Abstract

This package is made for typesetting recipes. You simply enter ingredients, preparation steps and get a beautiful recipe page. It is also possible to enter two pictures, which are displayed above the name of the recipe. Another option is to write an introduction, several suggestions and a hint, which is displayed at the bottom of the page.

Change History

v1.0
General: Initial version

v1.1
General: Load the non-free package 'emerald' per option. Possibility to change the picture width independently from the text columns

v1.2
General: Better way for displaying the ingredients (tabularx). Thanks to Bartosz Dziubaczyk for developing

v1.3
General: Fixed a problem with the default recipe name font. Thanks to Václav Zeman for reporting. New environment for the ingredients. Thanks to Andreas Pöge for this suggestion

v1.4
General: Added two commands for an introduction and recipe suggestions. Additional several hooks are implement in order to insert user-defined text blocks at different places in the recipe. Thanks to Heikki Lehvaslaiho for designing the graphical separator and sending suggestions. Added translations for German, English, French and Spanish

v1.5
General: Adding translations for Portuguese and Brazilian. Thanks to Thiago de Melo for submitting. Adding optional argument for the ingredients part. Thanks to Benjamin Steinwender for submitting

---

*This document corresponds to xcookybooky v1.5, dated 2015/03/03.
Contents

1 Introduction ........................................... 4

2 Usage ............................................... 4
   2.1 Main ........................................... 4
   2.2 Advanced ....................................... 6
   2.3 Support Commands ............................... 7
   2.4 Modify Design ................................... 7
   2.5 Translation ...................................... 9
   2.6 Options ......................................... 10
   2.7 Background graphic .............................. 10

3 Example ............................................. 11

4 Troubleshooting ...................................... 12
   4.1 MiKTeX ......................................... 12
   4.2 TeX Live ........................................ 12

5 Bugs .................................................. 12
   5.1 Known Issues ................................... 12

6 Implementation ...................................... 13
   6.1 Options ......................................... 13
   6.2 Required Packages ............................... 13
   6.3 Page Layout ..................................... 14
   6.4 Color Management ............................... 14
   6.5 Size Management ................................. 16
   6.6 Recipe Font ..................................... 17
   6.7 Headlines and Translations ..................... 17
   6.8 Recipe Lengths .................................. 20
   6.9 Recipe Commands ............................... 21
      Main Commands ................................... 21
      Supporting Commands ............................ 23
CONTENTS

Internal Commands .................................................. 24

6.10 Hooks ............................................................ 26

6.11 Recipe Environment ............................................. 28

6.12 Header and Footer ............................................... 30

6.13 Background Picture ............................................ 31

6.14 Configuration file ............................................... 33
1 Introduction

It all begin in 2011 when I wanted to make a cookbook with \LaTeX. Thus I was looking for recipe templates and found the cookybooky package by Jürgen Gilg (http://www.ctan.org/pkg/cookybooky). It looks very good, but I was unable to compile it (e.g. I haven’t got the Lucida fonts). Also there are some packages which have to be downloaded by hand, because there are not available at CTAN. Other handicaps are the missing possibility to create a PDF-file directly and a recipe cannot be longer than a single page. So decided to take a look at the code. Step by step I replaced all critical parts. Finally the code is nearly complete different from the original and now it is possible to create beautiful designed recipes much easier (at least in my opinion).

Please note that there is no compatibility between xcookybooky and cookybooky, even the name is associating it. I chose the name, because I was was inspired by the layout.

2 Usage

In this section the most important macros are explained. After this you should be able to insert recipes. For more details take a look at the implementation in section 6. Afterwards some macros are shown, which allows you to modify the design (e.g. colors, headlines).

2.1 Main

This commands are commonly used for typesetting recipes. For more details take a look at the implementation in section 6. You can also investigate the examples and the configuration file.

\begin{recipe}
\%
  \texttt{preparationtime} = \unit [1]{h},
  \texttt{bakingtime} = \unit [1]{h},
  \texttt{bakingtemperature} = \texttt{fanoven} = \unit [230]{°C},
  \texttt{portion} = \texttt{5–6},
  \texttt{calory} = \unit [3]{kJ},
  \texttt{source} = \texttt{Somebody you used know}
\]

\{Test Recipe\}

\end{recipe}

Example 1: Recipe Environment

\graph It is possible to set two pictures by using the macro \texttt{graph}. They are displayed at the top of the page and therefore above all other content. If no pictures are set, the space is used by the other content. The \texttt{big} picture should be larger than the \texttt{small} picture of course. The width of both graphics can be adapt to your needs by applying the command \texttt{setRecipeLengths}. 
\graph{
    small = strawberry,
    big = strawberrycake
  }

Example 2: Graph Command (simple)

Another possibility is shown in the following example. It is recommended if you only want to change the widths of the pictures for one recipe, because these values are one only used for one recipe. If you want to changed it for all recipes use `setRecipeLengths`.

\graph{
    small = strawberry,
    smallpicturewidth = 0.3\textwidth,
    big = strawberrycake,
    bigpicturewidth = 0.6\textwidth,
  }

Example 3: Graph Command (extended)

\ingredients

Here the ingredients of your recipe are set. They should be entered by number (unit) & ingredient. If the unit is a SI-Unit, it looks better if you use the `units` package instead of a blank. Since the version 1.2 there is only one possibility to enter ingredients, which is shown in the example. Note that a overlong ingredient will automatically create a line break.

\ingredients{
    2 bar & Dark Chocolate (above \unit[70]{\%})\
    3 & Eggs\
    \unit[200]{ml} & Cream\
    \unit[40]{g} & Sugar\
    \unit[50]{g} & Butter
  }

Example 4: Ingredients Command

Unfortunately the package `wrapfig` has a problem to detect the length of a table. In order to get a better floating of the ingredients you have to insert the number of table lines as shown in the example below. Please note that you have to take the headline and line breaks of overlong ingredients into account.

