LINUX VOICE TUTORIAL

COMPOSE BEAUTIFUL TEXT WITH LATEX

VALENTINE SINITSYN

Or: how one perfectionist PhD student was able to compose his thesis in a month and was completely happy with how it looked.

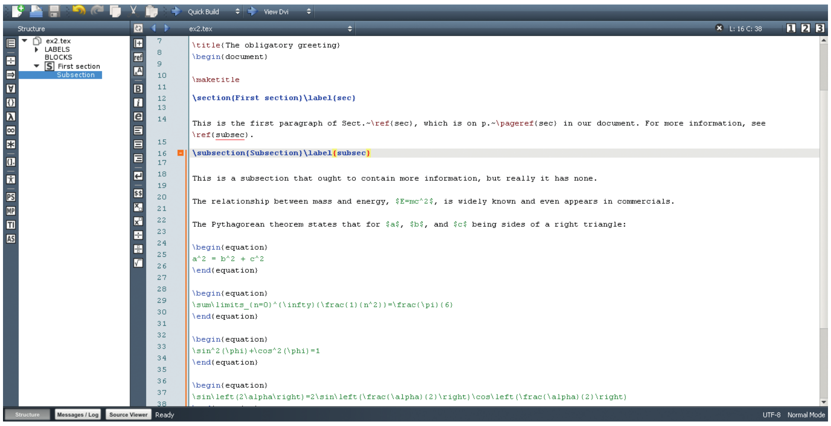
WHY DO THIS?

- Save time formatting your texts.
- Enter formulas quickly and intuitively.
- Make your documents look like CS classic.

Donald Knuth, the author of *The Art of Computer Programming*, is one of the biggest names in computer science. When he received proofs of the second edition of this book in early 1977, he found them awful – so awful he decided to write his own typesetting system. So *Tex* was born. By 1984, Leslie Lamport extended *Tex* with a set of macros known today as *Latex*. *Tex* provides layout features; *Latex*, (which translates to *Tex*) operates on higher-level objects.

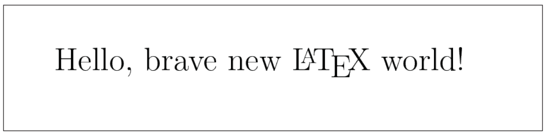
Linux already comes with plenty of modern options for processing text documents, so why waste time with a solution that's three decades old? There are plenty of reasons, but in short: *Latex* does a brilliant job for complex structured texts that need a professional look. You can use it for anything: my mom typesetted our family cookbook entirely in Latex back in the nineties, but nowadays there is probably not much reason to do so. However, if you are preparing a science report, a course project or even a thesis, *Latex* can save you a good amount of time. It lets you focus on the contents, and takes care of all the visualisation and "book keeping". It chooses the right fonts, indentations and spacings, does enumerations, tracks cross-references, generates tables of contents and indices. Sure, a word processor can do a lot of this too, but Latex takes it to the whole new level. Converting an article to a book with *Latex* is simply a matter of switching to another document class. Many science magazines provide their own *Latex* classes, and may charge you for papers not submitted in *Tex*. For documents with predefined formatting (like official reports) you are likely to find *Latex* templates where you just need to write original content and have everything else formatted properly automatically. And as *Latex* can produce PDFs or PostScript files, you never have to worry that the document will look or print differently elsewhere.

> "Latex does a brilliant job for complex structured texts that need a professional look."



```
\title{The obligatory greeting}
\begin{document}

\maketitle

\section{First section}\label{sec}

This is the first paragraph of Sect.~\ref{sec}, which is on p.~\pageref{sec} in our document. For more information, see
\ref{subsec}.

\subsection{Subsection}\label{subsec}

This is a subsection that ought to contain more information, but really it has none.

The relationship between mass and energy, $E=mc^2$, is widely known and even appears in commercials.

The Pythagorean theorem states that for $a$, $b$, and $c$ being sides of a right triangle:

\begin{equation}
a^2 = b^2 + c^2
\end{equation}

\begin{equation}
\sum\limits_{n=0}^{\infty}{\frac{1}{n^2}}=\frac{\pi}{6}
\end{equation}

\begin{equation}
\sin^2(\phi)+\cos^2(\phi)=1
\end{equation}

\begin{equation}
\sin\left(2\alpha\right)=2\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha}{2}\right)
```

Texmaker is one of many dedicated *Latex* editors.



