The polycv class

Martin C Schwarzer
polyluxus@gmail.com

v1.0rc1 from 2020/06/29

Abstract

Polycv is a document class to typeset curriculum vitaes with LaTeX. It also provides a template for a cover letter consistent with the layout of the resume.

Contents

1 Introduction

I previously used the package koma-moderncvclassic to prepare my CV, and I was actually quite satisfied with the look of it. In a way this class borrows some design elements from it. However, I noticed it was not using a lot of space, which got especially problematic when inserting a picture.

I looked through various templates and classes online (limecv, simple academic resume, DeedyResume, FancyCV by Adrien Friggeri, the Overleaf CVGallery, TeX Stack Exchange LaTeX template for resume/curriculum vitae, … , there are possibly some I forgot), but couldn't find anything that was exactly to my linking. So I decided to write a class for myself, not just only for my CV, but also for the training in writing such classes (or packages).

The main goal was to structure the information in a way that, if condensed enough, it would fit on one page; but additional pages could be added when the need arose, e.g. a long publication list, or many stations along the way.

Additionally I liked the idea of having a cover letter template, which resembles most of the elements of the CV.

The source code is available at GitHub, where there are also a few example uses of this class. If anything is unclear, needs improvement, is missing please submit an issue there (GitHub issues). Thank you.

1.1 Contributing

If you find this class useful, you are very welcome to contribute to it. The repository is hosted at GitHub repository. It will consist of one main branch, the release branch, possibly a release candidate branch and a developement branch. If you would like to contribute changes, please submit a pull request to the development branch. Once all changes have been pulled in, the commits will be squashed and pulled into the release candidate branch for final testing and adjustment. As such, there will always be only one commit for one release.

2 Installation

This work consists of the files **polycv**.dtx and **polycv**.ins and the derived filebase **polycv**.cls, which should be copied to a location where latex will look for it. How to build the package and documentation from its source files is explained in section 6. The source code is hosted at GitHub, where the derived class and documentation files are also available.

2.1 Prerequisites

In order for this class to function properly, the following packages must be installed:

- The package `etoolbox` provides additional hooks necessary for the setup within the source class.
- The `calc` package allows to derive some lengths from others.
- The `hyperref` package provides clickable cross-references, e.g. to references to the bibliography, or websites.
- Colorsupport comesss from `xcolor`.
- Most of the graphical elements within this class are produced with `tikz`, such as the header and footer boxes, and the progressbar.
- To include a signature and profile pictures, `graphicx` is lloaded.
- For most of the document the indentation is disabled with the `parskip` package.
- The page layout is produced with the `geometry` package.
- The `fontawesome` is needed for icons, e.g. , in the side bar.
- To include other reference letters as pdfs, `pdfpages` is also loaded.

3 Usage

You can load the class in the same way you would load any other class. The following listing defines a skeleton document containing a elements of the class.

```
\documentclass[⟨options⟩]{polycv}
  %
  % packages,
  % define personal information,
  % define colours, page layout, etc.
  %
\begin{document}
  \begin{polycvfirstpage}
    \begin{polycvsidebar}
      %
      % Contents of the side bar box
      % (personal information image, etc.)
      %
    \end{polycvsidebar}
    %
    % Body of the first page
    %
  \end{polycvfirstpage}
  \begin{polycvpage}
    %
    % Body of additional pages
    %
  \end{polycvpage}
  \begin{polycvletter}%
    [⟨Author Address⟩]%
    {⟨Subject⟩}%
    {⟨Recipient Address⟩}
    \opening{Dear Example,}
    %
    % Letter content
    %
    \closing{Sincerely,}
    \ps{more text}
  \end{polycvletter}
  \pdfappendix{⟨filename⟩}
\end{document}
```

3.1 Class Options

This class is build on top of the standard article class. It therefore inherits most options from that class. Some are incompatible with the design of this class and have been disabled. Other options allow for customisation of the used templates. In this section they will be briefly described, along with commands that can be used to set the values in the preamble.

3.1.1 Page Layout

The general page layout will be accomplished by the `geopetry` package. Depending on the environment currently in use, a few internal lengths are defined and used. Most of these can only be set via the class options, or within the preamble.

Opt `margins`
Default: `1.0 cm`

`\setmargins`

The length of `margins` is the baselength of most surrounding space, it defines and sets the length `\polycvmargins`. It is used additionally to the header and footer lengths. This setting also controls the space between the side bar and the main body on the first page of the CV environment. The command **`setmargins`**{⟨*length*⟩} is only available in the preable and can be used to set the margin length after loading the class.

Opt `headerheight`
Default: `3.0cm`
Opt `footerheight`
Default: `0.5cm`

`\setheaderheight`
`\setfooterheight`

The options `headerheight` and `footerheight` control the height of the boxes for the header and the footer, respectively. These lengths affect the whole document and can therfore only be set when loading the class or within the preamble. The will be the same for each of the environments used. Instead of choosing the key/value option at loadtime, the commands `\`**`setheaderheight`**{⟨*length*⟩} and `\`**`setfooterheight`**{⟨*length*⟩} can be issued to manipulate these values.

Opt `sidebarwidth`
Default: `5.0cm`

⚠
`\setsidebarwidth`

On the left side of first page of the CV a box is created, the side bar, which may hold personal information, a picture, skills, languages, hobbies, etc.. The width of this box is controlled with the option `sidebarwidth`. This must be defined before the sidebar is attempted to be set. There is currently no command analogous to the other lengths to set this in the preable, it must be defined via the options.

Opt `iconspace`
Default: `0.8cm`

`\seticonspace`

Closely related is the option `iconspace`, which defines the width that is reserved for the icon in a line in the side bar environment. It can also be set via `\`**`seticonspace`** {⟨*length*⟩} in the preamble.

Opt `hintcol`
Default: `2.7cm`
Opt `hintcolsep`
Default: `0.2cm`

`\sethintcolwidth`
`\sethintcolsep`

In the main body, each entry will consist of two parts: First, a small column for a keyword, date, bar, which is called hint in this documentation. The width of this column can be adjusted with the option `hintcol`. Second, a column for the description of this hint. This is just a parbox, and depends on the command used. The width will be automaticall calculated from the available space. These two columns are separated by a space. The width of which can be set via the option `hintcolsep`. Alternatively they can be set in the preamble via `\`**`sethintcolwidth`**{⟨*length*⟩} and via `\`**`sethintcolsep`**{⟨*length*⟩}, respectively.

Opt `letterindent`
Default: `1em`

In the letter environment the indentation of the first line of a paragraph can be adjusted with the `letterindent` option. This is another length that can (currently) only be controlled with passing the option to the class.

3.1.2 Colors

Opt `xcolor`
Default: `dvipsnames`

The class loads the `xcolor` package and passes by default the option `dvipsnames` to it. This can be changed by using the `xcolor` option to pass other parameters to this

package. It is loaded internally before the `tikz` package to avoid option conflicts.

Opt `primary`
Default: `blue`
Opt `shade`
Default: `black`
Opt `mixing`
Default: `10`
Opt `secondary`
Default: `white`
unused
Opt `highlight`
Default: `orange`
`\setcolprimary`
`\setcolsecondary`
`\setcolhighlight`
`\setcolshade`

The main colour of the CV is called `primary` and can be set via this option. It is used for all parts that will be highlighted, such as the header bar and the section headings. The textcolor is a composit color from the primary color and the `shade` color. By default this is defined as black, so the text will appear darker. The amount of primary color mixed in depends on the a factor which can be set via the `mixing` option.

