The Jolly Writer

The Jolly Writer

Dépot légal : août 2020
ISBN : 978-2-9572100-0-8
Première édition

Achevé d'imprimer en août 2020 par **WirmachenDruck**
Mühlbachstraße 7, 71522 Backnang, Allemagne
Imprimé en Allemagne

Scypress
6, rue de Versailles
78460 Chevreuse

The Jolly Writer

Joris van der Hoeven

Table of contents

Preface

The ultimate aim of a typesetter is to compose books in such a perfect manner that nobody will notice it. Nothing is more dishonorable than to make a reader stumble over an ill-placed comma. What the typesetter wants is to bury you cozily in your armchair, with your complete focus on the book, and in a spell not to be broken by trivial typographic details.

If you are a writer instead of a reader, then it takes far more than an armchair to put you in the right creative state of mind. The ideal text editor should help you to maintain this state for as long as possible. You don't want to be torn out of trance by a misguided comma key that suddenly starts inserting apostrophes. The ideal writing tool should behave as your first art pen from school: it should unleash your desire to write, make you feel one with the tool, and occasionally surprise you with the beauty of your own words.

Back in 1996, when I was writing my PhD in computer science, none of the existing text editors were even close to this ideal. General purpose editors such as MICROSOFT WORD produced documents of poor quality and made it difficult to type mathematical formulas. In order to compose professional-looking scientific documents, the main alternative was to use (LA)TEX. Although this solution was nice from the reader's perspective, it forced writers to encode their prose in a technical pseudo-language. (LA)TEX then relied on an akward "compilation" process in order to transform this pseudo-code into a printable and human-readable document.

After my PhD, this unsatisfactory state of the art led me to start the development of GNU TEXMACS, a free office suite for scientists. Before anything else, TEXMACS allows you to create beautiful scientific documents with special types of content, such as mathematical formulas or technical pictures. TEXMACS also provides interfaces for various external systems for symbolic and numeric computations. Recent versions further include a laptop presentation facility, versioning tools, integrated documentation, etc. In the areas of science and education, the ultimate aim is to provide a complete suite for the most frequent tasks on a computer.

One distinctive feature of TEXMACS is that it does not compromise on quality. First of all, the final documents have a professional typesetting quality, similar or even superior to what is achieved by (LA)TEX. Yet, user-friendliness has not been sacrificed, since the system also provides an intuitive wysiwyg (what-you-see-is-what-you-get) graphical interface. Finally, TEXMACS favors the composition of so-called "structured" documents, in contrast to existing wysiwyg interfaces of general purpose office suites, which are mainly "presentation oriented".

The name "GNU TEXMACS" is explained by the facts that TEXMACS is part of the GNU project and that some initial inspiration was drawn from the (LA)TEX systems [36, 38] and GNU EMACS [53]. However, it has become clear over time that this choice of name was one of the biggest mistakes of the project. Indeed, the name incorrectly suggests that TEXMACS is some kind of interface to (LA)TEX. So let me stress once and for all: the current TEXMACS system is completely independent from (LA)TEX. Of course, TEXMACS does provide converters between its native format and LATEX.

As a free software, the development of TEXMACS has benefitted from the help of a wide community of contributors around the globe. Particular thanks go to the following main co-developers throughout the years: David ALLOUCHE, Miguel DE BENITO DELGADO, Darcy CHEN, Andrey GROZIN, Massimiliano GUBINELLI, Philippe JOYEZ, Grégoire LECERF, Henri LESOURD, François POULAIN, and Denis RAUX. Further thanks go to the many other contributors and supporters; see `http://www.texmacs.org/tmweb/about/authors.en.html` for a more extensive list. I am indebted to Basile AUDOLY and Kostas OIKONOMOU for their careful proofreading.

I also wish to express my gratitude to those who have provided financial support to the TEXMACS project: CNRS, CRI-TECH de Haute-Savoie, DIGITEO, Rennes MÉTROPÔLE, SPRINGER-VERLAG, INRIA, Dan GRAYSON, and Christoph BENZMUELLER.

TEXMACS is built on top of a lot of other software. The QT, GUILE, FREETYPE, and HUMMUS libraries are particularly essential [47, 33, 63, 34]. Recent versions integrate an increasing amount of artwork, such as the TEX Gyre fonts [32] (the main font of this book is PAGELLA), various "Subtle Patterns" [61], more fonts from DAFONT [9], and further freely available pictures from WIKIMEDIA [65].

Last but not least, I wish to thank SYLVIE, JUDITH, and NIELS for their support and patience.

Chapter 1

Introduction

1.1 How to read this book

The fact that you are reading these lines makes it is quite likely that you are already a convinced TEXMACS user. Maybe you actually bought this book in order to support our development efforts rather than to actually read it. Indeed, we tried to make TEXMACS so user-friendly that you will naturally learn the software while using it. The system also ships with a lot of integrated documentation that is most easily searched and consulted from within TEXMACS itself.

So why read this book? If you are a TEXMACS user or plan to become one, then you probably spend a considerable amount of your time on writing documents, doing computations, or making presentations from your laptop. In comparison, reading parts of this book will be a minor investment that might help you get the most out of TEXMACS.

On top of being an up-to-date manual for TEXMACS 2.1, this book indeed contains a substantial amount of information that is not so obvious to acquire just by using the software. For example, you can learn some of the basics of professional typesetting; typing kilometers of text will not miraculously instill such knowledge into your mind. Similarly, although every single feature of TEXMACS is rather self-explanatory, interesting results can be obtained through less obvious combinations of them. In this book, we present several tips and tricks of this kind.

One of the original motivations for developing TEXMACS was that traditional office suites such as MICROSOFT OFFICE [42] and LIBREOFFICE [13] offered poor support for scientific users. For instance, the typesetting quality of mathematical formulas was low in comparison to (LA)TEX [36, 38] and visual formatting was favored over structural markup (see section 1.3). For this reason, TEXMACS is particularly well suited for scientists, teachers, and students. But other types of users may also appreciate the combination of high typesetting quality, a wysiwyg front-end (what you see is what you get), and the possibility to write structured documents [28, 21].

As a non-native English speaker, the one thing I cannot teach you is how to write English. Pinker's *The Sense of Style* [44] is one excellent book that I can recommend to any fellow scientist who wants to become a better writer. Of course, you have a wealth of other options, such as Strunk and White's *The Ele-*

ments of Style [55] or Fowler's *Dictionary to Modern English Usage* [14]. Besides these classics, a few more recent style guides are Thomas and Turner's *Clear and Simple as the Truth: Writing Classic Prose* [60] and Williams's *Toward clarity and grace* [66]. *The Cambridge Grammar of the English Language* [31] is one of the bibles on grammar, whereas *Merriam-Webster's Dictionary of English Usage* is a dictionary that elevates words beyond the ordinary.

This book is organized as follows. The remainder of the present chapter is devoted to a quick introduction to some of the main features of TEXMACS and its design philosophy. The next chapter discusses the the graphical user interface. These two chapters introduce the most important concepts and terminology, so we recommend that you read them first. The other chapters are fairly independent and organized by increasing order of difficulty.

Let us mention a few conventions that are followed throughout this book. We use a sans serif font for menu entries: "you may create a new file using File ▸ New". Sometimes, clicking on a menu button will open a new popup window in which you have to click on yet another series of buttons. The menu notation naturally extends to this case, as in Edit ▸ Preferences ▸ Convert ▸ Pdf.

We use boxes such as ⌘⇧x and -|> for keyboard shortcuts. The first example ⌘⇧x corresponds to simultaneously pressing the modifier keys ⌘ (meta) and ⇧ (shift) together with the letter x. In the second example -|>, you first have to press and release the - key and next press and release the > key. For more information, we refer to section 2.3.1.

Notice that the default behavior of the menus and keyboard depends on your operating system. Indeed, TEXMACS attempts to follow the same conventions as the other programs on your computer, and these conventions may vary with your operating system. The default behavior can be overridden by selecting another "look and feel" in the user preferences (Edit ▸ Preferences ▸ General). The keyboard shortcuts in this book hold for the EMACS look and feel on MAC computers. That said, many of them work for all look and feels, up to minor prefix adjustments. See section 2.3.6 for more details.

1.2 Making science beautiful

Modern word processors have made it easy for just about anyone to create nice-looking documents. Yet, there exists a difference between "nice" and "beautiful". A few decades ago, typesetting books was the job of professionals. Whereas most of the typesetting can nowadays be carried out automatically, some design decisions still have to be made by humans. So let us step back and ask ourselves what makes a document attractive.

Typography is a matter of technical craftsmanship. Although we might compare it with architecture or the building of a bridge, typography does not involve emotions, as do arts like poetry, painting, or literature.

One fundamental idea about the esthetics of technology is that beauty is strongly correlated with adequacy to the purpose. A skillfully conceived and presented book usually provokes a quiet feeling of harmony, whereas clumsy formatting or an abundance of exuberant fonts will distract and thereby irritate the reader. In a sense, the better the typesetting quality, the less you will notice it. An advertiser, on the contrary, will use flashy colors and bold fonts to attract your attention. This may convince you to buy washing powder, but it does not make you better at differential geometry.

How to put this into practice? When starting a new document, you will first have to decide on the global layout, such as the page size, the extent of the actual text, the number of columns, the main fonts, etc. Here are a few useful rules:

- Do *not* make your paragraphs too wide, since wide paragraphs make it harder for the eye to find the next line, and thereby cause fatigue for the reader.

- Consistently stick to the same layout throughout your document. Use a different layout only if there is a special reason for it.

- In particular, keep the number of fonts to a minimum. Use alternative fonts for specific purposes only.

- Use fonts and a general layout that are well suited to the medium of publication and to the intended purpose. For example, sans serif fonts are more readable on a computer screen (so you might prefer them for web pages and laptop presentations) but may look unprofessional on paper.

To ease the choice of a general layout, TEXMACS offers *document styles* for various purposes: you may select Document ▸ Style ▸ Article for writing an article, Document ▸ Style ▸ Beamer for creating a laptop presentation, and so on. TEXMACS also supports some of the most important styles used by publishing companies. For instance, Document ▸ Style ▸ Article ▸ Springer ▸ llncs can be used for articles in Springer's series of "Lecture Notes on Computer Science".

Style files rely on *markup* to map specific intentions of the author to appropriate layouts. For example, it is customary to use either an italic or a bold font for important text. Such choices are made once and for all in the document style and then consistently applied throughout the entire document. We will return to this issue in the next section about document structure. For now, we observe once again that the consistent use of a minimal number of fonts and layouts is helpful in several ways: it allows the reader to stay focussed, without getting distracted by incongruous design decisions; it eases navigation within the document; and ideas can be transmitted through the mere use of appropriate typography.

