

Documentation Home
(/learn/Main_Page)

Learn LaTeX in 30 minutes
(/learn/Learn_LaTeX_in_30_minutes)

Q Search help library....

Suggest new doc

ShareLaTeX guides

Creating a document in ShareLaTeX
(/learn/Creating_a_document_in_ShareLaTeX)

Uploading a project
(/learn/Uploading_a_project)

Copying a project
(/learn/Copying_a_project)

Creating a project from a template
(/learn/Creating_a_project_from_a_template)

Including images in ShareLaTeX
(/learn/Including_images_in_ShareLaTeX)

Exporting your work from
ShareLaTeX
(/learn/Exporting_your_work_from_ShareLaTeX)

Using bibliographies in ShareLaTeX
(/learn/Using_bibliographies_in_ShareLaTeX)

Sharing your work with others
(/learn/Sharing_your_work_with_others)

Debugging Compilation timeout
errors
(/learn/Debugging_Compilation_timeout_errors)

Knowledge Base
(/learn/Kb/Knowledge_Base)

LaTeX Basics

Creating your first LaTeX document
(/learn/Creating_a_document_in_LaTeX)

Choosing a LaTeX Compiler
(/learn/Choosing_a_LaTeX_Compiler)

Paragraphs and new lines
(/learn/Paragraphs_and_new_lines)

Bold, italics and underlining
(/learn/Bold_italics_and_underlining)

Lists (/learn/Lists)

Errors (/learn/Errors)

Mathematics

Mathematical expressions
(/learn/Mathematical_expressions)

Subscripts and superscripts
(/learn/Subscripts_and_superscripts)

Brackets and Parentheses
(/learn/Brackets_and_Parentheses)

Fractions and Binomials
(/learn/Fractions_and_Binomials)

Aligning Equations
(/learn/Aligning_equations)

Operators (/learn/Operators)

Spacing in math mode
(/learn/Spacing_in_math_mode)

Integrals, sums and limits
(/learn/Integrals_sums_and_limits)

Commands

Latex typesetting is made by using special tags or commands that provide a handful of ways to format your document. Sometimes standard commands are not enough to fulfil some specific needs, in such cases new commands can be defined and this article explains how.

Contents

- 1 Introduction
- 2 Commands
- 3 Defining a new command
 - 3.1 Simple commands
 - 3.2 Commands with parameters
 - 3.3 Commands with optional parameters
- 4 Overwriting existing commands
- 5 Further reading

Introduction

Most of the \LaTeX commands are simple words preceded by a special character.

In a document there are different types of `\textbf{commands}` that define the way the elements are displayed. This commands may insert special elements: α β γ

In a document there are different types of **com-
mands** that define the way the elements are
displayed. This command may insert special
elements: $\alpha\beta\gamma$

(/learn/File:CommsAndEnvsEx1.png)

In the previous example there are different types of commands. For instance, `\textbf` will make boldface the text passed as parameter to the command. In mathematical mode there are special commands to display Greek characters.

➔ Open an example in ShareLaTeX (<https://www.sharelatex.com/project/new/template?zipUrl=/project/530513e2f1814bd50b0001a5/download/zip&templateName=Commands&compiler=pdfplat>)

Commands

Commands are special words that determine \LaTeX behaviour. Usually this words are preceded by a backslash and may take some parameters.

```
Example a list
\begin{itemize}
  \item[\S] First item
  \item Second item
\end{itemize}
```

Example a list

§ First item

• Second item

(/learn/File:CommsAndEnvs2.png)

Display style in math mode
(/learn/Display_style_in_math_mode)

List of Greek letters and math symbols
(/learn/List_of_Greek_letters_and_math_symbols)

Mathematical fonts
(/learn/Mathematical_fonts)

Figures and tables

Inserting Images
(/learn/Inserting_Images)

Tables (/learn/Tables)

Positioning Images and Tables
(/learn/Positioning_images_and_tables)

Lists of Tables and Figures
(/learn/Lists_of_tables_and_figures)

Drawing Diagrams Directly in LaTeX
(/learn/Picture_environment)

TikZ package (/learn/TikZ_package)

References and Citations

Bibliography management in LaTeX
(/learn/Bibliography_management_in_LaTeX)

Bibliography management with biblatex
(/learn/Bibliography_management_in_LaTeX)

Biblatex bibliography styles
(/learn/Biblatex_bibliography_styles)

Biblatex citation styles
(/learn/Biblatex_citation_styles)

Bibliography management with natbib
(/learn/Bibliography_management_with_natbib)

Natbib bibliography styles
(/learn/Natbib_bibliography_styles)

Natbib citation styles
(/learn/Natbib_citation_styles)

Bibliography management with bibtex
(/learn/Bibliography_management_with_bibtex)

Bibtex bibliography styles
(/learn/Bibtex_bibliography_styles)

