Basic Document Formatting

Rwitaban Goswami and Mihir Vahanwala

July 3, 2020

Contents

1 Organising the document **1**
1.1 Segregation 1
1.2 Demonstrating the declarations 1
1.3 Referencing and Table of Contents 2
1.3.2 TOC: Adjusting Depth 2

2 Page Layout **4**

3 Headers, Footers and Footnotes **5**
3.1 Page Styles 5
3.2 Footnotes 6

4 Indices **7**
4.1 Introduction 7
4.2 Basic Indices 7
4.3 Entries & Subentries 9
4.4 Formatting the Index 9
4.5 Including Index in the TOC 10

5 Multiple Columns **11**
5.1 Introduction 11
5.2 Multicol 11

Document Index **14**

1 Organising the document

1.1 Segregation

The first step in writing a coherent document is organising your content into parts, chapters and sections. The following "divisions" are ordered (that is, LaTeX maintains an inbuilt counter for them), and here's the hierarchy of their depths:

Depth	How to declare
-1	`\part{Generic Part Name}`
0	`\chapter{Generic Chapter Name}`
1	`\section{Generic Section Name}`
2	`\subsection{More Intricate}`
3	`\subsubsection{Minutiae}`

This is a relatively small document, we have used the article document class. `\part` and `\chapter` are only available in the *book* and *report* document classes.

As you go deeper, you have `\paragraph` and `\subparagraph` as well. Although they aren't ordered, they can show up in the table of contents with the appropriate setting. More on that later.

1.2 Demonstrating the declarations

This is the second subsection. We will now make a subsubsection, and give you a sneak peek into the code.

1.2.1 Sample Subsection

Hey. Why are you reading this? Go ahead to the next subsubsection. It kinda sounds funny when you read it aloud though.

1.2.2 The straightforward code

Here is what the code looks like. The code we show will only typeset upto this sentence.

If we go beyond, will we be caught in an infinite loop?! (Mihir seems high.)

```
\subsection{Demonstrating the declarations}
\label{declaration-demo}
This is the second subsection. We will now make a
    subsubsection, and give you a sneak peek into the code
    .

\subsubsection{Sample Subsection}
\label{sample}
```

```
Hey. Why are you reading this? Go ahead to the next
    subsubsection. It kinda sounds funny when you read it
    aloud though.

\subsubsection{The straightforward code}
\label{straightforward-code}
Here is what the code looks like. The code we show will
    only typeset upto this sentence.
```

1.3 Referencing and Table of Contents

1.3.1 Referencing

Notice that we put a label at the beginning of each sub(sub)section. This allows us to reference them. You will have to declare `\usepackage{hyperref}` for the references to work. We typed in `\ref{declaration-demo}` and now can say, "Recall Section 1.2" like a Prof.

Why is this useful, you ask? Couldn't you manually keep track of the section numbers by yourself? Well, imagine this scenario, I just wrote the code we showed you in Section 1.2.2. Now I reference Section 1.2.1. To do that I manually lookup the Section number, i.e. 1.2.1, and write "Section 1.2.1" in my code. Now suppose I come along and want to add another section called "New Sample" *above* my Sample Subsection. I will have to change *all* instances of wherever I had referenced Section 1.2.1 and change it to "Section 1.2.2". If I simply write "Section `\ref{sample}`" instead of "Section 1.2.1", LaTeX will **automatically** do it for us!

Note: You will have to compile twice for the hyper-references to work properly. (Some LaTeX software like the VScode plugin will automatically compile it twice for you)

The package also makes the entries of the Table of Contents hyperlinks!

1.3.2 TOC: Adjusting Depth

To generate a table of contents, all you have to do is declare `\tableofcontents`. By default, all headings of depth up to 3 will be displayed.

However, in our document, we've chosen to limit our depth to 2 (unless we explicitly make an exception). To do that, simply declare `\setcounter{tocdepth}{2}` in the preamble. Notice that this subsubsection is an exception.

Before declaring this subsubsection, we did this:

```
\addtocontents{toc}
{\setcounter{tocdepth}{3}}
\subsubsection{TOC: Adjusting Depth}
```

Right before the point where you want a change in depth, make the above declaration.

Now, before proceeding to the next section, we will set the depth back to 2, so that subsubsections will no longer show up in the Table of Contents. (Just as before)

```
\addtocontents{toc}
{\setcounter{tocdepth}{2}}
\section{Page Layout}
```

2 Page Layout

The geometry package makes adjusting the page layout in terms of paper size and margins very convenient.