The same lines of thought also apply to the main text. Many authors put a lot of effort into the science, some good will into the English spelling, and little energy into punctuation and other typographic issues. Fortunately, professional typesetting systems such as (LA)TEX and TEXMACS have taken over a large part of this task. However, despite sophisticated algorithms for spacing, line-breaking, page-breaking, positioning individual characters, etc., they occasionally require some human assistance.

Basic punctuation rules specify that you enter a space after a period "." at the end of a sentence, before starting a new one. If you correctly apply this rule, then TEXMACS will understand your intent and use this during the typesetting process (for instance, the space between two sentences will be slightly larger than between two ordinary words). Although TEXMACS does not forbid you to omit spaces after periods, it will fail to do the typesetting correctly if you break basic punctuation rules.

Explicit human intervention is typically required when using periods "." for other purposes, such as decimal dots, or inside abbreviations. Note that the space after an abbreviation is somewhat smaller than after the final period of a sentence. By identifying abbreviations through the use of appropriate markup, you may help TEXMACS to get the typesetting right.

Punctuation, typography, and even spelling are often perceived as dull and unimportant details. Please keep in mind that some of your readers might even if *you* don't. Your sloppiness may easily irritate or confuse others and lead to ambiguities or even errors. The good news is that it requires only minimal awareness and discipline to do things right in the first place. Furthermore, TEXMACS will do a big part of the job for you, as long as you follow a few basic rules. In sections 3.1 and 3.7, we will come back to these issues in more detail.

1.3 Structuring your documents

We have already stressed the importance of a consistent layout and the use of appropriate fonts. TEXMACS has been designed so as to make it as easy as possible to create so-called *structured documents*. When properly used, this ensures in particular a uniform visual appearance.

The aim of structured documents is to focus on *meaning* or *intent* rather than *presentation*. For example, instead of manually typesetting all section titles in a bold 17 point sized font, we urge you to use a special *markup element* section (also called *tag*) that is dedicated to section titles. This has many advantages: TEXMACS will automatically number your sections, guarantee a uniform presentation of all section titles, build an automatic table of contents, etc. Furthermore, whenever you change your mind, and wish to opt for another rendering of section titles, then it will not be necessary to modify the layout of every individual title in your document: it suffices to tell TEXMACS once and for all how to display the section tag.

The rendering of a document usually reflects its structure, which makes it natural to edit structured documents in a wysiwyg manner. For instance, it is likely that a short line, typeset using a bold 17 point font, and starting with a number, is always a section title. However, some of the structure may not directly be apparent: from the mere rendering of

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n, \tag{1.1}$$

it is not clear whether $\binom{n}{k}$ was entered as a vector or as a binomial coefficient. Indeed, if we were not aware of the existence of the appropriate binom tag, then we might have entered the binomial coefficient as a vector instead, which has a similar visual appearance. The use of appropriate tags becomes essential whenever the precise semantics of the formula matters. This happens for instance when you want to check the correctness of the formula, by copying it into your favorite computer algebra system. Another example concerns French translations, in which case the traditional notation for binomial coefficients becomes C_n^k.

Even in cases when it seems overkill to use structural markup for producing a certain visual effect, doing so remains a good habit. On the one hand, the systematic use of appropriate markup helps you to formalize what you are doing. On the other hand, structured documents are more sustainable, in the sense that they can more easily be reused in situations that you did not anticipate, initially.

The precise way in which markup elements are rendered is specified in so-called *style* files and packages. If you are new to TEXMACS, then you do not have to worry about how the actual translations take place: just select a style file that suits your needs and aesthetic preferences. The standard style files and packages offer a large number of markup elements that can be used in common situations.

More advanced users may actually want to define or customize some of the document structure by themselves. This typically happens when you introduce your own notations or if you frequently need a specific non-standard layout pattern. The simplest way to add a new markup element to TEXMACS is through the definition of a *macro*, using Tools ▸ Macros ▸ New macro. By default, such macro definitions are put in the *preamble* of your document. If you want to reuse the same macros in several of your documents, then you may group them together in your own style files or packages. See chapter 12 for more details.

The design of more and more complex macros requires some skill, but one nice aspect of the integrated nature of TEXMACS is that it is easy to study the way the built-in style files and packages work. For example, if your cursor is inside the binomial coefficient of formula (1.1), then the precise markup element binom that was used to produce this coefficient is displayed on the status and focus bars of your window (see Figure 2.1 on page 27), as well as in the Focus menu.

Using Focus ▸ Preferences ▸ Edit macro you may then edit the macro definition that specifies the rendering of the binom tag. You may also inspect the style package in which the binom tag is defined, using Focus ▸ Preferences ▸ Edit source. Through trial and error, this allows you to progressively learn how to write more and more sophisticated style packages by yourself.

1.4 The joy of wysiwyg-ness

In the same way as professional typesetters strive to produce documents that allow readers to fully concentrate on what they are reading, we feel that good text editors should allow authors to remain entirely focussed on the creative process of writing. Whereas readers may get distracted by typos, clumsy formatting, or loud fonts, the writing process gets interrupted whenever you are obliged to address technical details before you can enter your thoughts into the computer.

One prerequisite for maximal user comfort is to use a *wysiwyg* (what you see is what you get) editor. It is indeed most natural to edit documents in the actual form that you want to look in print; this is probably also closest to your mental representation of the document. However, mere wysiwyg-ness is not enough, since the precise way in which documents are being edited should also be both natural and efficient.

For instance, several basic mathematical formula editors propose "palettes" with the available mathematical symbols, which tend to clutter the screen. If even simple symbols need to be entered *via* these palettes, then this also gives rise to a lot of back and forth movements between the keyboard and the mouse. Similarly, most front-ends to (LA)TEX force you to remember a long list of symbol names, or to interrupt the writing process just to look up the appropriate name of a symbol. In TEXMACS, a more efficient *and* user-friendly system was developed from scratch. Most symbols can be entered in an intuitive fashion, following a small number of basic rules (see section 4.2). For example, you may enter →, ↦, α, $\not\subseteq$, and $\subseteq$ via the keyboard shortcuts -|>, |-|>, a⇥, <|=|/, and <|⇥|⇥|=, respectively.

Historically speaking, early wysiwyg editors such as MICROSOFT WORD often lacked adequate tools for the efficient composition of scientific documents, especially those that contained many mathematical formulas. Moreover, their focus on visual rather than logical design often resulted in documents of poor quality. For this reason, such editors became a frequent target of mockery, "wysiwyg" being denigrated as "what you see is all you've got" [38, section 1.5]. As a result, the (LA)TEX movement took the rather dramatic step of abandoning the wysiwyg idea altogether. In (LA)TEX, documents are encoded in a textual pseudo-language, and a special compiler is required in order to obtain the desired printable versions. However, not everyone is comfortable using such a pseudo-language, which forces you to imagine the text you are typing, instead of actually seeing it.

Nevertheless, some of the historical criticism on wysiwyg editors is indeed justified and points to real problems. For example, the visual presentation of a document does not necessarily reflect its full structure, as we saw in the binomial coefficients example (1.1) of the previous section. But this drawback mostly vanishes for well written documents, under the reasonable assumption that the author (yourself) always used appropriate markup at the first place: if you *see* a binomial coefficient, then it *is* a binomial coefficient. In TEXMACS, the problem is further reduced by the fact that it is not only the final presentation on paper that matters, but rather the "full interaction" between the text and the author inside the editor. For instance, we already noticed that the structure becomes apparent when positioning the cursor inside. Similarly, one may select an alternative rendering style, and even a "LATEX-source-like" rendering style in which the full structure is visually exposed.

Another issue concerns the ability to position the cursor at all relevant places inside the text. For example, consider the text "**bold***italic*". In most wysiwyg editors, there is a single cursor position available between the bold '**d**' and the first italic '*i*', which makes it difficult to guess whether an extra keystroke typed at this cursor position will produce bold, italic or plain text. By contrast, TEXMACS offers three cursor positions [24], one after the '**d**' inside the bold environment, one between the two environments, and one before the '*i*' inside the italic environment. The user knows exactly which of these positions the cursor is in, thanks to visual clues in the interface: in particular, the first englobing environment is displayed at the bottom-right of the window.

Cursor movement raises some other interesting problems in structured documents with formulas that heavily rely on two dimensional layout. For instance, some early structured wysiwyg editors would move the cursor according to the abstract structure. This typically implied that pressing the → button at the end of a numerator of fraction would move the cursor to the start of the denominator. It turns out that this way of doing it can be very confusing. In TEXMACS, the cursor movement is mostly graphical: if you press →, then you will go to the right, if possible [24, 1]. There are some exceptions, such as a cursor at the end of the line, or the cursor movement in computer algebra sessions, in which case it is natural to jump from one input to the other when moving up and down.

Other historical arguments against wysiwyg editors are more dubious. For example, some people believe that a "format" such as (LA)TEX is "more scriptable" in the sense that (LA)TEX documents can more easily be generated, combined and manipulated by external tools than TEXMACS documents. On a superficial level and for some very simple tasks, this may be so, because operating systems include many tools for the manipulation of text files. However, since TEXMACS documents are truly structured, it is unnatural to manipulate them as strings: one should rather regard them as expression trees, to be manipulated in a scripting language that can handle such objects. A large part of the TEXMACS interface is written in the SCHEME language, whose syntax and other features are particularly well suited for this task. Moreover, for those

who really want to manipulate TEXMACS documents as text files, we offer several plain text formats, such as XML, SCHEME expressions, etc. The syntax of these formats is systematic and well-defined, contrary to (LA)TEX.

In summary, the most annoying drawbacks of early wysiwyg text editors have been addressed in TEXMACS through the implementation of various, occasionally novel, ideas. The further advantages of wysiwyg editors become most apparent for two dimensional layouts. For example, consider a table or a matrix. In TEXMACS, it is easy to insert a new column using a keyboard shortcut (⌥← or ⌥→), the menus, or the toolbars. In (LA)TEX or HTML, this requires you to modify every individual row of the table. Technical pictures are an even more convincing example. The original LATEX `picture` environment invites you to enter the coordinates of each individual component. No wonder that (LA)TEX eliminated pictures from mathematical textbooks more surely than the BOURBAKI movement did!

1.5 TEXMACS as a structured editor

We have explained why structured documents help you to ensure a uniform presentation. We also discussed how TEXMACS allows you to efficiently write such documents, using a user-friendly, wysiwyg interface. You may even create new kinds of markup yourself, using the integrated macro language. Another important advantage of structure is that it leads to new opportunities for enhancing the editing experience.

