The xcookybooky package*

Sven Harder
sven_one1@gmx.de

May 24, 2014

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 2 pictures, which are displayes above the name of the recipe. Another option is to write a hint, which is displayed at the bottom.

*This document corresponds to xcookybooky v1.1, dated 2012/04/06.

Contents

1 Introduction

When I was looking for template for recipes, I found the cookybooky package by Jï¿½rgen Gilg (http://www.ctan.org/tex-archive/macros/latex/contrib/cookybooky). It looks very good, but I was unable to compile it correctly (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 it is now possible to write recipes much easier (at least in my opinion).

2 Comparison between xcookybooky and cookybooky

There is no compatibility between xcookybooky and cookybooky, even the name is associating it. The reason for the naming is nearly similar design.

A comparison between `xcookybooky` and `cookybooky` is shown in table **??**.

Characteristic	**xcookybooky**	**cookybooky**
Maximum recipe length	unlimited	1 page
Support missing pictures	yes	no
Transparent background graphic	not part of package (see section **??**)	yes
Main Layout	wraptable	minipages
Support twoside option	only changing pictures above	full
Generate recipe	environment	macro

Table 1: Comparison between xcookybooky and cookybooky

3 Usage

First I want to explain the most important macros. After this you should be able to insert recipes. Afterwards some macros are shown, which allows you to modify the design (e.g. colors, headlines).

`recipe` This the environment to insert recipes. Each recipe is included in a single enviroment. It is necessary to enter the name of the recipe. It is possible to insert also the recipetime, the portions, the calory content and the source (where you get this recipe).

Example:

```
\begin{recipe}[
    preparationtime = 30 min,
    bakingtime = 1 hour,
    bakingtemperature = \unit[225]{ï¿½C},
    portion = 4 portions,
    calory = 4 kj,
    source = My neighbour Sarah
]{Currywurst}
...
\end{recipe}
```

`\graph` It is possible to set 2 pictures by the macro `graph`. They are displayed above the other content. If no pictures are set, the space is used by the other content. The `big` picture should be bigger than the `small` picture ;) but it also possible to modify this behaviour. The reason for this is that the widths of the pictures are predefined by the command `setRecipeLengths`.

Example (simple):

```
\graph{
    small = strawberry,
    big = strawberrycake
}
```

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. **Note:** These values are one only used for one recipe.

Example (extended):

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

`\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.

Example:

```
\ingredients{
    2 Tafeln & dunkle Schokolade (ï¿½ber \unit[70]{\%})\\
    3        & Eier\\
    \unit[200]{ml} & Sahne\\
    \unit[40]{g} & Zucker\\
    \unit[50]{g} & Butter
}
```

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 `tabularx` 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 thsi package
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 necessarry for the numeration.

Example:

```
\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!
}
```

`\portion` This macro is intented as an interface for an equal formating of portions entries. For this it can be easily modified. You just have to renew this command.

`\bakingtemperature` This macro allows you to enter some bakingtemperatures. 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 `xkeyval` package.

Example (an entry in options of an recipe):

```
\begin{recipe}
[
    bakingtemperature={\protect\bakingtemperature{fanoven=\unit[230]{ï¿½C},
        topbottomheat=\unit[195]{ï¿½C}, topheat=\unit[195]{ï¿½C}, bottomheat, gasstove=Stufe 2}}
]{Currywurst}
\end{recipe}
```

3.1 Modify design

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

`\setRecipeColors` If you want to change the default colors, you can use `setRecipeColors`. The colors are set by <key = value> as known by the `xkeyval` package. The names of colors can be found in table **??**. New colors can be easiliy defined by the `xcolor` package as shown the example below.

key	default value	description
recipename	green	recipe name
ing	gray	ingredients (text)
inghead	gray	headline of the ingredients
prep	black	preparation (text)
prephead	black	headline of the preparation
hint	red	hint (text)
hinthead	red	headline of the hint
hintline	red	lines leftside the hint text (cross)
numeration	red	numeration of the preparation steps

Table 2: Color Management

Example:

```
\definecolor{webgreen}{rgb}{0,.5,0}

\setRecipeColors{
    recipename = webgreen,
    ing = blue,
    numeration = black,
}
```

`\setRecipeLengths` By using the command `setRecipeLengths` it is possible to change the (default) values for the picture widths and the text column width.