The `secondary` color specifies the background color and is white by default.

One last color is defined to highlight items. It can be set via the `highlight` option. This option currently has little effect as the color itself is not used within the template.

To allow for custom colors to be used, all colors ca also be set with the `\setcol<...>` {⟨*color*⟩} commands. Note that the mixing parameter must be set via the options and has no special macro.

3.1.3 Progress and Level Bar Options

Opt `progbarheight`
Default: `1.5 ex`
`\setprogbarheight`
Opt `progbarunits`
Default: `5`
`\setprogbarunits`

Skills can be displayd as an CV entry and accompanied with a progress bar or a level (bar) indicator. The width of this bar is by default either determined by the environment in which it is used, or the width of the hint column. The height of the bar can be controlled with the `progbarheight` option. It can also be set via **`\setprogbarheight`** {⟨*length*⟩}. The level bar is devided into boxes or cicles. The number of these items can be set via `progbarunits` and defaults to five. It can also be set via **`\setprogbarunits`** {⟨*number*⟩}.

3.1.4 Miscellaneous Options

Opt `draft, final`

The class also extends the `draft` option of the standard article class. The otherwise invisible boxes, e.g. for the side bar, will be lightly shaded, so that they are visible. The complementary option is `final`. Note that if the final option is found, the draft option will have no effect and will be ignored.

Opt `biblatex`

The `biblatex` option will inject some code into a bibliography environemnt, so that the alingment of the hint column will be better. It is turned off by default. If the biblatex package is not loaded (this class will not take care of this), it might lead to problems.

Opt `signcv`

In some countries it is customary to sign the CV. The option `signcv` can be used to enable this behavior. If it is enabled, a box with a signature will be placed in the lower left corner of the first page, below the sidebar contents. The box width is also controlled with the sidebar width option. An image with the signature is necessary (see below).

3.1.5 Void Article Options

Opt `twocolumn` ⚠ Since every page has a specific layout passing the `twocolumn` option to the inherited article class will lead to problems. Therefore it has been disabled.

Opt `titlepage` ⚠ For the same reason as above, creating a title page doesn't make sense and will clash with other settings of this class. Therefore the `titlepage` option has also been disabled.

4 Changing Fonts and Text Alignment

`\setheaderfont` Default: `\sffamily\bfseries` By default the font used for the header is made bold and serifs are turned off. In order to change that layout, the command **\setheaderfont**{⟨*font specification*⟩} can be used. This command should be used in the preamble.

`\setentryragged` `\setentryjustified` The default layout of an entry in the CV is justified. This can be changed with **\setentryragged** to flush the contents to the left and leave the right side ragged (**\raggedright**). For symmetry the command **\setentryjustified** can be used to unset the aabove. Both commands will set the internal variable used in the templates at the end of the preamble, so they can only be used there.

`\setsigright` `\setsigleft` The commands **\setsigright** and **\setsigleft** determine whether the name below a signature field is right or left aligned. This does not affect where the field itself is placed. These commands can also only be used in the preamble.

4.1 Commands to Store and Access Information

Personal information can be stored in variables so that they become accessible to the defined templates.

`\author` `\insertauthor` The **\author**{⟨*author name*⟩} command remains unchanged and cannot be used outside of the preamble. The contents of that field is copied to the **\insertauthor** variable and is accessible throughout the document that way.

`\title` `\inserttitle` Similarly to the above, the **\title**{⟨*title*⟩} remains unchanged, with the same limitations applied. The contents can be accessed with the **\inserttitle** command. There is currently no template using this command, but it is recommended to set it to somethin like *Curriculum Vitae* for the PDF fields that might use it.

`\position` `\insertposition` The **\position**{⟨*position*⟩} macro is defined in a similar way to the above commands. The **\insertposition** macro can be used to access the content. It is used in the header box and can be set to contain, as the name suggest, the title of the current position. A common usage might also be a one line description of one's profession.

`\address` `\insertaddress` The address can either be specified by the direct invocation of **\address**{⟨*address*⟩}

\street
\location
\insertstreet
\insertlocation
⚠

and is accessible via the **\insertaddress** macro. It is used in the address template (see sectin 4.2.2), as well as the header of the letter (see section 4.4). Alternatively it the macros **\street**{⟨*street*⟩} and **\location**{⟨*location*⟩} can be used to set the values separately. The **\insertaddress** will then be contructed from these two values. If set, they can be printed with the corresponding **\insertstreet** and **\insertlocation** macros. If the address is set directly the other values remain unchanged. It is therefore also possible to extend the functionality in the follwoing way:

```
\street{123, Example Road}
\location{New Town}
\address{The Old Shack\\\insertstreet\\\insertlocation}
```

\email
\insertemail

The command **\email**{⟨*email*⟩} will wrap the input in a hyperlink, which can be accessed through the **\insertemail** command.

\phone
\mobile
\insertphone
\insertmobile

Telephone numbers can be stored with **\phone**{⟨*number*⟩} and **\mobile**{⟨*number*⟩} and accessed with the adjacent commands **\insertphone** and **\insertmobile**. Currently the template only provides templates for these two telephone numbers.

\github
\insertgithub

The macro **\github**{⟨*username*⟩} provides a convenient way to display the GitHub username, while also wrapping it into a hyperlink to the associated profile.

\orcid
\insertorcid

Likewise for an ORCID account, the **\orcid**{⟨*ID number*⟩} will wrap the displayed ID into a hyperlink to the respective profile.

\signaturefile
\insertsigfilename
⚠

In order to sign the CV and the cover letter, a file name must be provided with the macro **\signaturefile**{⟨*filename*⟩}. This file name is stored in **\insertsigfilename** and is passed to the command **includegraphics** in the corresponding templates. For the cover letter it is necessary to set this filename, otherwise it will (currently) lead to an error.

4.2 Templates

The templates that are provided with this class should make typesetting the contents of the resumee easier as they provide predefined spacing and markup.

4.2.1 General Templates for the CV

\section
\subsection

The **\section**{⟨*text*⟩} and **\subsection**{⟨*text*⟩} commands are redefined to have no numbers. For the former there will be a bar the length of the hint column, while for the latter there is jsut the same amount of space. Both commands will use the primary color and the predefined header fonts.

\cvline

The **\cvline**[⟨*width*⟩] {⟨*hint*⟩} {⟨*description*⟩} macro divides an entry line in the CV into two parts: a hint column with the reset hint, which can also be set with the optional

argument, and a description box spanning the remaining width of the line. It can be used to set any type of centent as a line in the CV.

\cvitemline The **\cvitemline**[⟨*width*⟩]{⟨*icon*⟩}{⟨*description*⟩} is similarly constructed as the above command. The general difference is that the hint coulmn is replaced with an icon column. If the optional ⟨*length*⟩ parameter is not given, it falls back to the definition from the preamle. The line itself is defined consisting of two parboxes, spanning the whole width that is available. This command is primarily intended for the sidebar, but it could be used somewhere else. It is also used for the specialised templates (see section 4.2.2).

\progressbar The **\progressbar**[⟨*width*⟩]{⟨*level fraction*⟩} typesets a progress bar. The height of this bar is defined via the class options or in the preamble. The width of the bar can be set via the optional parameter ⟨*length*⟩, however, in most use cases it will be automatically calculated depending on where it is used. The mandatory argument ⟨*level fraction*⟩ should be a number between 0 and 1. There are some special templates, which use this command (see section 4.2.2).

