WALLCALENDAR CODE DOCUMENTATION

v1.3.1 · 21st October 2017 · link

1 INTRODUCTION

This is the code documentation for the `wallcalendar` documentclass.

The **User Guide** is in `wallcalendar.pdf`. Clone or download from Github:

https://github.com/profound-labs/wallcalendar/

CONTENTS

2 HEADER

```
% wallcalendar.cls; Don't edit this file. Edit wallcalendar-code.org with emacs org-mode and call 'org-babel-tan
%
% Wall Calendar
%
% A wall calendar document class with custom layouts and support for internationalization.
%
% (c) Gambhiro Bhikkhu, 2017
% gambhiro.bhikkhu.85@gmail.com
%
% LPPL LaTeX Public Project License
```

3 IDENTIFICATION

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{wallcalendar}[2017/10/21 v1.3.1 A wall calendar class with custom layouts and support for interna
```

4 PRELIMINARY DECLARATIONS

```
\RequirePackage{pgfopts}
\RequirePackage{calc}

\newlength\calPaperWidth
\newlength\calPaperHeight
\newlength\@wall@leftMargin
\newlength\@wall@rightMargin
\newlength\@wall@topMargin
\newlength\@wall@bottomMargin

\def\@wall@eventsCsv{}
\def\@wall@markDefaultsCsv{}
\def\@wall@translationsInputFile{}

\newcommand*\theMonthName{}
```

5 OPTIONS

Paper sizes and ratios.

```
Choral Octavo = 6.75in x 10.5in = 171.45mm x 266.7mm , r = 1.5555
The calendar row is 40mm (without bleed) = 1.5748in
```

```
When photo is above the calendar row, it has only three side bleed.
So aspect ratio:
(10.5in - 40mm + 3mm) / (6.75in + 6mm) =
(266.7 - 40 + 3) / (171.45 + 6) =
= 1.2944
```

```
\newif\ifvarnishmask
\newif\ifshowframe
\newif\iftranslationsAutoload
\newif\ifloadBabel

\pgfkeys{
  /wall/.is family, /wall,
  defaults/.style = {
    year = \the\year,
    altyear = \empty,
    loadBabel = false,
    language = english,
    translationsAutoload = true,
    imageFolder = {./src-images},
    choralOctavoPaper,
    hMargin = 17mm,
    topMargin = 20.5mm,
    bottomMargin = 10mm,
  },
  year/.store in = \CalendarYear,
  year/.initial = \the\year,
  altyear/.store in = \CalendarAltYear,
  altyear/.initial = \empty,
  language/.store in = \@wall@calendarLanguage,
  loadBabel/.is if = loadBabel,
  translationsInputFile/.store in = \@wall@translationsInputFile,
  translationsInputFile/.initial = {},
  translationsAutoload/.is if = translationsAutoload,
  eventsCsv/.store in = \@wall@eventsCsv,
  eventsCsv/.initial = {},
  markDefaultsCsv/.store in = \@wall@markDefaultsCsv,
  markDefaultsCsv/.initial = {},
  varnishmask/.is if = varnishmask,
  showframe/.is if = showframe,
  imageFolder/.store in = \@wall@imageFolder,
  paperWidth/.code = {\setlength{\calPaperWidth}{#1}},
  paperHeight/.code = {\setlength{\calPaperHeight}{#1}},
  hMargin/.code = {\setlength{\@wall@leftMargin}{#1}%
                   \setlength{\@wall@rightMargin}{#1}},
  leftMargin/.code = {\setlength{\@wall@leftMargin}{#1}},
  rightMargin/.code = {\setlength{\@wall@rightMargin}{#1}},
  topMargin/.code = {\setlength{\@wall@topMargin}{#1}},
  bottomMargin/.code = {\setlength{\@wall@bottomMargin}{#1}},
  % Paper sizes
  choralOctavoPaper/.style = {paperWidth = 6.75in, paperHeight = 10.5in},
  a2Paper/.style = {paperWidth = 420mm, paperHeight = 594mm},
  a2PaperLandscape/.style = {paperWidth = 594mm, paperHeight = 420mm},
  a3Paper/.style = {paperWidth = 297mm, paperHeight = 420mm},
  a3PaperLandscape/.style = {paperWidth = 420mm, paperHeight = 297mm},
  a4Paper/.style = {paperWidth = 210mm, paperHeight = 297mm},
  a4PaperLandscape/.style = {paperWidth = 297mm, paperHeight = 210mm},
  a5Paper/.style = {paperWidth = 148mm, paperHeight = 210mm},% r = 1.4142 = sqrt(2)
  prevPaper/.style = {paperWidth = 170mm, paperHeight = 250mm},% r = 1.4705
}

% Debug with: \wlog{YEA: \the\@wall@bottomMargin}

\DeclareOption*{%
  \PassOptionsToClass{\CurrentOption}{memoir}
}

\pgfkeys{/wall, defaults}
\ProcessPgfOptions{/wall}
```

```
\ProcessOptions\relax
```

6 MORE DECLARATIONS

6.1 LOAD MEMOIR

```
\LoadClass[11pt,oneside]{memoir}
```

6.2 TEMP VARIABLES

```
\newlength\@tmp@a
\newlength\@tmp@b
\newlength\@tmp@c
\newlength\@tmp@width
\newlength\@tmp@height
```

6.3 REQUIRE PACKAGES

```
\ifloadBabel
\RequirePackage[\@wall@calendarLanguage]{babel}
\fi

\RequirePackage{nag}
%\RequirePackage{textcomp}
\RequirePackage[cmyk]{xcolor}
\RequirePackage{graphicx}
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
\graphicspath{{\@wall@imageFolder}}

\RequirePackage{eso-pic}
\RequirePackage{ccicons}
\RequirePackage{multicol}
\RequirePackage{wasysym}

\RequirePackage{pdftexcmds}
\RequirePackage{etoolbox}
\RequirePackage{luacode}

\RequirePackage{xcoffins}
%\RequirePackage{xstring}
%\RequirePackage{stringstrings}

\ifloadBabel
\RequirePackage[final,babel=true]{microtype}
\else
\RequirePackage[final]{microtype}
\fi

\RequirePackage{tikz}
\usetikzlibrary{calendar}
\usetikzlibrary{positioning}
```

```
\usetikzlibrary{fit}
\usetikzlibrary{shapes.geometric}

% http://tex.stackexchange.com/a/20426/831
\pgfdeclarelayer{one}
\pgfdeclarelayer{two}
\pgfsetlayers{main,one,two}

\pgfkeys{%
  /tikz/on layer/.code={
    \pgfonlayer{#1}\begingroup
    \aftergroup\endpgfonlayer
    \aftergroup\endgroup
  }
}
```

6.4 Colors

```
\definecolor{textbody}{gray}{0.15}

% pantone 1245C, RGB 191,145,12 HEX: #BF910C
% pantone 1255C, CMYK 0,27.5,100,34
\definecolor{gold}{cmyk}{0,0.275,1,0.34}
\definecolor{darkgold}{cmyk}{0.27,0.53,1,0.09}
%\definecolor{orangegold}{cmyk}{0,0.31,0.89,0}
\colorlet{orangegold}{darkgold}

\colorlet{gridcolor}{black!30}
\colorlet{weekday}{black}
\colorlet{weekend}{black!50}
\colorlet{mooncolor}{textbody}

\definecolor{datenum}{gray}{0.3}
\definecolor{plannerdatenum}{gray}{0.15}
\definecolor{quote}{gray}{0.3}
\definecolor{notes}{gray}{0.3}

\definecolor{gridcolor}{gray}{0.5}
\colorlet{pagebgcolor}{gridcolor!60}
```

showframe option colors:

```
\colorlet{photo-frame}{blue}
\colorlet{quote-frame}{red}
\colorlet{heading-frame}{brown}
\colorlet{calendar-frame}{orange}
\colorlet{events-frame}{green}

\ifshowframe
  \colorlet{calendarbg}{black!50}
\else
  \colorlet{calendarbg}{white}
\fi
```

6.5 Page Size and Trim Marks

```
\ifshowtrims
  \setstocksize{\calPaperHeight + 35mm}{\calPaperWidth + 35mm}
  \setlength{\paperheight}{\calPaperHeight}
  \setlength{\paperwidth}{\calPaperWidth}
  \trimXmarks
  \trimLmarks
  \quarkmarks
  \settrims{0.5\stockheight - 0.5\paperheight}{0.5\stockwidth - 0.5\paperwidth}
  \settrimmedsize{\calPaperHeight}{\calPaperWidth}{*}
\else\relax
  \setstocksize{\calPaperHeight}{\calPaperWidth}
  \settrims{0pt}{0pt}
  \settrimmedsize{\stockheight}{\stockwidth}{*}
\fi

%\settypeblocksize{\stockheight}{\stockwidth}{*}
% TODO: calculate margins by ratios to paper size
\setlrmarginsandblock{\@wall@leftMargin}{\@wall@rightMargin}{*}
\setulmarginsandblock{\@wall@topMargin}{\@wall@bottomMargin}{*}
\setheadfoot{0pt}{0pt}
\setheaderspaces{0pt}{*}{*}

\checkandfixthelayout% This will typeout values in pt.
\settypeoutlayoutunit{mm}% It is useful to see layout values in mm too.
\typeoutlayout
```

6.6 Base font sizes and indents

```
\def\@wall@fontSize{11}
\def\@wall@lineHeight{13.6}

\renewcommand{\normalsize}{%
  \@setfontsize\normalsize\@wall@fontSize\@wall@lineHeight
  \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
  \abovedisplayshortskip \z@ \@plus3\p@
  \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
  \belowdisplayskip \abovedisplayskip
  \color{textbody}
  \let\@listi\@listI}
\normalsize

\setlength{\vgap}{1.5em}
\setlength{\vindent}{\vgap}
\setlength{\vleftmargin}{2em}

\setlength{\parskip}{0pt}
\setlength{\parindent}{0pt}

\setlength{\fboxsep}{0pt}
```

6.7 Helpers

```
#1 -- [fill=red, opacity=0.2], additional options used for showframe
#2 -- "hasvarnish", to indicate varnishmask color replacement when =varnishmask= option is used
#3 -- the content to hold the place for
```

The `varnishmask` and `showframe` options are handled by the same `\placeholder` helper command to avoid having to use two commands and repeating the content argument.