\ingredients[7]{
    2 bar & Dark Chocolate (above \unit[70]{\%})\
    3 & Eggs\
    \unit[200]{ml} & Cream\
    \unit[40]{g} & Sugar\
    \unit[50]{g} & Butter
  }

Example 5: Ingredients Command with Optional Argument

In the versions up to 1.1 there was also a method to enter ingredients with three columns. Due to the substitution of `tabular` with `tabulary` I decided that the three columns method is no longer supported, because it would mean a lot of work. This method has also the drawback it needs more space and in my opinion it also looks ugly. If you are using the three columns solution there are two options:
1. Use the version 1.1 of this package, which can be found at https://code.google.com/p/xcookybooky/

2. Redefine the concerning command in your latex file or in the configuration file

\preparation

The preparation steps are inserted by this macro. Every step is written after step command. This is necessary for the numeration. Since version 1.4 it is allowed to use blank lines to create passages.

\preparation{
\step This is the first step for cooking a brilliant meal.
\step The second step follows...
\step and so on
\step Enjoy your meal!
}

Example 6: Preparation Command

If you don’t like the step number, you can either redefine it or simply use an enumeration as demonstrated in the next example.

\preparation{
\begin{enumerate}
\item This is the first step for cooking a brilliant meal.
\item The second step follows...
\item and so on
\item Enjoy your meal!
\end{enumerate}
}

Example 7: Preparation Command with Enumeration

\section{2.2 Advanced}

\introduction

The introduction macro can be used for inserting an introduction before the preparation and ingredients part. It has a frame and the text can contain blank lines to create passages.

\introduction{
\%
This text block can contain an introduction to the recipe.
\}

Example 8: Introduction Command

\suggestion

This command is displayed after the ingredients and preparation part of the recipe. It is expandable so that you can insert multiple suggestions. The optional parameter can be used to set a headline.

\suggestion{
\%
Another possibility to insert additional information to the recipe.
\}
\%
\suggestion[Headline]{%
Another possibility to insert additional information to the recipe.
\}

Example 9: Suggestion Command
2.3 Support Commands

\texttt{portion} This macro is intended as an interface for an equal formatting of portions entries. For this it can be easily modified. You just have to renew this command.

\texttt{\texttt{portion} \{5\}}
\texttt{\texttt{portion} \{People\}\{5\}}

Example 10: Portion Command

\texttt{bakingtemperature} This macro allows you to enter some baking temperatures. If you do not like the order of the temperatures you can change them by redefining the command. The temperatures are inserted as key/value pairs as known from the \texttt{xkeyval} package.

\begin{verbatim}
\begin{recipe}
  \bakingtemperature={\protect \bakingtemperature{
    fanoven=\unit[230]{\degree C},
    topbottomheat=\unit[195]{\degree C},
    topheat=\unit[195]{\degree C},
    bottomheat, gasstove=Level 2}
\end{recipe}
\end{verbatim}

Example 11: Baking Temperature Command

2.4 Modify Design

The following macros can be used outside the recipe environment to choose your personal color etc.

\texttt{\texttt{setRecipeColors}} If you want to change the default colors, you can use \texttt{setRecipeColors}. The colors are set by \texttt{<key = value>} as known by the \texttt{xkeyval} package. The names of colors can be found in table 1. New

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recipename</td>
<td>DarkGreen</td>
<td>recipe name</td>
</tr>
<tr>
<td>intro</td>
<td>black</td>
<td>introduction</td>
</tr>
<tr>
<td>ing</td>
<td>DarkGray</td>
<td>ingredients (text)</td>
</tr>
<tr>
<td>inghead</td>
<td>DarkGray</td>
<td>headline of the ingredients</td>
</tr>
<tr>
<td>prep</td>
<td>black</td>
<td>preparation (text)</td>
</tr>
<tr>
<td>prehead</td>
<td>black</td>
<td>headline of the preparation</td>
</tr>
<tr>
<td>suggestion</td>
<td>black</td>
<td>suggestions (text)</td>
</tr>
<tr>
<td>suggestionhead</td>
<td>black</td>
<td>headline of suggestions</td>
</tr>
<tr>
<td>separationgraph</td>
<td>black</td>
<td>graphical separator</td>
</tr>
<tr>
<td>hint</td>
<td>red</td>
<td>hint (text)</td>
</tr>
<tr>
<td>hinthead</td>
<td>red</td>
<td>headline of the hint</td>
</tr>
<tr>
<td>hintline</td>
<td>red</td>
<td>lines leftside the hint text (cross)</td>
</tr>
<tr>
<td>numeration</td>
<td>red</td>
<td>numeration of the preparation steps</td>
</tr>
</tbody>
</table>

Table 1: Color Management

colors can be easily defined by the \texttt{xcolor} package as shown the example below.
2.4 Modify Design

\definecolor{webgreen}{rgb}{0,1,0}
\setRecipeColors{
  recipename = webgreen,
  ing = blue,
  numeration = black,
}

Example 12: Setting Recipe Colors Command

\setRecipeLengths{\setRecipeSizes{\setRecipeColors{\setRecipeLengths{
  preparationwidth = 0.5\textwidth,
  ingredientswidth = 0.4\textwidth,
}}

Example 13: Setting Recipe Sizes Command

\setRecipeLengths{ preparationwidth = 0.5\textwidth,
  ingredientswidth = 0.4\textwidth,
}

Example 14: Setting Recipe Lengths Command

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pictureheight</td>
<td>height of both pictures</td>
</tr>
<tr>
<td>bigpicturewidth</td>
<td>width of the bigger picture</td>
</tr>
<tr>
<td>smallpicturewidth</td>
<td>width of the smaller picture</td>
</tr>
<tr>
<td>introductionwidth</td>
<td>width of introduction</td>
</tr>
<tr>
<td>preparationwidth</td>
<td>width of preparation</td>
</tr>
<tr>
<td>ingredientswidth</td>
<td>width of ingredients</td>
</tr>
</tbody>
</table>