\levelbarsquares
\levelbarcircles The macros **\levelbarsquares**[⟨*width*⟩]{⟨*level*⟩} and **\levelbarcircles**[⟨*width*⟩]{⟨*level*⟩} are also progress indicators, which use squares or circles, respectively. The number of items is set via the class options or in the preamble. The mandatory argument ⟨*level*⟩ should be a number within 0 and the set limit. The width can be given as the optional argument ⟨*width*⟩ and defaults to the width of the hint column. There are currently no templates that use these commands.

4.2.2 Special Templates for the CV

\cventry The **\cventry** macro is a more specialised template of the **\cvline** macro. It requires six mandatory arguments, which of course can be empty. It is intended to be used in the following form:
\cventry[⟨*width*⟩]{⟨*when*⟩}{⟨*what*⟩}{⟨*who*⟩}{⟨*where*⟩}{⟨*opt.*⟩}{⟨*descr.*⟩}.
The entry spans the whole available width, the optional ⟨*width*⟩ controls the hint column. The first argument ⟨*when*⟩ contains the hint, which can be a date for example. The second argument ⟨*what*⟩ will be set in bold font and usually contains the title of the position, or a short job description. This is followed by a comma. The third argument ⟨*who*⟩ is set in italic font and usually contains the name of the employer, which is again followed by a comma. The fourth argument ⟨*where*⟩ may contain the location of this job and it is set in normal font style. If the fifth argument ⟨*opt.*⟩ contains something, it is set in the same style as the preceding element, separated by a comma from it. This could be used to specify a grade in the education section for example. If this argument is empty, the comma is skipped. The line will then be terminated by a full stop. The sixth argument ⟨*descr.*⟩ can also be empty. If it contains text, it will be set on a new line and can be used as a more detailed description of the entry. The font size is reduced to small for this paragraph.

\cvskill The **\cvskill**{⟨*description*⟩}{⟨*level*⟩} is a template using **\cvline**, defining a progress

bar to ⟨*level*⟩ and a ⟨*description*⟩. If it is used in the main document part, the progress bar will be set into the hint column, and will take half of its width and is flushed to the right. The description is flushed to the left in the main column. If it is used in the side bar, the description will start the line, followed by the progress bar. The progress bar will then use a third of the whole line, and the description the rest exclluding son separator space.

`\cvlanguage`

The `\cvlanguage{⟨language⟩}{⟨level⟩}` is a wrapper to the above command. This was introduced, to allow for the possibility of customisation. For example, it could be redefined to use a different indicator.

⚠ numeric format

```
\renewcommand\cvlanguage[2]{%
    \cvline{\levelbarsquares{#2}}{#1}}
```

`\cvlineaddress` `\cvlineemail` `\cvlinephone` `\cvlinemobile`

The special versions of the `\cvitemline` macro have a preset icon to it. The address (), email (), phone (), and mobile () templates are predefined. These macros are primarily intended for the side bar environment. They may be used in the main part, too, but they could look somewhat odd. Other lines can be added with the generic command.

```
\cvitemline{\faTerminal}{bash, zsh, fish}
```

`\orcidiconfilename`

Since there is no LaTeX package available that includes an Orcid icon, it must be provided with an external file. The command `\orcidiconfilename` will look for a file with the name `orcidicon.pdf` and include this as a graphic. If such a file is not found, it will use a palceholder icon made of the letters.

`\cvlineorcid` `\cvlinegithub`

The orcid and GitHub macros will use their respective icons, similar to the above. The contents of the variables defined in the preamble will go into the description column.

4.3 The CV Environments

Env `polycvfirstpage`

The class comes with two environments to set the CV contents. The first page is special since it also contains a side bar with additional infomation. The environment itself sets up the geometry of the page, and the header box including the name and position. It will allso set up the footer box to close the page. It should be used to set a single page only.

Env `polycvsidebar`

The side bar is part of the first page environment and should be issued inside it. It will produce a box with the specified content on the left side of the page. The main templates of this class will have different definintions inside this environment.

The following code block contains an exemplary usage of the two environments.

```
\begin{polycvfirstpage}
  \begin{polycvsidebar}
  \includegraphics[width=1.0\linewidth]{example-image-1x1}\\[2ex]

  \section{Personal Details}
  \cvitemline{\faStar}{01. 01. 1984,\newline Birthplace, Country}
  \cvlineaddress
```

```
  \cvlineemail

  \section{Languages}
  \cvlanguage{English (native)}{1}

  % etc.
  \end{polycvsidebar}%
  \section{Experience}
  \cventry{years}
    {jobtitle}{employer}{place}
    {optional: ...}
    {optional: job description}

  \section{Skills}
  \cvskill{\LaTeX}{0.8}

  % etc.
\end{polycvfirstpage}
```

Env `polycvpage` The generic CV page environment is similar to the firstpage environment, but it doesn't include space for the ssside bar. The margins aare adjusted to fill the whole available width. It is useful to typeset a second (or backside) page of the CV. The header box of this page will be of the same size as for the first page, but remain empty.

The following codeblock briefly summarises an exaamplle usage:

```
\begin{polycvpage}
  \section{Education}
  \cventry{years}
    {degree}{institution}{place}
    {optional: grade/...}
    {optional: comment/ description}

  % etc.
\end{polycvpage}
```

4.4 The Cover Letter Environment

Env `polycvletter` The letter environment takes one optional and two mandatory arguments: The optional argument is the author's address, which, if unspecified, will be constructed from variables set in the preamble. The first mandatory argument is the subject line, while the second is the recipient's address.

```
  \begin{polycvletter}
    [⟨Author Address⟩]{⟨Subject line⟩}{⟨Recipient Address⟩}
    %
    % Letter content
    %
  \end{polycvletter}
```

At the end of the environment the signature, a file name provided in the preample with `\signaturefile{<image file>}`, along with the author name will be set.

`\opening` `\closing` The letter environment uses indentation specified with the `letterindent` option (see section 3.1.1). This looks odd for the opening and closing statement, therefore a wrapper is provided that turns of the indentation. They each take one argument, like `\opening{`⟨*statement*⟩`}`. These commands will be set in place, i.e. you could include text before and after them.

`\ps` Occasionally you would like to inlcude a postscript note. The contents of such a note can be stored with `\ps{`⟨*note*⟩`}` and will be produced below the signature of the letter. The fontsize is also smaller than the one of the text. This command can only be used once; a later occurrence will overwrite a previous one.

