An Overview

Lists

Tables
Lists

An Overview

Lists

The itemize environment

The enumerate Environment

The description Environment

Nesting Lists

Customizing itemize

The outlines package

More with outlines

Practice

Tables
\LaTeX{} provides three list environments: \texttt{itemize}, \texttt{enumerate}, and \texttt{description}.

Within each environment, \texttt{\item} gives you the next item. Also, \texttt{\item} must be the \texttt{first} thing in the environment.

Some things to notice about the \texttt{itemize} environment:

- It is a list with bullet points.
- It has no numbers.

\begin{itemize}
\item It is a list with bullet points.
\item It has no numbers.
\end{itemize}
The *enumerate* Environment

An Overview

Lists

The *itemize* environment
The *enumerate* Environment
The *description* Environment
Nesting Lists
Customizing *itemize*

Nestings

The *outlines* package
More with *outlines*
Practice

Tables

Things I like about the *enumerate* environment:

1. It also produces a list.
2. It enumerates the list.

\begin{enumerate}
\item It also produces a list.
\item It enumerates the list.
\end{enumerate}
The description environment is generally used for stating definitions.

**Spanish** The language spoken in Spain.

**French** The language spoken in France.

\begin{description}
\item[Spanish] The language spoken in Spain.
\item[French] The language spoken in France.
\end{description}

*Note*: the optional argument is the whole point here.
It is easy to nest the listing environments.

1. This is the first level in \texttt{enumerate}.
   
   (a) This is the second level.

\begin{enumerate}
\item This is the first level.
  \begin{enumerate}
  \item This is the second level.
  \end{enumerate}
\end{enumerate}

This can be done with \texttt{itemize} as well.
The standard “bullets” that come in the \texttt{itemize} environment are dictated by the document class that you use.

- Here is the first level.
  - Here is the second level.
    - Here is the third level.

These symbols can be changed on a case-by-case basis:

- This item begins with a star.
- This item begins with a dagger.

\texttt{\item[$\star$]} \texttt{\item[$\dagger$]}
Nesting lists is made easy with the `outlines` package. Everything goes inside of an `outline` environment.

\begin{outline}
\1 This is the first level.
\2 Easy to move to level two.
\3 Level three is easy too.
\1 Back to level one.
\end{outline}

- This is the first level.
  - Easy to move to level two.
    - Level three is easy too.
- Back to level one.
By default, the outline environment uses itemize. We can easily make it use enumerate.

\begin{outline}[enumerate]
  \1 This is the first level.
  \2 Easy to move to level two.
  \3 Level three is easy too.
  \1 Back to level one.
\end{outline}

1. This is the first level.
   
   (a) Easy to move to level two.
       
       i. Level three is easy too.

2. Back to level one.
Let’s practice!
Open up the first example PDF file from Sakai, and reproduce it.
Tables

An Overview

Lists

Tables
- Getting Started with Tables
- Column Specifications
- More Control
- Even More Control
- Practice

Justifying Single Cells
- More Column Specifications
- Column Width
- Example
- Intercolumn Command
- Example
- Extra Column Specifiers
- Practice
- Changing the Line Thickness
- The \texttt{multirow} Package
- Practice
- Tables on Multiple Pages
- An Example
A table is generated by the `tabular` environment. \LaTeX\ treats the table as one big symbol.

Consequences:

- Tables (usually) cannot stretch across pages.
- Usually tables should be *displayed*, like in the `center` environment.

The syntax for `tabular`:
```
\begin{tabular}{column specifications}
  table cells, separate columns with &
  separate rows with \\
\end{tabular}
```
```
draw horizontal lines with \hline
\end{tabular}
```
For each column, you type \texttt{l}, \texttt{c}, or \texttt{r}, depending on the justification you want. You can put in lines between columns with \textbackslash{}\texttt{|}.

\begin{center}
\begin{tabular}{|l|r|}
\hline
\texttt{duck} & \texttt{vulture} \\
\hline
\texttt{humming bird} & \texttt{bee} \\
\hline
\end{tabular}
\end{center}

\begin{verbatim}
\begin{center}
\begin{tabular}{|l|r|} \hline
\texttt{duck} & \texttt{vulture} \\
\hline
\texttt{humming bird} & \texttt{bee} \\
\hline
\end{tabular}\end{center}
\end{verbatim}
Instead of \texttt{\hline}, the command \texttt{\cline{1-2}} draws a horizontal line through columns 1 and 2 only (for example).

\begin{center}
\begin{tabular}{|l|r|c|}
\hline
duck & vulture & goose \\
\cline{1-2}
humming bird & bee & eagle \\
\hline
\end{tabular}
\end{center}

\begin{center}
\texttt{\begin{tabular}{|l|r|c|}\hline}
duck & vulture & goose \\
\cline{1-2}
humming bird & bee & eagle \\
\hline
\texttt{\end{tabular}}
\end{center}
The command \texttt{\multicolumn} in tables allows you to stretch an entry across columns.

