# 2 Basic structures

The Quick Start exercise in section 1.4 on page 10 was enough to show how a LaTeX document works. Now we're going to start looking at how a larger document is put together. If you skipped the whole of Chapter 1 starting on page 1, be prepared to go back to some of the sections in it, because I'll be referring to things you might not have come across yet.

LaTeX's approach to formatting is based on *consistency*. This means that as long as you identify each component element of your document correctly, it will be typeset in the same way as all the other elements like it, so that you achieve a consistent finish with minimum effort.

Consistency helps make documents easier to read and understand, as well as making them more visually attractive. Consistency is also what editors, reviewers, and publishers look for. Publishers have a house style, and often a reputation to keep, so they rightly insist that if you do something a certain way once, you should do it the same way each time.

'Elements' are the component parts of a document: all the pieces which make up the whole. Almost everyone who reads books, newspapers, magazines, reports, articles, and other classes of documents will be familiar with the common elements: parts, chapters, sections, subsections, headings, titles, subtitles, paragraphs, lists, tables, figures, sidebars, panels, exercises, and so on, even if they don't consciously think about them.

## 2.1 The Document Class Declaration

In order to set things up correctly, LaTeX needs to know up front what type of document you are going to be writing. There are probably lots of different types of document you deal with: in LaTeX they are called ***classes*** of documents — 'class' is just a computing science word for 'type'.

### 2.1.1 Document classes

To tell LaTeX what class of document you are going to create, the first line of your file MUST identify it.[1] To start a report, for example, you would type a `\documentclass` command like this as the first line of your document:

```
\documentclass{report}
```

There are four built-in classes provided, and many others that you can download (some may already be installed for you):

**report** for business, technical, legal, academic, or scientific reports; and for theses[2] and dissertations;

**article** for white papers, magazine or journal articles, reviews, conference papers, essays, or research notes;

**book** for books, booklets, or whole journals;

**letter** for letters.[3]

These default classes are fairly basic in terms of layout and design, in order to make them easier to customise by adding *packages*, which are the style and layout plug-ins that LaTeX uses to let you automate formatting and change the design of your documents. Packages and classes are explained in more detail in Chapter 3 starting on page 57.

The **article** class in particular can be used for almost any short piece of typesetting by simply omitting the titling, changing the layout, and adding the

[1] Readers familiar with SGML, HTML, and XML will recognise the concept as similar to the Document Type Declaration (it's still called a 'type' there, not a 'class').

[2] Theses and dissertations require an Abstract, which is provided in the **report** class but not in the **book** class. Many universities provide a special **thesis** class of their own.

[3] The built-in **letter** class is rather idiosyncratic: there are much better ones you can use which you will find in the memoir package and the komascript bundle.

relevant packages — like we saw in the Quick Start document in section 1.4 on page 10.

The **letter** class is not much used: it provides a very old-fashioned layout. There are other more up-to-date classes for letters available for download.

### 2.1.2 Extending the default classes

The built-in classes are intended as starting-points, especially for drafts, and for compatibility when exchanging documents with other LaTeX users. They come built into every installation of LaTeX and if left unmodified, are guaranteed to format identically everywhere. They are *not* intended as final-format publication-quality layouts, and should not be used as such. For most other purposes, especially for publication, you use LaTeX packages to extend these classes to do what you need. Some common ways to do this are:

- ☐ The memoir package and the komascript bundle contain more sophisticated replacements for all the built-in classes, as well as additional ones;
- ☐ Many academic and scientific publishers provide their own special class files for articles and books (some come with LaTeX, others are on the publishers' web sites for download);
- ☐ Conference organisers may also provide class files for authors to write papers for submission, presentation, preprints, and proceedings;
- ☐ Many universities provide their own thesis document classes in order to ensure exact fulfilment of their formatting requirements (many of these are on CTAN);
- ☐ Businesses and other organisations can provide their users with private corporate classes on a central server and configure LaTeX installations to look there first for packages, fonts, etc (not usually available to the public, of course);
- ☐ There are nearly 300 document classes on CTAN (see `www.ctan.org/topic/class`).

The four default built-in document classes are therefore adequate for drafts or for sending to a colleague to edit, but they are not really usable for final-format publishing. For this you need to use packages to design it yourself, or (better) use a class file designed by your publisher or institution (or yourself!) to fit the type

of publication. Quite often these are based on the default classes for compatibility, but typeset quite different output.