A typical declaration in the preamble would look something like:

```
\usepackage[a4paper,hmargin=1in,vmargin=1.5in]{geometry}
%We want A4 paper, and have specified horizontal and
    vertical margins respectively.
```

This setting is the default for the document. Now, for some reason, you may want to change the margins for a few consecutive pages, and then revert to the original settings specified in the preamble. It is here that the `\newgeometry{}` and `\restoregeometry` commands prove useful. These commands are only available *within* the document environment.

For example,

```
%the above layout L1 declared in the preamble.
\begin{document}

%content in layout L1

\newgeometry{left=1.5in, right=0.5in, top=1in, bottom=1.5
    in}
%here we specify each of the four margins
%let this new layout be L2.
%\newgeometry does not allow you to change paper size or
    portrait/landscape orientation!
%for two sided document classes like books, left and
    right correspond to inner and outer margins
    respectively.

%\clearpage declared implicitly with the above command, i
    .e. new page

%Content in layout L2 next page onward.

\restoregeometry
%\clearpage declared implicitly

%Content in layout L1 next page onward

\end{document}
```

This is just a guide to get you motivated, the complete arsenal of nuanced options the geometry package provides is documented here.

3 Headers, Footers and Footnotes

3.1 Page Styles

There's something different about this document. Compared to the last one, it just seems... fancier. Of course, it's because we defined custom headers and footers. All the work is done in the preamble; have a look.

```
\documentclass{article}
...
\usepackage{fancyhdr}
...
\pagestyle{fancy}
%declares that a custom page style is to be used

\fancyhf{} %clears the default header and footer that
    appear as part of the plain style

%plain style is default: no header, page number in the
    centre of the footer
%other inbuilt options include:
%empty: no header, no footer. Blank
%myheadings: try it out for a document yourself!

\lhead{Basic Document Formatting} %left of the header
\rhead{Section \thesection}
\lfoot{R Goswami and M Vahanwala}
\rfoot{Page \thepage} %right of the footer

%you can also define \chead and \cfoot:
%c for centre!

% "\thesection" is a reference to the counter that keeps
    track of what section we're in! Ditto for "\thepage"

\renewcommand{\footrulewidth}{1pt}
```

The above demo should help serve most of your purposes. Other two-sided document classes use a slightly different approach. If you are particularly enthusiastic, we encourage you to read up on that here. It's comprehensive.

You might have noticed that the fancy style doesn't apply to the page with the title and table of contents. Also, page numbering starts *after* that. Here's what we did:

```
\maketitle
\tableofcontents
\thispagestyle{empty}
```

```
\clearpage
\pagenumbering{arabic}
```

For reasons related to the working of the `\maketitle` and `\tableofcontents` commands, you must put the `\thispagestyle{}` command after them to force the empty style.

When you design headers and footers yourself, we recommend you tinker with the following commands and see what they do:

```
\leftmark
\rightmark
%for example
\rhead{\leftmark}
```

3.2 Footnotes

Suppose you are making a point in your document. The assertion needs a clarifying remark, but making it immediately will take the focus away from the actual story. Here's where footnotes come in handy.[1].To maintain the readability in the source code[2], consider this technique.

The current value of the footnote counter is at 2. You can refer to that footnote again[2]. Or to a previous justification[1].

Footnote indexing comes in various styles.[‡§]

```
Here's where footnotes come in handy.\footnote{You see
    them in lecture slides all the time}.To maintain the
    readability in the source code\footnotemark, consider
    this technique.

\footnotetext{Your collaborators are important!}

The current value of the footnote counter is at 2. You
    can refer to that footnote again\footnotemark[\value{
    footnote}]. Or to a previous justification\
    footnotemark[1].

\renewcommand{\thefootnote}{\fnsymbol{footnote}}
Footnote indexing comes in various styles.\footnote{This
    is one of them... as well.}\footnote{There are nine
    symbols from where this came from.}
```

[1] You see them in lecture slides all the time

[2] Your collaborators are important!

[‡] This is one of them. See the source code. Experiment with Roman, roman, arabic, Alph and alph. Put the renewcommand in the preamble to get this style for the whole document. Of course, you can toggle within the document as well.

[§] There are nine symbols from where this came from.

