1. Ordinary text

1.1 Punctuation

.	`.`	period	- -	`-`	hyphen
,	`,`	comma	- -	* `\-`	discretionary hyphen
;	`;`	semicolon	- –	`--`	en-dash, for ranges
:	`:`	colon	- —	`---`	em-dash, sentence dash
?	`?`	question mark	/	`/`	slash
!	`!`	exclamation	/	`\slash`	/ with break allowed
“ ”	`‘‘ ’’`	quotes	¿	`?‘`	open question
‘ ’	`‘ ’`	single quotes	¡	`!‘`	open exclamation
‘ ’	`\lq \rq`	single quotes	. . .	`\dots`	ellipsis
()	`()`	parentheses	[]	`[]`	brackets

“‘ ’”	`‘‘\thinspace‘ ’\thinspace’’`	quotes within quotes
‘“ ”’	`‘{}‘‘ ’’’`	quotes within quotes
one-of-a-kind	`one-of-a-kind`	hyphen
2-3 tree	`2-3 tree`	hyphen
pages 6–15	`pages 6--15`	en-dash
Wisconsin–Madison	`Wisconsin--Madison`	en-dash
I win—you lose.	`I win---you lose.`	em-dash

1.2 Accents

é	`\'e`	acute	ç	`\c c`	cedilla	ȯ	`\.o`	dot
è	`` \`e ``	grave	ñ	`\~n`	tilde	o̲	`\b o`	bar-under
ö	`\"o`	dieresis or umlaut	ř	`\v r`	check	ọ	`\d o`	dot-under
ő	`\H o`	long Hungarian umlaut	ı͡a	`\t \i a`	tie-after	ı	`\i`	dotless i
ô	`\^o`	circumflex	ā	`\=a`	macron	ȷ	`\j`	dotless j
			ă	`\u a`	breve			

* `\accent`⟨ccode⟩	General accent; put character ⟨ccode⟩ from the current font over the next character.

1.3 Special characters

#	`\#`	pound sign	@	`@`	at sign
$	`\$`	dollar sign	•	`$\bullet$`	bullet
£	`{\it \$}`	pounds sterling	†	`\dag`	obelisk
¢	`\hbox{\rm\rlap/c}`	cents sign	‡	`\ddag`	diesis
%	`\%`	percent sign	§	`\S`	section
&	`\&`	ampersand	¶	`\P`	paragraph, pilcrow
*	`*`	asterisk	©	`\copyright`	copyright
_	`\_`	underbar	♯	`$\sharp$`	
"	`{\tt "}`	ditto mark	♭	`$\flat$`	
æ, Æ	`\ae,\AE`	Latin and Scandinavian	♮	`$\natural$`	
œ, Œ	`\oe,\OE`	French	♠	`$\spadesuit$`	
ß	`\ss`	German	♡	`$\heartsuit$`	
å, Å	`\aa,\AA`	Scandinavian	♢	`$\diamondsuit$`	
ø, Ø	`\o,\O`	Scandinavian	♣	`$\clubsuit$`	
ł, Ł	`\l,\L`	Polish			

* `\uppercase{`⟨tokens⟩ `}`	Convert tokens to upper case.
* `\lowercase{`⟨tokens⟩ `}`	Convert tokens to lower case.
* `\number`⟨number⟩	Convert number to arabic digits.
* `\romannumeral`⟨number⟩	Convert number to lowercase roman numerals.
`\uppercase\expandafter{\romannumeral`⟨number⟩ `}`	Convert number to uppercase roman numberals.
* `\char`⟨ccode⟩	Print character with given code.
* `\chardef\cs=`⟨ccode⟩	Define symbolic equivalent to character code.

1.4 Fonts, styles, sizes

	`\rm`	roman	`\tenrm`	Ten point roman	`\sevenrm`	Seven point roman
	`\bf`	**boldface**	`\tenbf`	**Ten point bold**	`\sevenbf`	**Seven point bold**
	`\tt`	`typewriter`	`\tentt`	`Ten point typewriter`	`\fiverm`	Five point roman
	`\it`	*italic*	`\teni`	*Ten point italic*	`\fivebf`	**Five point bold**
	`\sl`	*slanted*	`\tensl`	*Ten point slanted*	* `\nullfont`	Font with no characters

* `\/` Italic correction — extra space compensating for switch from slanted to unslanted fonts, as in `{\it italic text\/} normal text`.

* `\font\cs=`⟨external name⟩ [`at` ⟨size⟩ | `scaled` ⟨magnification⟩] Make `\cs` select named font.

• `\fontdimen`⟨number⟩⟨font⟩ `=`⟨value⟩ Parameters describing font.

1.5 Hyphenation

* `\hyphenation{al-go-rithm zy-mur-gy}` Add to hyphenation dictionary.
`\-` Discretionary hyphen: `non\-negative`.
• `\uchyph=`⟨number⟩ Positive enables hypenation of capitalized words [`1`].
• `\hyphenchar`⟨font⟩ `=`⟨ccode⟩ Character code used for discretionary hyphen [`` `\- ``].
* `\discretionary{`⟨pre-break text⟩ `}{`⟨post-break text⟩ `}{`⟨no-break text⟩ `}` General discretionary break.
• `\brokenpenalty=`⟨penalty⟩ After line ending with discretionary break [`100`].
• `\doublehyphendemerits=`⟨demerit⟩ Demerits for consective discretionary breaks [`10000`].
• `\hyphenpenalty=`⟨penalty⟩ Penalty for non-empty pre-break text [`50`].
• `\exhyphenpenalty=`⟨penalty⟩ Penalty for empty pre-break text [`50`].

2. White space

2.1 Horizontal

	`\space`	Same as ␣ (blank space).
*	`\␣`	Skip this much (breakable), the normal interword amount.
	`~`	Skip this much (unbreakable), the normal interword amount.
	`\enskip`	Skip this much (breakable).
	`\enspace`	Skip this much (unbreakable).
	`\quad`	Skip this much (breakable).
	`\qquad`	Skip this much (breakable).
	`\thinspace`	Skip this much (unbreakable).
	`\negthinspace`	Skip thismuch (unbreakable).
*	`\hskip`⟨glue⟩	Skip given amount (breakable).
*	`\kern`⟨dimen⟩	Skip given amount (unbreakable).
	`\hglue`⟨glue⟩	Skip given amount (unbreakable, starts paragraph if needed).
*	`\hfil`	Skip by zero, with infinite stretch.
*	`\hfill`	Skip by zero, with second order infinite stretch.
*	`\hfilneg`	Skip by zero, with infinite negative stretch.
*	`\hss`	Skip by zero, with infinite stretch and shrink.
	`\removelastskip`	Discard previous skip, if any.
	`\nonfrenchspacing`	Extra space after sentences [`on`].
	`\frenchspacing`	No extra space after sentences [`off`].
•	`\spaceskip=`⟨glue⟩	Override interword glue.
•	`\xspaceskip=`⟨glue⟩	Override intersentence glue.
•	`\spacefactor=`⟨sfactor⟩	Set space factor (governs interword glue).
•	`\sfcode`⟨ccode⟩ `=`⟨sfactor⟩	Set character's space factor.

Use a tie (`~`) to avoid breaking line near short words or symbols, as in `Chapter~4`; `Donald~E. Knuth`; `$x$,~$y$, and~$z$`; `speeding is (a)~foolish, (b)~dangerous`; `algorithm~A`; `less than~0`.

Say `Call the FBI\null.` to end a sentence with a capital letter, lest TEX omit the extra space after the period.

Say `Murder, Inc.\ did it.` to end a word with a period in mid-sentence, lest TEX insert extra space.

Use `\frenchspacing` in bibliographies or other copy containing lots of abbreviations.

2.2 Vertical

`\smallskip`	Skip this much.
`\smallbreak`	Skip as above, and encourage page break slightly.
`\medskip`	Skip this much.
`\medbreak`	Skip as above, and encourage page break.
`\bigskip`	Skip this much.
`\bigbreak`	Skip as above, and encourage page break a lot.
`\filbreak`	Encourage page break, fill bottom of page with white space if break is made.
* `\kern`⟨dimen⟩	Skip given amount (unbreakable).
* `\vskip`⟨glue⟩	Skip given amount (breakable).
`\vglue`⟨glue⟩	Skip given amount (doesn't disappear at breaks).
* `\vss`	Skip by zero, with infinite stretch and shrink.
* `\vfil`	Skip by zero, with infinite strecth.
* `\vfill`	Skip by zero, with second order infinite stretch.
* `\vfilneg`	Skip by zero, with infinite negative stretch.
`\removelastskip`	Discard previous skip, if any.

