A short manual for TeXworks

Alain Delmotte*

August 4, 2009

Contents

*with the help of: Jonathan Kew, Stefan Löffler, David J. Perry, Joel C. Salomon and Joseph Wright; but the errors are still mine!

1 Introduction

Donald E. Knuth decided to create a new typesetting system, which would be called TEX, because there had been a change in the printing system used for the volumes of his book *The Art of Computer Programming*, Knuth found the result of the new system awful.

The goal of TEX was then to have a system which would always produce the same documents independently of the actual machine. Knuth also designed the *Computer Modern* family of typefaces and the METAFONT language for font description.

The work initiated in 1977 was finished (the languages were "frozen") in 1989. TEX and METAFONT are not evolving any more except for minor bug fixes (TEX versions are numbered following the decimals of π – now 3.1415926 – and METAFONT the decimals of the number "e" – now 2.718281).

TEX provides basic tools (commands/instructions/"primitives") to define typesetting; almost every detail has to be defined, but the language allows the creation macros for repeatedly used constructs. So collections of macros are loaded through format files, these are pre-compiled large macro collections.

Knuth created an original default format (more or less 600 commands), it is called *Plain TEX*. This facilitates creating documents.

The most widely used format is LaTeX (Leslie Lamport, 1985), which provides more global commands and structures for documents (article, book,...) allowing easier and faster work, but sometimes with loss of flexibility due to the more or less rigid framework. Other formats are $\mathcal{AMS}$-TeX, $\mathcal{AMS}$-LaTeX, ConTeXt, having each specific goals and advantages (and drawbacks).

To extent the format, one loads "packages" which are collection of macros specific to some aspects of typesetting.

From its specification, late 60s, till now, last version March 2008, the TeX family had to evolve to take into account the developments in the typesetting world outside TeX.

Some of the problems to answer were/are 1) taking into account other languages with "alphabets" larger than the ASCII one or with non latin characters altogether, 2) having more fonts, there is not much variety in the fonts created with METAFONT (few font creator use it), 3) creating documents in other formats than the normal DVI [1], 4) using the rich possibilities of other typesetting systems and formats like PostScript and PDF, 5) having more calculation and scripting facilities,...

To answer these questions and others, many "engines" and programmes have been created around TeX, for example: pdftex, pdflatex, dvips, ps2pdf, METAPOST to open the TeX world to the possibilities of PostScript and PDF, XeTeX and XeLaTeX to be able to use the "normal" fonts found on the different machines and to be able to cope with writing systems different from the left to right systems which originated in Europe (latin and Cyrillic letters and associates) – right to left, vertically, pictograms,..., LuaTeX and LuaLaTeX to have a powerful scripting language.

But to use TeX and the systems of its family, one has to create a "source" document as TeX is only a system to create a typeset document from a source document. This source is a simple text with typesetting instructions and one needs a programme to create it: the editor.

There are many editors able to create a TeX source; some are general editors, others are specifically designed for TeX: here comes TeXworks.

TeXworks is a project to create a text editor for use with TeX family of tools; we will refer to these as (La)TeX. Instead of creating a new sophisticated program, equipped with multiple tool-bars to meet any need, TeXworks provides a simple editor, offering at first sight only a limited set of tools for text editing as well as a single button and a menu to typeset a (La)TeX text.

The idea to create the editor came to *Jonathan Kew*, the initiator and leader of the project, after a long period of reflection on the reasons why potential users tend to keep away from (La)TeX, as well as from the success of the **TeXshop** editor on the Mac.

Finally the goal was also to provide the same editor on many operating systems: TeXworks currently runs on Linux, Mac OS X and Windows. The interface is always

[1] Device Independent: format of files outputed by TeX

the same and the program offers the same functionality on all three platforms.

The first section of this manual explains how to install the software. In the second section, we describe the interface and create a first document showing the basics of TEXworks. In the third section the advanced tools provided by TEXworks are presented; you should read this section only after mastering the basic working of TEXworks. These advanced tools allow much more effective working practices. The last section, as annexes, provides lists of the keyboard shortcuts, the regular expression search/replace system, and the keywords for auto-completion. Finally, a short bibliography closes this manual.

2 Installation

TEXworks is only a text editor; to be able to create documents with (LA)TEX and to typeset them to PDF, we also need what is called a *TEX distribution*. This is a collection of programs and other files which will be automatically called by TEXworks during its work. Thus you need to install a distribution: we will do that *before* starting TEXworks for the first time, as in this way TEXworks will automatically find what it needs.

TeX Live (http://www.tug.org/texlive/), a combination of teTeX, macTeX and XEmTeX, is available for the three operating systems (Linux, Mac OS X, Windows). The current version is TeX Live 2008.

For Linux: every Linux distribution includes a TEX distribution, but it may not be installed by default and you will have to use the Linux package management tools to do that. As well as TeXlive, one can use **teTeX** (http://www.tug.org/teTeX/, a predecessor system to TeXlive.

For the Mac: **MacTeX**, new distribution based on gwTeX and XeTeX, is available; see http://www.tug.org/mactex/.

For Windows: a very popular distribution is **MiKTeX** (http://www.miktex.org/). MikTeX has an update programme, which has also been ported to Linux. You can also use the XEmTeX distribution (http://www.xemtex.org/).

2.1 Under Windows

After the installation of the TEX distribution, obtain TEXworks by downloading the archived programmes from the TEXworks web site: http://tug.org/texworks/; you will find binaries for Mac and Windows at: http://code.google.com/p/texworks/downloads/list.

You need to get: `TeXworks-w32-v0.1r352.zip` the programme itself and some necessary library files [2].

We create a folder, for example `C:\Program files\TeXworks`, and unzip the downloaded files in it. Create, on the Desktop or in the Quick Launch bar a shortcut to the `TeXworks.exe` file.

[2] There are releases also available from `http://www.leliseron.org/texworks`

When you start the program the first time, it creates a folder named TeXworks in your home folder (on Windows XP, C:\Documents and Settings\<your name>, on Vista, C:\Users\<your name>). This folder will contain some sub-folders for auto-completion, configuration, dictionaries, templates, and localisation files — we will see these details later.[3]

NB. At the time of writing, if <your name> contains any non-ASCII characters (for example accented characters) some TEXworks functions will not work correctly. For example, the spell-checker and forward/reverse synchronization between the source and .pdf will be impaired.

2.2 Under Linux

After the installation of the TEX distribution, for TEXworks, download the archived programmes from the web site: https://launchpad.net/ texworks/ and look for +archive/ppa.

Under Linux, if a build of TEXworks exists for your system, use it, otherwise you will probably need to build the program from the source: see the annex section 6.4. Once the program is installed, start TEXworks. A folder .texworks will be created in your home directory.

2.3 Under Mac OS

After the installation of the TEX distribution, for TEXworks, download the archived programmes from the TEXworks web site: http://tug.org/texworks/; you will find binaries for Mac and Windows at: http://code.google.com/p/texworks/downloads/list.

You need to get: TeXworks-Mac-0.1r352.zip, this is a file with everything [4].

It is a standalone .app package that does not require any Qt files installed into /Library/Frameworks, or other libraries into /usr/local/lib. Just copy the .app to anywhere you like, and run it.

On Mac OS X, the TEXworks resource folder will be created in your Library folder (~/Library/.TeXworks/), inside your home directory. Preferences are stored in ~/Library/Preferences/org.tug.TeXworks.plist which you can delete if you ever suspect it is causing problems.

2.4 Ready!

Finally some files may need to be added to the "personal" files that TEXworks creates. As the exact location is dependent on your platform, this will be referred to as <home>\TeXworks or the **TEXworks resource folder** later in this manual. On Windows

[3] TEXworks will save its preferences in the Register: \HKEY_USERS\S-...\Software\TUG\TeXworks. If this is erased, it will be recreated with default values at the next use.

[4] Version at the time of writing this manual.

XP this will be at `C:\Documents and Settings\<your name>\TeXworks`, on Linux it is `~/TeXworks` and `~/Library/.TeXworks/` on the Mac.

After installation and first run, have a look in the sub-folders of the TEXworks resources folder and delete any `qt_temp.xxxx` files; they are temporary files left behind and they could later on interfere with the normal ones, which are installed in the same folder.

3 First steps

Let's now see how to create a first document: for this you'll need to type some text in the editor window of TEXworks. (LA)TEX is not a WYSIWIG software,[5] so you'll have to type the text and the instructions for formatting it and you'll see the result only after "typesetting" the text. This looks a little bit dry, but one very quickly gets used to it and this is worth the effort.

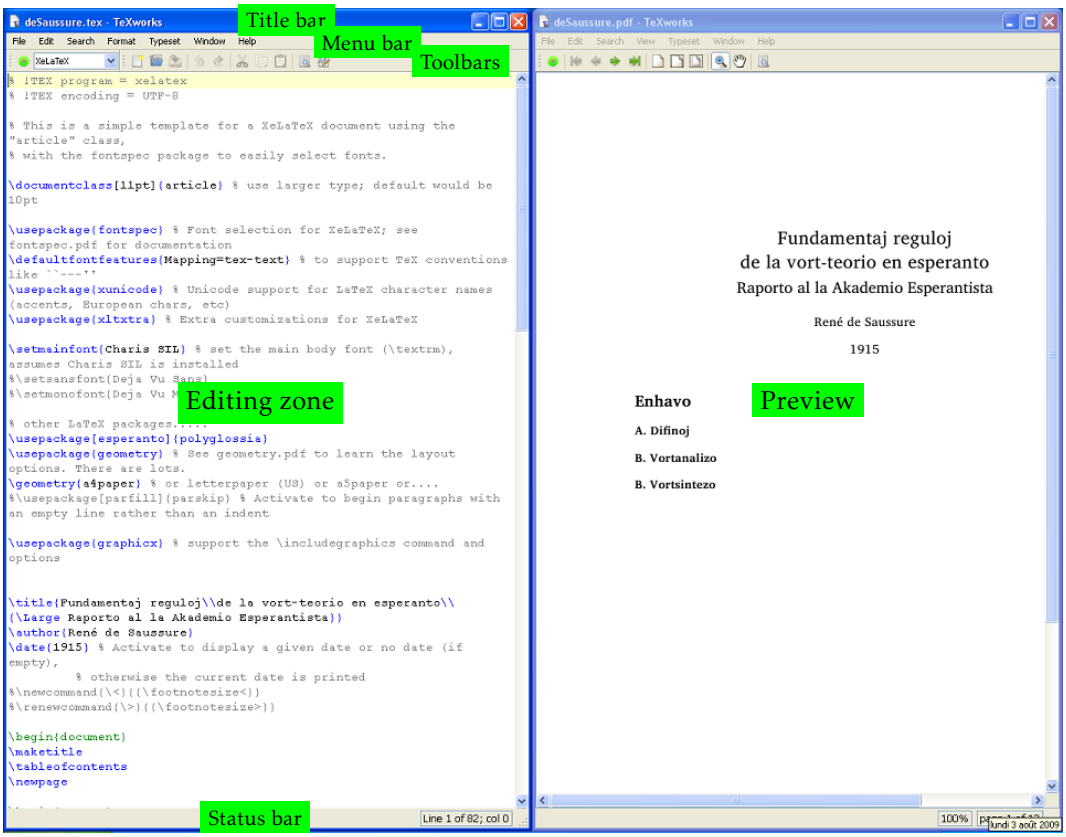
3.1 Interface summary

When one opens the editor, it shows a very sparse interface: a title bar, a menu bar, two small tool-bars, a large typing zone (white) and, at the bottom, a status bar. We are in the *source/editor* window. If you already typeset the document, the resulting .pdf will be shown on the right in the *preview* window.

