# 2

# *And*
# *a few more things…*

If life was perfect, we would not need this chapter. You would write perfect LaTeX, based on Chapter 1, no need to study how LaTeX works, what error messages mean… But life is not perfect, you will make mistakes, LaTeX will send messages, plain and mysterious.

In this chapter, we briefly explain how things work, the structure of LaTeX, the auxiliary files, the logical and visual design of an article, LaTeX error messages. See Appendix C for more detail. Finally, we present a long list of dos and don't to help you write good LaTeX.

## 2.1 Structure

LaTeX's core is a programming language called TeX, created by Donald E. Knuth, which provides low-level typesetting instructions. TeX comes with a set of fonts called *Computer Modern* (CM). The CM fonts and the TeX programming language form the foundation of a typical TeX system. TeX is extensible—new commands can be defined in terms of more basic ones. LaTeX is one of the best known extensions of TeX.

Figure 2.1: The structure of LaTeX.

The visual layout of a LaTeX document is primarily determined by the *document class*, such as amsart, article for articles, amsbook, book for books. Many journals, publishers, and schools have their own document classes for formatting articles, books, and theses.

Extensions of LaTeX are called *packages*. They provide additional functionality by adding new commands and environments, or by changing the way previously defined commands and environments work. It is essential that you find the packages that make your work easier. *The LaTeX Companion,* 2nd edition [56] discusses a large number of the most useful packages as of 2004.

The structure of LaTeX is illustrated in Figure 2.1. This figure suggests that in order to work with a LaTeX document, you first have to install TeX and the CM fonts, then LaTeX, and finally specify the document class and the necessary packages. The packages must include amsmath, amsthm, and so on. Of course, your LaTeX installation already includes all of these.

## 2.2   *Auxiliary files*

Figure 2.2 illustrates the steps in the production of a typeset document.