Hello, brave new LaTeX world!

No tutorial can go without a "Hello, World!" example.

Finally, *Latex* is not just about texts. You can use it to make beautiful (albeit non-interactive) presentations. Wikipedia also uses *Latex* to render formulas in the articles.

Let's start typing

In a nutshell, *Latex* is somewhat akin to HTML (albeit older). Documents are composed in plain text files (conventionally carrying a **.tex** suffix) that contain special "tags" recognised by the **latex** command. It compiles the document and produces a DVI (DeVice Independent) file that can be viewed directly or converted to PDF or PostScript. It is also possible to produce PDFs directly with *pdfTex*.

As *Latex* documents are plain text, you can write them in your editor of choice: basic *Latex* support like syntax highlighting is usually offered. There are, however, specialised *Latex* editors with more advanced features like smart autocompletion, output preview or navigation. Of those, my personal favourite is *Texmaker* (**www.xm1math.net/texmaker**). It's cross-platform, free and built with *Qt*.

Tex itself comes in various distributions (not to be confused with the Linux distributions it runs on). They contain all the tools, common packages and document classes (which we'll discuss shortly). For Linux, the most popular *Tex* distribution is probably *Tex Live* (**www.tug.org/texlive**); see the boxout for installation tips. If you still have Windows machines around, try *MikTex* (**www.miktex.org**). Both are free software, although commercial *Tex* distributions exist as well.

If you need more, you can always use CTAN: the Comprehensive Tex Archive Network (**www.ctan.org**). It's a central repository for almost any *Latex* package,

What's in the name?

The letter "X" in "Latex" (and "Tex") is a Greek letter "chi", pronounced as /k/. So the name has nothing to do with rubber. Letters in "LaTeX" are also traditionally aligned in a slightly unusual way (see the image). To do this in your documents, use the **\LaTeX{}** command.

class etc, and if you can't find something there, chances are it doesn't exists at all.

I guess you are a bit bored with reading words by now: let's write some of them. Open a text editor and compose a simple *Latex* document:

```
\documentclass[a4paper,12pt]{article}
\usepackage[utf8]{inputenc}
% Hyphenation patterns
\usepackage[english]{babel}
\author{Valentine Sinitsyn}
\title{The obligatory greeting}
\begin{document}
Hello, brave new \LaTeX{} world!
\end{document}
```

Despite being short, this example already introduces some important aspects. *Tex* commands begin with a slash, and accept parameters either in square or in curly brackets. As you probably guessed, square brackets are used for optional arguments. Comments start with a percentage sign; if you need a literal **%** symbol, use a **\%** command.

Latex documents start with a preamble that sets the document class and imports the required *Latex* packages with the **\usepackage{}** command. Here, the class is **article** with 12pt font size on A4 paper. Other standard classes include **book**, **report** and **letter**. Document class greatly influences the document appearance. For example, **\documentclass{book}** will make the document double-sided, start chapters on odd pages, and add some automatic headers and footers.

\begin{document} and **\end{document}** commands create an "environment", where the body of your document goes. Here, it's trivial (for the **\LaTeX{}** command, see the sidebar).

