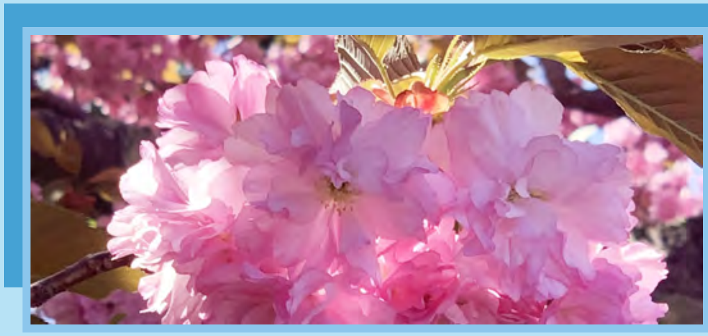

Making Tables

Making Tables

Making tables with LaTeX is not generally difficult, as long as you know how the necessary commands are used. These examples of each of the commands will help you get started.

Standard LaTeX Table

The standard table in LaTeX is made with

```
\begin{tabular}{<preamble>}
<table body>
\end{tabular}
```

We will see the commands that are used in the preamble and those in the body of the table made with **tabular**.

Tabular preamble

Each letter in the preamble signifies two things:

1) How many columns you want in this table, and

2) How are the contents of the columns to be positioned**?**
(left, right, or centered).

Syntax:

```
\begin{tabular}{lcrrrccc}
```

⇑
Table Preamble

- Each letter in the preamble will call for a column. The letter chosen will determine the justification within each column:
  l to the left, **r** to the right, **c** to the center.

(Important! Remember to use the letter l, not the number 1)

- Number of columns is optional, number of lines is optional
- Each new column is started with **&**
- May use less columns than are in preamble, but not more columns
- Each line ends with \\

Table Samples

Notice how the preamble sets the way the contents of the columns are positioned? In the first case, the contents of the first column goes to the left, the second column centers, and the third column moves to the right.

```
\begin{tabular}{lcr}
Here&is our&table\\
a&b&c\\
\end{tabular}
```

Here	is our	table
a	b	c

You don’t need to use every column:

```
\begin{tabular}{lcr}
Here&&table\\
a&b&\\
&&hello
\end{tabular}
```

Here		table
a	b	
		hello

Font changes are contained within the column entry where they are written:

```
\begin{tabular}{lcr}
Here&\it is our&table\\
a&b&c\\
\end{tabular}
```

Here	*is our*	table
a	b	c

All the things you can do in the preamble

Besides using letters to declare a column, we can also call for paragraphs of wrapping text.

To do this we can use the `p{<width>}` expression in the preamble:

```
\begin{tabular}{lp{1.5in}}
Our table&Here we have a place where we
can enter a lot of text and it will wrap
within a space of 1.5in.\\
a&b
\end{tabular}
```

Our table	Here we have a place where we can enter a lot of text and it will wrap within a space of 1.5in.
a	b

Vertical lines

Here we add vertical lines in the preamble, and see that vertical lines appear in the body of the table:

```
\begin{tabular}{|l||cr|}
Here& is our&table\\
a&b&c\\
\end{tabular}
```

Here	is our	table
a	b	c

Inserting characters or space between columns

The `@{}` command used in the preamble will let us either add a something between columns or get rid of space between columns.

When we ask one column to go to the right and the next to the left, notice there is still a space between the two columns:

```
\begin{tabular}{rlc}
Here&is our&table\\
a&b&c
\end{tabular}
```

```
Here  is our  table
   a  b         c
```

To get rid of the unwanted space between columns we can use an `@{}` expression in the preamble.

Now we can get numbers to line up on the decimal point by using two columns for the numbers, with the part of the number preceding the decimal point going in the first column, and the part of the number following the decimal point going in the next column.

```
\begin{tabular}{r@{}lc}
Here&is our&table\\
2130&.456\\
30&.33\\
1&.4568\\
\end{tabular}
```

```
Hereis our  table
2130.456
  30.33
   1.4568
```

Adding text between columns

If you want to, you can add something between the curly brackets. That text will then be added to every table line between the two columns:

```
\begin{tabular}{r@{ Hi! }lc}
Here&is our&table\\
a&b&c\\
\end{tabular}
```

```
Here Hi! is our    table
     a Hi! b           c
```

An author might use the following technique to get a decimal point inserted. Here is the previous decimal table with the decimal points added in the preamble:

```
\begin{tabular}{r@{.}lc}
2130&456\\
30&33\\
1&4568\\
\end{tabular}
```

```
2130.456
  30.33
   1.4568
```

Quick Summary of All Preamble Commands

rlc Letters that show how contents of column should be aligned, left, right, or center.

p{<dimen>} This column should wrap at the width specified. For example, **p{1in}** will make this column wrap at the width of one inch.

| Will make vertical line appear in table between the columns where **|** was used.

@{} Allow user to add something to every line in the table. In addition, it will get rid of space that normally appears between columns, useful if you are lining up decimal points.

