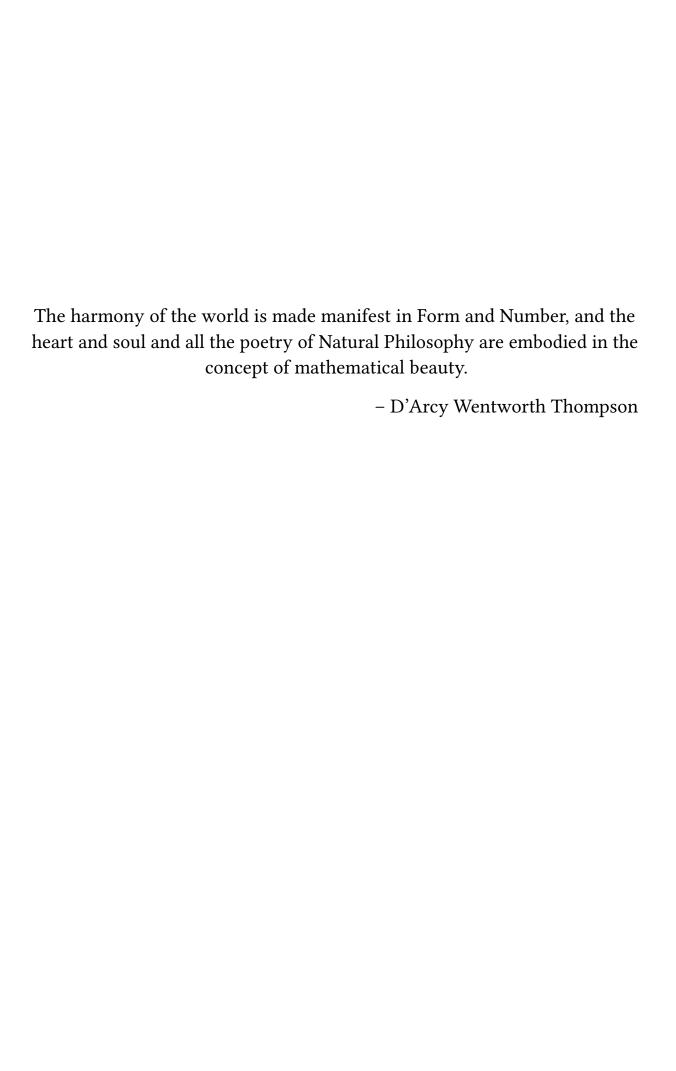**Use this document as a template**

Example and documentation of the kaobook class

Customise this page according to your needs

Federico Marotta[*]

October 1, 2024

An Awesome Publisher

[*] A LaTeX lover

The harmony of the world is made manifest in Form and Number, and the heart and soul and all the poetry of Natural Philosophy are embodied in the concept of mathematical beauty.

– D'Arcy Wentworth Thompson

Preface

I am of the opinion that every LaTeX geek, at least once during his life, feels the need to create his or her own class: this is what happened to me and here is the result, which, however, should be seen as a work still in progress. Actually, this class is not completely original, but it is a blend of all the best ideas that I have found in a number of guides, tutorials, blogs and tex.stackexchange.com posts. In particular, the main ideas come from two sources:

- Ken Arroyo Ohori's Doctoral Thesis, which served, with the author's permission, as a backbone for the implementation of this class;
- The Tufte-Latex Class, which was a model for the style.

The first chapter of this book is introductory and covers the most essential features of the class. Next, there is a bunch of chapters devoted to all the commands and environments that you may use in writing a book; in particular, it will be explained how to add notes, figures and tables, and references. The second part deals with the page layout and design, as well as additional features like coloured boxes and theorem environments.

I started writing this class as an experiment, and as such it should be regarded. Since it has always been intended for my personal use, it may not be perfect but I find it quite satisfactory for the use I want to make of it. I share this work in the hope that someone might find here the inspiration for writing his or her own class.

Federico Marotta

Contents

List of Figures

List of Tables

List of Listings

Introduction | 1

1.1 The Main Ideas

Many modern printed textbooks have adopted a layout with prominent margins where small figures, tables, remarks and just about everything else can be displayed. Arguably, this layout helps to organise the discussion by separating the main text from the ancillary material, which at the same time is very close to the point in the text where it is referenced.

This document does not aim to be an apology of wide margins, for there are many better suited authors for this task; the purpose of all these words is just to fill the space so that the reader can see how a book written with the kaobook class looks like. Meanwhile, I shall also try to illustrate the features of the class.

The main ideas behind kaobook come from this blog post, and actually the name of the class is dedicated to the author of the post, Ken Arroyo Ohori, which has kindly allowed me to create a class based on his thesis. Therefore, if you want to know more reasons to prefer a 1.5-column layout for your books, be sure to read his blog post.

Another source of inspiration, as you may have noticed, is the Tufte-Latex Class. The fact that the design is similar is due to the fact that it is very difficult to improve something which is already so good. However, I like to think that this class is more flexible than Tufte-Latex. For instance, I have tried to use only standard packages and to implement as little as possible from scratch;[1] therefore, it should be pretty easy to customise anything, provided that you read the documentation of the package that provides that feature.

[1]: This also means that understanding and contributing to the class development is made easier. Indeed, many things still need to be improved, so if you are interested, check out the repository on github!

In this book I shall illustrate the main features of the class and provide information about how to use and change things. Let us get started.

1.2 What This Class Does

The kaobook class focuses more about the document structure than about the style. Indeed, it is a well-known LaTeX principle that structure and style should be separated as much as possible (see also Section Section 1.3 on the following page). This means that this class will only provide commands, environments and in general, the opportunity to do things, which the user may or may not use. Actually, some stylistic matters are embedded in the class, but the user is able to customise them with ease.

The main features are the following:

Page Layout The text width is reduced to improve readability and make space for the margins, where any sort of elements can be displayed.
Chapter Headings As opposed to Tufte-Latex, we provide a variety of chapter headings among which to choose; examples will be seen in later chapters.

Page Headers They span the whole page, margins included, and, in twoside mode, display alternatively the chapter and the section name.[2]

2: This is another departure from Tufte's design.

Matters The commands `\frontmatter`, `\mainmatter` and `\backmatter` have been redefined in order to have automatically wide margins in the main matter, and narrow margins in the front and back matters. However, the page style can be changed at any moment, even in the middle of the document.

Margin text We provide commands `\sidenote` and `\marginnote` to put text in the margins.[3]

3: Sidenotes (like this!) are numbered while marginnotes are not

Margin figs/tabs A couple of useful environments is `marginfigure` and `margintable`, which, not surprisingly, allow you to put figures and tables in the margins (*cfr.* Figure 1.1).

Margin toc Finally, since we have wide margins, why don't add a little table of contents in them? See `\margintoc` for that.

Hyperref `hyperref` is loaded and by default we try to add bookmarks in a sensible way; in particular, the bookmarks levels are automatically reset at `\appendix` and `\backmatter`. Moreover, we also provide a small package to ease the hyperreferencing of other parts of the text.

