

Hello world! This is the beginning of L^AT_EX!

We can start a new paragraph at any time by leaving a blank line. Not that there are many special characters in L^AT_EX or just T_EX. In particular &, %, #, -, {, }, carets and tildes. and \ are all special characters – if L^AT_EX sees any of these without a preceding backslash, it thinks they are commands.

A macro is just a markup command.

Typesetting Mathematics!

Donnie Knuth is the creator of T_EX. To typeset mathematics and it is very good at it. We could type a vector as:

$$(v_1, v_2, v_3)^T.$$

We can create display math in a number of ways.

$$f(x) = x^n + 500x^n - 1 + \cdots + 501x$$

We can type many things in T_EX that were not possible in HTML, for example

$$(0.1) \qquad y = x$$

$$g(x) = \int_0^\infty G(t, x) dx$$

We can now do more complicated constructions. For example, we say that λ is an eigenvalue of a matrix A corresponding to a vector \vec{u} or u if $Au = \lambda u$. For example so -1 is an eigenvalue corresponding to eigenvector $[1, 0]^T$

L^AT_EX can do more sophisticated mathematical typesetting. We can make piecewise defined functions as follows:

$$f(x) = \begin{cases} x^2 & x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Here is an example involving some Greek letters and special functions:

$$\Lambda_N = \sum_{n=0}^N \left(\frac{1}{\cos \lambda^n} \right).$$

We could write the key part of the proof that the harmonic series diverges as

$$(0.2) \quad \sum_{i=1}^{2^N-1} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{2^N-1}$$

$$(0.3) \quad > \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \overbrace{\frac{1}{8} + \cdots + \frac{1}{8}}^{4 \text{ times}} + \cdots + \frac{1}{2^N-1} \quad \text{for } N > 3$$

Counting

As we saw in Section `LATEX` counts many things, such as chapters, sections and other document body parts, equations, citations, theorems and so on. We can manipulate those counters by using the `setcounter` and `addtocounter` macros.

To make our own counter, we just use the `setcounter` macro: for instance, the number of times the word "the" has occurred so far is 42 but now, the number of times "the" has occurred. is 44

Theorems & Stuff

In this subsection, numbered, , we will discuss the numbering scheme for theorems, definitions, corollaries, and such.

Definition 0.1 *The definition is where we put the meanings of the terms we use in our theorems.*

First, we prove a technical result.

Lemma 0.2 *here is where we prove some little technical result taht we will use in the proof of the theorem.*

Theorem 0.3 *Theorems are where we prove things about the terms we defined in the definitions.*

Lists

There are three kinds of lists in every markup language: ordered lists, unordered lists, and description lists.

- A) The first item
- B) The second item
- C) The ultimate item

Here is an unordered list.

- one
- two
- three

Here is a description list.

The first definition - the first def.

second - the second. etc etc

Page layout and spacing

Page layout in \TeX is not as transparent as it is in writing in WYSiWYG word processor instead of setting margins, as we would in a WYSiWYG program, \LaTeX knows only where the upper left corner of the text area is relative to the page, and how wide and tall the text area is. For this purpose it sets many variables that control the positions and sizes of the text area. We can change the values of these variables.

There are two ways to change the values of these variables: We can do it directly (set the variable equal to exactly what we want.); or we can use the `newdimension` command. Here is a list(possibly incomplete) of the variables of what we can change and control.

oddsidemargin- This is distance of the text area from the left edge of the page on **odd** numbered pages. This is useful when you are writing a book! In the `article` documentclass, this controls both odd and even numbered pages. Note that the distance we specify here is relative to default 1 inch margin. Note that the documentclass also imposes a separate default.

evensidemargin- this is the distance of the text area from the left edge of the page on even-numbered pages This only applies in some documentclasses, such as in `book`.

topmargin- This is the top of the page.

textwidth- the width of the text area.

textheight- the height of the text area.

headheight- The height of the header.

headsep- The height of the separation of the header from the text area.

footskip- The distance from the text area to the footer.

baselineskip- The distance from the bottom of one line to the next.

baselinestretch This is actually not a dimension; it is a macro that controls how much the space between lines is stretched. For example, to get double-spaced text you could type

parskip- The extra distance to place between paragraphs.

parindent- the amount of indentation at the beginning of every paragraph. Except the first paragraph in article class.

