

Documentation Home

(/learn/Main\_Page)

Learn LaTeX in 30 minutes

(/learn/Learn\_LaTeX\_in\_30\_minutes)

 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)

# Writing your own package

Sometimes the best option to use your own commands and macros in a document is to write a new package from scratch. This article explains the main structure of a new package.

## Contents

- 1 Introduction
- 2 General structure
  - 2.1 Identification
  - 2.2 Preliminary declarations
  - 2.3 Options
  - 2.4 More declarations
- 3 Handling errors
- 4 Reference guide
- 5 Further reading

## Introduction

The first thing to do before coding a new package is to determine whether you really need a new package or not. It's recommended to search on CTAN (Comprehensive T<sub>E</sub>X Archive Network) (<http://www.ctan.org/ctan-portal/search/>) and see if someone already created something similar to what you need.

Another important thing to have in mind is the difference between packages and classes (/learn/Understanding\_packages\_and\_class\_files). Making the wrong choice can affect the flexibility of the final product.

## General structure

The structure of all package files can be roughly described in the next four parts:

- **Identification.** The file declares itself as a package written with the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> syntax.
- **Preliminary declarations.** Here the external packages needed are imported. Also, in this part of the file the commands and definitions needed by the declared options are coded.
- **Options.** The package declares and processes the options.
- **More declarations.** The main body of the package. Almost everything a package does is defined here.

In the next subsections a more detailed description of the structure and a working example, *examplepackage.sty*, will be presented.

## Identification

There are two simple commands that all packages must have:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{examplepackage}[2014/08/24 Example LaTeX package]
```

The command `\NeedsTeXFormat{LaTeX2e}` sets the L<sup>A</sup>T<sub>E</sub>X version for the package to work. Additionally, a date can be added within brackets to specify the minimal release date required.

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 `\ProvidesPackage{examplepackage}` identifies this package as *examplepackage* and, inside the brackets, the release date and some additional information is included. The date should be in the form YYYY/MM/DD

➔ Open an example of how to write a package in ShareLaTeX  
([https://www.sharelatex.com/project/new/template?](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)  
[zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf](https://www.sharelatex.com/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf))

## Preliminary declarations

Most of the packages extend and customize existing ones, and also need some external packages to work. Below, some more code is added to the sample package "examplepackage.sty".

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{examplepackage}[2014/08/21 Example package]

\RequirePackage{imakeidx}
\RequirePackage{xstring}
\RequirePackage{xcolor}
\definecolor{greycolor}{HTML}{525252}
\definecolor{sharelatexcolor}{HTML}{882B21}
\definecolor{mybluecolor}{HTML}{394773}
\newcommand{\wordcolour}{greycolor}
```

The commands in this part either initialize some parameters that latter will be used to manage the options, or import external files.

The command `\RequirePackage` is very similar to the well-known `\usepackage`, adding optional parameters within brackets will also work. The only difference is that the `\usepackage` can not be used before `\documentclass` command. It's strongly recommended to use `\RequirePackage` when writing new packages or classes.

➔ Open an example of how to write a package in ShareLaTeX  
([https://www.sharelatex.com/project/new/template?](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)  
[zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf](https://www.sharelatex.com/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf))

## Options

To allow some flexibility in the packages a few additional options are very useful. The next part in the file "examplepackage.sty" handles the parameters passed to the package-importing statement.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{examplepackage}[2014/08/21 Example package]

\RequirePackage{imakeidx}
\RequirePackage{xstring}
\RequirePackage{xcolor}
\definecolor{greycolor}{HTML}{525252}
\definecolor{sharelatexcolor}{HTML}{882B21}
\definecolor{mybluecolor}{HTML}{394773}
\newcommand{\wordcolour}{greycolor}