You Try It!

Here is a little table that demonstrates each of these preamble commands. Please make a table like this now. You can copy the example to the right or make up your own.

```
\begin{tabular}{|l@{***}lr@{.}l||p{.5in}}
Stars&are found&in 99&9\%\\
of the&universe&&&Isn't that amazing?\\
\end{tabular}
```

Stars ***are found	in 99.9%	
of the***universe	.	Isn't that amazing?

Horizontal lines

`\hline`

Here we add horizontal lines in the table using the **`\hline`** command:

```
\begin{tabular}{lcr}
\hline
 Here& is our&table\\
\hline
\hline
a&b&c\\
\hline
\end{tabular}
```

Here	is our	table
a	b	c

`\cline{}`

If we want horizontal lines to extend under some columns but not others we use
`\cline{<number of first col-number of lastcol>}`
Two column numbers are needed with a dash in between them.

```
\begin{tabular}{lcr}
\hline
Here& is our&table\\
\cline{1-2}
a&b&c\\
\cline{2-3}
\end{tabular}
```

Here	is our	table
a	b	c

A horizontal line may be used under a single column. In that case we use the column number twice, as in `\cline{2-2}`, since cline needs two numbers with a dash between them to work correctly.

```
\begin{tabular}{lcr}
\hline
Here& is our&table\\
\cline{1-2}
a&b&c\\
\cline{2-2}
\end{tabular}
```

Here	is our	table
a	b	c

You Try It!

Make a small table that uses `\hline` and `\cline`.
Here’s an example. (Apologies to Robert Frost)

```
\begin{tabular}{ccc}
\hline\hline
I’m going out& to fetch& the little calf\\
\cline{3-3}
That’s standing by& the mother.&
          It’s so young,\\\cline{1-2}
It totters& when she licks it&
with her tongue.\\
\hline\hline
\end{tabular}
```

I’m going out	to fetch	the little calf
That’s standing by	the mother.	It’s so young,
It totters	when she licks it	with her tongue.

Spanning Columns

When we want to span more than one column with text, we can use the multicolumn command:

```
\multicolumn{<number of columns>}
                    {<position of text>}{<text>}.
```

```
\begin{tabular}{ccc}
\multicolumn{2}{c}{Hello}&table entry\\
One&Two&Three\\
&\multicolumn2c{Hi!}\\
\end{tabular}
```

Hello		table entry
One	Two	Three
	Hi!	

We can also add vertical lines to the multicolumn command in the second argument
ie, instead of `{c}` we might use `{|c|}`

This may be necessary if you have vertical lines in the table preamble, since the `\multicolumn` command ignores table preamble instructions. Here's and example where `\multicolumn` gets rid of the vertical rules that found in the rest of the table:

```
\begin{tabular}{||c|c|c|}
\multicolumn{2}{c}{Hello}&table entry\\
One&Two&Three\\
More Things&Here&You See\\
\end{tabular}
```

Hello		table entry
One	Two	Three
More Things	Here	You See

Making Tables continued...

Here's how to fix that problem. Add vertical lines to the second argument to `\multicolumn`:
`\multicolumn{2}{||c|}...`

```
\begin{tabular}{||c|c|c|}
\multicolumn{2}{||c|}{Hello}&table entry\\
One&Two&Three\\
More Things&Here&You See\\
\end{tabular}
```

Hello		table entry
One	Two	Three
More Things	Here	You See

Since `\multicolumn` ignores the table preamble instructions, sometimes we use this when we don't want to use something in the preamble. We can even use `\multicolumn{1}{}`, only spanning 1 column, in order to get around something in the preamble. Notice in this example how the double vertical lines go away in the bottom line because of using `\multicolumn`.

(Note: `\multicolumn1c` is the same as `\multicolumn{1}{c}`
Curly brackets are only needed for grouping when there is more than a single letter or number following the command.)

```
\begin{tabular}{|r||l|c|}
\multicolumn{1}{r}{Text in Multicolumn}
&\multicolumn1c{e}
&\multicolumn1c{f}\\
Here&is our&table\\
a&b&c\\
\end{tabular}
```

Text in Multicolumn	e	f
Here	is our	table
a	b	c

Making Tables continued...

We can use `\multicolumn` to make a column header which spans both parts of the decimal number:

```
\begin{tabular}{r@{.}l}
\multicolumn2c{\bf Decimal Numbers}\\
2130&456\\
30&33\\
1&4568\\
\end{tabular}
```

```
Decimal Numbers
2130.456
  30.33
   1.4568
```

You Try It!

Please try to make this table. The code is on the following page if you need to peek.

The table includes:

- Vertical lines
- `@{ hello }` in the preamble.
- Making decimal numbers in two adjoining columns; use `@{}` between the two columns so that the two columns don't have a space between them.
- Horizontal lines at the top and bottom of table
- `\cline{1-1}` and `\cline{3-4}`.
- `\multicolumn{}{}{}` command.

