

🔍 Search help library....

# Inserting Images

---

Images are essential elements in most of the scientific documents.  $\text{\LaTeX}$  provides several options to handle images and make them look exactly what you need. In this article is explained how to include images in the most common formats, how to shrink, enlarge and rotate them, and how to reference them within your document.

## Contents

- 1 Introduction
  - 2 The folder path to images
  - 3 Changing the image size and rotating the picture
  - 4 Positioning
  - 5 Captioning, labelling and referencing
    - 5.1 Captions
    - 5.2 Labels and cross-references
  - 6 Generating high-res and low-res images
  - 7 Reference guide
  - 8 Further reading
- 

## Introduction

Below is a example on how to import a picture.

```

\documentclass{article}
\usepackage{graphicx}
\graphicspath{ {./images/} }

\begin{document}
The universe is immense and it seems to be homogeneous,
in a large scale, everywhere we look at.

\includegraphics{universe}

There's a picture of a galaxy above
\end{document}

```

The universe is immense and it seems to be homogeneous, in a large scale,  
everywhere we look at.



There's a picture of a galaxy above

(/learn/File:InsertingImagesEx1.png)

Latex can not manage images by itself, so we need to use the **graphicx** package. To use it, we include the following line in the preamble: `\usepackage{graphicx}`

The command `\graphicspath{ {./images/} }` tells L<sup>A</sup>T<sub>E</sub>X that the images are kept in a folder named *images* under the directory of the main document.

The `\includegraphics{universe}` command is the one that actually included the image in the document. Here *universe* is the name of the file containing the image without the extension, then *universe.PNG* becomes *universe*. The file name of the image should not contain white spaces nor multiple dots.

*Note: The file extension is allowed to be included, but it's a good idea to omit it. If the file extension is omitted it will prompt LaTeX to search for all the supported formats. For more details see the section about generating high resolution and low resolution images.*

➡ Open an example in Overleaf (<https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex>)

## The folder path to images

When working on a document which includes several images it's possible to keep those images in one or more separated folders so that your project is more organised.

The command `\graphicspath{ {images/} }` tells L<sup>A</sup>T<sub>E</sub>X to look in the *images* folder. The path is **relative** to the current working directory - so, the compiler will look for the file in the same folder as the code where the image is included. The path to the folder is relative by default, if there is no initial directory specified, for instance

```
%Path relative to the .tex file containing the \includegraphics command
\graphicspath{ {images/} }
```

This is a typically straightforward way to reach the graphics folder within a file tree, but can leads to complications when .tex files within folders are included in the main .tex file. Then, the compiler may end up looking for the images folder in the wrong place. Thus, **it is best practice to specify the graphics path to be relative to the main .tex file**, denoting the main .tex file directory as `./` , for instance

```
%Path relative to the main .tex file
\graphicspath{ {./images/} }
```

as in the introduction.

The path can also be **absolute**, if the exact location of the file on your system is specified. For example:

```
%Path in Windows format:
\graphicspath{ {c:/user/images/} }

%Path in Unix-like (Linux, Mac OS) format
\graphicspath{ {/home/user/images/} }
```

Notice that this command requires a trailing slash `/` and that the path is in between double braces.

You can also set multiple paths if the images are saved in more than one folder. For instance, if there are two folders named *images1* and *images2*, use the command.

```
\graphicspath{ {./images1/}{./images2/} }
```

If no path is set L<sup>A</sup>T<sub>E</sub>X will look for pictures in the folder where the .tex file the image is included in is saved.

➡ Open an example in Overleaf (<https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex>)

# Changing the image size and rotating the picture

If we want to further specify how  $\text{\LaTeX}$  should include our image in the document (length, height, etc), we can pass those settings in the following format:

```
\begin{document}
```

Overleaf is a great professional tool to edit online, share and backup your  $\text{\LaTeX}$  projects. Also offers a rather large help documentation.

```
\includegraphics[scale=1.5]{lion-logo}
```

Overleaf is a great professional tool to edit online, share and backup your  $\text{\LaTeX}$  projects. Also offers a rather large help documentation.



(/learn/File:InsertingImagesEx2.png)

The command `\includegraphics[scale=1.5]{lion-logo}` will include the image *lion-logo* in the document, the extra parameter `scale=1.5` will do exactly that, scale the image 1.5 of its real size.

