A Small List of Built-In LaTeX Commands

Part 1: general symbols

Remember that < and > must be enclosed in dollar signs or they will give ¡ and ¿. Likewise use dollar signs to turn - into −. To negate symbols, precede them with `$\not$`; e.g., `$\not\sim$` produces $\not\sim$. Superscript is given by `$^{}$` and subscript by `$_{}$`; these may be combined: `$X^{A_s}_{B^2_4}$` for $X^{A_s}_{B^2_4}$.

$\leq$	`$\leq$`	$\geq$	`$\geq$`	$\equiv$	`$\equiv$`	$\models$	`$\models$`	$\vdash$	`$\vdash$`
$\subseteq$	`$\subseteq$`	$\supseteq$	`$\supseteq$`	$\subset$	`$\subset$`	$\supset$	`$\supset$`	$\in$	`$\in$`
$\backslash$	`$\backslash$`	$\neq$	`$\neq$`	$\notin$	`$\notin$`	$\vee$	`$\vee$`	$\wedge$	`$\wedge$`
$\forall$	`$\forall$`	$\exists$	`$\exists$`	$\emptyset$	`$\emptyset$`	$\neg$	`$\neg$`	∞	`$\infty$`
ℓ	`$\ell$`	$\approx$	`$\approx$`	$\sim$	`$\sim$`	$\simeq$	`$\simeq$`	$\cong$	`$\cong$`
$\parallel$	`$\parallel$`	$\top$	`$\top$`	$\bot$	`$\bot$, $\perp$`	$\S$	`$\S$`	∂	`$\partial$`
$\angle$	`$\angle$`	$\triangle$	`$\triangle$`	$\ldots$	`$\ldots$`	$\cdots$	`$\cdots$`	$\cdot$	`$\cdot$`
$\vdots$	`$\vdots$`	$\ddots$	`$\ddots$`	$\star$	`$\star$`	$\bullet$	`$\bullet$`	$\circ$	`$\circ$`
$\pm$	`$\pm$`	$\mp$	`$\mp$`	$\cup$	`$\cup$`	$\cap$	`$\cap$`	$\times$	`$\times$`
$\ll$	`$\ll$`	$\gg$	`$\gg$`	$\odot$	`$\odot$`	$\oplus$	`$\oplus$`	$\div$	`$\div$`
$\diamond$	`$\diamond$`	¶	`$\P$`	©	`$\copyright$`	$\dag$	`$\dag$`	$\ddag$	`$\ddag$`

Greek letters are made by their names, with the first letter capitalized for a capital Greek letter: ϕ (`$\phi$`), Φ (`$\Phi$`). Exceptions are letters which exist in English, such as A. Also note the commonly-used φ (`$\varphi$`). For "blackboard bold" letters, use `\mathbb{}`, as in $\mathbb{N}$ from `$\mathbb{N}$`. For cursive-type letters, use `\mathcal{}` (math calligraphic), as in $\mathcal{A}$ (`$\mathcal{A}$`).

Part 2: arrows

$\leftarrow$	`$\leftarrow$`	$\rightarrow$	`$\rightarrow$`	$\leftrightarrow$	`$\leftrightarrow$`
$\uparrow$	`$\uparrow$`	$\downarrow$	`$\downarrow$`	$\updownarrow$	`$\updownarrow$`
$\mapsto$	`$\mapsto$`	$\hookleftarrow$	`$\hookleftarrow$`	$\hookrightarrow$	`$\hookrightarrow$`

The first two rows of arrows may be made double-shafted by capitalizing the first letter: `$\Leftarrow$` produces $\Leftarrow$. The first row and "mapsto" have longer versions made by preceding the command by "long": `$\longleftarrow$` produces $\longleftarrow$. And finally, you may combine those by capitalizing the L of long: `$\Longleftarrow$` produces $\Longleftarrow$.

Part 3: brackets and multi-size symbols

$\langle$	`$\langle$`	$\rangle$	`$\rangle$`	$\{$	`$\{$`	$\}$	`$\}$`	[	`[`	]	`]`
$\lceil$	`$\lceil$`	$\rceil$	`$\rceil$`	$\lfloor$	`$\lfloor$`	$\rfloor$	`$\rfloor$`	(	`(`	)	`)`

Depending on whether you are in in-line math mode (`$ ... $`) or display math mode (`\[ ... \]` or `$\displaymath{...}$`), the following symbols adjust to be appropriately sized (display shown first).

$\displaystyle\sum$	$\sum$	`$\sum$`	$\displaystyle\prod$	$\prod$	`$\prod$`	$\displaystyle\bigoplus$	$\bigoplus$	`$\bigoplus$` (also dot)
$\displaystyle\int$	$\int$	`$\int$`	$\displaystyle\bigcup$	$\bigcup$	`$\bigcup$` (also cap)	$\displaystyle\bigvee$	$\bigvee$	`$\bigvee$` (also wedge)

Note that superscript and subscript bounds of integration or summation appear above and below the symbol only in display style. To make brackets and other matched pairs size themselves appropriately for the height of what they are bracketing, use `$\left$` and `$\right$`, as in `$\left[ ... \right]$`. Every left must have a matching right; if you wish only one half of a pair follow the other with a dot, as in `$\left\{ ... \right.$` (used for piecewise-defined functions, for example).