The first tool-bar has a button to typeset and an drop-down menu to choose the format for typesetting (we'll choose `pdfLaTeX`). Knowing that the keyboard shortcut for typesetting is `[Ctrl+T]` (Mac OS X: `[Cmd+T]`) and that we almost never change the format, we could even not show this tool-bar. The selection of a format for compiling can also be changed through the ***Typeset*** menu.

The second toolbar provides the standard functions: New document, Open, Save | Undo, Redo | Cut, Copy, Paste | Search, Replace.

[5] *What You See Is What You Get.*



3.2 Creating a document

3.2.1 Writing the document

As an example of the use of TEXworks we will work with LATEX, but any other TEX system is possible. In particular, if you need to use some special fonts – mandatory font for an official template, non latin alphabets – the XeTeX system is very powerful. [6]

Let's create now our first document: enter the following text exactly as shown. To show some of the features of TEXworks/LATEX, it is intentionally in French.[7]

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{geometry}
\geometry{a4paper}
```

[6] See the references for pointers to XeTeX and XeLaTeX.

[7] This also means that it matches the original manual in French!

```
\usepackage[francais]{babel}

\title{Premier document}
\author{Un TeXnicien}
\date{}

\begin{document}
\maketitle

Voici un texte accentué en français!

\end{document}
```

Save the file in a folder for test documents (for example: `<home>\TeXworks_tests`); call the file `first.tex`, note that it should have a `.tex` extension.

3.2.2 Typesetting the document and viewing it

Next we start typesetting[8] by clicking the green button  or by `[Ctrl+T]` (Mac OS X `[Cmd+T]`).

A new panel opens between the typing area and the status bar: the *output panel*, labelled *Log*; everything LaTeX is doing when it works is displayed here [9]. When LaTeX finishes (if there is no error) this panel disappears and a new window will appear; in this new window, the *Preview window*, you can see a page with a title "Premier document" followed by the name of the author "Un TeXnicien", both centred, the text "Voici un texte accentué en français !" and at the bottom centre a page number.

Notice that the mouse cursor is like a magnifier in the new window. If you push the left button of the mouse you can see the text under the magnifier much bigger (it is a magnifier, isn't it!); you can move the magnifier and so inspect the text in detail.

To go back to the source, you can just click in its window or better do `[Ctrl+']` (Mac OS X `[Cmd+']`). This shortcut is a toggle between the two windows.[10] See also section 5.1 to automatically move at a specified location of the view from the source or the inverse.

3.2.3 The work of LaTeX

Let's now shortly analyse the result to understand what LaTeX did and why. Introductions and full tutorials can be found on internet: see for example *lshort* which should

[8] We also use the words compilation and compile for the same action; indeed LaTeX works on the source file to produce a .pdf output, so there is a compilation.

[9] see page 10 for a picture of the bottom of the editor window

[10] On Windows one can use `Alt+Tab` to go to the last window opened before the one in use.

be installed as part of your TEX distribution, and is also available from CTAN.[11]

We ask to create a document of the *article* class: this defines the global layout of the document.

Next we say that the input document (the source) is saved with the Unicode encoding *utf-8* and that it will then contain characters which are not present in the standard ASCII without accents. We also want to use an output encoding T1 (the modern TEX encoding); we also want an *A4* document and not the default *US letter* size. Finally we make it clear that the typography should follow the French rules using the **babel** package. Those general instructions for the work are done by packages called with options.

Lastly, we finish the declaration part of the document, the *preamble*, giving the title, the author and the date of the document; here we specify no date.

Next comes the body of the document, between the lines `\begin{document}` and `\end{document}` (commands). It is here where everything which will appear in the document will be.

Let's do some experiments to show the effect of these instructions. For this we put a % in front of the instructions; the % and everything after it will be considered as comment, which will be ignored by LATEX.[12]

Comment out the line `\usepackage[utf8]{inputenc}`, and typeset the file. You should see that the accented characters are now incorrect in the preview window. If you now comment out the line `\usepackage[francais]{babel}`, LATEX will give an error. Just hit `[Enter]` to continue the typesetting.

After these experiments, let's modify the text as follows:

```
\begin{document}
\maketitle
\tableofcontents

\section{Petite démonstration}

Voici un texte accentué en français!
Suite du texte entré après avoir fait un retour chariot. Dans l'éditeur
on peut demander un passage automatique à la ligne du texte saisi; mais
le numéro de ligne n'est incrémenté que par un retour chariot.

Nouvelle ligne en passant une ligne dans la source: c'est la manière
d'indiquer un changement de paragraphe.

\end{document}
```

[11] *Comprehensive TeX Archives Network*, a network of mirror servers of the central CTAN; one can find there everything about TEX, LATEXand more: http://www.ctan.org

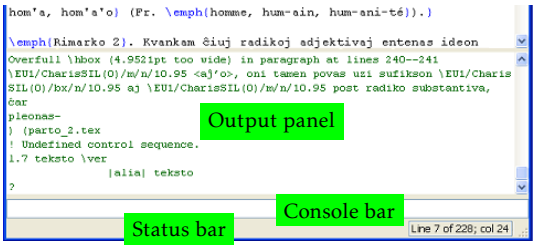
[12] Notice that the comments are, by default – this can be changed, coloured in red by TEXworks, so we see them well.

Redo the previous experiments and observe the changes which appear.

Note that entering only one carriage return doesn't create a new paragraph. In LaTeX one has to have an empty line for that. In TeXworks the line number of the source (on the right in the status bar) numbers the lines created with carriage return, not the wrapped lines.

3.3 And when errors occur!

When you create a document for typesetting with LaTeX, you cannot avoid making mistakes: forgetting a closing brace or an `\end{}` command to close an environment, using mathematical commands without switching to mathematical mode, *etc*. When you compile and there is an error, LaTeX stops, this is shown by scrolling actions in the output panel stopping, and an error message is displayed with LaTeX waiting for an instruction to know what it should do: one sees the *typing cursor* in the line between the output panel and the status bar: *the console bar*.



The error message is on many lines, for example like this:

```
! Undefined control sequence.
l.168 ... fermante ou d'une commande \veb
                                          +\end{}+ de fermeture d'un...
?
```

LaTeX says that it doesn't recognize the command name, sometimes suggests to see the manual or to type `h` (plus `[Return]`) for help, points to the line number (here 168) and shows the place of the error at the cut of the line (here at `\veb`) and finally with the question mark shows that it waits for an action from us.

There are different possible actions:

- Type `[Return]` and ask to continue as if nothing happened; sometimes this allows to finish compiling, but there will be an error in the result.
- Type `h[Return]` to ask for help; this help in not always clearer than the error message, but often this gives a clue.

- Type `i[Return]` to tell LaTeX that we will propose a replacement text. Enter the text followed by `[Return]`; it will be used, beginning at the start of the error, but you should correct the source afterwards. There is no correction in the source during compilation.

- Type `x[Return]` to stop compilation. This is the traditional (LA)TeX way to kill a typesetting process.

  We can also kill the typesetting by repeating the action used to start it: the green typesetting button will have changed to a red one with a white cross  . By clicking on that button or with `[Ctrl+T]` (Mac OS X `[Cmd+T]`), the LaTeX process is terminated. The output panel is still visible and so one can still see the error message.

You should note that sometimes an error appears far from its actual position. For example, when opening an environment but not closing it, LaTeX doesn't see the error before it encounters another end of environment without closing of the first one. The error is often only picked up at the `\end{document}` command, which shows that another environment was not closed!

3.4 Changing TeXworks parameters for convenience

If the default font of the editor doesn't suit you, it is possible to change it from ***Format / Font...*** by selecting a new one in the dialogue box which appears. This change will apply only until TeXworks is restarted.

From the ***Typeset*** menu or from the drop-down on the ***Typesetting tool bar***, you can change the compilation format. Again this change will only be temporary.

To have permanent changes, you need to change the *Preferences* through the ***Edit/ Preferences...*** menu, using the ***Editor*** tab for the font and ***Typesetting*** tab for the compilation format: the default format is at the bottom of the tab (let's choose `pdfLaTex` for this one).

4 Going further: Editing tools

When you have had some practise with TeXworks, you'll find the need for more effective tools. Many tools exist in TeXworks. We are going to see them now.

4.1 Creating a document from a template

Most documents you will create will use the same instructions in the preamble, the same layout settings, similar heading and so on. You can use predefined templates or create your own with all of these settings already in place.

Use ***File / New from template...*** or `[Ctrl+Shift+N]` (Mac OS X `[Cmd+Shift+N]`). A dialogue box opens to allow selection of the template. After selection and `OK` a document is created and you can start to work.

If you want to create a personal template, you just have to create a suitable document with everything you always want to do (and perhaps marking places to fill in) and save it as a `.tex` file in the `<home>\TeXworks\templates` folder, or possibly a sub-folder of it.

4.2 Creating a project using several source files

When the source becomes long, it is sometimes difficult to navigate in it. It is then useful to split the source in different smaller files: one file will be the main document, with the preamble, the `document` environment, as well as calls to the "sub-documents".[13]

But there might be a problem if, in a sub-document, you start typesetting/compilation: as there is no preamble nor `document` environment there is an immediate error stop.

To tell TEXworks that it should typeset the main document one adds at the very beginning of the sub-document the instruction:

```
% !TeX root = path/main_file.tex
```

for example:

```
% !TeX root = manual.tex
```

If the main file is in the same folder, its name is enough, as in the above example. Notice that the slash / and not the backslash \ should be used even on Windows.

Further, with MiKTeX, the call to a sub-document, `\input{name.tex}`, should include the extension `.tex` to have an active SyncTeX action.

4.3 Spell-check

You can turn on automatic spell-checking of your source from ***Edit / Spelling / <language>***: for example `en-GB` for UK English.

During typing, if there is an error, the word is underlined by a red wavy line. A right-click on the word opens a contextual menu in which there are some replacement suggestions. Click on the desired word to make the replacement.

Before using the spell-checker, you need to install dictionaries in the right folder of TEXworks: `<home>\TeXworks\dictionaries`.

One can use the available dictionaries for OpenOffice and other free software;[14], if you have Thunderbird with spell-check, you can copy the `.aff` and `.dic` files. It is possible to ask TEXworks to use by default a dictionary by ***Edit / Preferences... / Editor*** option ***Spell-check language***.

4.4 Search and replace

4.4.1 Standard functions

The options of the menu ***Search***: ***Find...***, ***Find again***, ***Replace...*** and ***Replace again*** (`[Ctrl+F]`, `[Ctrl+G]`, `[Ctrl+R]` and `[Ctrl+Shift+R]`, respectively) are standard ac-

[13] Called by the commands `\input{}` or `\include{}`, see LATEX manuals for more information.

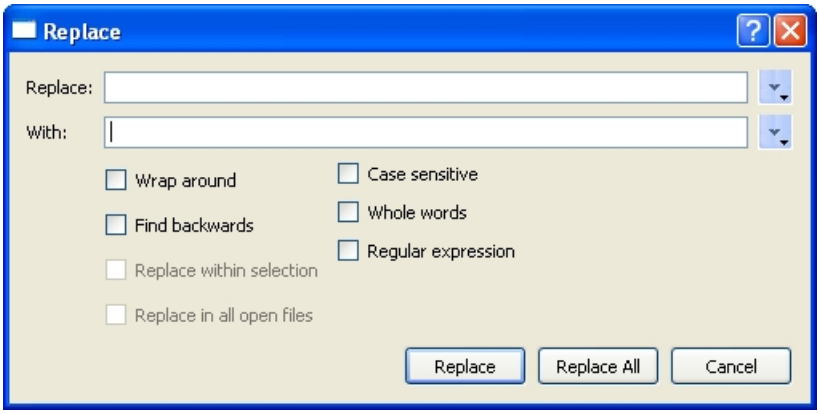
[14] See for example at http://lingucomponent.openoffice.org/download_dictionary.html

tions (Mac OS X [Cmd+F], [Cmd+G], [Cmd+R] and [Cmd+Shift+R]); the first and the third open a dialogue box: (replace)

There are the usual options: *Find backwards*, *Wrap around* or *Replace within selection*. The following options are also usual: *Case sensitive* and *Whole words*. By default the search is forward, towards the end of the document.

The option *Replace in all open files* is also a frequent choice, but not as much as the others; this allows, for example, replacement in all the files of a project – pay attention as this is very powerful.

The last option, *Regular expression*, is detailed in the next sub-section.



In the ***Search*** menu there are other options:

- ***Copy to Find***, copies the currently selected text into the ***Find*** zone of the Find or Replace dialogue; you still need to open ***Find*** or ***Replace*** separately;
- ***Copy to Replace***, the same for the ***Replace*** dialogue;
- ***Find Selection***, uses the current selection for a search without opening the ***Find*** dialogue: very fast;
- ***Show selection***, scrolls the view to the currently selected text: useful if word wrapping is turned off and one moved in the document using the vertical scroll bar on the right.

4.4.2 Regular expressions

The regular expressions provide a very powerful tool, but they require some effort to be well understood. To understand them fully would require a manual, so we'll give some simple ideas of use: see also section 6.2.

Suppose we have the following text:

```
Voici du texte pour tester les expressions régulières
dans du texte accentué.
```

```
Voici du texte pour tester les expressions régulières dans
du texte accentué.
Voici du texte pour tester les expressions régulières. Voici
du texte pour tester les expressions régulières.
truc          truc
tél.: 010-99-99-99
tél.: 00.32.10.99.99.99
tél.: 00/32-10/99.99.99
```

We want to 1) insert an empty line between the paragraphs after "accentué" (to create paragraphs in LaTeX) but not after the three telephone numbers; 2) replace each *tab* character between the two words "truc" of the fourth paragraph, by three spaces; and finally 3) make the telephone numbers consistent by replacing the various punctuation characters by spaces.

In the dialog box we have to select the option: ***Regular expression***.

For 1) in the dialogue box ***Replace*** (`[Ctrl+R]`) for *Find:* we put `>\n<` [15] and in *With:* `>\n\n<`. `>\n<` is the code to match or insert a line feed. You will need to select the first four paragraphs and the beginning of the fifth (first telephone number) and to tick the *Replace within selection*; if this was not done and an empty line has been inserted after each line, select the telephone lines and do the reverse action: replace `>\n\n<` by `>\n<`. So we replace one line feed by two, creating an empty line.

For 2) use `>\t<` and "`>␣␣␣<`. `>\t<` is the code which represent a tab, while a space is typed in literally (here represented as ␣).

For 3) find `>-|\.|/<` and replace with "`>␣<`. Here `>|<` provides different possibilities (`-` `.` `/`); for the point we have used `>\.<` because the dot alone is a regular expression code which represents any character and we would have replaced all the characters by a space!! we then use a code to get the point.

If one has strings of the same character but of different length (for example 2, 3, 4, 5 characters) and one wants to bring all strings to a string with less characters (for example 2), one can ask to replace the string `>e{3,5}<` by `>ee<`.

If one wants to insert at the beginning of some paragraphs separated or not by an empty line the same string, for example `>\noindent␣` or `>\item␣`, one can replace `>\n\n<` or `>\n<` by "`>\n\n\\noindent␣<` or "`>\n\\noindent␣<`. Pay attention, we have double `\` in front of `noindent` to get one (`\noindent`) because `\` is an ecape character in regular expressions!

If it was making sense, we could replace all the letters between "a" and "m" by "$" using `>[a-m]<` and `>$<`.

[15] the >< are used here only to show the limits of the entered text and they should not actually be entered

4.5 Other tools for editing and error tracking

4.5.1 Standard tools

It is always possible to undo an action using ***Edit / Undo*** or [Ctrl+Z] (Mac OS X [Cmd+Z]): this way you can undo stepwise! The inverse action, redo, is available as ***Edit / Redo*** or [Ctrl+Alt+Z] (Mac OS X [Cmd+Opt+Z]).

TEXworks also provides the standard editing tools such as the clipboard; therefore one can select, cut/copy and paste a piece of text.

You can select with the mouse by dragging over the desired text, or by double-clicking to select a word. Using the keyboard, holding down [Shift] while moving using the arrow keys will select text. You can also move and select word by word moving left or right holding [Ctrl+Shift] down ([Cmd+Shift] on Mac OS X). The clipboard shortcuts are the ones you'll finds in almost every program: [Ctrl+X] to cut, [Ctrl+C] to copy and [Ctrl+V] to paste ([Cmd+X], [Cmd+C] and [Cmd+V], respectively, on Mac OS X).

You can easily change the case of a selection — put everything upper case or lower case — using ***Edit / Changer case /*** and next, depending on the desired effect, ***ALL UPPERCASE*** or ***all lowercase***.

It is also convenient to show the line numbers, as all errors messages refer to these numbers; you can toggle the line numbers, on the left of the editing panel, on/of by ***Format / Line Numbers***.

4.5.2 Commenting

When preparing a document with (LA)TEX it is often useful to prevent compilation of a portion of text to be able to locate an error; you can do this piece by piece until you find a part which creates an error. To do this, commenting the source block by block is needed.

We have seen that the symbol % marks the beginning of a comment. To comment a big piece of text, it is sufficient to select it and ask to mark it as comment ***Format / Comment*** or "Ctrl+Shift+]" (Mac OS X "Cmd+Shift+]"). To remove the comment, select the lines and choose ***Format / Uncomment*** or "Ctrl+Shift+[" (Mac OS X "Cmd+Shift+[") [16].

4.5.3 Matching delimiters

A frequent error is to forget a closing symbol: parenthesis, bracket, square bracket, *etc*. TEXworks helps with a tool to show the pairs of symbols: when the cursor moves over one of these symbols, its partner is briefly highlighted in orange. You can also select an entire block using ***Edit / Balance Delimiters*** or by the shortcut [Ctrl+B] (Mac OS X [Cmd+B]). Thus you will immediately sees the scope of the block.

[16] On some keyboards, like the French keyboard, it is not possible to do "Ctrl+Shift+[" or "Ctrl+Shift+]"; so the shortcuts have been changed, see annex 6.1

4.5.4 Smart quotes

Another similar error, but this time semantic and having no influence on typesetting, is in the use of quotes when one wants to give focus to some text.

There are two types of quotes, in English, the 'single' quotes and the "double" quotes. They are obtained by the ` and ´; these are not the quotes used in programming and found on the keyboard " and ′. But using the TEXworks smart quotes system, one can use the latter to automatically produce the single/double opening **and** close quotes.

In a `.tex` document, select one of the smart quotes system: ***Format / Smart Quotes / TeX Ligatures|TeX Commands|Unicode Characters***. Then in your text when you want to start a quoted text, let say double quotes, type " then the text to be focused on and finish again by "; TEXworks will automatically insert the correct opening quotes `` and later the correct closing ones ′′. The three options give the same result in the typeset document.

Finally, it is possible to define personal quotes systems in the file containing the definitions (`smart-quotes-modes.txt` in the `configuration` folder of the resource folder).

4.6 Auto-completion

Another tool which rapidly becomes indispensable is auto-completion. Indeed, when you use (LA)TEX, you have to continuously enter codes to, for example, create environments; you also have to remember to close every group you open.

Auto-completion allows you to type a keyword, hit [Tab] and have TEXworks insert the (LA)TEX command or environment code automatically.

As an example to insert "LATEX", we have to type `\LaTeX`. This is not difficult, but entering "\"[17] followed by the word "LaTeX" with alternating capitals and lower case letters could become annoying after a while. With auto-completion you just enter `latex` and hit [Tab] to get `\LaTeX`. You just have to take care that there is no *letter* glued in front or after `latex`.

Another example is `bmin`, which gives gives:

```
\begin{minipage}{}
•
\end{minipage}•
```

with the cursor between the empty pair of brackets where you need to enter the size of the minipage. See the section 6.3 for a list of the keywords for auto-completion. Notice the "•" in the minipage environment. They are place holders which can be reached by [Ctrl+Tab] ([Option+Tab] on the Mac), repeating this shortcut goes forward in the created structure and by [Ctrl+Shift+Tab] ([Option+Shift+Tab]) one goes backward.

If a partial keyword is given, repeatedly hitting [Tab] will cycle through possible completions. For example, `bali` (the `b` means the beginning of an environment

[17] In particular with keyboard layouts where \ is not directly accessible.

\begin{}) creates the align environment after one [Tab], next align*, and after, in succession, alignat, alignat*, aligned, alignedat, alignedat with option; these last environments have their own code which starts by bali (balis, baliat, baliats, balied, baliedat and baliedato).

If you want to create your own keywords, you can add a .txt file in the completion folder inside <home>\TeXworks. The entries in the file should have the following format:

```
bfigo:=\begin{figure}[#INS#]#RET##RET#\end{figure}•
\bibliography{#INS#}•
```

In the first case, bfigo is the assigned keyword (with :=) to be converted into a figure environment with an optional argument; there is a carriage return after the begin, one empty and the cursor is between the square brackets. The codes #INS# and #RET# represent the cursor insertion point and line returns, respectively, "•" is a place holder.

In the second case we give ourselves a shortcut, which will let us type the first part of \bibliography{} and have TEXworks convert it to the full name plus braces (with the cursor between them). The keyword is the instruction itself.

Of course it is always possible to use "•"!

The .txt file containing the auto-completion information needs to be UTF-8 encoded: it can be created with TEXworks.

5 Going further: Other tools

5.1 SyncTeX-ing between source and preview

When you are reading a document in preview and see something to change, it is convenient to go immediately to the same place in the source. To do that, hold down [Ctrl] (Mac OS X [Cmd]) and click at the appropriate place in the preview window: the cursor will jump and highlight the same location in the source window. The same is true in the other direction: [Ctrl]+[click] in the source will highlight the same line in the preview window. [18]

Here a remark for users under Windows: this only works if **all** the names for folders/files/...do **not** have accented characters. If, for example, your document is in C:\Documents and Settings\Propriétaire\My Documents\thesis it will not work because of the é of Propriétaire!

5.2 Special comment strings

Special comments, at the very beginning of the files, can be used to manage two other aspects of the compilation. By default, TEXworks uses the "utf8" encoding for saving

[18] It is also possible to use a Right-click to open a context window and select "jump to PDF" or "jump to source".

files, but some files could be saved in another format. To ask another encoding for a specific file one can put at the beginning of this file:

`% !TeX encoding = latin1` : another encoding often used.

If we want to compile a file with another programme than the default TEX or LATEX, we put at the beginning of the file:

`% !TeX program = the_programme` for example:

`% !TeX program = xelatex`

Pay attention about this last instruction: you have to use here the name of the programme which should be used for the whole project, as it is the programme name which is first encountered when starting typesetting (the one from the sub-document in which you are), TEXworks will use that programme, even if another name appears in the main document!

When opening a document with a specified programme name, this will become the programme to use (its name will appear in the drop down menu in the Tools bar); but if you then manually change the programme, it is the one shown in the drop down menu which will be used.

5.3 Formatting the source for legibility

To facilitate legibility of the source, one can use indentation as programmers do:

```
\begin{itemize}
    \item First element of the list;
    \item second element;
    \item last element:
    \begin{itemize}      % beginning of a sub-list
        \item first sub-element;
        \item second sub-element.
    \end{itemize}
\end{itemize}
```

this increases legibility, but works well only on short lines, without text wrapping; or if one chooses not to use text wrapping by un-checking ***Format / Wrap lines***.

The command ***Format / Indent*** or the shortcut "`Ctrl+]`" (Mac OS X "`Cmd+]`") will indent the line, or the selected lines, by inserting a tab character. You can repeat the process to increase the indent.

To remove one indentation: ***Format / Unindent*** or the shortcut "`[Ctrl+[`" ("`Cmd+[`" on Mac OS X) [19].

As *indent* only indents the first line of multiline (if wrapped) lines, this is not very satisfactory. But one can ask TEXworks to split a long line (longer than the width of the editing panel) into short ones adding a hard coded line feed: ***Format / Hard Wrap...*** opens a dialog box in which you can specify the width of the lines; you can also re-format lines which have already been split.

[19] See the modified shortcuts for the keyboard not allowing these actions!

5.4 Showing the tags

When a document is becoming long and you want to move to a specific place (a chapter, a section, a sub-section, ...) you need to scroll the editing window to find the desired location.

You can also use the structural information in the document to navigate the source: ***Window / Show / Tags*** opens a panel showing the information detected by TEXworks. Clicking on an item in the panel selects the corresponding part of the source. That panel, like any other, can be resized by dragging its border.

The same action is possible in the .pdf window from ***Window / Show / Table of contents***, but this only works if one has created structure tags in the .pdf file using the package `hyperref`.

5.5 Organising the windows

By default, the editor/source window opens on the left and the preview one on the right (when the corresponding PDF file exists), thus splitting the screen in two.

You can change the position of the windows in the menu ***Window***. ***Stack*** and ***Side by side*** give the same effect if there is only one document open, if not, *Stack* creates a mosaic with all the windows. The other options allow to place the windows for your convenience. It is also always possible to resize and move the windows.

For the preview you can change the way it is presented and of course the zoom by ***Actual size***, ***Fit to width*** and ***Fit to window*** using the options of the ***View*** menu; you can also zoom **in** and **out**. Shortcuts exist for all these actions and are shown next to the option in the menu.

5.6 Cleaning the working folder

Very soon when one uses (LA)TEX, one discovers that the working folder is cluttered by many files which have the name of the source file but different extensions: `.aux`, `.log`,

`.toc`, `.lof`, `.lot`, `.bbl`,...

All these are files needed by (LA)TEX to be able to create the table of contents, lists of figures/tables, the bibliography, the cross references and, also very important, to keep track of what it did (the .log file).

Apart from the external files, images, pictures,..., the only files required are the `.tex` files, the sources of the document. One can erase all the others.

This can be done using a TEXworks command from the ***File*** menu: ***Remove Aux files...***.

When you ask for this command, a dialog box opens in which you can check/uncheck the files you want to remove [20]; you have the choice for `aux`, `log`, `toc` and others depending of what you did. If you already removed some of these files, the choices may differ in the dialog box; if you removed all, you get a message box saying that there is no file to remove at the moment.

The list of auxilliary files which are shown is in the file `texworks-config.txt` in the `configuration` folder of the TEXworks resources folder. You could add some if required.

5.7 Changing the configuration

We have seen in section 2 (page 4) that at first use TEXworks creates a resource folder in `home` (variable on the different systems) and also that it saves preference information (under Windows in a registry key).

But it is possible to define a personal place where one wants the resource folder and the preferences. This can be handy when one wants a portable system or when one wants to easily access the templates or completion folders for modifications.

For this create in the programme folder a file `texworks-setup.ini` in which one specifies the path to the folder containing the completion, configuration, dictionaries,... folder and the configuration file (`texworks.ini`); there will be two lines:

```
inipath=C:/myfolder/TW_conf/\index{configuration!inipath}
libpath=C:/myfolder/TW_conf/\index{configuration!libpath}
```

`inipath` for the configuration file and `libpath` for the necessary folders. Here `TW_conf` would replace the resource folder `TeXworks`. Notice first that the referenced folder (here `TW_conf`) should exist, it will not be created, and second the use of `/` and not the backslash generally used under Windows.

If one wants to put the resource folder in the programme folder, as a sub-folder, one can use an instruction like `inipath=./TW_conf/`; this reference and the other relative references are always related to the TEXworks programme folder.

One could add a first line:
`defaultbinpaths=C:/Program Files/MiKTeX 2.7/miktex/bin`
to specify where are the programmes of the TEX distribution; but this instruction is not yet completely operational, especially under Windows.

[20] The name of the main file is used to list the possible candidates for deletion.

6 Annexes

6.1 Keyboard shortcuts

The use of keyboard shortcuts greatly facilitates typing in and the management of the source and the preview. Their use is much more effective than the use of buttons for frequently-repeated actions.

You'll find below the shortcuts for the work in the source and those for the preview. Note that on Mac OS X, `Ctrl` actually refer to the *Command key*, which is the usual modifier for keyboard shortcuts. Although the keyboard shortcuts are specified with `Ctrl`, this will appear as the *Command-key* symbol in menus. (To refer to the actual *Control key* on the Mac, the shortcut file should use the name `Meta`).

We also show the (proposed) shortcuts implemented in the French localisation translation of the interface.

All the shortcuts can be redefined either to create new shortcuts or to modify the existing ones to match personal uses or change shortcuts not adapted to one particular computer. The list of possible actions to associate with shortcuts is given after the predefined shortcuts.

To define your own shortcuts, put a file named `shortcuts.ini` in the configuration folder of the TEXworks resources folder, next to `auto-indent-patterns.txt`, `delimiter-pairs.txt`, ..., `texworks-config.txt`.

For example, this file could contain:

```
actionHard_Wrap = Shift+F3
actionLast_Page = Ctrl+End
actionFirst_Page = Ctrl+Home
actionWrap_Lines = F3
actionLine_Numbers = F4
actionBalance_Delimiters = F9
```

The first line says that using `Shift+F3` opens the hardwrap dialog box in the source window; the second `Ctrl+End` brings you to the last page and `Ctrl+Home` (third line) to the first page; with `F3` you wrap/unwrap lines in the source, with `F4` you show/hide line numbers and with `F9` you select the text between corresponding delimiters in the source.

Predefined shortcuts:
For the work in the source:

shortcut	in French	Action
Ctrl+N	-	New
Ctrl+Shift+N	-	New from template
Ctrl+O	-	Open
Ctrl+S	-	Save
Ctrl+Shift+S	-	Save as

Ctrl+W	-	Close
Ctrl+Q	-	Quit TEXworks
Ctrl+Z	-	Undo
Ctrl+Shift+Z	-	Redo
Ctrl+X	-	Cut
Ctrl+C	-	Copy
Ctrl+V	-	Paste
Ctrl+F	-	Search
Ctrl+G	-	Search again
Ctrl+R	-	Replace
Ctrl+E	-	Copy to Find
Ctrl+Shift+E	-	Copy to replace
Ctrl+L	-	Go to line
Ctrl+H	-	Find selection
Ctrl+=	-	Show selection
Ctrl+A	-	Select All
Ctrl+B	-	Balance Delimiters
Ctrl+]	Ctrl+>	Indent
Ctrl+[	Ctrl+<	Unindent
Ctrl+Shift+]	Ctrl+(	Comment
Ctrl+Shift+[	Ctrl+)	Uncomment
Ctrl+	Ctrl+$	Show/hide output panel
Ctrl+'	-	Go to preview
Tab	-	Auto-completion
Ctrl+Tab	-	Move to next place holder

moves (and selections: Shift+)

→	-	1 character right
Ctrl+→	-	1 word right
←	-	1 character left
Ctrl+←	-	1 word left
↑	-	1 line up
↓	-	1 line down
PgUp	-	1 screen up
PgDown	-	1 screen down
Home	-	Begin of line
Ctrl+Home	-	Begin of document
End	-	End of line
Ctrl+End	-	End of document

For the work in the preview window:

shortcut	Action
Ctrl+N	New
Ctrl+Shift+N	New from template
Ctrl+O	Open
Ctrl+W	Close
Ctrl+Q	Quit TEXworks
Ctrl+Z	Undo
Ctrl+Shift+Z	Redo
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
Ctrl+RetArr	Erase
Home	First page
End	Last page
PgUp	Previous page
PgDown	Next page
Ctrl+G	Go to page ...
Ctrl++	Zoom +
Ctrl+-	Zoom -
Ctrl+1	Actual size
Ctrl+2	Fit to width
Ctrl+3	Fit to window
Ctrl+Shift+F	Full screen
Ctrl+T	Typeset
Ctrl+'	Go to source

Actions listed alphabetically:

actionAbout_TW	actionPaste
actionActual_Size	actionPlace_on_Left
actionAutoIndent_None	actionPlace_on_Right
actionBalance_Delimiters	actionPreferences
actionClear	actionPrevious_Page
actionClose	actionQuit_TeXworks
actionComment	actionRedo
actionCopy	actionRemove_Aux_Files
actionCopy_to_Find	actionReplace
actionCopy_to_Replace	actionReplace_Again
actionCut	actionRevert_to_Saved
actionFind	actionSave
actionFind_Again	actionSave_As

actionFind_Selection	actionScroll
actionFirst_Page	actionSelect_All
actionFit_to_Width	actionShow_Hide_Console
actionFit_to_Window	actionShow_Selection
actionFont	actionSide_by_Side
actionFull_Screen	actionSmartQuotes_None
actionGoToHomePage	actionStack
actionGo_to_Line	actionSyntaxColoring_None
actionGo_to_Page	actionTile
actionGo_to_Preview	actionToggle_Case
actionGo_to_Source	actionTo_Lowercase
actionHard_Wrap	actionTo_Uppercase
actionIndent	actionTypeset
actionLast_Page	actionUncomment
actionLine_Numbers	actionUndo
actionMagnify	actionUnindent
actionNew	actionWrap_Lines
actionNew_from_Template	actionWriteToMailingList
actionNext_Page	actionZoom_In
actionNone	actionZoom_Out
actionOpen	

Actions listed by menus:

**** Common ****	
[File]	
actionNew	actionNew_from_Template
actionOpen	actionClose
actionQuit_TeXworks	
[Edit]	
actionPreferences	
[Search]	
actionFind	actionFind_Again
[Typeset]	
actionTypeset	
[Windows]	
actionTile	actionStack
actionSide_by_Side	actionPlace_on_Left
actionPlace_on_Right	
[Help]	
actionAbout_TW	actionGoToHomePage

actionWriteToMailingList	
**** TeX source ****	
[File]	
actionSave	actionSave_As
actionRevert_to_Saved	actionRemove_Aux_Files
[Edit]	
actionUndo	actionRedo
actionCopy	actionCut
actionPaste	actionClear
actionBalance_Delimiters	actionSelect_All
actionToggle_Case	actionTo_Lowercase
actionTo_Uppercase	
[Search]	
actionReplace	actionReplace_Again
actionCopy_to_Find	actionCopy_to_Replace
actionFind_Selection	actionGo_to_Line
actionShow_Selection	
[Format]	
actionFont	actionIndent
actionUnindent	actionComment
actionUncomment	actionLine_Numbers
actionWrap_Lines	actionHard_Wrap
actionSyntaxColoring_None	actionAutoIndent_None
actionSmartQuotes_None	
[Windows]	
actionGo_to_Preview	actionShow_Hide_Console
**** PDF preview ****	
[View]	
actionActual_Size	actionFit_to_Width
actionFit_to_Window	actionFirst_Page
actionNext_Page	actionPrevious_Page
actionLast_Page	actionGo_to_Page
actionFull_Screen	actionMagnify
actionZoom_In	actionZoom_Out
[Windows]	
actionGo_to_Source	
**** not in menus ****	
actionNone	actionScroll

6.2 Regular expressions

As TEXworks is built on the base of Qt4, the regular expressions, often referred to as **regexp**, available are a sub-set of the one found for Qt4. See the site of Qt4 [21] for more complete information. It is possible to find other information about regexps on the net [22] or from books. But pay attention that all systems (programming languages, editors,...) do not use the same set of instructions; there is no "standard set".

6.2.1 Introduction

When searching and replacing, one has to define the text to be found. This can be the text itself "Abracadabra", but often it is necessary to define the strings in a more powerful way to avoid repeating the same operation many times with only small changes from one time to the next; example, one wants to replace sequences of the letter **a** by one **o** but not all of them, only the sequences of 3, 4, 5, 6 and 7 **a**; this would require repeating changing 5 times. Another example: replace the vowels by §, again 5 replace operations.

Here come the regular expressions!

A simple character (a or 9) represents itself. But a set of characters can be defined: **[aeiou]** will match any vowel, **[abcdef]** the letters **a b c d e f**; this last set can be shortened as **[a-f]** using "-" between the two ends of the range.

To define a set not to be taken, one uses "^": the caret negates the character set if it occurs as the first character, i.e. immediately after the opening square bracket. **[^abc]** matches anything except **a b c**.

6.2.2 Codes to represent special sets

When using regexps, very often one has to create strings which generally represent other strings, I mean, if you are looking for a string which represents an email address, the letters and symbols will vary; still you could search for any string which corresponds to an email address (`text@text.text` – roughly). So there are abbreviations to represent letters, figures, symbols,...

These codes replace and facilitate the definition of sets; for example to mean the set of digits **[0-9]**, one can use "**\d**". The following table lists the replacement codes. [23]

[21] `http://doc.trolltech.com/4.4/qregexp.html#details`, – this text is based on that information

[22] see for example Wikipedia

[23] simplified from Qt4 at trolltech, see note 21

Element	Meaning
c	Any character represents itself unless it has a special regexp meaning. Thus c matches the character c.
\c	A character that follows a backslash matches the character itself except where mentioned below. For example if you wished to match a literal caret at the beginning of a string you would write "\^".
\n	This matches the ASCII line feed character (LF, Unix newline, used in TEXworks).
\r	This matches the ASCII carriage return character (CR).
\t	This matches the ASCII horizontal tab character (HT).
\v	This matches the ASCII vertical tab character (VT).
\xhhhh	This matches the Unicode character corresponding to the hexadecimal number hhhh (between 0x0000 and 0xFFFF). \0ooo (i.e., zero-ooo) matches the ASCII/Latin-1 character corresponding to the octal number ooo (between 0 and 0377).
. **(dot)**	This matches any character (including newline). So if you want to match the dot, you have to escape it "\.".
\d	This matches a digit.
\D	This matches a non-digit.
\s	This matches a white space.
\S	This matches a non-white space.
\w	This matches a word character or "_").
\W	This matches a non-word character.
\n	The n-th back-reference, e.g. \1, \2, etc.

Using these abbreviations is better than describing the set, because the abbreviations remain valid in different alphabets.

Pay attention that the end of line is often taken as a white space. Under TEXworks the end of line is referred to by "\n".

6.2.3 Repetition

One doesn't work only on unique letter, digit, symbol; most of the time these are repeated (ex.: a number is a repetition of digits and symbols – in the right order).

To show the number of repetitions, one uses a so called "quantifier": **a{1,1}** means at least one and only one **a**, **a{3,7}** between 3 and 7; **{1,1}** can be dropped, so **a{1,1}** = **a**.

This can be combined with the set notation: **[0-9]{1,2}** will correspond to at least one digit and at most two, the integer numbers between 0 and 99. But this will match any group of 1 or 2 figures within a string; if we want that this matches the whole string (we only have 1 or 2 figures in the string) we will write the regular expression as **^[0-9]{1,2}$**; here ^ says that the required string should be the first character of the string, the **$** the last, so there is only one or two figures in the string (^ and **$** are "assertions" – see later for more).

Here a table of quantifiers.[24] E represents an expression (letter, abbreviation, set).

`E?`	Matches zero or one occurrence of E. This quantifier means *the previous expression is optional*. It is the same as **E{0,1}**.
`E+`	Matches one or more occurrences of E. This is the same as `E\{1,MAXINT\}`.
`E*`	Matches zero or more occurrences of E. This is the same as **E{0,MAXINT}**. The * quantifier is often used by a mistake for the + quantifier. Since it matches zero or more occurrences it will match no occurrence at all.
`E{n}`	Matches exactly **n** occurrences of the expression. This is the same as repeating the expression n times.
`E{n,}`	Matches at least **n** occurrences of the expression. This is the same as **E{n,MAXINT}**.
`E{,m}`	Matches at most **m** occurrences of the expression. This is the same as **E{0,m}**.
`E{n,m}`	Matches at least **n** occurrences of the expression and at most **m** occurrences of the expression.

`MAXINT` depends on the implementation, minimum 1024.

6.2.4 Alternatives and assertions

When searching, it is often necessary to search for alternatives, ex.: apple, pear, cherry, but not pineapple. To separate the alternatives, one uses |: apple|pear|cherry. But this will not prevent to find pineapple, so we have to specify that apple should be standalone, a whole word (as is often called in the search dialog boxes).

To specify that a string should be considered standalone, we specify that it is surrounded by word separators/boundaries (begin/end of sentence, space), like **\bapple\b**. For our alternatives example we will **group** them by parentheses and add the boundaries **\b(apple|pear|cherry)\b**. Apart from **\b** we have already seen ^ and **$**.

Here a table of the "assertions" which do not correspond to characters and will never be part of the result of a search. [25]

`^`	The caret signifies the beginning of the string. If you wish to match a literal ^ you must escape it by writing `\^`
`$`	The dollar signifies the end of the string. If you wish to match a literal **$** you must escape it by writing `\$`
`\b`	A word boundary.
`\B`	A non-word boundary. This assertion is true wherever `\b` is false.
`(?=E)`	Positive lookahead. This assertion is true if the expression matches at this point in the regexp.
`(?!E)`	Negative lookahead. This assertion is true if the expression does not match at this point in the regexp.

[24] see note 23
[25] see note 23

Notice the different meanings of ^ as assertion and as negation inside a set!

6.2.5 Final notes

Using rexexp is very powerful, but then also very dangerous; you could change your text at unseen places and sometimes reverting to the previous situation is not fully possible. If you immediately see the error, you could use `Ctrl+Z`.

Showing how to exploit the full power of regexp would require much more than this extremely short summary; in fact it would require a full manual on it own.

Also note that there are some limits in the implementation of regexps in TEXworks; in particular, the assertions (^ and $) only consider the whole file.

Finally, do not forget to "tick" the regexp option when using them in the *Find* and *Replace* dialogs and to un-tick the option when not using regexps.

6.3 Roots for completion

We give here the keywords for auto-completion as they are supplied by TEXworks. They are in the files `tw-basic.txt`, `tw-context.txt` (empty) and `tw-latex.txt` from the `<home>\TeXworks\Completion` folder.

We give them in three columns: the first two show the keywords, the third the (LA)TEX code produced. In some cases there is only the code, this means that you can start to enter the (LA)TEX code and try to complete it with `[Tab]`.

During completion, the system inserts line feeds and puts the cursor at the first place where one has to enter information to complete the typing. To represent the line feeds we used $\mathcal{R}$ and $\mathcal{I}$ for the input point.

It is possible to see that the keywords have some coherence, the mathematical variables have a keyword starting with `x`, when they are in a mathematical environment, when they are used alone in the text you add `d` in front; for example `xa` and `dxa` for `\alpha`, if there is a capital there is a `c`, as `xo` for `\omega` and `xco` for `\Omega`. The keywords for environments start with `b`: `bali` for `align` (`b` for `\begin`). When the environment has possible options, there is one or more `o` added to the base name: `bminp` for `\begin{minipage}{}...` and `bminpo` for `\begin{minipage}[]{}...`.

Keywords defined in `tw-basic.txt` (defined in TEX):

xa	\xa	\alpha
xb	\xb	\beta
xch	\xch	\chi
xd	\xd	\delta
xcd	\xcd	\Delta
xe	\xe	\epsilon
xve	\xve	\varepsilon
xet	\xet	\eta
xg	\xg	\gamma

xcg	\xcg	\Gamma
xio	\xio	\iota
xl	\xl	\lambda
xcl	\xcl	\Lambda
xm	\xm	\mu
xn	\xn	\nu
xo	\xo	\omega
xco	\xco	\Omega
xp	\xp	\pi
xcp	\xcp	\Pi
xvp	\xvp	\varpi
xph	\xph	\phi
xcph	\xcph	\Phi
xvph	\xvph	\varphi
xps	\xps	\psi
xcps	\xcps	\Psi
xs	\xs	\sigma
xcs	\xcs	\Sigma
xvs	\xvs	\varsigma
xz	\xz	\zeta
xr	\xr	\rho
xvr	\xvr	\varrho
xt	\xt	\tau
xth	\xth	\theta
xcth	\xcth	\Theta
xvth	\xvth	\vartheta
xu	\xu	\upsilon
xcu	\xcu	\Upsilon
xx	\xx	\xi
xcx	\xcx	\Xi
tex	\tex	\TeX
		\TeX
texs	\texs	\TeX\
		\TeX\
	\bsk	\bigskip
		\bigskip$\mathcal{R}$
		\hskip
		\indent
		\input
	\msk	\medskip
		\medskip$\mathcal{R}$
		\noindent
		\par

	\ssk	\smallskip$\mathcal{R}$
		\smallskip$\mathcal{R}$
		\scriptsize
		\vskip

Keywords defined in tw-latex.txt (defined in TEX):

-		\textendash
-		\textendash\
--		\textemdash
--		\textemdash\
		\author{$\mathcal{I}$}$\mathcal{R}$
		\begin{
adlen	\adl	\addtolength{$\mathcal{I}$}{}$\mathcal{R}$
		\addtolength{$\mathcal{I}$}{}
adcount	\adc	\addtocounter{$\mathcal{I}$}{}$\mathcal{R}$
		\addtocounter{$\mathcal{I}$}{}
barr		\begin{array}$\mathcal{RIR}$\end{array}
babs	\babs	\begin{abstract}$\mathcal{RIR}$\end{abstract}
bali	\bali	\begin{align}$\mathcal{RIR}$\end{align}
balis	\balis	\begin{align*}$\mathcal{RIR}$\end{align*}
baliat	\baliat	\begin{alignat}{$\mathcal{I}$}$\mathcal{RR}$\end{alignat}
baliats	\baliats	\begin{alignat*}{$\mathcal{I}$}$\mathcal{RR}$\end{alignat*}
balied	\balied	\begin{aligned}{$\mathcal{I}$}$\mathcal{RR}$\end{aligned}
baliedat	\baliedat	\begin{alignedat}$\mathcal{RIR}$\end{alignedat}
baliedato	\baliedato	\begin{alignedat}[$\mathcal{I}$]$\mathcal{RR}$\end{alignedat}
bapp	\bapp	\begin{appendix}$\mathcal{RIR}$\end{appendix}
bbmat	\bbmat	\begin{bmatrix}$\mathcal{RIR}$\end{bmatrix}
bcase	\bcase	\begin{cases}$\mathcal{RIR}$\end{cases}
bcent	\bcent	\begin{center}$\mathcal{RIR}$\end{center}
bcenum	\bcenum	\begin{compactenum}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactenum}
bcenumo	\bcenumo	\begin{compactenum}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{compactenum}
bcitem	\bcitem	\begin{compactitem}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactitem}
bcitemo	\bcitemo	\begin{compactitem}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{compactitem}
bdes	\bdes	\begin{description}$\mathcal{R}$\item[$\mathcal{I}$]$\mathcal{RR}$\end{description}
benu	\benu	\begin{enumerate}$\mathcal{R}$\item$\mathcal{RIR}$\end{enumerate}
benuo	\benuo	\begin{enumerate}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{enumerate}
bequ	\bequ	\begin{equation}$\mathcal{RIR}$\end{equation}
bequs	\bequs	\begin{equation*}$\mathcal{RIR}$\end{equation*}
beqn	\beqn	\begin{eqnarray}$\mathcal{RIR}$\end{eqnarray}
beqns	\beqns	\begin{eqnarray*}$\mathcal{RIR}$\end{eqnarray*}
bf	\bf	\textbf{$\mathcal{I}$}
bfd	\bfseries	\bfseries
bfig	\bfig	\begin{figure}$\mathcal{RIR}$\end{figure}

bfigo	\bfigo	\begin{figure}[$\mathcal{I}$]$\mathcal{RR}$\end{figure}
bmult		\begin{multiline}$\mathcal{RIR}$\end{multiline}
bflalig	\bflalig	\begin{flalign}$\mathcal{RIR}$\end{flalign}
bflaligs	\bflaligs	\begin{flalign*}$\mathcal{RIR}$\end{flalign*}
bfll	\bfll	\begin{flushleft}$\mathcal{RIR}$\end{flushleft}
bflr	\bflr	\begin{flushright}$\mathcal{RIR}$\end{flushright}
bgath	\bgath	\begin{gather}$\mathcal{RIR}$\end{gather}
bgaths	\bgaths	\begin{gather*}$\mathcal{RIR}$\end{gather*}
bgathed	\bgathed	\begin{gathered}$\mathcal{RIR}$\end{gathered}
bgathedo	\bgathedo	\begin{gathered}[$\mathcal{I}$]$\mathcal{RR}$\end{gathered}
bite	\bite	\begin{itemize}$\mathcal{R}$\item$\mathcal{RIR}$\end{itemize}
biteo	\biteo	\begin{itemize}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{itemize}
biblio		\bibliography{$\mathcal{I}$}
bibstyle	\bibstyle	\bibliographystyle{$\mathcal{I}$}
blett	\blett	\begin{letter}{$\mathcal{I}$}$\mathcal{RR}$\end{letter}
blist	\blist	\begin{list}{$\mathcal{I}$}{}$\mathcal{R}$\item$\mathcal{RR}$\end{list}
bminp	\bminp	\begin{minipage}{$\mathcal{I}$}$\mathcal{RR}$\end{minipage}
bminpo	\bminpo	\begin{minipage}[$\mathcal{I}$]{}$\mathcal{RR}$\end{minipage}
bmult	\bmult	\begin{multiline}$\mathcal{RIR}$\end{multiline}
bmults	\bmults	\begin{multiline*}$\mathcal{RIR}$\end{multiline*}
botr		\bottomrule$\mathcal{R}$
bpict	\bpict	\begin{picture}$\mathcal{RIR}$\end{picture}
bpmat	\bpmat	\begin{pmatrix}$\mathcal{RIR}$\end{pmatrix}
bquot	\bquot	\begin{quotation}$\mathcal{RIR}$\end{quotation}
bquo	\bquo	\begin{quote}$\mathcal{RIR}$\end{quote}
bsplit	\bsplit	\begin{split}$\mathcal{RIR}$\end{split}
bsubeq	\bsubeq	\begin{subequations}$\mathcal{RIR}$\end{subequations}
btab	\btab	\begin{tabular}{$\mathcal{I}$}$\mathcal{RR}$\end{tabular}
btabs	\btabs	\begin{tabular*}{$\mathcal{I}$}{}$\mathcal{RR}$\end{tabular*}
btabx	\btabx	\begin{tabularx}{$\mathcal{I}$}{}$\mathcal{RR}$\end{tabularx}
btabl	\btabl	\begin{table}$\mathcal{RIR}$\end{table}
btablo	\btablo	\begin{table}[$\mathcal{I}$]$\mathcal{RR}$\end{table}
btabls	\btabls	\begin{table*}$\mathcal{RIR}$\end{table*}
btablso	\btablso	\begin{table*}[$\mathcal{I}$]$\mathcal{RR}$\end{table*}
btbl	\btbl	\begin{table}$\mathcal{RIR}$\end{table}
btblo	\btblo	\begin{table}[$\mathcal{I}$]$\mathcal{RR}$\end{table}
btbls	\btbls	\begin{table*}$\mathcal{RIR}$\end{table*}
btblso	\btblso	\begin{table*}[$\mathcal{I}$]$\mathcal{RR}$\end{table*}
btabb	\btabb	\begin{tabbing}$\mathcal{RIR}$\end{tabbing}
bbib	\bbib	\begin{thebibliography}{$\mathcal{I}$}$\mathcal{R}$\bibitem{} $\mathcal{RR}$\end{thebibliography}
bindex	\bindex	\begin{theindex}$\mathcal{RIR}$\end{theindex}
btheo	\btheo	\begin{theorem}$\mathcal{RIR}$\end{theorem}

