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 MTHEME 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 MTHEME.

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

```
\end{document}
```

3 Customization

3.1 Package options

The theme provides a number of options. To use any of the options below, call them when invoking MTHEME in the preamble of the slides, i.e.

```
\usetheme[<options>]{m}
```

`usetitleprogressbar` Adds a thin progress bar similar to the section progress bar underneath each frame title.

`protectframetitle` In order to use `\cite`, `\ref` and similar commands in a frame title you have to protect the title. This can be done automatically with this option.

`blockbg` Adds background color to the blocks similar to other beamer themes.

`nooffset` By default, the MTHEME adds `\vspace{2em}` after the frametitle to center content vertically on the frame. This option removes this additional space in order to get more content per slide.

`nosectionslide` By default when using the `\section` command, a slide is created with just the title and the progress bar on it. This option prevents the creation of these additional slides.

`usetotalslideindicator` By default, only the current page number is printed in the lower right corner. This option changes the slide numbering format to `#current/#total`.

`noslidenumbers` Omits slide numbers entirely.

`darkcolors` Makes the background dark and the foreground light.

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 Formatting

The main title, section titles, and frame titles are all formatted according to the custom command `\mthemetitleformat`. By default, this is equivalent to `\scshape` and sets the titles in small capitals, but you can change it in your preamble. For example:

```
% no small capitals
\renewcommand{\mthemetitleformat}{}
% all small capitals
\renewcommand{\mthemetitleformat}{\scshape\MakeLowercase}
% all capitals
\renewcommand{\mthemetitleformat}{\MakeUppercase}
```

Note that `\MakeLowercase` and `\MakeUppercase` can have unexpected behaviour in math mode, are disabled when `protectframetitle` is used,

and cause crashes when an unprotected frametitle appears on a slide with `allowframebreaks`.

3.4 Commands

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

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

A good presentation uses colors that are

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

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

4 Known Issues

5 License

6 Contributors

For a full list of contributors please visit the GitHub Repository.

7 Implementation

Options

```
\newif\if@useTitleProgressBar
\@useTitleProgressBarfalse
\DeclareOptionBeamer{usetitleprogressbar}{
  \@useTitleProgressBartrue
}
```

usetotalslideindicator

```
\newif\if@useTotalSlideIndicator
\@useTotalSlideIndicatorfalse
\DeclareOptionBeamer{usetotalslideindicator}{
  \@useTotalSlideIndicatortrue
}
```

noslidenumbers

```
\newif\if@noSlideNumbers
\@noSlideNumbersfalse
\DeclareOptionBeamer{noslidenumbers}{
  \@noSlideNumberstrue
}
```

nosectionslide

```
\newif\if@noSectionSlide
\@noSectionSlidefalse
\DeclareOptionBeamer{nosectionslide}{
  \@noSectionSlidetrue
}
```

protectframetitle

```
\newif\if@protectFrameTitle
\@protectFrameTitlefalse
\DeclareOptionBeamer{protectframetitle}{
  \@protectFrameTitletrue
}
```

nooffset

```
\newlength{\@mtheme@voffset}
\setlength{\@mtheme@voffset}{2em}
\DeclareOptionBeamer{nooffset}{
  \setlength{\@mtheme@voffset}{0em}
}
```

blockbg

```
\DeclareOptionBeamer{blockbg}{
  \PassOptionsToPackage{blockbg}{beamercolorthememetropolis}%
}
```

darkcolors

```
\DeclareOptionBeamer{darkcolors}{
  \PassOptionsToPackage{darkcolors}{beamercolorthememetropolis}%
}
```

Unknown option error handling

```
\DeclareOptionBeamer*{
  \PackageWarning{beamerthemem}{Unknown option `\CurrentOption'}%
}
\ProcessOptionsBeamer
```

mthemetitleformat

\mthemetitleformat

```
\def\mthemetitleformat#1{\scshape #1}
```

```
\mode<presentation>
```