Bibliography We want the reader to be able to know what has been cited without having to go to the end of the document every time, so citations go in the margins as well as at the end, as in Tufte-Latex. Unlike that class, however, you are free to customise the citations as you wish.

The order of the title pages, table of contents and preface can be easily changed, as in any LaTeX document. In addition, the class is based on KOMA-Script's `scrbook`, therefore it inherits all the goodies of that.

Figure 1.1: The Mona Lisa. `https://commons.wikimedia.org/wiki/File:Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF_retouched.jpg`

1.3 What This Class Does Not Do

As anticipated, further customisation of the book is left to the user. Indeed, every book may have sidenotes, margin figures and so on, but each book will have its own fonts, toc style, special environments and so on. For this reason, in addition to the class, we provide only sensible defaults, but if these features are not needed, they can be left out. These special packages are located in the `style` directory, which is organised as follows:

kao.sty This package contains the most important definitions of macros and specifications of page layout. It is the heart of the `kaobook`. Special features include: most commonly used LaTeX packages are already loaded; there is some flexibility to change the default layout; some fancy environments (with coloured boxes around them, floating, and/or with a counter) are predefined.

kaobiblio.sty Where commands to print citations in the margins are defined.[4] It is the `kao-` analog of `biblatex`.

4: See Chapter 5.

kaorefs.sty It contains some useful commands to manage labeling and referencing, again to ensure that the same elements are referenced always in a consistent way.

kaotheorems.sty For the style of mathematical environments, which can be optionally wrapped in a colourful `mdframed` environment, like in this document, or not.

In the rest of the book, I shall assume that the reader is not a novice in the use of LaTeX, and refer to the documentation of the packages used in this class for things that are already explained there. Moreover, I assume that the reader is willing to make minor edits to the provided packages for styles, environments and commands, if he or she does not like the default settings.

The audacious users might feel tempted to edit some of these packages. I'd be immensely happy if they sent me examples of what they have been able to do!

1.4 How to Use This Class

Either if you are using the template from latextemplates, or if you cloned the GitHub repository, there are infinite ways to use the kaobook class in practice, but we will discuss only two of them. The first is to find the `main.tex` file which I used to write this book, and edit it; this will probably involve a lot of text-deleting, copying-and-pasting, and rewriting. The second way is to start almost from scratch and use the `./examples/minimal_book/main.tex` file, which is a cleaned-up version of the `./examples/documentation/main.tex`; even if you choose the second way, you may find it useful to draw inspiration from the `./examples/documentation/main.tex` file.

To compile the document, assuming that its name is `main.tex`, you will have to run the following sequence of commands:

```
pdflatex main # Compile template
makeindex main.nlo -s nomencl.ist -o main.nls # Compile nomenclature
makeindex main # Compile index
biber main # Compile bibliography
makeglossaries main # Compile glossary
pdflatex main # Compile template again
pdflatex main # Compile template again
```

You may need to compile the template some more times in order for some errors to disappear. For any support requests, please ask a question on `tex.stackexchange.org` with the tag "kaobook", open an issue on GitHub, or contact the author via e-mail.

Class Options, Commands and Environments

<h1 style="text-align:right">Class Options | 2</h1>

In this chapter I will describe the most common options used, both the ones inherited from scrbook and the kao-specific ones. Options passed to the class modifies its default behaviour; beware though that some options may lead to unexpected results...

2.1 **KOMA** Options

The kaobook class is based on scrbook, therefore it understands all of the options you would normally pass to that class. If you have a lot of patience, you can read the KOMA-Script guide.[1] Actually, the reading of such guide is suggested as it is very instructive.

Every KOMA-Script option you pass to the class when you load it is automatically activated. In addition, in kaobook some options have modified default values. For instance, the font size is 9.5pt and the paragraphs are separated by space,[2] not marked by indentation.

1: The guide can be downloaded from https://ctan.org/pkg/koma-script?lang=en.

2: To be precise, they are separated by half a line worth of space: the parskip value is "half".

2.2 **kao** Options

In the future I plan to add more options to set the paragraph formatting (justified or ragged) and the position of the margins (inner or outer in twoside mode, left or right in oneside mode).[3]

I take this opportunity to renew the call for help: everyone is encouraged to add features or reimplement existing ones, and to send me the results. You can find the GitHub repository at https://github.com/fmarotta/kaobook.

3: As of now, paragraphs are justified, formatted with \singlespacing (from the setspace package) and \frenchspacing.

> **To Do**
>
> Implement the justified and margin options. To be consistent with the KOMA-Script style, they should accept a simple switch as a parameter, where the simple switch should be true or false, or one of the other standard values for simple switches supported by KOMA-Script. See the KOMA-Script documentation for further information.

The above box is an example of a kaobox, which will be discussed more thoroughly in Chapter 7 (Mathematics and Boxes) on page 27. Throughout the book I shall use these boxes to remarks what still needs to be done.

2.3 **Other Things Worth Knowing**

A bunch of packages are already loaded in the class because they are needed for the implementation. These include:

- ▸ etoolbox
- ▸ calc
- ▸ xifthen
- ▸ xkeyval
- ▸ xparse
- ▸ xstring

Many more packages are loaded, but they will be discussed in due time. Here, we will mention only one more set of packages, needed to change the paragraph formatting (recall that in the future there will be options to change this). In particular, the packages we load are:

- ▸ ragged2e
- ▸ setspace
- ▸ hyphenat
- ▸ microtype
- ▸ needspace
- ▸ xspace
- ▸ xcolor (with options `usenames,dvipsnames`)

Some of the above packages do not concern paragraph formatting, but we nevertheless grouped them with the others. By default, the main text is justified and formatted with singlespacing and frenchspacing; the margin text is the same, except that the font is a bit smaller.

As a last warning, please be aware that the `cleveref` package is not compatible with `kaobook`. You should use the commands discussed in Section 5.3 instead.

2.4 Document Structure

We provide optional arguments to the `\title` and `\author` commands so that you can insert short, plain text versions of this fields, which can be used, typically in the half-title or somewhere else in the front matter, through the commands `\@plaintitle` and `\@plainauthor`, respectively. The PDF properties `pdftitle` and `pdfauthor` are automatically set by hyperref to the plain values if present, otherwise to the normal values.[4]

There are defined two page layouts, `margin` and `wide`, and two page styles, `plain` and `fancy`. The layout basically concern the width of the margins, while the style refers to headers and footer; these issues will be discussed in Chapter 6 (Page Design) on page 22.[5]

The commands `\frontmatter`, `\mainmatter`, and `\backmatter` have been redefined in order to automatically change page layout and style for these sections of the book. The front matter uses the `margin` layout and the `plain` page style. In the mainmatter the margins are wide and the headings are fancy. In the appendix the style and the layout do not change; however we use `\bookmarksetup{startatroot}` so that the bookmarks of the chapters are on the root level (without this, they would be under the preceding part). In the backmatter the margins shrink again and we also reset the bookmarks root.