One of the main ways to exploit structure inside the TEXMACS editor is through the Focus menu. The *current focus* refers to the innermost tag at the current cursor position. For example, assume that we positioned the cursor after the two in the following equation:

$$\frac{x^{2}}{y} = 1 + \frac{x}{y^3}. \tag{1.2}$$

The current focus is highlighted using a cyan box, whereas the grey boxes correspond to the other tags that contain the current cursor at higher levels.

The entries in the Focus menu allow you to directly perform editing operations on the current focus. For instance, in the example (1.2), the superscript can be turned into a subscript using Focus ▸ Superscript ▸ Subscript or the keyboard shortcut ^⇥. You may also jump to the next and previous scripts inside the document using Focus ▸ Next similar and Focus ▸ Previous similar (or ^⇞ and ^⇟), or obtain contextual help on the "superscript" tag rsup using Focus ▸ Describe. See sections 2.7 and 10.5–10.7 for more details.

The document structure is also taken into account for several common editing operations. For example, when searching "*a*" inside a mathematical formula, TEXMACS will only search for the symbol "*a*" inside other mathematical formulas. If you were editing a (LA)TEX source file, then such a search would typically result in many parasite non-mathematical "a" hits inside the main

text. Using the TEXMACS search facility, it is even possible to find text in a structured way (see section 10.2). For instance, when searching for $\frac{x}{\cdots}$, you will find any fraction in which x occurs in the numerator, such as

$$\frac{x}{y}, \quad \frac{1+x}{1+y}, \quad \frac{1+\sqrt{x}}{1-\sqrt{x}}, \ldots$$

Another example of a structured editing facility concerns the computation, presentation and editing of the differences between two versions of a file (see section 10.9). As a general rule, the number of TEXMACS features that exploit document structure steadily grows with the years.

1.6 Towards a scientific office suite

Besides a scientific text editor, there are several other functionalities that can be expected from an office suite dedicated to science and education: the possibility to perform scientific computations from within the editor, an easy way to draw technical pictures, a spreadsheet facility, a laptop presentation tool, version control, collaborative authoring over the web, maintaining databases with bibliographic references, etc.

From early on, TEXMACS comes with interfaces for various computer algebra systems and other mathematical software. This allows you to perform complex mathematical computations directly from inside TEXMACS, while ensuring a professional rendering and a flawless integration of such computer algebra sessions into papers, books, and class material. Here is an example of a session inside the MAXIMA system:

(%i1) $\mathrm{diff}(x^{x^x}, x, 2)$

(%o1) $$x^{x^x}\left(x^x \log(x)\left(\log(x)+1\right)+x^{x-1}\right)^2+x^{x^x}\left(x^x \log(x)\left(\log(x)+1\right)^2+x^{x-1}\left(\log(x)+\frac{x-1}{x}\right)+x^{x-1}\left(\log(x)+1\right)+x^{x-1}\log(x)\right)$$

(%i2) $$\int \frac{x^5+2x-1}{x^2-3x+7}\,\mathrm{d}x$$

(%o2) $$-\frac{57\log\left(x^2-3x+7\right)}{2}+\frac{37\arctan\left(\frac{2x-3}{\sqrt{19}}\right)}{\sqrt{19}}+\frac{x^4+4x^3+4x^2-60x}{4}$$

Besides this traditional kind of interface, recent versions of TEXMACS also propose other mechanisms to interact with external software. For instance, any TEXMACS plug-in can be used as the computational engine of the built-in spreadsheet facility. Currently, it is also possible to use external software for silent computations in the background, such as evaluating or simplifying a selection while editing some formula. Finally, it is fairly easy to add your own plug-ins to external software. Chapter 11 is dedicated to the use of TEXMACS as an interface.

The technical picture editor is another extremely useful tool in the TEXMACS office suite. Although the tool is not as complete as dedicated editors for scalable vector graphics, it is well suited for simple images. One important advantage is that the tool is fully integrated, which makes it easy to put formulas or hyperlinks inside your pictures. Of course, it remains possible to embed pictures that were made using another tool into your documents. The technical picture editor is presented in detail in chapter 8.

A few decades ago, scientific presentations at conferences were done on a blackboard or using slides on an overhead projector. Nowadays, it is most common to directly project the screen of your laptop using a beamer. TEXMACS contains special markup for creating such laptop presentations, allowing you to progressively uncover content, play simple animations and sounds, compose complex overlays, etc. Various standard themes can be used for the general appearance and it is easy to add your own ones using the style package mechanism. It should be stressed that TEXMACS presentations are very dynamic: you may add new content or make corrections during your show. Computer algebra sessions can also be replayed in real time. See chapter 9 for more information about the presentation tool.

Recent developments aim at making TEXMACS more "web-aware". Indeed, when coauthoring papers with several people, it is fashionable to create a central repository on the web from where such papers are accessible to everyone and where to keep a record of all successive versions. SVN [6, 7] and GIT [62, 51] are well known versioning tools that allow you to accomplish this task. TEXMACS comes with an interface for SVN, and a special mode for going through the differences between various versions. Future plans exist for a centrally available TEXMACS server, with many other features beyond mere file sharing and version control.

1.7 Creating your own extensions

(LA)TEX [36, 38] and the GNU EMACS [53] text editor share the common strength that they make it easy for users to extend the system. (LA)TEX allows you to extend the system through the definition of new macros. This makes (LA)TEX quite different from other common markup languages such as HTML [41], where the allowed markup elements often form a fixed set, specified by a so-called DTD. The EMACS editor comes with a so-called *extension language* (called EMACS-LISP, a special dialect of LISP), which allows you to customize the editor and to program new features. One common application is to add support for a new programming language (e.g. syntax highlighting, automatic indentation, etc.).

TEXMACS offers several similar mechanisms for customizations and extensions. We already mentioned the possibility of defining new macros and bundle such macro definitions together into style files and packages (see chapters 12 and 13). TEXMACS uses SCHEME instead of EMACS-LISP as its extension language. All high level editing routines are written in this language and new routines can easily be implemented by the user. In particular, the TEXMACS menus, key-

board shortcuts, and most elements of the graphical interface are programmed in SCHEME. Besides, certain TEXMACS macros may occasionally call SCHEME routines for computing their rendering. The last chapter 14 of this book contains an introduction to SCHEME programming for TEXMACS.

TEXMACS offers one more extension mechanism, which is also used for the creation of interfaces with external software. A TEXMACS *plug-in* consists of a directory with a collection of various files: TEXMACS style files and packages as above, SCHEME files with customizations or extensions of the editor, TEXMACS files with useful documentation, and any other files of interest, such as communication scripts with external programs. TEXMACS plug-ins can easily be shared among users and it suffices to put them at a dedicated place in order to activate them. For more details, we refer to the integrated documentation in Help ▸ Interfacing.

1.8 TEXMACS and (LA)TEX

Before anything else, we stress that recent versions of TEXMACS are completely independent from (LA)TEX [36, 38, 16]. Contrary to what its name might suggest, TEXMACS is *not* a (LA)TEX front-end.

Initially, the development of TEXMACS got some of its inspiration from (LA)TEX and EMACS, whence its name. For instance, TEXMACS uses the same fonts as TEX, and borrows several of its typesetting algorithms (such as the way paragraphs are hyphenated and broken into separate lines). The graphical interface of older versions of TEXMACS looked similar to the EMACS interface, especially concerning keyboard shortcuts and the menus. Finally, as mentioned in the previous section, TEXMACS offers similar extension mechanisms as (LA)TEX and EMACS.

Over time, the differences between TEXMACS, (LA)TEX, and EMACS have become more important. Recent versions of TEXMACS strive to use the same graphical user interface conventions as other software on your system, although you may still opt for an EMACS look and feel in the preferences. TEXMACS has also continued to grow out into a complete scientific office suite, whose capacities go far beyond the production of good looking scientific documents (see section 1.6).

Obviously, the major advantages of TEXMACS with respect to LATEX are its higher versatility and its user-friendly, wysiwyg graphical user interface. The integrated drawing tool and presentation mode are highly convenient as well. There are also a few less prominent improvements. For example, TEXMACS uses better algorithms for page breaking and the computation of space between successive lines. Mathematical formulas also carry more semantics than in LATEX, which is important for interfacing purposes with external software. For example, (LA)TEX makes no essential distinction between the formulas $f(x+y)$ and $a(b+c)$. However, we probably meant "f applied to $x+y$" versus "a multiplied with $b+c$"; TEXMACS encourages you to make the intended meaning explicit.

When it comes to more specialized features, (LA)TEX occasionally conserves its advantages over TEXMACS. Due to its large user base, there are many packages for exotic typesetting needs: if you are writing an essay on Egyptian mathematics and wish to combine mathematical formulas with hieroglyphs, then you will still need (LA)TEX. If you have special ideas on how to present algorithms, then you may be lucky to find a LATEX package that implements precisely your point of view. A few useful LATEX features have also not yet been implemented in TEXMACS, such as page breaking of long tables and wrapping text around figures. Of course, such features are on the TEXMACS developer's wish lists, so it will only be a matter of time before they will also appear in TEXMACS.

The main advantage of LATEX with respect to TEXMACS is of a social nature: most publishers require scientists to send final versions of papers in LATEX and many professors force their students to learn LATEX, sometimes even through dedicated courses. This is a sad situation for many of our users. Fortunately, PDF is an even more standard format nowadays, so we recommend users to systematically push publishers towards accepting either TEXMACS or PDF documents. For those who do not need to collaborate with colleagues who use LATEX or send their work to publishers, the above "advantage" of LATEX vanishes. This holds in particular for high school education or for less formal documents in academia.

Besides its lack of wysiwyg-ness, (LA)TEX also has another major drawback: there does not exist such a thing as a (LA)TEX document format. In fact, (LA)TEX is really a programming language, but, unlike other programming languages, (LA)TEX does not obey any well-defined formal syntax. This means that (LA)TEX documents can essentially only be processed in a reliable way by (LA)TEX itself. For instance, it is easy to write a (LA)TEX package in which $*$ means "plus" on even pages and "start a new theorem" on odd pages.

The lack of a well-defined (LA)TEX format has caused endless headaches and frustration for the TEXMACS developers, since it makes it impossible to write 100% reliable converters between LATEX and other formats, despite a huge amount of work in this direction. Unfortunately, few LATEX users are aware of this fact and its corollary that their "source code" might be of little use within ten or twenty years from now. It is also plausible that this "locking up in the LATEX non-format" effect has slowed down the development of other mathematical software besides TEXMACS.

Although TEXMACS comes with high quality converters from and to LATEX, the above discussion shows that one cannot consider these mechanisms as 100% reliable black boxes. For those who frequently need to convert documents between TEXMACS and LATEX, there exist various conversion options so as to make the conversion process as smooth as it can get.