Packages

```
\RequirePackage{etoolbox}
```

```
\RequirePackage{tikz}
\RequirePackage{pgfplots}
\RequirePackage{ifxetex,ifluatex}
\newif\ifxetexorluatex
\ifxetex
  \xetexorluatextrue
\else
  \ifluatex
    \xetexorluatextrue
  \else
    \xetexorluatexfalse
  \fi
\fi
\usetikzlibrary{backgrounds}
\usetikzlibrary{calc}
\usecolortheme{metropolis}
\ifxetexorluatex
  \usefonttheme{metropolis}
\else
  \PackageWarning{beamerthememem}{You need to compile with XeLaTeX or LuaLaTeX for the Fira fonts.}
\fi

\useinnertheme{metropolis}%
\useoutertheme{metropolis}

\AtEndPreamble{%
  \@ifpackageloaded{pgfplots}{%
    \RequirePackage{pgfplotsthemetol}
  }{}
}

```

Create a plain frame with dark background

`\plain`

```
\newcommand{\plain}[2][]{%
  \begingroup
    \setbeamercolor{background canvas}{use=palette primary,parent=palette primary}
```

```
    \begin{frame}{#1}
      \centering
      \vfill
      \vspace{1em}
      \usebeamercolor[fg]{palette primary}
      \usebeamerfont{section title}
      \mthemetitleformat{#2}
      \vfill
    \end{frame}
  \endgroup
}
```

misc

`\mreducelistspacing`

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

8 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.

8.1 Title page

`title page` Template for the title page.

```
\setbeamertemplate{title page}{
  \begin{minipage}[b][\paperheight]{\textwidth}
    \vspace*{\@mtheme@voffset}
```

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.

```
\ifx\inserttitlegraphic\@empty\else{%
  \vbox to 0pt {
    \vspace*{2em}
    \usebeamercolor[fg]{titlegraphic}%
    \inserttitlegraphic%
  }%
  \nointerlineskip%
}
\fi
\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.

```
\ifx\inserttitle\@empty\else{{%
  \raggedright%
  \linespread{1.0}%
  \usebeamerfont{title}%
  \usebeamercolor[fg]{title}%
  \mthemetitleformat{\inserttitle}%
  \par%
  \vspace*{0.5em}
}}
\fi
\ifx\insertsubtitle\@empty\else{{%
  \usebeamerfont{subtitle}%
  \usebeamercolor[fg]{subtitle}%
  \insertsubtitle%
  \par%
  \vspace*{0.5em}
}}
\fi
```

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

```
\begin{tikzpicture}
  \usebeamercolor{title separator}
```

```
    \draw[fg] (0, 0) -- (\textwidth, 0);
  \end{tikzpicture}%
  \par%
  \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.

```
  \ifx\beamer@shortauthor\@empty\else{{%
    \usebeamerfont{author}%
    \usebeamercolor[fg]{author}%
    \insertauthor%
    \par%
    \vspace*{0.25em}
  }}
  \fi
```

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

```
  \ifx\insertdate\@empty\else{{%
    \usebeamerfont{date}%
    \usebeamercolor[fg]{date}%
    \insertdate%
    \par%
  }}
  \fi
  \ifx\insertinstitute\@empty\else{{%
    \vspace*{3mm}
    \usebeamerfont{institute}%
    \usebeamercolor[fg]{institute}%
    \insertinstitute%
    \par%
  }}
  \fi
  \vfill
  \vspace*{\@mtheme@voffset}
\end{minipage}
```

```
}
```

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` `\titlepage` Inserts the title frame, or causes the current frame to use the `title page` template.

```
\def\maketitle{%
  \ifbeamer@inframe
    \titlepage
  \else
    \frame[plain]{\titlepage}
  \fi
}
\def\titlepage{%
  \usebeamertemplate{title page}
}
```

