

Learn LaTeX in 30 minutes

In this guide, we hope to give you your first introduction to \LaTeX . The guide does not require you to have any prior knowledge of LaTeX, but by the time you are finished, you will have written your first LaTeX document, and hopefully will have a good knowledge of some of the basic functions provided by LaTeX.

Contents

- 1 What is LaTeX?
 - 2 Why learn LaTeX?
 - 3 Writing your first piece of LaTeX
 - 4 The preamble of a document
 - 5 Adding a title, author and date
 - 6 Adding comments
 - 7 Bold, italics and underlining
 - 8 Adding images
 - 8.1 Captions, labels and references
 - 9 Creating lists in LaTeX
 - 9.1 Unordered lists
 - 9.2 Ordered lists
 - 10 Adding math to LaTeX
 - 11 Basic Formatting
 - 11.1 Abstracts
 - 11.2 Paragraphs and newlines
 - 11.3 Chapters and Sections
 - 12 Creating tables
 - 12.1 Creating a simple table in LaTeX
 - 12.2 Adding borders
 - 12.3 Captions, labels and references
 - 13 Adding a Table of Contents
 - 14 Downloading your finished document
-

What is LaTeX?

\LaTeX (pronounced *LAY-tek* or *LAH-tek*) is a tool used to create professional-looking documents. It is based on the WYSIWYM (what you see is what you mean) idea, meaning you only have focus on the contents of your document and the computer will take care of the formatting. Instead of spacing out text on a page to control formatting, as with Microsoft Word or LibreOffice Writer, users can enter plain text and let LaTeX take care of the rest.

Why learn LaTeX?

LaTeX is used all over the world for scientific documents, books, as well as many other forms of publishing. Not only can it create beautiful typesetted documents, but it allows users to very quickly tackle the more complicated parts of typesetting, such as inputting mathematics, creating tables of contents, referencing and creating bibliographies, and having a consistent layout across all sections. Due to the huge number of open source *packages* available (more on this later), the possibilities with LaTeX are endless. These *packages* allow users to do even more with LaTeX, such as add footnotes, draw schematics, create tables etc.

One of the most important reasons people use LaTeX is that it separates the content of the document from the style. This means that once you have written the content of your document, we can change its appearance with ease. Similarly, you can create one style of document which can be used to standardise the appearance of many different documents. This allows scientific journals to create templates for submissions. These templates have a pre-made layout meaning that only the content needs to be added. In fact there are hundreds of **templates** available for everything from CVs to slideshows.

Writing your first piece of LaTeX

The first step is to create a new LaTeX project. You can do this on your own computer by creating a new **.tex** file, or else you can **start a new project in ShareLaTeX**. Let's start with the simplest working example:

```
\documentclass{article}

\begin{document}
First document. This is a simple example, with no
extra parameters or packages included.
\end{document}
```

First document. This is a simple example, with no extra parameters or packages included.

You can see that LaTeX has already taken care of the first piece of formatting for you, by indenting the first line of the paragraph. Let's have a close look at what each part of our code does.

[Open an example in ShareLaTeX](#)

The first line of code declares the type of document, known as the *class*. The class controls the overall appearance of the document. Different types of documents will require different classes i.e. a CV/resume will require a different class than a scientific paper. In this case, the class is **article**, the simplest and most common LaTeX class. Other types of documents you may be working on may require different classes such as **book** or **report**.

After this, you write the content of our document, enclosed inside the `\begin{document}` and `\end{document}` tags. This is known as the *body* of the document. You can start writing here and make changes to the text if you wish. To see the result of these changes in the PDF you have to compile the document. To do this in ShareLaTeX, simply hit **Recompile**.

If you are using a basic text editor such as gedit, emacs, vim, sublime, notepad etc., you will have to compile the document manually. To do this, simply run **pdflatex <your document>** in your computers terminal/command line. See [here](#) for more information on how to do this.

If you are using a **dedicated LaTeX editor** such as TeXmaker or TeXworks, simply hit the **Recompile** button. Consult the programs documentation if you are unsure of where this is.

Now that you have learnt how to add content to our document, the next step is to give it a title. To do this, we must talk briefly about the **preamble**.

