

Modern Beamer Presentations with the METROPOLIS package

Matthias Vogelgesang
matthias.vogelgesang@gmail.com

v1.0 – 2015/12/04

Contents

1	Introduction	3
2	Getting Started	3
2.1	Installing from CTAN	3
2.2	Installing from GitHub	4
2.3	Installing the Debian Package	4
2.4	A Minimal Example	5
2.5	Dependencies	5
2.6	Pandoc	5
3	Customization	6
3.1	Package options	6
3.1.1	Main theme	7
3.1.2	Inner theme	7
3.1.3	Outer theme	7
3.1.4	Color theme	8
3.2	Color Customization	8
3.3	Font Customization	9
3.3.1	Old style figures	9
3.4	Commands	9
3.5	Paul Tol's colors: a <code>pgfplots</code> theme	10

4	Known Issues	10
4.1	Titleformats	10
4.2	Plain Frame	11
5	License	11
6	Implementation	11
6.1	METROPOLIS main theme	11
6.1.1	Options	11
6.1.2	Component sub-packages	13
6.1.3	Custom commands	14
6.2	METROPOLIS inner theme	15
6.2.1	Options	15
6.2.2	Title page	18
6.2.3	Section page	22
6.2.4	Block environments	23
6.2.5	Lists and floats	25
6.2.6	Footnotes	25
6.2.7	Text and spacing settings	26
6.3	METROPOLIS outer theme	26
6.3.1	Options	27
6.3.2	Head and footline	29
6.3.3	Frametitle	29
6.4	METROPOLIS font theme	31
6.4.1	Load Fira font	31
6.4.2	General font definitions	33
6.5	METROPOLIS color theme	34
6.5.1	Options	34
6.5.2	Base colors	35
6.5.3	Base styles	35
6.5.4	Derived colors	36
6.6	Tolpgfplots theme	38

1 Introduction

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

The goal of **METROPOLIS** is to provide a simple, modern Beamer theme suitable for anyone to use. It tries to minimize noise and maximize space for content; the only visual flourish it offers is an (optional) progress bar added to each slide or to the section slides.

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

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

2 Getting Started

2.1 Installing from CTAN

For the regular user it is recommended to install **METROPOLIS** from **CTAN**. In case you keep your \TeX distribution up-to-date, chances are good that **METROPOLIS** is already installed. If it is not, you need to update your packages. For \TeX Live (or Mac \TeX on OS X) the following command updates all packages.

```
sudo tlmgr update --all
```

For any other distribution please refer to its documentation on how to update your packages.

To get the most out of the theme you should also install the **Fira** fonts. Yet this is not mandatory. **METROPOLIS** also works with the standard fonts.

2.2 Installing from GitHub

Installing **METROPOLIS**, like any Beamer theme, involves four easy steps:

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

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

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

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

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

`make sty` builds the theme style files.

`make doc` builds this documentation manual.

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

`make all` builds the theme, manual, and demo presentation.

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

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

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

`make ctan` creates a package for CTAN distribution.

2.3 Installing the Debian Package

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

2.4 A Minimal Example

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

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

2.5 Dependencies

- TikZ
- XeLaTeX or LuaTeX
- **Fira Sans** and Mono font

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

2.6 Pandoc

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

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

3 Customization

3.1 Package options

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

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

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

```
\metroset{<key=value list>}
```

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

```
\metroset{inner/block=fill}
```

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

key *list of possible values* default value

A short description of the option.

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

3.1.1 Main theme

`titleformat` *regular, smallcaps, allsmallcaps, allcaps* regular

Shortcut option to change the titleformat of all titles together. Please refer to section 4.1 for known issues.

`titleformat plain` *regular, smallcaps, allsmallcaps, allcaps* regular

Control the titleformat of the plain title. Please refer to section 4.1 for known issues.

3.1.2 Inner theme

`block` *transparent, fill* transparent

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

`sectionpage` *none, simple, progressbar* progressbar

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

`titleformat title` *regular, smallcaps, allsmallcaps, allcaps* regular

Control the titleformat of the title. Please refer to section 4.1 for known issues.

`titleformat subtitle` *regular, smallcaps, allsmallcaps, allcaps* regular

Control the titleformat of the subtitle. Please refer to section 4.1 for known issues.

`titleformat section` *regular, smallcaps, allsmallcaps, allcaps* regular

Control the titleformat of the section title. Please refer to section 4.1 for known issues.

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 framenummer can be added additionally.

`progressbar` *none, head, frametitle, foot* none

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

`titleformat frame` *regular, smallcaps, allsmallcaps, allcaps* regular

Control the titleformat of the frame title. Please refer to section 4.1 for known issues.

