A Summary of Common TEX Control Sequences

Daniel J. Plonsey
Department of Astronomy
University of California, Berkeley
Berkeley, CA 94720

This document lists many of the most commonly used TEX control sequences, and may be useful as a sort of glossary and reference for beginning and intermediate TEX users. However, it is not intended to serve as an *introduction* to TEX; but as far as I know, there is no such thing. *The TEXBook* (Donald Knuth) itself is generally regarded as long, self-indulgent, and poorly organized... all of which is true, but if you have the time, and if you have some programming background, you might find it both enjoyable and worthwhile. Arthur L. Samuel's *First Grade TEX* is much shorter (34 pages), but there are significant omissions. Perhaps a good approach to learning TEX would be to read *First Grade TEX*, then perhaps chapters 16–19, 20, and 22 of *The TEXBook* (mathematics, macro definition, and alignment (tables), respectively). Then try to create a few TEX documents, using this "Summary" to locate the control sequences you need. Finally, turn to the index of *The TEXBook* to learn more about the design and use of various macros as necessary.

1 Control Sequences for Characters in Text

1.1 Special Characters

The following characters are reserved for special purposes in TEX:

- \ Signifies a TEX control sequence.
- { } Delimiters for a section of text to be treated as a unit.
- $ Delimiter for TEX math mode. $ appears at beginning and end of mathematical text.
- $$ Delimiter for TEX math display mode. Creates a separate paragraph for the enclosed mathematical text.
- # Used for passing parameters when defining new control sequences.
- % Comment character. All text on a line following a % is ignored.
- & Alignment character. Used in tables to indicate column separation.
- _ Used to indicate that the character immediately following is a subscript.
- ^ Used to indicate that the character immediately following is a superscript.
- ~ Tilde. Used as an "unbreakable" space—the line cannot be broken between words separated by a tilde.

The above reserved characters can be printed as follows:

To print	\	{}	$	#	%	&	_	^	~
You type	`$\backslash$`	`$\{ \}$`	`\$`	`\#`	`\%`	`\&`	`\_`	`\^{}`	`\~{}`

1.2 Dashes and Quotes

There are four lengths of dashes, and two kinds of quotes available in TEX:

Name	Hyphen	En-dash	Em-dash	Minus sign	Single quotes	Double quotes
To Print	-	–	—	−	‘text’	“text”
You type	`-`	`--`	`---`	`$-$`	`` `text' ``	``` ``text'' ```

The hyphen is used in words like ‘em-dash,’ an ‘en-dash’ is used in a page or date range (e.g., 26–30), and ‘em-dashes’ are used as text separators—usually without spaces on either side.

The important thing to notice about typing quotes is to use the special left-quote key (next to the ‘1’ on my keyboard) for left quotes, rather than the typewriter double quote key.

1.3 Accents, Special Letters, Ligatures, and a Few More

Type	*to get*	
`` \`o ``	ò	(grave accent)
`\'o`	ó	(acute accent)
`\^o`	ô	(circumflex or "hat")
`\"o`	ö	(umlaut or dieresis)
`\~o`	õ	(tilde or "squiggle")
`\=o`	ō	(macron or "bar")
`\.o`	ȯ	(dot accent)
`\u o`	ŏ	(breve accent)
`\v o`	ǒ	(háček or "check")
`\H o`	ő	(long Hungarian umlaut)
`\t oo`	o͡o	(tie-after accent)
`\c o`	o̧	(cedilla accent)
`\d o`	ọ	(dot-under accent)
`\b o`	o̱	(bar under accent)
`\oe, \OE`	œ, Œ	(French ligature OE)
`\ae, \AE`	æ, Æ	(Latin and Scandinavian ligature AE)
`\aa, \AA`	å, Å	(Scandinavian A-with-circle)
`\o, \O`	ø, Ø	(Scandinavian O-with-slash)
`\l, \L`	ł, Ł	(Polish suppresses-L)
`\ss`	ß	(German "es-zet" or sharp S)
`\dag`	†	(dagger or obelisk)
`\ddag`	‡	(double dagger or diesis)
`\S`	§	(section number sign)
`\P`	¶	(paragraph sign or pilcrow)

To put accents over the letters 'i' and 'j,' use the dotless 'ı' and

'ȷ,' obtained by typing `\i` and `\i`. For example, type 'na\"\i ve' to get 'naïve.'