A Sample Table		hello	341.143
and now	we use cline	hello	78.99
Finally, Multicolumn	This is in multicolumn		

Congratulations!

That wasn't easy!

(Code for sample table)

```
\begin{tabular}{|l||c|@{ hello }|r@{}l|}
\hline
A Sample Table&&341.&143\\
\cline{1-1}
and now&we use cline&78.&99\\
\cline{3-4}
Finally, Multicolumn&
\multicolumn{3}{c|}{This is in multicolumn}\\
\hline
\end{tabular}
```

More control over table spacing

Adding extra space between lines

If we want extra vertical space between lines we can add a skip after the `\\`; `\\[<dimension>]`, which will add space **beneath** that line.

If you use a negative dimension you will bring the lower line closer to the upper line, which is occassionally useful as well.

```
\begin{tabular}{|l||cr|}
\hline
 Here& is our&table\\[24pt]
\hline
\hline
a&b&c\\
\hline
\end{tabular}
```

Here	is our	table
a	b	c

Using vrule in the table

We may want to have more space above the table line as well as below it.

In that case, we can set a vrule to a specific height and depth, remembering that the height starts from the baseline of the letters and goes up, while the depth starts at the baseline and goes down. `\vrule` is a Plain TEX command. It expects to be followed by one or more of these **keywords** entered in any order: `height`, `width`, and `depth`.

(Notice that **keywords** do not have a backslash before them.)

```
\begin{tabular}{|l||cr|}
\hline
 Here\vrule height 16pt depth 6pt width0pt
                      & is our&table\\
\hline
\hline
a&b&c\\
\hline
\end{tabular}
```

Here	is our	table
a	b	c

Arraystretch

To add more vertical space to all the tables in a document, redefine `\arraystretch`.

The default value is `\def\arraystretch{1.0}`. If we want to add 50% more vertical space between lines in the tables, we could redefine it to be `\def\arraystretch{1.5}`.

This is a table using the default `\def\arraystretch{1}`:

```
\def\arraystretch{1}
\begin{tabular}{lrc}\hline
Here&is our&table\\
a&b&c\\
More&Things&Here\\\hline
\end{tabular}
```

Here	is our	table
a	b	c
More	Things	Here

Compared to `\def\arraystretch{1.5}`:

```
\def\arraystretch{1.5}
\begin{tabular}{lrc}\hline
Here&is our&table\\
a&b&c\\
More&Things&Here\\\hline
\end{tabular}
```

Here	is our	table
a	b	c
More	Things	Here

You can add `\def\arraystretch{<your dimen>}` at the beginning of your document to add extra vertical space to all the tables in the document, usually a good idea since tables tend to be too crowded vertically. I would suggest `\def\arraystretch{1.2}` for most documents.

Change space between columns

The space between columns is set with the `\tabcolsep` dimension. You can make this dimension wider, for better table visibility, or smaller, to try to fit a wider table into the width of a page.

The default width of `\tabcolsep` is 6pt. This space is used twice between columns so that if there is a vertical line between columns, there will be space left on either side of the vertical line, so keep this in mind when you make your new setting.

If you are not sure what the default tabcolsep dimension is for your document, you can add or reduce space by adding or subtracting from the original value, ie:
`\advance\tabcolsep by 6pt` or
`\advance\tabcolsep by -6pt`.

If you want to change the dimension for just one table, you can do it within `\begin{table}...\end{table}`, or surround the change with curly brackets to keep the change local. If you want this change for your whole document, you can set it before `\begin{document}`.

Examples to the right:

Default tabcolsep in a simple table:

```
\begin{tabular}{c|c|c|c}
Here&is a&very&simple table\\
\end{tabular}
```

Here	**is a**	**very**	**simple table**

Adding extra space between columns:

```
\tabcolsep=18pt
\begin{tabular}{c|c|c|c}
Here&is a&very&simple table\\
\end{tabular}
```

Here	**is a**	**very**	**simple table**

Subtracting space between columns:

```
\advance\tabcolsep-4pt
\begin{tabular}{c|c|c|c}
Here&is a&very&simple table\\
\end{tabular}
```

Here	**is a**	**very**	**simple table**

Using noalign between table lines

The `\noalign` command will allow us to temporarily stop the table, enter a space command or even text, and then continue with the table. The skip can either be used to add space or take away space between lines of the table.

```
\begin{tabular}{|l||cr|}
\hline
 Here& is our&table\\
\hline
\noalign{\vspace{-2pt}}
\hline
a&b&c\\
\noalign{\vspace{6pt}
We can even put some text in here.\vspace{6pt}
def&hij&klm\\
\hline
\end{tabular}
```

Here	is our	table
a	b	c

We can even put some text in here.

def	hij	klm

Making table spread out to particular width

If we want the table to spread out to any particular width we use **\begin{tabular*}...\end{tabular*}**
with an additional argument for the width of the table,
and **@{extracolsep\fill}** in the preamble:
\begin{tabular*}{<width>}{@{\extracolsep\fill}....

