Tables

To typeset tables you really want the booktabs package, which is incorporated within the memoir class. This package has been designed with the goal of book-quality output—the manual contains an excellent introduction on the technicalities of table layout. First and foremost is the mantra "never use vertical lines in tables". See the booktabs manual for full details.

To illustrate tables, I'm simply going to typeset a bunch of examples, each showing different table features. Tables are generally put in floats, which are dealt with more in the `floats-and-referencing` example document.

Basics

One way to set tables is with the `\begin{tabular}...\end{tabular}` environment. The first thing that must be defined in this environment is the number of columns and how they are aligned. For every column, the *column spec* sets how the column will behave. For example, the spec `{lcr}` would define three columns, the first of which was aligned left, the second is centred and the third is aligned right. See Table 1 for this example. Column widths expand to fit.

description	left	centered	right
column spec	`l`	`c`	`r`
example	left	center	right

Table 1: Basic column alignment

The other type of column formatting is by fixing the width and letting the text reflow in the table cells. If you want to align the text in these columns, you'll have to wait until the end to find out how. Look at Table 2 for a demonstration of these paragraph columns.

description	top	vertical center	bottom
column spec	`b{7em}`	`m{7em}`	`p{7em}`
example	fixed width in which text will wrap.	fixed width in which text will wrap.	fixed width in which text will wrap.

Table 2: Paragraph columns

It is very common to combine cells horizontally for table headings, etc. This is possible with the `multicolumn{<number of cells>}{<column spec>}{<text>}`

command. With the multi-column cell also comes the multi-column rule, `\cmidrule{<start col>-<end col>}`. See Table 3 for how these look.

	Column Type		
description	left	centered	right
spec	`l`	`c`	`r`
example	left	center	right

Table 3: Multi-column cells and rules

Now we know the basics, I'm going to start making the tables look even better. Notice in all the tables up until now that there's been a bit of extra space hanging off the ends? The `@{}` command in the column spec will remove inter-column space and insert whatever's in the brackets. Well, we want to eliminate the end space, so we will just surround our column spec with these: `{@{}llcr@{}}`. This is shown in Table 4. We'll also trim the multi-column rule (`(lr)`: the optional `l` trims the left and the optional `r` trims the right), since at the moment it's a little asymmetric.

	Column Type		
description	left	centered	right
spec	`l`	`c`	`r`
example	left	center	right

Table 4: Trimmed rule edges. Compare with Table 3. Much nicer.

There are some more things we can put into the column spec.[1] `>{}` will append every cell of that column with whatever's in the curlies; `<{}` will prepend. These are useful for specifying column formatting. Have a look in Table 6 to see how I made the left hand column set in an italic font.

	Column Type		
description	left	centered	right
spec	`l`	`c`	`r`
example	left	center	right

Table 5: Column formatting.

Finally, don't you think our column spec is getting a bit cluttered? In Table 6 it looks like `{@{}>{\itshape}llcr@{}}`. Pretty long and bloated for a four column table. It is possible to create our own to clear things up a whole lot. Let's define `x` to stand for trimmed space, `i` to stand for "make the next column italic", and `M` to stand for a centred maths column. Have a look in the

[1] If you're not using memoir, you'll need the array package

source (after this paragraph, but usually it'd be put in the preamble) to see how this is done. Now all we need is `{x M il lcr X}`, which makes a lot more sense.

$\mathcal{MATHS}$	*Italic words*	Column type		
τ	*description*	left	centered	right
ϵ	*spec*	`l`	`c`	`r`
χ	*example*	left	center	right

Table 6: New column types.

This has just scratched the surface of what you can do with tables in the memoir class and LaTeX in general. Now you know enough to at least be able to produce good looking tables under almost all conditions—even if you need to look up some documentation to work out how to print a table over multiple pages.