Note: The typewriter font (`\tt`) doesn't allow all the same accents as the other fonts. See *The TEXBook*, page 53, and Appendix F.

2 Control Sequences for Characters in Math Mode

The following symbols must all be used within math mode. To print them within text mode, switch to math mode by enclosing the symbols within dollar-signs.

2.1 Lowercase Greek Letters

α	`\alpha`	ι	`\iota`	ϱ	`\varrho`
β	`\beta`	κ	`\kappa`	σ	`\sigma`
γ	`\gamma`	λ	`\lambda`	ς	`\varsigma`
δ	`\delta`	μ	`\mu`	τ	`\tau`
ϵ	`\epsilon`	ν	`\nu`	υ	`\upsilon`
ε	`\varepsilon`	ξ	`\xi`	ϕ	`\phi`

ζ	\zeta	o	o	φ	\varphi
η	\eta	π	\pi	χ	\chi
θ	\theta	ϖ	\varpi	ψ	\psi
ϑ	\vartheta	ρ	\rho	ω	\omega

2.2 Uppercase Greek letters.

Γ	\Gamma	Ξ	\Xi	Φ	\Phi
Δ	\Delta	Π	\Pi	Ψ	\Psi
Θ	\Theta	Σ	\Sigma	Ω	\Omega
Λ	\Lambda	Υ	\Upsilon		

To get *slanted* Greek letters $\mathit{\Gamma}\mathit{\Delta}\gamma\delta$, type

```
${\mit \Gamma \Delta \gamma \delta}$
```

2.3 Calligraphic Capitals

To get the calligraphic capitals $\mathcal{ABCDE}\ldots\mathcal{Z}$, type

```
${\cal A B C D E} \ldots {\cal Z}$
```

2.4 Miscellaneous Symbols of type Ord

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists
$\imath$	\imath	∇	\nabla	$\neg$	\neg
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat
ℓ	\ell	$\top$	\top	$\natural$	\natural
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp
$\Re$	\Re	$\Vert$	`|`	$\clubsuit$	\clubsuit
$\Im$	\Im	$\angle$	\angle	$\diamondsuit$	\diamondsuit
∂	\partial	$\triangle$	\triangle	$\heartsuit$	\heartsuit
∞	\infty	$\backslash$	\backslash	$\spadesuit$	\spadesuit

2.5 Digits

To get italic digits *0123456789*, type {\it0123456789}.
To get boldface digits **0123456789**, type {\bf0123456789}.
To get "oldstyle" digits 0123456789, type {\oldstyle0123456789}.
These conventions work both inside and outside of math mode.

2.6 "Large" operators

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot
$\prod$	\prod	$\bigcup$	\bigcup	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\bigoplus$	\bigoplus
$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge		

2.7 Binary operations

Besides $+$ and $-$, you can type

$\pm$	`\pm`	$\cap$	`\cap`	$\vee$	`\vee`
$\mp$	`\mp`	$\cup$	`\cup`	$\wedge$	`\wedge`
$\setminus$	`\setminus`	$\uplus$	`\uplus`	$\oplus$	`\oplus`
$\cdot$	`\cdot`	$\sqcap$	`\sqcap`	$\ominus$	`\ominus`
$\times$	`\times`	$\sqcup$	`\sqcup`	$\otimes$	`\otimes`
$\ast$	`\ast`	$\triangleleft$	`\triangleleft`	$\oslash$	`\oslash`
$\star$	`\star`	$\triangleright$	`\triangleright`	$\odot$	`\odot`
$\diamond$	`\diamond`	$\wr$	`\wr`	$\dagger$	`\dagger`
$\circ$	`\circ`	$\bigcirc$	`\bigcirc`	$\ddagger$	`\ddagger`
$\bullet$	`\bullet`	$\bigtriangleup$	`\bigtriangleup`	$\amalg$	`\amalg`
$\div$	`\div`	$\bigtriangledown$	`\bigtriangledown`		