4: We think that this is an important point so we remark it here. If you compile the document with pdflatex, the PDF metadata will be altered so that they match the plain title and author you have specified; if you did not specify them, the metadata will be set to the normal title and author.

5: For now, suffice it to say that pages with the `margin` layout have wide margins, while with the `wide` layout the margins are absent. In `plain` pages the headers and footer are suppressed, while in `fancy` pages there is a header.

Margin Stuff | 3

Sidenotes are a distinctive feature of all 1.5-column-layout books. Indeed, having wide margins means that some material can be displayed there. We use margins for all kind of stuff: sidenotes, marginnotes, small tables of contents, citations, and, why not?, special boxes and environments.

3.1 Sidenotes

Sidenotes are like footnotes, except that they go in the margin, where they are more readable. To insert a sidenote, just use the command `\sidenote{Text of the note}`. You can specify a mark[O] with `\sidenote[mark]{Text}`, but you can also specify an offset, which moves the sidenote upwards or downwards, so that the full syntax is:

O: This sidenote has a special mark, a big O!

`\sidenote[mark][offset]{Text}`

If you use an offset, you always have to add the brackets for the mark, but they can be empty.[1]

1: If you want to know more about the usage of the `\sidenote` command, read the documentation of the `sidenotes` package.

In `kaobook` we copied a feature from the `snotez` package: the possibility to specify a multiple of `\baselineskip` as an offset. For example, if you want to enter a sidenote with the normal mark and move it upwards one line, type:

`\sidenote[][*-1]{Text of the sidenote.}`

As we said, sidenotes are handled through the `sidenotes` package, which in turn relies on the `marginnote` package.

3.2 Marginnotes

This command is very similar to the previous one. You can create a margin-note with `\marginnote[offset]{Text}`, where the offset argument can be left out, or it can be a multiple of `\baselineskip`, *e.g.*

While the command for margin notes comes from the `marginnote` package, it has been redefined in order to change the position of the optional offset argument, which now precedes the text of the note, whereas in the original version it was at the end. We have also added the possibility to use a multiple of `\baselineskip` as offset. These things were made only to make everything more consistent, so that you have to remember less things!

`\marginnote[-12pt]{Text}` or `\marginnote[*-3]{Text}`

> **To Do**
>
> A small thing that needs to be done is to renew the `\sidenote` command so that it takes only one optional argument, the offset. The special mark argument can go somewhere else. In other words, we want the syntax of `\sidenote` to resemble that of `\marginnote`.

We load the packages `marginnote`, `marginfix` and `placeins`. Since `sidenotes` uses `marginnote`, what we said for marginnotes is also valid for sidenotes. Side- and margin- notes are shifted slightly upwards (`\renewcommand{\marginnotevadjust}{3pt}`) in order to align them to the bottom of the line of text where the note is issued. Importantly, both sidenotes and marginnotes are defined as floating if the optional

argument (*i.e.* the vertical offset) is left blank, but if the offset is specified they are not floating. Recall that floats cannot be nested, so in some rare cases you may encounter errors about lost floats; in those cases, remember that sidenotes and marginnotes are floats. To solve the problem, it may be possible to transform them into non-floating elements by specifying an offset of 0pt.

3.3 Footnotes

Even though they are not displayed in the margin, we will discuss about footnotes here, since sidenotes are mainly intended to be a replacement of them. Footnotes force the reader to constantly move from one area of the page to the other. Arguably, marginnotes solve this issue, so you should not use footnotes. Nevertheless, for completeness, we have left the standard command \footnote, just in case you want to put a footnote once in a while.*

3.4 Margintoc

Since we are talking about margins, we introduce here the \margintoc command, which allows one to put small table of contents in the margin. Like other commands we have discussed, \margintoc accepts a parameter for the vertical offset, like so: \margintoc[offset].

The command can be used in any point of the document, but we think it makes sense to use it just at the beginning of chapters or parts. In this document I make use of a KOMA-Script feature and put it in the chapter preamble, with the following code:

The font used in the margintoc is the same as the one for the chapter entries in the main table of contents at the beginning of the document.

```
\setchapterpreamble[u]{\margintoc}
\chapter{Chapter title}
```

As the space in the margin is a valuable resource, there is the possibility to print a shorter version of the title in the margin toc. Thus, there are in total three possible versions for the title of a section (or subsection): the one for the main text, the one for the main table of contents, and the one for the margintoc. These versions can be specified at the same time when the section is created in the source TEXfile:

```
\section[alternative-title-for-toc]{title-as-written-in-text}[
    alternative-title-for-margintoc]
```

By default, the margintoc includes sections and subsections. If you only want to show sections, add

```
\setcounter{margintocdepth}{\sectiontocdepth}
```

somewhere in your preamble.

---

* And this is how they look like. Notice that in the PDF file there is a back reference to the text; pretty cool, uh?

3.5 Marginlisting

On some occasions it may happen that you have a very short piece of code that doesn't look good in the body of the text because it breaks the flow of narration: for that occasions, you can use a `marginlisting`. The support for this feature is still limited, especially for the captions, but you can try the following code:

Listing 3.1: An example of a margin listing.

```python
print("Hello World!")
```

```
\begin{marginlisting}[-0.5cm]
 \caption{My caption}
 \vspace{0.2cm}
 \begin{lstlisting}[language=Python,style=kaolstplain]
 ... code ...
 \end{lstlisting}
\end{marginlisting}
```

Unfortunately, the space between the caption and the listing must be adjusted manually; if you find a better way, please let me know.

Not only textual stuff can be displayed in the margin, but also figures. Those will be the focus of the next chapter.

4 Figures and Tables

4.1 Normal Figures and Tables

Figures and tables can be inserted just like in any standard LaTeX document. The graphicx package is already loaded and configured in such a way that the figure width is equal to the textwidth and the height is adjusted in order to maintain the original aspect ratio. As you may have imagined, the captions will be positioned...well, in the margins. This is achieved with the help of the floatrow package.

Here is a picture of Mona Lisa (Figure 4.1), as an example. The captions are formatted as the margin- and the side-notes; If you want to change something about captions you can use the command \captsetup from the caption package. Remember that if you want to reference a figure, the label must come *after* the caption!

While the format of the caption is managed by caption, its position is handled by the floatrow package. Achieving this result has been quite hard, but now I am pretty satisfied. In two-side mode, the captions are printed in the correct margin.

Tables can be inserted just as easily as figures, as exemplified by the following code:

Listing 4.1: Caption of a listing.

```
1  \begin{table}
2  \begin{tabular}{ c c c c }
3      \toprule
4      col1 & col2 & col3 & col 4 \\
5      \midrule
6      \multirow{3}{4em}{Multiple row} & cell2 & cell3 & cell4\\ &
7      cell5 & cell6 & cell7 \\ &
8      cell8 & cell9 & cell10 \\
9      \multirow{3}{4em}{Multiple row} & cell2 & cell3 & cell4 \\
       &
```

```
10   cell5 & cell6 & cell7 \\ &
11   cell8 & cell9 & cell10 \\
12   \bottomrule
13 \end{tabular}
14 \end{table}
```

which results in the useless Table Table 4.1.

col1	col2	col3	col 4
Multiple row	cell2	cell3	cell4
	cell5	cell6	cell7
	cell8	cell9	cell10
Multiple row	cell2	cell3	cell4
	cell5	cell6	cell7
	cell8	cell9	cell10

Table 4.1: A useless table.