3. Layout

3.1 Page image

Line of full width (`\hsize`).	`\line{Line ...}` Usually contains `fill` glue.
Left-justified line	`\leftline{Left-justified line}`
Centered line	`\centerline{Centered line}`
Right-justified line	`\rightline{Right-justified line}`
Move in both margins, making narrower paragraphs. This is traditional for long quotations.	`\narrower Move...quotations.`
3. A New Section. We shall solve a famous problem in Theorem 1. The first paragraph of a new section is not indented.	`\beginsection 3.  A New Section.` ⟨blank line⟩ `We shall...indented.` Skips some space (or starts a new page if near the bottom), and displays the title paragraph in bold.
Theorem 1. *All zeros of the zeta function lie on the line* $\Re z = 1/2$.	`\proclaim Theorem 1.  All...$\Re z=1/2$.` ⟨blank line⟩ Puts the title in bold and the paragraph in slanted. A period separates title from paragraph.

* `\hrule` $\begin{bmatrix}\text{width } \langle\text{dimen}\rangle \\ \text{height } \langle\text{dimen}\rangle \\ \text{depth } \langle\text{dimen}\rangle\end{bmatrix}^*$	Horizontal rule of given size [`height 0.4pt depth 0pt`].
* `\vrule` $\begin{bmatrix}\text{height } \langle\text{dimen}\rangle \\ \text{depth } \langle\text{dimen}\rangle \\ \text{width } \langle\text{dimen}\rangle\end{bmatrix}^*$	Vertical rule of given size [`width 0.4pt`].
`\break`	Break line or page.
`\eject`	Force page break.
`\goodbreak`	Desirable place for page break.

`\allowbreak`	Permit line or page break here.
`\nobreak`	Do not break line or page.
* `\penalty`⟨penalty⟩	Explicit penalty, encouraging or discouraging break.
`\supereject`	End page and print all pending insertions.
• `\maxdepth=`⟨dimen⟩	Maximum depth of page [`4pt`].

3.2 Paragraphs

This paragraph has no indentation. All its lines start at the left margin.	* `\noindent This...margin.`
This is a normal indented paragraph. Usually no command is required, but `\indent` also works.	* `\indent This is...works.`
Fully indented paragraph. All its lines are indented.	`\hang Fully...indented.`
Paragraph with hanging indentation. The indentation is twice the normal amount, and takes effect after two lines. The normal shape of paragraphs will be restored when this one is done.	• `\hangindent=2\parindent` Amount of hanging indentation (a ⟨dimen⟩). Positive indents at the left, negative at the right. • `\hangafter=2 Paragraph...done.` Duration of hanging indentation. Positive indents after n lines, negative indents first $-n$ lines.
• Normal paragraph with a tag in the indent. Only the first line is indented.	`\textindent{$\bullet$} Normal...indented.`
a) Indented paragraph with a tag in the indent. The whole paragraph is indented.	`\item{a)} Indented...indented.`
a.1) Doubly indented paragraph with a tag in the indent.	`\itemitem{a.1)} Doubly...indent.`
You can change the shape of paragraphs to leave room for figures, or create interesting designs filled with words. This paragraph has a two-tenths inch indent, and also has half an inch taken out of the right after two lines to leave room for a small figure.	`\parshape=3 .2in 2.8in 0pt 3in 0pt 2.5in You...figure.` The first number tells how many shape pairs there are; then comes pairs of dimensions, giving indents and lengths of each line. The last pair is repeated for extra lines, if any.

* `\par`	End current paragraph.
⟨blank line⟩	Generates `\par`, ending paragraph.
`\endgraf`	End the paragraph (synonym for `\par`).
• `\parskip=`⟨glue⟩	Glue before paragraphs [`0pt plus 1pt`].
• `\parindent=`⟨dimen⟩	Indentation for first line of paragraphs [`20pt`].
• `\looseness=`⟨number⟩	Try to make next paragraph n lines longer/shorter.
• `\parfillskip=`⟨glue⟩	Glue added to last line of paragraph [`0pt plus 1fil`].
• `\interlinepenalty=`⟨penalty⟩	Between every line in a paragraph [`0`].
• `\clubpenalty=`⟨penalty⟩	After first line of paragraph [`150`].
• `\widowpenalty=`⟨penalty⟩	Before last line of paragraph [`150`].
• `\displaywidowpenalty=`⟨penalty⟩	Before line preceding a display [`50`].
• `\finalhyphendemerits=`⟨demerit⟩	Demerits for ending penultimate line with discretionary break [`5000`].

Say `{\setbox0=\lastbox}` at the beginning of the paragraph to discard the indentation.

3.3 Lines

• `\leftskip=`⟨glue⟩	Glue at left of every line of paragraph.
• `\rightskip=`⟨glue⟩	Glue at right of every line of paragraph.
`\raggedright`	Don't align right margins.
`\ttraggedright`	Use font `\tt` and don't align right margins.

	`\normalbaselines`	Make lines the normal distance apart.
	`\openup`⟨dimen⟩	Add ⟨dimen⟩ to distance between lines, usually expressed in terms of `\jot`.
	`\strut`	Give line height and depth, separating it from its neighbors.
	`\strutbox`	Invisible box used for `\strut`.
•	`\baselineskip=`⟨glue⟩	Distance between baselines [`12pt`].
•	`\lineskip=`⟨glue⟩	Glue between close lines [`1pt`].
•	`\lineskiplimit=`⟨dimen⟩	Threshold for lines to be close [`0pt`].
•	`\normalbaselineskip=`⟨glue⟩	Distance between lines [`12pt`].
•	`\normallineskip=`⟨glue⟩	Glue between close lines [`1pt`].
•	`\normallineskiplimit=`⟨dimen⟩	Threshold for lines to be close [`0pt`].
	`\nointerlineskip`	Omit interline glue.
	`\offinterlineskip`	Turn off spacing between lines.
•	`\adjdemerits=`⟨demerit⟩	Demerits for adjacent incompatible lines [`10000`].
•	`\linepenalty=`⟨penalty⟩	Contributes to every line's badness [`10`].
•	`\tolerance=`⟨badness⟩	Badness threshold for acceptable line breaks [`200`].
•	`\pretolerance=`⟨badness⟩	Badness threshold for breaks without hyphens [`100`].

3.4 Horizontal

`\llap{`⟨contents⟩ `}`	Overlap contents to the left.
`\rlap{`⟨contents⟩ `}`	Overlap contents to the right.
`\underbar{`⟨contents⟩ `}`	<u>Underline contents</u>.
`\hrulefill`	Like `\hfill`, except use a rule ($\rule{4em}{0.4pt}$).
`\dotfill`	Like `\hfill`, except use dots (. . . . .).
`\leftarrowfill`	Like `\hfill`, except use a left arrow ($\longleftarrow\!\!\!-\!\!\!-$).
`\rightarrowfill`	Like `\hfill`, except use a right arrow ($-\!\!\!-\!\!\!\longrightarrow$).
`\upbracefill`	Like `\hfill`, except use an upbrace ($\underbrace{\qquad\qquad}$).
`\downbracefill`	Like `\hfill`, except use an downbrace ($\overbrace{\qquad\qquad}$).

4. Text constructions

4.1 Horizontal

*	`\hbox`$\left[\begin{array}{l}\texttt{to}\ \langle\text{dimen}\rangle\\ \texttt{spread}\ \langle\text{dimen}\rangle\end{array}\right]$ `{`⟨contents⟩ `}`	Create horizontal box.
*	`\lower`⟨dimen⟩	Move next box down.
*	`\raise`⟨dimen⟩	Move next box up.
*	`\leaders`$\left\{\begin{array}{l}\langle\text{box}\rangle\\ \langle\text{rule}\rangle\end{array}\right\}\left\{\begin{array}{l}\langle\text{horizontal glue}\rangle\\ \langle\text{vertical glue}\rangle\end{array}\right\}$	Fill the space occupied by the glue with copies of the box. Align the reference points of the boxes with the enclosing box.
*	`\cleaders`	Like `\leaders`, except align the boxes in the center of the glue, leaving white space at either end.
*	`\xleaders`	Like `\leaders`, except align the boxes within the glue, distributing leftover space equally between boxes.
	`\phantom{`⟨contents⟩ `}`	Invisible box of height, depth, and width of contents.
	`\hphantom{`⟨contents⟩ `}`	Invisible box of height and depth 0, width of contents.
	`\smash{`⟨contents⟩ `}`	Visible box of height and depth 0, width of contents.
	`\null`	Empty hbox.
	`\leavevmode`	Start a paragraph.
•	`\lastskip`	Size of last glue, if any.
•	`\lastpenalty`	Amount of last penalty, if any.
•	`\lastkern`	Dimension of last kern, if any.
*	`\unskip`	Discard the last glue or leaders, if any.
*	`\unpenalty`	Discard last penalty, if any.
*	`\unkern`	Discard last kern, if any.
*	`\lastbox`	Remove last `\hbox` or `\vbox` from current list and use it here.

- `\everyhbox={`⟨tokens⟩`}` — Insert tokens into every hbox [`null`].
- `\hfuzz=`⟨number⟩ — Grace for overfull hboxes [`0.1pt`].

Use `\leavevmode` to start a paragraph with a box.