Chapter 2
The user interface

The graphical user interface of TEXMACS has been designed to be as intuitive as possible. In particular, recent versions of TEXMACS strive to respect standard conventions that are also used by other software on your computer: basic keyboard shortcuts will be the same and parts of the menus are organized similarly. It is therefore likely that you will learn much of TEXMACS just by using it. Nevertheless, for getting the most out of the software, it may be useful to be aware of some general design principles and conventions that are used inside TEXMACS and this book.

Of course, explanations about user interfaces are easier to remember when you start putting them into practice. You may therefore wish to skip some of the details when reading this chapter for the first time. Once you understand the basics, you may use this chapter as a reference and return to it for more specific questions about the TEXMACS user interface.

2.1 Basic principles

First of all, different users have different backgrounds and distinct preferences when it comes to ergonomics. As long as individual habits do not conflict with each other, TEXMACS strives to support as many of them as possible. Take the example of a simple editing action like the insertion of a new section title. If you are new to TEXMACS, then the Insert ▸ Section menu or the icon is the natural place to look at. However, if you frequently need to insert new sections, then it is more convenient to use a keyboard shortcut. The appropriate one is `ϖ1`, as indicated in the Insert ▸ Section menu. Finally, if you already know LATEX, then you may prefer to use the `\section` command. For this, simply type `\section`, followed by enter `↵`. The `\`-key is heavily overloaded in TEXMACS and can also be used in order to enter other markup elements that do not necessarily exist in LATEX.

A second basic principle is that TEXMACS tries to be as *contextual* as possible. If we permanently advertised all features of TEXMACS on your screen, then blinking buttons would soon drive you crazy. Many features therefore only appear when they are most relevant. One problem with this is that you may miss certain functionalities if you are not looking at the right place. In order to reduce this risk, the interface contextualization is governed by a few simple rules:

1. On the most global level, you may enable or disable certain features in the user preferences or using the Tools menu.

2. On an intermediate level, many of the relevant editing actions depend on the *current editing mode*, which reflects the kind of object you are manipulating: text, mathematics, graphics, programs, etc. There are only a very limited number of editing modes, but they greatly affect the general behavior of the editor.

3. On the finest level, many structured editing actions depend on the *current focus*. The current focus corresponds to the innermost tag at the current cursor position. Since TEXMACS defines hundreds of different markup elements, the Focus menu and other focus-dependent editing actions are highly contextual.

Imagine for instance that we wish to enable a European numbering style of theorems inside TEXMACS (meaning that theorems, propositions, etc. are numbered individually instead of sharing a common counter). In this case, a search for "European numbering" in the documentation (using F1 or Help ▸ Search ▸ Documentation) will lead you to the relevant place. However, it is likely that the very terminology "European numbering style" is new to you. In order to find the appropriate information in that case, the idea is to first insert a theorem, using Insert ▸ Enunciation ▸ Theorem. The contextual Focus ▸ Preferences menu then contains an item European numbering style that you may try out and hope that it does what you want.

2.2 The main window

After starting a new document, the main TEXMACS window will usually look similar to the screenshot in Figure 2.1 below. The screenshot was taken on a computer that runs MACOS, so it might look slightly different on other systems. Furthermore, the TEXMACS menus are displayed at the top of the screen under MACOS, rather than in the window itself, as under GNU/LINUX or WINDOWS.

Let us briefly describe the components of the main TEXMACS window:

- The *title bar* displays the name of your document and also indicates whether the changes you made in your document have been saved on disk.

- The *main icon toolbar* contains context-independent icons that are generally useful (opening, saving, and printing files, opening the user preferences, closing TEXMACS, copying and pasting, search and replace, undo and redo, navigation buttons). Most of the icons correspond to entries in the File, Edit, and Go menus.

- The *mode-dependent icon toolbar* varies with the current editing mode and contains different icons when you are editing mathematical formulas or graphics instead of plain text. Most of these icons are used for inserting new markup elements into your document and correspond to entries in the Insert menu.

- The *focus toolbar* contains icons that depend on the current focus. This toolbar frequently changes when moving your cursor around in a structured document. Most of the icons correspond to entries in the Focus menu.

- The *status bar* contains useful information. The *left footer* generally shows the current mode and font properties, whereas the *right footer* indicates the structure at the current cursor position. The status bar is also used for certain warnings, error messages and interactive input from the user.

- The *search toolbar* is not displayed by default, but gets activated when starting a new search. The bar can be *expanded* into a separate window by pushing the button and closed by pushing . The *replace* and *spell toolbars* work in a similar way and it is likely that future versions of TEXMACS will implement more and more toolbars of this kind.

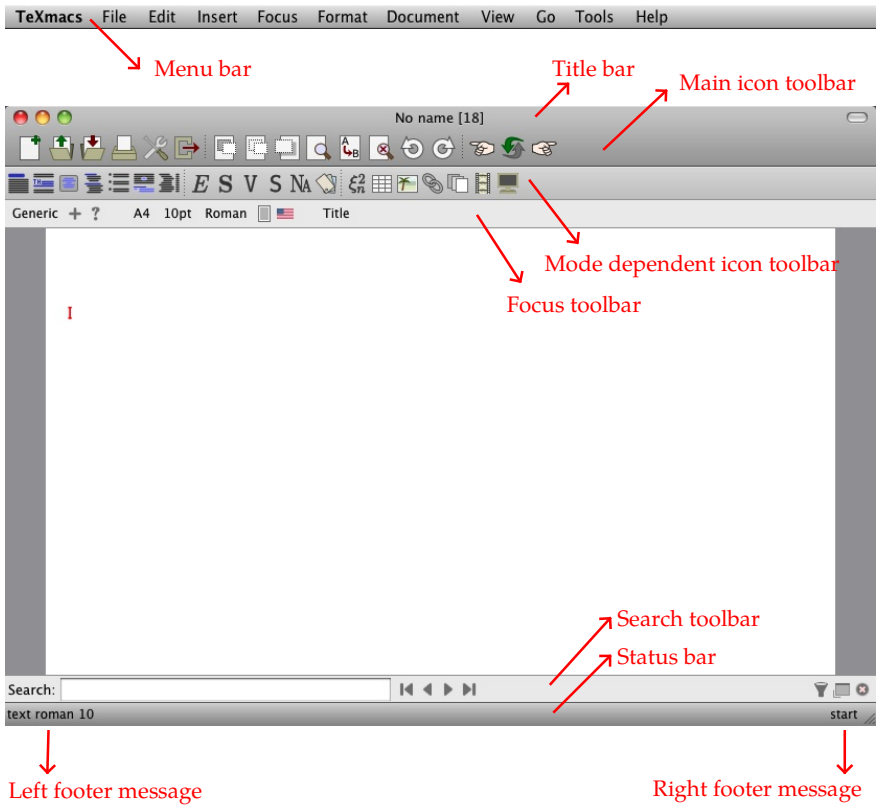

Figure 2.1. The main TEXMACS window. On most computers, the menu bar is part of the TEXMACS windows. Some operating systems such as MACOS rather display them at the top of the screen.

Key	Meaning	EMACS	GNU/LINUX	WINDOWS	MACOS
⇧	Shift	S-	Shift	Shift	shift
^	Control	C-	Ctrl	Ctrl	control
⌥	Alternate	A-	Alt	Alt	option
⌘	Meta	M-	Meta	Windows	command

Table 2.1. Notations for modifier keys in this book.

Not all icon toolbars are always visible. Their visibility can be controlled from within the View menu. It is also useful to know that most of the icons on the toolbars are self-explanatory: when keeping your mouse over the icon during a second or so, a help balloon with a short description of its purpose will appear. The help balloon also indicates an equivalent keyboard shortcut if there exists one.

2.3 Mastering the keyboard

Many editing operations in TEXMACS can most efficiently be executed using *keyboard shortcuts*. There is an abundance of such shortcuts in TEXMACS and new ones can be defined by the user. Before we present general principles that will help you to learn and remember them, let us start with explaining how they will be represented in this book.

2.3.1 Notation for keyboard shortcuts

Keyboard shortcuts are combinations of keys that are pressed either concurrently or in a sequence. In the case when several keys are pressed at the same time, all but one of the keys being pressed are so-called *modifier keys*. Such modifier keys are somewhat larger than the other keys on your keyboard, so your fingers can find them easily. TEXMACS uses four modifier keys: Shift ⇧, Control ^, Alternate ⌥, and Meta ⌘. Most keyboards contain Shift, Control (or Ctrl), and Alternate (or Alt) keys. The "Meta" key usually carries another name, which depends on your keyboard model and operating system. The naming conventions for different keyboard models and operating systems are summarized in Table 2.1.

We represent keyboard shortcuts by small boxes that graphically recall the idea of a key on your keyboard. When several keys are displayed in the same box, they are assumed to be pressed at the same time. Otherwise, we assume that you press them in succession. For instance, ⌘⇧a stands for "simultaneously press the three keys ⌘, ⇧, and a". The shortcut -> means "first press -,

Key	Meaning	Key	Meaning
⇧	Shift modifier	←	Cursor left
⇪	Caps lock	→	Cursor right
^	Control modifier	↑	Cursor up
⌥	Alternate modifier	↓	Cursor down
⌘	Meta modifier	↖	Home
↩	Return (or Enter)	↘	End
⌦	Forward delete	⇞	Page up
⌫	Backspace	⇟	Page down
⎋	Escape	␣	Space
⇥	Tab		

Table 2.2. Notations for special keys in this book.

then release -, next press >". Similarly, ^x ^f stands for "simultaneously press ^ and x, then release both keys (or at least x), next simultaneously press ^ and f.

It is convenient to extend the short graphical representations ⇧, ^, ⌥, and ⌘ for the modifier keys to other special keys on your keyboard. Table 2.2 summarizes the notations that are used in this book.

2.3.2 How to remember keyboard shortcuts

Unfortunately, there are far more useful editing operations than keys on your keyboard. This raises the question of how to associate keyboard shortcuts to editing actions in a way that is easy to remember. TEXMACS achieves this through a mixture of ideas.

First of all, recent of versions of TEXMACS try to respect as much as possible standard conventions used by other software on your computer; see also section 2.3.6. The most common keyboard shortcuts that you already learned for other programs should therefore also work inside TEXMACS.

Secondly, many editing operations are similar or at least share one common characteristic. For instance, we may single out the class of "structured insertions/removals". TEXMACS attempts to group shortcuts for operations in such a class together via a distinctive prefix, modifier key, or modifier key combination. Structured insertions and removals are assigned the common modifier key ⌥, with the consequence that you may insert a new column in a table or a new argument in a list of bibliographic citations using ⌥← or ⌥→. You may also perform the corresponding removals of columns and citations using ⌥⌫ and ⌥⌦.