2.8 Relations

Besides $<$, $>$ and $=$, you can type

$\leq$	`\leq`	$\geq$	`\geq`	$\equiv$	`\equiv`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\ll$	`\ll`	$\gg$	`\gg`	$\asymp$	`\asymp`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\smile$	`\smile`	$\mid$	`\mid`	$\doteq$	`\doteq`
$\frown$	`\frown`	$\parallel$	`\parallel`	$\perp$	`\perp`

2.9 Negated relations

Many of the relations listed above can be negated or "crossed out" by prefixing them with \not, as follows:

$\not<$	`\not<`	$\not>$	`\not>`	$\not=$	`\not=`
$\not\leq$	`\not\leq`	$\not\geq$	`\not\geq`	$\not\equiv$	`\not\equiv`
$\not\prec$	`\not\prec`	$\not\succ$	`\not\succ`	$\not\sim$	`\not\sim`
$\not\preceq$	`\not\preceq`	$\not\succeq$	`\not\succeq`	$\not\simeq$	`\not\si meq`
$\not\ll$	`\not\ll`	$\not\gg$	`\not\gg`	$\not\asymp$	`\not\asymp`
$\not\subset$	`\not\subset`	$\not\supset$	`\not\supset`	$\not\approx$	`\not\a pprox`
$\not\subseteq$	`\not\subseteq`	$\not\supseteq$	`\not\supseteq`	$\not\cong$	`\not\cong`
$\not\sqsubseteq$	`\not\sqsubseteq`	$\not\sqsupseteq$	`\not\sqsupseteq`		

2.10 Arrows

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\upar row`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Upar row`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`

$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarro w`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarro w`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`				

2.11 Openings

.

The following left delimiters are available, besides '(':

$[$	`\lbrack`	$\lfloor$	`\lfloor`	$\lceil$	`\lceil`
$\{$	`\lbrace`	$\langle$	`\langle`		

2.12 Closings

The corresponding right delimiters are:

$]$	`\rbrack`	$\rfloor$	`\rfloor`	$\rceil$	`\rceil`
$\}$	`\rbrace`	$\rangle$	`\rangle`		

2.13 Punctuation

TEX puts a thin space after commas and semicolons in math mode, and it does the same for a colon typed as `\colon`. Otherwise, colons are considered to be *relations* and thus get equal space before and after.

To get $x_1 + x_2 + \cdots + x_n$, type `$x_1 + x_2 + \cdots + x_n$`.

To get $x_1 x_2 \ldots x_n$, type `$x_1 x_2\ldots x_n$`.

2.14 Math Accents

Use the following control sequences to produce accents within math mode:

$\hat{a}$	`\hat a`	$\check{a}$	`\check a`	$\tilde{a}$	`\tilde a`
$\acute{a}$	`\acute a`	$\grave{a}$	`\grave a`	$\dot{a}$	`\dot a`
$\ddot{a}$	`\ddot a`	$\breve{a}$	`\breve a`	$\bar{a}$	`\bar a`
$\vec{a}$	`\vec a`				

The dotless letters `\imath` and `\jmath` should be used when i and j are accented; for example `$\hat\imath$` yields $\hat\imath$.

2.15 Bold Math Symbols

To get **bold** math symbols, use Donald Knuth's "poor man's bold" macro which typesets the symbol in three slightly different places (*The TEXBook*, p. 386):

```
\def\pmb#1{\setbox0=\hbox{#1}%
\kern-.025em\copy0\kern-\wd0
\kern.05em\copy0\kern-\wd0
\kern-.025em\raise.0433em\box0 }
```

Then typing '`\pmb{$\infty$}`' yields '$\boldsymbol{\infty}$.'

2.16 Defining Your Own Mathematical Symbols

You may find it necessary to define symbols that are not included in the above tables. For example, the symbol '$\lesssim$' can be defined as a combination of TEX symbols $<$ and $\sim$ as follows:

```
\def\ltsima{$\; \buildrel < \over \sim \;$}
\def\simlt{\lower.5ex\hbox{\ltsima}}
```

Then if you type `$A \simlt B$`, the result will be $A \lesssim B$.

