Introduction

This book originally accompanied a two-day course on using the LaTeX typesetting system. It was extensively revised and updated for publication, so that it could be used for self-study as well as in the classroom. For those with sufficient prior knowledge of computing and authoring, it has also successfully been used as the basis for a 1-day intensive introductory course. It is aimed at users of Unix & GNU/Linux, including Apple Macintosh OS X, and Microsoft Windows systems, but it can be used with LaTeX on any platform, especially online systems such as *Overleaf*, but also including other Unix workstations, mainframes, Android and Apple smartphones, and even some older Personal Digital Assistants (PDAs).

Who needs this book?

The course was originally designed for computer-literate but non-IT professionals in business, academic, and nonprofit organisations. You may be in a similar position, but you may also come from another background entirely; you may be a hobbyist, a school or college student, a home computer user or a volunteer worker, or you might just be interested in high-quality automated typesetting. However, it's likely that you have one or more of the following or similar objectives:

□ producing consistent, typeset-quality formatting;

□ formatting long *or* complex *or* highly-structured *or* repetitive *or* automatically-generated documents;

□ saving time and effort by automating common tasks;

□ gaining independence from expensive and restrictive proprietary hardware, software, or file formats;

□ creating robust, durable documents which will survive changes in technology;

□ having fun playing around with fonts and formatting.

Skills needed

LaTeX is a very easy system to learn, and requires no specialist knowledge to get started, although it's useful if you understand a little about writing, formatting, and readability. However, you do need to be completely familiar with using your computer, which means knowing the following topics thoroughly. Note that none

Table 1 – Using your computer — essential skills

Subject	**Detail**	**ECDL**
Using the mouse	know how to point and click with your mouse to run programs, pick from a menu, and highlight text (or how to use keyboard shortcuts to do the same)	1.1.4–1.1.6, 1.3.1, 1.4.1–1.4.2,
Handling files	know how to create, open, save, close, rename, copy, move, and delete files and folders (directories) using a directory browser (Windows: *File Explorer*, *My Computer*, or just *Computer*; Mac: *Finder*; Linux: *Nautilus*, *Thunar*, *Dolphin*, etc) or a typed command	2.3
Handling characters	know where to find all 95 of the printable (ASCII) characters on your keyboard, plus accents and symbols, if you need them	3.2.1.2
Using an editor	know how to use a good plaintext editor (*not* a wordprocessor like Microsoft *Word*, *Libre Office*, Lotus *Notes*, Apple *Pages*, or Corel *WordPerfect*; and *not* a context-insensitive editor like Apple *TextEdit* or Microsoft *Notepad*).	2.1.3

of these is in any way specialist; they're all basic, fundamental, standard computer skills that everyone should know.

If you don't know how to do these things yet, it's important to go and learn them first, at least the essential ones. Trying to become familiar with basic computer skills *at the same time* as learning LaTeX is not going to be as effective as doing them in the right order.

It is really important to understand that these are *not* specialist skills — they are standard for anyone who uses a computer, and they form a fundamental part of the basic knowledge of computers which everyone needs to be familiar with.

With the exception of software installation, they are all included in the 2000 European Computer Driving Licence (ECDL) (now the International Computer Driving Licence (ICDL)) course: the relevant module and section numbers of the original ECDL syllabus are noted in parentheses or in the margin above (Kelly and O'Connor 2005).

Table 2 – Using software (programs) — useful skills

Subject	Detail	ECDL
Downloading files	know how to use your Web browser and/or file transfer program to download and save files from the Internet	7.1.6 [7.3.1.6]
Unzip files	know how to uncompress and unwrap compressed 'archive' (zip) files	2.3.8
Install software	know how to install software, both manually and using automated installers	2.6
RTFM	know how to read and follow instructions and how (and where) to ask for help	1.7

Objectives of this book

By the end of this book or course, you should be able to undertake the following tasks:

1. use your editor to create and maintain your documents;
2. use LaTeX markup to identify your document structure and formatting requirements;
3. typeset LaTeX documents, correct simple formatting errors, and display or print the results;
4. identify and use additional LaTeX packages (using the Internet for downloading where necessary and installing them);
5. recognise the limitations of procedural markup systems and choose appropriate generic markup methods where appropriate.