4.2 Vertical

* `\vbox` [`to` ⟨dimen⟩ / `spread` ⟨dimen⟩] `{`⟨contents⟩`}`	Create vertical box.
* `\vtop` [`to` ⟨dimen⟩ / `spread` ⟨dimen⟩] `{`⟨contents⟩`}`	Create vertical box with reference point at top.
* `$\vcenter` [`to` ⟨dimen⟩ / `spread` ⟨dimen⟩] `{`⟨vertical mode⟩`}$`	Create vertical box centered around axis.
* `\moveleft`⟨dimen⟩	Shift next box left.
* `\moveright`⟨dimen⟩	Shift next box right.
`\vphantom{`⟨contents⟩`}`	Invisible box of width 0, height and depth of contents.
• `\lastskip`	Size of last glue, if any.
• `\lastpenalty`	Amount of last penalty, if any.
• `\lastkern`	Dimension of last kern, if any.
* `\unskip`	Discard the last glue or leaders, if any (except in main vertical list).
* `\unpenalty`	Discard last penalty, if any (except in main vertical list).
* `\unkern`	Discard last kern, if any (except in main vertical list).
* `\lastbox`	Remove last `\hbox` or `\vbox` from current list and use it here (except in main vertical list).
• `\everyvbox={`⟨tokens⟩`}`	Insert tokens into every vbox [`null`].
• `\prevdepth=`⟨dimen⟩	Depth of last box on current vertical list.
• `\boxmaxdepth=`⟨dimen⟩	Maximum depth of vertical boxes [∞].
* `\vsplit`⟨breg⟩ `to` ⟨dimen⟩	Split off material from beginning of box register to form vbox of height ⟨dimen⟩.
• `\splitmaxdepth=`⟨dimen⟩	Maximum depth of box formed via `\vsplit` [∞].
• `\splittopskip=`⟨glue⟩	Glue added before first box in a `\vsplit` [`10pt`].

4.3 Paragraphs

* `\vadjust{`⟨vertical mode⟩`}`	Add to paragraph after current line.
`\obeylines`	End of input line generates `\par`.
`\obeyspaces`	Don't ignore consecutive spaces between words.
• `\everypar={`⟨tokens⟩`}`	Read tokens at the beginning of every paragraph.
• `\prevgraf`	Number of lines in the last paragraph. Displays count as 3 lines.

To make a paragraph in a box, say `\vbox{\hsize=`⟨dimen⟩ `\indent` ⟨text⟩ `}`. `\vtop` works similarly.

5. Mathematics

`$`⟨text math mode⟩`$`	`^`	Superscript	`\sp`	Superscript
`$$`⟨display math mode⟩`$$`	`_`	Subscript	`\sb`	Subscript

5.1 Special characters

α A	`\alpha {\rm A}`	ι I	`\iota {\rm I}`	ϱ	`\varrho`
β B	`\beta {\rm B}`	κ K	`\kappa {\rm K}`	$\sigma\ \Sigma$	`\sigma \Sigma`
$\gamma\ \Gamma$	`\gamma \Gamma`	$\lambda\ \Lambda$	`\lambda \Lambda`	ς	`\varsigma`
$\delta\ \Delta$	`\delta \Delta`	μ M	`\mu {\rm M}`	τ T	`\tau {\rm T}`
ϵ E	`\epsilon {\rm E}`	ν N	`\nu {\rm N}`	$\upsilon\ \Upsilon$	`\upsilon \Upsilon`
ε	`\varepsilon`	$\xi\ \Xi$	`\xi \Xi`	$\phi\ \Phi$	`\phi \Phi`
ζ Z	`\zeta {\rm Z}`	o O	`o {\rm O}`	φ	`\varphi`
η H	`\eta {\rm H}`	$\pi\ \Pi$	`\pi \Pi`	χ X	`\chi {\rm X}`
$\theta\ \Theta$	`\theta \Theta`	ϖ	`\varpi`	$\psi\ \Psi$	`\psi \Psi`
ϑ	`\vartheta`	ρ R	`\rho {\rm R}`	$\omega\ \Omega$	`\omega \Omega`

$\aleph$	`\aleph`	∞	`\infty`	$\angle$	`\angle`
ℓ	`\ell`	$\surd$	`\surd`	∇	`\nabla`
$\hbar$	`\hbar`	$\prime$	`\prime`	$\triangle$	`\triangle`
$\emptyset$	`\emptyset`	f'	`f'`		

	`{\cal ABC...XYZ}`	$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ (calligraphic caps).
	`{\oldstyle 0123456789}`	0123456789 (old style digits).
	`{\mit \Gamma\Delta...\Omega}`	$\mathit{\Gamma\Delta\Theta\Lambda\Xi\Pi\Sigma\Upsilon\Phi\Psi\Omega}$ (math italic Greek caps).
*	`\mathchar`⟨mcode⟩	Use math char.
*	`\mathchardef\cs=`⟨mcode⟩	Define math char.
*	`\mathcode`⟨ccode⟩ `=`⟨mcode⟩	Define class, family, and position of math character.

5.2 Arithmetic

$+$	`+`	$\sum$ $\textstyle\sum$	`\sum`	$\prod$ $\textstyle\prod$	`\prod`
$-$	`-`	$\mathop{\sum}'$	`\mathop{{\sum}'}`	$\doteq$	`\doteq`
$\pm$	`\pm`	$=$	`=`	$\approx$	`\approx`
$\mp$	`\mp`	$\ne$	`\ne`	$\propto$	`\propto`
$\ast$	`\ast`	$\neq$	`\neq`	$\sim$	`\sim`
$\cdot$	`\cdot` (multiplication)	$> <$	`> <`	$\simeq$	`\simeq`
$\star$	`\star`	$\ge \le$	`\ge \le`	$\asymp$	`\asymp`
$\times$	`\times`	$\geq \leq$	`\geq \leq`	$\max$	`\max`
$\times$	`\*` (discretionary)	$\gg \ll$	`\gg \ll`	$\min$	`\min`
$/$	`/`	$\succ \prec$	`\succ \prec`	$\sqrt{x}$	`\sqrt{x}`
$\div$	`\div`	$\succeq \preceq$	`\succeq \preceq`	$\sqrt[3]{x}$	`\root 3 \of x`
$\left\vert x\right\vert$	`\left| x\right|` (absolute value)				

Put physical units in roman, preceded by a thin space: $2.54\,\rm cm$ (`$2.54\,\rm cm$`).
Put thin space after square root followed by letter or digit: $\sqrt{\pi}\,n$ (`\sqrt{\pi}\,n`).

5.3 Sets

$\{\ \}$	`\{ \}`	$\subset$	`\subset`	$\cup$	`\cup`	$\bigcup$ $\textstyle\bigcup$	`\bigcup`
$\in$	`\in`	$\subseteq$	`\subseteq`	$\cap$	`\cap`	$\bigcap$ $\textstyle\bigcap$	`\bigcap`
$\notin$	`\notin`	$\supset$	`\supset`	$\uplus$	`\uplus`	$\biguplus$ $\textstyle\biguplus$	`\biguplus`
$\ni$	`\ni`	$\supseteq$	`\supseteq`	$\setminus$	`\setminus`		
$\owns$	`\owns`	$\emptyset$	`\emptyset`	$:$	`:`		
				$\mid$	`\mid`		

Use thin spaces to set off set brackets when the set is defined by membership test:
$\{\, x:f(x)=0 \,\}$ (`$\{\, x:f(x)=0 \,\}$`).
Don't use thin spaces for explicitly given sets: $\{1,4,9\}$ (`$\{1,4,9\}$`).

5.4 Algebra

$\deg$	`\deg`	$\wp$	`\wp`	$\oplus$	`\oplus`	$\bigoplus$ $\textstyle\bigoplus$	`\bigoplus`
$\det$	`\det`	$\wr$	`\wr`	$\otimes$	`\otimes`	$\bigotimes$ $\textstyle\bigotimes$	`\bigotimes`
$\dim$	`\dim`	$\amalg$	`\amalg`	$\odot$	`\odot`	$\bigodot$ $\textstyle\bigodot$	`\bigodot`
$\hom$	`\hom`	$\coprod$ $\textstyle\coprod$	`\coprod`				
$\ker$	`\ker`						

5.5 Analysis

$\sin$	`\sin`	$\sinh$	`\sinh`	$\exp$	`\exp`	f'	`f'`
$\cos$	`\cos`	$\cosh$	`\cosh`	$\log$	`\log`	f''	`f''`
$\tan$	`\tan`	$\tanh$	`\tanh`	$\ln$	`\ln`	f'''	`f'''`
$\sec$	`\sec`	$\coth$	`\coth`	$\lg$	`\lg`	∂	`\partial`
$\csc$	`\csc`	$\lim$	`\lim`	$\Re$	`\Re`	∇	`\nabla`
$\cot$	`\cot`	$\inf$	`\inf`	$\Im$	`\Im`	$\displaystyle\int \textstyle\int$	`\int`
$\arcsin$	`\arcsin`	$\sup$	`\sup`	$\arg$	`\arg`	$\smallint$	`\smallint`
$\arccos$	`\arccos`	$\liminf$	`\liminf`	$\deg$	`\deg`		
$\arctan$	`\arctan`	$\limsup$	`\limsup`	$\asymp$	`\asymp`	$\displaystyle\oint \textstyle\oint$	`\oint`

Put a thin space before differentials: $f(x)\,dx$ (`f(x)\,dx`).

