

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 `MTHOME` 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

Build the manual.

manual

Build the manual.

demo

Build the 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 `MTHHEME`.

```
\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 following 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 sub-packages every option can and in most cases should be set on the main theme directly. If an option is listed in multiple themes, setting it on the main theme will set the option on every sub-package.

3.1.1 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. | | |

3.1.2 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</i> | none |
| | If enabled this option adds a thin progress bar similar to the sections progress bar underneath each frame title. | |
| frametitleoffset | <i><dimension></i> | 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 <code>noframetitleoffset</code> is defined. | |

3.1.3 Color 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. | |
| background | <i>dark, light</i> | 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 colors in `beamercolorthememetropolis`, including progress bar.

3.3 Title Case Formatting

The main title, section titles, frame titles and plain frame titles are all formatted according to the custom command `\@metropolis@titleformat`. By default, this is equivalent to `\MakeLowercase{#1}`, hence setting the titles in small capitals. You can change this behaviour in your preamble. For example:

```
% camel case
\renewcommand{\@metropolis@titleformat}{}
% lowercase
\renewcommand{\@metropolis@titleformat}\MakeLowercase
% uppercase
\renewcommand{\@metropolis@titleformat}\MakeUppercase
```

Be aware that these formatting macros will be replaced with theme options in the future.

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 `pgfplots-themetol` 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

8 Implementation: 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{pgfplots}
3\RequirePackage{ifxetex}
4\RequirePackage{ifluatex}
```

8.1 Options

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

```
5 \newcommand{\metropolisset}[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 }
```

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

```
17 \pgfkeys{/metropolis/.cd,  
18   usetitleprogressbar/.code=\pgfkeysalso{outer/progressbar=head},  
19   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},  
20   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},  
21   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},  
22   darkcolors/.code=\pgfkeysalso{color/background=dark},  
23   blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},  
24 }
```

Set default values for options.

```
25 \newcommand{\@metropolis@setdefaults}{  
26   \pgfkeys{/metropolis/.cd,  
27   }  
28 }
```


8.2 Component sub-packages

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

```
29 \useinnertheme{metropolis}
30 \useoutertheme{metropolis}
31 \usecolortheme{metropolis}
```

The **fira** font theme, which depends on **fontspec**, is only loaded if the document is being processed by Xe_ΛTeX or Lua_ΛTeX.

```
32 \ifboolexpr{bool {xetex} or bool {luatex}}{
33   \usefonttheme{metropolis}
34 }{
35   \PackageWarning{beamerthemem}{%
36     You need to compile with XeLaTeX or LuaLaTeX to use the Fira fonts.
37   }
38 }
```

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

```
39 \AtEndPreamble{%
40   \@ifpackageloaded{pgfplots}{%
41     \RequirePackage{pgfplotsthemetol}
42   }{}
43 }
```

8.3 Custom commands

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

| | |
|--|---|
| <code>\@metropolis@titleformat</code> | Creates hooks to change the case format of the four different titles. |
| <code>\@metropolis@sectiontitleformat</code> | |
| <code>\@metropolis@frametitleformat</code> | |
| <code>\@metropolis@plaintitleformat</code> | |

```
44 \def\@metropolis@titleformat#1{\MakeLowercase{#1}}
45 \def\@metropolis@sectiontitleformat#1{\@metropolis@titleformat{#1}}
46 \def\@metropolis@frametitleformat#1{\@metropolis@titleformat{#1}}
47 \def\@metropolis@plaintitleformat#1{\@metropolis@titleformat{#1}}
```