```
\definecolor{varnishmask}{gray}{0}

% http://tex.stackexchange.com/a/59571/831
\newcommand*{\strcmpblank}[3]{%
  \ifnum\pdf@strcmp{#1}{}=\z@ #2\else #3\fi
}

\newcommand\@placeholder@pre[1]{%
  \settowidth{\@tmp@width}{#1}%
  \settototalheight{\@tmp@height}{#1}%
  \def\@spacer{\rule{\@tmp@width}{0pt}\rule{0pt}{\@tmp@height}}%
}

\newcommand\placeholder[3][]{%
  \ifvarnishmask%
    \ifstrequal{#2}{hasvarnish}{%
      \@placeholder@pre{#3}%
      \tikz\node[fill=varnishmask, inner sep=0pt]{\@spacer};%
    }{}%
  \else%
    \ifshowframe%
      \@placeholder@pre{#3}%
      \tikz\node[inner sep=0pt, opacity=0.6, #1]{\@spacer};%
    \else%
      #3%
    \fi%
  \fi%
}
```

7 Keys setup

7.1 /put photo

```
\def\@wall@photo@putPhoto#1{#1}
\def\@wall@photo@setYOffset{}

\pgfkeys{
  /put photo/.is family, /put photo,
  simple/.code = {%
    \def\@wall@photo@putPhoto##1{%
      \includegraphics{##1}%
    }%
    \def\@wall@photo@setYOffset{}%
  },
  full page/.code = {%
```

```
    \def\@wall@photo@putPhoto##1{%
      \includegraphics[%
        keepaspectratio,%
        width={\calPaperWidth + 2\@t@bleed},%
      ]{##1}%
    }%
    \def\@wall@photo@setYOffset{}%
  },
  full width above calendar/.code = {%
    \def\@wall@photo@putPhoto##1{%
      \includegraphics[%
        keepaspectratio,%
        width={\calPaperWidth + 2\@t@bleed},%
      ]{##1}%
    }%
    \def\@wall@photo@setYOffset{%
      \setlength{\@t@yOffset}{\@t@calendar@height + \@t@bleed + 1pt}%
    }%
  },
  full width/.code = {%
    \def\@wall@photo@putPhoto##1{%
      \includegraphics[%
        keepaspectratio,%
        width={\calPaperWidth + 2\@t@bleed},%
      ]{##1}%
    }%
    \def\@wall@photo@setYOffset{%
      \setlength{\@t@yOffset}{\@t@calendar@height + \@t@bleed}%
    }%
  },
}
```

7.2 /Photo

```
\pgfkeys{
  /Photo/.is family, /Photo,
  init/.style = {
    defaults/.style = {file = {}, thumbFile = {}, caption = {}, bleed = 0pt, xOffset=0pt, yOffset=0pt},
    file/.initial = {},
    thumbFile/.initial = {},
    caption/.initial = {},
    bleed/.initial = 0pt,
    xOffset/.initial = 0pt,
    yOffset/.initial = 0pt,
  },
}
```

7.3 /Quote

```
\pgfkeys{
  /Quote/.is family, /Quote,
  init/.style = {
    defaults/.style = {position=center, text={}, xOffset=0pt, yOffset=0pt},
    position/.initial = {},
    text/.initial = {},
```

```
    xOffset/.initial = 0pt,
    yOffset/.initial = 0pt,
  },
}
```

7.4 /Calendar

```
\pgfkeys{
  /Calendar/.is family, /Calendar,
  init/.style = {
    defaults/.style = {bg/.style={opacity=0.5}},
    bg/.style = {},
    minimum height/.initial = {},
  },
}
```

7.5 /Events

```
\pgfkeys{
  /Events/.is family, /Events,
  init/.style = {
    defaults/.style = {day code = {}, marks = {}, text = {}},
    day code/.initial = {},
    marks/.initial = {},
    text/.initial = {},
  },
}
```

7.6 /Txt

```
\pgfkeys{
  /Txt/.is family, /Txt,
  init/.style = {
    defaults/.style = {text = {}},
    text/.initial = {},
  },
}
```

7.7 /parseMonthEvents

```
\def\eIdx{}
\def\eMaxIdx{}
\def\eMark{}
\def\eIsoDate{}
\def\eYear{}
\def\eMonth{}
```

```
\def\eMonthShort{}
\def\eDay{}
\def\eDayText{}
\def\eNote{}

\pgfkeys{
  /parseMonthEvents/.is family, /parseMonthEvents,
  month/.initial = {},
  filter pred/.initial = nil,
  format func/.initial = nil,
  format cmd/.initial = {},
  events csv/.initial = {},
  mark defaults csv/.initial = {},
  min events/.initial = {},
  defaults/.style = {
    month = \theMonthName,
    filter pred = nil,
    format func = nil,
    format cmd = {\textsuperscript{\eIdx}~\eMonthShort~\eDay:~\eNote\par},
    events csv = \@wall@eventsCsv,
    mark defaults csv = \@wall@markDefaultsCsv,
    min events = nil,
  },
}
```

7.8 /parseYearEvents

```
\pgfkeys{
  /parseYearEvents/.is family, /parseYearEvents,
  year/.initial = {},
  filter pred/.initial = nil,
  format func/.initial = nil,
  format cmd/.initial = {},
  events csv/.initial = {},
  mark defaults csv/.initial = {},
  min events/.initial = {},
  defaults/.style = {
    year = \CalendarYear,
    filter pred = nil,
    format func = nil,
    format cmd = {\textsuperscript{\eIdx}~\eMonthShort~\eDay:~\eNote\ifnumless{\eIdx}{\eMaxIdx}{,\space}{.}},
    events csv = \@wall@eventsCsv,
    mark defaults csv = \@wall@markDefaultsCsv,
    min events = nil,
  },
}
```

8 Event marks

8.1 Kite

```
\newcommand\StarMark{*}

\newcommand\NoteStarMark{*}
```

A kite mark that is the same height as the `\StarMark`. Note that the star (*) character is usually higher than the x-height, so it is not in the vertical center of its glyph box (i.e. not at the center of the character x):

```
\newlength\@wall@starHeight

% Measure the star's height here, so that we are measuring with the current typeface.
\newcommand\KiteMark{%
\setlength{\@wall@starHeight}{\totalheightof{*}}%
\begin{tikzpicture}
  \node (box) [
    rectangle, minimum height=\@wall@starHeight, minimum width=3.4pt,
    inner sep=0pt, line width=0pt,
  ] {};
  \node[
    kite, draw, textbody, fill=textbody,
    scale=0.15, kite vertex angles=60,
    above=0pt of box.north, anchor=north,
  ] {};%
\end{tikzpicture}%
}

% Same as \KiteMark but with scale=0.2
\newcommand\NoteKiteMark{\raisebox{1pt}{%
\setlength{\@wall@starHeight}{\totalheightof{*}}%
\begin{tikzpicture}
  \node (box) [
    rectangle, minimum height=\@wall@starHeight, minimum width=3.4pt,
    inner sep=0pt, line width=0pt,
  ] {};
  \node[
    kite, draw, textbody, fill=textbody,
    scale=0.2, kite vertex angles=60,
    above=0pt of box.north, anchor=north,
  ] {};%
\end{tikzpicture}%
}}
```

8.2 Moons

```
\tikzstyle{moon circle}=[
  circle,
  inner sep=0pt,
  line width=0pt,
  minimum height=8pt,
]

\newlength\@wall@moonRaise
\setlength{\@wall@moonRaise}{-0.7pt}

\newcommand\NewMoon{}
\newcommand\FirstQuarter{}
\newcommand\FullMoon{}
\newcommand\LastQuarter{}

\newcommand\@wall@NewMoon{%
```

```
\raisebox{\@wall@moonRaise}{%
  \tikz\node[moon circle, fill=mooncolor] {};%
}}