Put negative thin spaces between multiple integrals: $\int\!\!\!\int f(x,y)$ (`\int\!\!\!\int f(x,y)`).

5.6 Logic

$\neg$	`\neg`	$\Rightarrow$	`\Rightarrow`	$\models$	`\models`	$\bigvee \vee$	`\bigvee`
$\lnot$	`\lnot`	$\iff$	`\iff`	$\vdash$	`\vdash`	$\bigwedge \wedge$	`\bigwedge`
$\land$	`\land`	$\exists$	`\exists`	$\dashv$	`\dashv`		
$\lor$	`\lor`	$\forall$	`\forall`	$\bot$	`\bot`	$\aleph$	`\aleph`
				$\top$	`\top`		

5.7 Discrete math

!	`!` (factorial)	$\backslash$	`\backslash`
$\lg$	`\lg`	$\cong$	`\cong`
$\gcd$	`\gcd`	$\equiv$	`\equiv`
$\lfloor\rfloor$	`\lfloor \rfloor`	$a \bmod p = 1$	`a \bmod p = 1` (mod as binary operator)
$\lceil\rceil$	`\lceil \rceil`	$a\equiv 0\pmod{p}$	`a\equiv 0\pmod {p}` (mod as adverb)
$\binom{n}{k}$	`{n\choose k}`	$\prec \succ$	`\prec \succ`
		$\preceq \succeq$	`\preceq \succeq`
$\left\{ n \atop k \right\}$	`{n\brace k}`	$\left(\frac{a}{b}\right)$	`${a \overwithdelims() b}$` (Legendre symbol)
$\left[n \atop k \right]$	`{n\brack k}`	$\left\langle\frac{a}{b}\right\rangle$	`${a \overwithdelims<> b}$` (Eulerian number)

Put a thin space after factorials followed by a letter, digit, or left delimiter: $k!\,(n-k)!$ (`k!\,(n-k)!`)

5.8 Geometry and probability

$\angle$	`\angle`	$\parallel$	`\parallel`	$\Pr$	`\Pr`
$\triangle$	`\triangle`	$\perp$	`\perp`		

5.9 Accents and diacritics

$\hat{x}$	`\hat`	$\dot{x}$	`\dot`	$\acute{x}$	`\acute`	$\overline{w+x}$	* `\overline{w+x}`
$\widehat{x}$	`\widehat`	$\ddot{x}$	`\ddot`	$\grave{x}$	`\grave`	$\underline{w+x}$	* `\underline{w+x}`
$\bar{x}$	`\bar`	$\tilde{x}$	`\tilde`	$\check{x}$	`\check`	$\imath$	`\imath`
$\vec{x}$	`\vec`	$\widetilde{x}$	`\widetilde`	$\breve{x}$	`\breve`	$\jmath$	`\jmath`

`\skew`⟨number⟩⟨accent⟩⟨char⟩	Place accent over character, moved right ⟨number⟩ math units.
* `\mathaccent`⟨mcode⟩⟨formula⟩	Place character as accent over formula.
• `\skewchar`⟨font⟩ `=`⟨ccode⟩	Kern accents as if the accentee were followed by this character.

5.10 Miscellaneous operators and relations

$/$	`\not` (overstrikes next character)	$\dagger$	`\dagger`	$\wedge$	`\wedge`
$\bullet$	`\bullet`	$\ddagger$	`\ddagger`	$\vee$	`\vee`
$\circ$	`\circ`	$\smile$	`\smile`	$\sqsubseteq$	`\sqsubseteq`
$\bigcirc$	`\bigcirc`	$\frown$	`\frown`	$\sqsupseteq$	`\sqsupseteq`
$\triangleleft$	`\triangleleft`	$\oplus$	`\oplus`	$\sqcap$	`\sqcap`
$\triangleright$	`\triangleright`	$\ominus$	`\ominus`	$\sqcup$	`\sqcup`
$\bigtriangledown$	`\bigtriangledown`	$\otimes$	`\otimes`	$\bigwedge$ $\wedge$	`\bigwedge`
$\bigtriangleup$	`\bigtriangleup`	$\odot$	`\odot`	$\bigvee$ $\vee$	`\bigvee`
$\diamond$	`\diamond`	$\oslash$	`\oslash`	$\bigsqcup$ $\sqcup$	`\bigsqcup`
$\bowtie$	`\bowtie`	$\bigoplus$ $\oplus$	`\bigoplus`		
$\uplus$	`\uplus`	$\bigotimes$ $\otimes$	`\bigotimes`		
$\biguplus$ $\uplus$	`\biguplus`	$\bigodot$ $\odot$	`\bigodot`		

`\buildrel`⟨superscript⟩ `\over`⟨relation⟩ Form new relation with superscript:
$\buildrel * \over \longrightarrow$ (`\buildrel \ast \over \longrightarrow`).

5.11 Delimiters and punctuation

$()$	`( )`	$/$	`/`	$\vert$	`|`or `\vert`
$[]$	`[ ]`	$\backslash$	`\backslash`	$\Vert$	`|`or `\Vert`
$[]$	`\lbrack \rbrack`	$\downarrow$	`\downarrow`	$\arrowvert$	`\arrowvert`
$\{\}$	`\{ \}`	$\Downarrow$	`\Downarrow`	$\Arrowvert$	`\Arrowvert`
$\{\}$	`\lbrace \rbrace`	$\uparrow$	`\uparrow`	$\bracevert$	`\bracevert`
$\lfloor\rfloor$	`\lfloor \rfloor`	$\Uparrow$	`\Uparrow`	$\lgroup\rgroup$	`\lgroup \rgroup`
$\lceil\rceil$	`\lceil \rceil`	$\updownarrow$	`\updownarrow`	$\lmoustache\rmoustache$	`\lmoustache \rmoustache`
$\langle\rangle$	`\langle \rangle`	$\Updownarrow$	`\Updownarrow`		
			`.` (null delimiter)		

large delim	`\big`	giant delim	`\Big`	huge delim	`\bigg`	colossal delim	`\Bigg`
left	`\bigl`	left	`\Bigl`	left	`\biggl`	left	`\Biggl`
right	`\bigr`	right	`\Bigr`	right	`\biggr`	right	`\Biggr`
middle	`\bigm`	middle	`\Bigm`	middle	`\biggm`	middle	`\Biggm`
$\big(a\big)\,\big[b\big]\,\big\{c\big\}$		$\Big(a\Big)\,\Big[b\Big]\,\Big\{c\Big\}$		$\bigg(a\bigg)\,\bigg[b\bigg]\,\bigg\{c\bigg\}$		$\Bigg(a\Bigg)\,\Bigg[b\Bigg]\,\Bigg\{c\Bigg\}$	

65,536	`65{,}536`	Comma within number.
$2{\cdot}71828$	`2{\cdot}71828`	Raised decimal point.
$\colon$	`\colon`	Colon as punctuation, instead of operator (`f\colon A \to B`).
$(a_1,\ldots,a_n)$	`(a_1,\ldots,a_n)`	Ellipsis
$b_1\cdots b_n$	`b_1\cdots b_n`	Centered ellipsis
$\ddots$	`\ddots`	Diagonal ellipsis (for matrices)
$\vdots$	`\vdots`	Vertical ellipsis

* `\left`⟨del$_1$⟩⟨formula⟩ `\right`⟨del$_2$⟩ Variable size delimiters.
• `\nulldelimiterspace=`⟨dimen⟩ Width of null delimiter [`1.2pt`].

Put sentence punctuation *outside* text math mode: `Given~$a$, $b$, and~$c$, find their average` ;
but *inside* display math mode: `The answer is $${a+b+c \over 3}.$$`.
Put a thin space after an ellipsis followed by more punctuation:
Let $f(z)=1+z+z^2+\cdots\,$. (`Let $f(z)=1+z+z^2+\cdots\,$.`).