To give users the option to `\MakeUppercase` or `\MakeLowercase` the section title and frame title we need to patch the commands `\sectionentry`, `\beamer@section` and `\beamer@@frametitle`. This solution was suggested by Enrico Gregorio in an answer to [this StackExchange question](#).

```

48 \patchcmd{\sectionentry}
49   {\def\insertsectionhead{#2}}
50   {\def\insertsectionhead{\@metropolis@sectiontitleformat{#2}}}
51   {}{}
52 \patchcmd{\beamer@section}
53   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
54   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{\@metropolis@sectiontitle}}
55   {}{}
56
57 \patchcmd{\beamer@@frametitle}
58   {\beamer@ifempty{#2}{}{}}
59   {\gdef\insertframetitle{{#2\ifnum\beamer@autobreakcount>0\relax}}\space\usebeamerfont{frametitle}\fi}}
60   {\gdef\beamer@frametitle{#2}%
61   {\gdef\beamer@shortframetitle{#1}%
62   }}
63   {\beamer@ifempty{#2}{}{}}
64   {\gdef\insertframetitle{{\@metropolis@frametitleformat{#2}\ifnum\beamer@autobreakcount>0\relax}}\space\usebeamerfont{frametitle}\fi}}
65   {\gdef\beamer@frametitle{#2}%
66   {\gdef\beamer@shortframetitle{#1}%
67   }}
68   {}{}

```

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

```

69 \newcommand{\plain}[2][{}]{%
70   \begin{group}
71     \setbeamercolor{background canvas}{
72       use=palette primary,
73       parent=palette primary
74     }
75     \begin{frame}{#1}
76       \centering

```

```

77     \vfill
78     \vspace{1em}
79     \usebeamercolor[fg]{palette primary}
80     \usebeamerfont{section title}
81     \@metropolis@plaintitleformat{#2}
82     \vfill
83   \end{frame}
84 \endgroup
85 }

```

`\mreducelistspacing`

```

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

```

Process package options

```

87 \@metropolis@setdefaults
88 \ProcessPgfOptions{/metropolis}

```

9 Implementation: 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.

```

89 \RequirePackage{calc}
90 \RequirePackage{pgfopts}
91 \RequirePackage{tikz}

```

9.1 Options

block This option controls the block style.

```
92 \pgfkeys{
93   /metropolis/inner/block/.cd,
94   .is choice,
95   transparent/.code=\setlength{\@metropolis@blockskip}{0ex},
96   fill/.code=\setlength{\@metropolis@blockskip}{1ex},
97 }
```

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

```
98 \pgfkeys{
99   /metropolis/inner/sectionpage/.cd,
100   .is choice,
101   none/.code=\@metropolis@sectionpage@none,
102   progressbar/.code=\@metropolis@sectionpage@progressbar,
103 }
```

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

```
104 \newcommand{\@metropolis@inner@setdefaults}{
105   \pgfkeys{/metropolis/inner/.cd,
106     sectionpage=progressbar,
107     block=transparent,
108   }
109 }
```

9.2 Title page

title page Template for the title page.

```
110 \setbeamertemplate{title page}{
111   \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.

```
112   \ifx\inserttitlegraphic\@empty\else{%
113     \vbox to 0pt {
```

```

114         \vspace*{2em}
115         \usebeamercolor[fg]{titlegraphic}%
116         \inserttitlegraphic%
117     }%
118     \nointerlineskip%
119 }
120 \fi
121 \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.

```

122 \ifx\inserttitle\@empty\else{%
123     \raggedright%
124     \linespread{1.0}%
125     \usebeamerfont{title}%
126     \usebeamercolor[fg]{title}%
127     \@metropolis@titleformat{\inserttitle}%
128     \par%
129     \vspace*{0.5em}
130 }}
131 \fi
132 \ifx\insertsubtitle\@empty\else{%
133     \usebeamerfont{subtitle}%
134     \usebeamercolor[fg]{subtitle}%
135     \insertsubtitle%
136     \par%
137     \vspace*{0.5em}
138 }}
139 \fi

```

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

```

140 \begin{tikzpicture}
141     \usebeamercolor{title separator}
142     \draw[fg] (0, 0) -- (\textwidth, 0);
143 \end{tikzpicture}%
144 \par%
145 \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](#).

