Universidad de Córdoba

Verificación de la inferencia de tipos en el lenguaje de programación `Crystal`

Presentado ante la Facultad de Matemática, Astronomía, Física y Computación, como trabajo de grado

Por: Ramiro Barraco
Director: Dr. Soldevila Raffa, Mallku Ernesto
Director Representante: Dr. Ziliani, Beta

1 de diciembre de 2021

Crystal Lang

'Facts, facts, facts,' cries the scientist if he wants to emphasize the necessity of a firm foundation for science. What is a fact? A fact is a thought that is true. But the scientist will surely not recognize something which depends on men's varying states of mind to be the firm foundation of science.

– Gottlob Frege (1956). "The thought: A logical inquiry"

Resumen

Crystal es un lenguaje de programación orientado a objetos, diseñado y desarrollado por Ary Borenszweig, Juan Wajnerman, Brian Cardiff y más de 300 colaboradores. Con una sintaxis inspirada en el lenguaje Ruby, es un lenguaje compilado con verificación estática de tipos, pero sin la necesidad de especificar tipos de variables o los argumentos de los métodos. Los tipos se resuelven mediante un algoritmo de inferencia de tipos globales. Crystal presentó su versión 1.0.0 este año y sigue en desarrollo con una comunidad sumamente activa, con más de 6000 Shards (librerias de Crystal).

En este trabajo intentaremos formalizar una pequeña parte del lenguaje, en particular la inferencia de tipos. Utilizaremos PLT Redex para modelar Crystal-Clear y la probaremos con una suite de test de nuestra autoría. También usaremos herramientas que Redex ofrece para Random-testing y probaremos propiedades del lenguaje como seguridad.

Formalizar esta semántica puede otorgarle mayor seguridad y confianza a los proyectos desarrollados en Crystal

Abstract

Crystal is a general purpose object oriented programming language, designed and developed by Ary Borenszweig, Juan Wajnerman, Brian Cardiff and more than 300 collaborators. With a syntax inspired in Ruby, is a compiled language with static type-checking, but specifying the types of variables or method arguments is generally unneeded. Types are resolved by an advanced global type inference algorithm. Crystal is currently in active development. It is released as free and open-source software. This year Crystal launched it's 1.0.0 version and has an active community with more than 6000 shards (Crystal libraries).

In this work we will try to formalize a little part of the language, in particular it's type inference. We will model Crystal-Clear using PLT redex and test it with a test suite of our own authorship. Also we will use Redex tools like Random-testing to prove language properties as security.

Formalize this semantic may give more security and confidence to projects developed in Crystal.

Índice general

Índice de figuras

Índice de cuadros

Introducción | 1

1.1. Objetivos

En esta tesis se trabaja con la intención de definir un lenguaje formal llamado Crystal-Clear que modele parte de Crystal. En especial buscamos definir algunas funciones básicas en cualquier lenguaje de programación y luego concentrarnos en el algoritmo de inferencia de tipos. Para especificar un lenguaje de programación primero es importante entender que lo que se busca es describir la sintaxis y semántica del lenguaje , esto se logra a través de la utilización de diferentes herramientas que se verán a lo largo de este trabajo.

Otro objetivo de esta tesis es testear dicha formalización y convencernos de que Crystal-Clear realmente modela la parte del lenguaje que decidimos trabajar. Una vez estemos seguros de esto podremos utilizar herramientas como redex-check para probar que esté inferidor cumple con la propiedad safety(definida en pierce).

1.2. Motivación

Crystal es un lenguaje de programación que fue lanzado en 2021, como tal todavía tiene un largo camino por recorrer en temas como comunidad y desarrollo. Para ello un paso importante es conseguir una formalización completa de su lenguaje. En especial el inferidor de tipos que agrega nuevas cualidades no observadas en otros lenguajes ya trabajados.

1.3. Herramientas

En el trabajo de esta tesis se utilizaron múltiples herramientas tanto prácticas como teóricas. En la parte teórica utilizamos conceptos como gramáticas, (TODO). En cuanto a herramientas prácticas empleamos un lenguaje de programación

llamado Racket basado en Lisp. La utilización de esta herramienta es debido a la existencia de su librería redex, que nos permite definir estos objetos teóricos y usar las mismas para probar propiedades y desarrollar tests para convencernos de los resultados obtenidos.

1.4. Estructura de la tesis

La tesis estará dividida en 6 capítulos.

En el primero hablaremos de Crystal y Racket/Redex, Explicaremos propiedades interesantes de ambos lenguajes y se intentará transmitir un entendimiento acerca de las sintaxis de ambos.

En el segundo se explicarán conceptos teóricos utilizados que servirán para entender el trabajo de la tesis.

En el tercero se hablará de la gramática y semántica de Crystal-Clear nuestra formalización.