`\setpsmark` How the postscript is introduced can be changed with `\setpsmark{`⟨*psmark*⟩`}`.
Default: `P.S.`

The following is a minimal working example for a letter created with the polycv class.

```
\documentclass[primary=black,14pt,a4paper]{polycv}
\usepackage{mwe} % for this example
\title{Curriculum Vitae}
\author{Example Author}
\signaturefile{example-image-16x9}

\begin{document}
\begin{polycvletter}
  [Address line 1\\ Address line 2]
  {Subject line}
  {Company\\ Address line 1\\ Address line 2}
  \opening{Dear Example,}

  \blindtext[1-3]

  \closing{Sincerely,}
  \ps{\blindtext[1]}
\end{polycvletter}
\end{document}
```

4.5 Including external PDF as Appendix

`\pdfappendix` It is occasionally necessary to include certificates, letters of recommendation, or other external sources into an application. This class provides a wrapper to the `\includepdf` command from `pdfpages`. The difference to the original command (which still could be used) is that before including the document, the wrapper unsets the page color. This is necesssaary, even if it is defined as white, due to the way the page color is implemented. You can include pages from a PDF document to use it as appendix with the command `\pdfappendix[`⟨*kwargs*⟩`]{`⟨*filename*⟩`}`, where the optional argument accepts arguments like `pages=1-3` or `angle=90`. See the Documentation of the `pdfpages` package.
Default: `pages={1-}`

5 Implementation

5.1 Option definitions

These simple definitions for key-value pairs as class options are done with kvoptions as described in TUGBoat.[1] Using the prefix **polycv@** as an inbetween to distinguish values set from class options in the document.

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{
  family=polycv,
  prefix=polycv@
}
```

Define options and preset margins for the page setup via the geometry package.

```
\DeclareStringOption[1.0cm]{margins}
\DeclareStringOption[3.0cm]{headerheight}
\DeclareStringOption[0.5cm]{footerheight}
\DeclareStringOption[5.0cm]{sidebarwidth}
\DeclareStringOption[2.7cm]{hintcol}
\DeclareStringOption[0.2cm]{hintcolsep}
\DeclareStringOption[0.8cm]{iconspace}
\DeclareStringOption[1.5ex]{progbarheight}
```

The progressbar, which contains of individual items, can be customised to include a certain number of elements. Define the option for this number and set it to five as default.

```
\DeclareStringOption[5]{progbarunits}
```

Define the options for the colors which should be used throughout the document. The mixing option is used together with the shading color, to modify the primary color.

```
\DeclareStringOption[blue]{primary}
\DeclareStringOption[white]{secondary}
\DeclareStringOption[orange]{highlight}
\DeclareStringOption[black]{shade}
\DeclareStringOption[10]{mixing}
```

Define an option for a draft mode, analogous to the standard classes. This will be extended, see below.

```
\DeclareBoolOption{draft}
\DeclareComplementaryOption{final}{draft}
```

[1] J. Wright, C. Feuersänger, TUGBoat, Vol. 30 (2009), No. 1, p. 110-122.

Define a biblatex mode, which will adjust some space settings in the bibliography, if desired.

```
\DeclareBoolOption{biblatex}
```

Define an option that lets the user pass settings to xcolor, i.e. (the default) `dvipsnames`.

```
\DeclareStringOption[dvipsnames]{xcolor}
```

Define the option whether the CV (front page) is signed or not.

```
\DeclareBoolOption{signcv}
```

Define an option to set the indentation in the letter and let it default to the width or one m.

```
\DeclareStringOption[1em]{letterindent}
```

Warn about options which will not and cannot be used or passed on, because of potential clashes for thiss class.

```
\DeclareVoidOption{twocolumn}{%
  \ClassError{polycv}{%
    Option 'twocolumn' is incompatible because of the special page setup.}}
\DeclareVoidOption{titlepage}{%
  \ClassWarning{polycv}{%
    Option 'titlepage' is incompatible because of the special page setup.}}
```

Pass all other options to the satndart article class.

```
\DeclareDefaultOption{%
  \PassOptionsToClass{\CurrentOptionKey}{article}}
```

Finally, process these options and load the standard class.

```
\ProcessKeyvalOptions{polycv}
\LoadClass{article}
```

5.2 Necessary packages

Some packages are essential for this class and need to be loaded.

- `kvoptions` is necessary for the use of key-value-options (see above)

- `etoolbox` is necessary for additional hooks like `\AtEndPreamble`.
- `calc` is necessary to do mathematics with lengths
- `hyperref` provides hyperlinks within the CV and letter
- `xcolor` provides convenient color definitions, since hyperref is loaded, this option is set by default. Other options can be added with the class options switch. This package must be loaded before tikz, otherwise there will be an option clash.
- `tikz` is used for headerbox, symbols, etc.. (must be loaded after xcolor)
- `graphicx` is used for the inclusion of graphics, e.g. signature and profile picture.
- `parskip` turns of indentation and improves spacing
- `geometry` is used for the page layout
- `fontawesome` provides fancy symbols for the sidebar
- `pdfpages` are used to include a pdf in the appendix

```
\RequirePackage{etoolbox}
\RequirePackage{calc}
\RequirePackage{hyperref}
\PassOptionsToPackage{\polycv@xcolor,hyperref}{xcolor}
\RequirePackage{xcolor}
\RequirePackage{tikz}
\RequirePackage{graphicx}
\RequirePackage[indent=0pt]{parskip}
\RequirePackage{geometry}
\RequirePackage{fontawesome}
\RequirePackage{pdfpages}
```

5.3 Global settings

Issue an informative statement that the CV is signed. (To Do.)

```
  \AtEndPreamble{%
    \ifpolycv@signcv%
      \ClassInfo{polycv}{%
        Option 'signcv' enabled; CV will be signed.}%
    \fi
    }%
```

Set the page style to remove pagenumbers.

```
\pagestyle{empty}
```

5.3.1 Colors

Apart from the switches as options, define some commands that can be used to set the colors for the document.

`\setcolprimary` The primary color used for headers.

```
\newcommand*{\setcolprimary}[1]{%
  \renewcommand*{\polycv@primary}{#1}}
```

`\setcolsecondary` The secondary color is the background color, i.e. the page color.

```
\newcommand*{\setcolsecondary}[1]%
  {\renewcommand*{\polycv@secondary}{#1}}
```

`\setcolhighlight` If text should be highlighted another color can be set.

```
\newcommand*{\setcolhighlight}[1]%
  {\renewcommand*{\polycv@highlight}{#1}}
```

`\setcolshade` The color which is used for shading (damping) the primary color.

```
\newcommand*{\setcolshade}[1]%
  {\renewcommand*{\polycv@shade}{#1}}
```

Disable these changing color commands for body, then define the colors in internal macros along with some mixing. Then set the text color and the color for the page for the entire document.

```
\@onlypreamble\setcolprimary
\@onlypreamble\setcolsecondary
\@onlypreamble\setcolhighlight
\@onlypreamble\setcolshade
\AtEndPreamble{%
  \colorlet{polycvpagecolor}{\polycv@secondary}
  \colorlet{polycvheaderbackground}{\polycv@primary}
  \colorlet{polycvheadertext}{\polycv@secondary}
  \colorlet{polycvsectioncolor}{\polycv@primary}
  \colorlet{polycvtextcolor}{%
    \polycv@primary!\polycv@mixing!\polycv@shade}
  \colorlet{polycvhighlight}{\polycv@highlight}
  \colorlet{polycvsidebarbackground}{\polycv@secondary}
  \colorlet{polycvprogbarfilled}{\polycv@primary}
  \colorlet{polycvprogbarnofill}{%
    \polycv@primary!10!\polycv@secondary}
```

```
}
\AfterPreamble{%
  \color{polycvtextcolor}\pagecolor{polycvpagecolor}}
```

5.3.2 Extending draft mode

This section emulates and extends the draft mode. It writes an information message to the log. It then sets the overfull boxes identifier.[2] Passes the `showframe` option to the geometry package to make the page borders visible. It also sets the color of the side bar, to make the tikz boxes visible in draft mode.