You can also scale the image to a some specific width and height.

```
\begin{document}
```

Overleaf is a great professional tool to edit online, share and backup your  $\text{\LaTeX}$  projects. Also offers a rather large help documentation.

```
\includegraphics[width=3cm, height=4cm]{lion-logo}
```

Overleaf is a great professional tool to edit online, share and backup your L<sup>A</sup>T<sub>E</sub>X projects. Also offers a rather large help documentation.



(/learn/File:InsertingImagesEx3.png)

As you probably have guessed, the parameters inside the brackets [`width=3cm`, `height=4cm`] define the width and the height of the picture. You can use different units for these parameters. If only the *width* parameter is passed, the height will be scaled to keep the aspect ratio.

The length units can also be relative to some elements in document. If you want, for instance, make a picture the same width as the text:

```
\begin{document}
```

The universe is immense and it seems to be homogeneous,  
in a large scale, everywhere we look at.

```
\includegraphics[width=\textwidth]{universe}
```

The universe is immense and it seems to be homogeneous, in a large scale, everywhere we look at.



(/learn/File:InsertingImagesEx5.png)

Instead of `\textwidth` you can use any other default  $\text{\LaTeX}$  length: `\columnsep`, `\linewidth`, `\textheight`, `\paperheight`, etc. See the reference guide for a further description of these units.

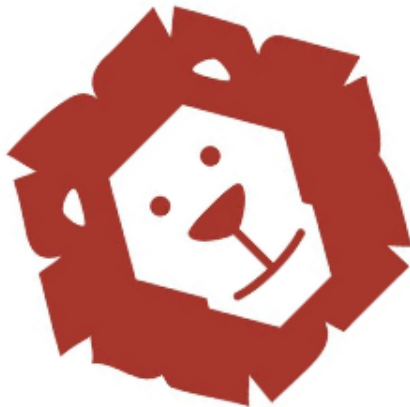
There is another common option when including a picture within your document, to **rotate** it. This can easily accomplished in  $\text{\LaTeX}$ :

```
\begin{document}
```

Overleaf is a great professional tool to edit online, share and backup your `\LaTeX{}` projects. Also offers a rather large help documentation.

```
\includegraphics[scale=1.2, angle=45]{lion-logo}
```

Overleaf is a great professional tool to edit online, share and backup your L<sup>A</sup>T<sub>E</sub>X projects. Also offers a rather large help documentation.



(/learn/File:InsertingImagesEx4.png)

The parameter `angle=45` rotates the picture 45 degrees counter-clockwise. To rotate the picture clockwise use a negative number.

↪ Open an example in Overleaf (<https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex>)

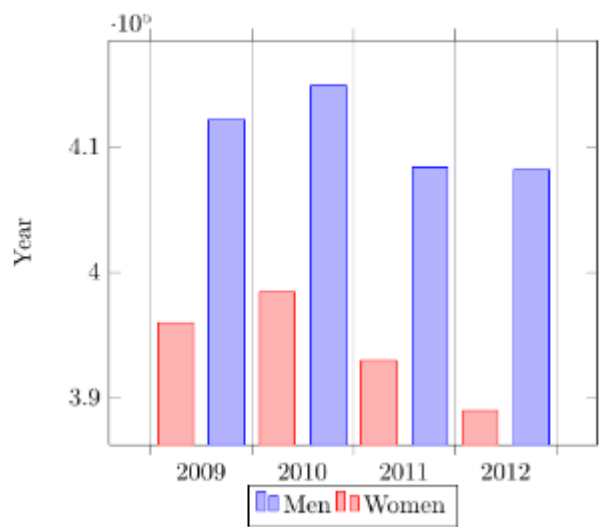
## Positioning

In the previous section was explained how to include images in your document, but the combination of text and images may not look as we expected. To change this we need to introduce a new *environment*.

In the next example the figure will be positioned right below this sentence.

```
\begin{figure}[h]
\includegraphics[width=8cm]{Plot}
\end{figure}
```

In the next example the figure will be positioned right below this sentence.



(/learn/File:InsertingImagesEx6.png)

The `figure` environment is used to display pictures as floating elements within the document. This means you include the picture inside the `figure` environment and you don't have to worry about it's placement,  $\text{\LaTeX}$  will position it in a such way that it fits the flow of the document.

