

Just Enough L^AT_EX

Clifford E. Weil¹

¹Department of Mathematics
Michigan State University
East Lansing, MI USA
`weil@math.msu.edu`

September 17, 2008

Objectives

The objectives for this short course are:

- ▶ to be able to prepare a simple \LaTeX document
- ▶ to be able to read a \LaTeX manual

First you will need a \TeX implementation on your computer.

Tex Implementations

Mac users: If you're running OS X, go to www.uoregon.edu/~koch/texshop. Follow the instructions for downloading and installing TeXShop.

Windows users: You'll need a T_EX editor and a T_EX compiler. For an editor go to www.toolscenter.org and download Texnik Center. For a T_EX compiler go to www.miktex.org and download MikTeX. Some Windows users prefer WinEdit, a commercial product that can be obtained from www.winedt.com

What is L^AT_EX?

L^AT_EX is many macros for T_EX.

T_EX is a professional typesetting computer program, developed by a computer scientist, Donald Knuth.

T_EX compiles a user-prepared source file to produce the typeset document. The source file contains text and commands.

Commands:

- ▶ enhance text

What is L^AT_EX?

L^AT_EX is many macros for T_EX.

T_EX is a professional typesetting computer program, developed by a computer scientist, Donald Knuth.

T_EX compiles a user-prepared source file to produce the typeset document. The source file contains text and commands.

Commands:

- ▶ enhance text
- ▶ produce special characters

What is \LaTeX ?

\LaTeX is many macros for \TeX .

\TeX is a professional typesetting computer program, developed by a computer scientist, Donald Knuth.

\TeX compiles a user-prepared source file to produce the typeset document. The source file contains text and commands.

Commands:

- ▶ enhance text
- ▶ produce special characters
- ▶ produce symbols

What is L^AT_EX?

L^AT_EX is many macros for T_EX.

T_EX is a professional typesetting computer program, developed by a computer scientist, Donald Knuth.

T_EX compiles a user-prepared source file to produce the typeset document. The source file contains text and commands.

Commands:

- ▶ enhance text
- ▶ produce special characters
- ▶ produce symbols
- ▶ design new commands

What is L^AT_EX?

L^AT_EX is many macros for T_EX.

T_EX is a professional typesetting computer program, developed by a computer scientist, Donald Knuth.

T_EX compiles a user-prepared source file to produce the typeset document. The source file contains text and commands.

Commands:

- ▶ enhance text
- ▶ produce special characters
- ▶ produce symbols
- ▶ design new commands

L^AT_EX, by Leslie Lamport, makes T_EX user-friendly.

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$\$$	changes to displayed math mode

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$$ $\$$	changes to displayed math mode
$_$	in math mode subscript marker

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$$ $\$$	changes to displayed math mode
$_$	in math mode subscript marker
\hat	in math mode superscript marker

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$$ $\$$	changes to displayed math mode
$_$	in math mode subscript marker
\hat	in math mode superscript marker
$\&$	alignment character

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$$ $\$$	changes to displayed math mode
$_$	in math mode subscript marker
\hat	in math mode superscript marker
$\&$	alignment character
\sim	sticky space

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$\$$	changes to displayed math mode
$_$	in math mode subscript marker
\hat	in math mode superscript marker
$\&$	alignment character
\sim	sticky space
$\%$	Comment out character

Altered Characters

The keystrokes \backslash , $\$$, $_$, \hat , $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and \hat can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$\$$	changes to displayed math mode
$_$	in math mode subscript marker
\hat	in math mode superscript marker
$\&$	alignment character
\sim	sticky space
$\%$	Comment out character
$\{, \}$	argument delimiters

Altered Characters

The keystrokes \backslash , $\$$, $_$, $\^$, $\&$, \sim , $\%$, $\{$, $\}$, and $\#$ don't produce text.

The commands $_$ and $\^$ can only be used in math mode.