The **@{\extracolsep\fill}** in the beginning of the preamble is necessary, since it causes the space between columns to expand.

```
\begin{tabular*}{3in}
{@{\extracolsep\fill}|lcr|}
\hline
\bf Here&\it is our&table\\
\hline
\hline
a&b&c\\
\hline
\end{tabular*}
```

Here	*is our*	**table**
a	**b**	**c**

To fit the width of the page, we can use **\textwidth** as the dimension given in the first argument of **\begin{tabular}**:

```
\begin{tabular*}
{\textwidth}{@{\extracolsep\fill}|lcr|}
\hline
\bf Here&\it is our&table\\
\hline\hline
a&b&c\\
\hline
\end{tabular*}
```

Here	*is our*	table
a	b	c

A table inside another table

Finally, we put one table inside another and see that it will work:

```
\begin{tabular}{lcr@{}}
 Here& is our table&
 %%% Here is the table within the larger table:
     \begin{tabular}{|l||cr|@{}}
     \hline
     Here& is our&table\\
     \hline
     \hline
     a&b&c\\
     \end{tabular}
 %%% End of table within table
\\
\hline
\hline
a&b&c\\
\hline
\end{tabular}
```

<table>
<tr><td>Here</td><td>is our table</td><td><table><tr><td>Here</td><td>is our</td><td>table</td></tr><tr><td>a</td><td>b</td><td>c</td></tr></table></td></tr>
<tr><td>a</td><td>b</td><td>c</td></tr>
</table>

This is not entirely a toy example– Putting a table within another table can be helpful when making complex column headers.

You Try It!

Now you do it– Try making a little table and try each of the features demo'ed above—have fun!

Things to try:

1. `\\[<dimen>]`, using a positive or negative dimension.

2. `\vrule height <dimen> depth <dimen> width<dimen>`
   in table body

3. Change the value of `\arraystretch` with
   `\def\arraystretch{<percent of 1>}`, ie,
   `\def\arraystretch{.8}`, or
   `\def\arraystretch{2}` (for a double spaced table).

4. Using `\noalign{}` between lines in a table.

5. Make a table that is as wide as `\textwidth`,
   using`\begin{tabular*}{\textwidth}...\end{tabular*}`;
   or a percentage of textwidth, ie:
   `\begin{tabular*}{.8\textwidth}...\end{tabular*}`.

How to get table columns to line up correctly

There are times when just one symbol or letter might stick out from the column, preventing the rest of the column from aligning correctly.

To fix this, we can use **\rlap{}** to allow something stick out to the right but not be seen by LaTeX in the table makeup; and **\llap{}** to allow something to stick out to the left but not be seen by LaTeX.

Compare these two table results:

Not using llap and rlap:

```
\begin{tabular}{lcr}
Here&\it is our&table\\
one&two&three\\
one&two&three*\\
$^1$one&two&three\\
\end{tabular}
```

Here	*is our*	table
one	two	three
one	two	three*
^{1}one	two	three

Using llap and rlap:

```
\begin{tabular}{lcr}
Here&\it is our&table\\
one&two&three\\
one&two&three\rlap{*}\\
\llap{$^1$}one&two&three\\
\end{tabular}
```

Here	*is our*	table
one	two	three
one	two	three*
^{1}one	two	three

You Try It!

Make a simple table like the one on the left, and see the results. Now, use **\llap{}** and **\rlap{}** to see the improvement.

Floating a table, and giving it a caption

To give the table a caption, and to make it float to the top or bottom of a page, we have the
`\begin{table}...\end{table}` environment.

The LaTeX float environment will place the table where you ask it to (more or less) and the text will automatically continue before and after the table or figure.

It is not necessary to put a tabular table in a floating environment: Using **`\begin{tabular}...\end{tabular}`** will work fine.

The advantage of surrounding your **`tabular`** table with
`\begin{table}...\end{table}`
is the floating capability, and the possibility of having a numbered caption inside of **`\begin{table}...\end{table}`** which is only possible in the floating table environment.

The caption also makes it possible to label and cross reference your table number, so that you can refer to it at other points in your document.

`\begin{table}[ht]` This argument determines what part of the page will be used for the float. We have these choices:

[h] for here

[t] for top of the page

[b] for bottom of the page

[p] for put on its own page

[!] Add this to encourage LaTeX to make an extra effort to position the float where you asked.

[H] Demand that LaTeX position float right at this point. Must call **`\usepackage{float}`** to activate this command.

You can use more than one letter. In case the first one doesn't work, the second one will be used.

Notice the square bracket argument used, showing that the argument is optional.

If no argument is given, the default position is to float the table to the top of the page, or the top of the next page if there is not enough room on the current page.

See the next page for an example table environment.

Example of floating table:

The `\centering` command makes the table center horizontally on the page. In some styles, short captions will also center, but if the caption is longer it will form a paragraph. (This style doesn't use that rule)

```
\begin{table}[ht]
\caption{Table caption here.}
\centering
\begin{tabular}{|l||cr|}
\hline
 Here& is our&table\\
\hline\hline
a&b&c\\
\hline
\end{tabular}
\end{table}
```

Table 1
Table caption here.