The next idea is based on the concept of "variants". In the case of a list of similar editing operations, you may use the Tab-key ⇥ to cycle among the different variants. This mechanism is highly useful for entering mathematical symbols: $\in$, $\subset$, and $\prec$ are examples of variants of $<$, which are entered using <⇥, <⇥⇥, and <⇥⇥⇥. The variant mechanism is also used when several editing operations compete for the same key. For example, inside mathematics, we use the keyboard shortcuts ^ and _ for entering superscripts and subscripts, as does (LA)TEX. Inside text, these shortcuts rather insert the symbols "^" and "_", yet superscripts and subscripts can be obtained through the shortcuts ^⇥ and _⇥. The Tab-key ⇥ is highly versatile: it is also used for "Tab-completion" of words and the selection of the appropriate object in the graphical mode, whenever ambiguities arise through superposition. The Tab-key also produces interesting effects when combining it with modifier keys: ⇧⇥ allows you to circle back, whereas ^⇥ is used for "structural variants" such as turning a theorem into a proposition.

There are a few other noteworthy mechanisms that are mode-specific. In math mode, one such mechanism is "graphical concatenation", which allows you to type $\rightarrow$ using ->, $\Longleftrightarrow$ using <==>, and $\leqslant$ using <=. We refer to section 4.2 for further examples and tables with mathematical symbols. In text mode, accented characters are formed using the ⌘ modifier key (e.g. ⌘'e yields "é") and special symbols are obtained using the prefix ⇧F5 (e.g. ⇧F5s yields ß): see section 3.1.

One should also be aware of the fact that the TEXMACS keyboard behavior is highly contextual: it depends on the user preferences, the current mode, as well as the current focus. For instance, inside "math mode", you have special keyboard shortcuts that are handy for typing mathematical formulas, but which are useless in text mode. This allows for a substantial amount of overloading, in the sense that the same keyboard shortcuts serve different purposes in different contexts. For example, after activating a computer algebra system as a scripting language (using Document ▸ Scripts), you may use ^↵ in order to evaluate a mathematical formula. When going through the differences between two TEXMACS documents using the versioning tool, the same shortcut ^↵ is rather used for retaining a selected version.

We finally recall that menu entries indicate equivalent keyboard shortcuts whenever they exist. The keyboard equivalent for an icon on the toolbars appears in a popup menu, when hovering the mouse pointer over the icon for a while. In math mode, the same holds for the submenus with "palettes" of mathematical symbols.

2.3.3 Hybrid commands and LaTeX emulation

TEXMACS uses \ as an important multi-purpose key. By hitting the \ key, you enter the hybrid LATEX/TEXMACS command mode. As soon as you finished

typing a LaTeX or TeXmacs command, the left footer displays something like

```
<return>: action to be undertaken
```

At this stage, hitting the return-key ↵ will execute your command. For instance, in math-mode, typing \ f r a c ↵ creates a fraction.

Technically speaking, TeXmacs first checks whether you typed a LaTeX command. If your command is not recognized as a LaTeX one, then TeXmacs looks whether it is an existing TeXmacs command, macro argument or environment variable (provided by the style file; see chapters 12 and 13). If so, the corresponding macro expansion, macro argument or environment variable is inserted, with the right number of arguments. The \ key is always equivalent to one of the more specific shortcuts ⌘i l (LaTeX command), ⌘i c (TeXmacs command), ⌘i a (macro argument), or ⌘i v (environment variable).

To insert a literal \ (backslash), you can use ⇧F5 \ or \ ⇥.

2.3.4 Common modifier combinations and prefix rules

We have seen that shared modifier combinations or prefixes make it easier to remember keyboard shortcuts for editing operations of a similar nature. The precise behavior heavily depends on the selected "look and feel" in Edit ▸ Preferences ▸ General. I personally own a MAC laptop, on which I use the EMACS look and feel. Under these circumstances, the most common modifier combinations and prefixes are as follows:

- ^. Standard shortcuts, that are similar to shortcuts used by other applications (for the selected look and feel). For example, ^y can be used for pasting text (as in EMACS).

- ⌥. TeXmacs shortcuts that may rely on the current editing mode. For example, ⌥s produces **strong** text in text mode and a square root √ in math mode.

- ⌘. Compound TeXmacs shortcuts. Usually, these shortcuts first indicate the kind of markup to which the command will apply and then specify the specific command. For instance, the ⌘e prefix is used for inserting executable markup, which is useful for writing style files (see chapter 13). One example is the shortcut ⌘e + for inserting an addition.

- ⌘. This modifier key is used in combination with arrow keys and certain other special keys for positioning and resizing objects (see section 10.8).

- ⌘⌥. This modifier combination is used in combination with arrow keys and some other special keys for structured cursor movements (see section 10.6).

Purpose	EMACS	GNOME	KDE	WINDOWS	MACOS
Standard	^	^	^	^	⌘
Mode-specific	⌥	⌥	⌥	⌥	^
TEXMACS-specific	⌘	⌘	⌘	⌘	⌘^
Position & resize	⌘	⌘^	⌘^	⌘^	⌘⌥
Structured navigation	⌘⌥	⌘⌥	⌘⌥	⌘⌥	⌘^
Extra	⌘^	⌥^	⌥^	⌥^	⌥^
Special symbols	⇧F5	⇧F5	⇧F5	⇧F5	⇧F5

Table 2.3. Common prefix rule correspondences for different look and feels. Recall that the shortcuts in this book correspond to the EMACS look and feel.

⌘^. This modifier combination is occasionally used in combination with letters and punctuation symbols for creating some additional easy to remind shortcuts.

⇧F5. This prefix can be used in combination with normal letters for the insertion of special symbols. For example, ⇧F5 s yields ß and ⇧F5 a yields LI. The ⇧F5 prefix is also used for the insertion of "literal characters". For instance, ⇧F5 \ will always produce the \ character, whereas the \ key is used for entering hybrid commands.

If you are using a different computer or look and feel, then some of these prefix rules may change: see Table 2.3.

2.3.5 Substitution rules

On certain keyboards, important modifier keys such as the "Meta" key (or a substitute) are sometimes missing, or reserved for other purposes. For instance, ⌥-based shortcuts are used for entering accented characters and common special symbols under MACOS. Certain important TEXMACS keyboard shortcuts can also be superseded by shortcuts from the operating system. For example, the MACOS application SPACES uses the shortcuts ^←, ^→, ^↑, ^↓, ^1, ^2, ^3, and ^4 to switch between multiple screens.

One solution to the second problem is to change the problematic global shortcuts that got overridden in the responsible applications. For example, SPACES can be configured to use ⌘⌥^ as a prefix instead of ^ (click on the popup menu behind "To switch between spaces" and simultaneously press ⌘, ⌥, and ^). Notice that fn is another key that is not used by TEXMACS.

If you cannot or do not want to change the system-wide shortcuts, or if one or more modifier keys are missing on your keyboard, then you may use the

Shortcut	EMACS	GNOME	KDE	WINDOWS	MACOS
⎋	⌘	⌘	⌘	⌘	⌥
⎋ ⎋	⌥	⌥	⌥	⌥	^
⎋ ⎋ ⎋	^	^	^	^	⌘
⇧⎋	⌘⌥	⌘^	⌘^	⌘^	⌘⌥
⇧⎋ ⇧⎋	⌘^	⌥^	⌥^	⌥^	⌥^
⇧⎋ ⇧⎋ ⇧⎋	⌥^	⌘⌥	⌘⌥	⌘⌥	⌘^

Table 2.4. Keyboard shortcuts for modifier keys or modifier key combinations. Recall that the shortcuts in this book correspond to the EMACS look and feel.

Escape-key ⎋ in order to produce equivalents for the modifier keys ⌘, ⌥, and ^. For instance, under MACOS, ^ is equivalent to ⎋ ⎋. Hence, the TEXMACS shortcut ^→ can also be obtained by typing ⎋ ⎋ →, which may coexist with the SPACES shortcut ^→. Table 2.4 shows the modifier key combinations that can be obtained using ⎋.

2.3.6 Standards conformance

TEXMACS attempts to be as standards-conforming as possible for each of the supported look and feels in Edit ▸ Preferences ▸ General. However, there are a few general situations in which TEXMACS reserves keyboard shortcuts for the sake of user-friendliness:

- The function keys F5–F12 are reserved for special actions.
- Most operating systems admit a "principal modifier key" for forming keyboard shortcuts (e.g. ^ for the EMACS look and feel) as well as a "secondary modifier key" for a few other shortcuts (e.g. the Windows key under WINDOWS and ⌥ under MACOS). The remaining free modifier key (e.g. ⌥ for the EMACS look and feel) is reserved for TEXMACS.
- TEXMACS contains many keyboard macros involving one or more modifier keys and the special keys ←, →, ↑, ↓, ↖, ↘, ⇞, ⇟, ⌫, ⌦, ␣, ⇥, and ↵. The behavior of shortcuts of this kind is occasionally non-standard.

The TEXMACS-specific shortcuts are rarely in conflict with standard conventions. Nevertheless, in Table 2.5, we have listed a few more or less standard shortcuts, that might work in other applications, but that will usually not work inside TEXMACS.

Look & feel	Shortcut	Alternative	Meaning
Emacs	F10		Show menu bar in window
Emacs	⌘!		Shell command
Emacs	⌘' / ⌘` / ⌘^		Needed for TEXMACS accents
Emacs	⌘/ / ⌘\ / ⌘: / ⌘;		
Emacs	⌘← / ⌘→	^← / ^→	Move word back/forward
Emacs	⌘a / ⌘e	^↑ / ^↓	Move paragraph back/forward
Emacs	⌘b / ⌘f	^← / ^→	Move word back/forward
Emacs	⌘l / ⌘t		Locase/transpose words (not impl.)
Windows	F5		Refresh/Switch to next pane
Windows	F6 / ^F6 / ^⇧F6		Switch to next/previous pane/tab
Windows	^␣		Remove formatting
Windows	^⇥		Switch to next child window
Windows	^⌫ / ^⌦		Delete word
MacOS	^F5 / ^F6 / ^⇧F6		Move focus to toolbar/panels
MacOS	^F7		Override keyboard access mode
MacOS	F9 / F10		Tile or untile windows
MacOS	F11 / F12		Hide or show windows/dashboard
MacOS	⇥ / ⇧⇥		Navigate through controls
MacOS	^⇥, ^⇧⇥		Move focus within control groups
MacOS	^␣ / ⌘^␣		Toggle between input sources
MacOS	^← / ^→	⌘⌥← / ⌘⌥→	Move one cell left/right in table
MacOS	^↑ / ^↓	⌘⌥↑ / ⌘⌥↓	Move one cell up/down in table
MacOS	↖ / ↘	⌘↑ / ⌘↓	Move to start/end of document
MacOS	⌥⇞, ^↑, ^⇞	⇞	Page up
MacOS	⌥⇞, ^↓, ^⇞	⇞	Page down
MacOS	^a / ^e	⌥↑ / ⌥↓	Move to start/end of block

Table 2.5. Shortcuts that might work in other applications, but usually not in TEXMACS.