8.2 Section page

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

```
\setbeamertemplate{section page}{
  \vspace{2em}
  \centering
  \begin{minipage}{22em}
    \usebeamercolor[fg]{section title}
    \usebeamerfont{section title}
    \insertsectionHEAD\\[-1ex]
    \usebeamertemplate*{progress bar in section page}
  \end{minipage}
  \par
}
\if@noSectionSlide\else%
  \AtBeginSection{
    \ifbeamer@inframe
```

```
      \sectionpage
    \else
      \frame[plain,c]{\sectionpage}
    \fi
  }
\fi
```

To give users the option to Or the section title, we need to expand `\insertsectionhead` before applying the relevant formatting command. This solution was suggested by Enrico Gregorio in an answer to this StackExchange question.

```
\newcommand{\insertsectionHEAD}{\expandafter\formatsectionhead\insertsectionhead}
\newcommand{\formatsectionhead}[3]{#1{#2}{\mthemetitleformat{#3}}}
```

`rogress 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`.

```
\RequirePackage{calc}
\newlength{\metropolis@progressonsectionpage}
\setbeamertemplate{progress bar in section page}{
  \setlength{\metropolis@progressonsectionpage}{%
    \textwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
  }%
  \begin{tikzpicture}
    \draw[bg, fill=bg] (0,0) rectangle (\textwidth, 0.4pt);
    \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressonsectionpage, 0.4pt);
  \end{tikzpicture}%
}
```

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.

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

8.3 Block environments

```
\newlength{\leftrightskip}
\if@beamer@metropolis@blockbg
  \setlength{\leftrightskip}{1ex}
\else
  \setlength{\leftrightskip}{0ex}
\fi
\setbeamertemplate{block begin}{%
  \vspace*{1ex}
  \begin{beamercolorbox}[%
    ht=2.4ex,
    dp=1ex,
    leftskip=\leftrightskip,
    rightskip=\leftrightskip]{block title}
      \usebeamerfont*{block title}\insertblocktitle%
  \end{beamercolorbox}%
  \vspace*{-1pt}
  \usebeamerfont{block body}%
  \begin{beamercolorbox}[%
    dp=1ex,
    leftskip=\leftrightskip,
    rightskip=\leftrightskip,
    vmode]{block body}%
}
\setbeamertemplate{block end}{%
  \end{beamercolorbox}
  \vspace*{0.2ex}
}
```

Alerted block environment

```
\setbeamertemplate{block alerted begin}{%
  \vspace*{1ex}
  \begin{beamercolorbox}[%
    ht=2.4ex,
    dp=1ex,
    leftskip=\leftrightskip,
    rightskip=\leftrightskip]{block title alerted}
      \usebeamerfont*{block title alerted}\insertblocktitle%
  \end{beamercolorbox}%
```

```
  \vspace*{-1pt}
  \usebeamerfont{block body alerted}%
  \begin{beamercolorbox}[%
    dp=1ex,
    leftskip=\leftrightskip,
    rightskip=\leftrightskip,
    vmode]{block body}%
}
\setbeamertemplate{block alerted end}{%
  \end{beamercolorbox}
  \vspace*{0.2ex}
}
```

Example block environment

```
\setbeamertemplate{block example begin}{%
  \vspace*{1ex}
  \begin{beamercolorbox}[%
    ht=2.4ex,
    dp=1ex,
    leftskip=\leftrightskip,
    rightskip=\leftrightskip]{block title example}
      \usebeamerfont*{block title example}\insertblocktitle%
  \end{beamercolorbox}%
  \vspace*{-1pt}
  \usebeamerfont{block body example}%
  \begin{beamercolorbox}[%
    dp=1ex,
    leftskip=\leftrightskip,
    rightskip=\leftrightskip,
    vmode]{block body}%
}
\setbeamertemplate{block example end}{%
  \end{beamercolorbox}
  \vspace*{0.2ex}
}
```