The preamble of a document

In the previous example the text was entered after the `\begin{document}` command. Everything in your `.tex` file *before* this point is called the **preamble**. In the preamble you define the type of document you are writing, the language you are writing in, the *packages* you would like to use (more on this later) and several other elements. For instance, a normal document preamble would look like this:

```
\documentclass[12pt, letterpaper]{article}
\usepackage[utf8]{inputenc}
```

Below a detailed description of each line:

```
\documentclass[12pt, letterpaper]{article}
```

As said before, this defines the type of document. Some additional parameters included in the square brackets brackets can be passed to the command. These parameters must be comma-separated. In the example, the extra parameters set the font size (**12pt**) and the paper size (**letterpaper**). Of course other font sizes (**9pt**, **11pt**, **12pt**) can be used, but if none is specified, the default size is **10pt**. As for the paper size other possible values are **a4paper** and **legalpaper**; see the article about [Page size and margins](#) for more details about this.

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

This is the encoding for the document. It can be omitted or changed to another encoding but utf-8 is recommended. Unless you specifically need another encoding, or if you are unsure about it, add this line to the preamble.

Adding a title, author and date

To add a title, author and date to our document, you must add three lines to the **preamble** (NOT the main body of the document). These lines are

```
\title{First document}
```

This is the title.

```
\author{Hubert Farnsworth}
```

Here you put the name of the Author(s) and, as a optional parameter, you can add the next command:

```
\thanks{funded by the ShareLaTeX team}
```

This can be added after the name of the autor, inside the braces of the **title** command. It will add a superscript and a footnote with the text inside the braces. Useful if you need to thank an institution in your article.

```
\date{February 2014}
```

You can enter the date manually or use the command `\today` so the date will be updated automatically at the time you compile your document

With these lines added, your preamble should look something like this

```
\documentclass[12pt, letterpaper, twoside]{article}
\usepackage[utf8]{inputenc}

\title{First document}
\author{Hubert Farnsworth \thanks{funded by the ShareLaTeX team}}
\date{February 2017}
```

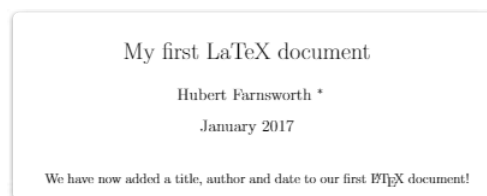
Now that you have given your document a title, author and date, you can print this information on the document with the `\maketitle` command. This should be included in the **body** of the document at the place you want the title to be printed.

```
\begin{document}

\maketitle

We have now added a title, author and date to our first \LaTeX{} document!

\end{document}
```



[Open an example in ShareLaTeX](#)

Adding comments

As with any code you are writing, it can often be useful to include comments. Comments are pieces of text you can include in the document which will not be printed, and will not affect the document in any way. They are useful for organizing your work, taking notes, or commenting out lines/sections when debugging. To make a comment in LaTeX, simply write a `%` symbol at the beginning of the line as shown below:

```
\begin{document}

\maketitle

We have now added a title, author and date to our first \LaTeX{} document!
```

% This line here is a comment. It will not be printed in the document.

\end{document}



[Open an example in ShareLaTeX](#)

Bold, italics and underlining

We will now look at some simple text formatting commands.

- **Bold**: Bold text in LaTeX is written with the `\textbf{...}` command.
- *Italics*: Italicised text in LaTeX is written with the `\textit{...}` command.
- Underline: Underlined text in LaTeX is written with the `\underline{...}` command.

An example of each of these in action is shown below:

Some of the `\textbf{greatest}`
`discoveries` in `\underline{science}`
were made by `\textbf{\textit{accident}}`.

Some of the **greatest** discoveries in science were
made by *accident*.

Another very useful command is the `\emph{...}` command. What the `\emph` command actually does with its argument depends on the context - inside normal text the emphasized text is italicized, but this behaviour is reversed if used inside an italicized text- see example below:

Some of the greatest `\emph{discoveries}`
in science
were made by accident.

`\textit{Some of the greatest \emph{discoveries}}`
in science
were made by accident.

`\textbf{Some of the greatest \emph{discoveries}}`
in science
were made by accident.

Some of the greatest *discoveries* in science were
made by accident.
*Some of the greatest discoveries in science were
made by accident.*
**Some of the greatest *discoveries* in science
were made by accident.**

Moreover, some packages, e.g. [Beamer](#), change the behaviour of `\emph` command.

[Open an example in ShareLaTeX](#)

Adding images

We will now look at how to add images to a LaTeX document. On ShareLaTeX, you will first have to [upload the images](#).

Below is a example on how to include 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

[Open an example in ShareLaTeX](#)

Latex can not manage images by itself, so you will need to use a *package*. Packages can be used to change the default look of your LaTeX document, or to allow more functionalities. In this case, you need to include an image in our document, so you should use the **graphicx** package. This package gives new commands, `\includegraphics{...}` and `\graphicspath{...}`. To use the **graphicx** package, include the following line in you preamble: `\usepackage{graphicx}`

The command `\graphicspath{ {images/} }` tells LaTeX that the images are kept in a folder named *images* under the current directory.