Now, save the file under the name **hello.tex** and compile with:

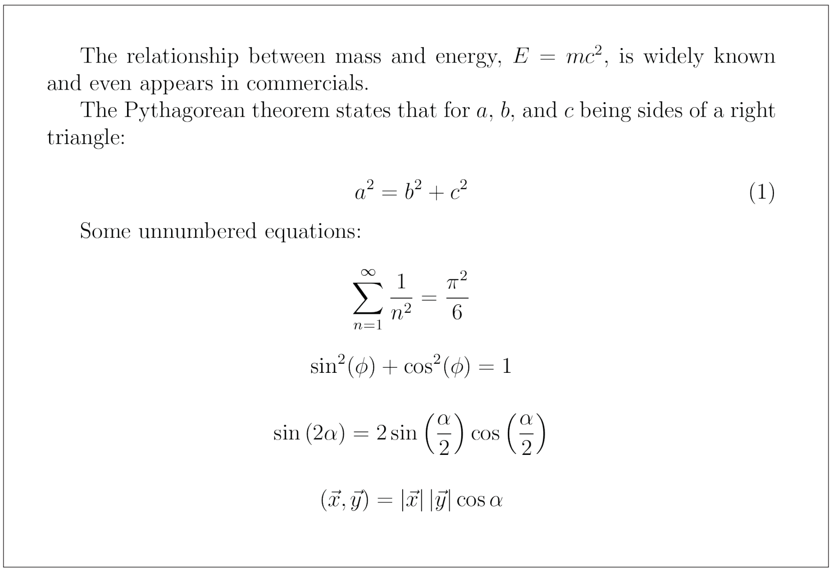
```
latex hello.tex
```

If you typed everything correctly, you'll get a **hello.dvi** file that you can view with *Evince* (Gnome/Unity), *Okular* (KDE) or *xdvi* (comes with *TeX Live*). To convert DVI to PDF or PostScript, use the **dvipdf** or **dvips** commands, respectively. If you use a dedicated *TeX* editor like *Texmaker*, these steps will be performed automatically when you build the document.

Some more words

This was of course a very basic example. To let *Latex* show its powers, something more sophisticated is needed, like this (this goes into the 'document' environment from the example above):

```
\maketitle
\section{First section}\label{sec}
This is the first paragraph of Sect.~\ref{sec}, which is on p.~\pageref{sec} in our document. For more information, see \ref{subsec}.
\subsection{Subsection}\label{subsec}
This is a subsection that ought to contain more information, but really it has none.
```



The relationship between mass and energy, $E = mc^2$, is widely known and even appears in commercials.

The Pythagorean theorem states that for a, b, and c being sides of a right triangle:

$$a^2 = b^2 + c^2 \qquad (1)$$

Some unnumbered equations:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\sin^2(\phi) + \cos^2(\phi) = 1$$

$$\sin(2\alpha) = 2\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha}{2}\right)$$

$$(\vec{x}, \vec{y}) = |\vec{x}|\,|\vec{y}|\cos\alpha$$

Latex formulas can be embedded in paragraph text or come on their own.

The **\maketitle** command just renders a title set previously in the preamble. Paragraphs are separated with a blank line. The **\subsection** command creates a subsection header, and again, *Latex* chooses the exact font size, typeface etc automatically (as per document class). The tilde character inserts a non-breaking space, so references will always stay on the same line with **Sect.** and **p.** (it's a recommended practice).

What's new here is the **\label{}** command. You can think of it as a way to give a place in the text a meaningful name (stubs like **sec** shouldn't appear in real world documents). Later, you can include a reference to the label with either the **\ref{}** or **\pageref{}** commands. The first one references a section (or equation, or figure, or something else) by number, like '1' or '1.1'. A neat thing is that *Latex* does the enumeration automatically, so if you put another subsubsection before the **subsec** label, the cross-references will stay correct (although the **latex** command might ask you to run itself twice to update references, otherwise they will appear in the text as **??**). **\pageref** puts a reference to the page where the