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/footer}{ ... }
\setbeamercolor{progress bar in section page}{ ... }

```

3.3 Font Customization

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

```

\setsansfont{Ubuntu}
\setmonofont{Ubuntu Mono}

```

3.3.1 Old style figures

The regular fontspec mechanism for changing glyph appearance applies also to this theme. In case you want to have old style figures in the text but regular lined figures for math, you have to add the following to your preamble:

```

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

```

3.4 Commands

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

3.5 Paul Tol’s colors: a `pgfplots` theme

A good presentation uses colors that are

- distinct from each other as much as possible, and
- distinct from black and white,
- under many different lighting and display environments, and
- to color-blind viewers,
- all while matching well together.

In a [technical note](#) for SRON, Paul Tol proposed a palette of colors satisfying these constraints. The sub-package `pgfplotsthemetol` defines palettes for `pgfplots` charts based on Tol’s work. Use the `mlineplot` key to plot line data and `mbarplot` or horizontal `mbarplot` to plot bar charts.

4 Known Issues

4.1 Titleformats

If you want to use either `smallcaps` or `allsmallcaps` be aware that not every font supports small caps. So make sure the font you are using does.

`allsmallcaps` and `allcaps` are quite nice from an aesthetic point of view, but they introduce some issues by using `\MakeLowercase` and `\MakeUppercase`, respectively.

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

4.2 Plain Frame

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

5 License

The theme itself is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#). This means that if you change the theme and re-distribute it, you must retain the copyright notice header and license it under the same CC-BY-SA license. This does not affect the presentation that you create with the theme.

6 Implementation

6.1 METROPOLIS main theme

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

Load the required packages.

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

6.1.1 Options

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

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

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

```
4 \pgfkeys{/metropolis/.cd,
5 .search also={
```

```

6    /metropolis/inner,
7    /metropolis/outer,
8    /metropolis/color,
9  },

```

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

```

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

```

titleformat plain Control the titleformat of the plain title

```

15 \pgfkeys{
16   /metropolis/titleformat plain/.cd,
17   .is choice,
18   regular/.code={%
19     \let\@metropolis@plaintitleformat\@empty%
20     \setbeamerfont{plain title}{shape=\normalfont}%
21   },
22   smallcaps/.code={%
23     \let\@metropolis@plaintitleformat\@empty%
24     \setbeamerfont{plain title}{shape=\scshape}%
25   },
26   allsmallcaps/.code={%
27     \let\@metropolis@plaintitleformat\MakeLowercase%
28     \setbeamerfont{plain title}{shape=\scshape}%
29     \PackageWarning{beamerthememetropolis}{%
30       Be aware that titleformat plain=allsmallcaps can lead to prob-
31       lems%
32     },
33   allcaps/.code={%
34     \let\@metropolis@plaintitleformat\MakeUppercase%
35     \setbeamerfont{plain title}{shape=\normalfont}%
36     \PackageWarning{beamerthememetropolis}{%
37       Be aware that titleformat plain=allcaps can lead to problems%
38     }
39   },

```

```
40 }
```

titleformat Control the titleformat of every title type together

```
41 \pgfkeys{
42   /metropolis/titleformat/.code=\pgfkeysalso{
43     inner/titleformat title=#1,
44     inner/titleformat subtitle=#1,
45     inner/titleformat section=#1,
46     outer/titleformat frame=#1,
47     titleformat plain=#1,
48   }
49 }
```

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

```
50 \pgfkeys{/metropolis/.cd,
51   usetitleprogressbar/.code=\pgfkeysalso{outer/progressbar=frametitle},
52   noslidenumbers/.code=\pgfkeysalso{outer/numbering=none},
53   usetotalslideindicator/.code=\pgfkeysalso{outer/numbering=fraction},
54   nosectionslide/.code=\pgfkeysalso{inner/sectionpage=none},
55   darkcolors/.code=\pgfkeysalso{color/background=dark},
56   blockbg/.code=\pgfkeysalso{color/block=fill, inner/block=fill},
57 }
```

Set default values for options.

```
58 \newcommand{\@metropolis@setdefaults}{
59   \pgfkeys{/metropolis/.cd,
60     titleformat plain=regular,
61   }
62 }
```

6.1.2 Component sub-packages

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

```
63 \useinnertheme{metropolis}
64 \useoutertheme{metropolis}
```

```

65 \usecolortheme{metropolis}
66 \usefonttheme{metropolis}

```

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

```

67 \AtEndPreamble{%
68   \ifpackageloaded{pgfplots}{%
69     \RequirePackage{pgfplotsthemetol}
70   }{}
71 }

```

6.1.3 Custom commands

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

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

```

72 \def\@metropolis@plaintitleformat#1{#1}

```

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

```

73 \newcommand{\plain}[2][{}]{%
74   \begingroup
75     \setbeamercolor{background canvas}{
76       use=palette primary,
77       parent=palette primary
78     }
79     \begin{frame}[c]{#1}
80       \begin{center}
81         \usebeamercolor{fg}{palette primary}
82         \usebeamerfont{plain title}
83         \@metropolis@plaintitleformat{#2}
84       \end{center}
85     \end{frame}
86   \endgroup
87 }

```

`\mreducelistspacing`

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

Process package options

```
89 \@metropolis@setdefaults
90 \ProcessPgfOptions{/metropolis}
```

6.2 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.

```
91 \RequirePackage{etoolbox}
92 \RequirePackage{calc}
93 \RequirePackage{pgfopts}
94 \RequirePackage{tikz}
```

6.2.1 Options

block This option controls the block style.