2.4 Organization of the menus

Recall that we use a sans serif font for menu entries in this book. For instance, the menu entry for creating a new file is denoted by File ▸ New. The top-level menus of TEXMACS were designed according to the following principles:

- The File menu contains the most common operations on files: opening, saving, closing, and printing of documents, as well as conversions to other formats.
- The Edit menu is also pretty standard and contains the most basic editing operations: copying and pasting of text, undoing and redoing changes, search and replace, as well as spell checking. TEXMACS also allows you to copy and paste fragments of documents that use other formats. For example, when copying and pasting a selection from your favorite web browser, most of the formatting properties will be preserved.

- The Insert menu is mode-dependent and used for the insertion of structural markup. In text mode, it allows you for example to insert section titles, theorems, numbered lists, emphasized text, etc. In math mode, it allows you to insert fractions, wide accents, matrices, etc.

- The Focus menu corresponds to editing operations that act on the current focus; see section 2.7. Assuming that your cursor is inside a section title, it allows you to transform the section into a subsection, to toggle the numbering, to jump to other sections, etc.

- The Format menu is dedicated to visual fine-tuning of the layout and presentation of pieces of your document. You may change the font, the color, the alignment and spacing of paragraphs, insert extra whitespace, control line and page breaking, adjust the position and size of pieces of text, etc.

- The Document menu allows you to specify the global properties of your document, such as its style and language, the page size and color, the main font, metadata, etc.

- The View menu is used for controlling the presentation of the TEXMACS windows. It specifies which toolbars are visible and the zoom factor that is applied to your document. It also allows you to enter and leave presentation mode.

- The Go menu allows you to change the document that is visible in the main window. Its Back and Forward entries can be used for browsing a collection of TEXMACS files, such as the built-in documentation.

- The Tools menu contains miscellaneous utilities. It also allows you to enable additional tools for debugging, code development, versioning, etc. When enabling the versioning tools, for example, a new top-level Version menu will appear with additional document versioning facilities.

- The Help menu gives access to the built-in documentation. TEXMACS includes a manual, a reference guide, and more technical information about its implementation. It is also possible to search for specific keywords inside the documentation.

2.5 Dialogue windows

In old versions of TEXMACS, most of the complex editing operations were accomplished through the menus. More recently, a big part of the user interface has been redesigned so as to make more extensive use of dialogue windows. For instance, fonts can be picked through a special font selector and the page layout can be modified using a dedicated widget.

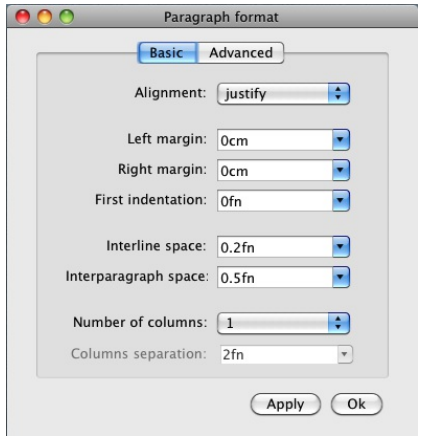

Figure 2.2. The dialogue window for changing the layout of paragraphs.

In order to explain how dialogue windows work, it is convenient to extend the notation that we introduced for menus. Let us see how this works on an example. In Figure 2.2 below, we have displayed the dialogue window for modifying the layout of paragraphs. This window is opened using Format ▸ Paragraph. Then Basic ▸ Alignment ▸ justify corresponds to the selection of justify for the paragraph alignment in the list of choices next to Alignment. The notation Format ▸ Paragraph ▸ Basic ▸ Alignment ▸ justify stands for the combined action of opening the dialogue window and selecting the justified paragraph alignment.

Some users do not like their screen to be cluttered by dialogue windows and prefer to use the menus as much as possible for complex editing operations. TEXMACS provides a special user preference for this: Edit ▸ Preferences ▸ General ▸ Complex actions ▸ Through the menus.

2.6 User preferences

Many functionalities of TEXMACS can be configured according to the preferences of the user. Some of the most important user preferences can be changed in Edit ▸ Preferences. However, many settings are relevant for specific tools or markup elements only. Changing such settings is done in the preference menu of the corresponding tool or in an appropriate context-dependent preference menu. Some of your likings are even determined automatically by TEXMACS. For example, the size and position of the TEXMACS window are remembered when closing and restarting the application.

In Edit ▸ Preferences ▸ General, you will find a few important global user preferences that control the general behavior of the user interface:

Look and feel. Each particular operating system comes with its own user interface conventions for common editing operations like "cut and paste". By default, TEXMACS tries to conform to the conventions of your

operation system, but you are free to choose another "look and feel". You may also choose the Emacs look and feel in order to enable many of the GNU EMACS keyboard bindings.

User interface language. Whenever appropriate translations are available, TEXMACS will use the specified language for the user interface (e.g. the menus, dialogue windows, etc.).

Complex editing actions. By default, TEXMACS opens dedicated dialogue windows for accomplishing complex editing operations. Some users prefer not to clutter their screen with other windows and rather perform such operations through the menus.

Interactive questions. Some editing operations require extra input from the user. Such additional feedback can either be entered in popup windows or via the footer.

Details in menus. TEXMACS offers many possibilities. For some applications (e.g. high school education), one may wish to simplify the user interface by only exposing a subset of it.

Buffer management. Many applications create a separate window for every new document that you open. Some users prefer the GNU EMACS convention of using a single window in which you may switch between multiple documents, also called *buffers*.

2.7 The current focus

TEXMACS offers a wide range of features. The main mechanism through which only the most relevant functionalities in a given context are exposed to the user is based on the concept of *current focus*. Recall that TEXMACS documents are *structured*. At a given cursor position, we find ourselves inside a list of structures (also called *tags* or *markup elements*), ordered from innermost to outermost. For instance, assume that our cursor is behind the "w" of "brown" inside the following quotation:

> The quick brown fox
>
> Jumps over the lazy dog.

Then we are inside two structures: a change of color into brown and a quotation. TEXMACS displays this information in the right footer. It also indicates the innermost structure through a cyan box and all other surrounding structures through light grey boxes. The innermost structure at the current cursor position is called the *current focus*.

Remark. When selecting text, the current focus rather becomes the innermost structure that contains the selection. Here it is useful to know that the ^␣ keyboard shortcut enlarges the current selection to its surrounding structure. Through the repeated use of this shortcut, any of the tags that contain the cursor can thus be selected efficiently and act as the *focus tag*.

The main editing operations that correspond to the current focus are accessible through the Focus menu and the *focus toolbar*. The items in this menu and on this toolbar are highly context-dependent by design. Whenever appropriate, they allow you to perform the following types of actions:

Navigation. The icons ⤒, ▲, ▼, and ⤓ (which correspond to the menu entries First similar, Previous similar, Next similar, and Last similar) can be used for jumping to similar tags in your document. Assuming that you are inside a theorem, clicking on ▼ will for example bring your cursor to the next theorem, proposition, lemma or corollary.

Insertion and removal. Certain tags (such as bibliographic citations, "switches" in presentations, tables, etc.) admit a variable number of arguments. Using the icons ⊕ and ⊕ (or the menu items Insert argument before and Insert argument after), you may add new arguments. Similarly, ⊖ and ⊖ (or Remove argument before and Remove argument after) can be used for the removal of arguments. In the case of tables, the above icons (or menu entries) can be used for the insertion and removal of columns. In addition, you may use ⊕, ⊕, ⊖, and ⊖ for the insertion and removal of rows.

Variants. Certain tags admit variants or alternative states. Theorems can be numbered or not for example and they behave much alike propositions, lemmas and corollaries. Other markup elements can be folded or unfolded (this is useful for presentations and exercises). Now the focus menu and toolbar always indicate the name of the tag that corresponds to the current focus. If this tag has interesting variants, then a submenu with the list of variants is proposed under its name. Similarly, the icons Ⅳ, ▭, ▭, and ▭ can be used to toggle the numbering or state of the tag, whenever applicable.

Exit. The icons ⇤ and ⇥ (or Exit left and Exit right) can be used in order to exit the focus tag at the left or the right.

Destroy. The focus tag can be destroyed using ✘, Delete or ^⌫. More precisely, the tag and its arguments will be replaced by the argument that contains the cursor.

Customize. The presentation of the focus tag can be customized via 🔧 or Preferences. Notice that customizations of this kind will act on the style properties and preamble of your document and not on the user preferences, strictly speaking. In other words, alternative settings will only impact on the document that you are editing and not on any other

documents that you may open in the future. The following types of customizations are most common:

- Some tags come with one or more *style options* that correspond to customized renderings through the use of special style packages. For example, theorems may be told to use an alternative "European numbering style" using Focus ▸ Preferences ▸ European numbering style.
- Various *style parameters* (also called *environment variables*) may influence the rendering of a tag. For example, list environments come with two style parameters *item-hsep* and *item-vsep* that control the horizontal and vertical spacing of list items. The Preferences menu comes with an entry for specifying each style parameter of the focus tag.
- All non-built-in TEXMACS tags are defined using macros in the TEXMACS style packages or directly by the user. If applicable, then you may use Focus ▸ Preferences ▸ Edit source in order to jump to the relevant place in the style package that defines the focus tag. You may also edit the corresponding macro inside a special dialogue window using Focus ▸ Preferences ▸ Edit macro; see section 12.1 for more information.

Documentation. Contextual help about the focus tag can be obtained using ? or Describe.

Arguments. Certain tags come with one or more "hidden" arguments that are invisible in the document itself and therefore require special mechanisms for being edited. Typical examples of hidden arguments are the color "brown" in the example at the beginning of this section or the names of labels and references. In many cases, such hidden arguments are short names, colors or lengths, that can be edited via extra input fields on the focus toolbar. In general, tags with hidden arguments can be deactivated and reactivated using or Show hidden. Alternatively, you may place the cursor just behind the tag and press ⌫ in order to deactivate it. After editing the hidden arguments, you may reactivate the tag using ↵.