\newcommand\@wall@FullMoon{%
\raisebox{\@wall@moonRaise}{%
  \tikz\node[moon circle, draw, mooncolor, line width=0.3pt] {};%
}}

\newcommand\@wall@FirstQuarter{%
\raisebox{\@wall@moonRaise}{%
\begin{tikzpicture}
\node [moon circle, name=waxing] {};
\path[fill=mooncolor]
  (waxing.north) --
  (waxing.south) to[out=-180,in=-90]
  (waxing.west) to[out=90,in=-180]
  (waxing.north);
\end{tikzpicture}%
}}

\newcommand\@wall@LastQuarter{%
\raisebox{\@wall@moonRaise}{%
\begin{tikzpicture}
\node [moon circle, name=waning] {};
\path[fill=mooncolor]
  (waning.north) --
  (waning.south) to[out=0,in=-90]
  (waning.east) to[out=90,in=0]
  (waning.north);
\end{tikzpicture}%
}}

\newcommand\@wall@useDefaultMoons{%
\renewcommand\NewMoon{\@wall@NewMoon}%
\renewcommand\FirstQuarter{\@wall@FirstQuarter}%
\renewcommand\FullMoon{\@wall@FullMoon}%
\renewcommand\LastQuarter{\@wall@LastQuarter}%
}

\newcommand\plannerMoonSize{\@setfontsize\plannerMoonSize{14}{14}}
\newcommand*\plannerMoonFormat[1]{{\plannerMoonSize #1}}

\newcommand\@wall@planner@NewMoon{\plannerMoonFormat{\CIRCLE}}
\newcommand\@wall@planner@FirstQuarter{\plannerMoonFormat{\LEFTcircle}}
\newcommand\@wall@planner@FullMoon{\plannerMoonFormat{\Circle}}
\newcommand\@wall@planner@LastQuarter{\plannerMoonFormat{\RIGHTcircle}}

\newcommand\@wall@usePlannerMoons{%
\renewcommand\NewMoon{\@wall@planner@NewMoon}%
\renewcommand\FirstQuarter{\@wall@planner@FirstQuarter}%
\renewcommand\FullMoon{\@wall@planner@FullMoon}%
\renewcommand\LastQuarter{\@wall@planner@LastQuarter}%
}

\@wall@useDefaultMoons
```

9 USER COMMANDS

9.1 \SetPhoto

```
\SetPhoto[bleed=3mm]{June}
```

```
\newcommand\SetPhoto[2][]{%
  \pgfkeys{%
    /Photo/#2/.is family, /Photo/#2,
    /Photo/init,
    defaults, file={#2},
    #1%
  }%
}
```

9.2 \SetCalendar

```
\SetCalendar[bg={opacity=0.8}]{June}
```

```
\newcommand\SetCalendar[2][]{%
  \pgfkeys{%
    /Calendar/#2/.is family, /Calendar/#2,
    /Calendar/init,
    defaults,
    #1%
  }%
}
```

9.3 \SetQuote

```
\SetQuote[position=top right, text align=right]{June}{%
first line\\
second line\\
third line\\
fourth line
}
```

```
\newcommand\SetQuote[3][]{%
  \pgfkeys{%
    /Quote/#2/.is family, /Quote/#2,
    /Quote/init,
    defaults, text={#3},
    #1%
  }%
}
```

9.4 \SetEvents

FIXME: Needs at least a \SetEvents{#2} to set defaults.

```
\SetEvents[yshift={10pt}]{August}{%
marks
}{%
text
}
```

```
#1 : options
#2 : month name
#3 : marks
#4 : text
```

```
\newcommand\SetEvents[4][]{%
  \pgfkeys{%
    /Events/#2/.is family, /Events/#2,
    /Events/init,
    defaults, marks={#3}, text={#4},
    #1%
  }%
}
```

9.5 \MonthPage

The /MonthPage key is set in-place for each page, not collecting options per month.

```
\pgfkeys{
  /MonthPage/.is family, /MonthPage,
  layout/.style = {layout handlers/#1/.get = \@wall@month@doLayout},
  put photo/.style = {/put photo/#1},
  defaults/.style = {layout = small landscape, put photo = simple},
  layout handlers/.cd,
    full page/.initial = \@wall@fullPageLayout,
    small landscape/.initial = \@wall@smallLandscapeLayout,
}
```

FIXME: handle case when no \SetPhoto{#2} was called, and so options are not initialized. This happens for example when bleed value is missing, and .get returns \@val:

```
! Missing number, treated as zero.
<to be read again>
\@val
l.30 \MonthPage[layout=full page]{August}
```

For now, just make sure there is a \SetPhoto{August} before \MonthPage{August}.

```
\MonthPage[layout=full page]{June}
```

```
\newcommand\MonthPage[2][]{%
  \pgfkeys{/MonthPage, defaults, #1}%
  \@wall@month@doLayout{#2}%
}
```

9.6 \SetTxt

```
\SetTxt{August Quote}{%
The text of the quote.
}
```

```
#1 : key
#2 : text
```

```
\newcommand\SetTxt[2]{%
  \pgfkeys{%
    /Txt/#1/.is family, /Txt/#1,
    /Txt/init,
    defaults, text={#2},
  }%
}
```

9.7 \txt

```
\txt{August Quote}
```

```
#1: text key
```

```
\newcommand\txt[1]{%
\pgfkeys{/Txt/#1/text/.get=\@val}%
\@val%
}
```

9.8 \parseMonthEvents

```
\def\@t@monthName{}
\def\@t@eventsCsv{}
\def\@t@markDefaultsCsv{}
\def\@t@filterPred{}
\def\@t@formatFunc{}
\def\@t@formatCmd{}
\def\@t@minEvents{}

\newcommand\monthMarkFmt{\color{textbody}}

\newcommand\symbolSpace{\thinspace}
\newcommand\symbolSeparator{,\symbolSpace}

% #1 = option keys
\newcommand*\parseMonthEvents[1][]{%
\pgfkeys{/parseMonthEvents, defaults, #1,
  month/.get=\@t@monthName,
  filter pred/.get=\@t@filterPred,
  format func/.get=\@t@formatFunc,
  format cmd/.get=\@t@formatCmd,
  events csv/.get=\@t@eventsCsv,
  mark defaults csv/.get=\@t@markDefaultsCsv,
  min events/.get=\@t@minEvents,
}%
\luadirect{
require("wallcalendar-helpers.lua")
monthEvents(
  \luastring{\@t@monthName},
  \@t@filterPred,
  \@t@formatFunc,
  \luastringO{\@t@formatCmd},
  \luastring{\@t@eventsCsv},
  \luastring{\@t@markDefaultsCsv},
  \@t@minEvents
)}}
```

9.9 \parseYearEvents

```
\def\@t@yearNum{}

% #1 = option keys
\newcommand*\parseYearEvents[1][]{%
\pgfkeys{/parseYearEvents, defaults, #1,
  year/.get=\@t@year,
  filter pred/.get=\@t@filterPred,
  format func/.get=\@t@formatFunc,
  format cmd/.get=\@t@formatCmd,
  events csv/.get=\@t@eventsCsv,
  mark defaults csv/.get=\@t@markDefaultsCsv,
  min events/.get=\@t@minEvents,
}%
\luadirect{
require("wallcalendar-helpers.lua")
yearEvents(
  tonumber(\@t@year),
```

```
  \@t@filterPred,
  \@t@formatFunc,
  \luastringO{\@t@formatCmd},
  \luastring{\@t@eventsCsv},
  \luastring{\@t@markDefaultsCsv},
  \@t@minEvents
)}}
```

9.10 \parseMonthMarksDayText

NOTE: Tikz will not work with parsing options as with `\parseMonthEvents`.

It will parse the CSV file set with the `eventsCsv` class option.

```
\newlength\dayTextXshift
\newlength\dayTextYshift
\setlength{\dayTextXshift}{0pt}
\setlength{\dayTextYshift}{0pt}

\newlength\markNumberAbove
\newlength\markNumberRight
\setlength{\markNumberAbove}{-10pt}
\setlength{\markNumberRight}{-3pt}

\newlength\markDayTextAbove
\newlength\markDayTextRight
\setlength{\markDayTextAbove}{-10pt}
\setlength{\markDayTextRight}{-3pt}

\newcommand\parseMonthMarksDayText{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksDayText(\luastring{\theMonthName}, nil, \luastring{\@wall@eventsCsv})
tex.sprint(';')
}}

\newcommand\parseMonthMarksDayTextMonth[1]{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksDayText(\luastring{#1}, nil, \luastring{\@wall@eventsCsv})
tex.sprint(';')
}}
```

9.11 \parseMonthMarksDayTextUsing

```
\newcommand*\parseMonthMarksDayTextUsing[1]{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksDayText(\luastring{\theMonthName}, nil, \luastring{#1})
}}

\newcommand\parseMonthMarksDayTextMonthUsing[2]{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksDayText(\luastring{#1}, nil, \luastring{#2})
```

```
}}
```

9.12 \parseMonthMarksNote

```
\newcommand*\parseMonthMarksNote{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksNote(\luastring{\theMonthName}, nil, \luastring{\@wall@eventsCsv}, \luastring{\@wall@markDefaultsCsv})
}}
```

9.13 \parseMonthMarksNoteUsing

```
\newcommand*\parseMonthMarksNoteUsing[1]{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksNote(\luastring{\theMonthName}, nil, \luastring{#1}, \luastring{\@wall@markDefaultsCsv})
}}
```

10 Page layouts

10.1 Initial setup

Applying a blank, bare pagestyle, the layout macro should position the parts of the page.