Example:

```
\setRecipeLengths{
    preparationwidth = 0.60\textwidth,
    ingredientswidth = 0.35\textwidth,
    pictureheight = 6cm,
    bigpicturewidth = 0.6\textwidth,
    smallpicturewidth = 0.35\textwidth
}
```

`\setRecipeSizes` The macro `setRecipeSizes` enables you to modify the fontsize of many parts of the recipe. The sizes are inserted by <key = value> (`xkeyval`).

Example (these are the default values):

```
\setRecipeSizes{
    recipename = \fontsize{25pt}{30pt},
    ing = \normalsize,
    inghead = \normalsize,
    prep = \normalsize,
    prephead = \normalsize,
    hint = \normalsize,
    hinthead = \Large
}
```

\setRecipenameFont The name of recipe is displayed with a handwritten font by default. 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/`. The other parameters are fontencoding, fontseries and fontshape. The size is changed by `setRecipeSizes`.

Example (some fonts you could try out, requires option `handwritten`):

```
\setRecipenameFont{
    % pbsi
    % fau
    % fwb
    % fjd    % default when using the option handwritten
    cmr      % default
}{T1}{m}{n}
```

3.2 Translation

\setHeadlines `setHeadlines` allows to modify the default headlines. This is also needed for translations (The default values are german, sorry). The headlines are also inserted in <key = value> form.

Example (a possible english translation):

```
\setHeadlines{
    inghead = Ingredients,
    prephead = Preparation,
    hinthead = Hint,
    calory = energy content,
    continuationhead = Continuation,
    continuationfoot = Continuation on next page
}
```

3.3 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 **??**.
- **myconfig** If you are using this package often or want to keep your main file 'clean' you can define your modfications in the configuration file. If you submit this option the file will be loaded otherwise **not**.

3.4 Background graphic

Inside this package there is only the option to set a background picture using the `setBackgroundPicture` macro. For creating transparent backgrounds I recommand **Inkscape**[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 or stockcenter. To modify these orientations you have the possibility to change it manually by setting the `x` and `y` option (as shown below).

Example:

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

[1] http://inkscape.org/

4 Example

Mousse au Chocolat

1 h
5 Personen
R. Gaus

Zubereitung

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

2 Eigelb in einer grossen Schuessel mit 2 EL heissem Wasser cremig schlagen, den Zucker einruehren 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 Kuehlschrank kalt stellen. Aber nicht zu kalt servieren.

Zutaten

2 Tafeln	dunkle Schokolade (ueber 70 %)
3	Eier
200 ml	Sahne
40 g	Zucker
50 g	Butter

5 Troubleshooting

5.1 MiKTeX

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

5.2 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.

6 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.

6.1 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.

7 Implementation

Let's have a look at the implementation.

7.1 Options

You can use the option `myconfig` to load your personal configuration (file). See section **??**

```
1 ⟨*package⟩
2 \newif\ifHandwrittenFont@required
3 \HandwrittenFont@requiredfalse
4 \DeclareOption{handwritten}
5 {% displays the recipename in a handwitten font
6     \HandwrittenFont@requiredtrue
```

```
    \AtEndOfPackage{
        \setRecipeSizes{recipename = \fontsize{25pt}{30pt}}
        \setRecipenameFont{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

```

7.2 Required Packages

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

```
\RequirePackage{tikz}                       % for creating the lines for the hint
\RequirePackage{graphicx}
\RequirePackage{xcolor}
\RequirePackage[clock, misc, weather]{ifsym} % Symbolpackage 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{ifthen}
\RequirePackage{xkeyval}
\RequirePackage{lettrine}                   % numbering the preparation steps
\RequirePackage{fancyhdr}
\RequirePackage{units}
\RequirePackage{eso-pic}                    % for background pictures
\RequirePackage{picture}                    % for modfifying the position of the bg pictures
\RequirePackage{tabularx}                   % line breaks in tabular

```

7.3 Page Layout

This will set up the page. It is optimized for the documentclass `article`. If you use a other package you probably have to modify this.