```
95 \pgfkeys{
96   /metropolis/inner/block/.cd,
97   .is choice,
98   transparent/.code=\setlength{\@metropolis@blockskip}{0ex},
99   fill/.code=\setlength{\@metropolis@blockskip}{1ex},
100 }
```

titleformat title Control the titleformat of the title

```
101 \pgfkeys{
102   /metropolis/inner/titleformat title/.cd,
```

```

103 .is choice,
104 regular/.code={%
105   \let\@metropolis@titleformat\@empty%
106   \setbeamerfont{title}{shape=\normalfont}%
107 },
108 smallcaps/.code={%
109   \let\@metropolis@titleformat\@empty%
110   \setbeamerfont{title}{shape=\scshape}%
111 },
112 allsmallcaps/.code={%
113   \let\@metropolis@titleformat\MakeLowercase%
114   \setbeamerfont{title}{shape=\scshape}%
115   \PackageWarning{beamerthememetropolis}{%
116     Be aware that titleformat title=allsmallcaps can lead to prob-
117     lems%
118   }
119 },
120 allcaps/.code={%
121   \let\@metropolis@titleformat\MakeUppercase%
122   \setbeamerfont{title}{shape=\normalfont}%
123   \PackageWarning{beamerthememetropolis}{%
124     Be aware that titleformat title=allcaps can lead to problems%
125   }
126 }

```

`titleformat subtitle` Control the titleformat of the subtitle

```

127 \pgfkeys{
128   /metropolis/inner/titleformat subtitle/.cd,
129   .is choice,
130   regular/.code={%
131     \let\@metropolis@subtitleformat\@empty%
132     \setbeamerfont{subtitle}{shape=\normalfont}%
133   },
134   smallcaps/.code={%
135     \let\@metropolis@subtitleformat\@empty%
136     \setbeamerfont{subtitle}{shape=\scshape}%
137   },
138   allsmallcaps/.code={%

```



```

139     \let\@metropolis@subtitleformat\MakeLowercase%
140     \setbeamerfont{subtitle}{shape=\scshape}%
141     \PackageWarning{beamerthememetropolis}{%
142     Be aware that titleformat subtitle=allsmallcaps can lead to prob-
143     lems%
144     },
145     allcaps/.code={%
146     \let\@metropolis@subtitleformat\MakeUppercase%
147     \setbeamerfont{subtitle}{shape=\normalfont}%
148     \PackageWarning{beamerthememetropolis}{%
149     Be aware that titleformat subtitle=allcaps can lead to prob-
150     lems%
151     },
152 }

```

titleformat section Control the titleformat of the section title

```

153 \pgfkeys{
154 /metropolis/inner/titleformat section/.cd,
155 .is choice,
156 regular/.code={%
157     \let\@metropolis@sectiontitleformat\@empty%
158     \setbeamerfont{section title}{shape=\normalfont}%
159 },
160 smallcaps/.code={%
161     \let\@metropolis@sectiontitleformat\@empty%
162     \setbeamerfont{section title}{shape=\scshape}%
163 },
164 allsmallcaps/.code={%
165     \let\@metropolis@sectiontitleformat\MakeLowercase%
166     \setbeamerfont{section title}{shape=\scshape}%
167     \PackageWarning{beamerthememetropolis}{%
168     Be aware that titleformat section=allsmallcaps can lead to prob-
169     lems%
170     },
171     allcaps/.code={%
172     \let\@metropolis@sectiontitleformat\MakeUppercase%

```

```

173     \setbeamerfont{section title}{shape=\normalfont}%
174     \PackageWarning{beamerthememetropolis}{%
175       Be aware that titleformat section=allcaps can lead to prob-
176       lems%
177     }
178 }

```

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

```

179 \pgfkeys{
180   /metropolis/inner/sectionpage/.cd,
181   .is choice,
182   none/.code=\@metropolis@sectionpage@none,
183   simple/.code=\@metropolis@sectionpage@simple,
184   progressbar/.code=\@metropolis@sectionpage@progressbar,
185 }

```

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

```

186 \newcommand{\@metropolis@inner@setdefaults}{
187   \pgfkeys{/metropolis/inner/.cd,
188     sectionpage=progressbar,
189     block=transparent,
190     titleformat title=regular,
191     titleformat subtitle=regular,
192     titleformat section=regular,
193   }
194 }

```

6.2.2 Title page

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

```

195 \def\@metropolis@titleformat#1{#1}
196 \def\@metropolis@subtitleformat#1{#1}
197 \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 solu-

tion was suggested by Enrico Gregorio in an answer to [this StackExchange question](#).

```

198 \patchcmd{\sectionentry}
199   {\def\insertsectionhead{#2}}
200   {\def\insertsectionhead{\@metropolis@sectiontitleformat{#2}}}
201   {}
202   {\PackageError{beamerinnerthememetropolis}{Patching section ti-
203     title failed}}
203 \patchcmd{\beamer@section}
204   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{#1}}}
205   {\def\insertsectionhead{\hyperlink{Navigation\the\c@page}{%
206     \@metropolis@sectiontitleformat{#1}}}}
207   {}
208   {\PackageError{beamerinnerthememetropolis}{Patching section ti-
209     title failed}}

```

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

```

209 \setbeamertemplate{title page}{
210   \begin{minipage}[b][\paperheight]{\textwidth}
211     \ifx\inserttitlegraphic\@empty\else\usebeamertemplate*{title graphic}\fi
212     \vfill%
213     \ifx\inserttitle\@empty\else\usebeamertemplate*{title}\fi
214     \ifx\insertsubtitle\@empty\else\usebeamertemplate*{subtitle}\fi
215     \usebeamertemplate*{title separator}

```

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

```

216   \ifx\beamer@shortauthor\@empty\else\usebeamertemplate*{author}\fi
217   \ifx\insertdate\@empty\else\usebeamertemplate*{date}\fi
218   \ifx\insertinstitute\@empty\else\usebeamertemplate*{institute}\fi
219   \vfill
220   \vspace*{1mm}
221 \end{minipage}
222 }