```

146   \ifx\beamer@shortauthor\@empty\else{%
147       \usebeamerfont{author}%
148       \usebeamercolor[fg]{author}%
149       \insertauthor%
150       \par%
151       \vspace*{0.25em}
152   }}
153   \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`.

```

154   \ifx\insertdate\@empty\else{%
155       \usebeamerfont{date}%
156       \usebeamercolor[fg]{date}%
157       \insertdate%
158       \par%
159   }}
160   \fi
161   \ifx\insertinstitute\@empty\else{%
162       \vspace*{3mm}
163       \usebeamerfont{institute}%
164       \usebeamercolor[fg]{institute}%
165       \insertinstitute%
166       \par%
167   }}
168   \fi
169   \vfill
170   \vspace*{1mm}
171   \end{minipage}
172 }

```

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`

```

173 \def\maketitle{%
174   \ifbeamer@inframe
175     \titlepage
176   \else
177     \frame[plain]{\titlepage}
178   \fi
179 }
180 \def\titlepage{%
181   \usebeamertemplate{title page}
182 }
```

9.3 Section page

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

```

183 \newcommand{\@metropolis@sectionpage@none}{
184   \AtBeginSection{
185     % intenionally empty
186   }
187 }
188 \defbeamertemplate{section page}{progressbar}{
189   \vspace{2em}
190   \centering
191   \begin{minipage}{22em}
192     \usebeamercolor[fg]{section title}
193     \usebeamerfont{section title}
194     \insertsectionhead\[-1ex]
195     \usebeamertemplate*{progress bar in section page}
196   \end{minipage}
197   \par
198 }
199 \newcommand{\@metropolis@sectionpage@progressbar}{
200   \setbeamertemplate{section page}[progressbar]
```

```

201 \AtBeginSection{
202   \ifbeamer@inframe
203     \sectionpage
204   \else
205     \frame[plain,c]{\sectionpage}
206   \fi
207 }
208 }

```

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**.

```

209 \newlength{\metropolis@progressonsectionpage}
210 \setbeamertemplate{progress bar in section page}{
211   \setlength{\metropolis@progressonsectionpage}{%
212     \textwidth * \ratio{\insertframenum pt}{\inserttotalframenum pt}}%
213   }%
214   \begin{tikzpicture}
215     \draw[bg, fill=bg] (0,0) rectangle (\textwidth, 0.4pt);
216     \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressonsectionpage, 0.4pt);
217   \end{tikzpicture}%
218 }

```

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.

```

219 \def\inserttotalframenum{100}

```

9.4 Block environments

```

220 \newlength{\@metropolis@blockskip}
221 \setbeamertemplate{block begin}{%
222   \vspace*{1ex}

```



```

223 \begin{beamercolorbox}[%
224     ht=2.4ex,
225     dp=1ex,
226     leftskip=\@metropolis@blockskip,
227     rightskip=\@metropolis@blockskip]{block title}
228     \usebeamerfont*{block title}\insertblocktitle%
229 \end{beamercolorbox}%
230 \vspace*{-1pt}
231 \usebeamerfont{block body}%
232 \begin{beamercolorbox}[%
233     dp=1ex,
234     leftskip=\@metropolis@blockskip,
235     rightskip=\@metropolis@blockskip,
236     vmode]{block body}%
237 }
238 \setbeamertemplate{block end}{%
239 \end{beamercolorbox}
240 \vspace*{0.2ex}
241 }

```

Alerted block environment

```

242 \setbeamertemplate{block alerted begin}{%
243 \vspace*{1ex}
244 \begin{beamercolorbox}[%
245     ht=2.4ex,
246     dp=1ex,
247     leftskip=\@metropolis@blockskip,
248     rightskip=\@metropolis@blockskip]{block title alerted}
249     \usebeamerfont*{block title alerted}\insertblocktitle%
250 \end{beamercolorbox}%
251 \vspace*{-1pt}
252 \usebeamerfont{block body alerted}%
253 \begin{beamercolorbox}[%
254     dp=1ex,
255     leftskip=\@metropolis@blockskip,
256     rightskip=\@metropolis@blockskip,
257     vmode]{block body}%
258 }
259 \setbeamertemplate{block alerted end}{%
260 \end{beamercolorbox}