4 Indices

4.1 Introduction

You've definitely found yourself opening the last few pages of a textbook, and you would've noticed a little handy something called the *index*. It contains a list of all the keywords found in the book, and where to find those keywords in the book. If you look at this document, you'll find a similar section right at the end.

4.2 Basic Indices

To insert an index section into your document, you do not have to do anything manually. Once again LaTeX packages have got you covered!

With LaTeX and the support program `imakeidx`, an index can be generated quite easily.

Let's see a simple working example.

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[parfill]{parskip}
%Other packages
\usepackage{imakeidx}
\makeindex

\begin{document}
\tableofcontents
%Other sections
\section{Indices}
\subsection{Introduction}
If you have find yourself that you opened the last few
    pages of a textbook, you'll notice a little handy
    something called the \textit{index}\index{index}. It
    contains a list of all the keywords\index{keywords}
    found in the book, and where to find those keywords in
     the book. If you look at this document\index{document
    }, you'll find a similar section\index{section} right
    at the end.
\subsection{Basic Indices}
To insert an index section\index{section} into your
    document, you do not have to do anything manually.
    Once again \LaTeX{}\index{\LaTeX} packages\index{
    package} have got you covered!

With \LaTeX{} and the support program \verb!imakeidx!\
    index{imakeidx}, an index can be generated quite
```

```
    easily.

%Rest of the document
\printindex

\end{document}
```

This creates the following index:

Figure 1: Simple Index defined in Section 4.2

First, the package is included in the preamble by the line `\usepackage{imakeidx}`

Then the command `\makeindex` is mandatory in the preamble for the index to work and can take some parameters to customize its appearance, in the next sections this will be clear.

To add an entry to the index the command `\index{}` is used, where the word to be added is inserted as the parameter. Take note that the `\index{}` command affects only the index section, and will not insert anything at the location you have used the `\index{}` command. It is kind of like using the `\label{}` command to label a section, instead you are labelling a keyword, and it is referenced automatically in the index section.

Finally, the command `\printindex` will actually render the index. Note that the index section will actually be inserted wherever you write the command `\printindex`

Note: instead of `imakeidx` *the package* `makeidx` *may be imported, but offers less customizations possibilities.*
Note: When adding an index to a project on Overleaf, it is important to note that for the index to compile properly, the main .tex file needs to be in the root directory of the project, outside of any folders. This ensures that the auxiliary files needed to generate the index are cached in a way that they are accessible to the compiler.

4.3 Entries & Subentries

If you go look up a real index in a book now, you'll see that indices are often *nested*. Suppose you have the keyword LaTeX in your index, but you also want the keyword "LaTeX document". Nested indices will allow you to do this.

```
1 \subsection{Introduction}
2 If you have find yourself that you opened the last few
    pages of a textbook, you'll notice a little handy
    something called the \textit{index}\index{index}. It
    contains a list of all the keywords\index{keywords}
    found in the book, and where to find those keywords in
     the book. If you look at this document\index{\LaTeX!
    document}, you'll find a similar section\index{index!
    section} right at the end.
3 \subsection{Basic Indices}
4 To insert an index\index{index} section\index{index!
    section} into your document, you do not have to do
    anything manually. Once again \LaTeX{}\index{\LaTeX}
    packages\index{\LaTeX!package} have got you covered!
5
6 With \LaTeX{} and the support program \verb!imakeidx!\
    index{\LaTeX!package!imakeidx}, an index can be
    generated quite easily.
```

This generates the index:

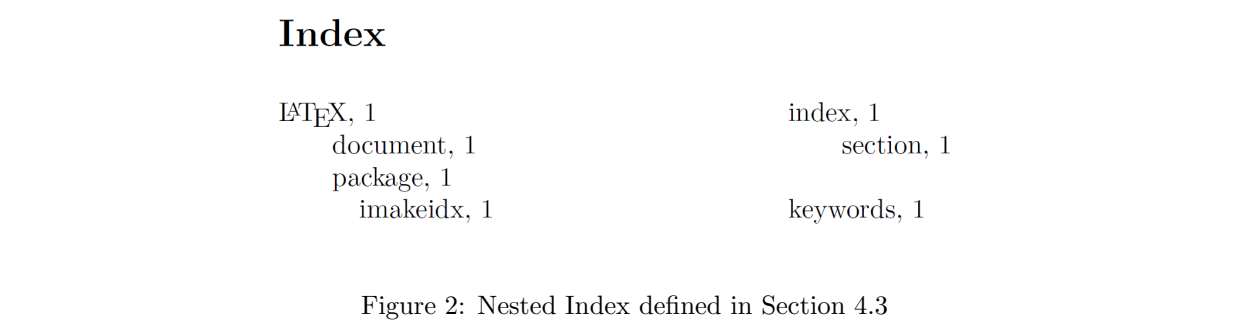


Figure 2: Nested Index defined in Section 4.3

It is a common practice to extend terms in the index to include special adjectives. For such cases you can add a exclamation mark "!" that will add the terms after this mark as sub entries of the main word.

In the example, the word "LaTeX" has the word "document" as sub entry.

4.4 Formatting the Index

Simple formatting for the index, such as changing the index title, adding several columns and changing the column width can be easily done passing optional

values to `\makeindex`