### 2.1.3 Document class options

The default class layouts were originally designed to fit as drafts on US 'Letter' size paper.[4] However, the default paper size in TEX Live is now A4, so to create documents in North America, you need to specify the paper size in an optional argument in square brackets before the document class name, eg

```
\documentclass[letterpaper]{report}
```

The geometry package, which we will see later, lets you specify other bigger and smaller paper sizes.[5]

Books and journals are not usually printed on office-size paper. Although for draft purposes LATEX's layouts fit on the standard A4 or Letter stationery in your printer, it makes them look odd: the margins are too wide and the font size is too small, because the finished print job will normally be trimmed to a completely different size entirely — try printing a few pages of the PDF version of this chapter and then trimming the margins at the pale blue crop marks to make it 188 mm × 235 mm (the same as the *Companion* series) and you'll see how it changes the appearance.

The other default settings in the built-in classes are for:

1. 10pt type (all document classes);
2. two-sided printing (books and reports) or one-sided (articles and letters);
3. separate title page (books and reports only).

[4] 'Letter' size is 8½″×11″, which is the trimmed size of the long-obsolete Demy Quarto, still in use in North America. Other common US office sizes are 'Legal', which is 8½″×14″, a 'bastard' (variant) cutting similar to the old Foolscap (8¼″×13¼″); Ledger or Tabloid (11″×17″, which is exactly twice 'Letter', in the same way that A3 is twice A4); and 'Executive' (7″×10″). The US still has many other sizes obsolete elsewhere, especially in government offices. International Organization for Standardization (ISO) standard 'A', 'B', and 'C' paper sizes, used everywhere else in the world, are still largely unknown in many parts of North America.

[5] Note that the standard built-in document classes (book, article, report, or letter) only use the paper size to adjust the margins: they do not embed the paper size name in the PostScript or PDF output. For this you need the geometry package in order to ensure that the paper size name gets embedded correctly in the output, otherwise printers may select the wrong paper tray, or reject the job.

These can be modified with the following document class options which you can add in the same set of square brackets, separated by commas (the *10pt* option is the default):

***11pt*** to specify 11pt type (headings, footnotes, etc get scaled up or down in proportion);

***12pt*** to specify 12pt type (again, headings etc get scaled to match);

***oneside*** to format one-sided printing for books and reports;

***twoside*** to format articles or letters for two-sided printing;

***titlepage*** to force articles to have a separate title page (books and reports get that automatically);

***draft*** makes LaTeX highlight any hyphenation or justification problems with a small square in the right-hand margin so they can be located quickly by you or a proofreader. This option also sets graphics to print as an empty rectangle containing just the filename of the image, so that image-heavy drafts will print more quickly and use less ink or toner.

So, if you were using LaTeX for a report to be in 12pt type on Letter paper, but printed one-sided in draft mode, you would use:

```
\documentclass[12pt,letterpaper,oneside,draft]{report}
```

The 10pt, 11pt, and 12pt settings cover between them probably 99.9% of all common text-document typesetting. There are extra options for other body type sizes in the extsizes bundle of document classes (extarticle, extbook, extreport, etc), and various national and international organisations supporting the visually-impaired have special large-type document class options.

## 2.2 The document environment

After the Document Class Declaration, the text of your document is enclosed between the two commands we saw in section 1.5 on page 12: `\begin{document}` and `\end{document}`. These identify the beginning and end of the text of your document (so in the example below, you would put your text where the dots are):

**Exercise 7 –** Create a new document

1. Use your editor to create a new, empty document

   If your editor insists on filling your new document with template material, delete it all so that the file is empty;
2. Type in a Document Class Declaration as shown above;
3. Add a font size option if you wish;
4. In North America, omit the *a4paper* option or change it to *letterpaper*;
5. Save the file (make up a name) ensuring the name ends with `.tex`.

### Global options

In addition to any options specific to the document class, it is also possible to put package options in the `\documentclass` options argument *instead of* in the `\usepackage` command (see section 3.1.2 on page 58), provided they are not implemented by more than one package. Packages which do not implement the named option at all are supposed to silently ignore it.