```

```

261 \vspace*{0.2ex}
262 }

```

Example block environment

```

263 \setbeamertemplate{block example begin}{%
264   \vspace*{1ex}
265   \begin{beamercolorbox}[%
266     ht=2.4ex,
267     dp=1ex,
268     leftskip=\@metropolis@blockskip,
269     rightskip=\@metropolis@blockskip]{block title example}
270     \usebeamerfont*{block title example}\insertblocktitle%
271   \end{beamercolorbox}%
272   \vspace*{-1pt}
273   \usebeamerfont{block body example}%
274   \begin{beamercolorbox}[%
275     dp=1ex,
276     leftskip=\@metropolis@blockskip,
277     rightskip=\@metropolis@blockskip,
278     vmode]{block body}%
279 }
280 \setbeamertemplate{block example end}{%
281   \end{beamercolorbox}
282   \vspace*{0.2ex}
283 }

```

9.5 Itemize/enumerate environments

```

284 \setlength{\leftmargini}{1em}
285 \setlength{\leftmarginii}{1em}
286 \setlength{\leftmarginiii}{1em}
287 \setbeamertemplate{itemize item}{\textbullet}
288 \setbeamertemplate{itemize subitem}{\textbullet}
289 \setbeamertemplate{itemize subsubitem}{\textbullet}

```

9.6 Figures and tables

```

290 \setbeamertemplate{caption label separator}{: }
291 \setbeamertemplate{caption}[numbered]

```

9.7 Footnotes

```
292 \setbeamertemplate{footnote}{%
293   \parindent 0em\noindent%
294   \raggedright
295   \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotetext
296 }
```

9.8 General text settings

```
297 \mode<all>
298 \setlength{\parskip}{0.5em}
299 \linespread{1.15}
```

Process package options

```
300 \@metropolis@inner@setdefaults
301 \ProcessPgfPackageOptions{/metropolis/inner}
```

10 Implementation: 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.

```
302 \RequirePackage{calc}
303 \RequirePackage{pgfopts}
```

10.1 Options

numbering This option controls the page numbering.

```
304 \pgfkeys{
305   /metropolis/outer/numbering/.cd,
306   .is choice,
307   none/.code=\setbeamertemplate{frame numbering}[none],
308   counter/.code=\setbeamertemplate{frame numbering}[counter],
309   fraction/.code=\setbeamertemplate{frame numbering}[fraction],
310 }
```

progressbar This option controls the progressbar.

```
311 \pgfkeys{
312   /metropolis/outer/progressbar/.cd,
313   .is choice,
314   none/.code=\setbeamertemplate{progress bar in head/foot}[none],
315   head/.code=\setbeamertemplate{progress bar in head/foot}[head],
316 }
```

frametitleoffset This option controls the frame title offset.

```
317 \pgfkeys{
318   /metropolis/outer/.cd,
319   frametitleoffset/.code=\setlength{\@metropolis@voffset}{#1},
320   noframetitleoffset/.code=\setlength{\@metropolis@voffset}{0em},
321 }
```

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

```
322 \newcommand{\@metropolis@outer@setdefaults}{
323   \pgfkeys{/metropolis/outer/.cd,
324     numbering=counter,
325     progressbar=none,
326     frametitleoffset=2em,
327   }
328 }
```

10.2 Head and footline

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

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

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

```
330 \defbeamertemplate{frame numbering}{none}{
331   % intentionally empty
332 }
333 \defbeamertemplate{frame numbering}{counter}{
```

```

334 \insertframenumbers
335 }
336 \defbeamertemplate{frame numbering}{fraction}{
337 \insertframenumbers/\inserttotalframenumbers
338 }