Note: I tried to use the KOMA class `scrartcl`, but unfortunately some strange effects occured.

```
\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 `recipesection` allows to modify the entry in the table of contents. **Note:** If you modify the macro you may change the two upper lines as well.

```
\newcommand*{\recipesection}[2][]
{%
    \subsubsection[#1]{#2}
}

```

7.4 Color Management

Below the (default) colors are defined using the `xkeyval` package.

```
\define@key{colorManagement}{recipename}[green]{\def\xcb@color@recipename{#1}}
\define@key{colorManagement}{ing}[gray]{\def\xcb@color@ing{#1}}
\define@key{colorManagement}{inghead}[gray]{\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}{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}
{%
    recipename,   % color of recipename
    ing,          % color of ingredients (text)
    inghead,      % color of headline of ingredients
    prep,         % color of preparation (text)
    prephead,     % color of headline of preparation
    hint,         % color of hint (text)
    hinthead,     % color of headline of hint
    hintline,     % color of lines of hint
    numeration    % color of numeration (preparation)
}

```

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

```
\newcommand*{\setRecipeColors}[1]
{%
    \setkeys{colorManagement}{#1}
}
```

This is the initialisation of all required recipe colors.

```
\setRecipeColors
{% initialisation
    recipename,
    ing,
    inghead,
    prep,
    prephead,
    hint,
    hinthead,
    hintline,
    numeration
}
```

7.5 Size Management

Below the (default) sizes are definded.

```
\define@key{sizeManagement}{recipename}[\fontsize{25pt}{30pt}]{\def\xcb@fontsize@recipename{#1}}
\define@key{sizeManagement}{ing}[\normalsize]{\def\xcb@fontsize@ing{#1}}
\define@key{sizeManagement}{inghead}[\normalsize]{\def\xcb@fontsize@inghead{#1}}
\define@key{sizeManagement}{prep}[\normalsize]{\def\xcb@fontsize@prep{#1}}
\define@key{sizeManagement}{prephead}[\normalsize]{\def\xcb@fontsize@prephead{#1}}
\define@key{sizeManagement}{hint}[\normalsize]{\def\xcb@fontsize@hint{#1}}
\define@key{sizeManagement}{hinthead}[\Large]{\def\xcb@fontsize@hinthead{#1}}
\savekeys{sizeManagement}
{
    recipename,
    ing,
    inghead,
    prep,
    prephead,
    hint,
    hinthead
}
```

`\setRecipeSizes` 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
{% initialisation
    recipename,
    ing,
    inghead,
    prep,
    prephead,
    hint,
    hinthead
}

```

\setRecipenameFont This macro sets the font for the name of the recipe. You have to insert the fontfamily, fontencoding, fontseries and fontshape. The size is set by the `sizeManagement`

```
%% Handwritten fonts: hlce, pbsi, hlcw, fjd, fau
%% Other fonts: fwb
\newcommand*{\setRecipenameFont}[4]
{% sets the font for the recipename
    \def\xcb@font@recipename
    {
        \fontfamily{#1}
        \fontencoding{#2}
        \fontseries{#3} %x1
        \fontshape{#4}
        \xcb@fontsize@recipename
        \selectfont
    }
}
```

This is the initialisation of the font for the recipename. The default is Computer Modern, because it is available on every LaTeX distribution.

```
\setRecipenameFont{cmr}{T1}{b}{n}

```

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}
}

```

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

```
\newcommand{\step}
{%
    \lettrine
    [%
```

```
        lines=2,
        lhang=0,          % space into margin, value between 0 and 1
        loversize=0.15,   % enlarges the height of the capital
        slope=0em,
        findent=1em,      % gap between capital and intended text
        nindent=0em       % shifts all intended lines, begining with the second line
    ]{\stepcounter{step}\thestep}{}%
}