You start by opening an existing LaTeX source file or creating a new one with a text editor. For this discussion, the source file is called myart.tex. Once the source file is ready, you typeset it. Depending on the document class options you choose and the packages the document loads, you end up with at least three additional files:
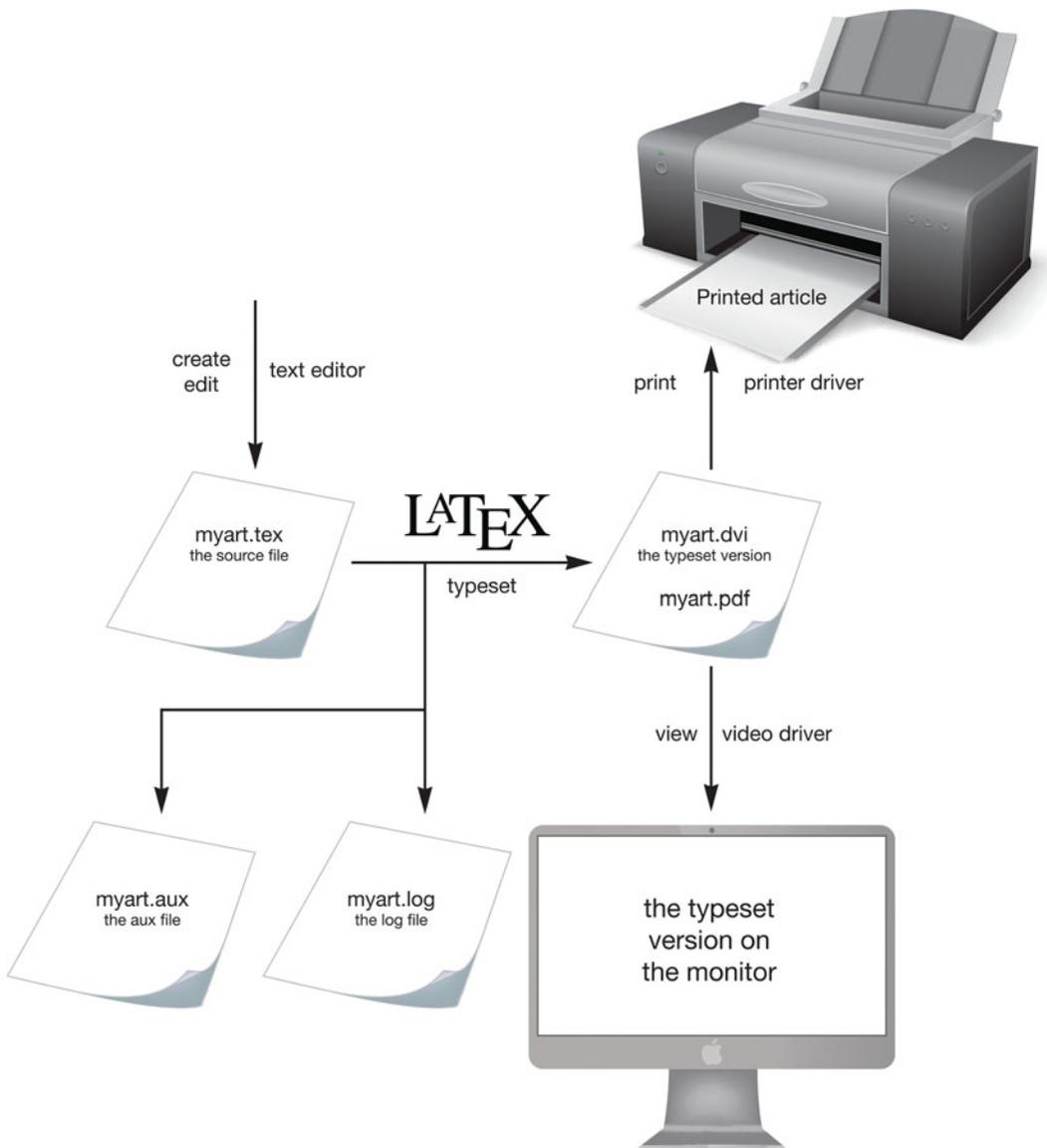
Figure 2.2: Using LaTeX.

1. `myart.pdf`    The typeset article in PDF format.

2. `myart.aux`    The auxiliary file, used by LaTeX for internal bookkeeping, including cross-references and bibliographic citations.

3. `myart.log`    The `log` file. LaTeX records the typesetting session in the `log` file, including any warnings and messages that appear on your monitor in the `log` window.

Your computer uses a *video driver* to display the typeset article on your monitor and a *printer driver* to print the typeset article on a printer. The video and printer drivers are computer and LaTeX implementation dependent.

It should be emphasized that of the three applications used, only one is the same for all computers and all implementations.

LaTeX always uses the `aux` file from the last typesetting. Here is an example. Your article has Theorems 1 (with `\label{T:first}`) and 2 (with `\label{T:main}`). The `aux` file has the two lines:

```
\newlabel{T:first}{{1}{1}}
\newlabel{T:main}{{2}{1}}
```

`\newlabel{T:first}{{1}{1}}` means that the label `T:first` is assigned the value 1 and appears on page 1. `\newlabel{T:main}{{2}{1}}` means that the label `T:main` is assigned the value 2 and appears on page 1. So the reference

```
see Theorems \ref{T:first} and \ref{T:first}.
```

is typeset as

see Theorems 1 and 2.

Now add a new theorem between Theorems 1 and 2. Typeset the article. In the typeset article, the three theorems are properly numbered, but it still contains the same typeset line:

see Theorems 1 and 2.

The `aux` file has the lines:

```
\newlabel{T:first}{{1}{1}}
\newlabel{T:main}{{3}{1}}
```

So at the next typesetting, the reference is displayed as

see Theorems 1 and 3.

## *2.3   Logical and visual design*

The typeset version of `firstarticle.tex` looks impressive on p. 4. To produce such articles, you need to understand that there are two aspects of article design: *visual* and *logical*.