```
\makepagestyle{month}
\makeoddhead{month}{}{}{}
\makeevenhead{month}{}{}{}
\makeoddfoot{month}{}{}{}
\makeevenfoot{month}{}{}{}

\pagestyle{month}
```

These elements are common to all layouts. One Coffin for each part of the page:

- Photo
- Quote
- Calendar
- Events

```
\NewCoffin\@wall@pageWrap
\NewCoffin\@wall@photo
\NewCoffin\@wall@quote
\NewCoffin\@wall@calendar
\NewCoffin\@wall@events
```

10.2 Formatting hooks and temp vars

```
\newlength\@t@bleed
\newlength\@t@rightOffset
\newlength\@t@minipageWidth
\newlength\@t@calendar@height
\newlength\@t@calendar@hmargin
\newlength\@t@calendar@dayYshift
\newlength\@t@calendar@dayXshift
\newlength\@t@calendar@gridHeight
\newlength\@t@calendar@gridHeightFiveRows
\newlength\@t@calendar@gridHeightSixRows
\newlength\@t@xOffset
\newlength\@t@yOffset
\def\@t@file{}

\newcommand*\monthFmt{}%
\newcommand*\yearFmt{}%
\newcommand*\dayLetterColor{}%
\newcommand*\dayLetterFmt{}%
\newcommand*\dayTextFmt{}%
\newcommand*\quoteFmt{}%
\newcommand*\headingFmt{}%
\newcommand*\calendarFmt{}%
\newcommand*\eventsFmt{}%
```

10.3 Layout Reset, \@wall@layout@reset

A reset macro for the beginning of a layout, to make sure parameters are not carried from one layout to the next.

```
\newcommand\@wall@layout@reset{%
  \renewcommand*\monthFmt{}%
  \renewcommand*\yearFmt{}%
  \renewcommand*\dayLetterColor{}%
  \renewcommand*\dayLetterFmt{}%
  \renewcommand*\dayTextFmt{}%
  \renewcommand*\quoteFmt{}%
  \renewcommand*\headingFmt{}%
  \renewcommand*\calendarFmt{}%
  \renewcommand*\eventsFmt{}%
  \def\@t@file{}%
  \setlength{\@t@calendar@height}{0pt}%
  \setlength{\@t@calendar@hmargin}{0pt}%
  \setlength{\@t@calendar@dayYshift}{0pt}%
  \setlength{\@t@calendar@dayXshift}{0pt}%
  \setlength{\@t@bleed}{0pt}%
  \setlength{\@t@rightOffset}{0pt}%
  \setlength{\@t@minipageWidth}{0pt}%
  \setlength{\@t@xOffset}{0pt}%
  \setlength{\@t@yOffset}{0pt}%
  \setlength{\@t@calendar@gridHeight}{0pt}%
  \setlength{\@t@calendar@gridHeightFiveRows}{0pt}%
  \setlength{\@t@calendar@gridHeightSixRows}{0pt}%
}
```

10.4 Full Page, \@wall@fullPageLayout

10.4.1 Setup formatting hooks

Renew this in your document when customizing the variables for this layout.

This must not introduce paragraph breaks or whitespace characters, so follow everything with %.

```
\newcommand\fullPageFmt{%
  \renewcommand*\monthFmt{\LARGE}%
  \renewcommand*\yearFmt{\LARGE}%
  \renewcommand*\dayLetterColor{}%
  \renewcommand*\dayLetterFmt{\tiny}%
  \renewcommand*\dayTextFmt{\small}%
  \renewcommand*\quoteFmt{}%
  \renewcommand*\headingFmt{\centering}%
  \renewcommand*\calendarFmt{\centering}%
  \renewcommand*\eventsFmt{\setlength{\parindent}{0pt}\raggedleft\footnotesize}%
}
```

10.4.2 Init

```
\newcommand\@wall@fullPageLayout[2][]{%
\makeatletter
\renewcommand*\theMonthName{#2}
\@wall@layout@reset
\fullPageFmt

\colorlet{weekday}{black}
\colorlet{weekend}{black!70}
```

10.4.3 Lengths and sizes

Bleed:

```
\pgfkeys{/Photo/#2/bleed/.get=\@val}
\setlength{\@t@bleed}{\@val}
```

```
\def\@t@monthnum{\monthToNum{#2}}
```

Calendar height:

```
% 40mm = 1.5748 inch
% 43mm = 1.6929 inch
\setlength{\@t@calendar@height}{40mm + \@t@bleed}

% See if there was a height given in the options
\pgfkeys{/Calendar/#2/minimum height/.get=\@val}
```

```
\ifx\@val\empty \relax
\else
  \setlength{\@t@calendar@height}{\@val + \@t@bleed}
\fi
```

Calendar horizontal margin:

```
\setlength{\@t@calendar@hmargin}{20pt}
```

Calculate offsets:

```
% NOTE: the -4pt and -2pt offset is somehow necessary for the sides to align
% with the edges
\setlength{\@t@rightOffset}{\@t@bleed +\@t@calendar@hmargin -4pt}
\setlength{\@t@minipageWidth}{\calPaperWidth -2\@t@calendar@hmargin -2pt}
```

10.4.4 Set the coffin contents

Page wrap:

```
\SetHorizontalCoffin\@wall@pageWrap{%
  \color{white}%
  \rule{\textwidth}{0pt}%
  \rule{0pt}{\textheight}%
}
```

Photo:

```
\pgfkeys{/Photo/#2/file/.get=\@t@file}

\SetHorizontalCoffin\@wall@photo{%
  \placeholder[fill=photo-frame]{hasvarnish}{%
    \@wall@photo@putPhoto{\@t@file}%
  }%
}
```

Quote:

```
\SetHorizontalCoffin\@wall@quote{%
  \placeholder[fill=quote-frame]{}{%
    \begin{minipage}{\linewidth}%
      \quoteFmt
      \pgfkeys{/Quote/#2/text/.get=\@val}%
      \@val%
    \end{minipage}%
  }%
}
```

Calendar:

```
\SetHorizontalCoffin\@wall@calendar{%
\ifvarnishmask\relax
\else
\begin{tikzpicture}

\node (bg) [
  fill = calendarbg,
  opacity = 0.5,
  minimum width = {\calPaperWidth + 2\@t@bleed},
  anchor=north west,
  /Calendar/#2/bg,
  % override the style, in case bleed was set above
  minimum height = {\@t@calendar@height},
] at (0,0) {};

\node (heading) [
  below right=10pt and {\@t@rightOffset} of bg.north west,
  anchor=north west,
] {%
\placeholder[fill=heading-frame]{}{%
\begin{minipage}{\@t@minipageWidth}%
\headingFmt
{\monthFmt\@tr@monthNumName{\@t@monthnum}}
\hfill
\strcmpblank{\CalendarAltYear}{%
   {\yearFmt\CalendarYear}%
}{%
  {\yearFmt\CalendarYear/\CalendarAltYear}%
}%
\end{minipage}%
}%
};

\node (calendar) [
  below right=35pt and {\@t@rightOffset} of bg.north west,
  anchor=north west,
] {%
\placeholder[fill=calendar-frame]{}{%
\begin{minipage}{\@t@minipageWidth}%
\calendarFmt
\tikz{\tikzMonthCalendar@fullpage{\@t@monthnum};}%
\end{minipage}%
}%
};

\node (events) [
  below=0pt of calendar.south west,
  anchor=north west,
] {%
\placeholder[fill=events-frame]{}{%
\begin{minipage}{\@t@minipageWidth}%
\eventsFmt
\pgfkeys{/Events/#2/text/.get=\@val}%
\@val%
\end{minipage}%
}%
};

\end{tikzpicture}%
\fi
}% \SetHorizontalCoffin
```

10.4.5 *Join the coffins*

```
\@wall@photo@setYOffset

\pgfkeys{/Photo/#2/xOffset/.get=\@val}
\addtolength{\@t@xOffset}{\@val}
\pgfkeys{/Photo/#2/yOffset/.get=\@val}
\addtolength{\@t@yOffset}{\@val}

% NOTE: a small whitespace is visible on the left edge of the page when bleed is Opt, a space must be getting in
\JoinCoffins*\@wall@pageWrap[l,t]\@wall@photo[l,t](-\spinemargin -\@t@bleed + \@t@xOffset, \uppermargin + \@t@bl

\pgfkeys{/Quote/#2/xOffset/.get=\@val}
\setlength{\@t@xOffset}{\@val}
\pgfkeys{/Quote/#2/yOffset/.get=\@val}
\setlength{\@t@yOffset}{\@val}

\JoinCoffins*\@wall@pageWrap[l,t]\@wall@quote[l,t](\@t@xOffset, \@t@yOffset)

\JoinCoffins*\@wall@pageWrap[l,b]\@wall@calendar[l,b](-\spinemargin -\@t@bleed, \textheight +\uppermargin -\pape

\makeatother
```

10.4.6 *Typeset*

```
\clearpage
\TypesetCoffin\@wall@pageWrap%
}% \@wall@fullPageLayout
```

10.4.7 *Tikz calendar*

```
#1 = number of month with leading zero
```

```
\newcommand*{\tikzMonthCalendar@fullpage}[1]{%
\pgfkeys{/Events/\monthName{#1}/marks/.get=\@eventmarks}%
\calendar (cal#1) [alnitak, dates=\CalendarYear-#1-01 to \CalendarYear-#1-last] \@eventmarks;%
}
```

10.5 Small Landscape, \@wall@smallLandscapeLayout

10.5.1 *Setup formatting hooks*

Renew this in your document when customizing the variables for this layout.

This must not introduce paragraph breaks or whitespace characters, so follow everything with %.