\define@key{headlines}{inghead}[Ingredients]{
    \def\xcb@name@inghead{\textcolor{\xcb@color@inghead}{\textbf{\xcb@fontsize@inghead{#1}}\ }}
}
\define@key{headlines}{prephead}[Directions]{
    \def\xcb@name@prephead{\textcolor{\xcb@color@prephead}{\textbf{\xcb@fontsize@prephead{#1}}\ }}
}
\define@key{headlines}{hinthead}[Tipp:]{
    \def\xcb@name@hinthead{\textcolor{\xcb@color@hinthead}{\xcb@fontsize@hinthead{#1}}}
}
\define@key{headlines}{continuationhead}[Fortsetzung]{\def\xcb@name@continuationhead{#1}}
\define@key{headlines}{continuationfoot}[Fortsetzung auf n\"achster Seite]{\def\xcb@name@continuationfoot{#
\define@key{headlines}{preparationtime}[\fontsize{14.5pt}{10pt}\Gloves]{\def\xcb@name@preparationtime{#1}}
\define@key{headlines}{bakingtime}[\fontsize{14.5pt}{10pt}\Oven]{\def\xcb@name@bakingtime{#1}}
\define@key{headlines}{bakingtemperature}[\Thermo{4}]{\def\xcb@name@bakingtemperature{#1}}
\define@key{headlines}{portion}[\fontsize{14.5pt}{10pt}\Dish]{\def\xcb@name@portion{#1}}
\define@key{headlines}{portionvalue}[Portions]{\def\xcb@name@portionvalue{#1}}
\define@key{headlines}{calory}[Brennwert]{\def\xcb@name@calory{#1}}
\define@key{headlines}{source}[\PaperPortrait]{\def\xcb@name@source{#1}}
\savekeys{headlines}
{
    inghead,
    prephead,
    hinthead,
    continuationhead,
    continuationfoot,
    preparationtime,
    bakingtime,
    bakingtemperature,
    portion,
    portionvalue,
    calory,
    source
}
```

`\setHeadlines` This allows you to modify the headlines for the recipe parts. This also needed for translations (The default values are german, sorry). The parameter are set in the form "key = value" as known by the `xkeyval` package.

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

\setHeadlines
{% initialisation
    inghead,
    prephead,
```

```
    hinthead,
    continuationhead,
    continuationfoot,
    preparationtime,
    bakingtime,
    bakingtemperature,
    portion,
    portionvalue,
    calory,
    source
}
```

The values xcb@bigpicturewidth and xcb@smallpicturewidth are the default values. They can be changed with the `graph` command.

```
\define@key{recipelengths}{preparationwidth}[0.60\textwidth]{\def\xcb@preparationwidth{#1}}
\define@key{recipelengths}{ingredientswidth}[0.35\textwidth]{\def\xcb@ingredientswidth{#1}}
\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}}
\savekeys{recipelengths}
{
    preparationwidth,
    ingredientswidth,
    pictureheight,
    bigpicturewidth,
    smallpicturewidth
}
```

`\setRecipeLengths` This command allows to modify most sizes of the recipe. The `pictureheight` is the height of the two pictures above the text.

```
\newcommand*{\setRecipeLengths}[1]
{%
    \setkeys{recipelengths}{#1}
}

\setRecipeLengths
{% initialisation
    preparationwidth,
    ingredientswidth,
    pictureheight,
    bigpicturewidth,
    smallpicturewidth
}

\define@key{picture}{small}[]{
    \def\xcb@picture@small{
        \ifthenelse{\equal{#1}{}}
        {% not set => not shown
        }
        {
            \vspace{-0.6cm}
            \includegraphics[width=\linewidth, height = \xcb@pictureheight, keepaspectratio]{#1}
```

```
        }
    }
}
\define@key{picture}{smallpicturewidth}[]{\def\xcb@picture@smallwidth{#1}}
\define@key{picture}{bigpicturewidth}[]{\def\xcb@picture@bigwidth{#1}}
\define@key{picture}{big}[]{
    \def\xcb@picture@big{
        \ifthenelse{\equal{#1}{}}
        {% not set => not shown
        }
        {
            \vspace{-0.6cm}
            \includegraphics[width=\linewidth, height = \xcb@pictureheight, keepaspectratio]{#1}
        }
    }
}

\define@key{recipe}{preparationtime}[]{\def\xcb@preparationtime{#1}}
\define@key{recipe}{bakingtime}[]{\def\xcb@bakingtime{#1}}
\define@key{recipe}{bakingtemperature}[]{\def\xcb@bakingtemperature{#1}}
\define@key{recipe}{portion}[]{\def\xcb@portion{#1}}
\define@key{recipe}{calory}[]{\def\xcb@calory{#1}}
\define@key{recipe}{source}[]{\def\xcb@source{#1}}