I don't have much else to say, so I will just insert some blind text. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Figure 4.1: It's Mona Lisa again. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2 Margin Figures and Tables

Marginfigures can be inserted with the environment `marginfigure`. In this case, the whole picture is confined to the margin and the caption is below it. Figure 1.1 is obtained with something like this:

```
1 \begin{marginfigure}
2     \includegraphics{monalisa}
3     \caption[The Mona Lisa]{The Mona Lisa.}
4     \labfig{marginmonalisa}
5 \end{marginfigure}
```

There is also the `margintable` environment, of which Table 4.2 is an example. Notice how you can place the caption above the table by just placing the `\caption` command before beginning the `tabular` environment. Usually, figure captions are below, while table captions are above. This rule is also respected for normal figures and tables: the captions are always on the side, but for figure they are aligned to the bottom, while for tables to the top.

Marginfigures and tables can be positioned with an optional offset command, like so:

Table 4.2: Another useless table.

col1	col2	col3
Multiple row	cell2	cell3
	cell5	cell6
	cell8	cell9

```
1 \begin{marginfigure}[offset]
2     \includegraphics{seaside}
3 \end{marginfigure}
```

Offset ca be either a measure or a multiple of `\baselineskip`, much like with `\sidenote`, `\marginnote` and `\margintoc`. If you are wondering how I inserted this orange bubble, have a look at the `todo` package.

> Improve this part.

4.3 Wide Figures and Tables

With the environments `figure*` and `table*` you can insert figures which span the whole page width. For example, here are a wide figure and a wide table.

Figure 4.2: A wide seaside, and a wide caption. Credits: By Bushra Feroz, CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=68724647`

Table 4.3: A wide table with invented data about three people living in the UK. Note that wide figures and tables are centered and their caption also extends into the margin.

Name	Surname	Job	Salary	Age	Height	Country
Alice	Red	Writer	4.000 £	34	167 cm	England
Bob	White	Bartender	2.000 £	24	180 cm	Scotland
Drake	Green	Scientist	4.000 £	26	175 cm	Wales

It is the user's responsibility to adjust the width of the table, if necessary, until it is aesthetically pleasing. The previous table was obtained with the following code:

Listing 4.3: How to typeset a wide table

```
\begin{table*}[h!]
    \caption{A wide table with invented data about three people
     living in the UK. Note that wide figures and tables are
    centered and their caption also extends into the margin.}
    \begin{tabular}{p{2.0cm} p{2.0cm} p{2.0cm} p{2.0cm} p{2.0cm
    } p{2.0cm} p{1.5cm}}
        \toprule
        Name    & Surname  & Job       & Salary         &
    Age   & Height    & Country \\
        \midrule
        Alice   & Red      & Writer    & 4.000 \pounds    & 34
        & 167 cm     & England \\
        Bob     & White    & Bartender & 2.000 \pounds    & 24
        & 180 cm     & Scotland \\
        Drake   & Green    & Scientist & 4.000 \pounds    & 26
        & 175 cm     & Wales \\
        \bottomrule
    \end{tabular}
\end{table*}
```

The floatrow package provides the "H" specifier to instruct LATEXto position the figure (or table) in precisely the same position it occupies in the source code. However, this specifier does not work with wide figures or tables: you should use "h!" instead, like so: \begin{figure*}[h!].

You may have noticed the full width image at the very beginning of this chapter: that, however, is set up in an entirely different way, which you'll read about in Chapter 6 on page 22.

kaobook also supports paginated tables (have a look at the longtable package). The longtable[1] environment behaves a bit differently from table, in that longtable encompasses both table and tabular, so that you can write, *e.g.*,

1: Interestingly, longtables may require up to four rounds of compilation before they are typeset correctly.

Listing 4.4: Example of a longtable

```
\begin{longtable}{|l c c|}
    \hline
    One & Two & Three \\
    Left & Center & Center \\
    \hline
    \caption{Caption of the longtable.}
\end{longtable}
```

to obtain the following table:

| One | Two | Three |

Left	Center	Center

Table 4.4: Caption of the longtable.

The caption of a `longtable` is always positioned below the table, and it has the same width as the text (it doesn't extend into the margin). However, sometimes you may need a `longtable` that is so wide that it trespass into the margins; in those cases, you may want to also increase the width of the caption. To do so, you'll have to write two additional commands, one before and one after the `longtable`:

Listing 4.5: Increasing the width of the caption of a `longtable`.

```
\floatsetup[longtable]{margins=centering,LTcapwidth=table} %
    Add this line before the longtable to increase the caption
    width
\begin{longtable}{lp{8cm}p{5cm}p{2cm}}
...
\end{longtable}
\floatsetup[longtable]{margins=raggedright,LTcapwidth=\
    textwidth} % Add this line after the longtable to revert
    the previous change
```

Having seen figures and tables, it is now time to tackle hyperreferences.

References | 5

5.1 Citations

To cite someone [1, 2] is very simple: just use the `\sidecite` command. It does not have an offset argument yet, but it probably will in the future. This command supports multiple entries, as you can see, and by default it prints the reference on the margin as well as adding it to the bibliography at the end of the document. Note that the citations have nothing to do with the text,[2] but they are completely random as they only serve the purpose to illustrate the feature.

[1]: Visscher et al. (2008), *Heritability in the genomics era–concepts and misconceptions.*
[2]: James et al. (2013), *An Introduction to Statistical Learning*

[2]: James et al. (2013), *An Introduction to Statistical Learning*

For this setup I wrote a separate package, `kaobiblio`, which you can find in the `styles` directory and include in your main tex file. This package accepts all the options that you can pass to `biblatex`, and actually it passes them to `biblatex` under the hood. Moreover, it also defines some commands, like `\sidecite`, and environments that can be used within a kao book.[1]

1: For this reason you should always use `kaobiblio` instead of `biblatex`, but the syntax and the options are exactly the same.

If you want to use `bibtex` instead of `biblatex`, pass the option `backend=bibtex` to `kaobiblio`. `kaobiblio` also supports two options that are not shared with `biblatex`: `addspace` and `linkeverything`, both of which are boolean options, meaning that they can take either "true" or "false" as a value. If you pass `addspace=true` when loading `kaobiblio`, a space will be automatically added before the citation marks. If you pass `linkeverything=true`, the author's name in the authoryear-* and authortitle-* styles will be a hyperlink like the year.[2]

2: The fact that the author name is not a hyperlink bothers more than one biblatex user. There are strong arguments *against* hyperlinking the author name, but in my personal opinion, linking the author's name does not result in any problems in most practical cases.