In the case when the current cursor is not inside any particular tag, the current focus is on the whole document. This happens for instance when the cursor is at the extreme top-left position in the document. Some of the most important entries of the Document menu are then exposed more directly on the focus toolbar:

Style. On the extreme left, you may specify the style of your document, and select any additional style packages to be used. We recall that style packages that are really style options for particular tags should rather be specified via the Focus ▸ Preferences menus for these tags.

Page style. The current page size is indicated on the focus bar and can be modified. Similarly, the orientation and number of columns are indicated through one of the icons , , , , , or . These settings can also be changed by clicking on the displayed icon.

The default page rendering mode is Paper, which corresponds to the way your document will be printed. However, in this mode, the positions of page breaks may heavily change while you are typing, which can be annoying. This can be avoided by using the alternative Papyrus mode, in which your document is displayed on a single page of fixed width and flexible height. The Papyrus mode also allows for faster typesetting, which makes the editor more responsive when your document gets large.

Font. The main document font and its size are indicated and can be modified from the focus bar. The proposed main fonts are on a restricted shortlist of fonts that are particularly well supported by TEXMACS.

Language. The icon with the little flag indicates the main document language. This language is used by the typesetter for punctuation and hyphenation of words. It is also relevant for the spell checker.

Section. If your document contains several sections, then the focus bar also indicates the current section. By clicking on it, a list with all sections appears, which allows you to jump to any other section inside the document.

2.8 Finding documentation

The easiest way to find documentation on a topic inside TEXMACS is to search for it using F1 or Help ▸ Search ▸ Documentation. Search queries are similar to those for popular search engines on the web: just type one or more keywords separated by spaces and TEXMACS will return a list of matching pages in the documentation, ordered by expected relevance.

We also recall from the previous subsection that Focus ▸ Describe provides contextual help on the current focus tag. If the focus is on the entire document, then Focus ▸ Help will provide help on the current document style and any selected style packages. Minimal contextual explanations of icons and menu entries are also available via help balloons that appear as soon as you hover your mouse pointer over them for a short while.

The bulk of the built-in documentation on TEXMACS is available from the Help menu. The available material is organized as follows:

- In the Help ▸ Manual menu, you find the main documentation about how to use TEXMACS. The manual is quite complete, but not very detailed.

- In the Help ▸ Reference guide menu, you find more detailed documentation about the TEXMACS format, its primitives and environment variables, and the standard style files.
- TEXMACS plug-ins may (or may not) include documentation. Any available auto-documentation of this kind is made available through the Help ▸ Plug–ins menu.
- The Help ▸ About menu contains more information about TEXMACS, like information about its authors, how to contact us, and changes in different versions of the program.
- More information is available for developers (Help ▸ Interfacing, Help ▸ Scheme extensions, Help ▸ Future plans) and anyone who wants to support the development of TEXMACS (Help ▸ Help us).

In addition, book versions of the existing documentation are available in Help ▸ Full manuals. An interesting feature of TEXMACS is that any documentation page can be expanded into an article or a full manual using Help ▸ Full manuals ▸ Compile article or Help ▸ Full manuals ▸ Compile book. For instance, if you start browsing the manual using Help ▸ Manual ▸ Browse, and click on Help ▸ Full manuals ▸ Compile book, then TEXMACS will compile the user manual [25]. Please be patient… Compiling a full manual may take a few minutes…

The Internet provides another valuable source of information. On the TEXMACS web site `http://www.texmacs.org`, you may find videos, tutorials and research papers about TEXMACS. The FAQ and the mailing list archives also contain answers for many questions by users. Do not hesitate to subscribe to the mailing lists and ask your own questions there, or help others out!

Chapter 3

Ready, steady, go!

3.1 Typing ordinary text

When it comes to typing ordinary text, we have nothing extraordinary to say: most of the time, TEXMACS will behave similarly to other programs on your computer. Nevertheless, there are a few points worth mentioning for entering certain symbols and for getting the punctuation, spacing, and other typographic details right.

Special symbols

TEXMACS reserves the \, $, and " keys for entering special commands, mathematical formulas, and quotes. The literal symbols \, $, and " can be produced using the shortcuts ⇧F5 \, ⇧F5 $, and ⇧F5 ", or alternatively using \ ⇥, $ ⇥, and " ⇥. The ⇧F5 keyboard prefix is also used for entering other special symbols; see Table 3.1.

Punctuation

TEXMACS assumes that you are familiar with basic punctuation rules for the language you are writing in. English rules stipulate that each of the principal punctuation symbols

. , : ; ! ?

should be followed by a space, except at the end of a paragraph, but not preceded by any spaces. French rules on the contrary require you to also put a space before these symbols, except for periods and commas.

Assuming that you correctly follow these language-specific rules, TEXMACS takes care of the more precise spacing conventions around punctuation symbols. For example, periods are followed by slightly larger spaces than commas. This makes it easier for the eye to spot separate sentences. Similarly, French typographers use as slightly smaller “thin” space before colons, semicolons, exclamations marks, and question marks.

Shortcuts							
⇧F5 a	æ	⇧F5 A	Æ	⇧F5 a e	æ	⇧F5 A E	Æ
⇧F5 o	ø	⇧F5 O	Ø	⇧F5 o e	œ	⇧F5 O E	Œ
⇧F5 s	ß	⇧F5 S	S				
⇧F5 !	¡	⇧F5 ?	¿	⇧F5 p	§	⇧F5 P	£

Table 3.1. Typing special characters using the ⇧F5 keyboard prefix.

Whitespace	
negative space xy	\ ! ↵
no space xy	
tiny space x y	\ , ↵
small space x y	\ : ↵
thin space x y	\ ; ↵
normal space x y	␣
quad space x y	␣ ⇥ ⇥ or \ q u a d ↵
double quad space x y	\ q q u a d ↵

Table 3.2. Common amounts of whitespace and how to obtain them.

There are a few exceptional cases that require additional care. The period sign is also used inside abbreviations, for example, in which case it should be followed by a normal space (instead of the slightly larger space between sentences). We recommend that you systematically "tag" abbreviations using Insert ▸ Content tag ▸ Abbreviation or the keyboard shortcut ⌥a. This in particular allows TEXMACS to adjust the spacing.

Whitespace

We have seen above that TEXMACS automatically determines the appropriate amount of whitespace to be inserted for each space character in the document. However, TEXMACS does not force you to enter a space after, say, a period. This flexibility is useful whenever periods are used for other purposes as ending sentences (e.g. inside numbers 1.234, abbreviations, names of websites `www.texmacs.org`, etc.).

It is generally a bad idea to insert several spaces in a row, just for the visual effect of obtaining a somewhat larger space. In the case when you need an explicit whitespace of, say, one inch, you should rather use Format ▸ Whitespace ▸ Rigid and enter `1 in` at the prompt. Certain amounts of whitespace are very common for tweaking the presentation of text; see Table 3.2 for some useful shortcuts to this effect. For more details about horizontal whitespace, we refer to section 7.5.1.

Shortcuts			
`	‘	'	’
` `	“	' '	”
< ⇥	‹	> ⇥	›
< <	«	> >	»
⇧F5 "	"	, ,	„

Table 3.3. Keyboard shortcuts for specific quotes.

When entering several spaces in a row, the default behavior of TEXMACS is to glue them together into a single big space. More precisely, hitting n times ␣ will produce a space with the width of $n-1$ “quad” spaces. Using Edit ▸ Preferences ▸ Space bar in text mode ▸ Allow multiple spaces, it is also possible to mimic the behavior of most other text editors: hitting n times ␣ will simply insert n spaces. Similarly, Space bar in text mode ▸ No multiple spaces forbids multiple spaces altogether: pressing ␣ any number of times always inserts a single space.

Hyphens and dashes

TEXMACS provides three types of hyphens and dashes (namely -, –, and —), which are entered using the shortcuts -, - -, and - - -. The shortest ones, called *hyphens*, are used for hyphenation and in order to join words (e.g. high-school teacher, Marie Skłodowska-Curie, etc.). So-called *en dashes* (–) are typically used for ranges (see pages 256–65536), relations, and connections (the Stone–Weierstrass theorem). The longest *em dashes* (—) mostly serve as substitutes for parentheses or colons (Archimedes—the great scientist—is also known for letting his bath overflow). TEXMACS finally provides a *non-breaking hyphen* after which line breaks are forbidden; you may enter this character using - ⇥.

Quotes

When you press the " key, an appropriate opening or closing quote will be inserted. The quote character is chosen according to the current language and the surrounding text. If the default quoting style is not appropriate for some reason, then you can change it in Edit ▸ Preferences ▸ Keyboard ▸ Automatic quotes. You can also insert specific quotes using the keyboard shortcuts in Table 3.3.

In most languages, opening quotes are preceded but not followed by a space, whereas closing quotes are followed but not preceded by one. The French use a thin space after opening *guillemets* « and before closing ones ». The spacing may need to be adjusted for nested quotations, such as “Archimedes exclaimed ‘Eureka!’ ”. In order to detach the single nested closing quote from the surrounding double closing quote, we manually inserted a small space using \ ; ↵.

Font	‘ ’	‘ ‘ ’ ’	‘ ⇥	’ ⇥	" ⇥
TEXMACS Computer Modern	‘ ’	“ ”	‘	’	"
TEX Gyre Termes (Times)	‘ ’	“ ”	`	'	"
TEX Gyre Pagella (Palatino)	‘ ’	“ ”	`	'	"
DejaVu	‘ ’	“ ”	`	'	"

Table 3.4. Rendering of simple quotes, double quotes, backquotes, apostrophes, and quotation marks in various fonts.

The single quotes ‘ ’ and the double closing quotes ” are easily confused with the backquote `, the ambidextrous apostrophe ' and the ambidextrous quotation mark ". Older fonts may even display some of these symbols in the same way, whereas modern UNICODE fonts tend to propose different glyphs (see Table 3.4). “Genuine” single and double quotes are used for quotations in ordinary text; they can for instance be entered using the natural shortcuts ‘, ’, ‘ ‘, and ’ ’, as explained above. The quotation mark ", the apostrophe ', and the backquote ` are important for programmers, who tend to use notations like `"Hello world"` or `'Hello world'` for representing textual string data; you may enter them using ‘ ⇥, ’ ⇥, and " ⇥. Notice that the keyboard shortcuts are “reversed” inside computer programs (e.g. " produces the quotation mark ", whereas " ⇥ produces ”).

Line breaking

Ordinary text is built up from paragraphs, each of which contain one or more sentences. Most paragraphs do not fit on a single line, so TEXMACS automatically breaks them into several lines. Sometimes, you may wish to control more precisely how this is done. There are several circumstances in which line breaks should be avoided:

- After initials of names: do not break after “A.” in “A. Einstein”.
- After very short words: I personally tend to avoid breaks after single-letter words such as the article “a”.
- Before references such as the number “3” in “Chapter 3”.
- Inside mathematical text, it is often considered bad taste to start a line with a mathematical formula.