As an example, let us look at a theorem from `firstarticle.tex` (see the typeset form of the theorem on page 4). You tell LaTeX that you want to state a theorem by using a `theorem` environment:

```
\begin{theorem}\label{T:technical}
Let $L$ be a finite lattice.


...
\end{theorem}
```

The logical part of the design is choosing to define a theorem by placing material inside a `theorem` environment. For the visual design, LaTeX makes hundreds of decisions. Could you have specified all of the spacing, font size changes, centering, numbering, and so on? Maybe, but would you *want* to? And would you want to repeat that process for every theorem in your document?

Even if you did, you would have spent a great deal of time and energy on the *visual design* of the theorem rather than on the *logical design* of your article. The idea behind LaTeX is that you should concentrate on what you have to say and let LaTeX take care of the visual design.

This approach allows you to easily alter the visual design by changing the document class (or its options, see Sections 9.5, 10.1.2, and 17.1). Section 9.1 provides some examples. If you code the visual design into the article—hard coding it, as a programmer would say—such changes are much harder to accomplish, for you and for the journal publishing the article.

For more on this topic, see Section C.4.

## *2.4   General error messages*

Now that you are ready to type your first document, we give you some pointers on using LaTeX.

You will probably make a number of mistakes in your first document. These mistakes fall into the following categories:

1.  Typographical errors, which LaTeX blindly typesets.

2.  Errors in formulas or in the formatting of the text.

3.  Errors in your instructions to LaTeX, that is, in commands and environments.

Typographical errors can be corrected by viewing and spell checking the source file, finding the errors, and then editing the typeset file. Mistakes in the second and third categories may trigger errors during the typesetting process, such as lines too wide of Section 1.4.

We now look at some examples of the third class of errors by deliberately introducing a number of mistakes into `firstarticle.tex` and examining the messages.

**Experiment 1.**   In `firstarticle.tex`, go to line 19 (use the `Go to Line` command of your editor) and remove the closing brace so that it reads `\begin{abstract`

When you typeset `firstarticle.tex`, LaTeX reports a problem:

```
{abstract We present a technical result for congruences on\ETC.
./firstarticle.tex:23:
Paragraph ended before \begin was complete.
<to be read again>
                   \par
l.23
```

Line 23 of the file is the line after `\maketitle`. The message informs you that the environment name was not completed.

Runaway `argument?` is a message that comes up often. It means that the argument of a command is either longer than expected or it contains material the argument cannot accept. Most often a closing brace solves the problem, as in this experiment.

**Experiment 2.**   Now restore line 19, then go to line 21 and change `\end{abstract}` to `\end{abstrac}` and typeset again. LaTeX informs you of another error:

```
./firstarticle.tex:21: LaTeX Error: \begin{abstract}
on input line 19 ended by \end{abstrac}.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...
l.21 \end{abstrac}
```

This is perfect. LaTeX correctly analyzes the problem and tells you where to make the change.

**Experiment 3.**   Correct the error in line 21, and introduce a new error in line 61. This line reads

```
z_1 \equiv y+ w \pmod{\delta}.
```

Change `\delta` to `\deta`. Now, when you typeset the document, LaTeX reports

```
./firstarticle.tex:61: Undefined control sequence.
<argument> {\operator@font mod}\mkern 6mu\deta

l.61 z_1 \equiv y+ w \pmod{\deta}
```

This mistake is easy to identify: `\deta` is a misspelling of `\delta`.

**Experiment 4.** In line 38, delete the closing brace of the \label command. This results in a message:

```
Runaway definition?
->E:cover\text {If $x$ is covered by $y,z \in L$ and\ETC.
! File ended while scanning definition of \df@label.
<inserted text>
                }
<*> firstarticle.tex
```

Undo the change to line 38.