Here	is our	table
a	b	c

Cross-Referencing Tables

To cross-reference table numbers, use **`\label{}`** either within the caption, or after the caption. If you enter it before the caption you will get the table number for the previous table since the table counter is only advanced with the caption command.

For example, here is how to label and reference a table number:

```
Here is the table reference: \ref{TestTable}
\begin{table}[ht]
\caption{Table caption here.}
\label{TestTable}
\centering
\begin{tabular}{|l||cr|}
\hline
 Here& is our&table\\
\hline\hline
a&b&c\\
\hline
\end{tabular}
\end{table}
```

Here is the table reference: 2

Table 2
Table caption here.

Here	is our	table
a	b	c

You Try It!

Can you imagine how to make a table that looks like this? This will review a number of tools that we've discussed earlier:

Table 3
Small Table

one[a]	two	three
Very	Small	Table Here[b]

[a] Refs. 19 and 20.

[b] $\kappa, \lambda > 1$.

Here's how the table was made:

```
\begin{table}[ht]
\caption{Small Table}
\begin{tabular}
{@{\vrule height 12pt depth 6pt width0pt}|ccc|}
\hline
one\rlap{$^a$}&two&three\\
\hline
Very&Small&Table Here\rlap{$^b$}\\
\hline
\multicolumn{3}{l}{$^a$Refs.~19 and 20.}\\
\multicolumn{3}{l}{$^b\kappa, \lambda>1$.}
\end{tabular}
\end{table}
```

Now, change the arraystretch and tabcolsep dimensions to see how the appearance changes.

(Hint: The syntax is
`\def\arraystretch{<percent of 1>}`
and `\tabcolsep=<dimen>`
or `\advance\tabcolsep<dimen>`.)

Rotating Tables and Figures

If you have included **`\usepackage{graphicx}`**, you will have the **`\rotatebox`** command available. Start your **`\begin{table}[p]`** or **`\begin{figure}[p]`** environment first, then rotate the contents. Usually you will want these rotated figures or tables to be placed on their own page, so you would use the **`[p]`** option, for 'page'.

Use
`\rotatebox{(angle)}{\vbox{ table or figure }}`;
then end with **`\end{table}`** or **`\end{figure}`**.

```
\begin{table}[p]
   \rotatebox{90}{\vbox{
\caption{This is the table caption.}
\begin{tabular}
{@{\vrule height 10pt depth 2pt width0pt}
ccrrcrcc@{}}
\multicolumn{3}{l}{\bf Parameters}&...
\end{tabular}
   }}%% <<== end{,  end vbox{
\end{table}
```

Table 4
This is the table caption.

Parameters			**Averaged Results**				
n	S^*_{MAX}	t_1	r_1	m_1	t_2	r_2	m_2
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

If you need to position the table or figure up or down on the page, you can add in **`\vskip <num>`** points above the **`\rotatebox`** command. A positive dimension will move the illustration down on the page; a negative dimension, ie. **`\vskip-1in`** will move it up on the page.

To move the table to the left or right on the page, add a **`\hskip`** before the rotated table environment, as seen below. A positive dimension will move the table to the right; a negative dimension to the left.

```
\begin{table}[p]
\vskip-1in %% <== this will move table up
           %%         or down, since it is
           %% used before the box is rotated
%%
\hskip3in %% <== this will move the table
          %% left or right depending
          %%  on whether the dimension
          %% is positive or negative
 \rotatebox{-90}{
\vbox{\caption{This is the table caption.}
\begin{tabular}
{@{\vrule height 10pt depth 2pt width0pt}
ccrrcrcc@{}}
\multicolumn{3}{l}{\bf Parameters}&...
\end{tabular}
   }}
\end{table}
```

Table 5
This is the table caption.

Parameters			**Averaged Results**				
n	S^*_{MAX}	t_1	r_1	m_1	t_2	r_2	m_2
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

Adding Color Screens to Table

To get colored columns, rows, or cells in your table, you must start with these packages:

```
\usepackage{graphicx}
\usepackage{xcolor}
\usepackage{array}
\usepackage{colortbl}
```

Documentation for each package:

Documentation for graphicx.sty:
https://mirror.las.iastate.edu/tex-archive/macros/latex/required/graphics/grfguide.pdf

Documentation for colortabl.sty:
http://tug.ctan.org/tex-archive/macros/latex/contrib/colortbl/colortbl.pdf.

Documentation for xcolor.sty:
https://ctan.math.washington.edu/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf

Documentation for array.sty:
https://mirrors.concertpass.com/tex-archive/macros/latex/required/tools/array.pdf

Column Colors

Enter the color for the column background in the table preamble, using `>{ }` which is defined in the **`array`** package.

`>{ }` is used before the column letter for the column that it will color.