```
\newcommand\smallLandscapeFmt{%
  \renewcommand*\monthFmt{\LARGE}%
  \renewcommand*\yearFmt{\LARGE}%
  \renewcommand*\dayLetterColor{}%
  \renewcommand*\dayLetterFmt{\tiny}%
  \renewcommand*\dayTextFmt{\small}%
  \renewcommand*\quoteFmt{\centering}%
  \renewcommand*\headingFmt{\centering}%
  \renewcommand*\calendarFmt{\centering}%
  \renewcommand*\eventsFmt{\setlength{\parindent}{0pt}\raggedright\footnotesize}%
}
```

10.5.2 Init

```
\newcommand\@wall@smallLandscapeLayout[2][]{%
\makeatletter
\renewcommand*\theMonthName{#2}
\@wall@layout@reset
\smallLandscapeFmt

\colorlet{weekday}{black}
\colorlet{weekend}{black!70}
```

10.5.3 Lengths and sizes

Bleed:

```
\pgfkeys{/Photo/#2/bleed/.get=\@val}
\setlength{\@t@bleed}{\@val}
```

```
\def\@t@monthnum{\monthToNum{#2}}
```

Calendar height:

Default height to fit:

- day headings
- days in a grid, 6 rows
- events
- bottom bleed

```
\setlength{\@t@calendar@height}{85mm + \@t@bleed}

% See if there was a height given in the options
\pgfkeys{/Calendar/#2/minimum height/.get=\@val}
\ifx\@val\empty \relax
\else
  \setlength{\@t@calendar@height}{\@val + \@t@bleed}
\fi
```

Calendar horizontal margin:

```
\setlength{\@t@calendar@hmargin}{20pt}
```

Calculate offsets:

```
% 5mm: day headings
% 20mm: events
% 6.5mm: vertical spacing
\setlength{\@tmp@a}{20mm + 5mm + 6.5mm}
\setlength{\@tmp@b}{\@t@calendar@height -\@tmp@a}
\setlength{\@t@calendar@dayYshift}{0.1666\@tmp@b}% 1/6 = 0.1666

\setlength{\@t@calendar@gridHeightSixRows}{\@tmp@b}
\setlength{\@t@calendar@gridHeightFiveRows}{\@tmp@b -0.1666\@tmp@b}

% NOTE: the -4pt and -2pt offset is somehow necessary for the sides to align
% with the edges
\setlength{\@t@rightOffset}{\@t@bleed +\@t@calendar@hmargin -4pt}
\setlength{\@t@minipageWidth}{\calPaperWidth -2\@t@calendar@hmargin -2pt}

\setlength{\@t@calendar@dayXshift}{0.1428\@t@minipageWidth}% 1/7 = 0.1428
```

10.5.4 Set the coffin contents

Page wrap:

```
\SetHorizontalCoffin\@wall@pageWrap{%
  \color{white}%
  \rule{\textwidth}{0pt}%
  \rule{0pt}{\textheight}%
}
```

Photo:

```
\pgfkeys{/Photo/#2/file/.get=\@t@file}

\SetHorizontalCoffin\@wall@photo{%
  \placeholder[fill=photo-frame]{hasvarnish}{%
    \@wall@photo@putPhoto{\@t@file}%
  }%
}
```

Quote:

```
\SetHorizontalCoffin\@wall@quote{%
  \placeholder[fill=quote-frame]{}{%
    \begin{minipage}{\linewidth}%
      \quoteFmt
      \pgfkeys{/Quote/#2/text/.get=\@val}%
      \@val%
    \end{minipage}%
```

```
  }%
}
```

Calendar:

```
\SetHorizontalCoffin\@wall@calendar{%
\ifvarnishmask\relax
\else
\begin{tikzpicture}

\node (bg) [
  fill = calendarbg,
  opacity = 1,
  minimum width = {\calPaperWidth + 2\@t@bleed},
  anchor=north west,
  /Calendar/#2/bg,
  % override the style, in case bleed was set above
  minimum height = {\@t@calendar@height},
] at (0,0) {};

\node (heading) [
  below right=10pt and {\@t@rightOffset} of bg.north west,
  anchor=north west,
] {%
\placeholder[fill=heading-frame]{}{%
\begin{minipage}{\@t@minipageWidth}%
\headingFmt
{\monthFmt\@tr@monthNumName{\@t@monthnum}}
\hfill
\strcmpblank{\CalendarAltYear}{%
   {\yearFmt\CalendarYear}%
}{%
  {\yearFmt\CalendarYear/\CalendarAltYear}%
}%
\end{minipage}%
}%
};

\node (calendar) [
  below right=35pt and {\@t@rightOffset} of bg.north west,
  anchor=north west,
] {%
\placeholder[fill=calendar-frame]{}{%
\begin{minipage}{\@t@minipageWidth}%
\calendarFmt
\hspace*{-5pt}% FIXME
\tikz{\tikzMonthCalendar@smallLandscape{\@t@monthnum};}%
\end{minipage}%
}%
};

\node (events) [
  above right=10mm and {\@t@rightOffset} of bg.south west,
  anchor=south west,
] {%
\placeholder[fill=events-frame]{}{%
\begin{minipage}{\@t@minipageWidth}%
\eventsFmt
\pgfkeys{/Events/#2/text/.get=\@val}%
\@val%
\end{minipage}%
```

```
}%
};

\end{tikzpicture}%
\fi
}% \SetHorizontalCoffin
```

10.5.5 *Join the coffins*

Only using `yOffset`. The quote should be centered on the x axis.

Join coffins so that the photo is in a fixed position, i.e. relative to the pageWrap, not relative to other coffins. It prevents accidental shifts when the other coffins are empty (when `varnishmask` is on) or have too much content.

```
% The quote has to be centered between the photo and the calendar with manual tweaking.
% The height of the quote is not known, the height of the calendar plus its heading is not known.

% TODO move the photo with /Photo/#2/xOffset and yOffset as at the full page layout

\pgfkeys{/Quote/#2/yOffset/.get=\@val}
\setlength{\@t@yOffset}{\@val}

\JoinCoffins*\@wall@photo[hc,b]\@wall@quote[hc,t](0pt, \@t@yOffset)

\pgfkeys{/Photo/#2/yOffset/.get=\@val}
\setlength{\@t@yOffset}{\@val}

% NOTE: Taking the bottom edge of the photo as fixed.
% yOffset must be set to pull the photo down into position

\JoinCoffins*\@wall@pageWrap[hc,t]\@wall@photo[hc,b](0pt, \uppermargin + \@t@yOffset)

% Calendar is aligned to the bottom of the page.