Table 2: Recipe Lengths

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recipename</td>
<td>font size of recipe name</td>
</tr>
<tr>
<td>intro</td>
<td>font size of introduction (text)</td>
</tr>
<tr>
<td>ing</td>
<td>font size of ingredients (text)</td>
</tr>
<tr>
<td>inghead</td>
<td>font size of headline of ingredients</td>
</tr>
<tr>
<td>prehead</td>
<td>font size of headline of preparation</td>
</tr>
<tr>
<td>suggestion</td>
<td>font size of suggestion (text)</td>
</tr>
<tr>
<td>hint</td>
<td>font size of hint (text)</td>
</tr>
<tr>
<td>hinthead</td>
<td>font size of headline of hint</td>
</tr>
</tbody>
</table>

Table 3: Font Size Management

\setRecipeSizes{ recipename = \fontsize{25pt}{30pt},
  ing = \normalsize,
  hinthead = \Large
}

Example 14: Setting Recipe Sizes Command
The name of recipe is only displayed with a handwritten font, if the corresponding option is selected. See section 2.6 for details. In my opinion it looks much better than a standard font. But if you want to change it you can use this macro. A well-arranged list of fonts can be found at [http://www.tug.dk/FontCatalogue/](http://www.tug.dk/FontCatalogue/). The four parameters are font name, font encoding, font series and font shape. The size is changed by \setRecipenameSizes. The following example shows a bunch of interesting fonts, which you should try out.

\setRecipenameFont{cmr}{T1}{m}{n} % probably the default setting
\usepackage{pbsi}
\setRecipenameFont{pbsi}{T1}{xl}{n}

Example 15: Setting Recipename Font Command (package pbsi)

\usepackage{emerald}
\setRecipenameFont{fau}{T1}{m}{n}
\setRecipenameFont{fwb}{T1}{m}{n}
\setRecipenameFont{fjd}{T1}{m}{n} % default by option handwritten

Example 16: Setting Recipename Font Command (package emerald)

2.5 Translation

\setHeadlines

The command \setHeadlines allows to modify the (translated) headlines. This is also needed for translations, if the used language is not supported yet. If you have improvements for a translation or created a new one, feel free to send me an e-mail. The headlines are also inserted in \texttt{<key = value>} form. All possible key are described in table 4.

<table>
<thead>
<tr>
<th>Key</th>
<th>Graphic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inghead</td>
<td>no</td>
<td>Headline of ingredients</td>
</tr>
<tr>
<td>prephed</td>
<td>no</td>
<td>Headline of preparation</td>
</tr>
<tr>
<td>hinthead</td>
<td>no</td>
<td>Headline of hint</td>
</tr>
<tr>
<td>continuationhead</td>
<td>no</td>
<td>Text at the top of the second page of a recipe</td>
</tr>
<tr>
<td>continuationfoot</td>
<td>no</td>
<td>Text at the bottom of the first page, if the recipe is longer than one page</td>
</tr>
<tr>
<td>preparationtime</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>bakingtime</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>bakingtemperature</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>portion</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>portionvalue</td>
<td>no</td>
<td>Unit for the portions</td>
</tr>
<tr>
<td>calory</td>
<td>no</td>
<td>Text for the calorific value</td>
</tr>
<tr>
<td>source</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Recipe Headlines

\setHeadlines{
  inghead = Ingredients ,
  hinthead = Straight Tip ,
  continuationhead = Continuation \texttt{\textbackslash dots} ,
}

Example 17: Setting Headlines Command
2.6 Options

This package provides three options which are explained in the following.

- **handwritten** This option requires the non-free package `emerald` and can only be used if your are not using TeX Live.

- **nowarnings** This options tries to eliminate all nasty warnings. Only important and interesting warnings are logged. This is in my opinion a good choice because the package `wrapfigure` generates a lot of warnings, see section 5.1.

- **myconfig** If you are using this package often or want to keep your main file 'clean' you can define your modifications in the configuration file. If you submit this option the file will be loaded otherwise not.

2.7 Background graphic

Inside this package there is only the option to set a background picture using the `setBackgroundPicture` macro. For creating transparent backgrounds I recommend Inkscape\textsuperscript{1}. You just have to import a picture and reduce the opacity of the picture (I prefer 22\%). The possible options for the orientation are:

- default
- pagecenter
- textcenter
- stockcenter

These orientations can be modified by using the x and y option. This is demonstrated in the following example.

\begin{verbatim}
\setBackgroundPicture
[ %
  x = 2cm, 
  y = -1cm, 
  width=\paperwidth-3cm, 
  height ,
  orientation=pagecenter
]{pic/bg_transparent} % filepath
\end{verbatim}

Example 18: Background Picture Command

\textsuperscript{1}http://inkscape.org/
Mousse au Chocolat

1 h
5 Personen
R. Gaus

Zutaten
2 Tafeln dunkle Schokolade
über 70 %
3 Eier
200 ml Sahne
40 g Zucker
50 g Butter

Zubereitung
1 Eier trennen, Eiweiß und Sahne separat steif schlagen. Butter und Schokolade vorsichtig im Wasserbad schmelzen.

2 Eigelb in einer großen Schüssel mit 2 EL heißem Wasser cremig schlagen, den Zucker einrühren bis die Masse hell und cremig ist.

3 Die geschmolzene Schokolade unterheben, anschließend sofort Eischnee und Sahne unterheben (nicht mit dem Elektro-Mixer!)

4 Mindestens 2 Stunden im Kühlschrank kalt stellen. Aber nicht zu kalt servieren.

Tipp
Der Schokoladenanteil kann auch gesenkt werden.
4 Troubleshooting

4.1 MiKTeX

If you have problems with the used fonts (e.g. cookingsymbols) it might be necessary to refresh the database of MiKTeX.

4.2 TeX Live

If you are using this distribution you may should not use the option handwritten, because this option load the font package emerald. This package is non-free and therefore not available in TeX Live.

5 Bugs

If you found a bug, which is not described in the following section, feel free to write me an email. It can be found at the titlepage. Alternatively you can visit https://code.google.com/p/xcookybooky/ and report the issue. All reported problems are registered there.