0.1 Spacing

Before we can really discuss spacing, we have to talk about horizontal and vertical mode. \LaTeX takes characters not separated by spaces and pastes them together into a word. It takes other white space and replaces it by so-called *glue*. This is illustrated below:

All of this word and line formation takes place in horizontal mode. Putting lines one after another requires going into vertical mode – where \LaTeX figures out how to fit the various lines on the page.

we can make horizontal spaces any time we are in horizontal mode.

Space can be either positive or negative. Note that \LaTeX cannot put in vertical space except when it is in vertical mode.

there are some special spacing commands that allow us to put in a lot of glue at once. In particular in horizontal mode, the `\hfil` and `\hfill` commands put in as much space as we need. For example, look below.

Left				Right
	Left		Right	
Left		middle		right

Likewise, there are commands called `vfil` and `vfill` that give as much vertical space as we need. There are a couple of similar macros that put dots or rules in place of these spaces. These are called `dotfill` and `hrulefill`, respectively.

Chapter 12 121

Another aspect of spacing is the boxes \TeX constantly uses form lines and so on. For example, we already saw that we could put a box around any text using `framebox` `framebox`. We can make a plain box any time using `makebox`

To get a paragraph of a certain size, use `parbox`. Not that the width of this is an obligatory argument.

1 Graphics

Before we start, we should probably mention one or two ways of storing graphics that are not traditional image formats. In particular, the EPS(Encapsulated PostScript) format is often used for storing images used in \TeX . This is an old-fashioned way to do

to incorporate images into your document, you must first load the `graphicx`. After that we can use the famous `includegraphics` command to put in your image: viz

Note that from now on we will never refer to figures using any formulations such as “in the figure below”. Instead, we always give the figure a name (number), and then refer to it using that.

there are many more packages to allow us to place our figures cleverly, attractively, etc., but we must load each one separately. Note that some of the these packages are not compatible with features of other packages – never load packages you do not need. For example, if we want to place a figure to one side of the page with text flowing around it, we use the `wrapfig` package.

Note that if we want to have figure numbers in the form section.figure, we can again use the `numberwithin` macro. We can also refer to using a `ref` command. Also note that the

figure number is not incremented until we use the `caption` macro – do not use your `label` macro until after that happens.

Another common requirement is to have two figures side by side. We could actually do that in an ordinary `figure` environment, but we would not get captions on each individual figure. Figure environment isn't about figures, but about floating elements. If we want captions then we must load another package: the `subfig` package.

Now we can refer to Figure ?? using the `label-ref` system as usual.

Tables

There is another environment that works exactly like the `figure` environment, except that the object in it that the objects in it that have captions are labeled “Table”. That environment is, not surprisingly, called the `table` environment. In order to make an actual table we must invoke the `tabular` environment. Table 1 shows an example of some of the simplest features of the `tabular` environment.

Table 1: A summary of pets

Animal	Sociability	something and feeding
Dog	high	blah blah bblah
Cat	low	blah blah blah
politician	Avoid this pet at all costs	

Note that the `wrapfig` package supplies a `wraptable` environment that lets tables float to one side or the other.

2 References

We often (always) need to make references to other sources in our documents. One way of handling those is to use the `thebibliography` environment. The key macros associated with this are called `bibitem` and `cite` command to actually refer to it. For example, this document contains some bogus references to Donald Knuth's work [1].

Note that most scholars create a database of references for the work they cite often. For this they use software called BibT_EX. We will say no more about this.

References

- [1] Donald Knuth, **The T_EXbook** Addison Wesley, New York, 1979.
- [2] Leslie Lamport, **The L^AT_EX Book**, Addison Wesley, New York, 1969.

3 Programming

We can let T_EX do a lot of our work for us. For example, if there is some text that appears very often in our paper, we can define our own macro to make that text. We can use `newcommand` to create a new command that will save us a bunch of time.

Sometimes we need to define commands that operate on some argument. For example we have needed to typeset the T_EX commands we were talking about, but we were too lazy to type all the stuff every time. Above we defined a T_EX command called `\tc` that does all that for us. This way we can talk about any command we like, such as `\numberwithin`, and get the output while only having to type the argument. Here is another.

This one makes a L^AT_EX command that raises the a as much as we want: viz. L^aχ.

There is also a `\newenvironment` command that defines an environment that way we want it.

4 Odds 'n ends

1. We have seen that tilde is a special character. It is actually a non-breaking space. The hyphenation program is not allowed to hyphenate there.
2. If you need to give T_EX a hint about where a convenient place to hyphenate is, you can use the `\-` command.
3. The `verbatim` environment allows us to typeset text obeying all spaces, line breaks and so on. This is used particularly in typesetting computer code.

```
function f(x)
f = 5
return f
```