```

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

```

339 \newlength{\@metropolis@voffset}

```

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

```

340 \setbeamertemplate{footline}{%
341 \begin{beamercolorbox}[%
342 wd=\textwidth,
343 ht=3ex,
344 dp=3ex,
345 leftskip=0.3cm,
346 rightskip=0.3cm
347 ]{footline}%
348 \hfill\usebeamerfont{page number in head/foot}%
349 \usebeamertemplate*{frame numbering}
350 \end{beamercolorbox}%
351 }

```

10.3 Frametitle

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

```

352 \setbeamertemplate{frametitle}{%
353 \nointerlineskip
354 \begin{beamercolorbox}[%
355 wd=\paperwidth,
356 leftskip=0.3cm,
357 rightskip=0.3cm,
358 ht=2.5ex,
359 dp=1.5ex
360 ]{frametitle}
361 \insertframetitle%
362 \end{beamercolorbox}%

```

```

363 \usebeamertemplate*{progress bar in head/foot}
364 \vspace{\@metropolis@voffset}
365 }

```

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**.

```

366 \newlength{\metropolis@progressinheadfoot}
367 \defbeamertemplate{progress bar in head/foot}{none}{}
368 \defbeamertemplate{progress bar in head/foot}{head}{
369   \nointerlineskip
370   \setlength{\metropolis@progressinheadfoot}{%
371     \paperwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}}%
372   }%
373   \begin{beamercolorbox}[
374     wd=\paperwidth,
375     ht=0.4pt,
376     dp=0pt]{progress bar in head/foot}
377     \begin{tikzpicture}
378       \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
379       \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
380     \end{tikzpicture}%
381   \end{beamercolorbox}
382 }

```

Process package options

```

383 \@metropolis@outer@setdefaults
384 \ProcessPgfPackageOptions{/metropolis/outer}

```

11 Implementation: Fira font theme

Font Definitions

```

385 \RequirePackage[no-math]{fontspec}
386 \defaultfontfeatures{Mapping=tex-text}
387 \setsansfont[BoldItalicFont={Fira Sans Italic},%
388               ItalicFont={Fira Sans Light Italic},%

```

```

389             BoldFont={Fira Sans}]{Fira Sans Light}
390 \setmonofont{Fira Mono}
391 \newfontfamily\ExtraLight{Fira Sans ExtraLight}
392 \newfontfamily\Light{Fira Sans Light}
393 \newfontfamily\Book{Fira Sans}
394 \newfontfamily\Medium{Fira Sans Medium}
395 \AtBeginEnvironment{tabular}{%
396     \setsansfont[BoldFont={Fira Sans},%
397         Numbers={Monospaced}]{Fira Sans Light}%
398 }

```

Font Assignment

```

399 \setbeamerfont{title}{family=\Book, size=\Large, shape=\scshape}
400 \setbeamerfont{author}{family=\ExtraLight, size=\small}
401 \setbeamerfont{date}{family=\ExtraLight, size=\small}
402 \setbeamerfont{section title}{family=\Book, size=\Large, shape=\scshape}
403 \setbeamerfont{block title}{family=\Book, size=\normalsize}
404 \setbeamerfont{block title alerted}{family=\Book, size=\normalsize}
405 \setbeamerfont{subtitle}{family=\Light, size=\fontsize{12}{14}}
406 \setbeamerfont{frametitle}{family=\Book, size=\large, shape=\scshape}
407 \setbeamerfont{caption}{size=\small}
408 \setbeamerfont{caption name}{family=\Book}
409 \setbeamerfont{description item}{family=\Book}
410 \setbeamerfont{page number in head/foot}{size=\scriptsize}