5.1 Known Issues

The wrapfig package has some problems with enumerations and the lettrine package. This is the reason, why some warnings are generated by compiling (the example), if do not use the package option nowarnings. But nevertheless the results look quite well, so I decided to ignore this issue. By using wrapfig it is not possible to switch ingredients and preparation (the result looks really ugly). This is the reason, why the layout is not changed when setting the twoside option in the document class. I also tried parcolumns and parallel, but the results were disappointing. If you want to use this feature, you have to use cookybooky.

Another limitation is the length of the ingredients. The maximum length is one page, because the wrapfigure package does not work with longtable. As a result no pagebreak in the ingredients is possible.
6 Implementation

Let’s have a look at the implementation.

6.1 Options

xcookybooky provides three options as explained in section 2.6. You can use the option myconfig to load your personal configuration (file). See section 6.14 for details.

\begin{verbatim}
\newif\ifHandwrittenFont@required
\HandwrittenFont@requiredfalse
\DeclareOption{handwritten}
\{\% displays the recipename in a handwritten font
  \HandwrittenFont@requiredtrue
  \AtEndOfPackage{
    \setRecipeSizes{recipename = \fontsize{25pt}{30pt}}
    \setRecipeFont{fjd}{T1}{m}{n}
  }
\}
\DeclareOption{nowarnings}
\{% tries to eliminate nasty warnings
  \AtEndOfPackage{
    \def\WF@conflict{}
    \%\def@fancywarning#1{}
  }
\}
\DeclareOption{myconfig}
\{% load personal configuration
  \AtEndOfPackage{
    \InputIfFileExists{xcookybooky.cfg}
    \{\PackageInfo{xcookybooky}{inputting 'xcookybooky.cfg'}}
    \{\PackageWarning{xcookybooky}{cannot find the configuration file 'xcookybooky.cfg'}}
  }
\}
\ProcessOptions\relax
\end{verbatim}

6.2 Required Packages

All needed packages are available at CTAN and as far as I know no general driver limitation are given.

\begin{verbatim}
\RequirePackage{tikz} % for creating the lines for the hint
\RequirePackage{graphicx}
\RequirePackage{xcolor}
\RequirePackage[clock, misc, weather]{ifsym} % Symbol package e.g. \Interval, \Wecker etc
\RequirePackage{cookingsymbols} % Cookings symbols e.g. \Oven, \Dish etc
\ifHandwrittenFont@required
\RequirePackage{emerald} % Handwritten Fonts
\fi
\RequirePackage{wrapfig} % To wrap the tabular with the ingredients
\RequirePackage{iflang} % detecting the language
\end{verbatim}
6.3 Page Layout

This will set up the page. It is optimized for the document class \texttt{article}. If you use an other class you probably have to modify this.

\textbf{Note:} I tried to use the KOMA class \texttt{scrartcl}, but unfortunately some strange effects occured.

\begin{verbatim}
\setlength{\parindent}{0cm}
\setlength{\parskip}{2ex plus 0.5ex}
\setlength{\textwidth}{15.5cm}
\setlength{\textheight}{24.0cm}
\setlength{\topmargin}{-0.8cm}
\setlength{\headheight}{14pt} %0cm
\setlength{\headsep}{1cm}
\setlength{\topskip}{0cm}
\setlength{\footskip}{1.4cm}
\setlength{\evensidemargin}{-0.5cm}
\setlength{\oddsidemargin}{0.5cm}
\setlength{\voffset}{0cm}
\setlength{\hoffset}{0cm}
\setcounter{secnumdepth}{2} % subsubsections not numbered
\setcounter{tocdepth}{3} % subsubsections in the .toc file

\recipesection

The macro \texttt{recipesection} allows to modify the entry in the table of contents.

\textbf{Note:} If you modify the macro you may change the two upper lines as well.
\end{verbatim}

6.4 Color Management

At first the used non-standard colors must be defined.

\begin{verbatim}
\providecolor{DarkGreen}{rgb}{0,.5,0}
\providecolor{DarkBlue}{rgb}{0, 0, 0.7}
\providecolor{DarkGray}{rgb}{0.23, 0.23, 0.23}
\end{verbatim}
Below the (default) colors are defined using the \texttt{xkeyval} package.

\begin{verbatim}
\define@key{colorManagement}{recipename}[DarkGreen]{\def\xcb@color@recipename{#1}}
\define@key{colorManagement}{intro}[black]{\def\xcb@color@intro{#1}}
\define@key{colorManagement}{ing}[DarkGray]{\def\xcb@color@ing{#1}}
\define@key{colorManagement}{inghead}[DarkGray]{\def\xcb@color@inghead{#1}}
\define@key{colorManagement}{prep}[black]{\def\xcb@color@prep{#1}}
\define@key{colorManagement}{prephead}[black]{\def\xcb@color@prephead{#1}}
\define@key{colorManagement}{suggestion}[black]{\def\xcb@color@suggestion{#1}}
\define@key{colorManagement}{suggestionhead}[black]{\def\xcb@color@suggestionhead{#1}}
\define@key{colorManagement}{separationgraph}[black]{\def\xcb@color@separationgraph{#1}}
\define@key{colorManagement}{hint}[black]{\def\xcb@color@hint{#1}}
\define@key{colorManagement}{hinthead}[red]{\def\xcb@color@hinthead{#1}}
\define@key{colorManagement}{hintline}[red]{\def\xcb@color@hintline{#1}}
\define@key{colorManagement}{numeration}[red]{\def\xcb@color@numeration{#1}}
\savekeys{colorManagement}
\end{verbatim}

This is the initialisation of all required recipe colors.

\begin{verbatim}
\setRecipeColors
\newcommand*{\setRecipeColors}[1]{\setkeys{colorManagement}{#1}}
\setkeys{colorManagement}{#1}
\end{verbatim}

This is the initialisation of all required recipe colors.
6.5 Size Management

Below the (default) font sizes of the recipe are defined.