### Picking suitable filenames

Never, *never*, NEVER create directories (folders) or file names which contain spaces or non-printing, non-ASCII characters. Although your operating system may support them, some don't, and they will only cause grief and tears, especially in automation software like document builders, web scripts, and app-based remote compilers.

Make filenames as short or as long as you wish, but strictly avoid spaces. Stick to upper- and lower-case letters *without* accents (A–Z and a–z), the digits 0–9, the hyphen (-), the underscore (_), and the dot (full point or period: `.`) — similar to the conventions for a Web URI: it will let you refer to TEX files over the Web more easily, make your files more portable, and make it easier to use standard system utilities and applications, as well as those distributed with TEX

```
\documentclass[11pt,a4paper,oneside]{report}

\begin{document}
...
\end{document}
```

The reason for marking the beginning of your document text is that LaTeX allows you to insert your setup and design specifications before it (where the blank line is in the example above: we'll be using this soon).

The reason for marking the end of your document text is that LaTeX stops processing at that point. You can therefore store comments or temporary text *underneath* the `\end{document}` in the knowledge that LaTeX will never see them and will never try to typeset them (they don't even need to be preceded by the `%` comment character), but they will remain in your file for you to see in your editor, maybe to copy and paste for re-use in a later edit.

```
...
\end{document}
Don't forget to get the extra chapter from Jim!
```

This `\begin`...`\end` pair of commands is an example of a common LaTeX structure called an ***environment***. Environments enclose text which is to be handled in a particular way. All environments start with `\begin{...}` and end with `\end{...}` (putting the name of the environment in the curly braces each time).

If you're familiar with HTML, SGML, or XML you'll recognise this technique: it's just like start-tags and end-tags.

**Exercise 8 –** Adding the document environment

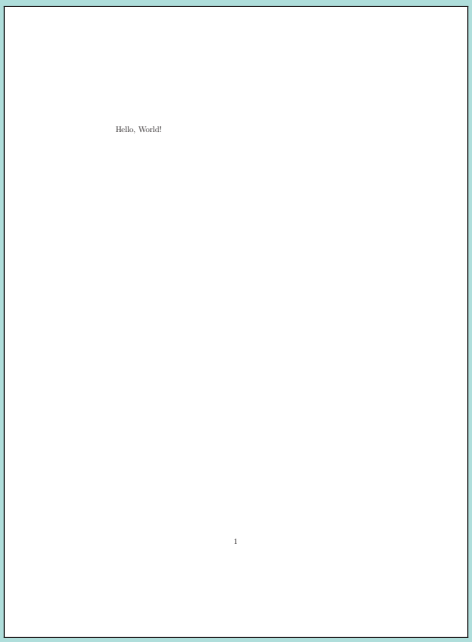
1. Add the *document* environment to your new file;
2. In between the Document Class Declaration and the \begin{document}, add the command which allows the use of UTF-8, TrueType, OpenType, and more:

```
\usepackage{fontspec}
```

3. In the *document* environment, type the phrase Hello, World!:

```
\begin{document}
Hello, world!
\end{document}
```

4. Save the file and typeset it with XeLaTeX and you should get some PDF output like this:



## 2.3 Titling

The first thing you actually put in the *document* environment is almost always the document title, the author's name, and the date (except in letters, which have a special set of commands for addressing). The title, author, and date are all examples of *metadata* (information *about* information).

```
\documentclass[11pt,a4paper,oneside]{report}
\begin{document}
\title{Practical Typesetting}
\author{Peter Flynn}
\date{January 2022}
\maketitle
\end{document}
```

The \title, \author, and \date commands are self-explanatory. You put the title, author name, and date in curly braces after the relevant command. The title and author are compulsory; if you omit the \date command, LaTeX uses today's date by default. If you don't want a date at all, use \date{} (ie an empty date).

You MUST finish the metadata with the \maketitle command, which tells LaTeX that it's complete and it can typeset the titling information at this point. If you omit \maketitle, the titling will never be typeset.