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](#).*

Captions, labels and references

Images can be captioned, labelled and referenced by means of the **figure** environment as shown below:

```
\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.

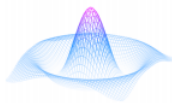


Figure 3: a nice plot

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

[Open an example in ShareLaTeX](#)

There are three important commands in the example:

- `\caption{a nice plot}`: As you may expect this command sets the caption for the figure. If you create a list of figures this caption will be used there. You can place it above or below the figure.
- `\label{fig:mesh1}`: If you need to refer the image within your document, set a label with this command. The label will number the image, and combined with the next command will allow you to reference it.
- `\ref{fig:mesh1}`: This code will be substituted by the number corresponding to the referenced figure.

When placing images in a LaTeX document, we should always put them inside a **figure** environment or similar so that LaTeX will position the image in a way that fits in with the rest of your text.

Note: *If you are using captions and references on your own computer, you will have to compile the document twice for the references to work. ShareLaTeX will do this for you automatically.*

Creating lists in LaTeX

Lists are very simple to create in LaTeX. You can create lists using different list *environments*. Environments are sections of our document that you want to present in a different way to the rest of the document. They start with a `\begin{...}` command and end with an `\end{...}` command.

There are two main different types of lists, *ordered* lists and *unordered* lists. Each will use a different environment.

Unordered lists

Unordered lists are produced by the **itemize** environment. Each entry must be preceded by the control sequence `\item` as shown below.

```
\begin{itemize}
  \item The individual entries are indicated with a black dot, a so-called bullet.
  \item The text in the entries may be of any length.
\end{itemize}
```

- The individual entries are indicated with a black dot, a so-called bullet.
- The text in the entries may be of any length.

By default the individual entries are indicated with a black dot, so-called bullet. The text in the entries may be of any length.

[Open an example in ShareLaTeX](#)

Ordered lists

Ordered list have the same syntax inside a different environment. We make ordered lists using the **enumerate** environment:

```
\begin{enumerate}
  \item This is the first entry in our list
  \item The list numbers increase with each entry we add
\end{enumerate}
```

1. This is the first entry in our list
2. The list numbers increase with each entry we add

As with unordered lists, each entry must be preceded by the control sequence `\item`, which will automatically generate the number labelling the item. The `enumerate` labels consists of sequential numbers starting at one.

[Open an example in ShareLaTeX](#)

Adding math to LaTeX

One of the main advantages of LaTeX is the ease at which mathematical expressions can be written. LaTeX allows two writing modes for mathematical expressions: the **inline** mode and the **display** mode. The first one is used to write formulas that are part of a text. The second one is used to write expressions that are not part of a text or paragraph, and are therefore put on separate lines. Let's see an example of the **inline** mode:

In physics, the mass-energy equivalence is stated
by the equation $E=mc^2$, discovered in 1905 by Albert Einstein.

In physics, the mass-energy equivalence is stated by the equation $E = mc^2$,
discovered in 1905 by Albert Einstein.

To put your equations in *inline* mode use one of these delimiters: `\(... \)`, `$... $` or `\begin{math} ... \end{math}`. They all work and the choice is a matter of taste.

The *displayed* mode has two versions: numbered and unnumbered.

The mass-energy equivalence is described by the famous equation

$E=mc^2$

discovered in 1905 by Albert Einstein.

In natural units ($c = 1$), the formula expresses the identity