```
\makeindex[columns=1, title=Document Index]
```

In this example the index is presented in a three-column format with the parameter `columns=1`, and the title is changed to "Document Index" by `title=Alphabetical Index`; these parameters are passed to `\makeindex` in the preamble. See the reference guide for a list of available parameters and their description.

The index generated looks like:

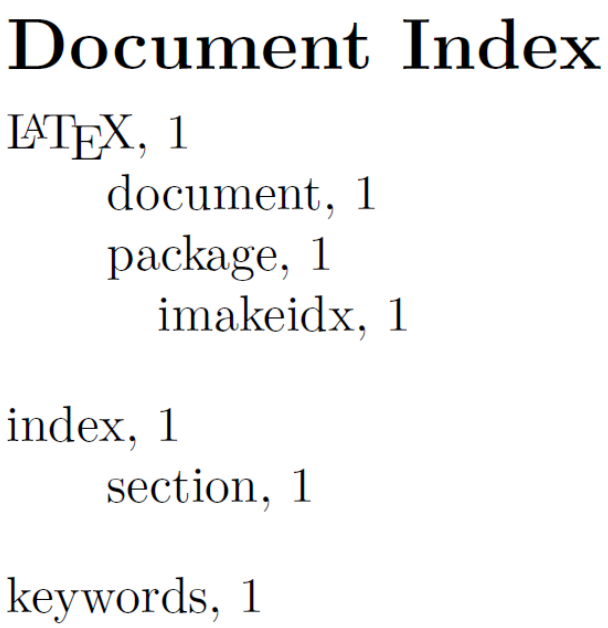


Figure 3: Stylized defined in Section 4.4

All the possible parameters to `\makeindex` can be found here `https://www.overleaf.com/learn/latex/Indices`. You can peruse the parameters list if you want to modify the look of your index.

4.5 Including Index in the TOC

You might notice that our TOC has the index section included in it. But while you are playing around with indices, your TOC will not have an index section.

What is that about? By default, the index is not included in the table of contents, this can be easily reverted.

```
\makeindex[columns=1, title=Document Index, intoc]
```

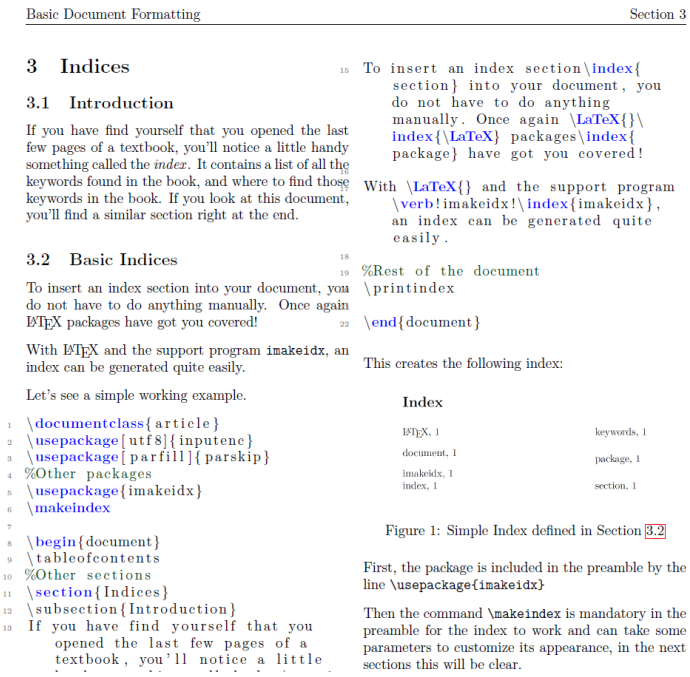
By adding the parameter `intoc` to the command `\makeindex` the inclusion of the index title into the table of contents is enabled.

5 Multiple Columns

5.1 Introduction

Have you ever seen newspaper articles where the content is divided into two or more columns? You can do that in LaTeX too. Just pass in the parameter `\twocolumn` in the `\documentclass{}` command, like `\documentclass[twocolumn]{article}`

The render looks like this:



Basic Document Formatting Section 3

3 Indices

3.1 Introduction

If you have find yourself that you opened the last few pages of a textbook, you'll notice a little handy something called the *index*. It contains a list of all the keywords found in the book, and where to find those keywords in the book. If you look at this document, you'll find a similar section right at the end.

3.2 Basic Indices

To insert an index section into your document, you do not have to do anything manually. Once again LaTeX packages have got you covered!

With LaTeX and the support program `imakeidx`, an index can be generated quite easily.

Let's see a simple working example.

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[parfill]{parskip}
%Other packages
\usepackage{imakeidx}
\makeindex

\begin{document}
\tableofcontents
%Other sections
\section{Indices}
\subsection{Introduction}
If you have find yourself that you
    opened the last few pages of a
    textbook, you'll notice a little
To insert an index section\index{
    section} into your document, you
    do not have to do anything
    manually. Once again \LaTeX{}\
    index{\LaTeX} packages\index{
    package} have got you covered!

With \LaTeX{} and the support program
    \verb!imakeidx!\index{imakeidx},
    an index can be generated quite
    easily.

%Rest of the document
\printindex

\end{document}
```