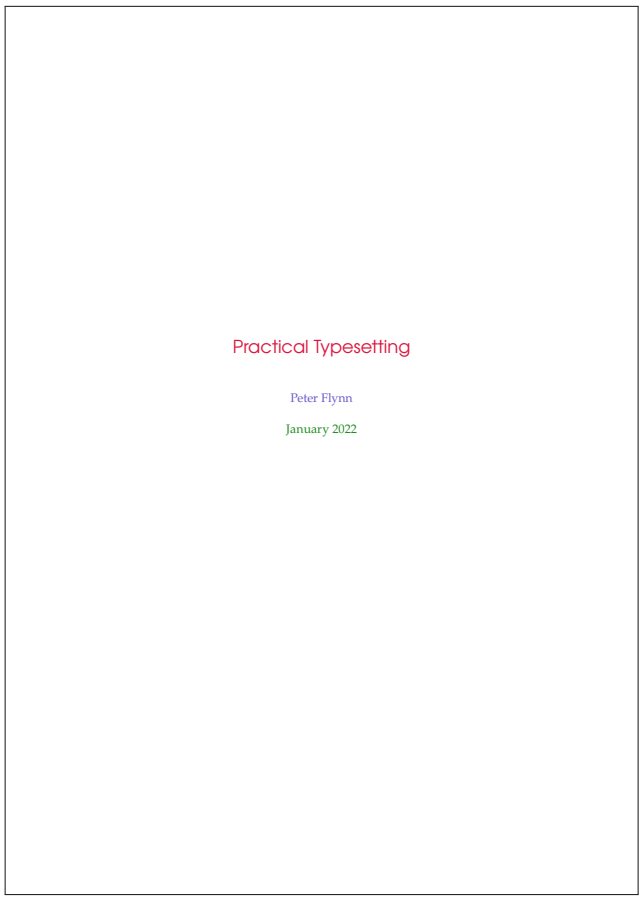
**Different title layouts**

The \maketitle command, like all LaTeX commands, is reprogrammable, so you can alter the appearance of titles (like I did for the printed version of this document). It also means publishers can create new commands like \datesubmitted and \editversion in their own document classes. Details about reprogrammability are in Chapter 7 starting on page 181.

When this file is typeset, you get something like Figure 2.1 on the following page (I've cheated and done it in colour for fun — yours will be in black and white for the moment). This is a report, so the title appears all by itself on a single page.

If you have mistyped a command, you may get an error message: see section B.3 on page 237 to resolve this.

**Figure 2.1 –** Titling information typeset on the title page




## 2.4 Abstracts and summaries

In reports and articles it is usual for the author to provide an Summary or Abstract, which describes the content and explains its importance. Abstracts in articles are usually only a couple of paragraphs long. Summaries in reports or theses can run to several pages, depending on the length and complexity of the document or the readership it's aimed at.

```
\documentclass[11pt,a4paper,oneside]{report}
\usepackage{fontspec}
\begin{document}
\title{Practical Typesetting}
```

**Exercise 9 –** Adding your metadata

1. Add the \title, \author, \date, and \maketitle commands to your file.
2. Use your own name, make up a title, and give a date.
3. Typeset the document and check that it's right.

The order of the first three commands is not important, but the \maketitle command must come last.

```
\author{Peter Flynn}
\date{January 2022}
\maketitle
\begin{abstract}
This document presents the basic concepts of typesetting
in a form usable by non-specialists. It is aimed at those
who find themselves (willingly or unwillingly) asked to
undertake work previously sent out to a professional
printer, and who are concerned that the quality of work
(and thus their corporate æsthetic) does not suffer.
\end{abstract}
\end{document}
```

In both cases (reports and articles) the Abstract or Summary is OPTIONAL (that is, LaTeX doesn't force you to have one), but it's rare to omit it because readers want and expect it, and it's used by web indexing engines to let people find your work. In practice, of course, you go back and type the Abstract or Summary *after* having written the rest of the document, but for the sake of the example we'll jump the gun and type it now.

You add the *abstract* environment *after* the \maketitle command,[6] and type your Abstract or Summary in it, leaving a blank line between paragraphs if there's more than one (see section 2.7 on page 54 for this convention).

In business and technical documents, the Abstract is often called a Management Summary, or Executive Summary, or Business Preview, or some similar phrase. LaTeX lets you change the name associated with the *abstract* environment to any other suitable text. The heading associated with the *abstract* environment is

[6] A few publishers' journal style packages ask for the Abstract to be typed *before* the \maketitle because they do special formatting with it along with the title block.

called the \abstractname, and you can use the \renewcommand command in your Preamble to give it a new value:

```
\renewcommand{\abstractname}{Summary}
```

This does not change the name of the environment, only its heading: you still use \begin{abstract} and \end{abstract}.

