

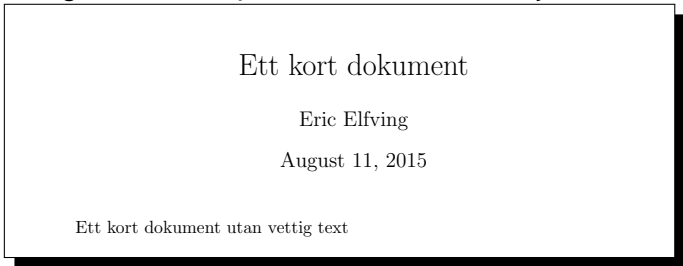
**L<sup>A</sup>T<sub>E</sub>X**

**Eric Elfving**  
**Institutionen för datavetenskap (IDA)**

**4 september 2015**

- L<sup>A</sup>T<sub>E</sub>X är ett "document preparation system"
- Vanliga ordbehandlare (t.ex. MS Word och OpenOffice) fungerar enligt principen **What You See Is What You Get** (WYSIWYG)
- L<sup>A</sup>T<sub>E</sub>X låter författaren fokusera på innehållet istället för utseendet - **What You See Is What You Mean**

- Antag att vi vill skapa ett dokument med följande innehåll:



- I en ordbehandlare krävs det ganska mycket jobb...

- I L<sup>A</sup>T<sub>E</sub>X skriver man "enkel" kod i en texteditor:

```
\documentclass{article}

\author{Eric Elfving}
\title{Ett kort dokument}

\begin{document}
\maketitle
Ett kort dokument utan vettig text
\end{document}
```

- Likt ett programmeringsspråk kompileras sedan din källkod till ett dokument
- Det finns stöd för flera olika filformat, men pdf är ju alltid trevligt
- På grund av referenser, inkluderingar och annat behöver man ofta kompilera i flera steg, därför finns det ett script `latexmk` som löser problemet åt en
  - Väljaren `-pdf` ger en pdf-fil som resultat
  - Väljaren `-pvc` uppdaterar resultatfilen när källkoden ändras
  - I Mint finns det flera paket för att installera L<sup>A</sup>T<sub>E</sub>X, men `latex-full` installerar "allt":  
`sudo apt-get install texlive-full latexmk`

- Ett L<sup>A</sup>T<sub>E</sub>X-dokument består av fyra grundläggande element:
  - Kommandon (command):  
`\namn[valfria parametrar]{parametrar}`
  - Block (group):  
`{ ... }`
  - Miljöer (environment):  
`\begin{miljönamn}`  
`...`  
`\end{miljönamn}`
  - Kommentarer:  
Endast enradskommentarer, inleds med `%`

```
\documentclass{...}  
% Beskriver vilken typ av dokument vi skriver.  
% Finns flera inbyggda som tex article , report och letter  
  
% Inledning (preamble)  
  
\begin{document}  
% Här står allt som ska synas i dokumentet  
\end{document}
```

- Området mellan `\documentclass` och `\begin{document}` kallas inledning (preamble)
- I inledningen kan man göra inställningar och ladda in extra paket för att lösa saker som språket inte har i grunden
- Ett paket laddas in med kommandot `usepackage`:  
`\usepackage[paketparametrar]{paketnamn}`



- `inputenc` - används för att få "specialtecken"(tex å,ä och ö) att fungera. Behövs endast när man använder `pdflatex` eller `latex`  
`\usepackage[utf8]{inputenc}`
- `fontspec` - Används istället för `inputenc` vid användning av `xelatex`
- `babel` - Språkinställningar såsom avstavning och formatering av tal och datum  
`\usepackage[swedish]{babel}`
- `graphicx` - Stöd för att inkludera bilder och grafik
- `listings` - Inkludera och formatera källkod
- `xcolor` - Massor av färger

1. Öppna ett tidigare dokument
2. Modifiera inledningen och ta bort resten
3. Spara som ett nytt dokument
4. Starta `latexmk` (med `-pvc`)
5. Låt texteditorn ta halva skärmen och din dokumentvisare andra halvan
6. Direkt du sparar uppdateras dokumentet i läsaren