Synopsis

The original course covered the following topics as separate sessions. Earlier versions of this document kept to this structure in the book as chapters, but recent versions have moved Installation (originally chapter 1) to Appendix A and merged it with the details of configuration; and Typesetting, viewing, and printing (originally chapter 4) to a new Appendix; as the procedures in both cases have been so much simplified that the previous level of detail is no longer needed.

1. How to create LaTeX documents (with a Quick-Start Guide for the impatient);

2. Basic document structures (the Document Class Declaration and its layout options; the *document* environment with sections and paragraphs);
3. Using packages and CTAN to adapt formatting to your needs;
4. Other document structures (lists, tables, figures, images, and verbatim text);
5. Textual tools (footnotes, marginal notes, cross-references, indexes and glossaries, and bibliographic citations);
6. Typographic considerations (white-space and typefaces; inline markup and font changes; extra font installation and automation);
7. Programmability and automation (macros and modifying LaTeX's behaviour);
8. Conversion and compatibility with other systems (XML, *Word*, etc).

A Where to get and how to install LaTeX;

B How to install new fonts;

C Typesetting, viewing, and printing (largely obsolete now that editors are better integrated with viewers and printers);

D User groups and the benefits of membership;

E The ASCII character set;

F The GNU Free Documentation License.

I have made a few other changes in the transition to printed and online form, but the basic structure is the same, and the document functions as a workbook for the course as well as a standalone self-teaching guide.

Where's the math?

Please understand that this document *does not cover* mathematical typesetting, complex tabular material, the design of large-scale macros and document classes, or the finer points of typography or typographic design, although it does refer to these topics in passing on a few occasions.

There are several other guides, introductions, and 'get-started' documents on the Web and on CTAN which cover these topics and more in great detail. Among the more popular are:

□ *Getting Started*, where all beginners should start;

□ *A (Not So) Short Introduction to LaTeX 2ε* is a good beginner's tutorial;

□ *Gentle Intro* is a classic tutorial on Plain TeX (not LaTeX);

□ *Imported graphics* shows you how to do (almost) anything with graphics: side-by-side, rotated, etc;

□ *Short Math Guide for LaTeX* gets you started with the American Math Society's extensions;

□ *TeX Symbol List* shows over 2,500 symbols available.

This list was taken from the CTAN search page. There are also lots of books published about TeX and LaTeX: the most important of these for users of this document are listed at the end of the on page xvii.

Availability of LaTeX systems

The standard implementations of TeX and related systems are published annually online by the TeX Users Group (TUG). As of 2024 the former TeX Collection is no longer available on Digital Versatile Disk (DVD) except by special request. These implementations are all derived from Knuth's master versions, and adapted for all major platforms (Unix & GNU/Linux, including Apple Mac OS X; and Microsoft *Windows*). You can also download the installation image file in ISO format from CTAN to burn your own DVD if you need to.

Rather than install TeX Live on your laptop, desktop, or tablet, many users prefer to use one of the online systems such as *Overleaf* via a web browser. These provide an editor and typeset display in the same way as installed systems, but contained inside your web browser. LaTeX still works identically no matter which way you choose to use it.

There are no longer any commercial implementations of TeX that I am aware of. The most recently available ones are listed in 'Commercial implementations' on page xxxiv.

Systems included in the TeX Live distribution from TUG

MiKTeX (Windows): This is the *MiKTeX* implementation plus the *TeXStudio* editor with its own built-in PDF viewer.

MacTEX (OS X): This is *TEX Live* for Mac plus the *TEXshop* editor (the Mac's own built-in *Preview* is used for the PDF display).