```

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` template.

```

223 \def\maketitle{%
224   \ifbeamer@inframe
225     \titlepage
226   \else
227     \frame[plain]{\titlepage}
228   \fi
229 }
230 \def\titlepage{%
231   \usebeamertemplate{title page}
232 }

```

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

```

233 \setbeamertemplate{title graphic}{
234   \vbox to 0pt {
235     \vspace*{2em}
236     \inserttitlegraphic%
237   }%
238   \nointerlineskip%
239 }

```

`title` Set the title on the title page.

```

240 \setbeamertemplate{title}{
241   \raggedright%
242   \linespread{1.0}%
243   \@metropolis@titleformat{\inserttitle}%
244   \par%
245   \vspace*{0.5em}
246 }

```

subtitle Set the subtitle on the title page.

```
247 \setbeamertemplate{subtitle}{
248   \@metropolis@subtitleformat{\insertsubtitle}%
249   \par%
250   \vspace*{0.5em}
251 }
```

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

```
252 \setbeamertemplate{title separator}{
253   \begin{tikzpicture}
254     \draw[fg, fill=fg] (0,0) rectangle (\textwidth, 0.4pt);
255   \end{tikzpicture}%
256   \par%
257 }
```

author Set the author on the title page.

```
258 \setbeamertemplate{author}{
259   \vspace*{2em}
260   \insertauthor%
261   \par%
262   \vspace*{0.25em}
263 }
```

date Set the date on the title page.

```
264 \setbeamertemplate{date}{
265   \insertdate%
266   \par%
267 }
```

institute Set the institute on the title page.

```
268 \setbeamertemplate{institute}{
269   \vspace*{3mm}
270   \insertinstitute%
271   \par%
272 }
```

6.2.3 Section page

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

```
273 \newcommand{\@metropolis@sectionpage@none}{
274   \AtBeginSection{
275     % intenionally empty
276   }
277 }
278 \defbeamertemplate{section page}{simple}{
279   \centering
280   \usebeamercolor[fg]{section title}
281   \usebeamerfont{section title}
282   \insertsectionhead\
283 }
284 \newcommand{\@metropolis@sectionpage@simple}{
285   \setbeamertemplate{section page}[simple]
286   \AtBeginSection{
287     \ifbeamer@inframe
288       \sectionpage
289     \else
290       \frame[plain,c]{\sectionpage}
291     \fi
292   }
293 }
294 \defbeamertemplate{section page}{progressbar}{
295   \centering
296   \begin{minipage}{22em}
297     \usebeamercolor[fg]{section title}
298     \usebeamerfont{section title}
299     \insertsectionhead\[-1ex]
300     \usebeamertemplate*{progress bar in section page}
301   \end{minipage}
302   \par
303 }
304 \newcommand{\@metropolis@sectionpage@progressbar}{
305   \setbeamertemplate{section page}[progressbar]
306   \AtBeginSection{
307     \ifbeamer@inframe
308       \sectionpage
```

```

309 \else
310 \frame[plain,c]{\sectionpage}
311 \fi
312 }
313 }

```

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**.

```

314 \newlength{\metropolis@progressonsectionpage}
315 \setbeamertemplate{progress bar in section page}{
316 \setlength{\metropolis@progressonsectionpage}{%
317 \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
318 }%
319 \begin{tikzpicture}
320 \draw[bg, fill=bg] (0,0) rectangle (\textwidth, 0.4pt);
321 \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressonsectionpage, 0.4pt);
322 \end{tikzpicture}%
323 }

```

The above code assumes that `\insertframenumber` is less than or equal to `\inserttotalframenumber`. However, this is not true on the first compile; in the absence of an `.aux` file, `\inserttotalframenumber` defaults to 1. This behaviour could cause fatal errors for long presentations, as `\metropolis@progressonsectionpage` would exceed TeX's maximum length (16383.99999pt, roughly 5.75 metres or 18.9 feet). To avoid this, we increase the default value for `\inserttotalframenumber`; presentations with over 4000 slides will still break on first compile, but users in that situation likely have deeper problems to solve.

```

324 \def\inserttotalframenumber{100}

```

6.2.4 Block environments

Regular block environment

```

325 \newlength{\@metropolis@blockskip}
326 \setbeamertemplate{block begin}{%
327 \setlength{\parskip}{\@metropolis@parskip}
328 \vspace*{1ex}

```

```

329 \begin{beamercolorbox}[%
330     ht=2.4ex,
331     dp=1ex,
332     leftskip=\@metropolis@blockskip,
333     rightskip=\@metropolis@blockskip]{block title}
334     \usebeamerfont*{block title}\insertblocktitle%
335 \end{beamercolorbox}%
336 \vspace*{-1pt}
337 \usebeamerfont{block body}%
338 \begin{beamercolorbox}[%
339     dp=1ex,
340     leftskip=\@metropolis@blockskip,
341     rightskip=\@metropolis@blockskip,
342     vmode]{block body}%
343 }
344 \setbeamertemplate{block end}{%
345     \end{beamercolorbox}
346     \vspace*{0.2ex}
347 }