\JoinCoffins*\@wall@pageWrap[hc,b]\@wall@calendar[hc,b](0pt, \textheight +\uppermargin -\paperheight -\@t@bleed)

\makeatother
```

10.5.6 *Typeset*

```
\clearpage
\TypesetCoffin\@wall@pageWrap%
}% \@wall@smallLandscapeLayout
```

10.5.7 *Tikz calendar*

```
#1 = number of month with leading zero
```

```
\newcommand*{\tikzMonthCalendar@smallLandscape}[1]{%
\pgfkeys{/Events/\monthName{#1}/marks/.get=\@eventmarks}%
\calendar (cal#1) [betelgeuse, dates=\CalendarYear-#1-01 to \CalendarYear-#1-last] \@eventmarks;%
}
```

10.6 Tikz styles

10.6.1 day letter headings

```
\tikzstyle{day letter headings}=[%
  day heading/.style={black!90},
  execute before day scope={%
    \ifdate{day of month=1}{%
      \pgfmathsetlength\pgf@xa{\tikz@lib@cal@xshift}%
      \pgfmathsetlength\pgf@ya{\tikz@lib@cal@yshift}%
      \foreach \d in {0,1,2,3,4,5,6} {%
        \pgf@xa=\d\pgf@xa%
        \pgftransformxshift{\pgf@xa}%
        \pgftransformyshift{\pgf@ya}%
        \node (d\d) [anchor=south, day heading] {\dayLetterFmt\@tr@dayLetter{\d}};%
      };%
    }{}%
  },%
]
```

10.6.2 days grid

```
\newcount\gridRows
\newcount\gridLines
\newcount\n

\newif\ifGridNoSurround
\GridNoSurroundfalse

\tikzstyle{no grid surround}=[execute before day scope={\GridNoSurroundtrue}]
\tikzstyle{grid surround}=[execute before day scope={\GridNoSurroundfalse}]

\tikzstyle{days grid}=[%
  execute before day scope={%
    \ifdate{day of month=1}{%
      %
      % Determine if the grid is five or six rows
      %
      % 31 day months
      \ifdate{between=01-01 and 01-31, between=03-01 and 03-31, between=05-01 and 05-31, between=07-01 and 07-31
        \ifdate{Saturday,Sunday}{%
          \gridRows=6%
          \setlength\@t@calendar@gridHeight{\@t@calendar@gridHeightSixRows}%
        }{%
          \gridRows=5%
          \setlength\@t@calendar@gridHeight{\@t@calendar@gridHeightFiveRows}%
        }
      }{
      % 30 day months and February
```

```
      \ifdate{between=02-01 and 02-last}{%
        \gridRows=5%
        \setlength\@t@calendar@gridHeight{\@t@calendar@gridHeightFiveRows}%
      }{%
        \ifdate{Sunday}{%
          \gridRows=6%
          \setlength\@t@calendar@gridHeight{\@t@calendar@gridHeightSixRows}%
        }{%
          \gridRows=5%
          \setlength\@t@calendar@gridHeight{\@t@calendar@gridHeightFiveRows}%
        }
      }
    }
    %
    % Horizontal lines
    %
    \gridLines=\gridRows
    \ifGridNoSurround\relax
    \else
      \advance\gridLines by 1
      \n=0
      \draw [
        gridcolor,
        line width=0.3pt,
        xshift=-0.5\@t@calendar@dayXshift,
        yshift=0.5\@t@calendar@dayYshift,
      ] (0,{-\n\@t@calendar@dayYshift}) -- (7\@t@calendar@dayXshift,{-\n\@t@calendar@dayYshift});
    \fi
    \foreach \n in {1,2,3,4,5,6} {
      \ifnum\n<\gridLines
        \draw [
          gridcolor,
          line width=0.3pt,
          xshift=-0.5\@t@calendar@dayXshift,
          yshift=0.5\@t@calendar@dayYshift,
        ] (0,{-\n\@t@calendar@dayYshift}) -- (7\@t@calendar@dayXshift,{-\n\@t@calendar@dayYshift});
      \fi
    }
    %
    % Vertical lines
    %
    \gridLines=7
    \ifGridNoSurround\relax
    \else
      \advance\gridLines by 1
      \n=0
      \draw [
        gridcolor,
        line width=0.3pt,
        xshift=-0.5\@t@calendar@dayXshift,
        yshift=0.5\@t@calendar@dayYshift,
      ] ({\n\@t@calendar@dayXshift},0) -- ({\n\@t@calendar@dayXshift},{-\gridRows\@t@calendar@dayYshift});
    \fi
    \foreach \n in {1,2,3,4,5,6,7} {
      \ifnum\n<\gridLines
        \draw [
          gridcolor,
          line width=0.3pt,
          xshift=-0.5\@t@calendar@dayXshift,
          yshift=0.5\@t@calendar@dayYshift,
        ] ({\n\@t@calendar@dayXshift},0) -- ({\n\@t@calendar@dayXshift},{-\gridRows\@t@calendar@dayYshift});
      \fi
    }
```

```
      %
      % If we are not drawing the surrounding lines, it looks better to hide
      % some of the top and bottom edges
      %
      \ifGridNoSurround
        \draw [
          color=white,
          line width=6pt,
          xshift=-0.5\@t@calendar@dayXshift,
          yshift={0.5\@t@calendar@dayYshift -1pt},
        ] (0,0) -- (7\@t@calendar@dayXshift,0);
        \draw [
          color=white,
          line width=6pt,
          xshift=-0.5\@t@calendar@dayXshift,
          yshift={0.5\@t@calendar@dayYshift +1pt},
        ] (0,-\gridRows\@t@calendar@dayYshift) -- (7\@t@calendar@dayXshift,-\gridRows\@t@calendar@dayYshift);
      \fi
    }{}%
  }
]
```

10.6.3 *headings background rule*

```
\tikzstyle{headings background rule}=[
  background rule/.style={black!90, line width=0.3pt, yshift={0.3\@t@calendar@dayYshift -3pt}},
  execute before day scope={%
    \ifdate{day of month=1}{%
      \draw [
        xshift=-0.5\@t@calendar@dayXshift,
        yshift=0.5\@t@calendar@dayYshift,
        background rule,
      ] (0,0) -- (7\@t@calendar@dayXshift,0);
    }{}%
  }
]
```

10.6.4 *headings background bar*

```
\tikzstyle{headings background bar}=[
  headings background rule,
  background rule/.style={black!20, line width=10pt, yshift={0.3\@t@calendar@dayYshift +3.5pt}},
]
```

10.7 Tikz calendar styles

10.7.1 *Betelgeuse, days in a grid*

(Alpha Orionis)

```
\tikzstyle{betelgeuse}=[
  no grid surround,
  days grid,
  week list,
  day xshift=\@t@calendar@dayXshift,
  day yshift=\@t@calendar@dayYshift,
  headings background bar,
  day letter headings,
  day heading/.style={black!90, yshift={-0.3\@t@calendar@dayYshift}},
  every day/.append style={anchor=base, inner xsep=0pt, yshift={-0.25\baselineskip}},
  day text={\dayTextFmt\%d-},
]
```

10.7.2 Alnitak, days in one line

(Zeta Orionis)

```
\tikzstyle{alnitak}=[
  day list right,
  day xshift={0pt},
  every month/.append style={anchor=base, inner xsep=0pt, yshift=4mm, xshift=-\widthof{\space}},
  day text={\dayTextFmt\%d-},
  every day/.append style={anchor=base, inner xsep=0pt},
  execute before day scope={
    %
    % === Moving the day number ===
    %
    % A small offset seems to improve the result, the last day gets closer to the end of the linewidth
    \pgftransformxshift{0.01em}
    \pgfcalendarjuliantodate{\pgfcalendarendjulian}{\currentyear}{\currentmonth}{\lastday}
    % Width of month day digits from 1 - 28
    \setlength{\@tmp@a}{\widthof{1234567891011121314151617181920212223242526272 8}}
    \setlength{\@tmp@b}{\widthof{30}}
    \pgfmathparse{(\linewidth - \@tmp@a - \@tmp@b * (\lastday-28)) / \lastday}
    \pgftransformxshift{\pgfmathresult}
    \let\%=\pgfcalendarshorthand
    \setlength{\@tmp@a}{\widthof{\%d-}}
    \pgftransformxshift{\@tmp@a}
    %
    % === Weekday letter above the day ===
    %
    \ifdate{weekend}{\def\dayLetterColor{\color{weekend}}}{\def\dayLetterColor{\color{weekday}}}%
    \node[anchor=south,yshift=5mm,inner sep=0pt]{%
      \dayLetterColor\dayLetterFmt\@tr@dayLetter{\pgfcalendarcurrentweekday}%
    };%
  },
]
```

11 YEAR PLANNER

11.1 TIKZ CALENDAR STYLES

11.1.1 Year Planner Portrait

```
\newcommand\plannerPortraitMonthFmt{\fontsize{11}{11}\selectfont\color{darkgold}}
\newcommand\plannerPortraitDayFmt{\fontsize{8}{8}\selectfont\color{plannerdatenum}}

\tikzstyle{year planner portrait}=[
  week list,
  month text={\plannerPortraitMonthFmt\@tr@monthNumName{\pgfcalendarcurrentmonth}},
  day text={\plannerPortraitDayFmt\%d-},
  every month/.append style={%
    anchor=south,
    inner xsep=0pt,
    yshift=5mm,
    xshift=2.5mm,
  },
  day letter headings,
  day heading/.style={gray, xshift=0pt, scale=0.85},
  month label above centered,
  every day/.style={anchor=mid},
]
```

11.1.2 Year Planner Landscape

```
\newcommand\plannerLandscapeDateDayFont{}
\newcommand\plannerLandscapeDayFont{}
\newcommand\plannerLandscapeMonthFont{}

\newcommand\plannerLandscapeMonthFmt{%
  \plannerLandscapeMonthFont%
  \fontsize{15}{18}\selectfont%
  \color{gold}\bfseries%
}

\newcommand\plannerLandscapeDayFmt{%
  \plannerLandscapeDayFont%
  \fontsize{10}{12}\selectfont%
}

\newcommand*\plannerLandscapeDateDayFormat[1]{%
  \plannerLandscapeDateDayFont%
  \fontsize{8}{11}\selectfont%
  #1%
}

