Packages

Formatting and Positioning Text

Font Sizes

Units of the Document

Spaces

Boxes Around Text
Packages

An Overview

Packages

Loading Packages
Packages in the Lab
Acquiring Packages
Elsewhere
The geometry Package
A geometry example
The setspace Package
Practice

Formatting and Positioning Text

Font Sizes

Units of the Document

Spaces

Boxes Around Text
“Packages” are add-ons to \LaTeX\ that can greatly extend its flexibility and your options. They are called in the preamble of your document with the command

\begin{verbatim}
\usepackage[options]{packagename}
\end{verbatim}

Most packages come with documentation, which tells you how to use the package properly. Find the relevant folder here: 
\texttt{C:\Program Files (x86)\MiKTeX 2.9\doc\latex}

Let’s open the documentation for \texttt{geometry} (\texttt{geometry.pdf}) as an example.
Believe it or not, every single \LaTeX{} package has been loaded onto your computer in this lab!

So, accessing \LaTeX{} packages in the lab should be easy.

It will be different on your own machine.
If you run MiKTeX, the easiest way to get a package onto your computer is to use the “Package Manager” program. (Go to the Start Menu, find the MiKTeX 2.9 folder, then Maintenance, then Package Manager.)

Once you find something there, MiKTeX will grab it and “refresh” your database.

Another way to get a package is to poke around on http://www.ctan.org. This is a central repository for \LaTeX\ packages.

Finally, if you call a package in your preamble and it isn’t on your computer, \LaTeX\ will either download and install it automatically, throw an error, or ask you if you’d like to download and install it. (It all depends upon the options you chose when installing MikTeX on your machine.)
The \textit{geometry} package controls the margins and page layout for your documents. It is \textit{very} useful.

A lot of the commands for the \textit{geometry} package are straightforward. See all of them in the documentation.

These are all examples of options for the \textit{geometry} package.

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>margin=1in</td>
<td>1-inch margins</td>
</tr>
<tr>
<td>top=1.5in</td>
<td>1.5-inch top margin</td>
</tr>
<tr>
<td>height=8in</td>
<td>8 vertical inches of text</td>
</tr>
<tr>
<td>width=4in</td>
<td>4 horizontal inches of text</td>
</tr>
</tbody>
</table>

The \textit{geometry} package is good at arithmetic, so you won’t have to specify everything.
Let’s create a quick file to test out this package.

```latex
\documentclass{article}
\usepackage[margin=1in]{geometry}
\begin{document}
Hello, world!
\end{document}
```

Now play around with the options above. Use `left`, `right`, `top`, `bottom`, or `margin`. To specify more than one option, separate them within the square brackets by commas (ex: `[top=3in,right=3.5cm]`). Remember to include a unit of measurement!
You’ve probably noticed that \LaTeX\ typesets everything single-spaced by default. This is actually quite challenging to change without the aid of the \texttt{setspace} package.

Put \texttt{\usepackage\{setspace\}} in your preamble. Then use the \texttt{\begin{spacing}{}{2}\end{spacing}} environment to change the spacing.

\begin{spacing}{2}
...
\end{spacing}

The mandatory argument is the spacing you desire. Most numerical values are fair game.

\textbf{Note on \texttt{setspace}:} This package is \textit{really} easy to use, so there isn’t much documentation to speak of. It’s commented out in the \texttt{.sty} file.
Let’s practice!
Open up the first example PDF file from Sakai, and reproduce it.
Formatting and Positioning Text

An Overview

Packages

Formatting and Positioning Text

Text Decorations
Commands and Declarations
Text Decorations with soul
Changing colors for soul

Text Justification
Footnotes
Accents and Ellipses
Practice

Font Sizes

Units of the Document
Spaces
Boxes Around Text
Short changes to the way your text looks (italics, bold, etc.) require *commands* in \LaTeX.