3 Font Style and Size

3.1 Available Font Styles

Plain TEX automatically loads the following text fonts:

Name	Control Sequences	Font Pixel File Name(s)
Roman	`\rm \tenrm \sevenrm \fiverm`	CMR10, CMR9, CMR8, CMR7, CMR6 , CMR5
Bold	`\bf \tenbf \sevenbf \fivebf`	CMBX10, CMBX9, CMBX8, CMBX7, CMBX6, CMBX5
Italics	`\it \tenit`	CMTI10, CMTI9, CMTI8, CMTI7
Slanted	`\sl \tensl`	CMSL10, CMSL9, CMSL8
`Typewriter`	`\tt \tentt`	CMTT10, CMTT9, CMTT8

To switch from the default 10 point Roman to any of the other above fonts, use the appropriate control sequence. If the control sequence appears within braces, TEX will switch back to 10 point Roman immediately after the closing (right) brace. Note that for each of these five font styles, the two-letter control sequence is equivalent to the control sequence for 10 point type (`\bf`, for instance, is the same as `\tenbf`). Also note that while these some of these fonts are also available in 9, 8, and 6 point versions, these must be loaded just as the "non-standard" fonts listed in the following table.

3.2 Non-standard TEX Fonts

To use any of the fonts in the following table, you must use the `\font` control sequence to load the font and assign it a name. For example:

```
\font\examplefont=CMDUNH10
{\examplefont an example of Computer Modern Dunhill 10 point type}.
```

produces an example of Computer Modern Dunhill 10 point type. The `\font` command should be placed near the top of your file, before the actual text begins.

Name	Font Pixel File Name
Bold-Slanted	CMBXSL10
Dunhill	CMDUNH10
Small Caps	CMCSC10
San-Serif 1 inch	AMINCH
Slanted Typewriter	CMSLTT10
San-Serif	CMSS10
San-Serif Bold	CMSSBX10
San-Serif Italics	CMSSI10
San-Serif Medium Condensed	CMSSMC10
San-Serif medium Condensed 40 pt	CMSSMC40
San-Serif Something-or-other 8 pt	CMSSQ8
San-Serif Something-or-other Italic 8 pt	CMSSQI8
TeX Typewriter	CMTEX10
Unslanted Text Italics	CMU10

3.3 Available Font Sizes

To magnify one of the above fonts, you must specify a *scale factor* when loading the font. A scale factor of 1000 produces the normal, unmagnified version of the font. Thus, a scale factor of 1200 magnifies the font by 1.2. For example:

```
\font\biggertenrm=CMR10 scaled 1200
```

will load 10 point roman type magnified by 1.2. This is the equivalent of

```
\font\biggertenrm=CMR10 at 12pt
```

TEX has a limited number of font *sizes*. On our VAX/VMS system, each valid size is a sub-directory of the directory `[TEX.FONTS]`, e.g. `[TEX.FONTS.1500]`. To list all available sizes, do a directory of `[TEX.FONTS]`. Other systems may have a corresponding setup.

To determine what *magnifications* are valid, use the formula

$$\text{size} = \text{magnification} * (\text{resolution}/200)$$

Our Imagen 8/300 printer has *resolution* = 300 pixels/inch, so all possible magnifications can be found by the equation

$$\text{magnification} = \text{size} * 2/3$$

Plain TEX provides the following control sequences as abbreviations for the scaling factor:

ABBREVIATION	EQUIVALENT MAGNIFICATION	SIZE FOR IMAGEN 8/300
`\magstep0`	1000	1500
`\magstephalf`	1095	1643
`\magstep1`	1200	1800
`\magstep2`	1440	2160
`\magstep3`	1728	2592
`\magstep4`	2074	3110

4 Units of Measurement

In the documentation following, ⟨dimen⟩ represents a unit of measurement. All measurements must be a number and a unit. This includes measurements of zero—for instance, to specify that paragraphs are not to be indented, you might type `\parindent=0in`.

TEX accepts the following units:

pt	point (these parentheses are 10 pt high)
pc	pica (1 pc = 12 pt)
in	inch (1 in = 72.27 pt)
bp	big point (72 bp = 1 in)
cm	centimeter (2.54 cm = 1 in)
mm	millimeter (10 mm = 1 cm)
dd	didôt point (1157 dd = 1238 pt)
cc	cicero (1 cc = 12 dd)
sp	scaled point (65536 sp = 1 pt)

All dimensions are multiplied by the *magnification*; in some cases this is desirable (e.g., you might want spacing between paragraphs to be proportional to the font size), in other cases not (usually you would want the margins around the text to be constant). To prevent a dimension from being magnified, insert the prefix `true` just before the unit. For instance, `\voffset=1truein`.