```

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

```
\newcommand*{\portion}[2][\xcb@name@portionvalue]{\unit[#2]{#1}}

```

`\bakingtemperature` This macro is used for inserting different bakingtemperatures. Only entered temperatures are displayed (with a symbol from `cookingsymbols` package). You can change the order of the temperatures by renew this command.

```
\define@key{bakingtemperature}{fanoven}[]{\def\xcb@bakingtemperature@fanoven{#1}}
\define@key{bakingtemperature}{topbottomheat}[]{\def\xcb@bakingtemperature@topbottomheat{#1}}
\define@key{bakingtemperature}{topheat}[]{\def\xcb@bakingtemperature@topheat{#1}}
\define@key{bakingtemperature}{bottomheat}[]{\def\xcb@bakingtemperature@bottomheat{#1}}
\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}{}}%
    {}%
    {%
        \xcb@bakingtemperature@fanoven \ {\fontsize{14.5pt}{10pt}\Fanoven}%
        \setboolean{xcb@bakingtemperature@firstentry}{false}%
    }%
    \ifthenelse{\equal{\xcb@bakingtemperature@topbottomheat}{}}%
    {}%
    {%
```

```
316         \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentr
317         \xcb@bakingtemperature@topbottomheat \ {\fontsize{14.5pt}{10pt}\Topbottomheat}%
318     }%
319     \ifthenelse{\equal{\xcb@bakingtemperature@topheat}{}}%
320     {}%
321     {%
322         \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentr
323         \xcb@bakingtemperature@topheat \ {\fontsize{14.5pt}{10pt}\Topheat}%
324     }%
325     \ifthenelse{\equal{\xcb@bakingtemperature@bottomheat}{}}%
326     {}%
327     {%
328         \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentr
329         \xcb@bakingtemperature@bottomheat \ {\fontsize{14.5pt}{10pt}\Bottomheat}%
330     }%
331     \ifthenelse{\equal{\xcb@bakingtemperature@gasstove}{}}%
332     {}%
333     {%
334         \ifthenelse{\boolean{xcb@bakingtemperature@firstentry}}{\setboolean{xcb@bakingtemperature@firstentr
335         \xcb@bakingtemperature@gasstove \ {\fontsize{14.5pt}{10pt}\Gasstove}%
336     }%
337 }
```