TEXMACS provides a special “non-breaking space” character that can be used instead of an ordinary space in order to prevent line breaks at a specific position. This character is entered using ␣ ⇥ and rendered on the screen using a barely visible steel-blue hyphen. You may also prevent all line breaks inside a selection of text by turning it into a “group” using Format ▸ Break ▸ Group ▸ Horizontal or ⌘=.

Conversely, you may insist on having a line break at a certain point using Format ▸ Line break. Notice that it is generally better to forbid breaks at places that are inappropriate for sure rather than to force them at places

that look nice at a time t. Indeed, as soon as you make further modifications inside your text, change the document style, or export it to another format, the nice looking break often becomes your worst possible choice.

Hyphenation

Long words may be hyphenated during the line breaking process. The hyphenation rules are determined by the language in which you wrote your text (as selected using Document ▸ Language or Insert ▸ Language).

Again, it is sometimes desirable to have more control over the hyphenation process. Now the above mechanism of "grouping" does not only forbid line breaks: it also prevents any words from being hyphenated. It is typically a good idea to turn off hyphenation inside names of persons (try hyphenating your own name).

For a given word, TEXMACS also allows you to explicitly specify its hyphenation: select the word and use Format ▸ Hyphenate as. You will be prompted for a hyphenation of the word. Just enter short dashes at the places where hyphenations should be allowed.

3.2 Foreign languages

In the past, typing foreign text on a computer was often a complex task, especially for languages with other alphabets or scripts, such as Russian or Chinese. The UNICODE standard was designed to simplify this task and provide a "universal" computer alphabet for all known languages in the world. Nowadays, most operating systems comply with this standard and provide built-in mechanisms for entering text in UNICODE format, while supporting a large number of languages and keyboard models.

TEXMACS is also compatible with a large part of UNICODE (notable exceptions are Arab and Hebrew which are written from right to left), even though another encoding is used internally. For the increasingly rare cases in which it is impossible or difficult to enter certain characters using the built-in facilities of the operating system, TEXMACS provides a few additional mechanisms of its own, that we will describe below.

Note that you can set the text language for the whole document with Document ▸ Language. It is also possible to select another language for a single word or fragment of text using Insert ▸ Language. TEXMACS uses language information for a variety of purposes, such as hyphenation, spell checking, available keyboard shortcuts, punctuation and spacing rules, etc. It is also possible to specify an alternative language for the graphical user interface in Edit ▸ Preferences ▸ General ▸ Language.

Shortcut		Example		Shortcut		Example	
⌘'	Acute ´	⌘' e	é	⌘`	Grave `	⌘` e	è
⌘^	Hat ˆ	⌘^ e	ê	⌘"	Umlaut ¨	⌘" e	ë
⌘~	Tilde ˜	⌘~ a	ã	⌘C	Cedilla ¸	⌘C c	ç
⌘U	Breve ˘	⌘U g	ğ	⌘V	Check ˇ	⌘V s	š
⌘O	Above ring ˚	⌘O a	å	⌘.	Above dot ˙	⌘. z	ż
⌘H	Hungarian ˝	⌘H o	ő				

Table 3.5. Typing accented characters.

3.2.1 Latin languages

Among the languages with Latin alphabets, the English language is particularly simple in the sense that its letters do not carry any accents. For other "Latin" languages, you may enter accented letters using the ⌘ prefix. For example, "é" is obtained by typing ⌘' e, and "à" by typing ⌘` a. See Table 3.5 for the complete list of available accents, with the corresponding keyboard shortcuts and examples. Note that these shortcuts are available regardless of the current language setting.

In Table 3.1 we already showed how to enter several other special letters and symbols using the prefix ⇧F5. Further keyboard shortcuts are available for specific languages only: see Table 3.6. Some of these shortcuts override general shortcuts from Table 3.1. Using the "variant" mechanism (see section 2.3.2), it is possible to recover the overridden behavior. For example, inside Hungarian text, you can still obtain "ø" through the shortcut ⇧F5 o ⇥.

3.3 Starting a new document

In TEXMACS, the first thing that you usually do when starting a new document is to choose an appropriate *document style*. This can be done using Document ▸ Style or directly via the focus bar.

Hungarian	
⇧F5 o	ő
⇧F5 O	Ő
⇧F5 u	ű
⇧F5 U	Ű

Spanish	
! ⇥	¡
? ⇥	¿
! `	¡
? `	¿

Polish			
⇧F5 a	ą	⇧F5 o	ó
⇧F5 A	Ą	⇧F5 O	Ó
⇧F5 c	ć	⇧F5 s	ś
⇧F5 C	Ć	⇧F5 S	Ś
⇧F5 e	ę	⇧F5 x	ź
⇧F5 E	Ę	⇧F5 X	Ź
⇧F5 l	ł	⇧F5 z	ż
⇧F5 L	Ł	⇧F5 Z	Ż
⇧F5 n	ń	⇧F5 z ⇥	ź
⇧F5 N	Ń	⇧F5 Z ⇥	Ź

Table 3.6. Language-specific text shorthands.

TEXMACS includes a few general purpose styles such as article, beamer, book, Education ▸ exam and letter, as well as a selection of more specialized styles. For example, in the menus Article and Book, you will find additional styles for writing articles and books, implementing specific layouts that are required by certain journals or publishers.

Assuming that you selected a document style such as "article" or "book", the next logical actions are to enter a title, the authors, an abstract, etc. Of course, the precise kind of information you provide depends on the style: in the case of an exam, you may wish to add the subject, the level, or the name of a classroom; the title page of a laptop presentation typically becomes more attractive if you add some artwork.

3.3.1 Entering the title

Let us detail how to enter title information in the case of an article. Such information should be regarded as a block of data. The specific order and way in which the data are presented is determined by the document style, although TEXMACS provides a few options for customization.

You may use Insert ▸ Title ▸ Insert title in order to create a block with title information. For a newly created document, you may also use the Title button on the focus bar. The main data field of the title block is the title itself. Extra data fields can be inserted from the Focus menu or the + icon on the focus toolbar. For instance, Focus ▸ Author allows you to insert one or more authors. You may also specify the creation date of your document using Focus ▸ Date (arbitrary dates) or Focus ▸ Today (the current date). Miscellaneous fields with a customized layout can be added using Focus ▸ Miscellaneous. For notes with people to be thanked or grant information, you should rather use Focus ▸ Note.

Many document styles do not only display the title at the top of the first page: often this information also shows up in the headers of individual pages. If a title is very long, then this may cause the headers to run off the pages. In such cases, you may specify an alternative *running title* using Focus ▸ Running title. The same remark applies to the authors: if your paper has many authors, then you should specify an abbreviated *running author* for page headers using Focus ▸ Running author.

For each author that you enter using Focus ▸ Author, you provide additional data via the items below Focus ▸ Author (alternatively you may use the + icon next to Author on the focus bar). Nowadays it is customary to provide an affiliation for the author (Focus ▸ Author ▸ Affiliation) as well as an email address (Focus ▸ Author ▸ Email). Optionally, you may also wish to point to a homepage (Focus ▸ Author ▸ Homepage).

Miscellaneous data fields and further notes can be inserted in a similar way as for titles.

Optimale Pareto-efficiëntie in genotssamenlevingen[1]

JAN KLAASSEN
Afdeling wiskunde
Katholieke Universiteit Tilburg

Email: jan@wis.kunt.nl

ADA P. NOOT-MIES
Afdeling econometrie
Universiteit van Amsterdam

Email: ada@uva.nl

PIET SNOT
Afdeling wiskunde
Katholieke Universiteit Tilburg

Email: piet@wis.kunt.nl

Optimale Pareto-efficiëntie in genotssamenlevingen[2]

JAN KLAASSEN[a], PIET SNOT[b]
Afdeling wiskunde
Katholieke Universiteit Tilburg
a. Email: jan@wis.kunt.nl
b. Email: piet@wis.kunt.nl

ADA P. NOOT-MIES
Afdeling econometrie
Universiteit van Amsterdam

Email: ada@uva.nl

Optimale Pareto-efficiëntie in genotssamenlevingen[3]

JAN KLAASSEN[ab], ADA P. NOOT-MIES[cd], PIET SNOT[ae]
a. Afdeling wiskunde
Katholieke Universiteit Tilburg
c. Afdeling econometrie
Universiteit van Amsterdam
b. Email: jan@wis.kunt.nl
d. Email: ada@uva.nl
e. Email: piet@wis.kunt.nl

Figure 3.1. Three ways to cluster title information: [1] none; [2] by affiliation; [3] maximal.

Notice that data fields need not be unique. You may for example specify more than one affiliation or email address for an author, or attach several miscellaneous data fields to the title.

Whenever several authors share the same affiliation, publishers may require you to save space by factoring out such common information. TEXMACS can do this automatically for you; the desired amount of "clustering" can be specified using the following options in the menu on the focus bar:

No clustering. Do not factor out any data.

Cluster by affiliation. Group authors by their affiliations.

Maximal clustering. Factor out as much common data as possible.

The effect of these options is illustrated in Figure 3.1.

3.3.2 Entering the abstract

The abstract of an article can be entered using Insert ▸ Title ▸ Abstract. This action really inserts *abstract blocks* which are similar to the title blocks from the previous section: besides the abstract itself, you may also insert several metadata such as keywords and subject classifiers.

More precisely, with the current focus on the abstract, a list of keywords can be entered using Focus ▸ Keywords: you start typing the first keyword and then insert additional keywords using Focus ▸ Insert argument after or ↹. Each topic (mathematics, physics, computer science, etc.) admits its own specific subject classification system. For example, mathematicians often use the A.M.S. subject classification (Focus ▸ A.M.S. subject class), whereas computer scientists tend to prefer the A.C.M. computing class (Focus ▸ A.C.M. computing class). In TEXMACS, we aim at supporting the most frequent classification schemes, and new schemes can easily be added on request.

We notice that certain metadata can also be entered using Document ▸ Metadata. This alternative mechanism attaches the metadata even more directly to the document. Currently, you may specify a title, an author, and a subject in this way, while TEXMACS attempts to automatically determine as many fields as possible. This kind of metadata is for instance used when exporting a document as a PDF file.

3.4 Subdividing your document into sections

Long documents are usually subdivided into chapters and/or sections (laptop and slide presentations rather consist of a sequence of "screens" or slides). As a structured text editor (see section 1.3), TEXMACS provides special markup for sections. We recommend to use this markup and to refrain from manually typesetting section titles using, say, a large underlined font.