Languages

International language support
(/learn/International_language_support)

Quotations and quotation marks
(/learn/Typesetting_quotations)

Arabic (/learn/Arabic)

Chinese (/learn/Chinese)

French (/learn/French)

German (/learn/German)

Greek (/learn/Greek)

Italian (/learn/Italian)

Japanese (/learn/Japanese)

Korean (/learn/Korean)

Portuguese (/learn/Portuguese)

Russian (/learn/Russian)

Spanish (/learn/Spanish)

The command `\begin{itemize}` defines an *environment*, see the article about environments (/learn/Environments) for a better description. Below the *environment* declaration is the command `\item`, this tells L^AT_EX that this is an item part of a list, and thus has to be formatted accordingly, in this case by adding a special mark (a small black dot called bullet) and indenting it.

Some commands need one or more parameters to work. The example at the introduction includes a command to which a parameter has to be passed, `\textbf`; this parameter is written inside braces and it's necessary for the command to do something.

There are also optional parameters that can be passed to a command to change its behaviour, this optional parameters have to be put inside brackets. In the example above, the command `<item>\item[\S]</item>` does the same as *item*, except that inside the brackets is `\S` that changes the black dot before the line for a special character.

➔ Open an example in ShareLaTeX (<https://www.sharelatex.com/project/new/template?zipUrl=/project/530513e2f1814bd50b0001a5/download/zip&templateName=Commands&compiler=pdfflatex>)

Defining a new command

L^AT_EX is shipped with a huge amount of commands for a large number of tasks, nevertheless sometimes is necessary to define some special commands to simplify repetitive and/or complex formatting.

Simple commands

New commands are defined by `\newcommand` statement, let's see an example of the simplest usage.

```
\newcommand{\R}{\mathbb{R}}
```

The set of real numbers are usually represented by a blackboard bold capital *r*: `\(\R \)`.

The set of real numbers are usually represented by a blackboard bold capital *r*.
(/learn/File:CommsAndEnvsEx3.png)

The statement `\newcommand{\R}{\mathbb{R}}` has two parameters that define a new command

`\R`

This is the name of the new command.

`\mathbb{R}`

This is what the new command does. In this case the letter *R* will be written in blackboard boldface style. (to use *mathbb* is needed the package *amssymb*)

After the command definition you can see how the command is used in the text. Even tough in this example the new command is defined right before the paragraph where it's used, is a good practice put all your user-defined commands in the preamble of your document.

➔ Open an example in ShareLaTeX (<https://www.sharelatex.com/project/new/template?zipUrl=/project/530513e2f1814bd50b0001a5/download/zip&templateName=Commands&compiler=pdfflatex>)

Commands with parameters

It is also possible to create new commands that accept some parameters.

```
\newcommand{\bb}[1]{\mathbb{#1}}
```

Other numerical systems have similar notations. The complex numbers `\(\bb{C} \)`, the rational numbers `\(\bb{Q} \)` and the integer numbers `\(\bb{Z} \)`.

Other numerical systems have similar notations. The complex numbers *C*, the rational numbers *Q* and the integer numbers *Z*.
(/learn/File:CommsAndEnvsEx4.png)

Document structure

Sections and chapters
(/learn/Sections_and_chapters)

Table of contents
(/learn/Table_of_contents)

Cross referencing sections and equations
(/learn/Cross_referencing_sections_and_equations)

Indices (/learn/Indices)

Glossaries (/learn/Glossaries)

Nomenclatures
(/learn/Nomenclatures)

Management in a large project
(/learn/Management_in_a_large_project)

Multi-file LaTeX projects (/learn/Multi-file_LaTeX_projects)

Hyperlinks (/learn/Hyperlinks)

Formatting

Lengths in L^AT_EX
(/learn/Lengths_in_LaTeX)

Headers and footers
(/learn/Headers_and_footers)

Page numbering
(/learn/Page_numbering)

Paragraph formatting
(/learn/Paragraph_formatting)

Line breaks and blank spaces
(/learn/Line_breaks_and_blank_spaces)

Text alignment
(/learn/Text_alignment)

Page size and margins
(/learn/Page_size_and_margins)

Single sided and double sided documents
(/learn/Single_sided_and_double_sided_documents)

Multiple columns
(/learn/Multiple_columns)

Counters (/learn/Counters)

Code listing (/learn/Code_listing)

Code Highlighting with minted
(/learn/Code_Highlighting_with_minted)

Using colours in LaTeX
(/learn/Using_colours_in_LaTeX)

Footnotes (/learn/Footnotes)

Margin notes (/learn/Margin_notes)

Fonts

Font sizes, families, and styles
(/learn/Font_sizes_families_and_styles)

Font typefaces
(/learn/Font_typefaces)

Supporting modern fonts with X_YL^AT_EX (/learn/XeLaTeX)

Presentations

Beamer (/learn/Beamer)

Powerdot (/learn/Powerdot)

Posters (/learn/Posters)

The line `\newcommand{\bb}[1]{\mathbb{#1}}` defines a new command that takes one parameter.