The `\columncolor` command may be followed with a color model in square brackets, with an argument of the percent of that color to be used, (oddly: the higher the percent, the lighter the color); or you may use a color name in curly brackets.

After the `\columncolor{}` declaration, follow with the color for the text. The full declaration will look like this:
`>{\columncolor{<color>}\color{<color>}}c`
repeated for every column that you want to color.

```
\columncolor[(color type)]
  {(percent of white in that color)}
 Or:
\columncolor{color name}
```

```
\begin{tabular}{
>{\columncolor[gray]{.8}\color{blue}}c
>{\columncolor[gray]{.2}\color{white}}c
>{\columncolor{CornflowerBlue}
                        \color{white}}c
>{\columncolor{red}\color{white}}c}
one&two&three&red\\
four&five&six& more red
\end{tabular}
```

one	two	three	red
four	five	six	more red

Choosing which color to use

These colors are generally always available: **`white, black, yellow, green, blue, purple, cyan, magenta`**

If you include **`\usepackage{xcolor}`**, you can use **`red, green, blue, cyan, magenta, yellow, black, gray, white, darkgray, lightgray, brown, lime, olive, orange, pink, purple, teal, violet`**

For a more elaborate color choice you can use the option **`[dvipsnames]`** with xcolor: **`\usepackage[dvipsnames]{xcolor}`**.

We were able to use ‘CornflowerBlue’ in the example on the previous page because we gave the option dvipsnames to xcolor.

You can see a list of the colors available when this option is used on page 38 in the xcolor documentation: https://ctan.math.washington.edu/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf

Row Colors

Row colors are added to the body of the table using **`\rowcolor[<color type>]{<percent of color>}`** or **`\rowcolor{color}`**.

```
\begin{tabular}{lc}
  \rowcolor{Apricot}
  one&two\\
  \rowcolor{Turquoise}
  three&four
  \end{tabular}
```

one	two
three	four

Making Tables continued...

What happens if you use both?

Seeing what happens if we use both row and column coloring– the most recent declaration is used, so the row color is the color we see.

```
\begin{tabular}{>{\columncolor{blue}
                           \color{red}}lc}
  \rowcolor[gray]{.8}\color{white}
   one&two\\
  \rowcolor[gray]{.5}\color{white}
  three&four
  \end{tabular}
```

one	two
three	four

Cell Colors

The command **\cellcolor{(color)}** can be used in any table cell, including **\multicolumn**:

```
\begin{tabular}{lc}
  one&\cellcolor{blue}\color{white} two\\
\multicolumn2c{\cellcolor{red}\color{white}
                        three and four}
\end{tabular}
```

one	two
three and four	

Setting Table Rows for Complete Table

The **`\rowcolors`** command is defined in xcolor.sty. For more information:
https://ctan.math.washington.edu/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf

```
\rowcolor [<commands>]{<row>}{<odd-row color>}{<even-row color>}
\rowcolors* [<commands>]{<row>}{<odd-row color>}{<even-row color>}
```

1. Setting the table rows for the complete table must be done before the table starts.
2. The optional argument **`<commands>`** are either **`\hline`** or **`\noalign{<stuff>}`**.
3. The first argument is the number of the line where you want the coloring to start. This Must be an odd numbered line.
4. Setting the colors in argument 2 and 3 are the same as setting the colors in an individual row, as we saw in the previous pages. Each of the color arguments may also be left empty (= no color).
5. In the starred version, (**`\rowcolors*`**) **`<commands>`** are ignored in rows with inactive rowcolors status, whereas in the non-starred version, **`<commands>`** are applied to every row of the table.
6. Additional commands are **`\showrowcolors`**: The rowcolors status is activated (i.e., use coloring scheme) by default or by using this command.
   `\hiderowcolors`: Row colors are inactivated (i.e., ignore coloring scheme) for a particular line by using **`\hiderowcolors`**.
7. The counter **`\rownum`** may be used within such a table to access the current row number.

Setting Table Rows for Complete Table (continued)

Example, from xcolor documentation

```
\rowcolors[\hline]{3}{green!25}
{yellow!50} \arrayrulecolor{red!75!gray}
\begin{tabular}{ll}
test & row \number\rownum\\
test & row \number\rownum\\
test & row \number\rownum\\
test & row \number\rownum\\
\arrayrulecolor{black}
test & row \number\rownum\\
test & row \number\rownum\\
\rowcolor{blue!25}
test & row \number\rownum\\
test & row \number\rownum\\
\hiderowcolors
test & row \number\rownum\\
test & row \number\rownum\\
\showrowcolors
test & row \number\rownum\\
test & row \number\rownum\\
\end{tabular}
```

Rowcolors without star:

test	row 1
test	row 2
test	row 3
test	row 4
test	row 5
test	row 6
test	row 7
test	row 8
test	row 9
test	row 10
test	row 11
test	row 12

Rowcolors with star:

test	row 1
test	row 2
test	row 3
test	row 4
test	row 5
test	row 6
test	row 7
test	row 8
test	row 9
test	row 10
test	row 11
test	row 12

Colors in Sophisticated Tables

Now let's see how we can use these tools for some sophisticated tables:

Table 6
Growth assumptions for the euro area (in percent)

	2007	3008	2009	2010
European Commission forecast 10/2008	2.7	1.2	0.1	0.9
IMF forecast 11/2008	2.6	1.2	−0.5	n/a
Bruegel scenario		1.2	−0.9	0.4

Source: European Commission, IMF, Bruegel calculations.