**Exercise 10 –** Using an Abstract or Summary

1. Add the \renewcommand as shown above to your Preamble (call it something other than Summary if you prefer).

   (The Preamble is at the start of your document, in between the \documentclass line and the \begin{document}: see the panel 'The Preamble' on p. 13).);

2. Add an ***abstract*** environment after the \maketitle and type in a paragraph or two of text.

3. Typeset the document.

Notice how the name of the command you are renewing (in this example, the \abstractname) goes in the first set of curly braces, and the new value you want it to have goes in the second set of curly braces (this is an example of a command with *two* arguments).

## 2.5 A little think about structure

It's very easy to sit down at a keyboard with a traditional wordprocessor and just start typing. If it's a very short document, or something short-lived or relatively unimportant, then you just want to type it in and make it 'look nice' by highlighting with the mouse and clicking on font styles and sizes.

In doing so, you may achieve the effect you wanted, but your actions have left no trace behind of *why* you made these changes. This is not important for trivial, ephemeral, or short-term documents. Sometimes, though, you may need to write longer, more permanent, or more complex documents, or documents arranged to a regular pattern like reports or articles. Making them consistent by manual methods

then becomes a nightmare, and an enormous waste of time, because everything has to be formatted and reformatted by hand.

LaTeX's automation is based on you providing the 'why' information, identifying the elements of your document by name, and letting the template or stylesheet take care of the formatting.

**Exercise 11 –** Reasoning

If your documents have any of the features below, then you have probably already started thinking about structure.

- ☐ My document naturally divides into sections (parts, chapters, etc).
- ☐ My document is long.
- ☐ There is lots of repetitive formatting in my document.
- ☐ My document is complex (intellectually or visually).
- ☐ There are lots of figures or tables (or examples, exercises, panels, sidebars, etc) in my document.
- ☐ Accuracy is important in formatting my document.
- ☐ A master copy of my document is needed for future reference or reprinting.
- ☐ This is a formal or official document needing special care and attention.
- ☐ My document (or part of it) may need ongoing or occasional re-editing and republishing.
- ☐ It's *my* thesis, book, white paper, leaflet, pamphlet, paper, article, etc. *That*'s why I care.

If you've got this far, you're over half-way done. Using a structural editor — even a simple outliner — can make a huge difference to the quality of your thinking because you are consciously organising your thoughts before setting them down. And it can make just as big a difference to your formatting as well: more consistent, better presented, easier for the reader to navigate through, and more likely to be read and understood — which is presumably why you are writing the document in the first place.

## 2.6 Sections

LaTeX provides seven levels of division or sectioning for you to use in structuring your text. They are all optional: it is perfectly possible to write a document consisting solely of paragraphs of unstructured text. But even novels are normally divided into chapters, although short stories are often made up just of paragraphs.

**Table 2.1 –** LaTeX's sectioning commands

| Depth | Division | Command | Notes |
|---|---|---|---|
| −1 | Part | `\part` | Not in letters |
| 0 | Chapter | `\chapter` | Books, reports |
| 1 | Section | `\section` | Not in letters |
| 2 | Subsection | `\subsection` | Not in letters |
| 3 | Subsubsection | `\subsubsection` | Not in letters |
| 4 | Titled paragraph | `\paragraph` | Not in letters |
| 5 | Titled subparagraph | `\subparagraph` | Not in letters |

Chapters are only available in the book and report document classes, because chapters don't have any meaning in articles or letters. Parts are also undefined in letters.[7]

In each case the title of the part, chapter, section, etc goes in curly braces after the command. LaTeX automatically calculates the correct numbering and prints the title in bold. You can turn section numbering off at a specific depth: details are in section 2.6.1 on the next page.

```
\section{New recruitment policies}
...
\subsection{Effect on staff turnover}
...
\chapter{Business plan 2020--2030}
```

There are packages to let you control the typeface, style, spacing, and appearance of section headings: it's much easier to use them than to try and reprogram the headings manually. Two of the most popular are section and sectsty.

