

# L<sup>A</sup>T<sub>E</sub>X in 10 Steps

## 1. It is not a WYSIWYG editor

Latex or L<sup>A</sup>T<sub>E</sub>X (usually written with this peculiar mix of uppercase and lowercase letters) is a word processor in which you cannot see the final result as you type, i.e., it is not a WYSIWYG editor (What You See Is What You Get). Unlike such word processors, like Word or OpenOffice.org Writer, you need to use different commands to tell it what to do with the text, achieving great precision, a considerable reduction in file size, and the occasional headache searching for the right command.

To work with L<sup>A</sup>T<sub>E</sub>X, you need to install two programs. One is the L<sup>A</sup>T<sub>E</sub>X distribution itself, which on Windows is called MiK<sub>T</sub>E<sub>X</sub>, and the other is the editor in which you write the texts and send them to MiK<sub>T</sub>E<sub>X</sub> for processing. There are several good ones: WinEdt, Texmaker, or T<sub>E</sub>XnicCenter. You write the text in the editor, tell it to send it to L<sup>A</sup>T<sub>E</sub>X, and it typesets the text and returns it to you in PDF or another format.

## 2. Some characters are special

A drawback of having to explain to the program what you want to do through commands is that some characters are reserved for those tasks, and if you want to type them, you need to use a command. These characters are:

Symbol	Command	Use
\	<code>\textbackslash</code>	Introduces a command
{	<code>\{</code>	Delimits the properties of a command
}	<code>\}</code>	Delimits the properties of a command
#	<code>\#</code>	Numbers the arguments of a command
&	<code>\&amp;</code>	Separates the columns of a table
%	<code>\%</code>	For commenting in the source code
~	<code>\~{}</code>	Prevents a line break between two words
\$	<code>\\$</code>	Delimits an equation
_	<code>\_</code>	In equations, subscript
^	<code>\^{}</code>	In equations, superscript

There are other characters that need to be specified through a command either because they are not on the keyboard or because L<sup>A</sup>T<sub>E</sub>X requires us to specify them. Here is a short list of them:

Symbol	Command	Use
« and »	<code>&lt;&lt; and &gt;&gt;</code>	French quotes
‘ and ’	<code>‘ and ’</code>	Single English quotes
“ and ”	<code>“ and ”</code>	Double English quotes
-	<code>-</code>	Hyphen
—	<code>---</code>	Em dash
-	<code>-\$-\$</code>	Minus sign
...	<code>\dots</code>	Ellipsis
°	<code>\textdegree</code>	Degree

Symbol	Command	Use
†	<code>\dag</code>	Dagger or obelisk
‡	<code>\ddag</code>	Double dagger
§	<code>\S</code>	Section sign
¶	<code>\P</code>	Pilcrow
®	<code>\textregistered</code>	Registered trademark
©	<code>\copyright</code>	Copyright
™	<code>\texttrademark</code>	Trademark
£	<code>\pounds</code>	Pound

### 3. The fundamentals are defined at the beginning

Writing a  $\text{\LaTeX}$  document is very simple. At the beginning, you need to define the fundamental characteristics of the document, such as the type of paper, the default font size, or its structure. Everything is specified in the following command:

```
\documentclass[options]{structure}
```

When writing a long document, structure is important.  $\text{\LaTeX}$  allows you to divide your work into (from most to least important): parts (`\part`), chapters (`\chapter`), sections (`\section`), subsections (`\subsection`), paragraphs (`\paragraph`), and subparagraphs (`\subparagraph`).

To give them a title, you just need to put it in braces after the division code:

```
\section{Title}
```

Depending on the type of document you want to write, you will need more or fewer of these divisions, so  $\text{\LaTeX}$  provides four different types of structures:

- **article**: The most used for short articles; the largest division it allows is the section.
- **report**: In addition to allowing all divisions, it allows adding an abstract (`\begin{abstract}`) and marking chapters as appendices (`\appendix`).
- **book**: Very similar to report except that it does not allow adding the abstract and also typesets the document as a book, always starting chapters on an odd page.
- **slides**: For making slide presentations.