```

Bibliography

```

411 \setbeamerfont{bibliography entry author}{family=\Light, size=\normalsize}
412 \setbeamerfont{bibliography entry title}{family=\Book, size=\normalsize}
413 \setbeamerfont{bibliography entry location}{family=\Light, size=\normalsize}
414 \setbeamerfont{bibliography entry note}{family=\Light, size=\small}
415 \linespread{1.15}

```

12 Implementation: METROPOLIS color theme

Load required packages.

```

416 \RequirePackage{pgfplots}

```

12.1 Options

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

```
417 \pgfkeys{
418   /metropolis/color/block/.cd,
419   .is choice,
420   transparent/.code=\@metropolis@block@transparent,
421   fill/.code=\@metropolis@block@fill,
422 }
```

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

```
423 \pgfkeys{
424   /metropolis/color/background/.cd,
425   .is choice,
426   dark/.code=\@metropolis@colors@dark,
427   light/.code=\@metropolis@colors@light,
428 }
```

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

```
429 \newcommand{\@metropolis@color@setdefaults}{
430   \pgfkeys{/metropolis/color/.cd,
431     background=light,
432     block=transparent,
433   }
434 }
```

12.2 Base colors

```
435 \definecolor{mDarkBrown}{HTML}{604c38}
436 \definecolor{mDarkTeal}{HTML}{23373b}
437 \definecolor{mLightBrown}{HTML}{EB811B}
438 \definecolor{mLightGreen}{HTML}{14B03D}
```


12.3 Base styles

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

```
439 \newcommand{\@metropolis@colors@dark}{
440   \setbeamercolor{normal text}{%
441     fg=black!2,
442     bg=mDarkTeal
443   }
444 }
445 \newcommand{\@metropolis@colors@light}{
446   \setbeamercolor{normal text}{%
447     fg=mDarkTeal,
448     bg=black!2
449   }
450 }
451 \setbeamercolor{alerted text}{%
452   fg=mLightBrown
453 }
454 \setbeamercolor{example text}{%
455   fg=mLightGreen
456 }
```

12.4 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`.

```
457 \setbeamercolor{titlelike}{%
458   use=normal text,
459   parent=normal text
460 }
461 \setbeamercolor{structure}{%
462   fg=normal text.fg
463 }
```

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.

```
464 \setbeamercolor{palette primary}{%
465   use=normal text,
466   fg=normal text.bg,
467   bg=normal text.fg
468 }
469 \setbeamercolor{frametitle}{%
470   use=palette primary,
471   parent=palette primary
472 }
```

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**.

```
473 \setbeamercolor{progress bar}{%
474   use=alerted text,
475   fg=alerted text.fg,
476   bg=normal text.bg!50!normal text.fg
477 }
478 \setbeamercolor{title separator}{
479   use=progress bar,
480   parent=progress bar
481 }
482 \setbeamercolor{progress bar in head/foot}{%
483   use=progress bar,
484   parent=progress bar
485 }
486 \setbeamercolor{progress bar in section page}{
487   use=progress bar,
488   parent=progress bar
489 }
```

Blocks

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

```

492 }
493 \newcommand{\@metropolis@block@fill}{
494   \setbeamercolor{block title}{%
495     use=normal text,
496     fg=normal text.fg,
497     bg=normal text.bg!80!fg
498   }
499 }
500 \setbeamercolor{block title alerted}{%
501   use={block title, alerted text},
502   bg=block title.bg,
503   fg=alerted text.fg
504 }
505 \setbeamercolor{block title example}{%
506   use={block title, example text},
507   bg=block title.bg,
508   fg=example text.fg
509 }
510 \setbeamercolor{block body alerted}{use=block body, parent=block body}
511 \setbeamercolor{block body example}{use=block body, parent=block body}
512 \setbeamercolor{block body}{
513   use={block title, normal text},
514   bg=block title.bg!50!normal text.bg
515 }

```

Footnotes

```

516 \setbeamercolor{footnote}{fg=normal text.fg!90}
517 \setbeamercolor{footnote mark}{fg=..}