8.4 Itemize/enumerate environments

```
\setlength{\leftmargini}{1em}
\setlength{\leftmarginii}{1em}
```

```
\setlength{\leftmarginiii}{1em}
\setbeamertemplate{itemize item}{\textbullet}
\setbeamertemplate{itemize subitem}{\textbullet}
\setbeamertemplate{itemize subsubitem}{\textbullet}
```

8.5 Figures and tables

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

8.6 Footnotes

```
\setbeamertemplate{footnote}{%
  \parindent 0em\noindent%
  \raggedright
  \usebeamercolor{footnote}\hbox to 0.8em{\hfil\insertfootnotemark}\insertfootnotet
}
```

8.7 General text

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

9 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.

9.1 Head and footline

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

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

The only element in the footline by default is the frame number. It can optionally be omitted or displayed as a fraction of the total frames.

```
\setbeamertemplate{footline}{%
```

```
  \begin{beamercolorbox}[%
      wd=\textwidth,
      ht=3ex,
      dp=3ex,
      leftskip=0.3cm,
      rightskip=0.3cm
    ]{footline}%
    \hfill\usebeamerfont{page number in head/foot}%
  \if@noSlideNumbers%
    %Purposefully left blank to display no slide number.%
    \else%
      \if@useTotalSlideIndicator%
      \insertframenumber/\inserttotalframenumber%
      \else%
      \insertframenumber%
      \fi%
    \fi%
  \end{beamercolorbox}%
}
```

9.2 Frametitle

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

```
\setbeamertemplate{frametitle}{%
  \nointerlineskip
  \begin{beamercolorbox}[%
      wd=\paperwidth,
      leftskip=0.3cm,
      rightskip=0.3cm,
      ht=2.5ex,
      dp=1.5ex
    ]{frametitle}
  \if@protectFrameTitle%
      \mthemetitleformat{\protect\insertframetitle}%
  \else%
      \mthemetitleformat{\insertframetitle}%
  \fi%
  \end{beamercolorbox}%
```

```
  \if@useTitleProgressBar
    \nointerlineskip
    \usebeamertemplate*{progress bar in head/foot}
  \fi
  \vspace{\@mtheme@voffset}
}
```

`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`.

```
\RequirePackage{calc}
\newlength{\metropolis@progressinheadfoot}
\setbeamertemplate{progress bar in head/foot}{
  \setlength{\metropolis@progressinheadfoot}{%
    \paperwidth * \ratio{\insertframenumber pt}{\inserttotalframenumber pt}%
  }%
  \begin{beamercolorbox}[wd=\paperwidth,ht=0.4pt,dp=0pt]{progress bar in head-
/foot}
    \begin{tikzpicture}
      \draw[bg, fill=bg] (0,0) rectangle (\paperwidth, 0.4pt);
      \draw[fg, fill=fg] (0,0) rectangle (\metropolis@progressinheadfoot, 0.4pt);
    \end{tikzpicture}%
  \end{beamercolorbox}
}
```

10 Implementation: Fira font theme

Font Definitions

```
\RequirePackage[no-math]{fontspec}
\defaultfontfeatures{Mapping=tex-text}
\setsansfont[BoldItalicFont={Fira Sans Italic},%
             ItalicFont={Fira Sans Light Italic},%
             BoldFont={Fira Sans}]{Fira Sans Light}
\setmonofont{Fira Mono}
\newfontfamily\ExtraLight{Fira Sans ExtraLight}
\newfontfamily\Light{Fira Sans Light}
\newfontfamily\Book{Fira Sans}
```

```
\newfontfamily\Medium{Fira Sans Medium}
\AtBeginEnvironment{tabular}{%
    \setsansfont[BoldFont={Fira Sans},%
                 Numbers={Monospaced}]{Fira Sans Light}%
    }
```

