En superkort introduktion til LaTeX

Torben Mogensen

13. september 2016

Resumé

I kurset "Programmering og problemløsning" (PoP) bruger vi værktøjerne Emacs og LaTeX.

Emacs er et tekstredigeringsværktøj, der er egnet til at redigere programmer og lignende. Et andet dokument introducerer Emacs. Ulig f.eks. Microsoft Word og Libre Office Writer redigerer man med Emacs "ren" tekst: Der er ikke formatering såsom fed skrift, kursiv, osv. Til dette bruger vi LaTeX.

LaTeX (som udtales "lahtek") er et dokumentformateringsværktøj. Det fungerer ligesom HTML ved, at man skriver formateringskoder ind i en ren tekstfil, som derefter konverteres til et formateret dokument. Hvor HTML konverteres i en browser til en webside, så konverteres LaTeX-dokumenter til PDF ved brug af værktøjet `pdflatex`.

Indhold

1 LaTeX **2**
1.1 Installering af LaTeX 2
1.2 Brug af LaTeX 3
1.2.1 Syntaks 5
1.3 Et struktureret dokument 6
1.4 Fed tekst, kursiv, m.m. 7
1.5 Krydsreferencer og figurer 9
1.6 Formler 10
1.7 Billeder 11
1.8 Udvidelsespakker 11
1.9 LaTeX og Emacs 12
1.10 Finpudsning 13
1.11 Supplerende litteratur 14
1.12 Fortvivl ikke! 14

Kapitel 1

LaTeX

LaTeX er et tekstformateringsværktøj designet til at skrive strukturerede dokumenter såsom videnskabelige artikler, rapporter, lærebøger, breve, Powerpoint-lignende præsentationer osv., men LaTeX kan også bruges til mange andre dokumentformer, for der er rige muligheder for at definere sine egne strukturer og sit eget layout.

LaTeX udtales "lahtek' og ikke "lateks", da de tre bogstaver i TeX faktisk er de græske bogstaver tau, epsilon og ksi, så denne del af ordet udtales "tech" eller "tek". LaTeX er en videreudvikling af Donald Knuths TeX lavet af Leslie Lamport, som satte bogstaverne La foran navnet. Det har altså intet med syntetisk gummi at gøre, og vi *har* allerede hørt alle vittighederne.

Den væsentligste forskel på LaTeX og f.eks. Microsoft Word er, at man forbereder sine LaTeX-dokumenter i et tekstredigeringsværktøj som f.eks. Emacs og derefter konverterer dem til PDF ved at køre et program. Det er i princippet ikke anderledes end den måde, et HTML-dokument konverteres til tekst og billeder i et vindue i en browser. Et LaTeX-dokument indeholder ud over den tekst, der skal indgå i det færdige PDF-dokument, et antal koder og kommandoer, der specificerer formatering, krydsreferencer, billedindsættelse, formler, tabeller, figurer osv. Det er LaTeX-programmets opgave at fortolke disse koder og bruge dem til at formatere dokumentets tekst. Endvidere opdeler LaTeX teksten i sider, som er egnede til udskrivning, visning på en projektor, eller lignende, og indsætter selv sidenumre osv.

1.1 Installering af LaTeX

LaTeX er gratis, open-source, og findes til en lang række platforme, og der findes også web-værktøjer, hvor man kan lave LaTeX-dokumenter i sin browser. Vi anbefaler, at man bruger Emacs på Linux, MacOS X eller Windows til at redigere sine LaTeX-dokumenter og derefter konverterer dem til PDF med programmet `pdflatex`. Her er en meget kort instruktion til installering af dette program på de tre hovedplatforme.

Linux. De fleste Linux-distributioner tillader installering af LaTeX gennem pakkesystemet, på samme måde som tidligere beskrevet for Emacs. Vi anbefaler, at du installerer TeX Live, der gør det nemt at hente udvidelser osv.

Hvis du bruger Ubuntu Softwarecenter, så vælg pakkerne `texlive-full` og `texlive-lang-all`. Det fylder en del, og det tager en del tid at hente, så sørg for at have en god Internetforbindelse.