```

Alerted block environment

```

348 \setbeamertemplate{block alerted begin}{%
349     \setlength{\parskip}{\@metropolis@parskip}
350     \vspace*{1ex}
351     \begin{beamercolorbox}[%
352         ht=2.4ex,
353         dp=1ex,
354         leftskip=\@metropolis@blockskip,
355         rightskip=\@metropolis@blockskip]{block title alerted}
356         \usebeamerfont*{block title alerted}\insertblocktitle%
357     \end{beamercolorbox}%
358     \vspace*{-1pt}
359     \usebeamerfont{block body alerted}%
360     \begin{beamercolorbox}[%
361         dp=1ex,
362         leftskip=\@metropolis@blockskip,
363         rightskip=\@metropolis@blockskip,
364         vmode]{block body alerted}%
365     }
366 \setbeamertemplate{block alerted end}{%

```



```

367 \end{beamercolorbox}
368 \vspace*{0.2ex}
369 }

```

Example block environment

```

370 \setbeamertemplate{block example begin}{%
371 \setlength{\parskip}{\@metropolis@parskip}
372 \vspace*{1ex}
373 \begin{beamercolorbox}[%
374 ht=2.4ex,
375 dp=1ex,
376 leftskip=\@metropolis@blockskip,
377 rightskip=\@metropolis@blockskip]{block title example}
378 \usebeamerfont*{block title example}\insertblocktitle%
379 \end{beamercolorbox}%
380 \vspace*{-1pt}
381 \usebeamerfont{block body example}%
382 \begin{beamercolorbox}[%
383 dp=1ex,
384 leftskip=\@metropolis@blockskip,
385 rightskip=\@metropolis@blockskip,
386 vmode]{block body example}%
387 }
388 \setbeamertemplate{block example end}{%
389 \end{beamercolorbox}
390 \vspace*{0.2ex}
391 }

```

6.2.5 Lists and floats

```

392 \setbeamertemplate{itemize items}{\textbullet}
393 \setbeamertemplate{caption label separator}{: }
394 \setbeamertemplate{caption}[numbered]

```

6.2.6 Footnotes

```

395 \setbeamertemplate{footnote}{%
396 \parindent 0em\noindent%
397 \raggedright
398 \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotetext%

```

```
399 }
```

6.2.7 Text and spacing settings

```
400 \newlength{\@metropolis@parskip}
401 \setlength{\@metropolis@parskip}{0.5em}
402 \setlength{\parskip}{\@metropolis@parskip}
403 \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](#).

```
404 \define@key{beamerframe}{c}[true]{% centered
405   \beamer@frametopskip=0pt plus 1fill\relax%
406   \beamer@framebottomskip=0pt plus 1fill\relax%
407   \beamer@frametopskipautobreak=0pt plus .4\paperheight\relax%
408   \beamer@framebottomskipautobreak=0pt plus .6\paperheight\relax%
409   \def\beamer@initfirstlineunskip{}}%
410 }
```

Process package options

```
411 \@metropolis@inner@setdefaults
412 \ProcessPgfPackageOptions{/metropolis/inner}
```

6.3 METROPOLIS outer theme

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

Load required packages.

```
413 \RequirePackage{etoolbox}
414 \RequirePackage{calc}
415 \RequirePackage{pgfopts}
```

6.3.1 Options

numbering This option controls the page numbering.

```
416 \pgfkeys{
417   /metropolis/outer/numbering/.cd,
418   .is choice,
419   none/.code=\setbeamertemplate{frame numbering}[none],
420   counter/.code=\setbeamertemplate{frame numbering}[counter],
421   fraction/.code=\setbeamertemplate{frame numbering}[fraction],
422 }
```

progressbar This option controls the progressbar.

```
423 \pgfkeys{
424   /metropolis/outer/progressbar/.cd,
425   .is choice,
426   none/.code={%
427     \setbeamertemplate{headline}[plain]
428     \setbeamertemplate{frametitle}[plain]
429     \setbeamertemplate{footline}[plain]
430   },
431   head/.code={\pgfkeys{/metropolis/outer/progressbar=none}
432     \addtobeamertemplate{headline}{}{}%
433     \usebeamertemplate*{progress bar in head/foot}
434   },
435   },
436   frametitle/.code={\pgfkeys{/metropolis/outer/progressbar=none}
437     \addtobeamertemplate{frametitle}{}{}%
438     \usebeamertemplate*{progress bar in head/foot}
439   },
440   },
441   foot/.code={\pgfkeys{/metropolis/outer/progressbar=none}
442     \addtobeamertemplate{footline}{}{}%
443     \usebeamertemplate*{progress bar in head/foot}%
444   },
445   },
446 }
```

frametitleformat Control the titleformat of the frame title

```

447 \pgfkeys{
448   /metropolis/outer/titleformat frame/.cd,
449   .is choice,
450   regular/.code={%
451     \let\@metropolis@frametitleformat\@empty%
452     \setbeamerfont{frametitle}{shape=\normalfont}%
453     \renewcommand{\@metropolis@frametitlestrut}{%
454       \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}%
455     }
456   },
457   smallcaps/.code={%
458     \let\@metropolis@frametitleformat\@empty%
459     \setbeamerfont{frametitle}{shape=\scshape}%
460     \renewcommand{\@metropolis@frametitlestrut}{%
461       \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}%
462     }
463   },
464   allsmallcaps/.code={%
465     \let\@metropolis@frametitleformat\MakeLowercase%
466     \setbeamerfont{frametitle}{shape=\scshape}%
467     \renewcommand{\@metropolis@frametitlestrut}{%
468       \vphantom{abcdefghijklmnopqrstuvwxyz}%
469     }
470     \PackageWarning{beamerthememetropolis}{%
471       Be aware that titleformat frame=allsmallcaps can lead to prob-
472       lems%
473     }
474   },
475   allcaps/.code={%
476     \let\@metropolis@frametitleformat\MakeUppercase%
477     \setbeamerfont{frametitle}{shape=\normalfont}%
478     \renewcommand{\@metropolis@frametitlestrut}{%
479       \vphantom{ABCDEFGHIJKLMNOPQRSTUVWXYZ}%
480     }
481     \PackageWarning{beamerthememetropolis}{%
482       Be aware that titleformat frame=allcaps can lead to problems%
483     }
484   },

```

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