Latex vs your favourite text suite

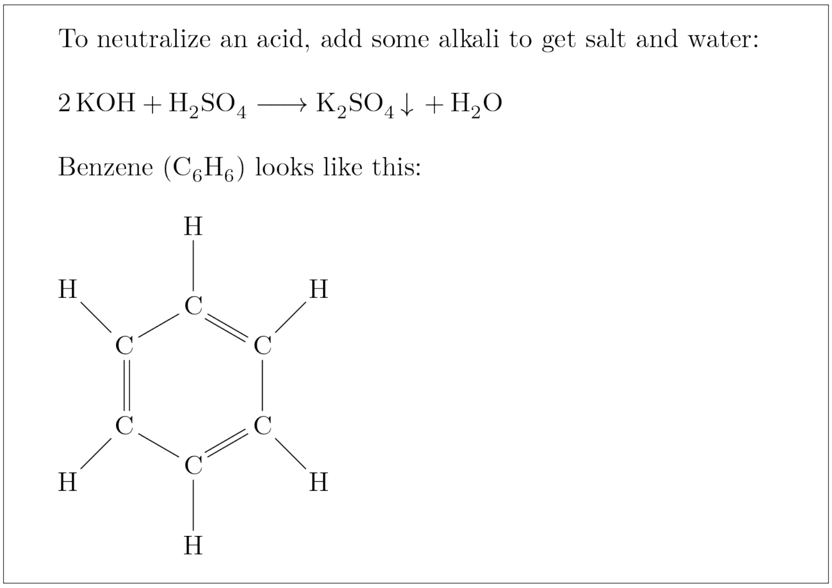
Latex is a great tool, but as with everything it has its pros and cons. They are quite subjective and depend on how skilled a *Latex* user you are – I know several people using *Latex* for all their documents with no trouble. Nevertheless, here is a quick side-by-side comparison:

Consider *Latex* for:

- Scientific texts, like papers or thesis.
- Texts with many formulas and cross-references.
- Texts that must adhere to strict formatting rules.

Better try something else for:

- Small texts with simple formatting (use *Writer*).
- Texts with artistic irregular structure, like in LV (*Scribus*).
- Interactive presentations or spreadsheets (*Impress*/*Calc*).



To neutralize an acid, add some alkali to get salt and water:

$$2\,\mathrm{KOH} + \mathrm{H_2SO_4} \longrightarrow \mathrm{K_2SO_4}\downarrow + \mathrm{H_2O}$$

Benzene (C_6H_6) looks like this:

If you are a chemist, *Latex* is here to help you make benzene look even cooler. Benzene's structure was discovered after Friedrich Kekulé had a crazy dream in front of the fire.

label resides, and again *Latex* does all the bookkeeping for you.

Do simple math

Formulas in *Latex* come in two flavours: text and displayed. The former are rendered inline; the latter are printed separately from the main text:

```
The relationship between mass and energy, $E=mc^2$, is widely
known and even appears in commercials.

The Pythagorean theorem states that for $a$, $b$, and $c$
being sides of a right triangle:

\begin{equation}
a^2 = b^2 + c^2
\end{equation}
```

Latex enumerates displayed formulas automatically (see the image on page 87). If you don't need this, use the **equation*** environment (defined in the **amsmath** package) instead of **equation**.

If you ever created formulas in *OpenOffice.org/LibreOffice Math*, *Latex* will feel a bit familiar to you. A circumflex (**^**) denotes a superscript, and underscore is used for subscripts. If they span more than one character, use curly brackets (this is the rule for many other formatting commands in *Latex* as well): **$a^xa^y=a^{x+y}$**. Fractions are created with **\frac{nominator}{denominator}**. They don't usually look good in word processor documents, but *Latex* does a great job of aligning them properly.

Of course, you're free to write more complex math, like series summation or integrals:

```
\sum\limits_{n=1}^{\infty}{\frac{1}{n^2}}=\frac{\pi^2}{6}
```

This example combines all of the concepts we've already discussed, and introduces some new ones. First, there's the **\limits** command to put summation limits at conventional positions (above and below the summation sign, not in the upper-right and lower-right corners, as **_** and **^** do alone). Then, it has the **\infty** command to render the infinity symbol, and finally **\pi** for a Greek letter 'pi'. If you need a capital 'pi', use the **\Pi** command, and **\Delta** produces the well known triangle-like letter. Yes, it's that simple.