- Rubriker skapas enkelt med kommandon:
  - Huvudrubriker med `\section{rubriknamn}`
  - Underrubrik med `\subsection{namn}`
  - Lägre nivå med `\subsubsection{namn}`
- Vid användning av dokumentklasserna `report` eller `book` finns även `chapter` i toppen av hierarkin
- Alla rubriker blir automatiskt numrerade
  - Går att stänga av med \*-varianten:  
`\section*{namn}`
- Innehållsförteckning skapas automatiskt med kommandot `tableofcontents`

- Precis som i HTML finns två typer av listor;
  - `itemize` - punktlista (`ul`)
  - `enumerate` - numrerad lista (`ol`)
- Exempel:

```
\begin{enumerate}  
\item Punkt ett  
  \begin{enumerate}  
    \item hej  
  \end{enumerate}  
\item Punkt två  
\end{enumerate}
```

1. Punkt ett
  - 1.1 hej
2. Punkt två

```
\begin{table}[!h]
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimalt \\
3700 & oktalt \\
11111000000 & binärt \\
\hline \hline
1984 & decimalt \\
\hline
\end{tabular}
\caption{Olika talbaser}
\label{tab:talbaser}
\end{table}
```

7C0	hexadecimalt
3700	oktalt
11111000000	binärt
1984	decimalt

Tabell : Olika talbaser

$$x = 2^3 * (4 + 2)$$

$$x = 2^3 * (4 + 2)$$

```
\begin{displaymath}
F_n =
\left\{ \begin{array}{ll}
n & n < 2 \\
F_{n-1} + F_{n-2} & n \geq 2
\end{array} \right.
\end{displaymath}
```

$$F_n = \begin{cases} n & n < 2 \\ F_{n-1} + F_{n-2} & n \geq 2 \end{cases}$$

- För att referera till andra delar i sitt dokument använder man sig av `\label` och `\ref`
  - `\label{namn}` används för att markera området (t.ex. en rubrik eller bildtext) man vill referera till
  - `\ref{namn}` används i löpande text för att sätta in en referens till markeringen `namn`
- För att referera till andra källor används `\cite`
  - Paketet `natbib` ger även tillgång till kommandona `citet` och `citep` som formaterar referenserna snyggare
  - Paketet `biblatex` är nyare och rekommenderas generellt

```
\documentclass{article}
\usepackage[swedish]{babel}
\usepackage{fontspec}

\begin{document}
\section{Historia}
\label{sec:history}
Här berättar vi lite historia

\section{Sammanfattning}
I del \ref{sec:history} ...
\end{document}
```

### 1 Historia

Här berättar vi lite historia

### 2 Sammanfattning

I del 1 ...



## BibTeX

- Om man har många externa källor kan man flytta ut dem på en separat fil
- Man strukturerar upp dem i en fil med filändelse `.bib`
- Där man sedan vill inkludera källorna kör man kommandot `\bibliography{filnamn}` (där `filnamn` är utan filändelse)
- Exempel på BibTeX-källa:

```
@article{berland13,  
author = {Berland, Matthew and Martin, Taylor and Benton, Tom and Petrick Smith  
, Carmen and Davis, Don},  
title = {Using Learning Analytics to Understand the Learning Pathways of Novice  
Programmers},  
journal = {Journal of the Learning Sciences},  
volume = {22},  
number = {4},  
pages = {564-599},  
year = {2013},  
}
```

- Det är ganska lätt att skriva ett snyggt dokument
  - Källkoden kan läsas i alla texteditorer - en viss ordbehandlares text är ofta svår att läsa i en annan (eller en annan version)
  - Du kan fokusera helt på texten
  - Indexering, fotnötter och citeringar sköts automatiskt
  - Eftersom allt är ren text går det att generera texten i vilket program (eller programspråk...) som helst!
  - Kompilatorn tvingar dig att göra rätt - din text byggs upp på ett vettigt sätt
  - Inbyggt stöd för att dela upp texten - lätt att samarbeta med andra
  - Källkod går bra att versionshantera...
-

- Hög inlärningströskel
- Finns massor av kommandon och miljöer (såklart också en fördel)
- Kan vara svårt att få din text att se ut exakt som du vill
- Normalt sett ser man inte resultatet medan man skriver utan måste vänta på kompilering

- <http://en.wikibooks.org/wiki/LaTeX/>  
En väldigt bra wiki med både grunder och avancerade ämnen
- Comprehensive TeX Archive Network (CTAN)  
<http://www.ctan.org>  
Samling av latexmoduler

[www.liu.se](http://www.liu.se)