TEX Live (Unix & GNU/Linux, including Apple Macintosh OS X; and Windows): The full *TEX Live* system from TUG.

Unix and GNU/Linux users can also choose to install the prepackaged implementation from their system's repositories (see section A.2.3 on page 220).

Because the TEX program (the internal 'engine' which does the actual typesetting) is independent of any other software, it doesn't have its own editor like a wordprocessor does. Instead, you get to choose whichever editor you prefer: there are lots available, and you can switch between them to find one you like: see 'Graphical interfaces (editors)' below and section 1.2 on page 5 for details.

Graphical interfaces (editors)

Most users run LATEX with a graphical plaintext editor which has a toolbar and menus like other windowing applications. These usually include all the common formatting features of LATEX plus writing tools like spellchecking, thesaurus, indexing, and bibliographic citation. They generally all work in a very similar way. Text-only interfaces are available for use on servers and automated production systems (see 'Command-line interfaces').

The Windows and Mac systems described in 'Availability of LATEX systems' on page xxxi come with a recommended editor (*TEXStudio* and *TEXshop* respectively), but you can install any other suitable editor you prefer (see section 1.2 on page 5). The Unix & GNU/Linux distribution does not install any editor because these systems usually have their own software repositories with suitable editors already available for installation, such as *Emacs*, *vi*, *TEXStudio*, or *Kile*.

Some fully synchronous typographic interfaces (editors) were available as commercial products, but so far as is known none of these is available any more: see the list item 'Synchronous typographic displays' on page xxxiii.

Command-line interfaces

While you would use a graphical interface to *set up* an automated system like a web server or e-commerce environment, it is useless where systems have to run in the background, unattended, with no human to click on buttons. In fact, the TEX typesetting engine is a Command-Line Interface (CLI) program, which can be used from any script or console or 'Command' window. You can type the command `xelatex` followed by the name of your document file (see Figure B.2 on page 236 for an example).

Commands like these let you run LaTeX in an automated or scripted environment like a Common Gateway Interface (CGI) script on a web server or a batch file on a document publishing system. All the popular distributions for all systems include this CLI interface as standard.

WYSIWYG displays

LaTeX usually displays your typeset results in a separate window such as a PDF viewer, updated automatically every time the document is retypeset, because the typesetting is kept separate from the editing. This is called an 'asynchronous' display. Some systems, however, can format the typesetting while you type each character, like a wordprocessor, although at the expense of some flexibility. These are called 'synchronous' displays.

Asynchronous typographic displays: The WYSIWYG display is updated when the document is reprocessed, rather than *while* you are still typing, as it would with a wordprocessor. To update the display, just click on the button which reformats the document. You are probably already familiar with this idea if you have used HTML, where you reload the page in a browser to see it, or if you have used a spreadsheet, where the ReCalc button (F9) does something similar.

TeX systems typeset the whole document at one go, including all indexing, cross-references, tables of contents, bibliographic citations, and the placement of figures and tables. TeX also formats whole paragraphs at a time, rather than line-by-line as wordprocessors do, in order to get the quality of spacing, hyphenation, and justification right. This approach makes it much faster than a wordprocessor in dealing with typical complex documents, as it can be done without holding the whole document in memory.

Synchronous typographic displays: The WYSIWYG display is the editing window, and it updates while you type, like a wordprocessor, for example *Scientific Word* (see the list item 'Scientific Word' section 8.1.1.1 on page 199) or *VTeX* (see 'Commercial implementations' on page xxxiv).

With a synchronous display you get Instant Textual Gratification™, but like a wordprocessor, your level of control is restricted to that of the system you use, which cannot provide access to everything that LaTeX can do. For complete control of complex material you may still need to use separate editing and display windows as for asynchronous implementations.

Near-synchronous displays: There are a few systems for very-close-to-synchronous WYSIWYG display. These include Jonathan Fine's *Instant Preview* with the TEX daemon, and David Kastrup's preview-latex package for embedding typographic fragments from the typeset display back into the editor window.