(From *European Recovery Programme, Bruegel Policy Brief*)

```
\definecolor{Salmon}{cmyk}{0,0.53,0.38,0}
\definecolor{ltSalmon}{cmyk}
                         {0,0.13,0.095,0}
\begin{table}
\def\arraystretch{2.0}\tabcolsep=6pt
\fcolorbox{black}{ltSalmon}{
\vtop{\normalsize\color{red}
\caption{Growth assumptions for the euro
area (in percent)}\color{black}
%%
\begin{tabular}{lccccc}
\rowcolor{Salmon} &2007&3008&2009&2010\\
\rowcolor{ltSalmon}European Commission
forecast 10/2008&2.7&1.2&0.1&0.9\\
IMF forecast 11/2008&2.6&1.2&
\llap{$-$}0.5&n/a\\
Bruegel scenario &&1.2&
\llap{$-$}0.9&0.4\\[3pt]
\multicolumn5l{\sf
Source: European Commission, IMF,
Bruegel calculations.}\\
\end{tabular}
}} % <<== end vtop, end colorbox
\vskip6pt
(From {\it European Recovery Programme,
Bruegel Policy Brief})
\end{table}
```

You Try It!

Study the code so that you can use some of these tools in one of your tables some time in the future.

Another Table Sample

The author of this table defines two colors, sets the arraystretch for more space between lines, (`\def\arraystretch{1.6}`) and uses `\noalign{\vskip2pt}` between rows in the table to get the white spaces that function like lines, ie, `\rowcolor{ltblue}360i&78\%&\$34 million&38\%&Nov  2018\\ \noalign{\vskip2pt}`

Vendor	Enterprise-level clients	Search marketing revenues	Percent of clients using vendor for paid search and SEO	Date evaluated
360i	78%	$34 million	38%	Nov 2018
iCrossing	73%	$79 million	30%	Nov 2018
IMPAQT	59%	$18 million	41%	Nov 2018
iProspect	65%	$40 million*	40%	Nov 2018
Oneupweb	40%	$13 million*	26%	Nov 2018
Razorfish	77%	$31 million	25%	Nov 2018

*The vendor cannot publicly disclose revenues.
This is Forrester's estimate of the vendor's 2017 search revenue.

Vendor selection criteria

Revenue for the vendor's search business is greater than $10 million.

Percent of the vendor's clients served that are enterprise level is greater than 40%.

Percent of the clients that work with the vendor for both paid search and SEO.

Source: Forrester Research, Inc.

Table MakeOver

We start with a crowded table, which would not be considered ‘professional’ by publishing houses:

```
\begin{tabular}{|c|c|c|} \hline
Index &$J\Delta$ Representation & $J\Phi$ Representation \\ \hline \hline
1     &$d=1,f=1,g=0,h=1$                & $d=1,h=1$      \\  \hline
2     &$d=1,e=1,f=1,g=0,h=1,i=1$        & $d=1,e=1$      \\  \hline
3     &$b=0,d=1,e=1,f=1,g=0,h=1,i=1$    & $b=0$          \\  \hline
\end{tabular}
```

Index	$J\Delta$ Representation	$J\Phi$ Representation
1	$d = 1, f = 1, g = 0, h = 1$	$d = 1, h = 1$
2	$d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$d = 1, e = 1$
3	$b = 0, d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$b = 0$

Add extra horizontal space with **tabular*{<size>}**

```
\begin{tabular*}{\textwidth}{@{\extracolsep\fill}|c|c|c|}\hline
Index &$J\Delta$ Representation & $J\Phi$ Representation \\ \hline \hline
1     &$d=1,f=1,g=0,h=1$                & $d=1,h=1$      \\  \hline
2     &$d=1,e=1,f=1,g=0,h=1,i=1$        & $d=1,e=1$      \\  \hline
3     &$b=0,d=1,e=1,f=1,g=0,h=1,i=1$    & $b=0$          \\  \hline
\end{tabular*}
```

Index	$J\Delta$ Representation	$J\Phi$ Representation
1	$d = 1, f = 1, g = 0, h = 1$	$d = 1, h = 1$
2	$d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$d = 1, e = 1$
3	$b = 0, d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$b = 0$

Table Makeover (continued)