Font Assignment

```
\setbeamerfont{title}{family=\Book, size=\Large}
\setbeamerfont{author}{family=\ExtraLight, size=\small}
\setbeamerfont{date}{family=\ExtraLight, size=\small}
\setbeamerfont{section title}{family=\Book, size=\Large}
\setbeamerfont{block title}{family=\Book, size=\normalsize}
\setbeamerfont{block title alerted}{family=\Book,size=\normalsize}
\setbeamerfont{subtitle}{family=\Light, size=\fontsize{12}{14}}
\setbeamerfont{frametitle}{family=\Book, size=\large}
\setbeamerfont{caption}{size=\small}
\setbeamerfont{caption name}{family=\Book}
\setbeamerfont{description item}{family=\Book}
\setbeamerfont{page number in head/foot}{size=\scriptsize}
```

Bibliograpy

```
\setbeamerfont{bibliography entry author}{family=\Light, size=\normalsize}
\setbeamerfont{bibliography entry title}{family=\Book, size=\normalsize}
\setbeamerfont{bibliography entry location}{family=\Light, size=\normalsize}
\setbeamerfont{bibliography entry note}{family=\Light, size=\small}
\linespread{1.15}
```

11 Implementation: METROPOLIS color theme

Options

```
\newif\if@beamer@metropolis@blockbg
\@beamer@metropolis@blockbgfalse
\DeclareOptionBeamer{blockbg}{
    \@beamer@metropolis@blockbgtrue
}
```

darkcolors

```
\newif\if@beamer@metropolis@darkcolors
\@beamer@metropolis@darkcolorsfalse
\DeclareOptionBeamer{darkcolors}{
    \@beamer@metropolis@darkcolorstrue
}
```

Unknown option error handling

```
\DeclareOptionBeamer*{%
  \PackageWarning{beamercolorthememetropolis}{Unknown option `\CurrentOption'}%
}
\ProcessOptionsBeamer
```

Colors

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

Base Colors

```
\if@beamer@metropolis@darkcolors
  \setbeamercolor{normal text}{%
    fg=black!2,
    bg=mDarkTeal
  }
\else
  \setbeamercolor{normal text}{%
    fg=mDarkTeal,
    bg=black!2
  }
\fi
\setbeamercolor{alerted text}{%
  fg=mLightBrown
}
\setbeamercolor{example text}{%
  fg=mLightGreen
}
```

Derived Colors

```
\setbeamercolor{titlelike}{use=normal text, parent=normal text}
\setbeamercolor{structure}{%
  fg=normal text.fg
}
```

Frame titles and plain slides

```
\setbeamercolor{frametitle}{use=palette primary, parent=palette pri-
  mary}
```

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

```
\setbeamercolor{palette primary}{%
  use=normal text,
  fg=normal text.bg,
  bg=normal text.fg
}
```

Progress bar and title separator

```
\setbeamercolor{title separator}{
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar in head/foot}{%
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar in section page}{
  use=progress bar,
  parent=progress bar
}
\setbeamercolor{progress bar}{%
  use=alerted text,
  fg=alerted text.fg,
  bg=normal text.bg!50!normal text.fg
}
```

Blocks

```
\if@beamer@metropolis@blockbg
  \setbeamercolor{block title}{%
    use=normal text,
    fg=normal text.fg,
    bg=normal text.bg!80!fg
  }
\else
  \setbeamercolor{block title}{use=normal text, parent=normal text}
\fi
\setbeamercolor{block title alerted}{%
    use={block title, alerted text},
    bg=block title.bg,
    fg=alerted text.fg
}
\setbeamercolor{block title example}{%
    use={block title, example text},
    bg=block title.bg,
    fg=example text.fg
}
\setbeamercolor{block body alerted}{use=block body, parent=block body}
\setbeamercolor{block body example}{use=block body, parent=block body}
\setbeamercolor{block body}{
  use={block title, normal text},
  bg=block title.bg!50!normal text.bg
}
```