7.6 Recipe Environment

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

```
338 \newenvironment{recipe}[2][]
339 {
340     %% macros
341     \newcommand*{\graph}[1]
342     {%
343         \setkeys{picture}{##1}
344     }
345
346     \newcommand*{\ingredients}[1]
347     {%
348         \def\xcb@ingredients
349         {%
350             \vspace{-1em}    % same height of ingredients and preparation
351             \xcb@name@inghead
352             %\\[4pt]
353             \\[1em]
354             \xcb@fontsize@ing\color{\xcb@color@ing}
355             \begin{tabularx}{\xcb@ingredientswidth}{rX}
356                 ##1
357             \end{tabularx}
358         }
359     }
360
361     \newcommand*{\preparation}[1]
362     {%
363         \def\xcb@preparation
364         {%
```

```
365             \xcb@name@prephead
366             %\\[4pt]
367             \xcb@fontsize@prep\color{\xcb@color@prep}##1
368         }
369         \setcounter{step}{0}
370     }
371
372     \newcommand*{\hint}[1]
373     {%
374         \def\xcb@hint
375         {%
376             \pagebreak[1]    % begin here a new page, if the space is small
377             \begin{tikzpicture}[line width = 1.5pt, \xcb@color@hintline]
378                 \useasboundingbox (0,0) -- ++(right:3.5cm) -- ++(up:3mm) -- ++(left:3.5cm) -- cycle;
379                 \draw (-3mm, 0mm) -- ++(right:3.5cm);
380                 \draw (0, 3mm) -- ++(down:2.4cm);
381             \end{tikzpicture}
382
383             \nopagebreak
384
385             \begingroup      % a group for the hint
386                 \leftskip1em
387                 \xcb@name@hinthead
388                 \nopagebreak
389
390                 \itshape\textcolor{\xcb@color@hint}{\xcb@fontsize@hint{##1}}
391                 \par
392             \endgroup
393         }
394     }
395
396     \newcommand{\xcb@cmd@recipeoverview}
397     {
398         \begin{tabular}{rl}
399             \ifthenelse{\equal{\xcb@preparationtime}{}}
400             {% not set => not shown
401             }
402             {
403                 \xcb@name@preparationtime & \xcb@preparationtime\\
404             }
405             \ifthenelse{\equal{\xcb@bakingtime}{}}
406             {% not set => not shown
407             }
408             {
409                 \xcb@name@bakingtime & \xcb@bakingtime\ifthenelse{\equal{\xcb@bakingtemperature}{}}
410                 {
411                     \\
412                 }
413                 {,\hspace{1em} \xcb@name@bakingtemperature \hspace{1ex} \xcb@bakingtemperature\\
414                 }
415             }
416             \ifthenelse{\equal{\xcb@portion}{}}
417             {% not set => not shown
418             }
419             {
420                 \xcb@name@portion & \xcb@portion\\
421             }
422             \ifthenelse{\equal{\xcb@calory}{}}
```

```
            {% not set => not shown
            }
            {
                \xcb@name@calory & \xcb@calory\\
            }
            \ifthenelse{\equal{\xcb@source}{}}
            {% not set => not shown
            }
            {
                \xcb@name@source & \xcb@source
            }
        \end{tabular}
    }
```

The following command is needed, if more ingredients than preparation (steps) are inserted. In this case the `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 http://mizine.de/latex/wrapfigure-austricksen/

```
    \newcommand\xcb@cmd@wrapfill
    {%
        \par
        \ifx\parshape\WF@fudgeparshape
        \nobreak
        \vskip-\baselineskip
        \vskip\c@WF@wrappedlines\baselineskip
        \allowbreak
        \WFclear
        \fi
    }
    %% end of macros


    % initialisation
    \setkeys{recipe}{preparationtime, bakingtime, bakingtemperature, portion, calory, source}
    \setkeys{picture}{small, big, smallpicturewidth=\xcb@smallpicturewidth, bigpicturewidth=\xcb@bigpicture
    \ingredients{}
    \preparation{}
    \def\xcb@hint{}

    \def\xcb@recipename{#2}
    \setkeys{recipe}{#1}  % reading the optional parameters

    \setcounter{xcb@newpagefoot}{1}
    \setcounter{xcb@newpagehead}{\value{page}}
}
{% this part is executed at \end{recipe}
%% FIRST BLOCK
    \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

%% SECOND BLOCK
    \recipesection[\normalsize\xcb@recipename]%
    {\hspace{-1em}\textcolor{\xcb@color@recipename}{\xcb@font@recipename\xcb@recipename}}
    \xcb@cmd@recipeoverview

%% THIRD BLOCK
    \begin{wraptable}{r}{\xcb@ingredientswidth}
    {
        \xcb@ingredients
    }
    \end{wraptable}
    \xcb@preparation

    \vfill

    \xcb@cmd@wrapfill
    \xcb@hint
    \setcounter{xcb@newpagefoot}{0}
}

```

7.7 Header and Footer

```
\pagestyle{fancy}
\renewcommand{\sectionmark}[1]
{%
    \markright{\MakeUppercase{\thesection.\ #1}}
}
\renewcommand{\headrulewidth}
{%
    0.5pt
}
\fancyhf{}
\if@twoside
    \fancyfoot[LE,RO]{\textbf{\thepage}}
    \fancyhead[LE,RO]{\rightmark}
\else
    \fancyfoot[R]{\textbf{\thepage}}
```

```
    \fancyhead[R]{\rightmark}
\fi
\fancyhead[C]{\@newpagehint{head}}
\fancyfoot[C]{\@newpagehint{foot}}
```

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}
```

`\@newpagehint` This macro is used to create a hint in the head or foot of the page, that the recipe is longer than one page.