5.12 Pointers

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\downarrow$	`\downarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Downarrow$	`\Downarrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\uparrow$	`\uparrow`
$\to$	`\to`	$\Longrightarrow$	`\Longrightarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\leftrightarrow$	`\leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\leftharpoondown$	`\leftharpoondown`	$\nearrow$	`\nearrow`
$\mapsto$	`\mapsto`	$\leftharpoonup$	`\leftharpoonup`	$\nwarrow$	`\nwarrow`
$\longmapsto$	`\longmapsto`	$\rightharpoondown$	`\rightharpoondown`	$\searrow$	`\searrow`
$\hookleftarrow$	`\hookleftarrow`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\hookrightarrow$	`\hookrightarrow`	$\rightleftharpoons$	`\rightleftharpoons`		

5.13 Character pieces

	`\braceld`	$-$	`\relbar` (extends `\leftarrow`)		`\lhook`
	`\bracelu`	$=$	`\Relbar` (extends `\Leftarrow`)		`\rhook`
	`\bracerd`	$\vert$	`\arrowvert`		`\mapstochar`
	`\braceru`	$\Vert$	`\Arrowvert`	$\surd$	`\surd`

6. Mathematics layout

6.1 Math text

$\frac{1}{1-x}$ — * `{1 \over 1-x}`. Fraction.

${1 \atop 1-x}$ — * `{1 \atop 1-x}`. Vertical pile.

$x \above 2pt y$ — * `{x \above 2pt y}`. Fraction with bar of given thickness.

* `\overwithdelims`⟨del$_1$⟩⟨del$_2$⟩ General fraction surrounded with delimiters.
* `\atopwithdelims`⟨del$_1$⟩⟨del$_2$⟩ General fraction surrounded with delimiters, no fraction bar.
* `\abovewithdelims`⟨del$_1$⟩⟨del$_2$⟩⟨dimen⟩ General fraction surrounded with delimiters, thickness of fraction bar specified.

$\begin{cases} f_1 & \text{first line} \\ f_2 & \text{second line} \end{cases}$ — `$\cases{f_1&first line\cr f_2&second line\cr}$`. Alternatives. First piece assumes math mode, second piece does not. Any number of lines allowed, each ending with `\cr`.

$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ — `\pmatrix{1&1\cr 0&1\cr}`. Two-dimensional array surrounded by parentheses. Any number of rows or columns allowed.

$\begin{array}{cc} & \begin{matrix} u & v \end{matrix} \\ \begin{matrix} u \\ v \end{matrix} & \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \end{array}$ — `\bordermatrix{&u&v\cr u&1&1\cr v&0&1\cr}`. Labelled two-dimensional array surrounded by parentheses.

$\begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix}$ — `\matrix{1&1\cr 0&1\cr}`. General two-dimensional array.

$\overbrace{\langle\text{formula}\rangle}$ — `\overbrace{`⟨formula⟩ `}`

$\underbrace{\langle\text{formula}\rangle}$ — `\underbrace{`⟨formula⟩ `}`

$\overleftarrow{\langle\text{formula}\rangle}$ — `\overleftarrow{`⟨formula⟩ `}`

$\overrightarrow{\langle\text{formula}\rangle}$ — `\overrightarrow{`⟨formula⟩ `}`

* `\limits` Put next super/subscripts over/under previous large operator:

$$\sum_{n>0} f(n) \quad (\texttt{\sum\limits\_\{n>0\}f(n)}).$$

* `\nolimits` Put next super/subscripts beside previous large operator:

$$\sum\nolimits_{n>0} f(n) \quad (\texttt{\sum\nolimits_{n>0}f(n)}).$$

* `\displaylimits` Put limits over operator in display style, beside in text style.

Use `\atop` and `\scriptstyle` for many-line limits:

$$\sum_{\scriptstyle 0<i<n \atop \scriptstyle 0<j<n} A[i,j]$$ (`\sum_{\scriptstyle 0<i<n \atop \scriptstyle 0<j<n} A[i,j]`).

Use `\displaystyle` and `\strut` to get continued fractions:

$$a_0 + {1 \over\displaystyle a_1 + {\strut 1 \over\displaystyle a_2 + {\strut 1 \over a_3}}}$$

`a_0 + {1 \over\displaystyle a_1 + {\strut 1 \over\displaystyle a_2 + {\strut 1 \over a_3}}}`

6.2 Displays

* `\eqno`⟨formula⟩ At end of display mode: Display ⟨formula⟩ as an equation number.
* `\leqno`⟨formula⟩ At end of display mode: Display ⟨formula⟩ as equation number at left.

$$\eqalign{LHS_1&=RHS_1\cr LHS_2&<RHS_2\cr}$$

`\eqalign{LHS_1&=RHS_1\cr LHS_2&<RHS_2\cr}`. Aligned equations. Any number of lines. Produces a vertically centered box.

`\eqalignno{LHS_1& =RHS_1& EQNO_1\cr}` Aligned equations with individual equation numbers. Any number of lines. Produces lines of full display width.

`\leqalignno` Like `\eqalignno`, but puts equation numbers on the left.

`\displaylines{`⟨formula$_1$⟩ `\cr` ⟨formula$_2$⟩ `\cr}` Unaligned formulas, one per line. Any number of lines.

`\openup`⟨dimen⟩ Add ⟨dimen⟩ to distance between lines, usually expressed in terms of `\jot`.

• `\displaywidth=`⟨dimen⟩ Goal width for current display [`\hsize`].
• `\displayindent=`⟨dimen⟩ Indentation for current display [`0pt`].
• `\predisplaysize` Width of line preceding current display.
• `\predisplaypenalty=`⟨penalty⟩ Penalty before a display [`10000`].
• `\postdisplaypenalty=`⟨penalty⟩ Penalty after a display [`0`].
• `\abovedisplayskip=`⟨glue⟩ Glue before a display [`12pt plus 3pt minus 9pt`].
• `\belowdisplayskip=`⟨glue⟩ Glue after a display [`12pt plus 3pt minus 9pt`].
• `\abovedisplayshortskip=`⟨glue⟩ Glue before a display preceded by a short line [`0pt plus 3pt`].
• `\belowdisplayshortskip=`⟨glue⟩ Glue after a display preceded by a short line [`7pt plus 3pt minus 4pt`].

Use `\eqalignno` instead of `\eqalign` when (1) equation numbers apply to one line rather than the whole display, (2) page breaks are allowed between lines, or (3) short lines of text come between aligned lines of display, as in `\eqalignno{x&=y\cr \noalign{\hbox{and}} y&=z\cr}`

Set `\postdisplaypenalty` to `-10000` inside a display to force it to appear at the bottom of the page.

6.3 Fonts, styles, and families

* `\displaystyle` Switch to display style.
* `\textstyle` Switch to text style.
* `\scriptstyle` Switch to script style.
* `\scriptscriptstyle` Switch to scriptscript style.
* `\mathchoice{`⟨display⟩ `}{`⟨text⟩ `}{`⟨script⟩ `}{`⟨scriptscript⟩ `}` Use alternative formula.

`\mathpalette\cs{`⟨stuff⟩ `}` Invoke `\cs` with first arg = current style selection (e.g., `\displaystyle`), second arg = ⟨stuff⟩.

* `\textfont`⟨family⟩ `=`⟨font⟩ Define text font of family.
* `\scriptfont`⟨family⟩ `=`⟨font⟩ Define subscript font of family.
* `\scriptscriptfont`⟨family⟩ `=`⟨font⟩ Defint subsubsript font of family.

`\newfam\cs` Allocate symbolic family.

• `\fam=`⟨number⟩ Select family.

`\rm`	Switch to roman: Ax = b + error.
`\bf`	Switch to bold: **Ax = b + error**.
`\it`	Switch to text italic: *Ax = b + error* (text size only).
`\sl`	Switch to slanted: *Ax = b + error* (text size only).
`\tt`	Switch to typewriter: `Ax = b + error` (text size only).
`\bffam`	Family for bold.
`\itfam`	Family for text italic.
`\slfam`	Family for slanted roman.
`\ttfam`	Family for typewriter.
`\tenex`	Ten-point extension font.
`\tensy`	Ten-point symbols font.

7. Mathematics constructions

* `\vcenter` [`to` ⟨dimen⟩ / `spread` ⟨dimen⟩] `{`⟨vertical mode⟩ `}`	Create vertical box centered around axis.
`\mathstrut`	Invisible box, forcing height and depth of left paren.
• `\everymath={`⟨tokens⟩ `}`	Insert ⟨tokens⟩ before every formula.
• `\everydisplay={`⟨tokens⟩ `}`	Insert ⟨tokens⟩ before every display.
• `\relpenalty=`⟨penalty⟩	Penalty for breaking line after Rel atom [`500`]
• `\binoppenalty=`⟨penalty⟩	Penalty for breaking line after Bin atom [`700`]

7.1 White space

`\,`	thin space (before dx, units, some left parens; after factorials)
`\>`	medium space
`\;`	thick space
`\!`	negative thin space
* `\mskip`⟨muglue⟩	Explicit math glue (breakable).
* `\mkern`⟨mudimen⟩	Explicit math space (unbreakable).
• `\mathsurround=`⟨glue⟩	Spacing before and after inline formulas [`0pt`].
• `\thinmuskip=`⟨muglue⟩	Thin space amount [`3mu`]
• `\medmuskip=`⟨muglue⟩	Medium space amount [`4mu plus 2mu minus 4mu`]
• `\thickmuskip=`⟨muglue⟩	Thick space amount [`5mu plus 5mu`]
* `\scriptspace=`⟨dimen⟩	Extra space after a sub/superscript [`0.5pt`].
* `\nonscript`	Next glue or kern ignored in script or scriptscript style.