Now, we take out those ugly vertical lines and add more vertical space.
Notice how we got rid of the extra space on the right side of the table
by adding `@{ }` to the end of the preamble?

```
\begin{tabular*}{\textwidth}{@{\vrule height 12pt depth 5pt width0pt
\extracolsep\fill}ccc@{ }}\hline
\bf Index &\bf$J\Delta$ Representation
&\bf$J\Phi$ Representation \\ \hline
1     &$d=1,f=1,g=0,h=1$                & $d=1,h=1$      \\
2     &$d=1,e=1,f=1,g=0,h=1,i=1$        & $d=1,e=1$      \\
3     &$b=0,d=1,e=1,f=1,g=0,h=1,i=1$    & $b=0$          \\  \hline
\end{tabular*}
```

Index	$J\Delta$ **Representation**	$J\Phi$ **Representation**
1	$d=1, f=1, g=0, h=1$	$d=1, h=1$
2	$d=1, e=1, f=1, g=0, h=1, i=1$	$d=1, e=1$
3	$b=0, d=1, e=1, f=1, g=0, h=1, i=1$	$b=0$

(***Doesn't that look better?***)

Second table to improve:

One problem with the table to the right is that the column headers are not centered above the columns. The asterisk also mispositions one column entry. **Improvements to make:**

1. Add extra vertical space to only the column header line, and a different amount of space for the other lines in the table.
2. Change the preamble so that the column headers center over the columns. (From `rrrrrrrr` to `ccrrcrcc`).
3. Surround the asterisk with `\rlap{}` to keep the column formatting correctly.
4. Use `\multicolumn1c{}` for the fourth header to center it above a column that is set to go to the right.

Code for the improved version:

```
\begin{tabular*}{\hsize}{@{\extracolsep{\fill}}
ccrrcrcc@{\vrule height 11pt
depth 4pt width0pt}}
\multicolumn{3}{l}{\bf Parameters}&
\multicolumn{5}{c}{\bf Averaged Results}\\
\noalign{\vskip4pt}\hline
\vrule height 11pt depth 6pt width0pt
$n$&$S^*_{MAX}$&$t_1$&
\multicolumn1c{$r_1$}&$m_1$&$t_2$&
$r_2$&$m_2$\\\hline
10&1\quad &4&235.0007&4&4&.0020&4\\
10&5\quad &50&.0008&8&50&.0020\rlap{*}
&12\\\hline
\end{tabular*}
```

The Original version

```
\begin{tabular}{rrrrrrrr}
\multicolumn{3}{l}{\bf Parameters}&
\multicolumn{5}{c}{\bf Averaged Results}\cr
\hline
$n$&$S^*_{MAX}$&$t_1$&$r_1$&$m_1$&$t_2$
&$r_2$&$m_2$\cr
\hline
10&1\quad &4&235.0007&4&4&.0020&4\cr
10&5\quad &50&.0008&8&50&.0020*&12\cr
\hline
\end{tabular}
```

Parameters			**Averaged Results**				
n	S^*_{MAX}	t_1	r_1	m_1	t_2	r_2	m_2
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

The Improved version:

Parameters			**Averaged Results**				
n	S^*_{MAX}	t_1	r_1	m_1	t_2	r_2	m_2
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

Table Continuing over Pages

Tables that continue over pages use the longtable.sty file. Access the commands with `\usepackage{longtable}`. Documentation here, where you can find many additional tools for working with long tables. The basics for using `\longtable` are found below.

These are the commands that must be used:

1. `\begin{longtable}{<preamble>}`
2. text...`\endfirsthead`
   This will make the column headers that are used on the first page of the table.
3. text...`\endhead`
   will make the text be used for column heads for the second and following pages.
4. text...`\endfoot` will be used for the bottom of every page of the table except for the last page of the table.
5. text...`\endlastfoot` will be used for the bottom of the last page of the table.
6. Now you can type in your table contents. End your table with `\end{longtable}`.

You Try It!

This is a good moment to try this out! Please use the commands listed above and make a sample long table that will continue over at least 2 pages.

```
\begin{longtable}{@{}ccc@{}} %<<<
\caption{ApJ costs from 1991 to 2013
\label{tab:table}} \\[2pt]
\hline
\bf Year &\bf Subscription &\bf Publication\\
 & \bf cost &\bf charges\\
 & \bf(\$) & \bf (\$/page)\\
\hline
\endfirsthead  %<<<
\multicolumn3{c}{{\bf Table \thetable}\
(continued)}\\
\hline
\bf Year &\bf Subscription &\bf Publication\\
 & \bf cost &\bf charges\\
 & \bf(\$) & \bf (\$/page)\\
\hline
\endhead  %<<<
\\[12pt]
\endfoot
\hline
\\[24pt]
\endlastfoot %<<<
1991 & 600 & 100 \\
1992 & 650 & 105 \\
...
\end{longtable} %<<<
```

Where to go for more information on Tables

For more things you can do with table specifications:
Wikibooks on LaTeX Tables

For a list of additional table related style files:
More table features

Overleaf Instructions on Tables:
Overleaf

Want a video to explain how to make a table?
Video on making LaTeX tables

More table examples:
Table Examples