Anyway, sometimes we need to have more control on the way the figures are displayed. An additional parameter can be passed to determine the figure positioning. In the example, `begin{figure}[h]`, the parameter inside the brackets set the position of the figure to **here**. Below a table to list the possible positioning values.

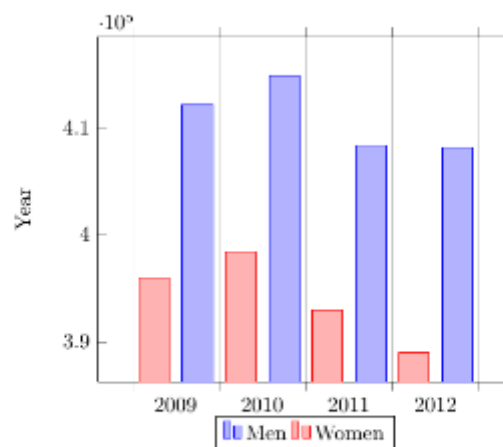
Parameter	Position
h	Place the float <i>here</i> , i.e., <i>approximately</i> at the same point it occurs in the source text (however, not <i>exactly</i> at the spot)
t	Position at the <i>top</i> of the page.
b	Position at the <i>bottom</i> of the page.
p	Put on a special <i>page</i> for floats only.
!	Override internal parameters $\text{\LaTeX}$ uses for determining "good" float positions.
H	Places the float at precisely the location in the $\text{\LaTeX}$ code. Requires the <code>float</code> package, though may cause problems occasionally. This is somewhat equivalent to <code>h!</code> .



In the next example you can see a picture at the top of the document, despite being declared below the text.

In this picture you can see a bar graph that shows the results of a survey which involved some important data studied as time passed.

```
\begin{figure}[t]
\includegraphics[width=8cm]{Plot}
\centering
\end{figure}
```



In this picture you can see a bar graph that shows the results of a survey which involved some tricky data studied as time passed.

(/learn/File:InsertingImagesEx7.png)

The additional command `\centering` will centre the picture. The default alignment is *left*.

It's also possible to **wrap** the text around a figure. When the document contains small pictures this makes it look better.

```

\begin{wrapfigure}{r}{0.25\textwidth} %this figure will be at
the right
\centering
\includegraphics[width=0.25\textwidth]{mesh}
\end{wrapfigure}

```

There are several ways to plot a function of two variables, depending on the information you are interested in. For instance, if you want to see the mesh of a function so it is easier to see the derivative you can use a plot like the one on the left.

```

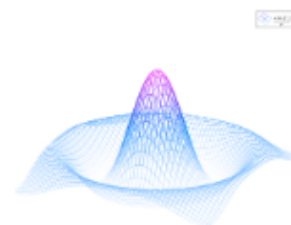
\begin{wrapfigure}{l}{0.25\textwidth}
\centering
\includegraphics[width=0.25\textwidth]{contour}
\end{wrapfigure}

```

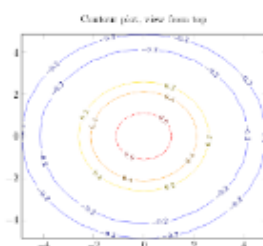
On the other side, if you are only interested on certain values you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, like the one on the left.

On the other side, if you are only interested on certain values you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, like the one on the left.

There are several ways to plot a function of two variables, depending on the information you are interested in. For instance, if you want to see the mesh of a function so it easier to see the derivative you can use a plot like the one on the left.



On the other side, if you are only interested on certain values you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, like the one on the left.



On the other side, if you are only interested on certain values you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, you can use the contour plot, like the one on the left.

(/learn/File:InsertingImagesEx8.png)

For the commands in the example to work, you have to import the package **wrapfig**. Add to the preamble the line `\usepackage{wrapfig}`.

Now you can define the *wrapfigure* environment by means of the commands

`\begin{wrapfigure}{l}{0.25\textwidth}` `\end{wrapfigure}`. Notice that the environment has two additional parameters enclosed in braces. Below the code is explained with more detail:

**`{l}`**

This defines the alignment of the figure. Set **l** for left and **r** for right. Furthermore, if you are using a book or any similar format, use instead **o** for the outer edge and **i** for the inner edge of the page.