Footnotes

```
\setbeamercolor{footnote}{fg=normal text.fg!90}
\setbeamercolor{footnote mark}{fg=.}
\mode<all>
```

12 Implementation: Tol `pgfplots` theme

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

```
\definecolor{TolDarkPurple}{HTML}{332288}
```

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

```
\definecolor{TolDarkBlue}{HTML}{6699CC}
\definecolor{TolLightBlue}{HTML}{88CCEE}
\definecolor{TolLightGreen}{HTML}{44AA99}
\definecolor{TolDarkGreen}{HTML}{117733}
\definecolor{TolDarkBrown}{HTML}{999933}
\definecolor{TolLightBrown}{HTML}{DDCC77}
\definecolor{TolDarkRed}{HTML}{661100}
\definecolor{TolLightRed}{HTML}{CC6677}
\definecolor{TolLightPink}{HTML}{AA4466}
\definecolor{TolDarkPink}{HTML}{882255}
\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.

```
\pgfplotscreateplotcyclelist{mbarplot cycle}{%
  {draw=TolDarkBlue,    fill=TolDarkBlue!70},
  {draw=TolLightBrown,  fill=TolLightBrown!70},
  {draw=TolLightGreen,  fill=TolLightGreen!70},
  {draw=TolDarkPink,    fill=TolDarkPink!70},
  {draw=TolDarkPurple,  fill=TolDarkPurple!70},
  {draw=TolDarkRed,     fill=TolDarkRed!70},
  {draw=TolDarkBrown,   fill=TolDarkBrown!70},
  {draw=TolLightRed,    fill=TolLightRed!70},
  {draw=TolLightPink,   fill=TolLightPink!70},
  {draw=TolLightPurple, fill=TolLightPurple!70},
  {draw=TolLightBlue,   fill=TolLightBlue!70},
  {draw=TolDarkGreen,   fill=TolDarkGreen!70},
}
```

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

```
\pgfplotscreateplotcyclelist{mlineplot cycle}{%
  {TolDarkBlue, mark=*, mark size=1.5pt},
  {TolLightBrown, mark=square*, mark size=1.3pt},
  {TolLightGreen, mark=triangle*, mark size=1.5pt},
  {TolDarkBrown, mark=diamond*, mark size=1.5pt},
}
```

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.

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

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

```
  mlineplot/.style={
    mbaseplot,
    xmajorgrids=true,
    ymajorgrids=true,
    major grid style={dotted},
    axis x line=bottom,
    axis y line=left,
    legend style={
      cells={anchor=west},
      draw=none
    },
    cycle list name=mlineplot cycle,
  },
```

`mbarplot` `horizontal 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`.

```
  mbarplot base/.style={
    mbaseplot,
    bar width=6pt,
    axis y line*=none,
  },
  mbarplot/.style={
    mbarplot base,
    ybar,
    xmajorgrids=false,
    ymajorgrids=true,
    area legend,
    legend image code/.code={%
      \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
```

```
    },
    cycle list name=mbarplot cycle,
  },
  horizontal mbarplot/.style={
    mbarplot base,
    xmajorgrids=true,
    ymajorgrids=false,
    xbar stacked,
    area legend,
    legend image code/.code={%
      \draw[#1] (0cm,-0.1cm) rectangle (0.15cm,0.1cm);
    },
    cycle list name=mbarplot cycle,
  },
```

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

```
  mbaseplot/.style={
    legend style={
      draw=none,
      fill=none,
      cells={anchor=west},
    },
    x tick label style={
      font=\footnotesize
    },
    y tick label style={
      font=\footnotesize
    },
    legend style={
      font=\footnotesize
    },
    major grid style={
      dotted,
    },
    axis x line*=bottom,
  },
  disable thousands separator/.style={
    /pgf/number format/.cd,
      1000 sep={}
```

```
556     },
557 }
```