As you have seen, the `\sidecite` command will print a citation in the margin. However, this command would be useless without a way to customise the format of the citation, so the `kaobook` provides also the `\formatmargincitation` command. By "renewing" that command, you can choose which items will be printed in the margins. The best way to understand how it works is to see the actual definition of this command.

```
\newcommand{\formatmargincitation}[1]{%
    \parencite{#1}: \citeauthor*{#1} (\citeyear{#1}), \citetitle{#1}%
}
```

Thus, the `\formatmargincitation` accepts one parameter, which is the citation key, and prints the parencite followed by a colon, then the author, then the year (in brackets), and finally the title.[3] Now, suppose that you wish the margin citation to display the year and the author, followed by the title, and finally a fixed arbitrary string; you would add to your document:

[3]: Battle et al. (2014), *Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individuals*

```
\renewcommand{\formatmargincitation}[1]{%
    \citeyear{#1}, \citeauthor*{#1}: \citetitle{#1}; very interesting!%
}
```

The above code results in citations that look like the following.[4] Of course, changing the format is most useful when you also change the default bibliography style. For instance, if you want to use the "philosophy-modern" style for your bibliography, you might have something like this in the preamble:

2005, Zou et al.: "Regularization and variable selection via the elastic-net"; very interesting!

```
\usepackage[style=philosophy-modern]{styles/kaobiblio}
\renewcommand{\formatmargincitation}[1]{%
    \sdcite{#1}%
}
\addbibresource{main.bib}
```

The commands like `\citeyear`, `\parencite` and `\sdcite` are just examples. A full reference of the available commands can be found in this cheatsheet, under the "Citations" section.

Finally, to compile a document containing citations, you need to use an external tool, which for this class is biber. You need to run the following (assuming that your tex file is called main.tex):

```
$ pdflatex main
$ biber main
$ pdflatex main
```

5.2 Glossaries and Indices

The kaobook class loads the packages `glossaries` and `imakeidx`, with which you can add glossaries and indices to your book. For instance, I previously defined some glossary entries and now I am going to use them, like this: computer. `glossaries` also allows you to use acronyms, like the following: this is the full version, Frame per Second (FPS), and this is the short one FPS. These entries will appear in the glossary in the backmatter.

Unless you use Overleaf or some other fancy IDE for LaTeX, you need to run an external command from your terminal in order to compile a document with a glossary. In particular, the commands required are:[3]

```
$ pdflatex main
$ makeglossaries main
$ pdflatex main
```

3: These are the commands you would run in a UNIX system, but see also Section 5.4 (A Final Note on Compilation); I have no idea about how it works in Windows.

Note that you need not run `makeglossaries` every time you compile your document, but only when you change the glossary entries.

To create an index, you need to insert the command `\index{subject}` whenever you are talking about "subject" in the text. For instance, at the start of this paragraph I would write `index{index}`, and an entry would be added to the Index in the backmatter. Check it out!

A nomenclature is just a special kind of index; you can find one at the end of this book. To insert a nomenclature, we use the package `nomencl` and add the terms with the command `\nomenclature`. We put then a `\printnomenclature` where we want it to appear.

In theory, you would need to run an external command for the index as well, but luckily the package we suggested, `imakeidx`, can compile the index automatically.

Also with this package we need to run an external command to compile the document, otherwise the nomenclature will not appear:

```
$ pdflatex main
$ makeindex main.nlo -s nomencl.ist -o main.nls
$ pdflatex main
```

These packages are all loaded in packages.sty, one of the files that come with this class. However, the configuration of the elements is best done in the main.tex file, since each book will have different entries and styles.

Note that the `nomencl` package caused problems when the document was compiled, so, to make a long story short, I had to prevent `scrhack` to load the hack-file for `nomencl`. When compiling the document on Overleaf, however, this problem seem to vanish.

5.3 Hyperreferences

Together with this class we provide a handy package to help you referencing the same elements always in the same way, for consistency across the book. First, you can label each element with a specific command. For instance, should you want to label a chapter, you would put `\labch{chapter-title}` right after the `\chapter` directive. This is just a convenience, because `\labch` is actually just an alias to `\label{ch: chapter-title}`, so it spares you the writing of "ch:". We defined similar commands for many typically labeled elements, including:

- ▸ Page: `\labpage`
- ▸ Part: `\labpart`
- ▸ Chapter: `\labch`
- ▸ Section: `\labsec`
- ▸ Figure: `\labfig`
- ▸ Table: `\labtab`
- ▸ Definition: `\labdef`

- ▸ Assumption: `\labassum`
- ▸ Theorem: `\labthm`
- ▸ Proposition: `\labprop`
- ▸ Lemma: `\lablemma`
- ▸ Remark: `\labremark`
- ▸ Example: `\labexample`
- ▸ Exercise: `\labexercise`

Of course, we have similar commands for referencing those elements. However, since the style of the reference should depend on the context, we provide different commands to reference the same thing. For instance, in some occasions you may want to reference the chapter by name, but other times you want to reference it only by number. In general, there are four reference style, which we call plain, vario, name, and full.

The plain style references only by number. It is accessed, for chapters, with `\refch{chapter-title}` (for other elements, the syntax is analogous). Such a reference results in: Chapter 5.

The vario and name styles rest upon the `varioref` package. Their syntax is `\vrefch{chapter-title}` and `\nrefch{chapter-title}`, and they result in: Chapter 5 on page 16, for the vario style, and: Chapter 5 (References), for the name style. As you can see, the page is referenced in `varioref` style.

The full style references everything. You can use it with `\frefch{chapter -title}` and it looks like this: Chapter 5 (References) on page 16.

Of course, all the other elements have similar commands (*e.g.* for parts you would use `\vrefpart{part-title}` or something like that). However, not all elements implement all the four styles. The commands provided should be enough, but if you want to see what is available or to add the missing ones, have a look at the attached package.

In order to have access to all these features, the `kaorefs` should be loaded in the preamble of your document. It should be loaded last, or at least after `babel` (or `polyglossia`) and `plaintheorems` (or `mdftheorems`).

Options can be passed to it like to any other package; in particular, it is possible to specify the language of the captions. For instance, if you specify "italian" as an option, instead of "Chapter" it will be printed "Capitolo", the Italian analog. If you know other languages, you are welcome to contribute the translations of these captions! Feel free to contact the author of the class for further details.

The `kaorefs` package also include `cleveref`, so it is possible to use `\cref` in addition to all the previously described referencing commands.