**`{0.25\textwidth}`**

This is the width of figure box. It's not the width of the image itself, that must be set in the *includegraphics* command. Notice that the length is relative to the text width, but normal units can also be used (cm, in, mm, etc). See the reference guide for a list of units.

**`\centering`**

This was already explained, but in this example the image will be centred by using its container as reference, instead of the whole text.

For a more complete article about image positioning see Positioning images and tables (/learn/Positioning\_images\_and\_tables)

➡ Open an example in Overleaf (<https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex>)

## Captioning, labelling and referencing

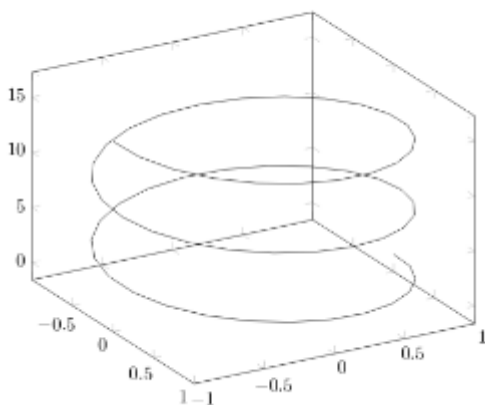
Captioning images to add a brief description and labelling them for further reference are two important tools when working on a lengthy text.

### Captions

Let's start with a caption example:

```
\begin{figure}[h]
\caption{Example of a parametric plot ( $\sin(x)$ ,  $\cos(x)$ ,  $x$ )}
\centering
\includegraphics[width=0.5\textwidth]{spiral}
\end{figure}
```

Figure 1: Example of a parametric plot  $(\sin(x), \cos(x), x)$



(/learn/File:InsertingImagesEx9.png)

It's really easy, just add the `\caption{Some caption}` and inside the braces write the text to be shown. The placement of the caption depends on where you place the command; if it's above the *includegraphics* then the caption will be on top of it, if it's below then the caption will also be set below the figure.

Captions can also be placed right after the figures. The **sidecap** package uses similar code to the one in the previous example to accomplish this.

```

\documentclass{article}
\usepackage[rightcaption]{sidecap}

\usepackage{graphicx} %package to manage images
\graphicspath{ {images/} }

\begin{SCfigure}[0.5][h]
\caption{Using again the picture of the universe.
This caption will be on the right}
\includegraphics[width=0.6\textwidth]{universe}
\end{SCfigure}

```



Figure 2: Using again the picture of the universe. This caption will be on the right

(/learn/File:InsertingImagesEx10.png)

There are two new commands

**`\usepackage[rightcaption]{sidecap}`**

As you may expect this line will import a package named **sidecap**, but there is an additional parameter: `rightcaption`. This parameter establishes the placement of the caption at the right of the picture, you can also use *leftcaption*. In book-like documents *outercaption* and *innercaption* are also available. The names of these are self-descriptive.

**`\begin{SCfigure}[0.5][h] \end{SCfigure}`**

Defines an environment similar to *figure*. The first parameter is the width of the caption relative to the size of the image, as declared in *includegraphics*. The second parameter `h` works exactly as in the *figure* environment. See the placement section for more information.

You can do a more advanced management of the caption formatting. Check the further reading section for references.

## Labels and cross-references

Figures, just as many other elements in a  $\text{\LaTeX}$  document (equations, tables, plots, etc) can be referenced within the text. This is very easy, just add a **label** to the *figure* or *SCfigure* environment, then later use that label to refer the picture.

```

\begin{figure}[h]
  \centering
  \includegraphics[width=0.25\textwidth]{mesh}
  \caption{a nice plot}
  \label{fig:mesh1}
\end{figure}

```

As you can see in the figure `\ref{fig:mesh1}`, the function grows near 0. Also, in the page `\pageref{fig:mesh1}` is the same example.

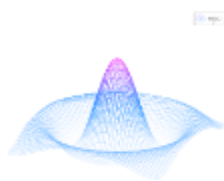


Figure 3: a nice plot

As you can see in the figure 3, the function grows near 0. Also, in the page 7 is the same example.

(/learn/File:InsertingImagesEx11.png)

There are three commands that generate cross-references in this example.

**`\label{fig:mesh1}`**

This will set a label for this figure. Since labels can be used in several types of elements within the document, it's a good practice to use a prefix, such as *fig*: in the example.