```
\ifpolycv@draft%
  \ClassInfo{polycv}{Option 'draft' is enabled}
  \setlength\overfullrule{1em}
  \PassOptionsToPackage{showframe}{geometry}
  \AtEndPreamble{%
    \colorlet{polycvsidebarbackground}{%
      \polycv@primary!20!\polycv@secondary}%
  }
\fi
```

5.3.3 Biblatex Mode

In this option the biblatex package is set to use the same width as the date column for the numeration.

```
\ifpolycv@biblatex%
  \ClassInfo{polycv}{Option 'biblatex' is enabled}
  \AtEndPreamble{%
    \setlength{\biblabelsep}{\polycvhintcolsep}%
    \DeclareFieldFormat{labelnumberwidth}{%
      \makebox[\polycvhintcol][r]{[#1]}}%
  }%
\fi
```

5.3.4 Lengths

Define the lengths and use defaults or parameters from the class options for the layout.

```
\newlength{\polycvheaderheight}
\setlength{\polycvheaderheight}{\polycv@headerheight}
\newlength{\polycvfooterheight}
```

[2] *How to test whether report has been called with "draft" option* https://tex.stackexchange.com/a/240128/33413

```
\setlength{\polycvfooterheight}{\polycv@footerheight}
\newlength{\polycvmargins}
\setlength{\polycvmargins}{\polycv@margins}
\newlength{\polycvsidebarwidth}
\setlength{\polycvsidebarwidth}{\polycv@sidebarwidth}
\newlength{\polycvhintcol}
\setlength{\polycvhintcol}{\polycv@hintcol}
\newlength{\polycvhintcolsep}
\setlength{\polycvhintcolsep}{\polycv@hintcolsep}
\newlength{\polycviconspace}
\setlength{\polycviconspace}{\polycv@iconspace}
\newlength{\polycvprogbarheight}
\setlength{\polycvprogbarheight}{\polycv@progbarheight}
\newlength{\polycvletterindent}
\setlength{\polycvletterindent}{\polycv@letterindent}
```

The lengths can also be altered with the following commands.

`\setheaderheight` Define the hight for the header box.

```
\newcommand*{\setheaderheight}[1]{%
  \setlength{\polycvheaderheight}{#1}}
```

`\setfooterheight` Define the height for the footer box.

```
\newcommand*{\setfooterheight}[1]{%
  \setlength{\polycvfooterheight}{#1}}
```

`\setmargins` Define the width of the margins.

```
\newcommand*{\setmargins}[1]{%
  \setlength{\polycvmargins}{#1}}
```

`\sethintcolwidth` Define the hint column width (where dates and level, etc. are set).

```
\newcommand*{\sethintcolwidth}[1]{%
  \setlength{\polycvhintcol}{#1}}
```

`\sethintcolsep` Define the space between the hint column and the description column.

```
\newcommand*{\sethintcolsep}[1]{%
  \setlength{\polycvhintcolsep}{#1}}
```

`\seticonspace` Define the space reserved for the items in the sidebar box.

```
\newcommand*{\seticonspace}[1]{%
  \setlength{\polycviconspace}{#1}}
```

\setprogbarheight Define the standard height of the progress bar. This will also apply to the levels bar.

```
\newcommand*{\setprogbarheight}[1]{%
  \setlength{\polycvprogbarheight}{#1}}
```

These commands need to be disabled for the body, so that automatically derived lengths will still be useable.

```
\@onlypreamble\setheaderheight
\@onlypreamble\setfooterheight
\@onlypreamble\setmargins
\@onlypreamble\sethintcolwidth
\@onlypreamble\sethintcolsep
\@onlypreamble\seticonspace
\@onlypreamble\setprogbarheight
```

Additional lengths, which are derived from the above, will be set at the end of the preamble.

```
\newlength{\polycvleftmargin}
\newlength{\polycvtopmargin}
\newlength{\polycvbottommargin}
\AtEndPreamble{%
  \setlength{\polycvleftmargin}{%
    \polycvsidebarwidth+2\polycvmargins}
  \setlength{\polycvtopmargin}{%
    \polycvheaderheight+\polycvmargins}
  \setlength{\polycvbottommargin}{%
    \polycvfooterheight+\polycvmargins}
}
```

5.3.5 Fonts and Styles

\polycv@headerfont \setheaderfont Define the font family for the header and make it bold (by default) and store it in a variable. Provide access to this cariable via a set command.

```
\providecommand\polycv@headerfont{}
\renewcommand{\polycv@headerfont}{\sffamily\bfseries}
\providecommand\setheaderfont{}
\renewcommand*{\setheaderfont}[1]{%
  \renewcommand*{\polycv@headerfont}{#1}}
\@onlypreamble\setheaderfont
```

\polycv@entry@ragged \setentryragged \setentryjustified Provide a variable to store how the output is set, i.e. the standard raggedness. Provide commands to change the raggedness of the output. These need to be set in the preamble, so they are disabled for the body.

```
\providecommand\polycv@entry@ragged{}
\newcommand*{\setentryragged}{%
  \renewcommand{\polycv@entry@ragged}{\raggedright}}
\newcommand*{\setentryjustified}{%
  \renewcommand{\polycv@entry@ragged}{}}
\@onlypreamble\setentryragged
\@onlypreamble\setentryjustified
```

`\polycv@sig@align` `\setsigright` `\setsigleft`

Define the alignment for the signature, which is stored in a variable. Provide commands to change it, and disable them for the body.

```
\providecommand\polycv@sig@align{right}
\newcommand*{\setsigright}{%
  \renewcommand{\polycv@sig@align}{right}}
\newcommand*{\setsigleft}{%
  \renewcommand{\polycv@sig@align}{left}}
\@onlypreamble\setsigright
\@onlypreamble\setsigleft
```

5.4 Macros and Environments

Define special commands and environment.s

`\setprogbarunits`

Control the number of items for the levels bar.

```
\newcommand*{\setprogbarunits}[1]{%
  \renewcommand*{\polycv@progbarunits}{#1}}
\@onlypreamble\setprogbarunits
```

5.4.1 Personal Information

This section defines the variables, which will store the personal information. These are used in the template lines of the sidebar.

`\insertauthor` `\inserttitle`

The author and title commands already exist, so they only need to be stored.

```
\providecommand*\insertauthor{}
\renewcommand{\insertauthor}{\@author}
\providecommand\inserttitle{}
\renewcommand{\inserttitle}{\@title}
```

`\insertposition` `\position`

Store the position, which is used as the sub-heading in the title banner. Disable this command because it is needed before the header is set.

```
\providecommand*\insertposition{}
\providecommand*\position{}
\renewcommand{\position}[1]{%
  \renewcommand{\insertposition}{#1}}
\@onlypreamble\position
```

`\cvitemline` This defines a generic line consisting of two parboxes. This command is primarily needed for the sidebar, but it could be used somewhere else. The left parbox is used as a hint column and contains an icon, the right the value of a variable.

```
\providecommand\cvitemline[3][\polycviconspace]{%
  \parbox[t]{#1}{\centering#2}%
  \parbox[t]{\linewidth-#1}{#3}\par\vspace{0.3\baselineskip}}
```

`\insertaddress` `\insertstreet` `\insertlocation` Variables to store an address.