```
\newcommand*{\@newpagehint}[1]
{
    \ifthenelse{\equal{#1}{head}}
    {
        \ifnum \value{xcb@newpagehead} < \value{page}
            \ifnum \value{xcb@newpagehead} > 0
                \xcb@name@continuationhead
            \fi
        \fi
    }
    {
        \ifthenelse{\value{xcb@newpagefoot} > 0}
        {
            \xcb@name@continuationfoot
        }
        {% if no message at the bottom is set, there is no need for a message on the next page
            \setcounter{xcb@newpagehead}{0}
        }
    }
}
```

7.8 Background Picture

```
\define@key{background}{width}[\linewidth]{\def\xcb@background@width{#1}}
\define@key{background}{height}[\paperheight]{\def\xcb@background@height{#1}}
\define@key{background}{x}[0cm]{\def\xcb@background@x{#1}}
\define@key{background}{y}[0cm]{\def\xcb@background@y{#1}}
\define@key{background}{orientation}[default]{\def\xcb@background@orientation{#1}}
\savekeys{background}
{%
    width,
    height,
    x,
    y,
    orientation
}
```

`etBackgroundPicture` 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)
            {%
                \parbox[b][\xcb@background@height]{\xcb@background@width}
                {%
                    \vfill
                    \includegraphics[width=\xcb@background@width, height=\xcb@background@height, keepaspect
                    \vfill
                }
            }
        }
        {%
            \ifthenelse{\equal{\xcb@background@orientation}{pagecenter}}
            {% pagecenter
                \AtPageCenter{%
                    \makebox(\xcb@background@x, \xcb@background@y)[c]
                    {%
                        \includegraphics[width=\xcb@background@width, height=\xcb@background@height, keepas
                    }
                }
            }
            {%
                \ifthenelse{\equal{\xcb@background@orientation}{textcenter}}
                {% textcenter
                    \AtTextCenter{%
                        \makebox(\xcb@background@x, \xcb@background@y)[c]
                        {%
                            \includegraphics[width=\xcb@background@width, height=\xcb@background@height, ke
                        }
                    }
                }
                {%
                    \ifthenelse{\equal{\xcb@background@orientation}{stockcenter}}
                    {% stockcenter
                        \AtStockCenter{%
                            \makebox(\xcb@background@x, \xcb@background@y)[c]
                            {%
                                \includegraphics[width=\xcb@background@width, height=\xcb@background@height
                            }
                        }
                    }
                    {% error message
                        \PackageError{xcookybooky}
                        {There is no '\xcb@background@orientation' option for background picture orientatio
                        {The possible options are: default, pagecenter, textcenter or stockcenter}
                    }
                }
            }
```

```
        }

    }
}
⟨/package⟩
```

7.9 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⟩
%% Change Layout
\setRecipeColors
{%
    recipename,
    ing,
    inghead,
    prep,
    prephead,
    hint,
    hinthead,
    hintline,
    numeration
}

\setRecipeSizes
{%
    recipename,
    ing,
    inghead,
    prep,
    prephead,
    hint,
    hinthead
}

\setRecipeLengths
{%
    preparationwidth,
    ingredientswidth,
    pictureheight
}

\setRecipenameFont{fjd}{T1}{m}{n}


%% Translation
\setHeadlines
{%
    inghead,
    prephead,
    hinthead,
    continuationhead,
    continuationfoot,
    preparationtime,
```

```
    bakingtime,
    bakingtemperature,
    portion,
    portionvalue,
    calory,
    source
}
⟨/config⟩
```