\newcommand\plannerLandscapeBeginDayScopeHook{%
  \ifdate{weekend}{\color{black!60}}{}
}

\newlength\plannerLandscapeDayXShift
\newlength\plannerLandscapeMonthYShift

% Assuming 25mm to fit 'September'
\setlength{\@tmp@a}{\calPaperWidth - \@wall@leftMargin - \@wall@rightMargin - 25mm}
% Max week shift is 6 = Sunday. 31 days + 6 = 37. 1/37 = 0.027
\setlength{\plannerLandscapeDayXShift}{0.027\@tmp@a}

% Assuming 10mm for day headings (M T W...) and some padding
\setlength{\@tmp@a}{\calPaperHeight - \@wall@topMargin - \@wall@bottomMargin - 10mm}
% 1/12 = 0.0833
\setlength{\plannerLandscapeMonthYShift}{0.0833\@tmp@a}
```

```
\tikzstyle{year planner landscape}=[
  month list,
  day xshift=\plannerLandscapeDayXShift,
  month yshift=\plannerLandscapeMonthYShift,
  every month/.append style={anchor=base, inner xsep=0pt},
  month text={\plannerLandscapeMonthFmt \%mt},
  day text={\plannerLandscapeDayFmt \%d-},
  every day/.append style={anchor=base, inner xsep=0pt},
  month label left,
  execute at begin day scope={
    \plannerLandscapeBeginDayScopeHook
    % Weekday headers in January
    \ifdate{between=01-01 and 01-31}{
      \def\l{\ifcase\pgfcalendarcurrentweekday M\or T\or W\or T\or F\or S\or S\fi}%
      \ifdate{weekend}{\def\daylabelcolor{gold}}{\def\daylabelcolor{black}}%
      \node[anchor=south,yshift=5mm,inner sep=0pt]{\color{\daylabelcolor}\plannerLandscapeDateDayFormat{\l}};%
    }{}
    % Weekday headers in December
    \ifdate{between=12-01 and 12-31}{
      \pgftransformyshift{-3mm}%
      \def\l{\ifcase\pgfcalendarcurrentweekday M\or T\or W\or T\or F\or S\or S\fi}%
      \ifdate{weekend}{\def\daylabelcolor{gold}}{\def\daylabelcolor{black}}%
      \node[anchor=south,yshift=5mm,inner sep=0pt]{\color{\daylabelcolor}\plannerLandscapeDateDayFormat{\l}};%
    }{}
  }
]
```

11.1.3 *Year Planner Landscape Grid*

```
\newcommand\plannerGridMonthFont{}
\newcommand\plannerGridDateDayFont{}
\newcommand\plannerGridNotesFont{}
\newcommand\plannerGridDayFont{}
\newcommand\plannerGridYearNumberFont{}

\newcommand\plannerGridYearNumberFmt{%
  \plannerGridYearNumberFont%
  \fontsize{11}{11}\selectfont%
  \color{white}%
}

\newcommand\plannerGridMonthFmt{%
  \plannerGridMonthFont%
  \fontsize{13}{13}\selectfont%
  \color{white}%
}

\newcommand\plannerGridDayFmt{%
  \plannerGridDayFont%
  \fontsize{11}{11}\selectfont%
}

\newcommand\plannerGridDateDayFmt{%
  \plannerGridDateDayFont%
  \fontsize{9}{9}\selectfont%
  \color{black!80}%
}
```

```
\newcommand{\plannerGridNotesFmt}{%
  \plannerGridNotesFont%
  \fontsize{11}{11}\selectfont%
}

\newcommand\plannerGridBeginDayScopeHook{%
  \ifdate{weekend}{\color{black!60}}{}
}

\newlength\plannerGridColXShift
\newlength\plannerGridRowYShift

\newlength\plannerGridCalendarWidth
\newlength\plannerGridCalendarHeight

\newlength\plannerGridDayHeadingsHeight
\newlength\plannerGridMonthNamesWidth

\newlength\plannerGridNotesHeight
\setlength{\plannerGridNotesHeight}{10mm}

\setlength{\plannerGridCalendarWidth}{\calPaperWidth - \@wall@leftMargin - \@wall@rightMargin}
\setlength{\plannerGridCalendarHeight}{\calPaperHeight - \@wall@topMargin - \@wall@bottomMargin - \plannerGridNo

\setlength{\plannerGridDayHeadingsHeight}{10mm}
\setlength{\plannerGridMonthNamesWidth}{10mm}

\setlength{\@tmp@a}{\plannerGridCalendarWidth - \plannerGridMonthNamesWidth}
% 37 columns. 1/37 = 0.027. Max week shift is 6 = Sunday. 31 days + 6 = 37.
\setlength{\plannerGridColXShift}{0.027\@tmp@a}

\setlength{\@tmp@a}{\plannerGridCalendarHeight - 2\plannerGridDayHeadingsHeight}
% 12 rows. 1/12 = 0.0833
\setlength{\plannerGridRowYShift}{0.0833\@tmp@a}

\newcounter{verticalCount}

\tikzstyle{year planner landscape grid days}=[
  month list,
  day xshift=\plannerGridColXShift,
  month yshift=\plannerGridRowYShift,
  day text={\plannerGridDayFmt \%d-},
  every day/.append style={anchor=base, yshift=-12pt, inner xsep=0pt},
  execute at begin day scope={%
    % White fill to cover the page background color
    \node (cell) [
      rectangle,
      fill = white,
      minimum height = \plannerGridRowYShift - 1pt,
      minimum width = \plannerGridColXShift - 1pt,
      xshift = -0.5\plannerGridColXShift,
      yshift = 4.5pt,
    ] {};%
    %
    \ifdate{Monday}{%
      \node (a) [above left = 15pt and -3.5pt of cell.west] {};
      \node (b) [below left = 15pt and -3.5pt of cell.west] {};
      \node (c) [right = 10pt of cell.west] {};
      \path [on layer = one, fill = pagebgcolor] (a.south) -- (b.north) -- (c.west) -- (a.south);%
    }{}%
    %
    \ifdate{Sunday}{%
      \node (a) [above left = 15pt and {-3.5pt + 1pt} of cell.east] {};
```

```
      \node (b) [below left = 15pt and {-3.5pt + 1pt} of cell.east] {};
      \node (c) [right = {10pt - 1pt} of cell.east] {};
      \path [on layer = two, fill = white] (a.south) -- (b.north) -- (c.west) -- (a.south);%
    }{}%
    %
    \plannerGridBeginDayScopeHook%
  },
]
```

11.2 Helpers

```
\newcommand\plannerMarksDayTextCSV{\@wall@eventsCsv}
\newcommand\plannerMarksNoteCSV{\@wall@eventsCsv}
\newcommand\plannerMarkDefaultsCsv{}

\newcommand\@wall@plm[1]{%
\luadirect{
require("wallcalendar-helpers.lua")
monthMarksDayText(\luastring{#1}, nil, \luastring{\plannerMarksDayTextCSV})
tex.sprint(';')
}}

\newcommand\@wall@plmYear{%
\luadirect{
require("wallcalendar-helpers.lua")
yearMarksDayText(tonumber(\CalendarYear), nil, \luastring{\plannerMarksDayTextCSV})
tex.sprint(';')
}}

\newcommand\@wall@plNotes{%
\luadirect{
require("wallcalendar-helpers.lua")
yearMarksNote(tonumber(\CalendarYear), nil, \luastring{\plannerMarksNoteCSV}, \luastring{\plannerMarkDefaultsCsv
}}

\newcommand\@wall@plNotesOneCalendar{%
\luadirect{
require("wallcalendar-helpers.lua")
yearMarksNote(tonumber(\CalendarYear), nil, \luastring{\plannerMarksNoteCSV}, \luastring{\plannerMarkDefaultsCsv
}}

\newcommand\plannerEvents{%
\parseYearEvents[filter pred = hasNote]%
}
```

11.3 \YearPlannerPortrait

```
\newcommand\YearPlannerPortrait{%
\@wall@useDefaultMoons%
\begin{tikzpicture}[every calendar/.style={year planner portrait}]%
  \matrix[column sep=1.5em, row sep=5mm] {
    \calendar(cal01)[dates=\CalendarYear-01-01 to \CalendarYear-01-last] \@wall@plm{January};   \pgfmatrixnextce
    \calendar(cal02)[dates=\CalendarYear-02-01 to \CalendarYear-02-last] \@wall@plm{February};  \pgfmatrixnextce
    \calendar(cal03)[dates=\CalendarYear-03-01 to \CalendarYear-03-last] \@wall@plm{March};     \\
    \calendar(cal04)[dates=\CalendarYear-04-01 to \CalendarYear-04-last] \@wall@plm{April};     \pgfmatrixnextce
```