7.2 Packaging

* `\mathord{`⟨formula⟩ `}`	Ordinary character.
* `\mathbin{`⟨formula⟩ `}`	Binary operation.
* `\mathop{`⟨formula⟩ `}`	Large operator.
* `\mathrel{`⟨formula⟩ `}`	Relation.
* `\mathopen{`⟨formula⟩ `}`	Opening.
* `\mathclose{`⟨formula⟩ `}`	Closing.
* `\mathpunct{`⟨formula⟩ `}`	Punctuation.
* `\mathinner{`⟨formula⟩ `}`	Formula.

* `\radical`⟨dcode⟩⟨formula⟩	Form radical using given delimiter code.
* `\mathaccent`⟨mcode⟩⟨formula⟩	Place character as accent over formula.
• `\delcode`⟨ccode⟩ `=`⟨dcode⟩	Define small and large delimiter codes for character.
* `\delimiter`⟨dcode⟩	Use explicit delimiter code.

8. Tables

8.1 Tabbing

`\settabs` { ⟨number⟩ `columns` / `\+`⟨sample line⟩ `\cr` }	Set tab stops in equally spaced columns, or as in sample line.
`\+`⟨text$_1$⟩ `&`⟨text$_2$⟩ `\cr`	Print line with tabs; successive ⟨text⟩s start at successive tab stops. Any number of ⟨text⟩s. A `&` after existing tabs sets a new tab stop.
`\tabalign`	Like `\+`, except permissible in “inner” contexts.
`\cleartabs`	Erase tab stops to the right of current stop.
`\tabs`	Register number of tab box.

Normal tab fields are flushed left. To flush a field right, say `\hfill`⟨text⟩ ; to center a field, say `\hfill`⟨text⟩ `\hfill`.

8.2 Alignment

* \halign [to ⟨dimen⟩ / spread ⟨dimen⟩] {⟨preamble⟩ \cr ⟨body⟩ }	Form table from body with column templates given in preamble. In body, columns are separated by &, each row ends with \cr.
* \valign [to ⟨dimen⟩ / spread ⟨dimen⟩] {⟨preamble⟩ \cr ⟨body⟩ }	Form table in vertical mode from body, with row templates given in preamble. In body, rows are separated by &, each column ends with \cr.
&	Column (row) separator.
* \cr	End of aligned row (column).
\endline	Synonym for \cr.
* \crcr	Force \cr.
* &&	In preamble, or & at beginning of preamble: Cycle through remaining templates as often as needed.
* \noalign{⟨contents⟩ }	Insert contents between rows (columns) of \halign (\valign).
• \tabskip=⟨glue⟩	Glue between columns (rows).
* \omit	As first token in a column: Use # instead of template from the preamble.
* \span	In body (in place of &): Merge adjacent columns. In preamble: Expand the next token while reading preamble.
\multispan⟨number⟩	Merge ⟨number⟩ columns without their templates.
\hidewidth⟨entry⟩	Allow entry to stick out of column at the left.
⟨entry⟩ \hidewidth	Allow entry to stick out of column at the right.
• \everycr={⟨tokens⟩ }	Insert tokens after every \cr.
\oalign{⟨char$_1$⟩ \cr⟨char$_2$⟩ }	Put characters over each other.
\ooalign{⟨char$_1$⟩ \cr⟨char$_2$⟩ }	Superimpose characters.

To make tables with butting \vrules in each row, use \offinterlineskip (to butt rows together) and \strut (to make height of rows the same).

9. Figures and footnotes

\footnote⟨mark⟩ {⟨text⟩ }	Print reference mark, and insert footnote at bottom of page.
\vfootnote⟨mark⟩ {⟨text⟩ }	Insert footnote at bottom of page. Use near an insertion to generate a footnote to the insertion.
\footnoterule	Line separating page from footnotes [\hrule width 2truein, raised 3pt].
\midinsert⟨vertical mode⟩ \endinsert	Insert nearby.
\topinsert⟨vertical mode⟩ \endinsert	Insert at top of nearby page.
\pageinsert⟨vertical mode⟩ \endinsert	Insert a full page nearby.
\footins	Insertion class for footnotes.
\topins	Insertion class for top insertions.
\newinsert\cs	Allocate symbolic insertion class.
* \insert⟨number⟩	Insertion class.
• \floatingpenalty=⟨penalty⟩	Penalty for splitting insertions across page [0].

10. Computer programs

$:=$	:=
$\leftarrow$	\gets

Use text italic for identifiers, boldface for keywords, math mode for expressions.

One way to get indenting using tabs, indenting to fit text:

	`\cleartabs  % get rid of previous tab stops.`
for $k \gets 1$ **to** n	`\+{\bf for} $k\gets 1$ {\bf to} $n$ \cr`
do	`\+ \quad& {\bf do} \cleartabs& \cr % reset indent for do-loop`
value $\gets f(k)$	`\+&& ${\it value} \gets f(k)$ \cr`
if *value* $>$ *max* **then**	`\+&& {\bf if} \cleartabs& ${\it value} > {\it max}$ {\bf then} \cr`
max $\gets$ *value*	`\+&&& ${\it max} \gets {\it value}$ \cr`
index $\gets k$	`\+&&& ${\it index} \gets k$ \cr`
fi	`\+&& {\bf fi} \cr`
od	`\+& {\bf od} \cr`

Another way, using `\leftskip` with fixed indentation:

	`\newdimen\progindent \progindent=20pt`
	`\def\bumpindent{\advance\leftskip by \progindent}`
	`\def\exdent{\advance\leftskip by -\progindent}`
	`{\obeylines \everypar={\hangindent 2\progindent}`
for $k \gets 1$ **to** n	`{\bf for} $k \gets 1$ {\bf to} $n$`
do	`    \bumpindent {\bf do}`
value $\gets f(k)$	`        \bumpindent ${\it value} \gets f(k)$`
if *value* $>$ *max* **then**	`        {\bf if} ${\it value} > {\it max}$ {\bf then}`
max $\gets$ *value*	`            \bumpindent ${\it max} \gets {\it value}$`
index $\gets k$	`            ${\it index} \gets k$`
fi	`        \exdent {\bf fi}`
od	`    \exdent {\bf od}`
	`\exdent`
	`} % turn off \obeylines`

11. Macros

11.1 Definitions

* `\def\cs`⟨parameter text⟩ `{`⟨replacement text⟩ `}`	Define macro `\cs`.
* `\gdef`	Like `\def`, but definition is global.
* `\edef`	Like `\def`, but expands replacement text before definition.
* `\xdef`	Like `\gdef`, but expands replacement text before definition.
* `\long`	Next-defined macro allows its arguments to include `\par`.
* `\outer`	Next-defined macro cannot appear as an argument, in a definition, in an alignment preamble, or in conditional text.
`#`n	Parameter number n (in parameter or replacement text).
`##`	In replacement text, yields a single `#`.
`#{`	At end of parameter text, delimits last argument by `{` and yields `{` at the end of replacement text.
* `\let\cs=`⟨token⟩	Give `\cs` the token's current meaning.
* `\futurelet\cs`$\langle\text{token}_1\rangle\langle\text{token}_2\rangle$	Give `\cs` the second token's meaning, then read $\langle\text{token}_1\rangle\langle\text{token}_2\rangle$.

11.2 Scoping and expansion

`{}`	Block structure.
`\bgroup \egroup`	Implicit `{}`.
* `\begingroup \endgroup`	Alternative block structure.
* `\global`	Next command affects all blocks.
* `\afterassignment`⟨token⟩	Save token and reread it after the next assignment. If the assignment is a `\setbox`, reread the token as first token of `\hbox`, `\vbox`, or `\vtop`.
* `\aftergroup`⟨token⟩	Save token and reread it after leaving the current group.

* `\expandafter`⟨token$_1$⟩⟨token$_2$⟩	Expand token$_2$, then read token$_1$ followed by the expansion.
* `\noexpand`⟨token⟩	Don't expand token.
* `\ignorespaces`	Ignore input tokens until a non-white one.
* `\relax`	Do nothing.
`\empty`	Macro that expands to nothing.