```
485 \newcommand{\@metropolis@outer@setdefaults}{  
486   \pgfkeys{/metropolis/outer/.cd,  
487     numbering=counter,  
488     progressbar=none,  
489     titleformat frame=regular,  
490   }  
491 }
```

6.3.2 Head and footline

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

```
492 \setbeamertemplate{navigation symbols}{}%
```

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

```
493 \defbeamertemplate{frame numbering}{none}{}  
494 \defbeamertemplate{frame numbering}{counter}{\insertframenumbers}  
495 \defbeamertemplate{frame numbering}{fraction}{  
496   \insertframenumbers/\inserttotalframenumbers  
497 }  
  
498 \defbeamertemplate{headline}{plain}{}  
499 \defbeamertemplate{footline}{plain}{%  
500   \begin{beamercolorbox}[wd=\textwidth, sep=3ex]{footline}%  
501     \hfill%  
502     \usebeamerfont{page number in head/foot}%  
503     \usebeamertemplate*{frame numbering}  
504   \end{beamercolorbox}%  
505 }
```

6.3.3 Frametitle

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

```
506 \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](#).

```

507 \patchcmd{\beamer@@frametitle}
508   {\beamer@ifempty{#2}{}{}}{%
509     \gdef\insertframetitle{#2\ifnum\beamer@autobreakcount>0\relax{}\space%
510       \usebeamertemplate*{frametitle continuation}\fi}}%
511   \gdef\beamer@frametitle{#2}%
512   \gdef\beamer@shortframetitle{#1}%
513   }}
514 {\beamer@ifempty{#2}{}{}}{%
515   \gdef\insertframetitle{{\@metropolis@frametitleformat{#2}\ifnum%
516     \beamer@autobreakcount>0\relax{}\space%
517     \usebeamertemplate*{frametitle continuation}\fi}}%
518   \gdef\beamer@frametitle{#2}%
519   \gdef\beamer@shortframetitle{#1}%
520   }}
521 {}
522 {\PackageError{beamerouterthememetropolis}{Patching frame ti-
  tle failed}}

```

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

```

523 \newlength{\@metropolis@frametitlestrut}
524 \defbeamertemplate{frametitle}{plain}{%
525   \nointerlineskip%
526   \begin{beamercolorbox}[%
527     wd=\paperwidth,%
528     sep=1.5ex,%
529   ]{frametitle}%
530   \@metropolis@frametitlestrut\insertframetitle\@metropolis@frametitlestrut%
531   \end{beamercolorbox}%
532 }

```

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**.

```

533 \newlength{\metropolis@progressinheadfoot}
534 \setbeamertemplate{progress bar in head/foot}{

```

```

535 \nointerlineskip
536 \setlength{\metropolis@progressinheadfoot}{%
537 \paperwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
538 }%
539 \begin{beamercolorbox}[wd=\paperwidth]{progress bar in head/-
    foot}
540 \begin{tikzpicture}
541 \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
542 \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
543 \end{tikzpicture}%
544 \end{beamercolorbox}
545 }

```

Process package options

```

546 \@metropolis@outer@setdefaults
547 \ProcessPgfPackageOptions{/metropolis/outer}

```

6.4 METROPOLIS font theme

Load required packages.

```

548 \RequirePackage{etoolbox}
549 \RequirePackage{ifxetex}
550 \RequirePackage{ifluatex}

```

6.4.1 Load Fira font

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

```

551 \ifboolexpr{bool {xetex} or bool {luatex}}{
552 \RequirePackage[no-math]{fontspec}

```

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

\checkfont Checks if a font is installed and increases **fontsnofound** counter if not.

```

553 \newcounter{fontsnofound}
554 \newcommand{\checkfont}[1]{%
555   \suppressfontnotfounderror=1%
556   \font\x = "#1" at 10pt
557   \selectfont
558   \ifx\x\nullfont%
559     \stepcounter{fontsnofound}%
560   \fi%
561   \suppressfontnotfounderror=0%
562 }
563

```

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

```

564 \newcommand{\iffontsexist}[3]{%
565   \setcounter{fontsnofound}{0}%
566   \expandafter\forcsvlist\expandafter%
567   \checkfont\expandafter{#1}%
568   \ifnum\value{fontsnofound}=0%
569     #2%
570   \else%
571     #3%
572   \fi%
573 }