\define@key{sizeManagement}{recipename}{\fontsize{25pt}{30pt}\selectfont}{\def\xcb@fontsize@recipename{#1}}
\define@key{sizeManagement}{intro}{\selectfont}{\def\xcb@fontsize@intro{#1}}
\define@key{sizeManagement}{ing}{\selectfont}{\def\xcb@fontsize@ing{#1}}
\define@key{sizeManagement}{inghead}{\selectfont}{\def\xcb@fontsize@inghead{#1}}
\define@key{sizeManagement}{prep}{\selectfont}{\def\xcb@fontsize@prep{#1}}
\define@key{sizeManagement}{prephead}{\selectfont}{\def\xcb@fontsize@prephead{#1}}
\define@key{sizeManagement}{suggestion}{\selectfont}{\def\xcb@fontsize@suggestion{#1}}
\define@key{sizeManagement}{hint}{\selectfont}{\def\xcb@fontsize@hint{#1}}
\define@key{sizeManagement}{hinthead}{\Large}{\def\xcb@fontsize@hinthead{#1}}

\savekeys{sizeManagement}

The parameter are set in the form "key = value" as known by the xkeyval package.

\newcommand*{\setRecipeSizes}[1]{\setkeys{sizeManagement}{#1}}

This is the initialisation of all required recipe sizes.

\setRecipeSizes

\setRecipeSizes
6.6 Recipe Font

This macro sets the font for the name of the recipe. You have to insert the font family, font encoding, font series and font shape. The size is set by the command \sizeManagement.

\begin{verbatim}
\setRecipenameFont\{familydefault\}{encodingdefault\}{b\}{n}
\end{verbatim}

We need a counter for the numeration of the preparation steps.

\newcounter{step}
\setcounter{step}{0}

Set the color of the numeration of preparation steps.

\renewcommand{\LettrineFontHook}{% \color{\xcb@color@numeration}

6.7 Headlines and Translations

The following section defines the default English headlines of the recipe. They are only used if no supported language is detected.

\define@key{headlines}{inghead}[Ingredients]{
\def\xcb@name@inghead{\textcolor{\xcb@color@inghead}{\textbf{\xcb@fontsize@inghead{#1}}}}}
\define@key{headlines}{prephead}[Preparation]{
\def\xcb@name@prephead{\textcolor{\xcb@color@prephead}{\textbf{\xcb@fontsize@prephead{#1}}}}}
\define@key{headlines}{hinthead}[Hint]{
\def\xcb@name@hinthead{\textcolor{\xcb@color@hinthead}{\xcb@fontsize@hinthead{#1}}}}
6.7 Headlines and Translations

\setHeadlines

This allows you to modify the headlines for the recipe parts. This also needed for translations. If you are going to define a language which is not supported by \texttt{xcookybooky} yet, please send me an email and I will integrate it in the next version! The parameter are set in the form 'key = value' as known by the \texttt{xkeyval} package.

\newcommand*{\setHeadlines}[1]{\setkeys{headlines}{#1}}
\setHeadlines

\IfLanguagePatterns{german}{% German
\setHeadlines

After the initialization the supported languages are used as default values.
6.8 Recipe Lengths

The values \texttt{xcb@bigpicturewidth} and \texttt{xcb@smallpicturewidth} are the default values. They can be temporarily changed with the \texttt{graph} command, i.e for a single recipe.

\begin{verbatim}
define@key{recipelengths}{pictureheight}[6cm]{\def{xcb@pictureheight}{#1}}
define@key{recipelengths}{bigpicturewidth}[0.60\textwidth]{\def{xcb@bigpicturewidth}{#1}}
define@key{recipelengths}{smallpicturewidth}[0.35\textwidth]{\def{xcb@smallpicturewidth}{#1}}
define@key{recipelengths}{introductionwidth}[\textwidth]{\def{xcb@introductionwidth}{#1}}
define@key{recipelengths}{preparationwidth}[0.60\textwidth]{\def{xcb@preparationwidth}{#1}}
define@key{recipelengths}{ingredientswidth}[0.35\textwidth]{\def{xcb@ingredientswidth}{#1}}
savekeys{recipelengths}
{
pictureheight, \% height of both pictures
bigpicturewidth, \% width of the bigger picture
smallpicturewidth, \% width of the smaller picture
introductionwidth, \% width of introduction
preparationwidth, \% width of preparation
ingredientswidth \% width of ingredients
}
\end{verbatim}

\texttt{\setRecipeLengths} This command allows to modify most sizes of the recipe.

\begin{verbatim}
newcommand*{\setRecipeLengths}[1]
{%
\setkeys{recipelengths}{#1}
}
\end{verbatim}
6.9 Recipe Commands

The following commands are implemented for inserting the recipe content.

Main Commands

These commands provide the main functionality for inserting a recipe.

\introduction

\%\% MAIN COMMANDS
\newcommand{\introduction}[1]
\% Introduction before the preparation and ingredients
\def\xcb@introduction
\%\color{\xcb@color@intro}
\begin{framed}
\ xcb@fontsize@intro#1
\end{framed}
\
\graph
This command is used for entering the filename of the pictures. If a picture is not set, it is not shown.

\newcommand*{\graph}[1]
\%\ \setkeys{picture}{#1}
\}

\ingredients
The command is used to insert the ingredients, which are displayed as a table. The optional argument contains the number of table lines. It can be used to adjust the floating of the ingredients table. The package wrapfig has a problem to detect the length of the table, which is shown at \url{http://tex.stackexchange.com/questions/83231/text-not-spanning-full-page-after-wraptable} for example.

\newcommand*{\ingredients}[2][\empty]
\% The optional argument contains the number of lines
\def\xcb@ingredientslines(#1)
\def\xcb@ingredients
\%\ xcb@name@inghead
\\[1em]
This command enables the user to insert the preparation steps.

\newcommand{\preparation}[1]\[
\begin{tabulary}{\ingredientswidth}{rL}
#1
\end{tabulary}
\]
\setcounter{step}{0}

The command is intended to insert some suggestions after the preparation and ingredients part. It is expandable in order to create more than one suggestion in one recipe.