The options, separated by commas, that  $\text{\LaTeX}$  allows to characterize your document are:

- **Font size**: 10pt, 11pt, 12pt... If nothing is specified, the default size is 10 points.
- **Paper size**: a4paper, a5paper, b5paper, executivepaper, legalpaper, or letterpaper.
- **Number of columns**: You can specify that the text be arranged in one column (`onecolumn`) or two (`twocolumn`).
- **Number of sides**:  $\text{\LaTeX}$  typesets the text in the most readable way depending on whether we want to use one (`oneside`) or both sides of the paper (`twoside`).
- **Orientation**:  $\text{\LaTeX}$  will place the page in landscape mode if we indicate it with (`landscape`).
- **Equation alignment**: By default,  $\text{\LaTeX}$  centers equations and right-aligns the equation number. If we write (`fleqn`), the equations will be left-aligned, and (`leqno`) will left-align the equation numbers.

- **Chapter start:** You can force, regardless of the structure set at the beginning, chapters to start on an odd page (`openright`) or on any page (`openany`).
- **Title page:** Except in the article structure, you can tell it to reserve a page for the title (`titlepage`).

After defining the document type, you need to load the extensions you need using the command:

```
\usepackage{name}
```

Extensions are add-ons or modifications that people make to  $\text{\LaTeX}$ . There are all kinds, and  $\text{MiKTeX}$  usually installs them automatically when needed. In my opinion, these four are essential:

```
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}
\usepackage{amsmath}
\usepackage{graphicx}
```

The `latin1` option of `inputenc` allows  $\text{\LaTeX}$  to interpret characters with accents, umlauts, tildes, cedillas. . . making text writing much more convenient, because otherwise we would have to indicate each special character through a code (for example: á as `\'a`).

The `babel spanish` package gives  $\text{\LaTeX}$  the hyphenation rules for Spanish so that it correctly splits words at the end of a line.

`amsmath` loads a large number of mathematical symbols into  $\text{\LaTeX}$ 's memory, needed for writing proper equations.

`graphicx` increases the range of things you can do with images.

Another useful package is `eurosym`, which allows us to insert the euro symbol every time we write the command `\euro`.

If we want to give the document a title, indicate the author, and things like that, this is the moment. The following commands are used:

```
\title{document name}
\author{author name}
\date{creation date}
\thanks{acknowledgments}
```

And now all that remains is the hardest part, which is writing the document.  $\text{\LaTeX}$  understands that all the text that makes up the document is between the commands:

```
\begin{document}
bla bla bla document bla bla bla
\end{document}
```

So please be kind and write everything inside, as anything written outside these two commands will not appear in the final document.

Something that always looks great is letting  $\text{\LaTeX}$  compose the document title from the data we provided in the preamble. To do this, right after opening the document environment, write the command:

```
\maketitle
```

## 4. Units of measurement in $\LaTeX$

Many commands require you to specify sizes to work.  $\LaTeX$  accepts several units of measurement. Some are absolute:

Symbol	Name	Description
sp	Scaled point	The smallest unit in $\TeX$ (65536 sp = 1 pt)
pt	Point	Equals 0.351 mm or 1/(72.27) inches
bp	PostScript or big point	Equals 0.353 mm or 1/72 inches
dd	Didôt point	Equals 0.376 mm or 1/72 of the French inch
mm	Millimeter	Equals 2.845 points
pc	Pica	Equals 4.218 mm or 12 points
cc	Cicero	Equals 4.531 mm or 12 Didôt points
cm	Centimeter	Equals 10 mm or 2.371 picas
in	Inch	Equals 25.4 mm or 72.27 points or 6.022 picas

And others are relative to the font in use:

Symbol	Name	Description
ex	Ex	Height of the letter x
em	Em	Width of the letter M
mu	Math unit	18 mu equal one em

To use them, simply put the number followed by the unit with no space between them. For example, 14 millimeters would be `14mm`.

## 5. $\LaTeX$ takes care of typesetting the paragraphs

When we send a document to  $\LaTeX$  for typesetting, it takes care of laying it out following a complex set of rules to make the final document as attractive and harmonious as possible. By default,  $\LaTeX$  justifies paragraphs, hyphenating words across lines if necessary.