\bb

This is the name of the new command.

[1]

The number of parameters the new command will take.

\mathbb{#1}

This is what the command actually does. In this case the parameter, referenced as #1, will be written using blackboard boldface characters. If the defined new command needs more than one parameter, you can refer each parameter by #1, #2 and so on.

➔ Open an example in ShareLaTeX (<https://www.sharelatex.com/project/new/template?zipUrl=/project/530513e2f1814bd50b0001a5/download/zip&templateName=Commands&compiler=pdfflatex>)

Commands with optional parameters

User-defined commands are even more flexible than the examples shown above. You can define commands that take optional parameters:

```
\newcommand{\plusbinomial}[3][2]{(#2 + #3)^#1}
```

To save some time when writing too many expressions with exponents is by defining a new command to make simpler:

```
\[ \plusbinomial{x}{y} \]
```

And even the exponent can be changed

```
\[ \plusbinomial[4]{y}{y} \]
```

To save some time when writing too many expressions with exponents is by defining a new command to make simpler:

$$(x + y)^2$$

(/learn/File:CommsAndEnvsEx5.png)

And even the exponent can be changed

$$(y + y)^4$$

Let's analyse the syntax of the line `\newcommand{\plusbinomial}[3][2]{(#2 + #3)^#1}`:

\plusbinomial

This is the name of the new command.

[3]

The number of parameters the command will take, in this case 3.

[2]

Is the default value for the first parameter. This is what makes the first parameter optional, if not passed it will use this default value.

(#2 + #3)^#1

This is what the command does. In this case it will put the second and third parameters in a "binomial format" to the power represented by the first parameter.

➔ Open an example in ShareLaTeX (<https://www.sharelatex.com/project/new/template?zipUrl=/project/530513e2f1814bd50b0001a5/download/zip&templateName=Commands&compiler=pdfflatex>)

Overwriting existing commands

Commands

Commands (/learn/Commands)

Environments (/learn/Environments)

Field specific

Theorems and proofs

(/learn/Theorems_and_proofs)

Chemistry formulae

(/learn/Chemistry_formulae)

Feynman diagrams

(/learn/Feynman_diagrams)

Molecular orbital diagrams

(/learn/Molecular_orbital_diagrams)

Chess notation

(/learn/Chess_notation)

Knitting patterns

(/learn/Knitting_patterns)

Circuitikz package

(/learn/Circuitikz_package)

Pgfplots package

(/learn/Pgfplots_package)

Typing exams in LaTeX

(/learn/Typing_exams_in_LaTeX)

Knitr (/learn/Knitr)

Attribute Value Matrices

(/learn/Attribute_Value_Matrices)

Class files

Understanding packages and class files

(/learn/Understanding_packages_and_class_files)

List of packages and class files

(/learn/List_of_packages_and_class_files)

Writing your own package

(/learn/Writing_your_own_package)

Writing your own class

(/learn/Writing_your_own_class)

Tips (/learn/Tips)

If you define a command that has the same name as an already existing $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ command you will see an error message in the compilation of your document and the command you defined will not work. If you really want to override an existing command this can be accomplished by `renewcommand`:

```
\renewcommand{\S}{\mathbb{S}}
```

The Riemann sphere (the complex numbers plus ∞) is sometimes represented by `\(\S\)`

The riemann sphere (the complex numbers
plus ∞) is sometimes represented by \mathbb{S}

(/learn/File:CommsAndEnvsEx6.png)

In this example the command `\S` (see the example in the commands section) is overwritten to print a blackboard bold S .

`renewcommand` uses the same syntax as `newcommand`.

➔ Open an example in ShareLaTeX (<https://www.sharelatex.com/project/new/template?zipUrl=/project/530513e2f1814bd50b0001a5/download/zip&templateName=Commands&compiler=pdfflat>)

Further reading

For more information see:

- Environments (/learn/Environments)
- Understanding packages and class files (/learn/Understanding_packages_and_class_files)
- Writing your own package (/learn/Writing_your_own_package)
- Writing your own class (/learn/Writing_your_own_class)
- Lengths in LaTeX (/learn/Lengths_in_LaTeX)
- Using colours in LaTeX (/learn/Using_colours_in_LaTeX)
- Page size and margins (/learn/Page_size_and_margins)
- List of packages and class files (/learn/List_of_packages_and_class_files)
- The not so short introduction to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ (<http://www.ctan.org/tex-archive/info/lshort/>)
- LaTeX/Creating Packages on WikiBooks (http://en.wikibooks.org/wiki/LaTeX/Creating_Packages)