MacOs X. Vi anbefaler MacTeX, der indeholder TeX Live og et par MacOS-specifikke udvidelser. Den kan hentes på `http://www.tug.org/mactex/`.

Windows. Følg instruktionen på
`http://www.tug.org/texlive/acquire-netinstall.html`.

1.2 Brug af LaTeX

For at afprøve installeringen af LaTeX, anbefaler vi, at du starter med et meget simpelt dokument. Brug Emacs til at oprette en fil `Hello.tex`, der indeholder følgende tekst:

```
\documentclass{report}

\begin{document}

Hello World

\end{document}
```

Vi kommer tilbage til, hvad de enkelte dele af filen betyder senere.

Når du har gemt filen, så kør følgende kommando i et kommandolinjevindue: `pdflatex Hello.tex`. Du skulle gerne få noget output i stil med dette på din skærm (detaljerne kan variere fra system til system):

```
This is pdfTeX, Version 3.1415926-2.5-1.40.14 (TeX Live 2013/Debian)
 restricted \write18 enabled.
entering extended mode
(./Hello.tex
LaTeX2e <2011/06/27>
Babel <3.9h> and hyphenation patterns for 78 languages loaded.
(/usr/share/texlive/texmf-dist/tex/latex/base/report.cls
Document Class: report 2007/10/19 v1.4h Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size10.clo))
No file Hello.aux.
[1{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map}] (./Hello.aux) )<
/usr/share/texlive/texmf-dist/fonts/type1/public/amsfonts/cm/cmr10.pfb>
Output written on Hello.pdf (1 page, 11544 bytes).
Transcript written on Hello.log.
```

Hvis du ser i filkataloget, vil du se, at der er oprettet tre nye filer: `Hello.aux`, `Hello.log` og `Hello.pdf`. Sidstnævnte skulle gerne være et et-sides PDF-dokument med teksten "Hello World". `Hello.aux` indeholder information, som LaTeX bruger til krydsreferencer og lignende, så det er ikke interessant at læse indholdet af dette. `Hello.log` indeholder en masse information om kørslen af LaTeX, som for det meste også er uinteressant, men som kan bruges til at finde oplysninger om fejl. Lad os tage et eksempel: Vi staver med vilje forkert i den sidste linje:

```
\documentclass{report}

\begin{document}

Hello World

\end{dokument}
```

Nu får vi følgende uddata, når vi kører `pdflatex Hello.tex`:

```
This is pdfTeX, Version 3.1415926-2.5-1.40.14 (TeX Live 2013/Debian)
 restricted \write18 enabled.
entering extended mode
(./Hello.tex
LaTeX2e <2011/06/27>
Babel <3.9h> and hyphenation patterns for 78 languages loaded.
(/usr/share/texlive/texmf-dist/tex/latex/base/report.cls
Document Class: report 2007/10/19 v1.4h Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size10.clo)) (./Hello.aux)

! LaTeX Error: \begin{document} ended by \end{dokument}.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.6 \end{dokument}

?
```

Linjen, der starter med `LaTeX Error` forklarer, at der er en fejl, og hvad LaTeX tror, fejlen er. Lidt længere nede er der en indikation af linjenummeret (`l.6`) og en gengivelse af den fejlbehæftede linje. Til sidst står et spørgsmålstegn. Tryk `x` og Enter for at komme ud af `pdflatex`, der nu skriver

```
No pages of output.
Transcript written on Hello.log.
```

Bemærk, at forekomsten af en fejl medfører, at der ikke produceres nogen PDF fil. Hvis der allerede findes en PDF fil, bliver denne dog liggende uændret.

Hvis vi i stedet ændrer sidste linje til

```
\edn{document}
```

får vi følgende fejlmeddelelse (efter samme indledende tekst):

```
! Undefined control sequence.
l.6 \edn
        {document}
?
```

Meddelelsen `Undefined control sequence` betyder, at LaTeX ikke kender kommandoen `\edn`. Den viser dette ved at opdele den fejlbehæftede linje der, hvor den har fundet fejlen. Igen kan man trykke `x` og Enter for at komme ud.