If we want to change the text alignment, we can begin and end a text block as follows:

<code>\begin{flushleft}</code>	<code>\begin{center}</code>	<code>\begin{flushright}</code>
Left-aligned text	Centered text	Right-aligned text
<code>\end{flushleft}</code>	<code>\end{center}</code>	<code>\end{flushright}</code>

$\LaTeX$  also distinguishes between a line break and a paragraph break. The difference between them is that a paragraph break only occurs when changing concepts in an exposition, while a line break only occurs when you want to express a longer pause than a full stop.

In  $\LaTeX$ , a line break is made by placing two consecutive backslashes (`\\`), while a paragraph break is made by leaving a blank line (pressing the Enter key twice). If you press the Enter key only once,  $\LaTeX$  interprets it as a space. If you press it more than twice,  $\LaTeX$  interprets it as a simple paragraph break, no matter how many times you pressed the key.

If we want to increase the space between two paragraphs on a specific occasion, we can use the command:

```
\vspace{size}
```

which separates the paragraphs by the space we specify.

If what we want is to modify the space between all paragraphs in the document, in the preamble (before `\begin{document}`) we can place the following command:

```
\parskip=size
```

Similarly, with spaces between words,  $\text{\LaTeX}$  only places one space between two words no matter how many times we press the space bar. If we want a wider space, we can use the command:

```
\hspace{length}
```

There is another break, which is the page break.  $\text{\LaTeX}$  provides two commands for this purpose:

```
\newpage
```

for a regular page break and:

```
\cleardoublepage
```

for a break to the next odd page.

Style guidelines recommend that the beginning of a paragraph be indented. The indentation length that  $\text{\LaTeX}$  adds by default at the beginning of a paragraph can be modified by writing in the document's preamble:

```
\parindent=size
```

You can even eliminate it at a specific point by writing at the beginning of the paragraph:

```
\noindent
```

You can also change the line spacing of the text, which is 1 by default. To do this, in the preamble you need to redefine the value using the code:

```
\renewcommand*{\baselinestretch}{amount}
```

where *amount* can be any decimal value.

Finally,  $\text{\LaTeX}$  chooses the appropriate margins for each paper and font size to make reading the text as comfortable as possible. These sizes can be modified in the preamble using the command:

```
\addtolength{property}{size}
```

where the properties that can be modified take a positive amount to increase or a negative amount to decrease. All of them are listed in the following table:

<b>Property</b>	<b>Meaning</b>
<code>\textwidth</code>	Width of the text
<code>\textheight</code>	Height of the text
<code>\hoffset</code>	Initial position of the left margin
<code>\voffset</code>	Initial position of the top margin
<code>\evensidemargin</code>	Initial position of the outer margin on even pages
<code>\oddsidemargin</code>	Initial position of the outer margin on odd pages

## 6. Playing with fonts is essential

L<sup>A</sup>T<sub>E</sub>X was created to be a personal professional typesetter. That is why it has all the modifications that can be applied to a typeface:

Command	Name
<code>\textbf{}</code>	Bold
<code>\textit{}</code>	Italic
<code>\textsl{}</code>	Slanted
<code>\underline{}</code>	Underlined
<code>\emph{}</code>	Emphasized
<code>\textsc{}</code>	Small caps
<code>\$^\text{}</code>	Superscript
<code>\$_\text{}</code>	Subscript
<code>\texttt{}</code>	Typewriter
<code>\textsf{}</code>	Sans serif

You can also change the font size using the following commands, listed from smallest to largest, with the normal size corresponding to what we declared in the preamble:

```
\tiny < \scriptsize < \footnotesize < \small <
\normalsize
< \large < \Large < \LARGE < \huge < \Huge
```

The way to insert these commands in a L<sup>A</sup>T<sub>E</sub>X document is different so that they can be combined with other codes, such as those for changing the shape of letters. These commands are placed between braces, at the beginning of the phrase you want to change:

```
{\Large bla bla bla text}
```

## 7. Using images

Inserting an image in L<sup>A</sup>T<sub>E</sub>X is as simple as writing the command:

```
\includegraphics{options}{file}
```

Where in *file* you need to put the path to the image on your computer, unless it is located in the same folder as the L<sup>A</sup>T<sub>E</sub>X file itself.