Headings also get put automatically into the Table of Contents, if you specify one (it's optional). But if you make manual styling changes to your heading, for

[7] It is arguable that chapters also have no place in reports, either, as these are conventionally divided into sections as the top-level division. LaTeX, however, assumes your reports have chapters, but this is only the default, and can be changed very simply (see section 7.6 on page 190).

example a very long title, or some special line-breaks or unusual font-play, this would appear in the Table of Contents as well, which you almost certainly *don't* want. LaTeX allows you to give an optional extra version of the heading text which only gets used in the Table of Contents and any running heads, if they are in effect (see section 6.1.2 on page 146). This alternative heading goes in [square brackets] before the curly braces:

```
\section[Effect on staff turnover]{An analysis of the
effects of the revised corporate recruitment policies
on staff turnover at divisional headquarters}
```

**Exercise 12 –** Start your document text

1. Add a `\chapter` command after your Abstract or Summary, giving the title of your first chapter.
2. If you're planning ahead, add a few more `\chapter` commands for subsequent chapters. Leave a few blank lines between them to make it easier to add paragraphs of text later.
3. Typeset the document.

### 2.6.1 Section numbering

All document divisions get numbered automatically. Parts get Roman numerals (Part I, Part II, etc); chapters and sections get decimal numbering like this document, and Appendixes (which are just a special case of chapters, and share the same structure) are lettered (A, B, C, etc). You can easily change this default if you want some special scheme.

You can change the depth to which section numbering occurs, so you can turn it off selectively. In this document the depth is set to 3, using the depth column in Table 2.1 on the preceding page. If you only want parts, chapters, and sections numbered, not subsections, subsubsections, or lower levels, you can change the value of the `secnumdepth` counter using the the `\setcounter` command, giving the depth value from Table 2.1 on the facing page:

```
\setcounter{secnumdepth}{1}
```

Notice that the \setcounter command, like \renewcommand which we saw earlier, has two arguments: the name of the counter you want to set, and the number you want to set it to.

A related counter is tocdepth, which specifies what depth to take the Table of Contents to. It can be reset independently, in exactly the same way as secnumdepth. The setting for this document is 2.

```
\setcounter{tocdepth}{3}
```

To get a one-time (special case) *unnumbered* section heading which does *not* go into the Table of Contents, follow the command name with an asterisk before the opening curly brace:

```
\subsection*{Shopping List}
```

All the divisional commands from \part* to \subparagraph* have this 'starred' version which can be used in isolated circumstances for an unnumbered heading when the setting of secnumdepth would normally mean it would be numbered.

### 2.6.2 Table of contents

All auto-numbered headings (parts, chapters, sections, subsections, etc) get entered in the Table of Contents (ToC) automatically. You don't have to print a ToC, but if you want to, add the command \tableofcontents at the point where you want it printed (usually after the Abstract or Summary).

Entries for the ToC are recorded each time you typeset your document, and only reproduced the *next* time you typeset it, so you need to run LaTeX an extra time to ensure that all ToC page-number references are correctly resolved.

**Table of Contents automation**

Your editor should automatically run LaTeX twice when needed, if you are using the Build, Compile, Typeset, or Make button or menu entry (see /1:note@tocprocess belowtypeset-examples). It is also done automatically by processing tools like *latexmk*, but if you are processing LaTeX manually by typing the commands in a terminal window, you will need to do this yourself.

The commands \listoffigures and \listoftables work in exactly the same way as \tableofcontents to automatically list all your tables and figures. If you use them, they normally go after the \tableofcontents command.

We've already seen in section 2.6 on page 50 how to use the optional argument to the sectioning commands to add text to the ToC which is slightly different from the one printed in the body of the document. It is also possible to add extra lines to the ToC, to force extra or unnumbered section headings to be included.

**Exercise 13 –** Using a Table of Contents

1. Add the \tableofcontents command to your document, before or after the Abstract, as you prefer.
2. Typeset the document.
3. Check that the Table of Contents is now showing. If not, typeset the document again.

If what you expect doesn't appear, you should always check the log file or error display: you might have made a typing mistake in a command.