```
\providecommand\insertaddress{}
\providecommand\insertstreet{}
\providecommand\insertlocation{}
```

`\address` `\street` `\location` Commands to set the values used for the address

```
\providecommand\address{}
\providecommand\street{}
\providecommand\location{}
\renewcommand{\address}[1]{%
  \renewcommand{\insertaddress}{#1}}
\renewcommand{\street}[1]{%
  \renewcommand{\insertstreet}{#1}}
\renewcommand{\location}[1]{%
  \renewcommand{\insertlocation}{#1}}
```

Build the address for street and location.

```
\renewcommand{\insertaddress}{\insertstreet\\\insertlocation}
```

`\cvlineaddress` Template to produce the address for the sidebar.

```
\providecommand{\cvlineaddress}{%
  \cvitemline{\faMapMarker}{\insertaddress}}
```

`\insertemail` `\email` `\cvlineemail` Store and set email, template for the sidebar.

```
\providecommand*\insertemail{}
\providecommand*\email{}
\renewcommand{\email}[1]{%
```

```
  \renewcommand{\insertemail}{\href{mailto:#1}{#1}}}
\providecommand{\cvlineemail}{%
  \cvitemline{\faEnvelope}{\insertemail}}
```

`\insertphone` `\phone` `\cvlinephone` Store and set phone number, template for the sidebar.

```
\providecommand*\insertphone{}
\providecommand*\phone{}
\renewcommand{\phone}[1]{\renewcommand{\insertphone}{#1}}
\providecommand{\cvlinephone}{%
  \cvitemline{\faPhone}{\insertphone}}
```

`\insertmobile` `\mobile` `\cvlinemobile` Store and set mobile number, template for the sidebar.

```
\providecommand*\insertmobile{}
\providecommand*\mobile{}
\renewcommand{\mobile}[1]{\renewcommand{\insertmobile}{#1}}
\providecommand{\cvlinemobile}{%
  \cvitemline{\faMobile}{\insertmobile}}
```

`\insertgithub` `\github` `\cvlinegithub` Store and set username for GitHub, template for the sidebar.

```
\providecommand*\insertgithub{}
\providecommand*\github{}
\renewcommand{\github}[1]{\renewcommand{\insertgithub}{%
  \href{https://github.com/#1}{github.com/#1}}}
\providecommand{\cvlinegithub}{%
  \cvitemline{\faGithub}{\insertgithub}}
```

`\orcidiconfilename` This defines the orcid icon either from the file `orcidicon.pdf` if it is found by LaTeX. I am unaware of a LaTeXpackage that defines this symbol.[3] If such a file is not found, it will place as small square of bold letters spelling orcid.

```
\providecommand*\orcidiconfilename{}
\IfFileExists{orcidicon.pdf}{%
  \renewcommand{\orcidiconfilename}{%
    \raisebox{-0.2em}{%
      \includegraphics[height=1.0em]{orcidicon.pdf}}}%
}{%
  \renewcommand{\orcidiconfilename}{%
    \raisebox{-0.2em}{%
      \resizebox{1em}{1em}{\textbf{ORCID}}}}%
}
```

[3] For more details see: *Is there a standard way to include ORCID in TeX / PDF?* (https://tex.stackexchange.com/q/275578/33413)

\insertorcid
\orcid
\cvlineorcid

Store and set the orcid number, template for the sidebar.

```
\providecommand*\insertorcid{}
\providecommand*\orcid{}
\renewcommand{\orcid}[1]{\renewcommand{\insertorcid}{%
  \href{http://orcid.org/#1}{orcid.org/#1}}}
\providecommand{\cvlineorcid}{%
  \cvitemline{\orcidiconfilename}{\insertorcid}}
```

\insertsigfilename
\signaturefile

Store the filename of an image used to produce the signature.

```
\providecommand\insertsigfilename{}
\providecommand\signaturefile{}
\renewcommand{\signaturefile}[1]{%
  \renewcommand{\insertsigfilename}{#1}}
```

5.4.2 Redefinition of Section Headers

\section
\subsection

The section and subsection commands are redefined to have no numbers. Instead there will be a bar the length of the hint column, and they will use the primary color.

```
\renewcommand{\section}[1]{%
  \ignorespaces%
  \parbox[b]{1\linewidth}{%
    \color{polycvsectioncolor}%
    \Large\polycv@headerfont%
    \rule{\polycvhintcol}{1ex}\hspace{\polycvhintcolsep}%
    \parbox{1.0\linewidth-\polycvhintcol-\polycvhintcolsep}{%
      #1%
    }%
  }\vspace{0.5\baselineskip}}
\renewcommand{\subsection}[1]{%
  \parbox[b]{1\linewidth}{%
    \color{polycvsectioncolor}%
    \polycv@headerfont%
    \hspace{\polycvhintcol}\hspace{\polycvhintcolsep}%
    \parbox{1.0\linewidth-\polycvhintcol-\polycvhintcolsep}{%
      #1%
    }%
  }\vspace{0.3\baselineskip}}
```

5.4.3 Header and Footer Boxes

\polycv@header

The header box will be used internally only. It is a simple box produced with tikz, and uses the author and position variables. It is typeset in the primary color of the template.

```
\newcommand{\polycv@header}[3][\polycvheaderheight]{%
  \begin{tikzpicture}[remember picture,overlay]%
    \node [rectangle, %
           fill           = polycvheaderbackground, %
           anchor         = north, %
           minimum width  = \paperwidth, %
           minimum height = #1%
          ] (headerbox) at (current page.north){};%
    \node [anchor = mid] (cv-name) at (headerbox) {%
      \Huge\polycv@headerfont\color{polycvheadertext}\textbf{%
        #2
      }%
    };%
    \node [anchor = north] at (cv-name.south) {%
      \Large\polycv@headerfont\color{polycvheadertext}%
      #3%
    };%
  \end{tikzpicture}%
  \ignorespaces%
}
```

`\polycv@footer` The footer box will be used internally only. It is a simple box produced with tikz, which can be used to ddisplay page numbers. If the argument to the macro is empty, nothing will be printed, otherwise dashes are used to surround the command. It is typeset in the primary color of the template.

```
\newcommand{\polycv@footer}[2][\polycvfooterheight]{%
  \begin{tikzpicture}[remember picture,overlay]%
    \node [rectangle, %
           fill           = polycvheaderbackground, %
           anchor         = south, %
           minimum width  = \paperwidth, %
           minimum height = #1%
          ] (footerbox) at (current page.south){};%
    \node [anchor = center] (pagenumber) at (footerbox) {%
      \polycv@headerfont\color{polycvheadertext}%
      \if\relax\detokenize{#2}\relax%
      \else%
      - #2 -%
      \fi%
    };%
  \end{tikzpicture}%
  \ignorespaces%
}
```

`\polycv@signature` The signature box will be used internally only. It is a simple box produced with tikz, which wraps around an image defined earlier. It will also print the author name below

the image. By default the box has the same background color as the page.

```
\newcommand{\polycv@signature}[2][\polycvsidebarwidth]{%
  \begin{tikzpicture}[remember picture, overlay]%
    \node [rectangle, %
           anchor = south west, %
           text width = #1, %
           align = \polycv@sig@align, %
           minimum height = 0.75#1, %
           yshift = \polycvbottommargin, %
           xshift = \polycvmargins, %
           fill = polycvsidebarbackground %
          ] (cv-signbox) at (current page.south west) {%
      \includegraphics[width=#1]{#2}\\%
      \insertauthor%
    };
  \end{tikzpicture}%
  \ignorespaces%
}
```

5.4.4 Level and Progress Bar

First a few simple filled and unfilled symbols are defined as tikz pictures.

`\polycv@square@nofill` `\polycv@square@filled` Square symbols, the height and width are set to the height of the progressbar, which is defined via the options.