5 Document Formatting

The following control sequences are generally given at the top of your TEX file. They set dimensions which will usually remain constant throughout the document.

`\magnification`=⟨number⟩

Sets the amount by which the entire document will be magnified. You cannot reset this parameter within a document, and it is best to put this line at the very top of your file. The ⟨number⟩ is 1000 times the actual magnification; i.e., 1000 is the default magnification. You may use the control sequences listed above ("Available Font Sizes") instead of numeric values.

`\hoffset`=⟨dimen⟩

Sets the horizontal (left) margin of the page to ⟨dimen⟩.

`\voffset=`⟨dimen⟩
Sets the vertical (top) margin to ⟨dimen⟩.

`\hsize=`⟨dimen⟩
Sets the horizontal text width to ⟨dimen⟩. (Thus, the right margin is $8.5'' - (hoffset + hsize)$).

`\vsize=`⟨dimen⟩
Sets the vertical text length to ⟨dimen⟩. (Thus, the bottom margin is $11'' - (hoffset + hsize)$).

`\nopagenumbers`
Turns off the automatic printing of page numbers at the bottom of each page. Does *not* turn off automatic numbering, though. See *The TEXBook* pp. 252–253 for a discussion of page numbers, and an example of how to put page numbers at the top of each page.

`\pageno=`⟨number⟩
Sets the initial page number. Defaults to 1. If you use a negative number, the page numbers will be printed as (positive) Roman numerals.

`\parindent=`⟨dimen⟩
Sets the amount of space to indent the first line in a paragraph. The default value should be 20pt.

`\parskip=`⟨dimen⟩
Sets the amount of space to skip between paragraphs. To get the space of a single blank line, set to 12pt. Most books do not skip space between paragraphs (set to 0pt).

`\baselineskip=`⟨dimen⟩
Sets the distance between consecutive baselines of text. `\baselineskip=`12pt is the default setting; for double-spaced lines, set to 24pt.

`\raggedbottom`
Tells TEX not to force each page of output to be exactly the same length. This helps keep TEX from putting page breaks at bad spots—like just after the header of a section (see also `\goodbreak`)

`\raggedright`
Tells TEX not to make the right margins exactly flush (but TEX will still make them *almost* flush). If you want to set a long passage in `typewriter type` (`\tt`), use `\ttraggedright` instead of `\tt`.

`\tolerance=`⟨number⟩
Sets the "tolerance" TEX will have for "overfull hboxes." That is, if set relatively high (up to 10000), TEX will be less bothered by lines of text with cramped spacing; you will be less likely to get error messages and those black boxes on your output. On the other hand, the general quality of spacing will be degraded.

`\pretolerance=`⟨number⟩
If set high to a high value (up to 10000), it will have not only the same effect as setting a high value for `\tolerance`, but will also reduce the use of hyphenation. Many journals discourage hyphenation on manuscripts; set `\pretolerance=`10000 for such papers.

6 Grouping

Often it is necessary to make "local" modifications to the default format you have set up. For instance, you might want to set a single word or phrase in a different font, or you might want several indented paragraphs with less space between lines. These areas of local variation are delimited by braces: { and }, or, equivalently, by `\begingroup` and `\endgroup`. Any parameters which are reset within these groups return to their original state when the group ends. Many levels of nesting of groups are permitted. Most frequently, grouping is used to set *parameters*; for example, setting a different font, or for setting new values for `\parskip`, `\parindent`, and `\baselineskip`.

Braces are also used as delimiters for arguments to macros. TEX macros usually expect that their arguments will be either one character, or that they will be a string of characters within braces. When learning about a new TEX control sequence, it is important to determine what arguments, if any, it expects.

7 Page and Line Breaks

TEX dislikes being told what to do; it would rather decide for itself. Consequently, many of these control sequences tell TEX that a certain place is a relatively "good" or "bad" place for a break.

7.1 Page Breaks

`\eject`
: Forces a page break at this point. Usually you should precede `\eject` with `\vfill`, to fill the bottom of the current page with blank space. Otherwise, the current page may contain large vertical spaces between paragraphs, as TEX tries to expand the page to the bottom.