Luego seguiremos a la mecanización donde intentaremos comparar nuestro lenguaje a Crystal, observaremos resultados obtenidos y hablaremos de su utilidad para la comunidad, también comentaremos sobre carencias de la documentación.

En el penúltimo capítulos comentaremos trabajos relacionados, ya sea de otras formalizaciones de otros lenguajes como de conceptos teóricos similares desarrollados por otros investigadores.

Por último daremos a conocer nuestras conclusiones y posibles trabajos futuros.

1.5. 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 wich 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.

1.6.   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 1.7 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]

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]

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.* Figura 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.

1.7. 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

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

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

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

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`.

kaobiblio.sty Contains commands to add citations and customise the bibliography.

packages.sty Loads additional packages to decorate the writing with special contents (for instance, the `listing` package is loaded here as it is not required in every book). There are also defined some useful commands to print the same words always in the same way, *e.g.* latin words in italics or `packages` in verbatim.

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

environments.sty Provides special environments, like boxes. Both simple and complex environments are available; by complex we mean that they are endowed with a counter, floating and can be put in a special table of contents.[4]

theorems.sty The style of mathematical environments. Acutally, there are two such packages: one is for plain theorems, *i.e.* the theorems are printed in plain text; the other uses `mdframed` to draw a box around theorems. You can plug the most appropriate style into its document.

4: See Capítulo 8 on page 33 for some examples.

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!

1.8. 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. To get started, 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.

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.

Conceptos importantes y herramientas

Crystal y Racket | 2

2.1. Resumen de cualidades de Crystal

2.2. ¿Que es racket/Redex y porque lo usamos?

CRYSTAL-CLEAR Y MECANIZACIÓN

Trabajo relacionado y conclusiones

Class Options, Commands and Environments

Class Options | 3

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. . .

3.1. **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».

3.2. **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 Capítulo 8 (Mathematics and Boxes) on page 33. Throughout the book I shall use these boxes to remarks what still needs to be done.

3.3.   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 6.3 instead.

3.4. 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 Capítulo 7 (Page Design) on page 28.[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 | 4

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.

4.1.   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.

4.2.   Marginnotes

This command is very similar to the previous one. You can create a marginnote 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

```
\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 margin-
notes is also valid for sidenotes. Side- and margin- notes are
shifted slightly upwards (\renewcommand{\marginnotevadjust}{3pt})
in order to allineate them to the bottom of the line of text
where the note is issued.

4.3. 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.[*]

4.4. 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

---

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

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:

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

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.

4.5.   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 4.1: An example of a margin listing.

```
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.

5 Figures and Tables

5.1. 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 (Figura 5.1), 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:

The credits for the image above the chapter title go to: Bushra Feroz — Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=68724647

```
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 Cuadro 5.1.

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

Cuadro 5.1: A useless table.

I don't have much else to say, so I will just insert some blind text. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Figura 5.1: It's Mona Lisa again. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

5.2.   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. Figura 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}
```

Listing 5.2: Another caption.

There is also the `margintable` environment, of which Cuadro 5.2 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:

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

Cuadro 5.2: Another useless table.

col1	col2	col3
Multiple row	cell2	cell3
	cell5	cell6
	cell8	cell9

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.

5.3.   Wide Figures and Tables

With the environments `figure*` and `table*` you can insert figures which span the whole page width. The caption will be positioned below or above, according to taste.

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 Capítulo 7 on page 28. Now it is time to tackle hyperreferences.

Figura 5.2: A wide seaside, and a wide caption. Credits: By Bushra Feroz — Own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=68724647

References | 6

6.1. 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 y col. (2008), «Heritability in the genomics era–concepts and misconceptions.»
[2]: James y col. (2013), *An Introduction to Statistical Learning*

[2]: James y col. (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]

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.

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

```
\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 y col. (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 y col.: «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
```

6.2.   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:[2]

2: These are the commands you would run in a UNIX system; I have no idea on how it works in Windows.

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

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.

This brief section was by no means a complete reference on the subject, therefore you should consult the documentation of the above package to gain a full understanding of how they work.