\DeclareOption{red}{\renewcommand{\wordcolour}{sharelatexcolor}}
\DeclareOption{blue}{\renewcommand{\wordcolour}{mybluecolor}}
\DeclareOption*{\PackageWarning{examplepackage}{Unknown '\CurrentOption'}}
\ProcessOptions\relax
```

Below a description of the main commands that can handle the options passed to the package.

The command `\DeclareOption{}` handles a given option. It takes two parameters, the first one is the name of the option and the second one is the code to execute if the option is passed.

The command `\OptionNotUsed` will print a message in the compiler and the logs, the option won't be used.

The command `\Declareoption*{}` handles every option not explicitly defined. It takes only one parameter, the code to execute when an unknown option is passed. In this case it will print a warning by means of the next command:

## 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<sup>A</sup>T<sub>E</sub>X

(/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<sub>3</sub>L<sup>A</sup>T<sub>E</sub>X (/learn/XeLaTeX)

## Presentations

Beamer (/learn/Beamer)

Powerdot (/learn/Powerdot)

Posters (/learn/Posters)

`\PackageWarning{}{}`. See handling errors for a description about what this command does.

`\CurrentOption` stores the name of the package option being handled at a determined moment.

The command `\ProcessOptions\relax` executes the code fore each option and must be inserted after all the option-handling commands were typed. There's a starred version of this command that will execute the options in the exact order specified by the calling commands.

In the example, if the options *red* or *blue* are passed to the `\usepackage` command within the document, the command `\wordcolor` is redefined. Both colours and the default grey colour were defined in the preliminary declarations after importing the *xcolor* package.

➡ Open an example of how to write a package in ShareLaTeX

([https://www.sharelatex.com/project/new/template?](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)

`zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf`)

## More declarations

In this part most of the commands will appear. In "examplepackage.sty". Below you can see the full package file.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{examplepackage}[2014/08/21 Example package]

\RequirePackage{imakeidx}
\RequirePackage{xstring}
\RequirePackage{xcolor}
\definecolor{greycolour}{HTML}{525252}
\definecolor{sharelatexcolour}{HTML}{882B21}
\definecolor{mybluecolour}{HTML}{394773}
\newcommand{\wordcolour}{greycolour}

\DeclareOption{red}{\renewcommand{\wordcolour}{sharelatexcolour}}
\DeclareOption{blue}{\renewcommand{\wordcolour}{mybluecolour}}
\DeclareOption*{\PackageWarning{examplepackage}{Unknown '\CurrentOption'}}
\ProcessOptions\relax

%%Numbered environment
\newcounter{example}[section]
\newenvironment{example}[1][\refstepcounter{example}\par\medskip
\noindent \textbf{My~environment~\theexample. #1} \rmfamily}{\medskip}

%%Important words are added to the index and printed in different colour
\newcommand{\important}[1]
{\IfSubStr{#1}{!}
{\textcolor{\wordcolour}{\textbf{\StrBefore{#1}{!}~\StrBehind{#1}
{!}}}\index{#1}}
{\textcolor{\wordcolour}{\textbf{#1}}\index{#1}\kern-1pt}
}
```

This package defines the new environment `example`, and a new command `\important`, that prints the words in a special colour and adds them to the index.

To fully understand each command see the reference guide and the links in the further reading section.

Below, a document that uses the package *examplepackage.sty*.

## 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)

```
\documentclass{article}
\usepackage[utf8]{inputenc}

\usepackage[red]{examplepackage}

\makeindex

\title{Package Example}
\author{Team Learn ShareLaTeX}
\date{ }

\begin{document}

\maketitle

\section{Introduction}
In this document a new package is tested. This package allows special
numbered
environments

\begin{example}
This text is inside a special environment, some boldface text is printed
at the beginning and a new indentation is set.
\end{example}

Also, there's a special command for \important{important!words} that will
be
printed in a special \important{colour} depending on the parameter used in
the
\important{package} importation statement. Because it's
\important{important}.

\printindex

\end{document}
```

## Package Example

Team Learn ShareLaTeX

### 1 Introduction

In this document a new package is tested. This package allows special numbered environments

**My environment 1.** This text is inside a special environment, some boldface text is printed at the beginning and a new indentation is set.