**`\ref{fig:mesh1}`**

This command will insert the number assigned to the figure. It's automatically generated and will be updated if insert some other figure before the referenced one.

**`\pageref{fig:mesh1}`**

This prints out the page number where the referenced image appears.

The `\caption` is mandatory to reference a figure.

Another great characteristic in a  $\text{\LaTeX}$  document is the ability to automatically generate a *list of figures*. This is straightforward.

`\listoffigures`

List of Figures

1	Example of a parametric plot $(\sin(x), \cos(x), x)$ . . . . .	6
2	Using again the picture of the universe. This caption will be on the right . . . . .	6
3	a nice plot . . . . .	7
4	a nice contour plot . . . . .	7

(/learn/File:InsertingImagesEx12.png)

This command only works on captioned figures, since it uses the caption in the table. The example above lists the images in this article.

Important Note: *When using cross-references your  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  project must be compiled twice, otherwise the references, the page references and the table of figures won't work.*

➡ Open an example in Overleaf (<https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex>)

# Generating high-res and low-res images

So far while specifying the image file name in the `\includegraphics` command we have omitted file extensions. However, that is not necessary, though it is often useful. If the file extension is omitted, LaTeX will search for any supported image format in that directory, and will search for various extensions in the default order (which can be modified).

This is useful in switching between development and production environments. In a development environment (when the article/report/book is still in progress), it is desirable to use low-resolution versions of images (typically in .png format) for fast compilation of the preview. In the production environment (when the final version of the article/report/book is produced), it is desirable to include the high-resolution version of the images.

This is accomplished by

- Not specifying the file extension in the `\includegraphics` command, and
- Specifying the desired extension in the preamble.

Thus, if we have two versions of an image, `venndiagram.pdf` (high-resolution) and `venndiagram.png` (low-resolution), then we can include the following line in the preamble to use the `.png` version while developing the report -

```
\DeclareGraphicsExtensions{.png,.pdf}
```

The command above will ensure that if two files are encountered with the same base name but different extensions (for example `venndiagram.pdf` and `venndiagram.png`), then the `.png` version will be used first, and in its absence the `.pdf` version will be used, this is also a good idea if some low-resolution versions are not available.

Once the report has been developed, to use the high-resolution `.pdf` version, we can change the line in the preamble specifying the extension search order to

```
\DeclareGraphicsExtensions{.pdf,.png}
```

Improving on the technique described in the previous paragraphs, we can also instruct  $\text{\LaTeX}$  to generate low-resolution `.png` versions of images on the fly while compiling the document if there is a PDF that has not been converted to PNG yet. To achieve that, we can include the following in the preamble after `\usepackage{graphicx}`

```
\usepackage{epstopdf}
\epstopdfDeclareGraphicsRule{.pdf}{png}{.png}{convert #1
\OutputFile}
\DeclareGraphicsExtensions{.png,.pdf}
```

If `venndiagram2.pdf` exists but not `venndiagram2.png`, the file `venndiagram2-pdf-converted-to.png` will be created and loaded in its place. The command `convert #1` is responsible for the conversion and additional parameters may be passed between `convert` and `#1`. For example - `convert -density 100 #1`.

There are some important things to have in mind though:

- For the automatic conversion to work, we need to call `pdflatex` with the `--shell-escape` option.
- For the final *production* version, we must comment out the `\epstopdfDeclareGraphicsRule`, so that only high-resolution PDF files are loaded. We'll also need to change the order of precedence.



➡ Open an example in Overleaf (<https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex>)

# Reference guide

## LaTeX units and legths

Abbreviation	Definition
pt	A point, is the default length unit. About 0.3515mm
mm	a millimetre
cm	a centimetre
in	an inch
ex	the height of an x in the current font
em	the width of an m in the current font
<code>\columnsep</code>	distance between columns
<code>\columnwidth</code>	width of the column
<code>\linewidth</code>	width of the line in the current environment
<code>\paperwidth</code>	width of the page
<code>\paperheight</code>	height of the page
<code>\textwidth</code>	width of the text
<code>\textheight</code>	height of the text
<code>\unitlegt</code>	units of length in the <i>picture</i> environment.