6.3. Hyperreferences

In this class we provide a handy sub-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`
- Theorem: `\labthm`
- Proposition: `\labprop`
- Lemma: `\lablemma`
- Remark: `\labremark`
- Example: `\labexample`
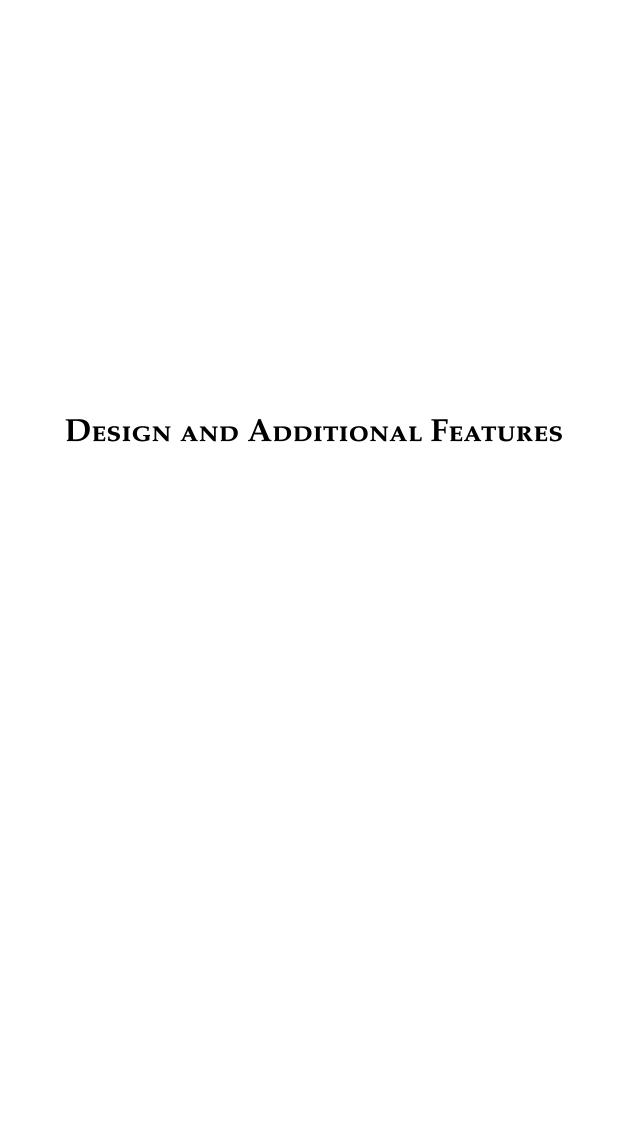- Exercise: `\labexercise`

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: Capítulo 6.

The vario and name styles rest upon the `varioref` package. Their syntax is `\vrefch{chapter-title}` and `\nrefch{chapter-title}`, and they result in: Capítulo 6 on page 23, for the vario style, and: Capítulo 6 (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: Capítulo 6 (References) on page 23.

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.

Design and Additional Features

7 Page Design

7.1. 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 appendix (on page 38); 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», 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.

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

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.

```
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 invide 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 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.

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!

7.2. Headers & Footers

Headers and footers in KOMA-Script are handled by the scrlayer-scrpage package. There are two basic style: «scrheadings» and «plain.scrheadings». 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 stles, 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 reminescent of the «kao» style for chapter titles.

To Do

The twoside class option is still unstable and may lead to unexpected behaviours. As always, any help will be greatly appreciated.

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

Cuadro 7.1: Commands to add a particular entry to the table of contents.

7.3. 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 Cuadro 7.1, each item corresponds to a possible entry in the TOC, and its description is the command you need to provide to have such entry. These commands are specified in the attached style package,[4] so if you don't want the entries, just comment the corresponding lines.

4: 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 Capítulo 6, we will biefly 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.[5]

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

7.4. 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}
```

7.5.   Numbers & Counters

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

By default, dispositions are numbered up to the section. This is achieved by setting: \setcounter{secnumdepth}{1}.