Many of the options are related to the image sizes:

- **width=size**: Using this command, we can specify the width we want the image to have; it will be scaled accordingly.
- **height=size**: Similarly, we can do the same with the image height.
- **keepaspectratio**: If set to **=false**, when we change the size of an image, it will not maintain its width-to-height ratio.
- **scale=number**: Displays the image at a size proportional to the value we specify. For example, if we put 2, the image will be twice its original size.
- **clip**: If declared as **=true**, the image will be clipped according to the dimensions declared in the other options.

- **draft**: Another boolean parameter. If set to `=true`, instead of the image, a box of the same size appears with the file name written in it. Very useful when printing drafts.

Inserting an image is that easy, although we also have the option of including an image the way professionals do, inside a floating object.

A floating object is a space that the program reserves on the page to include an image or a table with its caption. L<sup>A</sup>T<sub>E</sub>X always places the floating object in the location that is most comfortable for reading, so you need to be careful when referring to the image because it might not be where you expect. The code to include an image inside a floating object is:

```
\begin{figure}[Position]
\includegraphics{options}{file}
\caption{Caption}
\end{figure}
```

As mentioned, L<sup>A</sup>T<sub>E</sub>X places the image in the most readable location. However, we can suggest where we want it to go in what the code calls *Position*. If you put **h** (for *here*) between those brackets, L<sup>A</sup>T<sub>E</sub>X will try to place the image in the same position it occupies in the source code. If you put **t** or **b** (for *top* and *bottom*), the figure will be placed, respectively, at the top or bottom of the nearest page. Finally, you can also specify **p** (for *page*) so that the figure goes to a new page containing only images. If for some reason you really want to place a figure in a specific location and L<sup>A</sup>T<sub>E</sub>X ignores your suggestion, you can force it to comply by adding an exclamation mark to the position letter, e.g., **h!**.

Floating objects also allow adding a caption so that the author can explain what they contain. You just need to add the `\caption` command and, between braces, the caption you want to include.

## 8. Creating tables

Writing a table in L<sup>A</sup>T<sub>E</sub>X is very intuitive since it only uses two symbols as separators: `&` to separate columns and `\\` to separate rows. The `tabular` command declares that a table begins and is written row by row, as shown in the code:

```
\begin{tabular}{number and type of columns}
Cell & Cell & Cell \\
Cell & Cell & Cell \\
\end{tabular}
```

The `tabular` command requires that you indicate the number of columns and the alignment of all cells in each column, and this is as easy as putting one letter per column you want. Which letter? It depends on the column format: **l** if you want all cells in the column to be left-aligned, **c** if you want them centered, **r** if you want them right-aligned, and **p{width}** if you want to force L<sup>A</sup>T<sub>E</sub>X to typeset the column with a specific width (otherwise, the width would be that of the cell with the longest text).

If you want a cell to span multiple columns, the following code exists:

```
\multicolumn{number}{type}{cell text}
```

which is placed when it is time to write the specific cell. *number* is the number of columns it will span and *type* is the cell alignment.

Regarding drawing cell borders with lines, L<sup>A</sup>T<sub>E</sub>X gives you total flexibility to place them where and when you want. For vertical lines, you just need to add a pipe symbol (`|`) next to the

letter representing the column in the initial declaration. If placed to the left, the column will have a vertical line on the left; if placed between two letters, a vertical line will appear between the two columns represented by those letters. For example, a table with three left-aligned columns with lines on both sides of each column: `{|l|l|l|}`. If you want the line to be double, you can similarly put two pipe symbols (`||`).

For horizontal lines, the `\hline` command is used, placed at the end of each row (after the `\\` separator) that you want to have a bottom border. For the top line above the first row, you can place the `\hline` command after the declaration of the number and type of columns. If you want a double horizontal line, you can use the `\hline` command twice.

