parent=block body}
579 \setbeamercolor{block body example}{use=block body, parent=block body}
580 \setbeamercolor{block body}{
581   use={block title, normal text},
582   bg=block title.bg!50!normal text.bg
583 }

```

Footnotes

```

584 \setbeamercolor{footnote}{fg=normal text.fg!90}
585 \setbeamercolor{footnote mark}{fg=..}

```

Process package options

```

586 \@metropolis@color@setdefaults
587 \ProcessPgfPackageOptions{/metropolis/color}

588 \mode<all>

```

7.6 Tol **pgfplots** theme

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

```
589 \definecolor{TolDarkPurple}{HTML}{332288}
590 \definecolor{TolDarkBlue}{HTML}{6699CC}
591 \definecolor{TolLightBlue}{HTML}{88CCFF}
592 \definecolor{TolLightGreen}{HTML}{44AA99}
593 \definecolor{TolDarkGreen}{HTML}{117733}
594 \definecolor{TolDarkBrown}{HTML}{999933}
595 \definecolor{TolLightBrown}{HTML}{DDCC77}
596 \definecolor{TolDarkRed}{HTML}{661100}
597 \definecolor{TolLightRed}{HTML}{CC6677}
598 \definecolor{TolLightPink}{HTML}{AA4466}
599 \definecolor{TolDarkPink}{HTML}{882255}
600 \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.

```
601 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
602   {draw=TolDarkBlue,    fill=TolDarkBlue!70},
603   {draw=TolLightBrown,  fill=TolLightBrown!70},
604   {draw=TolLightGreen,  fill=TolLightGreen!70},
605   {draw=TolDarkPink,    fill=TolDarkPink!70},
606   {draw=TolDarkPurple,  fill=TolDarkPurple!70},
607   {draw=TolDarkRed,     fill=TolDarkRed!70},
608   {draw=TolDarkBrown,   fill=TolDarkBrown!70},
609   {draw=TolLightRed,    fill=TolLightRed!70},
610   {draw=TolLightPink,   fill=TolLightPink!70},
611   {draw=TolLightPurple, fill=TolLightPurple!70},
612   {draw=TolLightBlue,   fill=TolLightBlue!70},
613   {draw=TolDarkGreen,   fill=TolDarkGreen!70},
614 }
```

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

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

```

615 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
616   {TolDarkBlue, mark=*, mark size=1.5pt},
617   {TolLightBrown, mark=square*, mark size=1.3pt},
618   {TolLightGreen, mark=triangle*, mark size=1.5pt},
619   {TolDarkBrown, mark=diamond*, mark size=1.5pt},
620 }

```

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.

```

621 \pgfplotsset{
622   compat=1.9,

```

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

```

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

```

`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`.

```

636   mbarplot base/.style={
637     mbaseplot,
638     bar width=6pt,
639     axis y line*=none,
640   },
641   mbarplot/.style={

```



```

642     mbarplot base,
643     ybar,
644     xmajorgrids=false,
645     ymajorgrids=true,
646     area legend,
647     legend image code/.code={%
648         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
649     },
650     cycle list name=mbarplot cycle,
651 },
652 horizontal mbarplot/.style={
653     mbarplot base,
654     xmajorgrids=true,
655     ymajorgrids=false,
656     xbar stacked,
657     area legend,
658     legend image code/.code={%
659         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
660     },
661     cycle list name=mbarplot cycle,
662 },

```

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

```

663 mbaseplot/.style={
664     legend style={
665         draw=none,
666         fill=none,
667         cells={anchor=west},
668     },
669     x tick label style={
670         font=\footnotesize
671     },
672     y tick label style={
673         font=\footnotesize
674     },
675     legend style={
676         font=\footnotesize
677     },
678     major grid style={

```

```

679         dotted,
680     },
681     axis x line*=bottom,
682 },
683 disable thousands separator/.style={
684     /pgf/number format/.cd,
685     1000 sep={}
686 },
687 }

```