11.3 Control flow

* `\`⟨if-command⟩⟨true text⟩ `\else`⟨false text⟩ `\fi`	Conditional. Else part (`\else`⟨false text⟩) is optional.
* `\ifnum`⟨number$_1$⟩⟨relation⟩⟨number$_2$⟩	Numeric comparison; ⟨relation⟩ is `<`, `=`, or `>`.
* `\ifodd`⟨number⟩	True if ⟨number⟩ is odd.
* `\ifcase`⟨number⟩⟨text$_1$⟩ `\or`⟨text$_2$⟩ `\else`⟨default text⟩ `\fi`	Multi-way conditional. The ⟨number⟩ selects the text to use. Any number of cases allowed.
* `\ifdim`⟨dimen$_1$⟩⟨relation⟩⟨dimen$_2$⟩	Numeric comparison of dimensions; ⟨relation⟩ is `<`, `=`, or `>`.
* `\ifx`	True if next two tokens (without expansion) are either (a) macros with same parameters and definitions, or (b) non-macros with same character codes and same category codes.
* `\if`	True if next two tokens (after expanding macros) have same character codes. Unexpandable control sequence tokens (like `\relax`) have character code 256.
* `\ifcat`	True if next two tokens (after expanding macros) have same category codes. Unexpandable control sequence tokens (like `\relax`) have category code 16.
* `\ifhbox`⟨number⟩	True if `\box`⟨number⟩ is an hbox.
* `\ifvbox`⟨number⟩	True if `\box`⟨number⟩ is a vbox.
* `\ifvoid`⟨number⟩	True if `\box`⟨number⟩ is void.
* `\ifhmode`	True if TeX is in horizontal or restricted horizontal mode.
* `\ifvmode`	True if TeX is in vertical or internal vertical mode.
* `\ifmmode`	True if TeX is in math or display math mode.
* `\ifinner`	True if TeX is in restricted horizontal, internal vertical or text math mode.
* `\ifeof`⟨stream⟩	True if stream cannot supply more tokens.
* `\iftrue`	Always true.
* `\iffalse`	Always false.
`\newif\iffoo`	Define switch and three macros: `\footrue` turns switch on, `\foofalse` turns switch off, `\iffoo` tests switch. Use any word instead of `foo`.
`\loop`⟨text$_1$⟩⟨if-command⟩⟨text$_2$⟩ `\repeat`	Iterate commands in the two texts until the `\if` fails.

11.4 Numeric registers

`\newcount\cs`	Allocate symbolic count register.
`\newdimen\cs`	Allocate symbolic dimen register.
`\newskip\cs`	Allocate symbolic skip register.
`\newmuskip\cs`	Allocate symbolic math glue register.
* `\count`⟨number⟩	Count register.
* `\dimen`⟨number⟩	Dimension register.
* `\skip`⟨number⟩	Glue register.
* `\muskip`⟨number⟩	Math glue register.
* `\countdef\cs=`⟨number⟩	Define symbolic count register.
* `\dimendef\cs=`⟨number⟩	Define symbolic dimen register.
* `\skipdef\cs=`⟨number⟩	Define symbolic skip register.
* `\muskipdef\cs=`⟨number⟩	Define symbolic math glue register.
* `\advance`⟨register⟩ `by` ⟨number⟩	Add to register.
* `\multiply`⟨register⟩ `by` ⟨number⟩	Multiply register.
* `\divide`⟨register⟩ `by` ⟨number⟩	Divide register.

\count0 holds the page number.
\count0 through \count9 are logged and displayed on the terminal when page is shipped out.

11.5 Box registers

\newbox\cs	Allocate symbolic box register.
* \setbox⟨breg⟩ =⟨box⟩	Assign to box register.
* \box⟨breg⟩	Use box register.
* \unhbox⟨breg⟩	Use box register without outer level of horizontal boxing.
* \unvbox⟨breg⟩	Use box register without outer level of vertical boxing.
* \copy⟨breg⟩	Copy box register.
* \unhcopy⟨breg⟩	Copy box register without outer level of horizontal boxing.
* \unvcopy⟨breg⟩	Copy box register without outer level of vertical boxing.

11.6 Converting tokens, variables, strings, csnames, etc

• \day	Day of month (a ⟨number⟩).
• \month	Month of year (a ⟨number⟩).
• \year	Current year (a ⟨number⟩).
• \time	Time of day (a ⟨number⟩), in minutes since midnight.
* \uppercase{⟨tokens⟩ }	Convert tokens to upper case.
* \lowercase{⟨tokens⟩ }	Convert tokens to lower case.
* \number⟨number⟩	Convert number to arabic digits.
* \romannumeral⟨number⟩	Convert number to lowercase roman numerals.
• \fontname⟨font⟩	Name of font, including scaling.
• \jobname	Name of TEX job, used to name .dvi and .log files.
* \csname⟨tokens⟩ \endcsname	Expand tokens and convert to control sequence.
* \string\cs	Convert name of \cs to token list.
• \escapechar=⟨ccode⟩	Code used when converting \cs to tokens [‘\\].
* \the⟨internal quantity⟩	Token list describing a TEX variable.
\newtoks\cs	Allocate symbolic token list.
* \toks⟨number⟩	Token list register.
* \toksdef\cs=⟨number⟩	Define symbolic token list register.

12. Page design

12.1 Size

• \hsize=⟨dimen⟩	Define width of lines.
• \vsize=⟨dimen⟩	Define height of pages.
• \hoffset=⟨dimen⟩	Move page image to the right.
• \voffset=⟨dimen⟩	Move page image down.
\magnification=⟨number⟩	Enlarge document by factor of ⟨number⟩/1000.
\magstep⟨number⟩	$\lfloor 1000 * 1.2^{\langle\text{number}\rangle} \rfloor$, for use in \magnification and scaled size of fonts.
\magstephalf	$\lfloor 1000 * \sqrt{1.2} \rfloor$, likewise.
• \mag=⟨number⟩	Global magnification factor $*\,1000$ [1000].

12.2 Headers, footers, page numbers

\headline={⟨contents⟩ }	Print contents at top of every page.
\footline={⟨contents⟩ }	Print contents at bottom of every page.
• \pageno	Page number.
\folio	Page number in arabic if \pageno ≥ 0, in roman numerals if \pageno < 0.
\nopagenumbers	Don’t print page numbers.
\advancepageno	Increase magnitude of page number.
• \topskip=⟨glue⟩	Glue before first box of each page [10pt].
\normalbottom	Move last line to bottom of page [on].
\raggedbottom	Don’t move last line to bottom of page [off].

Increase `\voffset` for nonempty `\headline` to keep 1in margin at top.

12.3 Marks

* `\mark{`⟨contents⟩`}`	Put mark into page.
* `\topmark`	Last mark seen before current page.
* `\firstmark`	First mark on page if any; if none, same as `\topmark`.
* `\botmark`	Last mark seen up to end of page.
* `\splitfirstmark`	First mark in most recent `\vsplit`.
* `\splitbotmark`	Last mark in most recent `\vsplit`.

Put `\mark`s after boxes, not before.

12.4 Output routines

* `\output={`⟨tokens⟩`}`	Define output routine.
`\pagebody`	Box containing page contents.
`\pagecontents`	Vertical list of page, including insertions.
* `\outputpenalty`	Penalty observed at page break.
`\box255`	In output routine: Contents of page.
* `\shipout`⟨box⟩	Send box to `DVI` file.
• `\deadcycles=`⟨number⟩	Number of `\output`s since last `\shipout`.
• `\maxdeadcycles=`⟨number⟩	Error if `\deadcycles` exceeds this [25].

12.5 Page in progress

• `\pagetotal=`⟨dimen⟩	Height of current page so far.
• `\pagegoal=`⟨dimen⟩	Page height goal.
• `\pagestretch=`⟨dimen⟩	Amount of normal stretch on page so far.
• `\pagefilstretch=`⟨dimen⟩	Amount of infinite stretch on page so far.
• `\pagefillstretch=`⟨dimen⟩	Amount of doubly infinite stretch on page so far.
• `\pagefilllstretch=`⟨dimen⟩	Amount of triply infinite stretch on page so far.
• `\pageshrink=`⟨dimen⟩	Amount of shrink on page so far.
• `\pagedepth=`⟨dimen⟩	Depth of page so far.
• `\insertpenalties=`⟨number⟩	Penalty due to insertions on page so far.

13. Controlling TEX

13.1 Starting, stopping, reading files

* `\input`⟨file⟩	Read from file.
* `\endinput`	Stop reading from current file.
⟨blank line⟩	Generate `\par`, ending paragraph.
`\obeylines`	End of input line generates `\par`.
`\obeyspaces`	Don't ignore consecutive spaces between words.
`\bye`	End the document.
* `\end`	Stop processing.
* `\everyjob=`⟨tokens⟩	Read tokens at start of every document.

13.2 Auxiliary files

`\newread\cs`	Allocate symbolic input stream.
* `\openin`⟨stream⟩ `=`⟨file⟩	Prepare to read file through a stream.
* `\read`⟨stream⟩ `to \cs`	Define `\cs` to be a macro without parameters whose replacement text is next line of stream (or more if necessary to balance braces). If stream isn't open, use terminal (suppressing prompt if stream is negative).
* `\closein`⟨stream⟩	Stop reading from stream.
`\newwrite\cs`	Allocate symbolic output stream.
* `\openout`⟨stream⟩ `=`⟨file⟩	Prepare to write to file through a stream.