## About image types in LaTeX

### latex

When compiling with *latex*, we can only use EPS images, which is a vector format.

### pdflatex

If we are compiling using "pdflatex" to produce a PDF, then we can use a number of image formats -

JPG: Best choice if we want to insert photos

PNG: Best choice if we want to insert diagrams (if a vector version is available)

PDF: Even though we are used to seeing PDF documents, a PDF can be included directly

EPS: EPS images can be included using the *epstopdf* package (we don't need to use `\usepackage{}` to include it in our document)

### Vector format or Bit-map format?

Images can be of either vector format or bit-map format. Generally we don't need to worry about it, but if we do happen to know the format the image is in, we can use that information to choose an appropriate image format to include in our latex document. If we have an image in vector format, we should go for PDF or EPS. If we have it in bit-map format, we should go for JPG or PNG, as storing bit-map pictures in PDF or EPS takes a lot of disk space.

➡ Open an example of the pgfplots package in Overleaf

([https://www.sharelatex.com/project/new/template?](https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex)

[zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex](https://www.sharelatex.com/project/new/template?zipUrl=/project/52f4102815577d644c0017fb/download/zip&templateName=Inserting%20Images&compiler=pdflatex))

## Further reading

For more information see

- Positioning images and tables ([/learn/Positioning\\_images\\_and\\_tables](/learn/Positioning_images_and_tables))
- Lists of tables and figures ([/learn/Lists\\_of\\_tables\\_and\\_figures](/learn/Lists_of_tables_and_figures))
- Single sided and double sided documents ([/learn/Single\\_sided\\_and\\_double\\_sided\\_documents](/learn/Single_sided_and_double_sided_documents))
- Lengths in LaTeX ([/learn/Lengths\\_in\\_LaTeX](/learn/Lengths_in_LaTeX))
- **floatrow** package for advanced caption managing of floating elements (tables and figures) (<http://tug.ctan.org/tex-archive/macros/latex/contrib/floatrow/floatrow.pdf>)
- **sidecap** package documentation, for side-figure captions (<ftp://ctan.tug.org/tex-archive/macros/latex/contrib/sidecap/sidecap.pdf>)
- **epstopdf** package documentation (PDF to EPS conversion) (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&ved=0CEEQFjAC&url=http%3A%2F%2Fmirror.hmc.edu%2Fctan%2Fmacros%2Flatex%2Fcontrib%2Foberdiek%2Fepstopdf.pdf&ei=N4v2UpK7D4eHqQH2IDACg&usq=AFQjCNG9CxlXjrbc22kl-8QweEXeEauFew&sig2=qamKo3qpid8CEqzsHhplrQ&bvm=bv.60983673,d.aWM>)

Documentation Home ([/learn/Main\\_Page](/learn/Main_Page))

Learn LaTeX in 30 minutes ([/learn/Learn\\_LaTeX\\_in\\_30\\_minutes](/learn/Learn_LaTeX_in_30_minutes))

## Overleaf guides

[Creating a document in Overleaf \(/learn/Kb/Creating\\_a\\_document\\_in\\_Overleaf\)](/learn/Kb/Creating_a_document_in_Overleaf)

[Uploading a project \(/learn/Kb/Uploading\\_a\\_project\)](/learn/Kb/Uploading_a_project)

[Copying a project \(/learn/Kb/Copying\\_a\\_project\)](/learn/Kb/Copying_a_project)

[Creating a project from a template \(/learn/Kb/Creating\\_a\\_project\\_from\\_a\\_template\)](/learn/Kb/Creating_a_project_from_a_template)

[Including images in Overleaf \(/learn/Kb/Including\\_images\\_in\\_ShareLaTeX\)](/learn/Kb/Including_images_in_ShareLaTeX)

[Exporting your work from Overleaf \(/learn/Kb/Exporting\\_your\\_work\\_from\\_ShareLaTeX\)](/learn/Kb/Exporting_your_work_from_ShareLaTeX)

[Working offline in Overleaf \(/learn/Kb/Working\\_Offline\\_in\\_Overleaf\\_v2\)](/learn/Kb/Working_Offline_in_Overleaf_v2)

[Using Track Changes in Overleaf \(/learn/Kb/Track\\_Changes\\_in\\_Overleaf\\_v2\)](/learn/Kb/Track_Changes_in_Overleaf_v2)