```
    \calendar(cal05)[dates=\CalendarYear-05-01 to \CalendarYear-05-last] \@wall@plm{May};       \pgfmatrixnextce
    \calendar(cal06)[dates=\CalendarYear-06-01 to \CalendarYear-06-last] \@wall@plm{June};      \\
    \calendar(cal07)[dates=\CalendarYear-07-01 to \CalendarYear-07-last] \@wall@plm{July};      \pgfmatrixnextce
    \calendar(cal08)[dates=\CalendarYear-08-01 to \CalendarYear-08-last] \@wall@plm{August};    \pgfmatrixnextce
    \calendar(cal09)[dates=\CalendarYear-09-01 to \CalendarYear-09-last] \@wall@plm{September}; \\
    \calendar(cal10)[dates=\CalendarYear-10-01 to \CalendarYear-10-last] \@wall@plm{October};   \pgfmatrixnextce
    \calendar(cal11)[dates=\CalendarYear-11-01 to \CalendarYear-11-last] \@wall@plm{November};  \pgfmatrixnextce
    \calendar(cal12)[dates=\CalendarYear-12-01 to \CalendarYear-12-last] \@wall@plm{December};  \\
  };
  \@wall@plNotes
\end{tikzpicture}%
}
```

11.4 \YearPlannerLandscape

```
\newcommand\YearPlannerLandscape{%
\@wall@usePlannerMoons%
\begin{tikzpicture}%
  \calendar (cal) [
    year planner landscape,
    dates=\CalendarYear-01-01 to \CalendarYear-12-31,
  ] \@wall@plmYear;
  \@wall@plNotesOneCalendar
\end{tikzpicture}%
}
```

11.5 \YearPlannerLandscapeGrid

```
\newcommand\YearPlannerLandscapeGrid{%
\@wall@usePlannerMoons%
\begin{tikzpicture}

  % Background frame

  \node (bg) [
    rectangle,
    inner sep = 0pt,
    minimum width = \plannerGridCalendarWidth,
    minimum height = \plannerGridCalendarHeight,
  ] {};

  \foreach \x/\m in {1/\xJanShort, 2/\xFebShort, 3/\xMarShort, 4/\xAprShort,
    5/\xMayShort, 6/\xJunShort, 7/\xJulShort, 8/\xAugShort, 9/\xSepShort,
    10/\xOctShort, 11/\xNovShort, 12/\xDecShort}
  {
    \setlength{\@tmp@a}{\x\plannerGridRowYShift + \plannerGridDayHeadingsHeight}

    % Month labels

    \node [
      below = \@tmp@a of bg.north west,
      yshift = 0.5\plannerGridRowYShift,
      xshift = 0.5\plannerGridColXShift - 2pt,
      anchor = base,
      rotate = 90,
```

```
    ] {\plannerGridMonthFmt\MakeUppercase{\textls*{\m}}};

    % Horizontal lines

    \addtolength{\@tmp@a}{-\plannerGridRowYShift}

    \node (h-a-\x) [below = \@tmp@a of bg.north west] {};
    \node (h-b-\x) [below = \@tmp@a of bg.north east] {};
    \draw [gridcolor] (h-a-\x) -- (h-b-\x);
  }

  % 13th line
  \setlength{\@tmp@a}{12\plannerGridRowYShift + \plannerGridDayHeadingsHeight}
  \node (h-a-13) [below = \@tmp@a of bg.north west] {};
  \node (h-b-13) [below = \@tmp@a of bg.north east] {};
  \draw [gridcolor] (h-a-13) -- (h-b-13);

  % Year number in the corner
  \node [
    right = 0pt of bg.north west,
    xshift = 0.5\plannerGridMonthNamesWidth,
    yshift = -0.5\plannerGridDayHeadingsHeight - 5pt,
    anchor = base,
  ] {\plannerGridYearNumberFmt\CalendarYear};

  \setcounter{verticalCount}{1}

  \foreach \i in {0,1,2,3,4,5} {
    \foreach \j/\d in {
      0/\xMondayDayLetter,
      1/\xTuesdayDayLetter,
      2/\xWednesdayDayLetter,
      3/\xThursdayDayLetter,
      4/\xFridayDayLetter,
      5/\xSaturdayDayLetter,
      6/\xSundayDayLetter}
    {
      \addtocounter{verticalCount}{1}
      \ifnumgreater{\value{verticalCount}}{38}{\relax}{
        \setlength{\@tmp@a}{7\plannerGridColXShift}
        \setlength{\@tmp@a}{\i\@tmp@a + \j\plannerGridColXShift + \plannerGridMonthNamesWidth}

        % Day headings (top and bottom)

        \node [
          right = \@tmp@a of bg.north west,
          xshift = 0.5\plannerGridColXShift + 4pt,
          yshift = -0.5\plannerGridDayHeadingsHeight - 5pt,
          anchor = base,
        ] {\plannerGridDateDayFmt \d};

        \node [
          right = \@tmp@a of bg.south west,
          xshift = 0.5\plannerGridColXShift + 4pt,
          yshift = 0.5\plannerGridDayHeadingsHeight - 5pt,
          anchor = base,
        ] {\plannerGridDateDayFmt \d};

        % % Vertical lines

        % \node (v-a-\i-\j) [right = \@tmp@a of bg.north west] {};
        % \node (v-b-\i-\j) [right = \@tmp@a of bg.south west] {};
        % \draw [gridcolor] (v-a-\i-\j) -- (v-b-\i-\j);
```

```
      }
    }
  }

  \calendar (cal) [
    year planner landscape grid days,
    dates=\CalendarYear-01-01 to \CalendarYear-12-31,
    below right = 0pt and 0pt of bg.north west,
    anchor = north west,
    yshift = -\plannerGridDayHeadingsHeight - 3mm,
    xshift = \plannerGridMonthNamesWidth + 0.5\plannerGridColXShift + 4pt,
  ] \@wall@plmYear;

  \@wall@plNotesOneCalendar

\end{tikzpicture}%
}
```

11.6 \YearPlannerLandscapeGridPage

```
\newcommand{\YearPlannerLandscapeGridPage}{%
\pagecolor{pagebgcolor}%
\AddToShipoutPictureFG*{\put(\LenToUnit{\@wall@leftMargin},\LenToUnit{\@wall@bottomMargin})%
{\begin{minipage}[b]{\plannerGridCalendarWidth}%
\YearPlannerLandscapeGrid

\begin{minipage}[b][10mm][c]{\plannerGridCalendarWidth}%
\plannerGridNotesFmt
\hspace*{2mm}%
\parseYearEvents[%
  filter pred = hasNote,
  format cmd = {\textsuperscript{\eIdx}~\eMonthShort~\eDay:~\textit{\eNote}\ifnumless{\eIdx}{\eMaxIdx}{,\space}{
]%
\end{minipage}%
\end{minipage}%
}}}
```

12 Translation keys

```
% Load internal translations
\InputIfFileExists{i18n/wallcalendar-\@wall@calendarLanguage.tex}{}%
{\ClassError{wallcalendar}{File Not Found: i18n/wallcalendar-\@wall@calendarLanguage.tex}{}}

% Load user translations if the option was set and translationsAutoload is true

\newcommand\LoadTranslations{%
\InputIfFileExists{\@wall@translationsInputFile}{}%
{\ClassWarning{wallcalendar}{File Not Found: \@wall@translationsInputFile}{}}}

\notblank{\@wall@translationsInputFile}{%
\iftranslationsAutoload \LoadTranslations \fi}
```

Use `\ifcase` instead of a `\luadirect` array lookup, otherwise Portuguese day headings such as `\def\xMondayDayLetter{{2\textsuperscript{a}}}` cause an er-

ror.

```
#1 : month number, returns the translation
```

```
\newcommand*\@tr@monthNumName[1]{%
\ifcase#1 \relax\or \xJanuary\or \xFebruary\or \xMarch\or \xApril\or \xMay\or \xJune\or
\xJuly\or \xAugust\or \xSeptember\or \xOctober\or \xNovember\or \xDecember\fi}
```

```
#1 : weekday number, returns the one letter translation
```

```
\newcommand*\@tr@dayLetter[1]{%
\ifcase#1 \xMondayDayLetter\or \xTuesdayDayLetter\or \xWednesdayDayLetter\or
\xThursdayDayLetter\or \xFridayDayLetter\or \xSaturdayDayLetter\or \xSundayDayLetter\fi}
```

13 HELPER MACROS

Doing this in Lua to make blasted sure the result is just a string

```
#1 : month name in English, returns zero padded number
```

```
\newcommand*{\monthToNum}[1]{%
\luadirect{
  local monthToNum = {
    january   = '01',
    february  = '02',
    march     = '03',
    april     = '04',
    may       = '05',
    june      = '06',
    july      = '07',
    august    = '08',
    september = '09',
    october   = '10',
    november  = '11',
    december  = '12',
  }
  local key = string.lower('#1')
  tex.sprint(monthToNum[key])
}}
```

```
#1 : month number, returns the name in English
```

```
\newcommand*\monthName[1]{%
% \ifcase might be just fine here
\luadirect{
  local monthName = {
    'January', 'February', 'March', 'April', 'May', 'June', 'July',
    'August', 'September', 'October', 'November', 'December',
  }
  local key = tonumber('#1')
  tex.sprint(monthName[key])
}}
```

14 Epilogue

```
% End of wallcalendar.cls
```