\newcommand{\suggestion}[2]\[
\g@addto@macro{\suggestion}{\xcb@separationgraph{\color{\xcb@color@separationgraph}}
\ifthenelse{\equal{#1}{}}{
\textbf{\color{\xcb@color@suggestionhead}{#1}}\color{\xcb@color@suggestion}{#2}
}\{}
\{}
\]

This command allows the user to insert an optional hint at the bottom of the recipe.

\newcommand{\hint}[1]\[
\pagebreak[1] \% begin here a new page, if the space is small
\begin{tikzpicture}[line width = 1.5pt, \color{\ xcb@color@hintline}]
\useasboundingbox (0,0) -- ++(right:3.5cm) -- ++(up:3mm) -- ++(left:3.5cm) -- cycle;
\draw (-3mm, 0mm) -- ++(right:3.5cm);
\draw (0, 3mm) -- ++(down:2.4cm);
\end{tikzpicture}
\]
Supporting Commands

The following commands are intended to simplify inserting some parts of the recipe like backing temperatures.

\step The command is used for numbering the steps of preparation. Take a look at the lettrine package for details.

\portion This macro is for inserting the portions correctly. By renewing this command you can easily define your own design for the portions.

\bakingtemperature This macro is used for inserting different baking temperatures. Only entered temperatures are displayed (with a symbol from cooking symbols package). You can change the order of the temperatures by renew this command.
\define@key{bakingtemperature}{gasstove}{\def\xcb@bakingtemperature@gasstove{#1}}

\newboolean{xcb@bakingtemperature@firstentry}
\newcommand*{\bakingtemperature}[1]{
    \setboolean{xcb@bakingtemperature@firstentry}{true}
    \setkeys{bakingtemperature}{fanoven, topbottomheat, topheat, bottomheat, gasstove}
    \setkeys{bakingtemperature}{#1}
    \ifthenelse{\equal{\xcb@bakingtemperature@fanoven}{}}{}
    \else{
        \xcb@bakingtemperature@fanoven \ \Fanoven\%
        \setboolean{xcb@bakingtemperature@firstentry}{false}
    }\fi
    \ifthenelse{\equal{\xcb@bakingtemperature@topbottomheat}{}}{}
    \else{
        \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
        \xcb@bakingtemperature@topbottomheat \ \Topbottomheat\%
    }\fi
    \ifthenelse{\equal{\xcb@bakingtemperature@topheat}{}}{}
    \else{
        \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
        \xcb@bakingtemperature@topheat \ \Topheat\%
    }\fi
    \ifthenelse{\equal{\xcb@bakingtemperature@bottomheat}{}}{}
    \else{
        \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
        \xcb@bakingtemperature@bottomheat \ \Bottomheat\%
    }\fi
    \ifthenelse{\equal{\xcb@bakingtemperature@gasstove}{}}{}
    \else{
        \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
        \xcb@bakingtemperature@gasstove \ \Gasstove\%
    }\fi}

%% INTERNAL COMMANDS
\newcommand{\xcb@cmd@recipeoverview}{
\begin{tabular}{rl}
    \ifthenelse{\equal{\xcb@preparationtime}{}}{% not set => not shown
        \else{
            \xcb@name@preparationtime & \xcb@preparationtime\%
        }\fi
    \ifthenelse{\equal{\xcb@bakingtime}{}{}\}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
    \xcb@bakingtemperature@bakingtime \ \Bakingtime\%
    \ifthenelse{\equal{\xcb@bakingtime}{}{}\}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
    \xcb@bakingtemperature@bakingtime \ \Bakingtime\%
    \ifthenelse{\equal{\xcb@bakingtime}{}{}\}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
    \xcb@bakingtemperature@bakingtime \ \Bakingtime\%
    \ifthenelse{\equal{\xcb@bakingtime}{}{}\}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
    \xcb@bakingtemperature@bakingtime \ \Bakingtime\%
    \ifthenelse{\equal{\xcb@bakingtime}{}{}\}{\setboolean{xcb@bakingtemperature@firstentry}{false}}{, }
    \xcb@bakingtemperature@bakingtime \ \Bakingtime\%
\end{tabular}
}
The following command is needed, if more ingredients than preparation (steps) are inserted. In this case the \texttt{wrapfigure} package behaves in a strange way. The result is that the hint is also floating around the ingredients, although the "box" is already closed. This command repairs this effect. It is taken from \texttt{http://mizine.de/latex/wrapfigure-austricksen/}.

\newcommand*{\cbx@cmd@wrapfill}{% Bug fix for wrapfigure package
\par
\ifx\parshape\WF@fudgeparshape
\nobreak
\vskip-\baselineskip
\vskip\c@WF@wrappedlines\baselineskip
\allowbreak
\WFclear
\fi}
\fi}
\end{tabular}

The following code lines define the behaviour of the two pictures

\define@key{picture}{small}[]{
\def\cbx@picture@small{
\ifthenelse{\equal{#1}{}}{% not set => not shown
\else
\cbx@name@portion & \cbx@portion
\fi
\cbx@name@calory & \cbx@calory
\cbx@name@source & \cbx@source
\end{tabular}

\cbx@cmd@wrapfill
6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.

\texttt{\textbackslash newcommand*{xcb@separationgraph}[2][1.5]}
\begin{center}
\begin{tikzpicture}[scale=#1]
\fill[color=#2] (0,0) -- (-0.2, 0.1) -- (-4, 0) -- (-0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) -- (0.2, 0.1) -- (4, 0) -- (0.2, -0.1) -- cycle;
\fill[color=#2] (0,0) circle (0.1);
\end{tikzpicture}
\end{center}

6.10 Hooks

The following hooks allow the user to insert data many different places in the recipe in order to customize the recipe.
The content of this hook command is inserted before the (two) pictures. Therefore it is the first page content (aside from the page headline etc).

\newcommand*{\pregraph}{% Hook: entered before the pictures
  \def\xcb@hook@pregraph
    \begin{minipage}[T]{\textwidth}
      \hspace{0.6cm}
    \end{minipage}
}\}

This command is inserted before the title of the recipe. That means it is displayed after the pictures.