Also, there's a special command for **important words** that will be printed in a special **colour** depending on the parameter used in the **package** importation statement. Because it's **important**.

(/learn/File:WritingPackagesEx1.png)

Notice the command

```
\usepackage[red]{examplepackage}
```

➔ Open an example of how to write a package in ShareLaTeX

([https://www.sharelatex.com/project/new/template?](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)

[zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)

## Handling errors

When it comes to develop new packages it's important to handle possible errors to let know the user that something went wrong. There are four main commands to report errors in the compiler.

- `\PackageError{package-name}{error-text}{help-text}`. Takes three parameters, each one inside braces: the package name, the error text which is going to be displayed (the

compilation process will be paused), and the help text that will be printed if the user press "h" when the compilation pauses because of the error.

- `\PackageWarning{package-name}{warning-text}`. In this case the text is displayed but the compilation process won't stop. It will show the line number where the warning occurred.
- `\PackageWarningNoLine{package-name}{warning-text}`. Works just like the previous command, but it won't show the line where the warning occurred.
- `\PackageInfo{package-name}{info-text}`. In this case the information in the second parameter will only be printed in the transcript file, including the line number.

➡ Open an example of how to write a package ShareLaTeX

([https://www.sharelatex.com/project/new/template?](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)

[zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf](https://www.sharelatex.com/project/new/template?zipUrl=/project/53f11ec1eceb82a67658cb02/download/zip&templateName=PackageExample&compiler=pdf)

## Reference guide

### List of commands commonly used in packages and classes

- `\newcommand{name}{definition}`. Defines a new command ([/learn/Commands#Defining\\_a\\_new\\_command](/learn/Commands#Defining_a_new_command)), the first parameter is the name of the new command, the second parameter is what the command will do.
- `\renewcommand{}{}`. The same as `\newcommand` but will overwrite an existing command.
- `\providecommand{}{}`. Works just as `\newcommand` but if the command is already defined this one will be silently ignored.
- `\CheckCommand{}{}`. The syntax is the same as `\newcommand`, but instead it will check whether the command exists and has the expected definition, L<sup>A</sup>T<sub>E</sub>X will show a warning if the command is now what `\CheckCommand` expected.
- `\setlength{}{}`. Sets the length of the element passed as first parameter to the value written as second parameter.
- `\mbox{}`. Creates a box that contains the elements written inside the braces.
- `\fbox{}`. The same as `\mbox`, but a box is actually printed around the contents.

## Further reading

For more information see

- Understanding packages and class files ([/learn/Understanding\\_packages\\_and\\_class\\_files](/learn/Understanding_packages_and_class_files))
- Writing your own class ([/learn/Writing\\_your\\_own\\_class](/learn/Writing_your_own_class))
- Commands (</learn/Commands>) and Environments (</learn/Environments>)
- Lengths in LaTeX ([/learn/Lengths\\_in\\_LaTeX](/learn/Lengths_in_LaTeX))
- Using colours in LaTeX ([/learn/Using\\_colours\\_in\\_LaTeX](/learn/Using_colours_in_LaTeX))
- Management in a large project ([/learn/Management\\_in\\_a\\_large\\_project](/learn/Management_in_a_large_project))
- L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for package and package writers (<http://www.latex-project.org/guides/clsguide.pdf>)
- Notes on programming in tex (<http://pgfplots.sourceforge.net/TeX-programming-notes.pdf>)
- Minutes in less than hours: Using L<sup>A</sup>T<sub>E</sub>X Resources (<http://tutex.tug.org/pracjourn/2005-4/hefferon/hefferon.pdf>)
- The L<sup>A</sup>T<sub>E</sub>X Companion. Second edition (<http://ptgmedia.pearsoncmg.com/images/9780201362992/samplepages/0201362996.pdf>)