```
\begin{equation}
E=m
\end{equation}
```

The mass-energy equivalence is described by the famous equation

$$E = mc^2$$

discovered in 1905 by Albert Einstein. In natural units ($c = 1$), the formula expresses the identity

$$E = m \tag{1}$$

To print your equations in *display* mode use one of these delimiters: `\[... \]`, `$$... $$`, `\begin{displaymath} ... \end{displaymath}` or `\begin{equation} ... \end{equation}`

Important Note: *equation** environment is provided by an external package, consult the [amsmath article](#).

[Open an example in ShareLaTeX](#)

Many math mode commands require the **amsmath** package, so be sure to include it when writing math. An example is shown below of some basic math mode commands.

Subscripts in math mode are written as a_b and superscripts are written as a^b . These can be combined and nested to write expressions such as

$T^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} = T(x^{i_1}, \dots, x^{i_p}, e_{j_1}, \dots, e_{j_q})$

We write integrals using \int and fractions using $\frac{a}{b}$. Limits are placed on integrals using superscripts and subscripts:

$\int_0^1 \frac{1}{e^x} = \frac{e-1}{e}$

Lower case Greek letters are written as `\omega` `\delta` etc. while upper case Greek letters are written as `\Omega` `\Delta`.

Mathematical operators are prefixed with a backslash as `\sin(\beta)`, `\cos(\alpha)`, `\log(x)` etc.

Subscripts in math mode are written as a_b and superscripts are written as a^b . These can be combined and nested to write expressions such as

$$T^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} = T(x^{i_1}, \dots, x^{i_p}, e_{j_1}, \dots, e_{j_q})$$

We write integrals using \int and fractions using $\frac{a}{b}$. Limits are placed on integrals using superscripts and subscripts:

$$\int_0^1 \frac{1}{e^x} = \frac{e-1}{e}$$

Lower case Greek letters are written as ω δ etc. while upper case Greek letters are written as Ω Δ .

Mathematical operators are prefixed with a backslash as $\sin(\beta)$, $\cos(\alpha)$, $\log(x)$ etc.

[Open an example in ShareLaTeX](#)

The possibilities with math in LaTeX are endless and it is impossible to list them all here. Be sure to check out our other articles on

- [Mathematical expressions](#)
- [Subscripts and superscripts](#)
- [Brackets and Parentheses](#)
- [Fractions and Binomials](#)
- [Aligning Equations](#)
- [Operators](#)
- [Spacing in math mode](#)
- [Integrals, sums and limits](#)
- [Display style in math mode](#)
- [List of Greek letters and math symbols](#)
- [Mathematical fonts](#)

Basic Formatting

We will now look at how to write abstracts, as well as how to format a LaTeX document into different chapters, sections and paragraphs.

Abstracts

In scientific documents it's a common practice to include a brief overview of the main subject of the paper. In LaTeX there's the **abstract** environment for this. The **abstract** environment will put the text in a special format at the top of your document.

```
\begin{document}

\begin{abstract}
This is a simple paragraph at the beginning of the
document. A brief introduction about the main subject.
\end{abstract}
\end{document}
```

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.

[Open an example in ShareLaTeX](#)

Paragraphs and newlines

```
\begin{document}

\begin{abstract}
This is a simple paragraph at the beginning of the
document. A brief introduction about the main subject.
\end{abstract}
```

Now that we have written our abstract, we can begin writing our first paragraph.

```
This line will start a second Paragraph.
\end{document}
```

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.

Now that we have written our abstract, we can begin writing our first paragraph.

This line will start a second Paragraph.

[Open an example in ShareLaTeX](#)

When writing the contents of your document, if you need to start a new paragraph you must hit the "Enter" key twice (to insert a double blank line). Notice that LaTeX automatically indents paragraphs.

To start a new line without actually starting a new paragraph insert a *break line* point, this can be done by `\\` (a double backslash as in the example) or the `\newLine` command

You can find more information in the [Paragraphs and new lines](#) article.

Chapters and Sections

Commands to organize a document vary depending on the document type, the simplest form of organization is the sectioning, available in all formats.

```
\chapter{First Chapter}

\section{Introduction}

This is the first section.

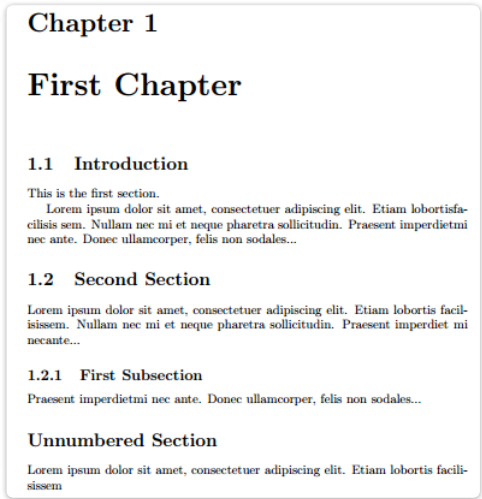
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam lobortisfacilisis sem. Nullam nec mi et
neque pharetra sollicitudin. Praesent imperdietmi nec ante.
Donec ullamcorper, felis non sodales...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilissem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi necante...

\subsection{First Subsection}
Praesent imperdietmi nec ante. Donec ullamcorper, felis non sodales...