```
btitpg      \btitpg     \begin{titlepage}ℛℐℛ\end{titlepage}
btrivl      \btrivl     \begin{trivlist}ℛℐℛ\end{trivlist}
bvarw       \bvarw      \begin{varwidth}{ℐ}ℛℛ\end{varwidth}
bverb       \bverb      \begin{verbatim}ℛℐℛ\end{verbatim}
bvers       \bvers      \begin{verse}ℛℐℛ\end{verse}
bibitem                 \bibitem{ℐ}ℛ
bibitemo                \bibitem[ℐ]{}ℛ
                        \bottomruleℛ
                        \boxed{ℐ}
center                  \centering
chap        \chapter{ℐ} \chapter{ℐ}ℛ
                        \cite{ℐ}
                        \citep{ℐ}
                        \citet{ℐ}
                        \caption{ℐ}ℛ
                        \cdots
                        \cline{ℐ}
cmidr                   \cmidrule(I){}
cmidro                  \cmidrule[ℐ](){}
                        \date{ℐ}ℛ
dd          \dd         \(ℐ\)
                        \ddots
                        \ddot{ℐ}
                        \dddot{ℐ}
                        \ddddot{ℐ}
                        \documentclass{ℐ}ℛ
                        \documentclass[ℐ]{}ℛ
                        \dots
                        \dotsc
                        \dotsb
                        \dotsm
                        \dotsi
                        \dotso
em          \emph{ℐ}    \emph{ℐ}
emd                     \em
                        \end{ℐ}ℛ
                        \eqref{ℐ}
foot        \footnote{ℐ}\footnote{ℐ}
                        \footnotesize
fbox                    \fbox{ℐ}
fboxo       \fboxo      \framebox[ℐ]{}
                        \framebox[ℐ]{}
fboxoo      \fboxoo     \framebox[ℐ][]{}
```

		\framebox[$\mathcal{I}$][]{}
		\fboxrule{$\mathcal{I}$}
		\fboxsep{$\mathcal{I}$}
geometry	\geometry{}	\geometry{}
		\hline$\mathcal{R}$
		\hspace{$\mathcal{I}$}
		\hspace*{$\mathcal{I}$}
hw		\headwidth
hw2tw		\setlength{\headwidth}{\textwidth}$\mathcal{R}$
href		\href{$\mathcal{I}$}{}
		\include{$\mathcal{I}$}$\mathcal{R}$
incg		\includegraphics{$\mathcal{I}$}$\mathcal{R}$
incgo		\includegraphics[$\mathcal{I}$]{}$\mathcal{R}$
it		\item$\mathcal{R}\mathcal{I}$
ito		\item[$\mathcal{I}$]$\mathcal{R}$
		\intertext{$\mathcal{I}$}
ti	\ti	\textasciitilde{$\mathcal{I}$}
itd	\itshape	\itshape
latex	\latex	\LaTeX
		\LaTeX
latexs	\latexs	\LaTeX\
		\LaTeX\
latexe	\latexe	\LaTeXe
		\LaTeXe
latexes	\latexes	\LaTeXe\
		\LaTeXe\
lbl	\lbl	\label{$\mathcal{I}$}
		\label{$\mathcal{I}$}
		\large
		\Large
		\ldots
listf	\listf	\listoffigures$\mathcal{R}$
		\listoffigures$\mathcal{R}$
listt	\listt	\listoftables$\mathcal{R}$
		\listoftables$\mathcal{R}$
mbf	\mbf	\mathbf{$\mathcal{I}$}
		\mathbf{$\mathcal{I}$}
mrm	\mrm	\mathrm{$\mathcal{I}$}
		\mathrm{$\mathcal{I}$}
mcal	\mcal	\mathcal{$\mathcal{I}$}
		\mathcal{$\mathcal{I}$}
msf	\msf	\mathsf{$\mathcal{I}$}
		\mathsf{$\mathcal{I}$}

mtt	\mtt	\mathtt{$\mathcal{I}$}
		\mathtt{$\mathcal{I}$}
mit	\mit	\mathit{$\mathcal{I}$}
		\mathit{$\mathcal{I}$}
mnorm	\mnorm	\mathnormal{$\mathcal{I}$}
		\mathnormal{$\mathcal{I}$}
mdd		\mdseries
mbox		\mbox{$\mathcal{I}$}
mboxo	\mboxo	\makebox[$\mathcal{I}$]{}
		\makebox[$\mathcal{I}$]{}
mboxoo	\mboxoo	\makebox[$\mathcal{I}$][]{}
		\makebox[$\mathcal{I}$][]{}
mpar	\mpar	\marginpar{$\mathcal{I}$}
		\marginpar{$\mathcal{I}$}
midr		\midrule$\mathcal{R}$
multc	\multc	\multicolumn{$\mathcal{I}$}{}{}
multic	\multic	\multicolumn{$\mathcal{I}$}{}{}
		\multicolumn{$\mathcal{I}$}{}{}
ncol	\ncol	&
ncm		\newcommand{$\mathcal{I}$}{}$\mathcal{R}$
newc		\newcommand{$\mathcal{I}$}{}$\mathcal{R}$
ncmo		\newcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
newco		\newcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
ncmoo		\newcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
newcoo		\newcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
nct		\newcolumntype{$\mathcal{I}$}{}
newct		\newcolumntype{$\mathcal{I}$}{}
newpg		\newpage$\mathcal{R}$
npg	\npg	\newpage$\mathcal{R}$
nline		\newline$\mathcal{R}$
newlin		\newline$\mathcal{R}$
nlen		\newlength{$\mathcal{I}$}$\mathcal{R}$
newlen		\newlength{$\mathcal{I}$}$\mathcal{R}$
nenv		\newenvironment{$\mathcal{I}$}{}{}$\mathcal{R}$
newe		\newenvironment{$\mathcal{I}$}{}{}$\mathcal{R}$
nenvo		\newenvironment{$\mathcal{I}$}[]{}{}$\mathcal{R}$
neweo		\newenvironment{$\mathcal{I}$}[]{}{}$\mathcal{R}$
nenvoo		\newenvironment{$\mathcal{I}$}[][]{}{}$\mathcal{R}$
neweoo		\newenvironment{$\mathcal{I}$}[][]{}{}$\mathcal{R}$
newt		\newtheorem{$\mathcal{I}$}{}$\mathcal{R}$
newto		\newtheorem{$\mathcal{I}$}{}[]$\mathcal{R}$
newtoo		\newtheorem{$\mathcal{I}$}[]{}$\mathcal{R}$

		\nocite{$\mathcal{I}$}
		\normalsize
pgref		\pageref{$\mathcal{I}$}
par		\paragraph{$\mathcal{I}$}$\mathcal{R}$
pars		\paragraph*{$\mathcal{I}$}$\mathcal{R}$
paro		\paragraph[$\mathcal{I}$]{}$\mathcal{R}$
		\pagebreak$\mathcal{R}$
pgs		\pagestyle{$\mathcal{I}$}$\mathcal{R}$
parbox		\parbox{$\mathcal{I}$}{}
parboxo		\parbox[$\mathcal{I}$]{}{}
pbox	\pbox	\parbox{$\mathcal{I}$}{}
pboxo	\pboxo	\parbox[$\mathcal{I}$]{}{}
pbox		\pbox{$\mathcal{I}$}{}
ref		\ref{$\mathcal{I}$}
rncm		\renewcommand{$\mathcal{I}$}{}$\mathcal{R}$
rnewc		\renewcommand{$\mathcal{I}$}{}$\mathcal{R}$
rnewco		\renewcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
rncmo		\renewcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
rnewcoo		\renewcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
rncmoo		\renewcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
		\rm
rmc		\rmfamily
rbox	\rbox	\raisebox{$\mathcal{I}$}{}
		\raisebox{$\mathcal{I}$}{}
rboxo	\rboxo	\raisebox{$\mathcal{I}$}[]{}
		\raisebox{$\mathcal{I}$}[]{}
rboxoo	\rboxoo	\raisebox{$\mathcal{I}$}[][]{}
		\raisebox{$\mathcal{I}$}[][]{}
		\rule{$\mathcal{I}$}{}
		\rule[$\mathcal{I}$]{}{}
		\sc
		\scshape
sec		\section{$\mathcal{I}$}$\mathcal{R}$
secs		\section*{$\mathcal{I}$}$\mathcal{R}$
seco		\section[$\mathcal{I}$]{}$\mathcal{R}$
		\setlength{$\mathcal{I}$}{}
ssec	\ssec	\subsection{$\mathcal{I}$}$\mathcal{R}$
		\subsection{$\mathcal{I}$}$\mathcal{R}$
ssecs	\ssecs	\subsection*{$\mathcal{I}$}$\mathcal{R}$
		\subsection*{$\mathcal{I}$}$\mathcal{R}$
sseco	\sseco	\subsection[$\mathcal{I}$]{}$\mathcal{R}$
		\subsection[$\mathcal{I}$]{}$\mathcal{R}$
sssec	\sssec	\subsubsection{$\mathcal{I}$}$\mathcal{R}$

		\subsubsection{$\mathcal{I}$}$\mathcal{R}$
sssecs	\sssecs	\subsubsection*{$\mathcal{I}$}$\mathcal{R}$
		\subsubsection*{$\mathcal{I}$}$\mathcal{R}$
ssseco	\ssseco	\subsubsection[$\mathcal{I}$]{}$\mathcal{R}$
		\subsubsection[$\mathcal{I}$][]$\mathcal{R}$
spar	\spar	\subparagraph{$\mathcal{I}$}
		\subparagraph{$\mathcal{I}$}
spars	\spars	\subparagraph*{$\mathcal{I}$}
		\subparagraph*{$\mathcal{I}$}
sparo	\sparo	\subparagraph[$\mathcal{I}$]{}
		\subparagraph[$\mathcal{I}$]{}
stcount		\stepcounter{$\mathcal{I}$}$\mathcal{R}$
sf	\sf	\textsf{$\mathcal{I}$}
sfd	\sffamily	\sffamily
scd		\scshape
sl	\sl	\textsl{$\mathcal{I}$}
		\textsl{$\mathcal{I}$}
sld		\slshape
		\text{$\mathcal{I}$}
		\textbf{$\mathcal{I}$}
		\textsf{$\mathcal{I}$}
sc		\textsc{$\mathcal{I}$}
		\textit{$\mathcal{I}$}
		\textup{$\mathcal{I}$}
tt	\tt	\texttt{$\mathcal{I}$}
ttd		\ttfamily
		\texttt{$\mathcal{I}$}
tw	\tw	\textwidth
		\textwidth
		\thanks{$\mathcal{I}$}$\mathcal{R}$
		\title{$\mathcal{I}$}$\mathcal{R}$
tilde	\tilde	\textasciitilde
topr		\toprule$\mathcal{R}$
toc	\toc	\tableofcontents$\mathcal{R}$
tableofcontents		\tableofcontents$\mathcal{R}$
up	\up	\textup{$\mathcal{I}$}
upd		\upshape
url		\url{$\mathcal{I}$}
usep		\usepackage{$\mathcal{I}$}$\mathcal{R}$
usepo		\usepackage[$\mathcal{I}$]{}$\mathcal{R}$
		\vdots
		\vspace{$\mathcal{I}$}$\mathcal{R}$
		\vspace*{$\mathcal{I}$}$\mathcal{R}$

dxa	\dxa	\(\alpha\)
dxb	\dxb	\(\beta\)
dxch	\dxch	\(\chi\)
dxd	\dxd	\(\delta\)
dxcd	\dxcd	\(\Delta\)
dxe	\dxe	\(\epsilon\)
dxve	\dxve	\(\varepsilon\)
dxet	\dxet	\(\eta\)
dxg	\dxg	\(\gamma\)
dxcg	\dxcg	\(\Gamma\)
dxio	\dxio	\(\iota\)
dxl	\dxl	\(\lambda\)
dxcl	\dxcl	\(\Lambda\)
dxm	\dxm	\(\mu\)
dxn	\dxn	\(\nu\)
dxo	\dxo	\(\omega\)
dxco	\dxco	\(\Omega\)
dxp	\dxp	\(\pi\)
dxcp	\dxcp	\(\Pi\)
dxvp	\dxvp	\(\varpi\)
dxph	\dxph	\(\phi\)
dxcph	\dxcph	\(\Phi\)
dxvph	\dxvph	\(\varphi\)
dxps	\dxps	\(\psi\)
dxcps	\dxcps	\(\Psi\)
dxs	\dxs	\(\sigma\)
	\dxcs	\(\Sigma\)
dxvs	\dxvs	\(\varsigma\)
dxz	\dxz	\(\zeta\)
dxr	\dxr	\(\rho\)
dxvr	\dxvr	\(\varrho\)
dxt	\dxt	\(\tau\)
dxth	\dxth	\(\theta\)
dxcth	\dxcth	\(\Theta\)
dxvth	\dxvth	\(\vartheta\)
dxu	\dxu	\(\upsilon\)
dxcu	\dxcu	\(\Upsilon\)
dxx	\dxx	\(\xi\)
dxcx	\dxcx	\(\Xi\)
sqrt	\sqrt	\sqrt{$\mathcal{I}$}
sqrto	\sqrto	\sqrt[$\mathcal{I}$]{}
frac		\frac{$\mathcal{I}$}{}