```

Using the previously defined macros it is tried to load the **Fira** fonts. First the default **Fira** name will be tried. Second the **Fira** fonts with the suffix OT – used by some Linux distributions – will be tried. If this also fails a warning will be displayed and the standard fonts will be used.

```

574 \iffontsexist{Fira Sans Light,%
575               Fira Sans Light Italic,%
576               Fira Sans,%
577               Fira Sans Italic}{%
578   \setsansfont[BoldFont={Fira Sans}]{Fira Sans Light}%
579 }{%
580   \iffontsexist{Fira Sans Light OT,%
581                 Fira Sans Light Italic OT,%
582                 Fira Sans OT,%

```



```

583             Fira Sans Italic OT}{%
584         \setsansfont[BoldFont={Fira Sans OT}]{Fira Sans Light OT}%
585     }{%
586         \PackageWarning{beamerthememetropolis}{%
587             Could not find Fira Sans fonts%
588         }
589     }
590 }
591 \iffontsexist{Fira Mono, Fira Mono Bold}{%
592     \setmonofont{Fira Mono}%
593 }{%
594     \iffontsexist{Fira Mono OT, Fira Mono Bold OT}{%
595         \setmonofont{Fira Mono OT}%
596     }{%
597         \PackageWarning{beamerthememetropolis}{%
598             Could not find Fira Mono fonts%
599         }
600     }
601 }
602 \AtBeginEnvironment{tabular}{%
603     \addfontfeature{Numbers={Monospaced}}}%
604 }
605 }{%
606     \PackageWarning{beamerthememetropolis}{%
607         You need to compile with XeLaTeX or LuaLaTeX to use the Fira fonts%
608     }
609 }

```

6.4.2 General font definitions

```

610 \setbeamerfont{title}{size=\Large,%
611             series=\bfseries}
612 \setbeamerfont{author}{size=\small}
613 \setbeamerfont{date}{size=\small}
614 \setbeamerfont{section title}{size=\Large,%
615             series=\bfseries}
616 \setbeamerfont{plain title}{size=\Large,%
617             series=\bfseries}
618 \setbeamerfont{block title}{size=\normalsize,%
619             series=\bfseries}

```

```

620 \setbeamerfont{block title alerted}{size=\normalsize,%
621                               series=\bfseries}
622 \setbeamerfont*{subtitle}{size=\large}
623 \setbeamerfont{frametitle}{size=\large,%
624                               series=\bfseries}
625 \setbeamerfont{caption}{size=\small}
626 \setbeamerfont{caption name}{series=\bfseries}
627 \setbeamerfont{description item}{series=\bfseries}
628 \setbeamerfont{page number in head/foot}{size=\scriptsize}
629 \setbeamerfont{bibliography entry author}{size=\normalsize,%
630                               series=\normalfont}
631 \setbeamerfont{bibliography entry title}{size=\normalsize,%
632                               series=\bfseries}
633 \setbeamerfont{bibliography entry location}{size=\normalsize,%
634                               series=\normalfont}
635 \setbeamerfont{bibliography entry note}{size=\small,%
636                               series=\normalfont}

```

6.5 METROPOLIS color theme

Load required packages.

```

637 \RequirePackage{pgfopts}

```

6.5.1 Options

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

```

638 \pgfkeys{
639   /metropolis/color/block/.cd,
640   .is choice,
641   transparent/.code=\@metropolis@block@transparent,
642   fill/.code=\@metropolis@block@fill,
643 }

```

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

```

644 \pgfkeys{
645   /metropolis/color/background/.cd,

```

```

646     .is choice,
647     dark/.code=\@metropolis@colors@dark,
648     light/.code=\@metropolis@colors@light,
649 }

```

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

```

650 \newcommand{\@metropolis@color@setdefaults}{
651   \pgfkeys{/metropolis/color/.cd,
652     background=light,
653     block=transparent,
654   }
655 }

```

6.5.2 Base colors

```

656 \definecolor{mDarkBrown}{HTML}{604c38}
657 \definecolor{mDarkTeal}{HTML}{23373b}
658 \definecolor{mLightBrown}{HTML}{EB811B}
659 \definecolor{mLightGreen}{HTML}{14B03D}

```

6.5.3 Base styles

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

```

660 \newcommand{\@metropolis@colors@dark}{
661   \setbeamercolor{normal text}{%
662     fg=black!2,
663     bg=mDarkTeal
664   }
665 }
666 \newcommand{\@metropolis@colors@light}{
667   \setbeamercolor{normal text}{%
668     fg=mDarkTeal,
669     bg=black!2
670   }
671 }
672 \setbeamercolor{alerted text}{%
673   fg=mLightBrown

```

```

674 }
675 \setbeamercolor{example text}{%
676   fg=mLightGreen
677 }

```

6.5.4 Derived colors

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

```

678 \setbeamercolor{titlelike}{use=normal text, parent=normal text}
679 \setbeamercolor{author}{use=normal text, parent=normal text}
680 \setbeamercolor{date}{use=normal text, parent=normal text}
681 \setbeamercolor{institute}{use=normal text, parent=normal text}
682 \setbeamercolor{structure}{use=normal text, fg=normal text.fg}

```

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

```

683 \setbeamercolor{palette primary}{%
684   use=normal text,
685   fg=normal text.bg,
686   bg=normal text.fg
687 }
688 \setbeamercolor{frametitle}{%
689   use=palette primary,
690   parent=palette primary
691 }

```

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`.

```

692 \setbeamercolor{progress bar}{%
693   use=alerted text,
694   fg=alerted text.fg,