\section*{Unnumbered Section}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilissem
```



[Open an example in ShareLaTeX](#)

The command `\section{}` marks the beginning of a new section, inside the braces is set the title. Section numbering is automatic and can be disabled by including a `*` in the section command as `\section*{}`. We can also have `\subsection{}`s, and indeed `\subsubsection{}`s. The basic levels of depth are listed below:

-1	<code>\part{part}</code>
0	<code>\chapter{chapter}</code>
1	<code>\section{section}</code>
2	<code>\subsection{subsection}</code>
3	<code>\subsubsection{subsubsection}</code>
4	<code>\paragraph{paragraph}</code>
5	<code>\subparagraph{subparagraph}</code>

Note that `\part` and `\chapter` are only available in *report* and *book* document classes.

For a more complete discussion about the document structure see the [article about sections and chapters](#).

Creating tables

Creating a simple table in LaTeX

Below you can see the simplest working example of a table

```
\begin{center}
\begin{tabular}{c c c}
cell1 & cell2 & cell3 \\
cell4 & cell5 & cell6 \\
cell7 & cell8 & cell9
\end{tabular}
\end{center}
```

```
\end{tabular}
\end{center}
```

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

The **tabular** environment is the default LaTeX method to create tables. You must specify a parameter to this environment, in this case `{c c c}`. This tells LaTeX that there will be three columns and that the text inside each one of them must be centred. You can also use `r` to align the text to the right and `l` for left alignment. The alignment symbol `&` is used to specify the breaks in the table entries. There must always be one less alignment symbol in each line than the number of columns. To go to the next line of your table, we use the *new line* command `\\`. We wrap the entire table inside the **center** environment so that it will appear in the center of the page.

[Open an example in ShareLaTeX](#)

Adding borders

The *tabular* environment is more flexible, you can put separator lines in between each column.

```
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
cell1 & cell2 & cell3 \\
cell4 & cell5 & cell6 \\
cell7 & cell8 & cell9 \\
\hline
\end{tabular}
\end{center}
```

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

You can add borders using the horizontal line command `\hline` and the vertical line parameter `|`.

- `{ |c|c|c| }`: This declares that three columns, separated by a vertical line, are going to be used in the table. The `|` symbol specifies that these columns should be separated by a vertical line.
- `\hline`: This will insert a horizontal line. We have included horizontal lines at the top and bottom of the table here. There is no restriction on the number of times you can use `\hline`.

Below you can see a second example.

```
\begin{center}
\begin{tabular}{||c c c c||}
\hline
Col1 & Col2 & Col2 & Col3 \\
\hline\hline
1 & 6 & 87837 & 787 \\
\hline
2 & 7 & 78 & 5415 \\
\hline
3 & 545 & 778 & 7507 \\
\hline
4 & 545 & 18744 & 7560 \\
\hline
5 & 88 & 788 & 6344 \\
\hline
\end{tabular}
\end{center}
```

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

[Open an example in ShareLaTeX](#)

Captions, labels and references

You can caption and reference tables in much the same way as images. The only difference is that instead of the **figure** environment, you use the **table** environment.

Table `\ref{table:data}` is an example of referenced `\LaTeX{}` elements.

```
\begin{table}[h!]
```



```
\centering
\begin{tabular}{||c c c c||}
\hline
Col1 & Col2 & Col2 & Col3 \\\ [0.5ex]
\hline\hline
1 & 6 & 87837 & 787 \\\
2 & 7 & 78 & 5415 \\\
3 & 545 & 778 & 7507 \\\
4 & 545 & 18744 & 7560 \\\
5 & 88 & 788 & 6344 \\\ [1ex]
\hline
\end{tabular}
\caption{Table to test captions and labels}
\label{table:data}
\end{table}
```

Table 1 is an example of referenced L^AT_EX elements.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1: Table to test captions and labels

[Open an example in ShareLaTeX](#)

Note: If you are using captions and references on your own computer, you will have to compile the document twice for the references to work. ShareLaTeX will do this for you automatically.'

Adding a Table of Contents

To create the table of contents is straightforward, the command `\tableofcontents` does all the work for you:

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

\title{Sections and Chapters}
\author{Gubert Farnsworth}
\date{ }

\begin{document}

\maketitle

\tableofcontents

\section{Introduction}

This is the first section.

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam lobortisfacilisis sem. Nullam nec mi et
neque pharetra sollicitudin. Praesent imperdietmi nec ante.
Donec ullamcorper, felis non sodales...

\addcontentsline{toc}{section}{Unnumbered Section}
\section*{Unnumbered Section}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisissem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi necante...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisissem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi necante...

\end{document}
```

Sections and Chapters

Gubert Farnsworth

Contents

1 Introduction	1
Unnumbered Section	1
2 Second Section	2

1 Introduction

This is the first section.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultrices vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Unnumbered Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue,

Sections, subsections and chapters are automatically included in the table of contents. To manually add entries, for example when you want an unnumbered section, use the command `\addcontentsline` as shown in the example.