```
{array}RIR\end{array}
{abstract}RIR\end{abstract}
{align}RIR\end{align}
{align*}RIR\end{align*}
{alignat}{I}RR\end{alignat}
{alignat*}{I}RR\end{alignat*}
{aligned}RIR\end{aligned}
{aligned}[I]RR\end{aligned}
{alignedat}{I}RR\end{alignedat}
{appendix}RIR\end{appendix}
{bmatrix}RIR\end{bmatrix}
{compactenum}R\itemRIR\end{compactenum}
{compactenum}[I]R\itemRR\end{compactenum}
{compactitem}R\itemRIR\end{compactitem}
{compactitem}[I]R\itemRR\end{compactitem}
{cases}RIR\end{cases}
{center}RIR\end{center}
{description}R\item[I]RR\end{description}
{enumerate}R\itemRIR\end{enumerate}
{enumerate}[I]R\itemRR\end{enumerate}
{equation}RIR\end{equation}
{eqnarray}RIR\end{eqnarray}
{eqnarray*}RIR\end{eqnarray*}
{figure}RIR\end{figure}
{figure}[I]RR\end{figure}
{flalign}RIR\end{flalign}
{flalign*}RIR\end{flalign*}
{flushleft}RIR\end{flushleft}
{flushright}RIR\end{flushright}
{gather}RIR\end{gather}
{gather*}RIR\end{gather*}
{gathered}RIR\end{gathered}
{gathered}[I]RR\end{gathered}
{itemize}R\itemRIR\end{itemize}
{itemize}[I]R\itemRR\end{itemize}
{letter}{I}RR\end{letter}
{list}{I}{}R\itemRR\end{list}
{minipage}{I}RR\end{minipage}
{minipage}[I]{}RR\end{minipage}
{multiline}RIR\end{multiline}
{multiline*}RIR\end{multiline*}
{picture}RIR\end{picture}
{pmatrix}RIR\end{pmatrix}
```