\begin{tabular}{|l|r|c|}
\hline
duck & vulture & goose \\
\hline
humming bird & & bee \\
\hline
\end{tabular}

\begin{verbatim}
\begin{tabular}{|l|r|c|}\hline
duck & vulture & goose \\
\hline
humming bird & \multicolumn{2}{c|}{bee} \\
\hline
\end{tabular}
\end{verbatim}

Syntax:
\texttt{\multicolumn[#1]{#2}{#3}}

\#1 — number of columns to span
\#2 — alignment of the new super-column
\#3 — text of column
Try to reproduce this table:

<table>
<thead>
<tr>
<th></th>
<th>one</th>
<th>two</th>
<th>three</th>
<th>four</th>
</tr>
</thead>
<tbody>
<tr>
<td>five</td>
<td></td>
<td>six</td>
<td>seven</td>
<td>eight</td>
</tr>
<tr>
<td>nine</td>
<td></td>
<td></td>
<td></td>
<td>wide ten</td>
</tr>
<tr>
<td>longer words</td>
<td>longer words</td>
<td>longer words</td>
<td>longer words</td>
<td></td>
</tr>
</tbody>
</table>
We saw how to put different justifications on different columns within a table (or array). What I didn’t show you was how to change the justification for a *single cell*. This is often used for table headings.

\begin{tabular}{|l|l|} 
\hline
words & words \\
\hline
longer words & longer words \\
\hline
\multicolumn{1}{|c|}{word} & word \\
\hline
\end{tabular}
Within the `tabular` environment, so far we know the column commands `l`, `c`, and `r`. Two more are provided by default.

- `p{size}` — This makes a column of a fixed width. Everything in that column is left justified. Entries in other rows are justified vertically at the top of the cell.
- `@{command}` — This is a way of putting commands between columns.
Note the difference:

```
\begin{tabular}{|l|c|} \hline
This entry goes on and on and on for a while \hline
\end{tabular}
```

```
\begin{tabular}{|p{1.5in}|c|} \hline
This entry goes on and on and on for a while \hline
\end{tabular}
```
The @@.@@ command is especially useful when you want to align along decimal points. Notice:

\begin{tabular}{|c|r@{.}l|} 
\hline
A nice number & 2.5177 \\
A better number & 3.14 \\
A super number & 554.8 \\
\hline
\end{tabular}

\begin{verbatim}
\begin{tabular}{|c|r@{.}l|} \hline
A nice number & 2&5177 \\
A better number & 3&14 \\
A super number & 554&8 \\
\hline
\end{tabular}
\end{verbatim}
If you load the `array` package in your preamble, you get two other column specifiers.

- `m{size}` — The same as `p{size}` except other entries in the row are vertically center justified.
- `b{size}` — The same as `p{size}` except other entries in the row are vertically bottom justified.
Let’s practice!
Open up the second example PDF file from Sakai, and reproduce it.
Sometimes you’d like a thicker line width in your tables (default is 0.4 pt). We can change the `arrayrulewidth` to do this.

```
\setlength{\arrayrulewidth}{2pt}
\begin{tabular}{|l|r|} \hline
left & right \\
left left & right right \\
\hline
\end{tabular}
```

**Note:** This will make a “global” change unless it is inside of some environment. One solution is to put `\{` before `\setlength`... and `\}` after `\end{tabular}`.
With `\usepackage{multirow}` in your preamble, you can have entries that span multiple rows, just the same way that we can have entries that span multiple columns now.

\multirow{nrows}{width}{contents}

- `nrows` number of rows to span
- `width` width of entry, can type *
- `contents` contents of the entry

\begin{tabular}{|l|l|l|}
\hline
one & two & \multirow{2}{*}{three} \\
\cline{1-2}
four & five & \\
\hline
six & seven & eight \\
\end{tabular}

\begin{tabular}{|l|l|l|} \hline  \\
one & two & \multirow{2}{*}{three} \\
\cline{1-2}
four & five & \\
\cline{1-2}
four & five & \hline
\end{tabular}
Try to reproduce this table:

<table>
<thead>
<tr>
<th>span #1</th>
<th>duck</th>
</tr>
</thead>
<tbody>
<tr>
<td>owl</td>
<td>goose</td>
</tr>
<tr>
<td>sparrow</td>
<td>dove</td>
</tr>
<tr>
<td>vulture</td>
<td>finch</td>
</tr>
<tr>
<td></td>
<td>porcupine</td>
</tr>
</tbody>
</table>
We said before that a table could not be split across pages. Without extra packages, that’s true. We can make it happen with the \texttt{supertabular} package.

Since this splits tables across pages, it has a way to specify the \textit{table header} that shows up on every page.

\begin{itemize}
\item \texttt{\tablehead{rows}} — repeated on every page
\item \texttt{\tablefirsthead{rows}} — table header for only first page
\item \texttt{\tabletail{rows}} — repeated on every page
\item \texttt{\tablelasttail{rows}} — table footer for only last page
\end{itemize}
Access the third example file (.tex) on Sakai.

Use the LaTeX => PDF profile.

Build and view.