A \tableofcontents command normally shows only numbered section headings, and only down to the level defined by the tocdepth counter (see section 2.6.1 on page 51), but you can add extra entries with the \addcontentsline command. For example if you use an unnumbered section heading command to start a preliminary piece of text like a Foreword or Preface, you can write:

```
\subsection*{Preface}
\addcontentsline{toc}{subsection}{Preface}
```

This will format an unnumbered ToC entry for 'Preface' in the 'subsection' style. You can use the same mechanism to add lines to the List of Figures or List of Tables by substituting lof or lot for toc.

There is also a command \addtocontents which lets you add any LaTeX commands to the ToC file. For example, to add a horizontal rule and a 6pt gap at some special place, you could say

```
\addtocontents{toc}{\par\hrule\vspace{6pt}}
```

at the place where you want it to occur. You should probably only use this command once you know what you are doing.

There are several packages to help you restyle these lists of contents automatically; perhaps the best-known is tocloft.

## 2.7 Ordinary paragraphs

After section headings comes your text. Just type it and leave a blank line between paragraphs. That's all LaTeX needs.

The blank line means 'end the current paragraph here': it is *not* (repeat: ***not***) for creating a blank line in the typeset output.

**Multiple blank lines**

Leaving multiple blank lines between paragraphs in your source document does *not* create extra white-space. As we saw in the Note on p. 16, all extra blank lines are ignored by LaTeX: the space between paragraphs is controlled *only* by the value of `\parskip`.

The spacing between paragraphs is an independently definable quantity, a ***dimension*** or ***length*** called `\parskip`. This is normally zero (no space between paragraphs, because that's how books and articles are normally typeset, but see below), but you can easily set it to any size you want with a `\setlength` command in your Preamble — like the `\setcounter` command we saw in section 2.6.1 on page 51 it takes two arguments: the name of the length, and the value to set it to:

```
\setlength{\parskip}{5mm}
```

This will set the space between paragraphs to 5mm. See section 1.10.1 on page 26 for details of the various size units LaTeX can use.

Most books and articles are set with no space between paragraphs (and indentation at the start of them). If you want to use the popular office-document style of having space between paragraphs (and no indentation), use the parskip package, which does it for you. It also makes adjustments to the spacing of lists and other structures which use paragraph spacing, so they don't get too far apart.

White-space in LaTeX can also be made flexible (what Lamport calls 'rubber' lengths). This means that values such as \parskip can have a default dimension plus an amount of expansion minus an amount of contraction. This is useful on pages in complex documents where not every page may be an exact number of fixed-height lines long, so some give-and-take in vertical space is useful. You can specify this in a \setlength command:

```
\setlength{\parskip}{1cm plus4mm minus3mm}
```

Paragraph indentation can also be set with the \setlength command, although you would always make it a fixed size, never a flexible one, otherwise you would have very ragged-looking paragraphs.

```
\setlength{\parindent}{6mm}
```

By default, the first paragraph after a chapter or section heading follows the standard Anglo-American publishers' practice of *no* indentation. Subsequent paragraphs are indented by the value of \parindent (default 18pt).[8] You can change the value of \parindent in the same way as any other length.

In the printed version of this document, the paragraph indentation is set to 12.0pt and the space between paragraphs is set to 0.0pt plus 1.0pt. These values do not apply in the Web (HTML) version because not all browsers are capable of that fine a level of control, and because users can apply their own stylesheets regardless of what this document proposes.

To turn off indentation completely, set it to zero (but you still have to provide units: it's still a measure!).

```
\setlength{\parindent}{0in}
```

If you do this, though, and leave \parskip set to zero, your readers won't easily be able to tell where each paragraph begins!

[8] Paragraph spacing and indentation are cultural settings. If you are typesetting in a language other than English, you should use the babel or polyglossia packages, which alter many things, including the spacing and the naming of sections, to conform with the standards of different cultures, countries, and languages.

**Exercise 14 –** Start typing!

By now you know enough about LaTeX's basic commands to write a whole document.

1. Give a title and your name as author (and a date if you want) and don't forget the `\maketitle` (see Figure 2.1 on page 46);
2. Add an Abstract if you need to (see section 2.4 on page 46).
3. Start with a `\chapter` (for books or reports) or a `\section` (for articles) with the title of the chapter or section.
4. Type your text in paragraphs. Leave a blank line between each. Don't bother about line-wrapping or formatting — LaTeX will take care of all that.
5. Use the geometry package to change the margins and size of the text body.