Latex renders most mathematical functions you know about (and maybe some you aren't even aware of). The respective commands are named after the functions, and you only need to prepend a slash, like this:

```
\sin^2(\phi)+\cos^2(\phi)=1
```

Plain parentheses don't adjust their sizes to match arguments. To produce scaling parentheses, use the **\left(** and **\right)** commands like so:

```
\sin\left(\alpha\right)=2\sin\left(\frac{\alpha}{2}\right)\cos\
left(\frac{\alpha}{2}\right)
```

\left and **\right** also work for brackets and curly braces. *Latex* is smart enough to match **\left**s to **\right**s, and will issue a compilation error if you missed anything:

```
LaTeX2e <2011/06/27>
Babel <3.9h> and hyphenation patterns for 2 languages loaded.
...
! Missing \right. inserted.
<inserted text>
           \right .
```

Finally, *Latex* can easily add all sorts of decoration you may need for your math texts, like arrows (for vectors) or hats (for matrices and operators). Consider the following:

```
\left(\vec x,\vec y\right)=\left|\vec x\right|\left|\vec y\right|\
cos\alpha
```

Note that for single-letter arguments, like **x** and **y** above, you can omit curly brackets. Also keep in mind that accents don't scale (try **\vec{x+y}**), as it wouldn't make much sense (mathematically).

And even fine arts

At this point you may start thinking that *Latex* is cool but of a little use to you, as you don't write math. While this might be true, *Latex* has something to offer for those form other branches of science as well.

Let's take chemistry. I'm not very good in it, but I was able to recall that alkalis neutralise (otherwise quite dangerous) acids. For sulphuric acid,

Where do I get Tex?

The easiest way to obtain software (*Tex* included) in Linux is to use packages from your distribution repositories. These usually contain everything you need to build a basic *Tex* system and many popular extensions from CTAN, only a mouse click away. Depending on which Linux flavour you use, they can be cutting-edge or quite outdated. In Ubuntu, these packages names start with **texlive-**. The **texlive-base** command installs a bare minimum, while **texlive** provides a decent selection of the *Tex Live* packages.

If your distribution packages miss something crucial for you, install latest *Tex Live* by yourself and use tlmgr utility to get any package you need from CTAN. You'll miss automated updates from your Linux vendor, so be prepared. If you only need a single specific package from CTAN, you can also install it in the prepackaged *Tex Live* manually, following instructions in the package manual. However, this is the last resort, so better stick to the completely prebuilt (simpler) or 'vanilla' variant.

If you get stuck, remember that **StackOverflow.com** has a complete sister site dedicated to *Tex*: **http://tex.stackexchange.com**.

Latex inside

This tutorial showed how to use *Latex* on its own. However, *Latex* also empowers several well-known software suits.

First, there is *Lyx* (**www.lyx.org**) – a visual graphical WYSIWYM (What You See Is What You Mean) document processor that matches *Writer*'s intuitiveness to *Latex*'s abilities. *Lyx* is not *Latex*, but is a good alternative with a flat learning curve.

LilyPond (**www.lilypond.org**) is non-visual, more *Latex*-like system for music engraving. If the **abc** package seems limited, you should probably give *LilyPond* a try. For easier editing, look at *Frescobaldi* (**www.frescobaldi.org**).

neutralisation can be represented by the reaction on show above-left.

To reproduce the equation from that figure, try this:

```
\documentclass{article}
\usepackage[version=3]{mhchem}
\begin{document}
\ce{2KOH + H2SO4 -> K2SO4 v + H_2O}
\end{document}
```

The key is the **mhchem** package (from **texlive-science**) that I included on the second line. Chemical species and equations are passed as **\ce{}** command arguments. Indices are subscripted (or superscripted) automatically, and you can put whatever sorts of arrows you need.