```
{quotation}ℛℐℛ\end{quotation}
{quote}ℛℐℛ\end{quote}
{split}ℛℐℛ\end{split}
{subequations}ℛℐℛ\end{subequations}
{tabular}{ℐ}ℛℛ\end{tabular}
{tabular*}{ℐ}{}ℛℛ\end{tabular*}
{tabularx}{ℐ}{}ℛℛ\end{tabularx}
{table}ℛℐℛ\end{table}
{table}[ℐ]ℛℛ\end{table}
{table*}ℛℐℛ\end{table*}
{table*}[ℐ]ℛℛ\end{table*}
{tabbing}ℛℐℛ\end{tabbing}
{thebibliography}ℛℐℛ\end{thebibliography}
{theindex}ℛℐℛ\end{theindex}
{theorem}ℛℐℛ\end{theorem}
{titlepage}ℛℐℛ\end{titlepage}
{trivlist}ℛℐℛ\end{trivlist}
{varwidth}{ℐ}ℛℛ\end{varwidth}
{verbatim}ℛℐℛ\end{verbatim}
{verse}ℛℐℛ\end{verse}
```

There are also environment codes (above) without `\begin{` (which is itself a keyword); this allows, if one started to input an environment code, to finish it by [Tab].

6.4 Compiling TEXworks

There are some documents giving detailed instructions to compile TEXworks on different machines.

For Linux:
http://code.google.com/p/texworks/ and
http://code.google.com/p/texworks/wiki/Building.

For the Mac:
http://code.google.com/p/texworks/issues/detail?id=107&q=MAc

For Windows:
http://code.google.com/p/texworks/wiki/BuildingOnWindowsMinGW

References

[1] **D. Knuth**, *The TEXbook*, Addison Wesley, 1986-1992

[2] **D. Knuth**, *The METAFONT book*, Addison Wesley, 1986-1992

[3] **L. Lamport**, *LATEX: A Document Preparation System*, Addison Wesley, 1985 (LATEX 2.09), 1994 (LATEX 2_ε)

[4] **M. Goossens, F. Mittelbach & A. Samarin**, *The LaTeX Companion*, Addison Wesley, 1994

[5] **M. Goossens**, *The XeTeX Companion*, July 2009, http://xml.web.cern.ch/XML/lgc2/xetexmain.pdf

[6] **D. J. Perry**, *Creating Scholarly Multilingual Documents Using Unicode, OpenType, and XeTeX*, June 2009, http://scholarsfonts.net/xetextt.pdf

Index