\newcommand*{\pretitle}{% Hook: entered before the recipe title
  \def\xcb@hook@pretitle
    \hspace{0.6cm}
}\}

The next macro is integrated before the overview of the recipe and therefore after the recipe title.

\newcommand*{\prerecipeoverview}{% Hook: entered before the recipe overview
  \def\xcb@hook@prerecipeoverview
    \hspace{0.6cm}
}\}

This command is inserted before the preparation headline and steps, i.e. it is beside the ingredients.

\newcommand*{\prepreparation}{% Hook: entered before the preparation (steps)
  \def\xcb@hook@prepreparation
    \hspace{0.6cm}
}\}

This command is inserted after the preparation headline and steps, i.e. it is beside the ingredients.

\newcommand*{\postpreparation}{% Hook: entered after the preparation (steps)
  \def\xcb@hook@postpreparation
    \hspace{0.6cm}
}\}
The following hook command is integrated before the ingredients. That means it parallel to the preparation.

\newcommand*{\preingredients}{% Hook: entered before the ingredients
\def\xcb@hook@preingredients{% 
#1 
}
}

The following hook command is integrated after the ingredients. That means it parallel to the preparation.

\newcommand*{\postingredients}{% Hook: entered after the ingredients
\def\xcb@hook@postingredients{% 
#1 
}
}

### 6.11 Recipe Environment

This the environment for inserting recipes. Inside you define your ingredients, preparation, pictures (graph) and maybe suggestions and/or a hint. The commands xcb@cmd@myrecipeoverview and xcb@cmd@wrapfill are intended for internal use only.

\newenvironment{recipe}[2][]{% initialisation
\setkeys{recipe}{preparationtime, bakingtime, bakingtemperature, portion, calory, source}
\setkeys{picture}{small, big, smallpicturewidth=\xcb@smallpicturewidth, bigpicturewidth=\xcb@bigpicturewidth}
\def\xcb@hook@pregraph{}
\def\xcb@hook@pretitle{}
\def\xcb@introduction{}
\def\xcb@hook@prepreparation{}
\preparation{}
\def\xcb@hook@postpreparation{}
\def\ xcb@hook@preingredients{}
\ingredients{}
\def\xcb@hook@postingredients{}
\def\ xcb@hook@suggestion{}
\def\ xcb@hint{}
\def\ xcb@recipename{#2}
\setkeys{recipe}{#1} % reading the optional parameters
\setcounter{xcb@newpagefoot}{1}
6.11 Recipe Environment

\setcounter{xcb@newpagehead}{\value{page}}

\% this part is executed at \end{recipe}
%% FIRST BLOCK
\xcb@hook@pregraph
\if@twoside
\ifodd\arabic{page}
\begin{minipage}[T]{\xcb@picture@bigwidth}
\xcb@picture@big
\end{minipage}
\hfill
\begin{minipage}[T]{\xcb@picture@smallwidth}
\xcb@picture@small
\end{minipage}
\else
\begin{minipage}[T]{\xcb@picture@smallwidth}
\xcb@picture@small
\end{minipage}
\hfill
\begin{minipage}[T]{\xcb@picture@bigwidth}
\ xcb@picture@big
\end{minipage}
\fi
\else
\begin{minipage}[T]{\xcb@picture@bigwidth}
\ xcb@picture@big
\end{minipage}
\hfill
\begin{minipage}[T]{\xcb@picture@smallwidth}
\ xcb@picture@small
\end{minipage}
\fi
\else
%% SECOND BLOCK
\xcb@hook@pretitle
\recipesection[\normalsize xcb@recipename]%%
\hspace{-1em}\textcolor{xcb@color@recipename}{xcb@font@recipename xcb@recipename}\
 xcb@cmd@recipeoverview
\xcb@introduction
%% THIRD BLOCK
\ifthenelse{\equal{\ xcb@ingredientslines}{\empty}}%\%
\begin{wraptable}{r}{\ xcb@ingredientswidth}
\% No line number given by the user
\vspace{-1em} % same height of ingredients and preparation
 xcb@hook@preingredients
\xcb@ingredients
\xcb@hook@postingredients
\end{wraptable}
\%
\begin{wraptable}[\ xcb@ingredientslines]{r}{\ xcb@ingredientswidth}
 xcb@hook@preingredients
\xcb@ingredients
\xcb@hook@postingredients
\end{wraptable}
\%
\begin{wraptable}[\ xcb@ingredientslines]{r}{\ xcb@ingredientswidth}
 xcb@hook@preingredients
\xcb@ingredients
\xcb@hook@postingredients
\end{wraptable}
\% Use the given line number by the user
\vspace{-1em} % same height of ingredients and preparation
\ xcb@hook@preingredients
\ xcb@ingredients
\ xcb@hook@postingredients
} \end{wraptable}
}
\ xcb@hook@prepreparation
\ xcb@preparation
\ xcb@hook@postpreparation
\ xcb@suggestion
\ vfill
\ xcb@cmd@wrapfill
\ xcb@hint
\setcounter{xcb@newpagefoot}{0}
\}
\}

6.12  Header and Footer

\pagestyle{fancy}
\renewcommand{\sectionmark}{\@newpagehint{head}}
\renewcommand{\headrulewidth}{0.5pt}
\fancyhf{}
\if@twoside
\fancyfoot[LE,RO]{\textbf{\@newpagehint{head}}}
\fancyhead[LE,RO]{\textbf{\#1}}
\fancyhead[LE,RO]{\rightmark}
\else
\fancyfoot[R]{\textbf{\@newpagehint{head}}}
\fancyhead[R]{\textbf{\@newpagehint{foot}}}
\fi
\setcounter{xcb@newpagefoot}{0}

Below the counter for the header and the footer are defined and initialised.

\newcounter{xcb@newpagehead}
\setcounter{xcb@newpagehead}{0}
\newcounter{xcb@newpagefoot}
\setcounter{xcb@newpagefoot}{0}
6.13 Background Picture

This macro is intended to set easily a (transparent) background. The command uses the `eso-pic` package. If the implemented options are not satisfying you, you may take a look at the documentation of `eso-pic` when redefining this command.