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textbf{This is bold}.</td>
<td>This is bold.</td>
</tr>
<tr>
<td>\textit{This is italic}.</td>
<td>This is italic.</td>
</tr>
<tr>
<td>\emph{This is emphasized}.</td>
<td>This is emphasized.</td>
</tr>
<tr>
<td>\underline{Underlined}.</td>
<td>Underlined.</td>
</tr>
<tr>
<td>\textsc{Small caps}.</td>
<td>SMALL CAPS.</td>
</tr>
<tr>
<td>\texttt{Typewriter type}.</td>
<td>Typewriter type.</td>
</tr>
<tr>
<td>\textsl{Slanted shape}.</td>
<td>Slanted shape.</td>
</tr>
</tbody>
</table>

Commands like this can be **nested**.
For text changes that stretch on for longer periods of time, use *declarations* instead of commands.

Instead of $\textbf{Bold text}$ use

$\texttt{\bfseries Bold text}$

or as an environment: $\begin{bfseries}$.

Here is the correspondence:

<table>
<thead>
<tr>
<th>Command</th>
<th>Declaration</th>
<th>Command</th>
<th>Declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textrm</td>
<td>\rmfamily</td>
<td>\textsf</td>
<td>\sffamily</td>
</tr>
<tr>
<td>\texttt</td>
<td>\ttfamily</td>
<td>\textmd</td>
<td>\mdseries</td>
</tr>
<tr>
<td>\textbf</td>
<td>\bfseries</td>
<td>\textup</td>
<td>\upshape</td>
</tr>
<tr>
<td>\textit</td>
<td>\itshape</td>
<td>\textsl</td>
<td>\slshape</td>
</tr>
<tr>
<td>\textsc</td>
<td>\scshape</td>
<td>\emph</td>
<td>\em</td>
</tr>
</tbody>
</table>
You can do other cool things to text by loading the `soul` package. (You will need to load the `color` package too.)

- strikethrough `\st{...}`
  Example: I went to the doctor’s dentist’s office this morning.
- highlighting `\hl{...}`
  Example: Color can be used for emphasis instead of italics.
- underlining `\ul{...}`
  Example: Underlining is not used very frequently for emphasis.
- letterspacing `\so{...}`
  Example: Letterspacing makes some text look funny.
You can change the strikethrough, highlighting, and underline colors.

\setstcolor{red} \hspace{1cm} Makes the strikethrough red.
\sethlcolor{green} \hspace{1cm} Makes the highlight color \textcolor{green}{green}.
\setulcolor{orange} \hspace{1cm} Makes the underline color \textcolor{orange}{orange}.
**Environments** are necessary to change the standard full justification that \LaTeX uses.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>center</td>
<td>Centered text</td>
</tr>
<tr>
<td>flushright</td>
<td>Right justification</td>
</tr>
<tr>
<td>flushleft</td>
<td>Left justification</td>
</tr>
</tbody>
</table>

Example:

```latex
\documentclass{article}
\begin{document}
\begin{center}
Hello, world!
\end{center}
\end{document}
```
\LaTeX{} handles footnotes beautifully.

\begin{document}
\underline{War and Peace} is a great book.\footnote{Actually, I haven’t read it.}
\end{document}
Accents

If you need accented characters, like from a different language, you’ll need some special commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;{a}</td>
<td>ä</td>
</tr>
<tr>
<td>’{e}</td>
<td>é</td>
</tr>
<tr>
<td>^{o}</td>
<td>ô</td>
</tr>
</tbody>
</table>

Consult the guide posted on Sakai for a full list.

Ellipses

\textit{type:} It was the best of times, \ldots\textit{see:} It was the best of times, …
Let’s practice!

Open up the second PDF file posted on Sakai and reproduce it.
Font Sizes
All of \LaTeX's font size adjustments are set based upon the \emph{normal} size for the document.

1. The default text size for a document is 10 point.
2. To make a change, use an optional argument on the \texttt{\documentclass} command.

\begin{verbatim}
\documentclass[12pt]{article}
\end{verbatim}

3. Permitted font sizes for the \texttt{article} class: 9, 10, 11, 12.
From smallest to largest:

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\tiny</td>
</tr>
<tr>
<td>\scriptsize</td>
</tr>
<tr>
<td>\footnotesize</td>
</tr>
<tr>
<td>\small</td>
</tr>
<tr>
<td>\normalsize</td>
</tr>
<tr>
<td>\large</td>
</tr>
<tr>
<td>\Large</td>
</tr>
<tr>
<td>\LARGE</td>
</tr>
<tr>
<td>\huge</td>
</tr>
<tr>
<td>\Huge</td>
</tr>
</tbody>
</table>
An example of changing font sizes within a document.