5.4 A Final Note on Compilation

Probably the easiest way to compile a latex document is with the `latexmk` script, as it can take care of everything, if properly configured, from the bibliography to the glossary. The command to issue, in general, is:

```
1  latexmk [latexmk_options] [filename ...]
```

`latexmk` can be extensively configured (see https://mg.readthedocs.io/latexmk.html). For convenience, I print here an example configuration that would cover all the steps described above.

```
1   # By default compile only the file called 'main.tex'
2   @default_files = ('main.tex');
3
4   # Compile the glossary and acronyms list (package 'glossaries')
5   add_cus_dep( 'acn', 'acr', 0, 'makeglossaries' );
6   add_cus_dep( 'glo', 'gls', 0, 'makeglossaries' );
7   $clean_ext .= " acr acn alg glo gls glg";
8   sub makeglossaries {
9       my ($base_name, $path) = fileparse( $_[0] );
10      pushd $path;
11      my $return = system "makeglossaries", $base_name;
12      popd;
13      return $return;
14  }
15
16  # Compile the nomenclature (package 'nomencl')
17  add_cus_dep( 'nlo', 'nls', 0, 'makenlo2nls' );
18  sub makenlo2nls {
19      system( "makeindex -s nomencl.ist -o \"$_[0].nls\" \"$_[0].
        nlo\"" );
20  }
```

However, if you'd rather not use an external package and want to do everything manually, here are some tips.[4]

Compiling the examples in the kaobook repository

To compile the examples, and in particular the documentation, that are in the `examples` directory of the kaobook repository on GitHub, do as follows. **cd** into the root directory of the repository, and run `pdflatex -output-directory examples/documentation main.tex`. With this trick, you can compile the documentation using the class files pertaining to the repository (and not, say, those in your texmf tree). The "-output-directory" option works with the other LATEX-related commands such as biber and makeglossaries.

4: As the author only uses Linux and compiles everything from the command line, he doesn't know how the compilation works in Windows or Mac. The tips, therefore, refer to the usage with Linux from the command line.

A note of warning: sometimes LaTeX needs more than one run to get the correct position of each element; this is true in particular for the positioning of floating elements like figures, tables, and margin notes. Occasionally, LaTeX can need up to four re-runs, so If the alignment of margin elements looks odd, or if they bleed into ther main text, try running pdflatex one more time.

Design and Additional Features

6 Page Design

6.1 Headings

So far, in this document I used two different styles for the chapter headings: one has the chapter name, a rule and, in the margin, the chapter number; the other has an image at the top of the page, and the chapter title is printed in a box (like for this chapter). There is one additional style, which I used only in the Chapter 7.3 (Appendix); there, the chapter title is enclosed in two horizontal rules, and the chapter number (or letter, in the case of the appendix) is above it.[1]

Every book is unique, so it makes sense to have different styles from which to choose. Actually, it would be awesome if whenever a kao-user designs a new heading style, he or she added it to the three styles already present, so that it will be available for new users and new books.

The choice of the style is made simple by the \setchapterstyle command. It accepts one option, the name of the style, which can be: "plain", "kao", "bar", or "lines".[2] If instead you want the image style, you have to use the command \setchapterimage, which accepts the path to the image as argument; you can also provide an optional parameter in square brackets to specify the height of the image. \setchapterimage automatically sets the chapter style to "bar" for that chapter (and also for subsequent chapters).

Let us make some examples. In this book, I begin a normal chapter with the lines:

```
1  \setchapterstyle{kao}
2  \setchapterpreamble[u]{\margintoc}
3  \chapter{Title of the Chapter}
4  \labch{title}
```

In Line 1 I choose the style for the title to be "kao". Then, I specify that I want the margin toc. The rest is ordinary administration in LaTeX, except that I use my own \labch to label the chapter. Actually, the \setchapterpreamble is a standard KOMA-Script one, so I invite you to read about it in the KOMA documentation. Once the chapter style is set, it holds until you change it.[3] Whenever I want to start a chapter with

1: To be honest, I do not think that mixing heading styles like this is a wise choice, but in this document I did it only to show you how they look.

2: Plain is the default LaTeX title style; the other ones are self explanatory.

3: The \margintoc has to be specified at every chapter. Perhaps in the future this may change; it all depends on how this feature will be welcomed by the users, so keep in touch with me if you have preferences!

an image, I simply write:

```
1 \setchapterimage[7cm]{path/to/image.png} % Optionally specify
      the height
2 \setchapterpreamble[u]{\margintoc}
3 \chapter{Catchy Title} % No need to set a chapter style
4 \labch{catchy}
```

If you prefer, you can also specify the style at the beginning of the main document, and that style will hold until you change it again.

6.2 Headers & Footers

Headers and footers in KOMA-Script are handled by the `scrlayer-scrpage` package. There are two basic style: "scrheadings" and "plain.scr-headings". The former is used for normal pages, whereas the latter is used in title pages (those where a new chapter starts, for instance) and, at least in this book, in the front matter. At any rate, the style can be changed with the `\pagestyle` command, *e.g.* `\pagestyle{plain.scrheadings}`.

In both styles, the footer is completely empty. In plain.scrheadings, also the header is absent (otherwise it wouldn't be so plain…), but in the normal style the design is reminiscent of the "kao" style for chapter titles.

6.3 `twoside` mode

> **To Do**
>
> The `twoside` class option is still unstable and may lead to unexpected behaviours. Great strides have been done since the first version of kaobook, but some work still needs to be done. As always, any help will be greatly appreciated.

By passing the `twoside` option to the kaobook, the style of left and right pages will be different, similarly to a printed book. In digital books, having a symmetrical layout for left and right pages is less important, and you may be tempted to use the `twoside=false` option. However, keep in mind that in "oneside" mode the `\uppertitleback` and `\lowertitleback` commands are not available.[4] If you want to have the upper/lower title-back in a one-side document, just add manually the contents that you'd put using the upper/lower titleback commands.

4: Another useful thing to keep in mind is that, when `twoside=true`, an extra white page will be added to the front-matter.

6.4 Table of Contents

Another important part of a book is the table of contents. By default, in kaobook there is an entry for everything: list of figures, list of tables, bibliographies, and even the table of contents itself. Not everybody might like this, so we will provide a description of the changes you need to do in order to enable or disable each of these entries. In the following Table 6.1, each item corresponds to a possible entry in the TOC, and its description is the command you need to provide to have such entry.

Entry	Command to Activate
Table of Contents	\setuptoc{toc}{totoc}
List of Figs/Tabs	\PassOptionsToClass{toc=listof}{\@baseclass}
Bibliography	\PassOptionsToClass{toc=bibliography}{\@baseclass}

Table 6.1: Commands to add a particular entry to the table of contents.

These commands are specified in the attached style package,[5] so if you don't want the entries, just comment the corresponding lines.

5: In the same file, you can also choose the titles of these entries.

Of course, some packages, like those for glossaries and indices, will try to add their own entries. In such cases, you have to follow the instructions specific to that package. Here, since we have talked about glossaries and notations in Chapter 5, we will briefly see how to configure them.

In a later section, we will see how you can define your own floating environment, and endow it with an entry in the TOC.

For the `glossaries` package, use the "toc" option when you load it: \ usepackage[toc]{glossaries}. For `nomencl`, pass the "intoc" option at the moment of loading the package. Both `glossaries` and `nomencl` are loaded in the attached "packages" package.

Additional configuration of the table of contents can be performed through the packages `etoc`, which is loaded because it is needed for the margintocs, or the more traditional `tocbase`. Read the respective documentations if you want to be able to change the default TOC style.[6]

6: (And please, send me a copy of what you have done, I'm so curious!)