[Using bibliographies in Overleaf \(/learn/Kb/Using\\_bibliographies\\_in\\_ShareLaTeX\)](/learn/Kb/Using_bibliographies_in_ShareLaTeX)

[Sharing your work with others \(/learn/Kb/Sharing\\_your\\_work\\_with\\_others\)](/learn/Kb/Sharing_your_work_with_others)

[Debugging Compilation timeout errors \(/learn/Kb/Debugging\\_Compilation\\_timeout\\_errors\)](/learn/Kb/Debugging_Compilation_timeout_errors)

[How-to guides \(/learn/Kb/Knowledge\\_Base\)](/learn/Kb/Knowledge_Base)

## LaTeX Basics

[Creating your first LaTeX document \(/learn/Creating\\_a\\_document\\_in\\_LaTeX\)](/learn/Creating_a_document_in_LaTeX)

[Choosing a LaTeX Compiler \(/learn/Choosing\\_a\\_LaTeX\\_Compiler\)](/learn/Choosing_a_LaTeX_Compiler)

[Paragraphs and new lines \(/learn/Paragraphs\\_and\\_new\\_lines\)](/learn/Paragraphs_and_new_lines)

[Bold, italics and underlining \(/learn/Bold,\\_italics\\_and\\_underlining\)](/learn/Bold,_italics_and_underlining)

[Lists \(/learn/Lists\)](/learn/Lists)

[Errors \(/learn/Errors\)](/learn/Errors)

## Mathematics

[Mathematical expressions \(/learn/Mathematical\\_expressions\)](/learn/Mathematical_expressions)

[Subscripts and superscripts \(/learn/Subscripts\\_and\\_superscripts\)](/learn/Subscripts_and_superscripts)

[Brackets and Parentheses \(/learn/Brackets\\_and\\_Parentheses\)](/learn/Brackets_and_Parentheses)

[Fractions and Binomials \(/learn/Fractions\\_and\\_Binomials\)](/learn/Fractions_and_Binomials)

[Aligning Equations \(/learn/Aligning\\_equations\)](/learn/Aligning_equations)

[Operators \(/learn/Operators\)](/learn/Operators)

[Spacing in math mode \(/learn/Spacing\\_in\\_math\\_mode\)](/learn/Spacing_in_math_mode)

[Integrals, sums and limits \(/learn/Integrals,\\_sums\\_and\\_limits\)](/learn/Integrals,_sums_and_limits)

[Display style in math mode \(/learn/Display\\_style\\_in\\_math\\_mode\)](/learn/Display_style_in_math_mode)

[List of Greek letters and math symbols \(/learn/List\\_of\\_Greek\\_letters\\_and\\_math\\_symbols\)](/learn/List_of_Greek_letters_and_math_symbols)

[Mathematical fonts \(/learn/Mathematical\\_fonts\)](/learn/Mathematical_fonts)

## Figures and tables

[Inserting Images \(/learn/Inserting\\_Images\)](/learn/Inserting_Images)

[Tables \(/learn/Tables\)](/learn/Tables)

[Positioning Images and Tables \(/learn/Positioning\\_images\\_and\\_tables\)](/learn/Positioning_images_and_tables)

[Lists of Tables and Figures \(/learn/Lists\\_of\\_tables\\_and\\_figures\)](/learn/Lists_of_tables_and_figures)

[Drawing Diagrams Directly in LaTeX \(/learn/Picture\\_environment\)](/learn/Picture_environment)

[TikZ package \(/learn/TikZ\\_package\)](/learn/TikZ_package)

## References and Citations

[Bibliography management in LaTeX \(/learn/Bibliography\\_management\\_in\\_LaTeX\)](/learn/Bibliography_management_in_LaTeX)

[Bibliography management with biblatex \(/learn/Bibliography\\_management\\_in\\_LaTeX\)](/learn/Bibliography_management_in_LaTeX)

[Biblatex bibliography styles \(/learn/Biblatex\\_bibliography\\_styles\)](/learn/Biblatex_bibliography_styles)

[Biblatex citation styles \(/learn/Biblatex\\_citation\\_styles\)](/learn/Biblatex_citation_styles)

[Bibliography management with natbib \(/learn/Bibliography\\_management\\_with\\_natbib\)](/learn/Bibliography_management_with_natbib)