```

```

695   bg=alerted text.fg!50!black!30
696 }
697 \setbeamercolor{title separator}{
698   use=progress bar,
699   parent=progress bar
700 }
701 \setbeamercolor{progress bar in head/foot}{%
702   use=progress bar,
703   parent=progress bar
704 }
705 \setbeamercolor{progress bar in section page}{
706   use=progress bar,
707   parent=progress bar
708 }

```

Blocks

```

709 \newcommand{\@metropolis@block@transparent}{
710   \setbeamercolor{block title}{use=normal text, parent=normal text}
711 }
712 \newcommand{\@metropolis@block@fill}{
713   \setbeamercolor{block title}{%
714     use=normal text,
715     fg=normal text.fg,
716     bg=normal text.bg!80!fg
717   }
718 }
719 \setbeamercolor{block title alerted}{%
720   use={block title, alerted text},
721   bg=block title.bg,
722   fg=alerted text.fg
723 }
724 \setbeamercolor{block title example}{%
725   use={block title, example text},
726   bg=block title.bg,
727   fg=example text.fg
728 }
729 \setbeamercolor{block body alerted}{use=block body, parent=block body}
730 \setbeamercolor{block body example}{use=block body, parent=block body}
731 \setbeamercolor{block body}{
732   use={block title, normal text},

```

```

733 bg=block title.bg!50!normal text.bg
734 }

```

Footnotes

```

735 \setbeamercolor{footnote}{fg=normal text.fg!90}
736 \setbeamercolor{footnote mark}{fg=..}

```

Process package options

```

737 \@metropolis@color@setdefaults
738 \ProcessPgfPackageOptions{/metropolis/color}

739 \mode<all>

```

6.6 Tol pgfplots theme

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

```

740 \definecolor{TolDarkPurple}{HTML}{332288}
741 \definecolor{TolDarkBlue}{HTML}{6699CC}
742 \definecolor{TolLightBlue}{HTML}{88CCEE}
743 \definecolor{TolLightGreen}{HTML}{44AA99}
744 \definecolor{TolDarkGreen}{HTML}{117733}
745 \definecolor{TolDarkBrown}{HTML}{999933}
746 \definecolor{TolLightBrown}{HTML}{DDCC77}
747 \definecolor{TolDarkRed}{HTML}{661100}
748 \definecolor{TolLightRed}{HTML}{CC6677}
749 \definecolor{TolLightPink}{HTML}{AA4466}
750 \definecolor{TolDarkPink}{HTML}{882255}
751 \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.

```

752 \pgfplotscreateplotcyclelist{mbarplot cycle}{%
753   {draw=TolDarkBlue, fill=TolDarkBlue!70},

```

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

```

754 {draw=TolLightBrown, fill=TolLightBrown!70},
755 {draw=TolLightGreen, fill=TolLightGreen!70},
756 {draw=TolDarkPink, fill=TolDarkPink!70},
757 {draw=TolDarkPurple, fill=TolDarkPurple!70},
758 {draw=TolDarkRed, fill=TolDarkRed!70},
759 {draw=TolDarkBrown, fill=TolDarkBrown!70},
760 {draw=TolLightRed, fill=TolLightRed!70},
761 {draw=TolLightPink, fill=TolLightPink!70},
762 {draw=TolLightPurple, fill=TolLightPurple!70},
763 {draw=TolLightBlue, fill=TolLightBlue!70},
764 {draw=TolDarkGreen, fill=TolDarkGreen!70},
765 }

```

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

```

766 \pgfplotscreateplotcyclelist{mlineplot cycle}{%
767 {TolDarkBlue, mark=*, mark size=1.5pt},
768 {TolLightBrown, mark=square*, mark size=1.3pt},
769 {TolLightGreen, mark=triangle*, mark size=1.5pt},
770 {TolDarkBrown, mark=diamond*, mark size=1.5pt},
771 }

```

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.

```

772 \pgfplotsset{
773   compat=1.9,

```

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

```

774 mlineplot/.style={
775   mbaseplot,
776   xmajorgrids=true,
777   ymajorgrids=true,
778   major grid style={dotted},
779   axis x line=bottom,
780   axis y line=left,
781   legend style={
782     cells={anchor=west},

```

```

783     draw=none
784   },
785   cycle list name=mlineplot cycle,
786 },

```

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**.

```

787 mbarplot base/.style={
788   mbaseplot,
789   bar width=6pt,
790   axis y line*=none,
791 },
792 mbarplot/.style={
793   mbarplot base,
794   ybar,
795   xmajorgrids=false,
796   ymajorgrids=true,
797   area legend,
798   legend image code/.code={%
799     \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
800   },
801   cycle list name=mbarplot cycle,
802 },
803 horizontal mbarplot/.style={
804   mbarplot base,
805   xmajorgrids=true,
806   ymajorgrids=false,
807   xbar stacked,
808   area legend,
809   legend image code/.code={%
810     \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
811   },
812   cycle list name=mbarplot cycle,
813 },

```

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

```

814 mbaseplot/.style={
815   legend style={

```



```

816     draw=none,
817     fill=none,
818     cells={anchor=west},
819 },
820 x tick label style={
821     font=\footnotesize
822 },
823 y tick label style={
824     font=\footnotesize
825 },
826 legend style={
827     font=\footnotesize
828 },
829 major grid style={
830     dotted,
831 },
832 axis x line*=bottom,
833 },
834 disable thousands separator/.style={
835     /pgf/number format/.cd,
836     1000 sep={}
837 },
838 }

```