6.5 Paper Size

Recent versions of Kaobook support paper sizes different from the default A4. It is possible to pass the name of the paper as an option to the class, as we are accustomed for any other LaTeX class. For example, the class option `b5paper` would set the paper size to the B5 format.

We also support the paper sizes specified in this web page and some additional sizes requested by the users, with the option names specified in Table 6.2.

Table 6.2: Some non-standard paper sizes supported by kaobook.

Dimension	Option name
12.0cm x 19.0cm	smallpocketpaper
13.5cm x 21.5cm	pocketpaper
14.8cm x 21.0cm	a5paper
15.5cm x 22.0cm	juvenilepaper
17.0cm x 17.0cm	smallphotopaper
21.0cm x 15.0cm	appendixpaper
17.0cm x 22.0cm	cookpaper
19.0cm x 27.0cm	illustratedpaper
17.0cm x 17.0cm	photopaper
16.0cm x 24.0cm	f24paper

For instance, to use the "smallpocketpaper" add the correct description at the beginning of the documentclass instruction:

```
1  \documentclass[
2          smallpocketpaper,
3          fontsize=10pt,
4          twoside=false,
5          %open=any,
6          secnumdepth=1,
7  ]{kaobook}
```

Sometimes it is convenient to adopt a landscape view; `kaobook` provides two additional options, `a4paperlandscape` and `169paperlandscape`, which set the page in landscape mode with width-to-height ratios of, respectively, 1.414 and 16:9.

6.6 Page Layout

Besides the page style, you can also change the width of the content of a page. This is particularly useful for pages dedicated to part titles,

where having the 1.5-column layout might be a little awkward, or for pages where you only put figures, where it is important to exploit all the available space.

In practice, there are two layouts: "wide" and "margin". The former suppresses the margins and allocates the full page for contents, while the latter is the layout used in most of the pages of this book, including this one. The wide layout is also used automatically in the front and back matters.

To change page layout, use the `\pagelayout` command. For example, when I start a new part, I write:

```
1  \pagelayout{wide}
2  \addpart{Title of the New Part}
3  \pagelayout{margin}
```

Beyond these two basic layouts, it is also possible to finely tune the page layout by redefining the `\marginlayout` command. This command is called internally by the higher-level `\pagelayout`, and it is responsible for setting the width of the margins and of the text. The default definition is:

```
1   \newcommand{\marginlayout}{%
2       \newgeometry{
3           top=27.4mm,              % height of the top margin
4           bottom=27.4mm,           % height of the bottom margin
5           inner=24.8mm,            % width of the inner margin
6           textwidth=107mm,         % width of the text
7           marginparsep=8.2mm,      % width between text and margin
8           marginparwidth=49.4mm,   % width of the margin
9       }%
10  }
```

so if you want to, say, decrease the width of the margin while increasing the width of the text, you could write in the preamble of your document something like:

```
1   \renewcommand{\marginlayout}{%
2       \newgeometry{
3           top=27.4mm,              % height of the top margin
4           bottom=27.4mm,           % height of the bottom margin
5           inner=24.8mm,            % width of the inner margin
6           textwidth=117mm,         % width of the text
7           marginparsep=8.2mm,      % width between text and margin
8           marginparwidth=39.4mm,   % width of the margin
9       }%
10  }
```

where the text width has been increased by 10mm and the margin width has been decreased by 10mm.

6.7 Numbers & Counters

In this short section we shall see how dispositions, sidenotes and figures are numbered in the kaobook class.

Sometimes it is desirable to increase the width for just one or a few paragraphs; the widepar environment does that: wrap your paragraphs in this environment, and they will occupy the full width of the page.

By default, dispositions are numbered up to the section in kaobook and up to the subsection in kaohandt. This can be changed by passing the option secnumdepth to kaobook or kaohandt (e.g. 1 corresponds to section and 2 corresponds to subsections).

The sidenotes counter is the same across all the document, but if you want it to reset at each chapter, just uncomment the line

```
\counterwithin*{sidenote}{chapter}
```

in the styles/style.sty package provided by this class.

Figure and Table numbering is also per-chapter; to change that, use something like:

```
\renewcommand{\thefigure}{\arabic{section}.\arabic{figure}}
```

6.8 White Space

One of the things that I find most hard in LaTeX is to finely tune the white space around objects. There are not fixed rules, each object needs its own adjustment. Here we shall see how some spaces are defined at the moment in this class.

Attention! This section may be incomplete.

Space around sidenotes and citations marks

There should be no space before or after sidenotes and citation marks, like so:

sidenote[7]sidenote
citation[2]citation

7: This paragraph can be used to diagnose any problems: if you see whitespace around sidenotes or citation marks, probably a % sign is missing somewhere in the definitions of the class macros.

Space around figures and tables

```
\renewcommand\FBaskip{.4\topskip}
\renewcommand\FBbskip{\FBaskip}
```

Space around captions

```
\captionsetup{
    aboveskip=6pt,
    belowskip=6pt
}
```

Space around displays (*e. g.* equations)

```
\setlength\abovedisplayskip{6pt plus 2pt minus 4pt}
\setlength\belowdisplayskip{6pt plus 2pt minus 4pt}
\abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
```

Mathematics and Boxes | 7

7.1 Theorems

Despite most people complain at the sight of a book full of equations, mathematics is an important part of many books. Here, we shall illustrate some of the possibilities. We believe that theorems, definitions, remarks and examples should be emphasised with a shaded background; however, the colour should not be to heavy on the eyes, so we have chosen a sort of light yellow.[1]

> **Definition 7.1.1** *Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X, d).*

1: The boxes are all of the same colour here, because we did not want our document to look like Harlequin.

Definition 7.1.1 is very important. I am not joking, but I have inserted this phrase only to show how to reference definitions. The following statement is repeated over and over in different environments.

> **Theorem 7.1.1** *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

> **Proposition 7.1.2** *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

> **Lemma 7.1.3** *A finite intersection[a] of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*
>
> ---
> [a] I'm a footnote

You can even insert footnotes inside the theorem environments; they will be displayed at the bottom of the box.

You can safely ignore the content of the theorems...I assume that if you are interested in having theorems in your book, you already know something about the classical way to add them. These example should just showcase all the things you can do within this class.

> **Corollary 7.1.4** (Finite Intersection, Countable Union) *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

Proof. The proof is left to the reader as a trivial exercise. Hint: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift –

not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. □

Definition 7.1.2 *Let (X,d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x,r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X,d).*

Here is a random equation, just because we can:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$

Example 7.1.1 Let (X,d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x,r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X,d).

Remark 7.1.1 Let (X,d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x,r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X,d).

As you may have noticed, definitions, example and remarks have independent counters; theorems, propositions, lemmas and corollaries share the same counter.

Remark 7.1.2 Here is how an integral looks like inline: $\int_a^b x^2 dx$, and here is the same integral displayed in its own paragraph:

$$\int_a^b x^2 dx$$

There is also an environment for exercises.