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}}
```

7.6.   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 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 | 8

8.1. 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]

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

Definition 8.1.1 *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).*

Definition 8.1.1 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 8.1.1 *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 8.1.2 *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 8.1.3 *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 8.1.4** (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).*

Demostración. The proof is left to the reader as a trivial exercise. Hint: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. □

> **Definition 8.1.2** *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).*

> **Example 8.1.1** 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 8.1.1** 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.

Here is a random equation, just because we can:

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

> **Remark 8.1.2** 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:
>
> $$\int_a^b x^2 dx$$

We provide two files for the theorem styles: plaintheorems.sty, which you should include if you do not want coloured boxes around theorems; and mdftheorems.sty, which is the one used for this document.[2] Of course, you will have to edit these files according to your taste and the general style of the book.

2: The plain one is not showed, but actually it is exactly the same as this one, only without the yellow boxes.

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

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.

> **Title of the box**
>
> Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

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

Comment 8.2.1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.
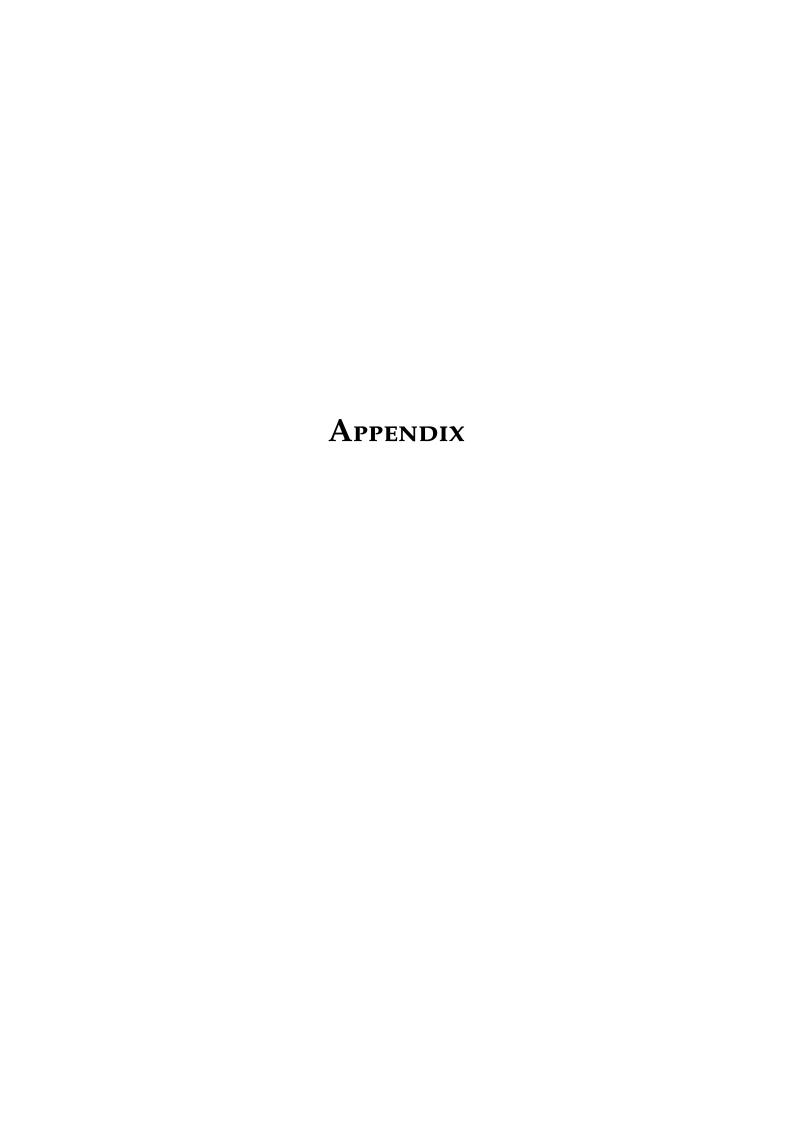
8.3. Experiments

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

title of margin note

Margin note inside a kaobox. (Actually, kaobox inside a marginnote!)

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.

APPENDIX

A
Heading on level 0 (chapter)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

A.1. Heading on level 1 (section)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Heading on level 2 (subsection)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Heading on level 3 (subsubsection)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Heading on level 4 (paragraph)   Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque

placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

A.2.   Lists

Example for list (itemize)

- First itemtext
- Second itemtext
- Last itemtext
- First itemtext
- Second itemtext

Example for list (4*itemize)

- First itemtext
  - First itemtext
    - First itemtext
      - First itemtext
      - Second itemtext
    - Last itemtext
  - First itemtext
- Second itemtext

Example for list (enumerate)

1. First itemtext
2. Second itemtext
3. Last itemtext
4. First itemtext
5. Second itemtext

Example for list (4*enumerate)

1. First itemtext
   a) First itemtext
      1) First itemtext
         a' First itemtext

        b′ Second itemtext
      2) Last itemtext
    b) First itemtext
  2. Second itemtext

Example for list (description)

First itemtext
Second itemtext
Last itemtext
First itemtext
Second itemtext

Example for list (4*description)

First itemtext
    First itemtext
        First itemtext
            First itemtext
            Second itemtext
        Last itemtext
    First itemtext
Second itemtext

Bibliography

Here are the references in citation order.

[1] Peter M Visscher, William G Hill y Naomi R Wray. «Heritability in the genomics era–concepts and misconceptions.» En: *Nat. Rev. Genet.* 9.4 (2008), págs. 255-266. DOI: `10.1038/nrg2322` (vid. pág. 23).

[2] Gareth James y col. *An Introduction to Statistical Learning*. 2013 (vid. pág. 23).

[3] Alexis Battle y col. «Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individuals». En: *Genome Res.* 24.1 (2014), págs. 14-24. DOI: `10.1101/gr.155192.113` (vid. pág. 23).

[4] Hui Zou y Trevor Hastie. «Regularization and variable selection via the elastic-net». En: *J. R. Stat. Soc.* 67.2 (2005), págs. 301-320. DOI: `10.1111/j.1467-9868.2005.00503.x` (vid. pág. 24).

Greek Letters with Pronounciation

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.

Alphabetical Index