Tables can also be placed inside a floating object. The code is:

```
\begin{table}[Position]
\begin{tabular}{number and type of columns}
cells cells and more cells
\end{tabular}
\caption{Caption}
\end{table}
```

where everything is the same as in floating objects for figures.

## 9. Equations. The great advantage of L<sup>A</sup>T<sub>E</sub>X

Since it was its original purpose, L<sup>A</sup>T<sub>E</sub>X is designed to write mathematical equations easily and quickly so that they look truly beautiful. There are three different ways to write an equation in L<sup>A</sup>T<sub>E</sub>X:

	<code>\begin{equation}</code>	<code>\begin{equation*}</code>
<code>\$</code> Inline equation <code>\$</code>	Numbered equation	Unnumbered equation
	<code>\end{equation}</code>	<code>\end{equation*}</code>

- **Between dollar signs:** Used when you need to include an occasional equation within the text.
- **`{equation}`:** Using this code, L<sup>A</sup>T<sub>E</sub>X places the equation in a separate, centered paragraph, numbered on the right.
- **`{equation*}`:** Exactly the same as the previous one, except that the equation is not numbered and is not counted when numbering the rest of the equations.

As typographic conventions dictate, L<sup>A</sup>T<sub>E</sub>X italicizes all text within an equation. If you want to introduce text that should not be in italics within an equation, such as units of measurement, you need to use the command `\text{text}`.

When writing symbols, remember that it is very important to install the `amsmath` extension to ensure that all important symbols are available. Each symbol has its own command, which you do not need to memorize since most L<sup>A</sup>T<sub>E</sub>X editing programs include tables for inserting them. In any case, there are many useful symbol lists available on the internet (the most comprehensive: Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List).

Even so, there are a few special commands that we will summarize in the following table:

Name	Code	Example
Fractions	<code>\frac{numerator}{denominator}</code>	$\frac{a+b}{c}$
Subscript	<code>_</code>	$a_n$
Superscript	<code>^</code>	$a^2$
Square root	<code>\sqrt{radicand}</code>	$\sqrt{2}$
General roots	<code>\sqrt[index]{radicand}</code>	$\sqrt[3]{2}$
Binomial coefficient	<code>\binom{upper}{lower}</code>	$\binom{1}{6}$
Summation	<code>\sum_{from}^{to}</code>	$\sum_{i=1}^n$
Product	<code>\prod_{from}^{to}</code>	$\prod_{i=1}^n$
Definite integral	<code>\int_{from}^{to}</code>	$\int_0^\infty$
Double integral	<code>\iint_{from}^{to}</code>	$\iint_0^\infty$
Triple integral	<code>\iiint_{from}^{to}</code>	$\iiint_0^\infty$
Limit	<code>\lim_{}</code>	$\lim_{x \rightarrow \infty}$

When an equation containing a fraction or another element with a special height uses parentheses, brackets, braces, etc., they will not change their size, appearing very small compared to the element. To make them adjust their height to that of the fraction or element, you need to use the commands:

```
\left§ \right§
```

Where § can be: (, {, [, <, their opposites >, ], }, ) or any other of the dozens of delimiters available in L<sup>A</sup>T<sub>E</sub>X tables.

To write a matrix or a determinant, we also need a special command. Within the mathematical environment, the following code is written, very similar to that of tables:

```
\begin{array}{number and type of columns}
Element & Element & Element \\
Element & Element & Element \\
\end{array}
```

Then you just need to declare with a delimiter whether it is a matrix ( ) or a determinant ||. The `array` command is also very useful for writing multiple equations in different rows, such as in a system of equations. These equations, within the `\begin{equation} \end{equation}` environment, are numbered as if they were a single one. If you want to write multiple equations within the same environment and have each one with its own numbering, you need to use the code:

```
\begin{eqnarray}
Equation 1 \\
Equation 2 \\
\end{eqnarray}
```

## 10. L<sup>A</sup>T<sub>E</sub>X organizes everything for you

When typesetting a document, L<sup>A</sup>T<sub>E</sub>X counts and stores in its memory all the chapters, references, figures, tables, etc. that we have written, allowing us to take advantage of this. The most common use is creating an index or table of contents with the list of parts, chapters, sections, and subsections (depending on the class, because for example, in article there are no chapters) that make up the document. To do this, you just need to write the code:

```
\tableofcontents
```

in the place where you want to insert the index.