Exercise 7.1.1 Prove (or disprove) the Riemann hypothesis.

We provide one package for the theorem styles: kaotheorems.sty, to which you can pass the `framed` option you do want coloured boxes around theorems, like in this document.[2] You may want to edit this files according to your taste and the general style of the book. However, there is an option to customise the background colour of the boxes if you use the `framed` option: when you load this package, you can pass it the `background=mycolour` option (replace "mycolour" with the actual colour, for instance, "red!35!white"). This will change the colour of all the boxes, but it is also possible to override the default colour only for some elements. For instance, the `propositionbackground=mycolour` option will change the colour for propositions only. There are similar options for theorem, definition, lemma, corollary, remark, and example.

2: The styles without `framed` are not showed, but actually the only difference is that they don't have the yellow boxes.

7.2 Boxes & Custom Environments [3]

Say you want to insert a special section, an optional content or just something you want to emphasise. We think that nothing works better than a box in these cases. We used `mdframed` to construct the ones

3: Notice that in the table of contents and in the header, the name of this section is "Boxes & Environments"; we achieved this with the optional argument of the `section` command.

shown below. You can create and modify such environments by editing the provided file kao.sty.

> **Title of the box**
>
> Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

If you set up a counter, you can even create your own numbered environment.

> **Comment 7.2.1**
>
> Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.3 Experiments

It is possible to wrap marginnotes inside boxes, too. Audacious readers are encouraged to try their own experiments and let me know the outcomes.

I believe that many other special things are possible with the kaobook class. During its development, I struggled to keep it as flexible as possible, so that new features could be added without too great an effort. Therefore, I hope that you can find the optimal way to express yourselves in writing a book, report or thesis with this class, and I am eager to see the outcomes of any experiment that you may try.

> **title of margin note**
>
> Margin note inside a kaobox. (Actually, kaobox inside a marginnote!)

APPENDIX

A

Heading on Level 0 (chapter)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1 Heading on Level 1 (section)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1.1 Heading on Level 2 (subsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 3 (subsubsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected

font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 4 (paragraph)   Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.2 Lists

A.2.1 Example for list (itemize)

- First item in a list
- Second item in a list
- Third item in a list
- Fourth item in a list
- Fifth item in a list

Example for list (4*itemize)

- First item in a list
  - First item in a list
    - First item in a list
      - First item in a list
      - Second item in a list
    - Second item in a list
  - Second item in a list
- Second item in a list

A.2.2 Example for list (enumerate)

1. First item in a list
2. Second item in a list
3. Third item in a list
4. Fourth item in a list
5. Fifth item in a list

Example for list (4*enumerate)

1. First item in a list
    a) First item in a list
        i. First item in a list
            A. First item in a list
            B. Second item in a list
        ii. Second item in a list
    b) Second item in a list
2. Second item in a list

A.2.3 Example for list (description)

First item in a list
Second item in a list
Third item in a list
Fourth item in a list
Fifth item in a list

Example for list (4*description)

First item in a list
    First item in a list
        First item in a list
            First item in a list
            Second item in a list
        Second item in a list
    Second item in a list
Second item in a list

B
Fonts Testing

B.1 Font Sizes

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

B.2 Font Families

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

The quick brown fox jumps over the lazy dog. Medium.

The quick brown fox jumps over the lazy dog. Bold.

The quick brown fox jumps over the lazy dog. Upright.

The quick brown fox jumps over the lazy dog. Italics.

The quick brown fox jumps over the lazy dog. Slanted.

The quick brown fox jumps over the lazy dog. Small Caps.

```
Hello, here is some text without a meaning. This text should
show what a printed text will look like at this place. If you
read this text, you will get no information. Really? Is there
no information? Is there a difference between this text and
some nonsense like ``Huardest gefburn''? Kjift -- not at all!
A blind text like this gives you information about the selected
font, how the letters are written and an impression of the
look. This text should contain all letters of the alphabet
and it should be written in of the original language. There
is no need for special content, but the length of words should
match the language.
```

```
The quick brown fox jumps over the lazy dog. Medium.
```

```
The quick brown fox jumps over the lazy dog. Bold.
```

```
The quick brown fox jumps over the lazy dog. Upright.
```

```
The quick brown fox jumps over the lazy dog. Italics.
```

```
The quick brown fox jumps over the lazy dog. Slanted.
```

```
The quick brown fox jumps over the lazy dog. Small Caps.
```

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

The quick brown fox jumps over the lazy dog. Medium.

The quick brown fox jumps over the lazy dog. Bold.

The quick brown fox jumps over the lazy dog. Upright.

The quick brown fox jumps over the lazy dog. Italics.

The quick brown fox jumps over the lazy dog. Slanted.

The quick brown fox jumps over the lazy dog. Small Caps.

Test C

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Bibliography

Here are the references in citation order.

[1]   Peter M Visscher, William G Hill, and Naomi R Wray. "Heritability in the genomics era–concepts and misconceptions." In: *Nat. Rev. Genet.* 9.4 (2008), pp. 255–266. DOI: 10.1038/nrg2322 (cited on page 16).

[2]   Gareth James et al. *An Introduction to Statistical Learning.* 2013 (cited on pages 16, 26).

[3]   Alexis Battle et al. "Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individuals." In: *Genome Res.* 24.1 (2014), pp. 14–24. DOI: 10.1101/gr.155192.113 (cited on page 16).

[4]   Hui Zou and Trevor Hastie. "Regularization and variable selection via the elastic-net." In: *J. R. Stat. Soc.* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x (cited on page 16).

Notation

The next list describes several symbols that will be later used within the body of the document.

c Speed of light in a vacuum inertial frame

h Planck constant

Greek Letters with Pronunciations

Character	Name	Character	Name
α	alpha *AL-fuh*	ν	nu *NEW*
β	beta *BAY-tuh*	ξ, Ξ	xi *KSIGH*
γ, Γ	gamma *GAM-muh*	o	omicron *OM-uh-CRON*
δ, Δ	delta *DEL-tuh*	π, Π	pi *PIE*
ϵ	epsilon *EP-suh-lon*	ρ	rho *ROW*
ζ	zeta *ZAY-tuh*	σ, Σ	sigma *SIG-muh*
η	eta *AY-tuh*	τ	tau *TOW (as in cow)*
θ, Θ	theta *THAY-tuh*	υ, Υ	upsilon *OOP-suh-LON*
ι	iota *eye-OH-tuh*	ϕ, Φ	phi *FEE, or FI (as in hi)*
κ	kappa *KAP-uh*	χ	chi *KI (as in hi)*
λ, Λ	lambda *LAM-duh*	ψ, Ψ	psi *SIGH, or PSIGH*
μ	mu *MEW*	ω, Ω	omega *oh-MAY-guh*

Capitals shown are the ones that differ from Roman capitals.

Special Terms

C

computer is a programmable machine that receives input, stores and manipulates data, and provides output in a useful format. 17

F

FPS Frame per Second. 17

T

TOC Table of Contents. 23, 24

Alphabetical Index