Part 4: "lids" for symbols

Many things can go on top of symbols and symbol groups. If you want to cap off an i or j, use `$\imath$` ($\imath$) or `$\jmath$` ($\jmath$) to "undot" first.

$\bar m$	`$\bar m$`	$\overline m$	`$\overline m$`	$\underline m$	`$\underline m$`	$\vec m$	`$\vec m$`
$\tilde m$	`$\tilde m$`	$\widetilde m$	`$\widetilde m$`	$\hat m$	`$\hat m$`	$\widehat m$	`$\widehat m$`
m̌	`\v m`	ḿ	`\'m`	m̀	``\`m``	m̈	`\"m`
ṁ	`\.m`	$\dot m$	`$\dot m$`	$\ddot m$	`$\ddot m$`	$\dddot m$	`$\dddot m$`

Note the difference in math mode. Also, there needs to be a space between an alphanumeric command and the character it is modifying, but not if the command is purely punctuation. To, say, underline a group of symbols, enclose them in braces: `$\underline{abc}$` produces $\underline{abc}$, but `$\underline abc$` produces $\underline abc$. Where applicable, use the "wide" option for groups of symbols.

Part 5: spacing

You can manually change the spacing between symbols, though usually it's better to let the compiler do it. To pull things closer together, `\negthinspace` (shortcut: `\!`), `\negmedspace`, and `\negthickspace` are available. To push things apart, remove the "neg" from the commands above; shortcuts are, respectively, `\,`, `\:`, and `\;`. Larger spaces may be made by `\enskip`, `\quad`, and `\qquad`, in order of increasing size. For space between lines of text, try `\smallskip`, `\medskip`, and `\bigskip`. To force a line break, insert `\\`.

When you need a particular amount of horizontal or vertical space use `\hspace{}` and `\vspace{}`, respectively. In the brackets you put the amount of space you want, which may be measured in the relative units `ex` or `em` (the height of a lowercase x or width of a capital M in the current font) or in absolute units including `in` and `cm`; e.g., `\hspace{.5in}`. The amount may be negative.

Two useful spacing commands: `\hfill` and `\vfill`. The former right-justifies the rest of the line, the latter bottom-justifies the rest of the page. Multiple uses of `\vfill` will evenly space paragraphs on the page. A `\vfill` with nothing following it will not insert space, so to space paragraphs such that there are equal amounts of space *following* each one, insert something invisible under the last `\vfill` such as `\mbox{}`.

Finally, `\phantom{}`. Put text in the braces to make a blank space the width of that typeset text. Useful for justification. This, `\hspace{}`, and `\hfill` may be used inside commands like `\underline{}`.

Part 6: font styles and justification

To italicize or boldface text, enclose it in curly brackets and precede it with `\emph` or `\textbf`, as in `\emph{this is slanty}`: *this is slanty.* You may see, instead, `{\em italic text}` or `{\bf bold text}`, but avoid those; they mess up spacing.

To center text, use `\begin{center} ... \end{center}`. To change its size, precede it with a command of the following sort, in order from smallest font size to largest: `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`. To change it back use `\normalsize`, though this is not necessary if your size modifier was inside some `\begin...\end` environment.

Part 7: lists and tables

Lists in LaTeX are made by `\begin{}...\end{}`, with the appropriate command in the braces. Use `enumerate` for numbered lists, `itemize` for bullet-point lists, and `description` for dictionary-entry-style lists. Individual list items are headed with `\item`. Lists may be nested.

Example (note the placement of the header for the `description` entry):

```
\begin{description}
\item[Entry] has a description here.
\begin{enumerate}
\item first!
\item second!
\end{enumerate}
\end{description}
```

Entry: has a description here.

(1) first!

(2) second!

Whether there is a colon after the description header depends on the document class you are using.

You may change the default bullet for `itemize` lists by using `\item[desired symbol]`.

Tables are made with `\begin{tabular}{} ... \end{tabular}`. Inside that empty pair of braces you put letters `l`, `r`, and `c` to create columns which are left-, right-, or center-justified, respectively. The number of letters gives the number of columns and they are justified according to the choice of letter, read left to right. Between the `\begin` and `\end` the table entries are listed left to right, top to bottom, with `&` separating entries on the same line and `\\` ending lines. There is no need to put `\\` at the end of the final line.

A note on something outside the scope of this brief note: while list entries will linebreak themselves appropriately, table entries will not linebreak without, say, making them into minipages.

Part 8: the rest

Footnotes are made by `\footnote{Footnote text.}`. To get math operators like min, lim, sin to typeset nicely in math mode, precede them with a slash (this works only for built-ins, but they are numerous). The `\noindent` command does exactly what it should.