Command	use
\backslash	indicates a command
$\$$	changes to in-line math mode
$\$$ $\$$	changes to displayed math mode
$_$	in math mode subscript marker
$\^$	in math mode superscript marker
$\&$	alignment character
\sim	sticky space
$\%$	Comment out character
$\{$, $\}$	argument delimiters
$\#$	used in designing new commands

Producing the Altered Characters

To get	Type
<code>\</code>	<code>\backslash</code>
<code>\$</code>	<code>\\$</code>
<code>-</code>	<code>_</code>
<code>^</code>	<code>\^{} </code>
<code>&</code>	<code>\&</code>
<code>~</code>	<code>\~{} </code>
<code>%</code>	<code>\%</code>
<code>{</code>	<code>\{</code>
<code>}</code>	<code>\}</code>
<code>#</code>	<code>\#</code>

Commands

Commands begin with a `\`. The simplest kind of command is a backslash followed by a non-alphabetic character; for example `\$`, `_`, `\&`, `\%`, `\{`, `\}`, `\#`, `\:`, `\;`, `\!`, `\|` and `\,`. These examples produce something once they have been entered. Others must be followed immediately by a character to which the command applies. For example

To get	Type	To get	Type	To get	Type
ò	<code>\'o</code>	ó	<code>\'o</code>	ô	<code>\^o</code>
ö	<code>\"o</code>	ō	<code>\=o</code>	ô	<code>\.o</code>
õ	<code>\~o</code>				

Still other examples are `\-`, `\|`, `\>`.

More on Commands

A second type of simple command is a backslash followed by one or more alphabetic characters. For example, Greek letters: α , β , etc. Math mode is necessary. Other examples are \sin , \cos , \exp , and all mathematical symbols. A very important one is `\dots` which produces three dots. Three periods gets ... but typing `\dots` gets (Note that math mode isn't needed.)

Also of special interest are `\i`, `\j` and `\ell`.

Commands with Arguments

These are commands that require arguments; mandatory ones, enclosed in a pair of braces $\{ \}$ and also possibly optional ones, enclosed in a pair of brackets, $[]$. Additional accents use such commands.

To get	Type	To get	Type	To get	Type
ö	<code>\u{o}</code>	ø	<code>\v{o}</code>	ő	<code>\H{o}</code>
ø	<code>\c{o}</code>	⊙	<code>\b{o}</code>	ö	<code>\r{o}</code>

Many of the above accents are created with different commands in math mode.

To get	Type	Type	To get	Type	
\acute{a}	<code>\\$acute{a}\$</code>	\bar{a}	<code>\\$bar{a}\$</code>	\breve{a}	<code>\\$breve{a}\$</code>
\check{a}	<code>\\$check{a}\$</code>	\grave{a}	<code>\\$grave{a}\$</code>	\mathring{a}	<code>\\$mathring{a}\$</code>
\tilde{a}	<code>\\$tilde{a}\$</code>	\hat{a}	<code>\\$hat{a}\$</code>	\vec{a}	<code>\\$vec{a}\$</code>
\widetilde{abc}	<code>\\$widetilde{abc}\$</code>	\widehat{abc}	<code>\\$widehat{abc}\$</code>	\dot{a}	<code>\\$dot{a}\$</code>
\ddot{a}	<code>\\$ddot{a}\$</code>	\ddot{a}	<code>\\$dddot{a}\$</code>	$\dot{\dot{a}}$	<code>\\$ddddot{a}\$</code>

New Commands

L^AT_EX provides commands to produce new commands.

- ▶ `\newcommand{\name}{the command}`
- ▶ `\DeclareMathOperator`

Examples: `\newcommand{\e}{\ensuremath{\epsilon}}`.

Typing `\e` produces ϵ .

`\DeclareMathOperator{\cl}{cl}`. Typing `$\cl A$` produces $cl A$.

Some Typing Peculiarities

- ▶ The space bar in text mode and in math mode
- ▶ The return key or carriage return
- ▶ After some commands
- ▶ The hyphen key
- ▶ ".