```
\newcommand{\polycv@square@nofill}[1][\polycvprogbarheight]{%
  \begin{tikzpicture}[x=#1, y=#1]%
    \filldraw[polycvprogbarnofill] (0,0) rectangle (1,1);%
    \draw[polycvtextcolor, thick] (0,0) rectangle (1,1);%
  \end{tikzpicture}}
\newcommand{\polycv@square@filled}[1][\polycvprogbarheight]{%
  \begin{tikzpicture}[x=#1, y=#1]%
    \filldraw[polycvprogbarfilled] (0,0) rectangle (1,1);%
    \draw[polycvtextcolor, thick] (0,0) rectangle (1,1);%
  \end{tikzpicture}}
```

`\polycv@circle@nofill` `\polycv@circle@filled` Analogous circle symbols.

```
\newcommand{\polycv@circle@nofill}[1][\polycvprogbarheight]{%
  \begin{tikzpicture}[x=#1, y=#1]%
    \filldraw[polycvprogbarnofill] (0.5,0.5) circle [radius=0.5];%
    \draw[polycvtextcolor, thick] (0.5,0.5) circle [radius=0.5];%
  \end{tikzpicture}}
\newcommand{\polycv@circle@filled}[1][\polycvprogbarheight]{%
  \begin{tikzpicture}[x=#1, y=#1]%
```

```
    \filldraw[polycvprogbarfilled] (0.5,0.5) circle [radius=0.5];%
    \draw[polycvtextcolor, thick] (0.5,0.5) circle [radius=0.5];%
  \end{tikzpicture}}
```

\polycv@repeating This is a helper function to repeat typesetting a symbol. It will fill the spaces between the symbols.

```
\newcommand{\polycv@repeating}[2][5]{%
  \ifnum#1 > 0%
    \newcount\rep%
    \rep0%
    \loop\ifnum\rep < \number\numexpr#1-1\relax%
      #2\hfill%
      \advance\rep by 1%
    \repeat%
    #2%
  \fi%
}
```

\polycv@level@bar The level bar is also a helper function, which wraps around the repeating function. It sets its contents as a parbox with the width of the hint column (#1 by default). It also needs a level to display (#2), which is the number of filled elements (#3). It the computes from the options of total units the number of unfilled elements(#4) to typeset.

```
\newcommand{\polycv@level@bar}[4][\polycvhintcol]{%
  \newcount\total@units%
  \total@units\polycv@progbarunits%
  \newcount\level%
  \ifnum#2 < \total@units \level#2 \else \level\total@units \fi%
  \newcount\elevel%
  \elevel\numexpr\total@units-\level\relax%
  \def\symbol@filled{#3}%
  \def\symbol@nofill{#4}%
  \parbox{#1}{%
    \ifnum\level = 0%
      \polycv@repeating[\total@units]{\symbol@nofill}%
    \else
      \ifnum\level = \total@units%
        \polycv@repeating[\total@units]{\symbol@filled}%
      \else
        \polycv@repeating[\level]{\symbol@filled}%
        \hfill%
        \polycv@repeating[\elevel]{\symbol@nofill}%
      \fi%
    \fi%
  }%
}
```

\levelbarsquares
\levelbarcircles

Wrapper commands as user interfaces to the level bar.

```
\newcommand{\levelbarsquares}[2][\polycvhintcol]{%
  \polycv@level@bar[#1]{#2}{%
    \polycv@square@filled}{\polycv@square@nofill}%
}
\newcommand{\levelbarcircles}[2][\polycvhintcol]{%
  \polycv@level@bar[#1]{#2}{%
    \polycv@circle@filled}{\polycv@circle@nofill}%
}
```

\progressbar

The progressbar is a simple tikz image consiting of three rectangles: one denoting the level using the fill color, one using the empty color, and an empty box encompasing all as a frame.

```
\newcommand{\progressbar}[2][\polycvhintcol]{%
  \hfill%
  \begin{tikzpicture}[x=0.99*#1, y=\polycvprogbarheight]%
    \filldraw[polycvprogbarfilled] (0, 0) rectangle (#2, 1) ;%
    \filldraw[polycvprogbarnofill] (#2, 0) rectangle (1, 1);%
    \draw[polycvtextcolor, thick] (0, 0) rectangle (1, 1) ;%
  \end{tikzpicture}%
}%
```

5.4.5 Commands for the main Body

\cvline

A generic command to typeset a line in the CV. It consists of two parboxes, the left being the hint column, and the right the description. One example usage is: `\cvline[width]{hint}{description}`.

```
\providecommand\cvline{}
\renewcommand{\cvline}[3][\polycvhintcol]{%
  \par%
  \parbox[t]{#1}{\strut\raggedleft #2}%
  \hspace{\polycvhintcolsep}%
  \parbox[t]{\linewidth-\polycvhintcolsep-#1}{%
    \strut\polycv@entry@ragged #3}%
  \par\vspace{0.2\baselineskip}%
}
```

\cventry

The entry defines a block. It uses the command above, but fills in more details in the description parbox of the template. An example usage is: `\cventry[width]{when}{what}{who}{where}{optional}{optional}`.

```
\providecommand\cventry{}
\renewcommand{\cventry}[7][\polycvhintcol]{%
```

```
  \cvline[#1]{#2}{%
    \textbf{#3}, \textit{#4}, #5%
    \if\relax\detokenize{#6}\relax%
    \else%
    , #6%
    \fi%
    \if\relax\detokenize{#7}\relax%
    .%
    \else%
    .\newline\small #7%
    \fi%
  }%
}
```

`\cvskill` This command uses a progressbar in the hint column.

```
\providecommand\cvskill[2]{%
  \cvline[\polycvhintcol]{%
    \progressbar[0.5\linewidth]{#2}}{#1}}
```

`\cvlanguage` This is basically a wrapper to the above command.

```
\providecommand\cvlanguage[2]{%
  \cvskill{#1}{#2}}
```

Env `polycvfirstpage` This environment redefines the page layout for the first page, and calls the header and footer templates.

```
\newenvironment{polycvfirstpage}{%
  \newgeometry{%
    left=\polycvleftmargin,
    top=\polycvtopmargin,
    right=\polycvmargins,
    bottom=\polycvbottommargin,
    nohead,nofoot}%
  \polycv@header{\insertauthor}{\insertposition}%
  \polycv@footer{}%
}{%
  \restoregeometry%
  \ignorespaces%
}
```

Env `polycvpage` This environment defines the page layout for a 'regular' page. It may cause problems, when the content is overflowing, i.e. producing more than one page. Unfortunately, page handling has (still) to be done manually. It also calls the header and footer templates, but they are empty by default.

```
\newenvironment{polycvpage}{%
  \newgeometry{%
    top=\polycvtopmargin,%
    left=2\polycvmargins,%
    right=2\polycvmargins,%
    bottom=\polycvbottommargin,%
    nohead,nofoot}%
  \polycv@header{}{}%
  \polycv@footer{}%
}{%
  \restoregeometry%
  \ignorespaces%
}
```

5.4.6 Side Bar Definitions

`\polycv@sidebar@skill` This template is used in the sidebar. It consists of one parbox spanning the full width of the line, encapsulating two more parboxes to set the progressbar and the description.

