LaTeX

LaTeX is a document preparation system for the TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. LaTeX was originally written in 1984 by Leslie Lamport and has become the dominant method for using TeX; few people write in plain TeX anymore. The current version is LaTeX 2_ε.

$$E = mc^2 \tag{1}$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2}$$

This is an example document from the Wikipedia page about LaTeX.

Main points:
* LaTeX is a type-setting system
* Behold, the power of automation!
* We'll get to the specifics about TeX vs. LaTeX and stuff a little later.
* Note the nicely numbered equations, with the equals signs lined up.

Why use LaTeX?

- **aesthetics, elegance, logical structure**
- **convenience, ease, portability, reproducibility**
- **necessity? peer pressure**
- **source documents are amenable to version control**

For what?

- **abstracts**
- **journal articles**
- **problem sets, solution sets**
- **theses!**

What are some things at which LaTeX excels?

- **handling of mathematical symbols**
- **auto-numbering of figures, equations, references, etc.**
- **auto-generation of T.O.C., etc**

You can become a TeXspert if you want, but there are huge advantages to just being a TeXnician.

It's very easy to get very nicely formatted, professional-looking material, without ever doing more than typing into a text file.

The programming aspect tends to appeal to scientists, mathematicians, engineers, etc.

If someone gives you a problem set in the form of a .tex file, it's nice to know what to do with it.

You don't have to mess with font sizes and justification and stuff (though if you want to get really into it, you can start defining your own styles, writing your own packages, etc.).

Examples next.

TeX is . . .

a typesetting system written by Donald E. Knuth.
a macro processor.
written in WEB.
currently on version 3.1415926.

To produce a document, you write macros and text interleaved with each other. The macros define an environment in which the text is to be typeset.

LaTeX is . . .

a TeX-based language in which you actually write documents.
a TeX macro package, originally written by Leslie Lamport.
currently on version 2e.

LaTeX allows markup to describe the structure of a document, so that the user need not think about presentation. By using document classes and add-on packages, the same document can be produced in a variety of different layouts.

PDFTeX is . . .

a modified tex executable that implements the the LaTeX format.
a development of TeX that is capable of generating typeset PDF output in place of DVI.

PDFTeX is (today) the main stream of TeX distributions: most LaTeX users nowadays use PDFTeX whether they know it or not.

much of the above is from "The UK List of TeX Frequently Asked Questions on the Web", or the TeX Users Group (http://tug.org/levels.html)

Knuth says in the Preface to his book on TeX that it is "intended for the creation of beautiful books — and especially for books that contain a lot of mathematics". (http://www.tex.ac.uk/cgi-bin/texfaq2html?label=whatTeX)

Even though Donald Knuth himself has suggested a few areas in which TeX could have been improved, he indicated that he firmly believes that having an unchanged system that will produce the same output now and in the future is more important than introducing new features. For this reason, he has stated that the "absolutely final change (to be made after my death)" will be to change the version number to π, at which point all remaining bugs will become features. (from Wikipedia, citing Knuth's article "The future of TeX and METAFONT")