`\goodbreak`
: Causes a `\par` and tells TEX that this would be a good place to have a page break if you are anywhere near the bottom of a page. When `\raggedbottom` has been specified above, will usually cause a break if the next paragraph cannot be fit on the current page.

`\bigbreak`
: Causes a `\par` and tells TEX that this would be a good place to have a page break if you are anywhere near the bottom of a page, but isn't as strong as `\goodbreak`. Also, causes a `\bigskip`.

`\medbreak`
: Same as `\bigbreak`, but still weaker. Causes a `\medskip`.

`\smallbreak`
: Same as `\bigbreak`, but even weaker than `\medbrea k`. Causes a `\smallskip`.

7.2 Line Breaks

`\break`
: Forces TEX to break the line at this point. Usually, you will want to precede it with an `\hfil`.

`\nobreak`
: Prohibits TEX from breaking the line at this point.

See also `\obeylines`, below.

8 Paragraph Formatting

`\par`
: Forces an end-of-paragraph. Equivalent to a blank line (or multiple consecutive blank lines).

`\indent`
: Can be used to put a horizontal blank space in a line. The width of the space is equal to the current value of `\parindent`.

`\noindent`
: Placed at the beginning of a paragraph, `\noindent` prevents TEX from inserting the usual `\parindent`.

`\item{...}`
: The ellipsis represents an item number or symbol. `\item` is used for outline format—the symbol within braces following `\item` appears flush left, and the entire paragraph which follows is indented by the amount of `\indent`. The section on "Special Characters" uses the `\item` macro.

`\itemitem{...}`
: The ellipsis represents an item number or symbol. `\itemitem` does the same thing as `\item`, but its symbol is indented further, and the text further still.

`\obeylines`
: Tells TEX to treat the $\boxed{\texttt{ret}}$ at the end of each line of text in your file as an end-of-paragraph (`\par`). Useful for addresses and poetry. Usually you will want to use `\obeylines` within a group (delimited by braces), in conjunction with `\parskip`=0pt (so that you don't get that excess space between lines).

`\narrower`
: Brings in both the left and right margins by the current value of `\parindent` (by resetting `\leftskip` and `\rightskip`). Useful for setting an extended quotation within a paper. In such cases, `\baselineskip` will often be temporarily reset (to a smaller value). Use within a group, just before the closing right-brace, insert a `\smallskip`.

`\parshape`=⟨number⟩
: Allows you to specify an arbitrary paragraph shape. ⟨number⟩ $= n\ i_1\ l_1\ i_2\ l_2 \ldots i_n\ l_n$. `\parshape` $= n\ i_1\ l_1\ i_2\ l_2 \ldots i_n\ l_n$ specifies a paragraph shape in which the first n lines will have lengths $l_1\ l_2 \ldots l_n$, respectively, and they will be indented from the left margin by the amounts $i_1\ i_2 \ldots i_n$. If the paragraph has fewer than n lines, the additional specifications will be ignored; if it has more than n lines, the specifications for all n lines will repeat.

`\hangafter`=n
: This is a special case of `\parshape`. It specifies that the first n lines of the paragraph are not to be indented, but all subsequent lines are to be indented by the amount specified by `\hangindent`. If n is a negative number, then the first $|n|$ lines will be indented, and all lines thereafter will not be.

`\hangindent=`⟨dimen⟩
: The amount by which lines are to be indented, as specified by `\hangafter`. If ⟨dimen⟩ is negative, the lines will have indentation from the right margin rather than the left.

`\everypar={...}`
: The ellipsis represents a set of control sequences to execute at the beginning of each subsequent paragraph. For instance, in order to give all paragraphs the same shape, use in conjunction with `\parshape` or `\hangafter`.

9 Line Formatting

`\centerline{`*some text*`}`
: Causes *some text* to be centered in the middle of a line of its own.

`\leftline{`*some text*`}`
: Causes *some text* to be left-justified on a line of its own.

`\rightline{`*some text*`}`
: Causes *some text* to be right-justified on a line of its own.

`\line{`*some text*`}`
: Creates an "hbox" for the enclosed text, whose width is the current value of `\hsize`. Usually you will need some kind of fill within the line (e.g., `\hfill`, or `\dotfill`).