* `\write`⟨stream⟩ `{`⟨tokens⟩ `}`	Expand and append tokens to the stream, followed by a newline. Doesn't actually happen until page is shipped out (unless preceded by `\immediate`).
* `\closeout`⟨stream⟩	Stop writing to stream.
* `\immediate`	Do next stream operation now, instead of waiting for `\shipout`.
`\wlog{`⟨tokens⟩ `}`	Write tokens in log file (only).
* `\endlinechar=`⟨ccode⟩	Append this char to every input line.
* `\newlinechar=`⟨ccode⟩	Character to use for newlines in `\write` [`‘\^^J`].
* `\special{`⟨tokens⟩ `}`	Expand tokens and send them to `DVI` file at the current position on the page.

13.3 Reserved characters

`\`	Escape character	`$`	Math mode shift
`{}`	Grouping	`^`	Superscript
`%`	Comment	`_`	Subscript
`~`	Tie	`#`	Parameter
		`&`	Alignment tab

`\active`	Category 13, one-character macro.
* `\catcode`⟨ccode⟩ `=`⟨category⟩	Assign category to character

13.4 Interaction

* `\message{`⟨tokens⟩ `}`	Print message on terminal (and in log file).
* `\errmessage{`⟨tokens⟩ `}`	Print error message on terminal (and in log file).
• `\newhelp\cs{`⟨tokens⟩ `}`	Define symbolic help message.
* `\errhelp=\cs`	Define help message to print after `\errmessage`, if user asks for it.

If prompt is `?`:

`i`⟨text⟩	Insert text.
⟨number⟩	Delete ⟨number⟩ tokens.
⟨return⟩	Try to recover.
`x`	Quit.
`e`	Switch to editor.
`h`	Print help message.
`?`	Print help message.
`s`	Don't stop for TEX errors.
`r`	Don't stop for file errors either.
`q`	Don't even print errors on screen.

If prompt is `*`:

⟨anything⟩	Process.

If prompt is `**`:

`\cs`⟨anything⟩	Start processing.
⟨file name⟩	Do `\input` from given file.

In TEX source:

* `\errorstopmode`	Stop for TEX errors [`on`].
* `\scrollmode`	Don't stop for TEX errors.
* `\nonstopmode`	Don't stop for file errors.
* `\batchmode`	Don't print errors on screen.

13.5 Dimensions and Glue

`in`	inch	1.4 in
`cm`	centimeter	3 cm
`mm`	millimeter	35 mm
`pt`	printer's point	100 pt
`bp`	big point	100 bp
`pc`	pica	8 pc
`sp`	scaled point	7000000 sp
`dd`	didôt point	100 dd
`cc`	cicero	8 cc

`em`	Quad width of current font.
`ex`	x-height of current font.
`mu`	1/18 em of math symbol font.
`\jot`	Incremental baseline distance [`3pt`].
• `\ht`⟨breg⟩	Height of box register.
• `\dp`⟨breg⟩	Depth of box register.
• `\wd`⟨breg⟩	Width of box register.
`\maxdimen`	Largest dimension [`16383.99999pt`]
`true`	Prefix to unit to bypass magnification.

`fil`	Infinite stretch or shrink.
`fill`	Second order infinite stretch or shrink.
`filll`	Third order infinite stretch or shrink.

13.6 Debugging

`\ddt`	Cause an error (for debugging).
* `\show`⟨token⟩	Display definition of token.
* `\showthe`⟨internal quantity⟩	Display TEX variable.
* `\showbox`⟨number⟩	Display contents of box register.
* `\showlists`	Display current vertical and horizontal lists.
• `\showboxbreadth=`⟨number⟩	Number of components displayed for boxes [`5`].
• `\showboxdepth=`⟨number⟩	Deepest nesting level displayed for boxes [`3`].
`\showhyphens{`⟨text⟩ `}`	Display where hyphens could occur in text.
* `\meaning`⟨token⟩	Meaning of token, in human readable form.
* `\tracingonline=`⟨number⟩	Positive displays logged data.
* `\tracingmacros=`⟨number⟩	Positive logs macro expansions. Greater than 2 also logs dæmon macros like `\output` and `\everypar`.
* `\tracingpages=`⟨number⟩	Positive logs page breaking data.
* `\tracingparagraphs=`⟨number⟩	Positive logs line-breaking data.
* `\tracingoutput=`⟨number⟩	Positive logs boxes as they are shipped out.
* `\tracinglostchars=`⟨number⟩	Positive logs missing characters in fonts.
* `\tracingrestores=`⟨number⟩	Positive logs block-exit actions.
* `\tracingstats=`⟨number⟩	Positive logs memory usage at end of job.
* `\tracingcommands=`⟨number⟩	Positive logs every command. Greater than 2 also logs conditionals.
`\tracingall`	Turn on all tracing.
* `\hbadness=`⟨number⟩	Badness threshold for reporting bad hboxes.
• `\pausing=`⟨number⟩	Positive displays each line from input files for confirmation or editing.

14. Legend

*	(Before a control sequence): TEX primitive command.
•	(Before a control sequence): TEX variable or parameter.
$\begin{bmatrix}\text{option}_1\\ \text{option}_2\end{bmatrix}$	Optional material; use zero or one of the given options.
$\begin{Bmatrix}\text{alternative}_1\\ \text{alternative}_2\end{Bmatrix}$	Alternatives; use one of the given options.
$\begin{bmatrix}\text{option}_1\\ \text{option}_2\\ \text{option}_3\end{bmatrix}^*$	Optional repetition; use zero or more of the given options in any order.
⟨badness⟩	Integer measuring amount of stretch or shrink in a line.
⟨box⟩	A `\hbox`, `\vbox`, `\vtop`, `\copy`, `\box`, `\vsplit`, or `\lastbox`.
⟨breg⟩	Box register, usually a symbolic named (defined by `\newbox`).
⟨ccode⟩	Character code, usually given by quoting a character [`` `\a``], or in octal [`'40`].
⟨contents⟩	Text, control sequences, and whatever.
⟨dcode⟩	Delimiter code, usually given in hexadecimal [`"123456`], meaning small version is in family `1` position `"23`, large version is in family `4` position `"56`.
⟨demerit⟩	An integer measuring desirability of line breaking decisions within a paragraph.
⟨dimen⟩	An optional factor and a dimension unit [`.5\vsize`].
⟨external name⟩	Name of font, including size [`cmr10`].
⟨family⟩	A number between `0` and `15`, usually a symbolic name (defined by `\newfam`). A family of math fonts describes a `\textfont`, a `\scriptfont`, and a `\displayfont`. TEX expects roman in family `0`, math italic in family `1`, symbols in family `2`, and extensible characters in family `3`.
⟨font⟩	A symbolic font name (defined by `\font\cs=`$\cdots$), or `\font` (denoting the current font).
⟨glue⟩	A ⟨dimen⟩ with optional stretch [`plus 1em`] and shrink [`minus 3pt`] components.
⟨internal quantity⟩	A TEX variable, ⟨register⟩, or parameter.
⟨magnification⟩	1000 times the magnification factor, usually given as `\magstep`n or `\magstephalf`.

⟨mcode⟩	Math code, usually given in hexadecimal [`"1234`], meaning class `1`, family `2`, position `"34`.
⟨mudimen⟩	A math dimenson, like ⟨dimen⟩ except units must be `mu` (or something which is defined in terms of `mu`).
⟨muglue⟩	A math glue, like ⟨glue⟩ except units must be `mu` (or something which is defined in terms of `mu`).
⟨number⟩	Explicit integer [`25`], or ⟨register⟩ [`\pageno`], or variable [`\year`], or (by coercion) ⟨dimen⟩ [`\hsize`], or ⟨glue⟩ (`\parskip`].
⟨parameter text⟩	Template for finding arguments to a macro by pattern matching, where tokens in parameter text match themselves except `#`n, which matches any string and defines the n-th argument.
⟨penalty⟩	An integer giving desirability of break, from `-10000` (required) to `0` (neutral) to `10000` (forbidden).
⟨register⟩	A `\count`, `\skip`, `\dimen`, or `\muskip` register, usually a symbolic name (defined by `\newcount`, `\newskip`, `\newdimen`, or `\newmuskip`).
⟨replacement text⟩	Result of macro expansion, where `#`n is replaced by the n-th argument.
⟨sfactor⟩	Space factor, an integer roughly 1000 times the amount to multiply the next interword space.
⟨stream⟩	Stream number, usually a symbolic name (defined with `\newread` or `\newwrite`), or `16` for the terminal (and log file), or `-1` for the log file.
⟨tokens⟩	A list of tokens.
⟨vertical mode⟩	Material to be made into a vertical list.

TEX Reference Card

Samuel W. Bent
Dartmouth College

(revised March 31, 1987)

This guide is intended to remind the TEX user what the control sequences of plain TEX do. See the legend to find out the meanings of some of the notation used herein. For more information, see the TEXbook.