**Experiment 5.** Add a blank line following line 61:

```
x+ z = z + z_1 \equiv z + (y+ w) = y+ z \pmod{\delta},
```

This change results in the message

```
./firstarticle.tex:62: Missing $ inserted.
<inserted text>
                $
l.62
```

There can be no blank lines within a displayed math environment. LaTeX catches the mistake, but the message itself is misleading.

**Experiment 6.** Add a $ before \pmod in line 61 (such an error often occurs when cutting and pasting a formula). You get the message:

```
./firstarticle.tex:61: Display math should end with $$.
<to be read again>
                    \penalty
l.61 z_1 \equiv y+ w $\pmod{\delta}
```

Maybe this could be more to the point?

---

**Tip** LaTeX's messages are not very useful with displayed formulas. Comment out some of the lines to try to localize the problem.

---

**Tip** Typeset often.

Typesetting my book *First Steps into* LaTeX with the closing brace of the first \caption command on line 480 of the source file missing, I get the error message

```
! Text line contains an invalid character.
l.1227 ...pletely irreducible^^?
```

where the reference is to line 1227, about 700 lines removed from the actual error. However, if the only thing I did before typesetting was to insert that figure with its incorrect caption command, at least I would know where to look for errors. If you make a dozen changes and then typeset, you may not know where to start.

## 2.5   *Errors in math*

Even in such a simple note there are opportunities for errors. To help familiarize yourself with some of the most commonly seen LATEX errors in formulas, we introduce mistakes into formulanote.tex.

**Experiment 1**   In line 6 of formulanote.tex, delete the third $ symbol; save the file under the name formulanotebad1.tex in the work folder.

Typeset formulanotebad1.tex. LATEX generates the following message:

```
! Missing $ inserted.
<inserted text>
                $
l.6 as $(u, v)$ and  (u, \infty
                                )$. Such an interval
```

LATEX reads (u, \infty) as text; but the \infty command instructs LATEX to typeset a math symbol, which can only be done in a formula. So LATEX offers to put a $ in front of \infty while typesetting the source file—it does not put the $ in the source file itself. LATEX attempts a cure, but in this example it comes too late, because the formula *should* start just before (u.

**Experiment 2**   In line 16 of formulanote.tex, delete the second } symbol and save it under the name formulanotebad2.tex in the work folder. This introduces an error: the closing brace of the subscript (see page 14) is missing. Now typeset the note. You get the message

```
Missing } inserted.
<inserted text>
                }
l.12 such as $\lim_{x \to \infty f(x)$
```

LATEX reports that a closing brace (}) is missing, but it is not sure where the brace should be. LATEX noticed that a subscript started with {, but it reached the end of the formula before finding a closing brace }. To remedy this, you must look in the formula for an opening brace { that is not balanced, and insert the missing closing brace }. Make the necessary change and typeset again to view the difference.

**Experiment 3**   In mathnote.tex, delete the two $ signs in line 19, that is, replace $a$ by a. Typeset the file. It typesets with no errors. Here is the last line of the typeset file you get:

> we assume that the function is defined and continuous in a neighborhood of a.

instead of

> we assume that the function is defined and continuous in a neighborhood of *a*.

This is probably the error most often made by beginners. There is no message by LaTeX and the typeset version looks good. Notice the difference in the shape of the letter a in the two cases. You need sharp eyes to catch such an error.

---

**Tip**   After an error is corrected, LaTeX can refuse to typeset your document. If your document is document.tex, look in the same folder for the *auxiliary file* document.aux that was created by LaTeX. Delete document.aux and **typeset twice**. See Section 2.2.

---

## 2.6   *Your errors: Davey's Dos and Don'ts*

Based on his many years of experience correcting LaTeX articles for the journal *Algebra Universalis*, Brian Davey collected the LaTeX mistakes most often made by authors. Here are some items from his list, divided into three categories.

### *Commands*

1. Place ALL custom commands and environments in the preamble!
   If you have trouble with custom commands, then you know where to find them.