\newcommand*{\setBackgroundPicture}[2][]

{%
\ClearShipoutPictureBG
\setkeys{background}{width, height, x, y, orientation} \% initialisation
\setkeys{background}{#1}
\AddToShipoutPicture
%}
\ifthenelse{\equal{\xcb@background@orientation}{default}}{% \% default
\put(\xcb@background@x, \xcb@background@y)
6.14 Configuration file

This file is intended to change the design central. The most important commands are already written down. You have to submit the option myconfig, if you want to load the configuration file.

```
(*config)
\setRecipeColors {%
    recipename, intro, ing, inghead, prep, prehead, suggestion, suggestionhead, separationgraph, hint, hinthead, hintline, numeration }
\setRecipeSizes {%
    recipename, intro, ing, inghead, prep, prehead, suggestion, hint, hinthead }
\setRecipeLengths {%
    pictureheight, bigpicturewidth, smallpicturewidth, introductionwidth, preparationwidth, ingredientswidth }
\setRecipenameFont{cmr}{T1}{m}{n}
\setHeadlines {%
    inghead, prehead, hinthead, continuationhead, continuationfoot, preparationtime, }
```
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.

Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>\v</td>
<td>308, 310, 322, 324</td>
</tr>
<tr>
<td>\fancywarning</td>
<td>16</td>
</tr>
<tr>
<td>\newpagehint</td>
<td>781, 782, 788</td>
</tr>
<tr>
<td>-</td>
<td>305, 307, 309, 319, 321, 323</td>
</tr>
<tr>
<td>_</td>
<td>192, 195, 476, 483, 489, 495, 501, 767</td>
</tr>
<tr>
<td>\AddToShipoutPicture</td>
<td>828</td>
</tr>
<tr>
<td>\allowbreak</td>
<td>554</td>
</tr>
<tr>
<td>\arabic</td>
<td>688</td>
</tr>
<tr>
<td>\AtEndOfPackage</td>
<td>7, 14, 21</td>
</tr>
<tr>
<td>\AtPageCenter</td>
<td>845</td>
</tr>
<tr>
<td>\AtStockCenter</td>
<td>865</td>
</tr>
<tr>
<td>\AtTextCenter</td>
<td>855</td>
</tr>
<tr>
<td>\bakingtemperature</td>
<td>460</td>
</tr>
<tr>
<td>\baselineskip</td>
<td>482, 488, 494, 500</td>
</tr>
<tr>
<td>\boolean</td>
<td>495</td>
</tr>
<tr>
<td>\c</td>
<td>305, 307, 309, 319, 321, 323</td>
</tr>
<tr>
<td>\c@WF@wrappedlines</td>
<td>553</td>
</tr>
<tr>
<td>\ClearShipoutPictureBG</td>
<td>825</td>
</tr>
<tr>
<td>\color</td>
<td>188, 363, 382, 395, 407, 410, 412, 435</td>
</tr>
<tr>
<td>\DeclareOption</td>
<td>4, 12, 19</td>
</tr>
<tr>
<td>\Dish</td>
<td>33, 205</td>
</tr>
<tr>
<td>\draw</td>
<td>424, 425</td>
</tr>
<tr>
<td>\encodingdefault</td>
<td>182</td>
</tr>
<tr>
<td>environments:</td>
<td></td>
</tr>
<tr>
<td>recipe</td>
<td>662</td>
</tr>
<tr>
<td>\evensidemargin</td>
<td>60</td>
</tr>
<tr>
<td>\familydefault</td>
<td>775, 778, 782</td>
</tr>
<tr>
<td>\fancyfoot</td>
<td>776, 779, 781</td>
</tr>
<tr>
<td>\fancyhead</td>
<td>773</td>
</tr>
<tr>
<td>\Fanoven</td>
<td>476</td>
</tr>
<tr>
<td>\fill</td>
<td>595-597</td>
</tr>
<tr>
<td>\fontencoding</td>
<td>175</td>
</tr>
<tr>
<td>\fontfamily</td>
<td>174</td>
</tr>
<tr>
<td>\fontseries</td>
<td>176</td>
</tr>
<tr>
<td>\fontshape</td>
<td>177</td>
</tr>
<tr>
<td>\fontsize</td>
<td>8, 128, 202, 203, 205, 476, 483, 489, 495, 501</td>
</tr>
<tr>
<td>\footskip</td>
<td>59</td>
</tr>
<tr>
<td>\g@addto@macro</td>
<td>402</td>
</tr>
<tr>
<td>\Gasstove</td>
<td>501</td>
</tr>
<tr>
<td>\Gloves</td>
<td>202</td>
</tr>
<tr>
<td>\graph</td>
<td>370</td>
</tr>
<tr>
<td>\HandwrittenFont@requiredfalse</td>
<td>3</td>
</tr>
<tr>
<td>\HandwrittenFont@requiredtrue</td>
<td>6</td>
</tr>
<tr>
<td>\headheight</td>
<td>56</td>
</tr>
<tr>
<td>\headrulewidth</td>
<td>769</td>
</tr>
<tr>
<td>\headsep</td>
<td>57</td>
</tr>
<tr>
<td>\hfill</td>
<td>692, 700, 709</td>
</tr>
<tr>
<td>\hint</td>
<td>417</td>
</tr>
<tr>
<td>\hoffset</td>
<td>63</td>
</tr>
<tr>
<td>\hspace</td>
<td>523, 718</td>
</tr>
<tr>
<td>\if@twoside</td>
<td>687, 774</td>
</tr>
<tr>
<td>\ifHandwrittenFont@required</td>
<td>2, 34</td>
</tr>
<tr>
<td>\IfLanguagePatterns</td>
<td>244, 258, 272, 286, 300, 314</td>
</tr>
<tr>
<td>\ifodd</td>
<td>688</td>
</tr>
<tr>
<td>\includegraphics</td>
<td>566, 579, 837, 848, 858, 868</td>
</tr>
<tr>
<td>\ingredients</td>
<td>375, 673</td>
</tr>
</tbody>
</table>