```

Process package options

```

518 \@metropolis@color@setdefaults
519 \ProcessPgfPackageOptions{/metropolis/color}

520 \mode<all>

```

13 Implementation: Tol pgfplots theme

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

```
521 \definecolor{TolDarkPurple}{HTML}{332288}
522 \definecolor{TolDarkBlue}{HTML}{6699CC}
523 \definecolor{TolLightBlue}{HTML}{88CCEE}
524 \definecolor{TolLightGreen}{HTML}{44AA99}
525 \definecolor{TolDarkGreen}{HTML}{117733}
526 \definecolor{TolDarkBrown}{HTML}{999933}
527 \definecolor{TolLightBrown}{HTML}{DDCC77}
528 \definecolor{TolDarkRed}{HTML}{661100}
529 \definecolor{TolLightRed}{HTML}{CC6677}
530 \definecolor{TolLightPink}{HTML}{AA4466}
531 \definecolor{TolDarkPink}{HTML}{882255}
532 \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.

```
533 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
534   {draw=TolDarkBlue,    fill=TolDarkBlue!70},
535   {draw=TolLightBrown,  fill=TolLightBrown!70},
536   {draw=TolLightGreen,  fill=TolLightGreen!70},
537   {draw=TolDarkPink,    fill=TolDarkPink!70},
538   {draw=TolDarkPurple,  fill=TolDarkPurple!70},
539   {draw=TolDarkRed,     fill=TolDarkRed!70},
540   {draw=TolDarkBrown,   fill=TolDarkBrown!70},
541   {draw=TolLightRed,    fill=TolLightRed!70},
542   {draw=TolLightPink,   fill=TolLightPink!70},
543   {draw=TolLightPurple, fill=TolLightPurple!70},
544   {draw=TolLightBlue,   fill=TolLightBlue!70},
545   {draw=TolDarkGreen,   fill=TolDarkGreen!70},
546 }
```

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.

```

547 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
548   {TolDarkBlue, mark=*, mark size=1.5pt},
549   {TolLightBrown, mark=square*, mark size=1.3pt},
550   {TolLightGreen, mark=triangle*, mark size=1.5pt},
551   {TolDarkBrown, mark=diamond*, mark size=1.5pt},
552 }

```

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.

```

553 \pgfplotsset{
554   compat=1.9,

```

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

```

555   mlineplot/.style={
556     mbaseplot,
557     xmajorgrids=true,
558     ymajorgrids=true,
559     major grid style={dotted},
560     axis x line=bottom,
561     axis y line=left,
562     legend style={
563       cells={anchor=west},
564       draw=none
565     },
566     cycle list name=mlineplot cycle,
567   },

```

`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`.

```

568   mbarplot base/.style={
569     mbaseplot,
570     bar width=6pt,
571     axis y line*=none,
572   },
573   mbarplot/.style={

```

```

574     mbarplot base,
575     ybar,
576     xmajorgrids=false,
577     ymajorgrids=true,
578     area legend,
579     legend image code/.code={%
580         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
581     },
582     cycle list name=mbarplot cycle,
583 },
584 horizontal mbarplot/.style={
585     mbarplot base,
586     xmajorgrids=true,
587     ymajorgrids=false,
588     xbar stacked,
589     area legend,
590     legend image code/.code={%
591         \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
592     },
593     cycle list name=mbarplot cycle,
594 },

```

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

```

595 mbaseplot/.style={
596     legend style={
597         draw=none,
598         fill=none,
599         cells={anchor=west},
600     },
601     x tick label style={
602         font=\footnotesize
603     },
604     y tick label style={
605         font=\footnotesize
606     },
607     legend style={
608         font=\footnotesize
609     },
610     major grid style={

```

```

611     dotted,
612 },
613 axis x line*=bottom,
614 },
615 disable thousands separator/.style={
616     /pgf/number format/.cd,
617     1000 sep={}
618 },
619 }

```