10 Horizontal and Vertical Spacing

10.1 Horizontal spaces

A horizontal space is called "unbreakable" if it prevents TEX from inserting a line break at that point.

Breakable horizontal spaces:		Unbreakable horizontal spaces	
`\␣`	normal interword space	`~`	normal interword space
`\enskip`	this much	`\enspace`	this much
`\quad`	this much	`\thinspace`	this much
`\qquad`	this much	`\negthinspace`	thismuch
`\hskip`	⟨arbitrary dimen⟩	`\kern`	⟨arbitrary dimen⟩

The following "infinite" space fillers are used within the various `\line` macros above, or more generally, within `\hbox`es. See *The TEXBook*, pp. 71–72.

`\hfil`
: "Infinite" horizontal space.

`\hfill`
: "Infinite" horizontal space, *bigger* infinity than `\hfil`.

`\dotfill`
: An "Infinite" row of dots; as infinite as `\hfill`.

`\hrulefill`

An "infinite" line; as infinite as `\hfill`.

10.2 Vertical spaces

`\bigskip`	12 pt
`\medskip`	6 pt
`\smallskip`	3 pt
`\vskip`	⟨arbitrary dimen⟩

`\vfill`

Fills from current position to the bottom of the page with blank space. **`\vfill`** is useful immediately before `\eject`. See also *The TEXBook*, pp. 71–72.

11 Footnoting

The `\footnote` macro takes two parameters: first is the reference mark *, and second is the footnote text. The footnote in the last sentence, for example, was produced by typing:

```
$\ldots$ first is the reference mark,\footnote*{The asterisk is
the ``reference mark'' here.} and second$\ldots$
```

TEX does not have a macro to produce automatically numbered footnotes, but in **EXERCISE 15.12** of *The TEXBook*, which I have copied here,[1] Knuth shows how such a macro, to be called `\note`, can be designed:[2]

```
\newcount\notenumber
\def\clearnotenumber{\notenumber=0}
\def\note{\advance\notenumber by 1
\footnote{$^{\the\notenumber}$}}
```

* The asterisk is the "reference mark" here.

[1] I also have copied it at the top of my text-file.

[2] And it seems to work.

12 Tables

`\settabs` is the macro used to typeset tables. If you are interested in doing anything fancy with tables, you should read Chapter 22 (pp. 231–249) of *The TEXBook*. Here, however, is a simple sample table, copied from page 232:

```
\settabs\+\indent&Horizontal lists\quad&\cr % sample line
\+&Horizontal lists&Chapter 14\cr
\+&Vertical lists&Chapter 15\cr
\+&Math lists&Chapter 17\cr
```

This produces the following three-line table:

Horizontal lists	Chapter 14
Vertical lists	Chapter 15
Math lists	Chapter 17

The control sequence `\settabs` is immediately followed by the characters `\+` which also begin every line of the actual table. The remainder of the line following `\settabs\+` is called the *sample line*. The sample line determines the width of each column. Column positions are specified by the `&` character. Thus, in this example, the first column has a width of a single paragraph indentation; column 2 is as wide as the string "Horizontal lists" plus one quad of space; column 3 extends to the end of the page. Note that the sample line itself is not printed in the resulting table. In choosing a sample line, it is best to choose the widest entry for each column, plus a little extra space.

13 `\input` and `\end`

`\input` ⟨filespec⟩
Causes TEX to immediately begin reading the file ⟨filespec⟩. There must be a space between `\input` and ⟨filespec⟩. ⟨filespec⟩ is a standard VAX VMS file-specification. For VAX/VMS TEX, if the file has no file-type specified, TEX will look for ⟨file⟩.TEX. If no directory is given, TEX will first search the directory in which you are currently working; if the file is not found there, TEX will look in the directory given by logical name **TEX_INPUTS**. Presumably similar defaults are provided on other operating systems. If the file still is not found, TEX will print an error message, and give you the opportunity to enter the correct file-specification.

`\end`
Must appear at the end of every TEX document. However, if you are using the `\input` command to load other files, only the main document should contain an `\end`. The `\end` macro automatically executes a `\vfill\eject`.

14 Math

Sorry—it would take too long; maybe in the next edition. Also, this is one thing that *The TEXBook* is good for. Read Chapters 16–19, or just look at the examples.