**type:** These `\large` words `\Large` get `\LARGE` big.

**see:** These words get *big*.

Another example:

**type:** Lots `\huge` of different `\small` sizes `\tiny` of `\normalsize` text.

**see:** Lots *of different* sizes of text.
There are three ways to get a new line of text.

1. Insert a blank line in the `.tex` file. This produces a new paragraph.

2. Use the `\` command. Text after the double backslash will appear on the next line, without any indentation. (Make sure you have a space in the `.tex` file after the `\` to achieve this.)

3. Use `\newline`. This functions the same way as `\`.

Most prefer `\` because of this added functionality:

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\[length]</code></td>
<td>creates interline space of <code>length</code></td>
</tr>
<tr>
<td><code>\*</code></td>
<td>prevents page break after line</td>
</tr>
<tr>
<td><code>\*[length]</code></td>
<td>combination</td>
</tr>
</tbody>
</table>
Type:

I think we should continue this conversation \[.3in\] down here.

See:

I think we should continue this conversation down here.

Acceptable units of measurement: in, cm, pt
We’ve seen that a blank line in your file produces a new paragraph. Sometimes you might not want this paragraph indented. (It is indented by default.)

Use the command `\noindent` before the first line of the next paragraph.

```
\documentclass{article}
\begin{document}
This is indented.
\noindent This is not.
\end{document}
```
This command is exactly what you might expect now:

\newpage.

Everything after this command will appear on the next page. That new paragraph will be indented.
Let’s practice!
Open up the third PDF file posted on Sakai and reproduce it.
Spaces
We’ve learned about \, and \,\, as ways to add horizontal space.

We’ll mainly use the \hspace command:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\quad</td>
<td>\quad</td>
</tr>
<tr>
<td>\qquad</td>
<td>\qquad</td>
</tr>
<tr>
<td>\hspace{8pt}</td>
<td>\hspace{8pt}</td>
</tr>
<tr>
<td>\hspace{.6in}</td>
<td>\hspace{.6in}</td>
</tr>
<tr>
<td>\hspace{1.2cm}</td>
<td>\hspace{1.2cm}</td>
</tr>
</tbody>
</table>

Horizontal space can be negative if needed.
The analog to \hspace is \vspace.

\LaTeX adds vertical space after the typeset line in which that command appears.

This means that \vspace commands should usually be on their own line in the .tex file.
Filling Space

- `\hfill` fills available space in the line with blank space
- `\vfill` fills available space on the page with blank space

Example:

```latex
\documentclass{article}
\begin{document}
\noindent The top. \[
\text{The left \hfill and the right.}\]
\vfill
The bottom.
\end{document}
```
Boxes Around Text
An added text effect can be achieved by framing text.

Like this!

\fbox{Like this!}

**Restriction**: You cannot break a line within \fbox.
The text within a paragraph box works just like a paragraph, except that the author sets the width.

\parbox{2in}{The text within...sets the width.}

You can accomplish the same thing with the \texttt{minipage} environment.

\begin{tabular}{|l|}
\hline
\textbf{Syntax} \\
\hline
\parbox{width}{text} \\
\begin{minipage}{width}text\end{minipage} \\
\hline
\end{tabular}
As you can see, we can combine these techniques together to form a boxed paragraph of any width we choose.

\fbox{\parbox{2in}{As you can... we choose.}}

As you can see, we can combine these techniques together to form a boxed paragraph of any width we choose.

\fbox{\begin{minipage}{2in}
As you can ... we choose.
\end{minipage}}
We can also change the background color for a paragraph box:

Look at this yellow box. Look at this yellow box. Look at this yellow box. Look at this yellow box.

\colorbox{yellow}{\parbox{2in}{...}}

If you want this to happen within a document, you’ll need to use a width of \textwidth within the parbox. (Alternatively, the minipage environment can be used in place of the parbox.)
Let’s practice!

Open up the fourth PDF file posted on Sakai and reproduce it.