This creates the following index:

Index

LaTeX, 1 keywords, 1
document, 1 package, 1
imakeidx, 1
index, 1 section, 1

Figure 1: Simple Index defined in Section 3.2

First, the package is included in the preamble by the line `\usepackage{imakeidx}`

Then the command `\makeindex` is mandatory in the preamble for the index to work and can take some parameters to customize its appearance, in the next sections this will be clear.

5.2 Multicol

5.2.1 Multicol Sample

The following text is an example usage of the multicol package. Notice that we use the lipsum package and command to generate placeholder text

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.2.2 Multicol Code

If you need more flexibility in the column layout, or to create a document with multiple columns, the package `multicol` provides a set of commands for that. You will need to import this package using `\usepackage{multicol}` to use this.

```
\begin{multicols}{3}
[
\subsubsection{Multicol Sample}
The following text is an example usage of the multicol
    package. Notice that we use the lipsum package and
    command to generate placeholder text
]
\lipsum[1]
\end{multicols}
```

The environment takes two parameters:

- Number of columns. This parameter must be passed inside braces, and its value is 3 in the example.
- "Header text", which is inserted in between square brackets. This is optional and will be displayed on top of the multicolumn text. Any LaTeX command can be used here, except for floating elements such as figures and tables. In the example, the section title and a small line are set here.

The text enclosed inside the tags `\begin{multicols}` and `\end{multicols}` is printed in multicolumn format.

5.2.3 Additional Parameters

You can set the separation between columns, which is defined by `\columnsep`. You can set it to whatever you want using `\setlength{\columnsep}{1cm}`. This will affect all of the columns below wherever this command is used.

In the default `multicols` environment the columns are balanced so each one contains the same amount of text. This default format can be changed by the stared environment `multicols*`

You can break the content into the next column by using the `\columnbreak` command. This command inserts a column breakpoint. In this case, the behaviour of the text is different from what you may expect. The column break is inserted, then the paragraphs before the breakpoint are evenly distributed to fill all available space. In the example, the second paragraph is at the bottom of the column and a blank space is inserted in between the second and the first paragraphs.

5.2.4 Vertical Rulers

You can add vertical rulers between your columns by:

```
%Part of the preamble
\setlength{\columnseprule}{1pt}
\def\columnseprulecolor{\color{blue}}

\begin{multicols}{3}
[
\section{First Section}
The following text has columns separated by rulers
]
\lipsum[1]
\end{multicols}
```

As you see, the column separator can be set to a specific colour also. Below a description of each command:

- `\setlength{\columnseprule}{1pt}` This determines the width of the ruler to be used as column separator, it's set to 0 by default. In the example a column whose width is 1pt is printed.
- `\def\columnseprulecolor{\color{blue}}` The colour of the separator ruler can be set to any color you want

Document Index

LaTeX, 7
    document, 7
    package, 7
        imakeidx, 7

index, 7
    section, 7

keywords, 7