Το `SuperCollider` βιβλίο (class)

Use this document as a template

Example and documentation of the `kaobook` class

Customise this page according to your needs

Federico Marotta *

August 18, 2021

An Awesome Publisher

* A LaTeX lover

Το `SuperCollider` βιβλίο (class)

Disclaimer
You can edit this page to suit your needs. For instance, here we have a no copyright statement, a colophon and some other information. This page is based on the corresponding page of Ken Arroyo Ohori's thesis, with minimal changes.

Είναι εξαιρετικό ότι μπορούμε να γράφουμε και ελληνικά εδώ

Colophon
This document was typeset with the help of KOMA-Script and LaTeX using the kaobook class. The source code of this book is available at:
https://github.com/fmarotta/kaobook
(You are welcome to contribute!)

Publisher
First printed in May 2019 by An Awesome Publisher

Κάθε μέρα που δεν αγάπησες είναι μια μέρα που δεν έζησες

– Στέφανος Ξενάκης

Πίνακας Περιεχομένων

List of Figures

List of Tables

You can edit this page to suit your needs. For instance, here we have a no copyright statement, a colophon and some other information. This page is based on the corresponding pag I am slowly beginning to get nervous whether the thinky is compiled. This building from scratch is slow but it will succeed. Made glossaries. Made nomenclature. ..., greek. accident, and ***ατύχημα.***

Now trying to fix this by enclosing it in a textgreek statement: The following word is going to be in Greek: Ελληνικά. And continuing in English...

Some more greek here: ελληνικά. Μπράβο.

Ελληνικά.

Καλημέρα Μάνο,

ας βρεθούμε μεθαύριο Παρασκευή στις 14:00, γιατί αύριο πρέπει να βοηθήσω σε δουλειές τη μάνα μου να μη βγαίνει με τη ζέστη.

Μπορούμε γενικότερα να αντικαταστήσουμε τις συναντήσεις που προτείνω για Δευτέρα, με αντίστοιχες τις Παρασκευές κάθε εβδομάδα.

ok let's try some English here for fun.

OK για αύριο Παρασκευή 14:00. Επίσης και για Δευτέρες-παρασκευές από δώ κα ιπέρα

Όσο *περισσότερο* ξανα-επισκέπτομαι το Pure `Data` αυτές τις μέρες, τόσο `συνειδητοποιώ` τη δυσκολία του εγχειρήματος μας.

Σίγουρα μια safe θεώρηση είναι ένα step-by-step tutorial για το κάθε **προγραμματιστικό** περιβάλλον.

Σίγουρα μια safe θεώρηση είναι ένα step-by-step tutorial για το κάθε προγραμματιστικό περιβάλλον.

Ας το δούμε μαζί. Μπορούμε να κάνουμε μαι γενική σύγκριση σε θεωρητικό επίπεδο σε ένα ξεχωριστό κεφαλαιο, και μετά να κάνουμε μια λίστα από βασικές εφαρμογες πχ. additive synthesis που μπορούν να γίνουν και με τια δυο περιβάλλοντα, και να δείξουμε πως γίνεται η καθε εφαρμογή στο καθένα, σε αντοπαράθεση και σύγκριση. Αρκει να φτιάξουμε την λίστα και να αρχίσουμε, και θα δούμε.

Πιο δύσκολο, εντούτοις ενδιαφέρον, να είναι κάθε Κεφάλαιο project based, όπου οι αναγνώστες να φτάνουν στο ίδιο αποτέλεσμα από δύο δρόμους.

Νομ'ιζω αυτό εννοοώ και γώ παραπάνω. Ας αρχίσουμε και βλέπουμε.

You can edit this page to suit your needs. For instance, here we have a no copyright statement, a colophon and some other information. This page is based on the corresponding pag I am slowly beginning to get nervous whether the thinky is compiled. This building from scratch is slow but it will succeed. Made glossaries. Made nomenclature. ..., greek. accident, and ***ατύχημα.***

Now trying to fix this by enclosing it in a textgreek statement: The following word is going to be in Greek: Ελληνικά. And continuing in English...

Some more greek here: ελληνικά. Μπράβο.

Ελληνικά.

Καλημέρα Μάνο,

ας βρεθούμε μεθαύριο Παρασκευή στις 14:00, γιατί αύριο πρέπει να βοηθήσω σε δουλειές τη μάνα μου να μη βγαίνει με τη ζέστη.

Μπορούμε γενικότερα να αντικαταστήσουμε τις συναντήσεις που προτείνω για Δευτέρα, με αντίστοιχες τις Παρασκευές κάθε εβδομάδα.

ok let's try some English here for fun.

OK για αύριο Παρασκευή 14:00. Επίσης και για Δευτέρες-παρασκευές από δώ κα ιπέρα

Όσο *περισσότερο* ξανα-επισκέπτομαι το Pure `Data` αυτές τις μέρες, τόσο `συνειδητοποιώ` τη δυσκολία του εγχειρήματος μας.

Σίγουρα μια safe θεώρηση είναι ένα step-by-step tutorial για το κάθε **προγραμματιστικό** περιβάλλον.

Σίγουρα μια safe θεώρηση είναι ένα step-by-step tutorial για το κάθε προγραμματιστικό περιβάλλον.

Ας το δούμε μαζί. Μπορούμε να κάνουμε μαι γενική σύγκριση σε θεωρητικό επίπεδο σε ένα ξεχωριστό κεφαλαιο, και μετά να κάνουμε μια λίστα από βασικές εφαρμογες πχ. additive synthesis που μπορούν να γίνουν και με τια δυο περιβάλλοντα, και να δείξουμε πως γίνεται η καθε εφαρμογή στο καθένα, σε αντοπαράθεση και σύγκριση. Αρκει να φτιάξουμε την λίστα και να αρχίσουμε, και θα δούμε.

Πιο δύσκολο, εντούτοις ενδιαφέρον, να είναι κάθε Κεφάλαιο project based, όπου οι αναγνώστες να φτάνουν στο ίδιο αποτέλεσμα από δύο δρόμους.

Νομ'ιζω αυτό εννοοώ και γώ παραπάνω. Ας αρχίσουμε και βλέπουμε.

Εισαγωγή 1

1.1 The Main Ideas

This is a test to see if input works.

This is a new paragraph within the input test file.

N.B. Org files for chapters should be placed directly in the chapters folder. Their content should be exported to latex as body-only.

The end.

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.

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: 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!

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 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]

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. 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.

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

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

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`.

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]

4: See Chapter ?? on page ?? for some examples.

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.

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. 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.