2. Don't use \def; rather use \newcommand or \renewcommand.
   \def is a TeX command. It is like \newcommand (see Sections 1.9 and 14.1), but it can redefine an existing command. Redefining your own commands is bad enough, redefining a TeX command can be a disaster.

3. Do not simply type the name of an operator into a formula. Declare the appropriate operator; see Sections 1.9 and 14.1.
   For instance, do not type $length I$; it typesets as $lengthI$. It should be length $I$, typed as $\length I$. Of course, you have to add

   \DeclareMathOperator{\length}{length}

   to the preamble (see Section 1.8).

4. When you send a document to a coauthor or submit an article to a journal, remove all the custom commands not used.
   This is a real time saver for your coauthor and editor.

*Text*

1. Do not produce a list with horizontal and vertical spacing commands. Use a list environment; see Sections 3.8 and 4.2.

2. Do not type numbers for citations and internal references. Use \cite{...} for citations and \ref{...} for references. For references to equations, use \eqref; see Sections 1.7.1 and 5.3.

3. Do not number proclamations (see Section 4.4). Use the standard amsart environments for theorems, and so on, and let LaTeX number them.

4. When writing a document for a journal requiring a document class file, **do not**

   (a) change any of the size parameters: for instance, do not use options like 12pt to change the font size or the \setlength command to change any parameter of the page size;

   (b) insert vertical white space via \bigskip, \smallskip, \vskip, \vspace, etc, nor via your own custom commands. Do not adjust horizontal space without a very good reason.

      So if you want to display some text:

      Please, display this text.

   don't do this:

   ```
   \medskip
   \hspace*{6pt} Please, display this text.
   \medskip
   ```

   but rather

   ```
   \begin{itemize}
   \item[] Please, display this text.
   \end{itemize}
   ```

   or

   ```
   \begin{quote}
   Please, display this text.
   \end{quote}
   ```

5. Do not leave a blank line before \end{proof} or before a text environment (see Section 4.1).

6. Do not use the geometry package.

### Formulas

1. Do not leave a blank line before a displayed formula.

2. Don't use the symbol | in a set description, use the binary relation \mid; see Section 5.5.4.
   For instance, \{ x | x^2 < 2 \} typesets as $\{x|x^2 < 2\}$. The correct form is $\{x \mid x^2 < 2\}$, typed as $\{x \mid x^2 < 2\}$.

3. Don't put punctuation marks inside an inline math environment.
   For instance, $\sin x$. typed as $\sin x.$; use $\sin x$. This typesets as $\sin x$. Notice the smaller space between "$\sin x$." and "typed" and the wider space between "$\sin x$." and "This"; see Sections 1.3 and 3.2.2.

4. Don't use two or more displayed formulas one after another. Use an appropriate environment such as \align, \alignat, \gather, and so on (see Section 7.1.1).

5. Don't use \left\{, \right\}, \left(, \right), and so on, by default (see page 15 and Section 5.5.1 for the commands \left and \right). Even when \left and \right do not change the size of the symbol, they add extra space after the closing delimiter.

6. Use \colon for functions. For instance, $f(x) \colon x \to x^2$ typesets as $f(x)\colon x \to x^2$. If you type $f(x) : x \to x^2$, you get $f(x) : x \to x^2$; the spacing is bad.

7. Use \[ and \] (or equation*) to type a displayed math environment (see Section 1.7) rather than the old TeX $$ matched by $$. While display math produced via the latter does work properly most of the time, there are some LaTeX commands that do not; for example, \qedhere.

8. Do not use the center environment to display formulas.

9. Use \dots first and let LaTeX make the decision whether to use \dots or \cdots; see page 15 and Section 5.4.3. If LaTeX gets it wrong, then use \cdots or \ldots.

10. If you can, avoid constructs (for instance, $\overset{\text{up}}{\to}$) in inline formulas that disrupt the regular line spacing. Although LaTeX automatically leaves room for it, it does not look good, as a rule.

# Springer