```
\providecommand\polycv@sidebar@skill[3][\polycvhintcolsep]{%
  \parbox{1.0\linewidth}{%
    \hspace{#1}%
    \parbox[b]{2\linewidth/3-4#1}{#2}%
    \hspace{#1}\hfill%
    \parbox[b]{\linewidth/3}{%
      \progressbar[\linewidth]{#3}%
      }%
    \vspace{0.2\baselineskip}}%
}
```

Env `polycvsidebar` This environment provides a box as a tikz picture for the sidebar. It first saves the original definition of the section command, then redefines it for a better fit. It also redefines the skill and language commands for a better fit within the box. At the end, it restores the original definitions. If the option to sign the CV is active, another box will be inserted at the bottom off the paaage.

```
\newenvironment{polycvsidebar}{%
  \let\section\save@section%
  \newcommand{\section}[1]{%
  \parbox[b]{1\linewidth}{%
    \color{polycvsectioncolor}\large\polycv@headerfont{##1}%
  }\par\vspace{0.5\baselineskip}}%
  \let\cvlanguage\save@polycvlanguage%
  \newcommand{\cvlanguage}[2]{%
    \polycv@sidebar@skill{##1}{##2}}%
  \let\cvskill\save@polycvskill%
  \newcommand{\cvskill}[2]{%
```

```
    \polycv@sidebar@skill{##1}{##2}}%
  \begin{tikzpicture}[remember picture, overlay]%
    \node [anchor = north west, %
           text width = \polycvsidebarwidth, %
           xshift = \polycvmargins, %
           yshift = -\polycvheaderheight-\polycvmargins, %
           fill = polycvsidebarbackground %
          ] (polycvsidebarbox) at (current page.north west) %
    \bgroup%
}{%
    \egroup;%
  \end{tikzpicture}%
  \let\save@section\section%
  \let\save@polycvlanguage\cvlanguage%
  \let\save@polycvlanguage\cvskill%
  \ifpolycv@signcv%
    \polycv@signature{\insertsigfilename}%
  \fi%
  \ignorespaces%
}%
```

5.5 Cover Letter Definitions

In this section the commands for the letter template are set up.

\polycv@letter@from

The header box is now substituted to include the address. The colourscheme is also adjusted to make the box lesss bulky. Therefore the usual background colour and the text color will be used.

```
\newcommand{\polycv@letter@from}[3][\polycvheaderheight]{%
  \begin{tikzpicture}[remember picture, overlay]%
    \node [rectangle, %
           fill           = polycvheadertext, %
           anchor         = north, %
           minimum width  = \paperwidth, %
           minimum height = #1-1ex%
          ] (headerbox) at (current page.north){};
    \node [rectangle, %
           fill           = polycvheaderbackground, %
           anchor         = north, %
           minimum width  = \paperwidth, %
           minimum height = 1ex%
          ] (headerline) at (headerbox.south){};
    \node [anchor = east, %
           text width = 5cm, %
           xshift     = -2\polycvmargins
          ] (polycv-letter-address) at (headerbox.east) {%
```

```
      \polycv@headerfont\color{polycvheaderbackground}%
      \parbox[t]{1.0\linewidth}{%
        \raggedleft%
        \textbf{\Large #2}\\%
        \textit{\normalsize #3}%
        }%
    };%
  \end{tikzpicture}%
  \ignorespaces%
}
```

`\polycv@letter@to` This defines the field of the addressee.

```
\newcommand{\polycv@letter@to}[1]{%
  \noindent\parbox[t][4cm][c]{0.4\linewidth}{\raggedright#1}\par%
  \vspace{1\baselineskip}%
}
```

`\polycv@letter@date` This defines and formates the date field.

```
\newcommand{\polycv@letter@date}{%
  \noindent\parbox[t]{1.00\linewidth}{\raggedleft\today}\par%
}
```

`\polycv@letter@subject` This defines and formates the subject field.

```
\newcommand{\polycv@letter@subject}[1]{%
  \noindent\parbox[t]{1.00\linewidth}{%
    \raggedright\textbf{#1}}%
}
```

`\opening` `\closing` This defines and formates the opening and closing remarks

```
\providecommand\opening{}
\renewcommand{\opening}[1]{\noindent{}#1\par}
\providecommand\closing{}
\renewcommand{\closing}[1]{\noindent{}#1\par}
```

`\polycv@letter@psmark` Define the default postscript mark.

```
\newcommand{\polycv@letter@psmark}{P.S.~}
```

`\setpsmark` Command to change the default postscript mark.

```
\providecommand\setpsmark{}
```

```
\renewcommand*{\setpsmark}[1]{%
  \renewcommand{\polycv@letter@psmark}{#1}}
```

`\polycv@letter@ps` `\ps` Variable to store the content of the postscript remark and command to set it.

```
\providecommand\polycv@letter@ps{}%
\providecommand\ps{}
\renewcommand{\ps}[1]{%
  \renewcommand{\polycv@letter@ps}{%
    \noindent\footnotesize\polycv@letter@psmark#1\par%
  }%
}%
```

Env `polycvletter` Here the actual letter environment is defined. Like before, this is intended to be only one page and hence may cause problems for sufficiently long letters. The environment requires two parameters, the recipient address, and the body. First the page setup is changed, then the templates for header, recipient address, subject, and footer are called. The indentation is set to the specified value. At the end, the signature will be inserted, and below that - if present - the postscript remark.

```
\newenvironment{polycvletter}[3][\insertaddress]{%
  \newgeometry{%
    top=\polycvtopmargin,%
    left=2\polycvmargins,%
    right=2\polycvmargins,%
    bottom=\polycvbottommargin,%
    nohead,nofoot}%
  \polycv@letter@from{\insertauthor}{#1}%
  \polycv@letter@to{#3}%
  \polycv@letter@date%
  \polycv@letter@subject{#2}%
  \setlength{\parindent}{\polycvletterindent}%
}{%
  \vspace{0.5\baselineskip}%
  \noindent\includegraphics[width=5cm]{\insertsigfilename}\\%
  \noindent\insertauthor\par%
  \if\relax\detokenize{\polycv@letter@ps}\relax%
  \else%
  \polycv@letter@ps\par%
  \fi%
  \polycv@footer{}%
  \restoregeometry%
}
```

5.6 Appendix

\pdfappendix The appendix command is an interface to include (parts of) a pdf, e.g. for certificates or recommendations. The page colour is first reset to having none, otherwise it would overlay on top off the included pdf. By default all pages will be included, but this can be overwritten to specific range (and other options). The command can be used multiple times.

```
\providecommand*\pdfappendix{}
\renewcommand*{\pdfappendix}[2][pages={1-}]{%
  \nopagecolor%
  \includepdf[#1]{#2}}%
```

6 Building polycv and its Documentation

To compile the polycv package:

1. `pdflatex` **`polycv`**`.ins`

To compile the polycv documentation

1. `pdflatex` **`polycv`**`.dtx`
2. `makeindex -s gglo.ist -o` **`polycv`**`.gls` **`polycv`**`.glo`
3. `makeindex -s gind.ist -o` **`polycv`**`.ind` **`polycv`**`.idx`
4. `pdflatex` **`polycv`**`.dtx` (several times)

Change History

v1.0
General: Initial version 1

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

L

M

O

P

R

S

T

X