L<sup>A</sup>T<sub>E</sub>X puts the division title in the index. If for some reason you want it to display something different, when naming the division you can put the name you want to appear in the index between brackets:

```
\section[Title for the index]{Title of the division}
```

Similarly, you can create indexes of the figures and tables in the document. Their respective commands are:

```
\listoffigures  
\listoftables
```

In these indexes, the figure or table number that L<sup>A</sup>T<sub>E</sub>X assigns by default and the caption or description that we had placed in `\caption` appear. As before, this can be changed using the following command inside the floating object:

```
\caption[Title for the index]{Caption}
```

By introducing labels using the command `\label{label name}` at the beginning of a chapter, section, or other division, in a floating object, or in an equation, we can then refer to the labeled structure in the text using the command `\ref{label name}`, which will be replaced by the number that L<sup>A</sup>T<sub>E</sub>X assigned to the cited object. Another option is the command `\pageref{label name}`, which allows us to refer to the page where it is located.

Bibliographic citations are another thing you do not have to worry about when working with L<sup>A</sup>T<sub>E</sub>X. The traditional way to write the bibliography is to use the following command where you want it to appear in your work:

```
\begin{thebibliography}{number}  
\bibitem[alternative name]{label name} Bibliographic citation  
Bibliography  
and more bibliography  
\end{thebibliography}
```

You need to put a number in braces after opening the `thebibliography` environment. The number of digits in that number is the space that L<sup>A</sup>T<sub>E</sub>X will reserve in its memory for the bibliography numbering, so it must be equal to or longer than the longest number or alternative name in our bibliography.

Citations are introduced within the environment using the `\bibitem` command. We give them whatever format we want and also assign them a label in braces to cite them whenever we want in the text using the command `\cite{label name}`. L<sup>A</sup>T<sub>E</sub>X will number the bibliographic citations in the order they appear in the text. If you want something other than a number to appear (it will not be counted in the numbering), just put it in brackets after the command as shown in the codes above.

There is another way to write the bibliography that is much more convenient when working with a large number of citations. It consists of creating a `.bib` file separate from the document file and saved in the same folder, in which all bibliographic citations will be entered with their label and with the information broken down into author, title, year, etc. In practice, everyone uses independent programs to write and manage that file, such as JabRef. Once we have all the bibliography in that file, we just need to cite the references with the command `\cite{label name}` and, where we want to place the bibliography list, insert the code:

```
\bibliography{file name}  
\bibliographystyle{style}
```

Writing the bibliography this way has the advantage that a specific style can be automatically applied to all citations in the document using the `\bibliographystyle{style}` command. There are many styles, some of them official, for writing citations with a specific format for each of their fields. Using different styles also allows you to change the ordering of citations (alphabetically, by date, etc.) or the way you refer to them in the text (author name and year, a specific code, number with a different format, etc.). There are four standard styles:

- **plain**: Citations numbered alphabetically.
- **unsrt**: Citations numbered in the order they appear in the text.
- **alpha**: Citations identified by the author's name and publication year.
- **abbrv**: Citations numbered alphabetically but with abbreviated author and journal names.

### **Much more information and sources:**

These are the books I consulted to write this introduction:

*El libro de L<sup>A</sup>T<sub>E</sub>X*, Bernardo Cascales, Ed. Pearson Educación, Madrid, 2003.

*The not so short introduction to L<sup>A</sup>T<sub>E</sub>X<sub>2</sub> $\epsilon$* , Tobias Oetiker,

<http://tobi.oetiker.ch/lshort/lshort.pdf>

They are very good references for learning more about L<sup>A</sup>T<sub>E</sub>X. For specific questions, the main source of information on any L<sup>A</sup>T<sub>E</sub>X topic is the internet, which is full of forums, wikis, and tutorials. Since anyone can contribute an extension to the program, you can also find many solutions to common problems there. L<sup>A</sup>T<sub>E</sub>X is a very powerful program and you can get a lot out of it.