[Natbib bibliography styles \(/learn/Natbib\\_bibliography\\_styles\)](/learn/Natbib_bibliography_styles)

[Natbib citation styles \(/learn/Natbib\\_citation\\_styles\)](/learn/Natbib_citation_styles)

[Bibliography management with bibtex \(/learn/Bibliography\\_management\\_with\\_bibtex\)](/learn/Bibliography_management_with_bibtex)

[Bibtex bibliography styles \(/learn/Bibtex\\_bibliography\\_styles\)](/learn/Bibtex_bibliography_styles)

## Languages

[International language support \(/learn/International\\_language\\_support\)](/learn/International_language_support)

[Quotations and quotation marks \(/learn/Typesetting\\_quotations\)](/learn/Typesetting_quotations)

[Arabic \(/learn/Arabic\)](/learn/Arabic)

[Chinese \(/learn/Chinese\)](/learn/Chinese)

[French \(/learn/French\)](/learn/French)

[German \(/learn/German\)](/learn/German)

[Greek \(/learn/Greek\)](/learn/Greek)

[Italian \(/learn/Italian\)](/learn/Italian)

[Japanese \(/learn/Japanese\)](/learn/Japanese)

[Korean \(/learn/Korean\)](/learn/Korean)

[Portuguese \(/learn/Portuguese\)](/learn/Portuguese)

[Russian \(/learn/Russian\)](/learn/Russian)

[Spanish \(/learn/Spanish\)](/learn/Spanish)

## Document structure

[Sections and chapters \(/learn/Sections\\_and\\_chapters\)](/learn/Sections_and_chapters)

[Table of contents \(/learn/Table\\_of\\_contents\)](/learn/Table_of_contents)

[Cross referencing sections and equations \(/learn/Cross\\_referencing\\_sections\\_and\\_equations\)](/learn/Cross_referencing_sections_and_equations)

[Indices \(/learn/Indices\)](/learn/Indices)

[Glossaries \(/learn/Glossaries\)](/learn/Glossaries)

[Nomenclatures \(/learn/Nomenclatures\)](/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>Y</sub>L<sup>A</sup>T<sub>E</sub>X \(/learn/XeLaTeX\)](#)

## Presentations

[Beamer \(/learn/Beamer\)](#)

[Powerdot \(/learn/Powerdot\)](#)

[Posters \(/learn/Posters\)](#)

## Commands

[Commands \(/learn/Commands\)](#)

[Environments \(/learn/Environments\)](#)

## Field specific

[Theorems and proofs \(/learn/Theorems\\_and\\_proofs\)](#)

[Chemistry formulae \(/learn/Chemistry\\_formulae\)](#)

[Feynman diagrams \(/learn/Feynman\\_diagrams\)](/learn/Feynman_diagrams)

[Molecular orbital diagrams \(/learn/Molecular\\_orbital\\_diagrams\)](/learn/Molecular_orbital_diagrams)

[Chess notation \(/learn/Chess\\_notation\)](/learn/Chess_notation)

[Knitting patterns \(/learn/Knitting\\_patterns\)](/learn/Knitting_patterns)

[Circuitikz package \(/learn/Circuitikz\\_package\)](/learn/Circuitikz_package)

[Pgfplots package \(/learn/Pgfplots\\_package\)](/learn/Pgfplots_package)

[Typing exams in LaTeX \(/learn/Typing\\_exams\\_in\\_LaTeX\)](/learn/Typing_exams_in_LaTeX)

[Knitr \(/learn/Knitr\)](/learn/Knitr)

[Attribute Value Matrices \(/learn/Attribute\\_Value\\_Matrices\)](/learn/Attribute_Value_Matrices)

## Class files

[Understanding packages and class files \(/learn/Understanding\\_packages\\_and\\_class\\_files\)](/learn/Understanding_packages_and_class_files)

[List of packages and class files \(/learn/List\\_of\\_packages\\_and\\_class\\_files\)](/learn/List_of_packages_and_class_files)

[Writing your own package \(/learn/Writing\\_your\\_own\\_package\)](/learn/Writing_your_own_package)

[Writing your own class \(/learn/Writing\\_your\\_own\\_class\)](/learn/Writing_your_own_class)

[Tips \(/learn/Tips\)](/learn/Tips)

---