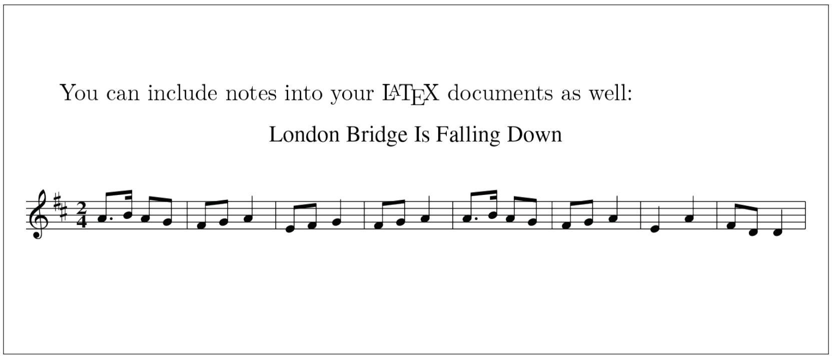
Another thing you often see in chemical texts is structural formulae. You can draw them in specialised software, but with *Latex*, it is easy to include such diagrams directly in text, like this:

```
\documentclass{article}
\usepackage{chemfig}
\begin{document}
\chemfig{C*6(-C(-[6]H)=C(-[7]H)-C(-[1]H)=C(-[2]H)-C(-[3]H)=)
(-[5]H)}
\end{document}
```

This time, I used the **chemfig** *Latex* package (which comes in **texlive-pictures**) to render a benzene molecule (**\ce{C6H6}**). The **C*6** part means we are drawing a hexagon (six sides), and **C** is its first vertex. The first pair of parentheses contain the hexagon's sides (hyphens mean single bond and equals symbols mean double bonds) and remaining vertices (carbon atoms marked as **C**). The inner parentheses (and the last ones) are for branches (hydrogen atoms). Numbers in brackets set a branch direction (in 45 degree units, counter-clockwise). Note that DVI displays the diagram wrong, and to preview the results, you'll need to convert the output to PostScript or PDF first, or use the **pdflatex** command to produces a PDF directly:

```
pdflatex benzene.tex
evince benzene.pdf
```

If you are not into sciences, but into arts, *Latex* can also prove itself useful. For example, you can use it to print music sheets. There is a specialised *Tex*-based software called *LilyPond* built just for these purposes (see the sidebar), but pure *Latex* will fit the bill as well. Packages like **musixtex** or **abc** (found in **texlive-music**) can be used to enrich your texts with some tunes:



You can include notes into your LaTeX documents as well:

London Bridge Is Falling Down

Barely understandable for those like me, but it looks good nevertheless.

```
\documentclass{article}
\usepackage{abc}
\begin{document}
You can include notes into your \LaTeX{} documents as well:
% ABC notation is used here, see http://en.wikipedia.org/wiki/
ABC_notation
\begin{abc}
X:1
T:London Bridge Is Falling Down
M:2/4
L:1/8
K:D
A>B AG|FGA2|EFG2|FGA2|
A>B AG|FGA2|E2A2|FDD2|
\end{abc}
\end{document}
```

For this to compile, you'll need to install the **abcm2ps** tool with your package manager. Then, pass the **--shell-escape** option to the **latex** or **pdflatex** command:

```
pdflatex --shell-escape london_bridge.tex
```

As with **chemfig**, the notes aren't directly viewable in DVI.

Follow your route

Here we come to an end of our brief excursion into the *Latex* world. I hope you agree now that *Latex* isn't a scary beast from the pre-PC era, and can save you time and effort even 30 years after its initial introduction. And you've probably already guessed that we merely scratched the surface in this tutorial. With *Latex*, you can do many other things we haven't even mentioned: generate tables of contents and insert figures (with automatic enumeration and references, of course), prepare nice PDF presentations with the **beamer** package, maintain a bibliography, and much more.

There are books written on *Latex*, and there's so much more that it can do. If you do anything with text, you may well have found your new favourite tool.

Dr Valentine Sinitsyn has committer rights in KDE but prefers to spend his time mastering virtualisation and doing clever things with Python.