What You See Is What You Get (WYSIWYG) refers to the accuracy with which the typographical display shows your document. Most modern PDF viewers are pretty good, given the fact that your screen is probably only a fraction of the accuracy of your printer — from 96 dots per inch (DPI) on an old desktop screen, 140 DPI on a small laptop, 240 DPI on a high-end 4k laptop or desktop, and up to around 300 DPI on some phones and tablets; as opposed to 600 DPI on your printer, or 1200 DPI or more in photo-quality, and 3,600 DPI up to well over 5,000 DPI on industrial digital printers and laser phototypesetters. More commonly, what people are really trying to express with WYSIWYG is What You See Is What You Mean (WYSIWYM); that is, your intent is accurately conveyed by the formatting.

Commercial implementations

Although TEX Live is available free of charge, there have been some excellent commercial implementations of TEX and LATEX, mentioned below, which provided enhanced support and additional features, and some of these are still in use. These companies, founders, and staff were good friends of the TEX and LATEX communities for many years; but regrettably, all the products listed are now at end-of-life, although they are in some cases still available for download.

Y&Y, Mackichan Software, PC-TEX, Blue Sky (Textures), True TEX, and BaKoMa TEX (RIP Basil K Malyshev), and MicroPress, Inc (VTEX), who produced TEX distributions for many years, all appear to have ceased trading.

Some of the Y&Y add-on fonts are still distributed by the TEX Users Group (see Appendix 3 starting on page 243), or have been replaced by Open Source implementations, and there is a mailing list at the TUG web site for the support of former Y&Y users.

Symbols and conventions

There are several typographic conventions about how you represent computer-related material in print which are shown in Table 3 on page xxxvi. Typed commands, keywords, examples of input, and related text are in a fixed-width (monospace) font, like a typewriter, because that's how program code is usually displayed and edited (this also helps avoid ambiguities, as explained in section 4.7.1.1 on page 115). Special values, like numeric quantities represented by a name or symbol, are in italics, as in mathematics. Terms or references to products, programs, packages, and other components of LaTeX have their own typographic form. Finally there are some symbols like keyboard keys and menus, which are shown graphically.

Production note

This document is written and maintained in XML, using a customized version of the *DocBook 5* DTD. Conversions are made to HTML and LaTeX using XSLT3 scripts and Michael Kay's *Saxon* processor.

The complete source of the 2002 published version, with all ancillary files, is available online at `www.ctan.org/tex-archive/info/beginlatex/src/`. More recent versions are on the Silmaril web site at `latex.silmaril.ie/formattinginformation`. If you want to try processing it yourself you will need *Java* and *Saxon* in addition to a full installation of LaTeX.

This document is published under the terms and conditions of the GNU Free Documentation License. Details are in Appendix 5 starting on page 255.

Table 3 – Typographic notations used in this document

Notation	Meaning
MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL	These keywords have a specific mandatory meaning when shown in THIS STYLE (small capitals of the prevailing font), according to RFC 2119 (Bradner 1997).
`\command`	LaTeX commands (control sequences) that you type which perform an action or identify your text eg `\footnote{...}`
`\`*`length`*	Control sequences which store a dimension (a measurement in units), like `\textwidth`
`counter`	Values used for counting (whole numbers only), like `section`
term	The defining instance of a new technical term
`environment`	A LaTeX formatting or identification environment, like *`quotation`*
class	A LaTeX document class (available from CTAN), like **memoir**
package	A LaTeX add-on package (available from CTAN), like footmisc
product	A program or product name
`typewriter type`	Literal examples
value	Mnemonic examples of things you must type, where you have to supply real-life values of your own, like `\author{your name}` means you must replace your name with your own real name.
x	A specific key on your keyboard
Ctrl + x	Two or more keys pressed together, not separately
Esc q	Two keys pressed one after another
Submit	An on-screen button to click
Menu ❯ Submenu ❯ Item	A drop-down menu with items to select
␣	A normal space, just so it's visible