Der er flere mulige fejlmeddelelser, som nogen gange kan være kryptiske. En beskrivelse af de hyppigst forekommende fejlmeddelelser, og deres mest sandsynlige årsag, kan findes på `http://en.wikibooks.org/wiki/LaTeX/Errors_and_Warnings`. En mere komplet liste kan findes på `http://www.eng.fsu.edu/~dommelen/l2h/errors.html`.

1.2.1 Syntaks

Vi kan af ovenstående se, at ikke alle tekstfiler kan konverteres til PDF af `pdflatex`: De skal opfylde nogle regler for tekstens struktur. Disse regler kaldes *syntaks*, et begreb, der også findes for "rigtige" sprog såsom dansk og engelsk. Groft sagt dækker begrebet syntaks de *grammatiske regler og staveregler*, der kendetegner et skrevet sprog, men uden at tage hensyn til *betydningen* af den skrevne tekst.

Et LaTeX dokument skal starte med kommandoen `\documentclass` efterfulgt af navnet på en dokumenttype indeholdt i krøllede parenteser (kendt som "mængdeparenteser" i matematik). I eksemplet herover har vi brugt dokumenttypen `report`, som bruges til at skrive rapporter. Andre dokumenttyper er `article`, `book` og `letter`, som mere eller mindre siger sig selv, og `beamer`, der bruges til Powerpoint-lignende præsentationer. Vi holder os primært til `report` i denne korte vejledning.

Derefter kan komme et antal kommandoer, som man kalder "indledningen" (eller *the preamble*), hvorefter dokumentets hovedindhold kommer omsluttet af kommandoerne `\begin{document}` og `\end{document}`. Al tekst efter dette vil blive ignoreret. Vores "Hello World" eksempel er altså stort set et minimalt LaTeX-dokument.

Da LaTeX er designet i USA, er standardindstillingerne, at PDF-filen bruger det amerikanske letter-papirformat. Dette format er en smule bredere end A4, men ikke helt så højt. For at få LaTeX til at producere PDF formateret til A4-papir, ændrer vi første linje til

```
\documentclass[a4paper]{report}
```

Man kan angive andre papirformater i stedet for A4, blandt andet `a5paper` og `b5paper`. Vi holder os dog til A4 papir, da det passer til de fleste printere. For at kunne bruge de danske bogstaver æ, ø og å og accenter såsom é, for at få automatisk orddeling tilpasset det danske sprog, og et par andre ting, som vi kommer tilbage til senere, tilføjer vi et antal kommandoer til indledningen, som dermed bliver

```
\documentclass[a4paper]{report}

\usepackage[utf8x]{inputenc}
\usepackage{latexsym}
\usepackage[danish]{babel}
\usepackage{graphicx}
\usepackage{hyperref}
\usepackage[all]{hypcap}
```

Du behøver ikke at forstå, hvad alle disse kommandoer betyder, men du bør bemærke, at tegnene `\`, `{`, `}`, `[` og `]` bruges i kommandoer. LaTeX bruger yderligere et antal tegn til kommandoer, så følgende tegn kan ikke uden videre bruges i tekster, uden at LaTeX måske fortolker dem som dele af kommandoer:

```
\ { } [ ] $ & # % ^ _ ~
```

Hvis man i sin tekst gerne vil bruge disse tegn, kan man i de fleste tilfælde bare sætte et \ foran, så teksten `\{\$\_\%\}` producerer tegnfølgen {$_%} i PDF-filen. Der er dog undtagelser: Tegnet \ produceres med kommandoen `$\backslash$`, tegnet ~ med kommandoen `\~{}`, tegnet ^ med kommandoen `\^{}`, og tegnene [og] kan uden videre skrives, som de er, så længe de ikke kommer lige efter et kommandonavn.

Hvis man bruger disse tegn til andet end kommandoer, vil man enten få fejlmeddelelser, eller man vil få et andet resultat, end det man venter. Specielt vil tegnet ~ i reglen vises som et blanktegn, og alle tegn mellem et % og næste linjeskift ignoreres af LaTeX. Dette kan man bruge til at skrive noter om ting, man gerne vil skrive ind i dokumentet senere, f.eks.

```
H.~C.~Andersen sagde »At rejse er at leve«.
% check lige, om citatet er rigtigt.
```

som blot giver outputtet

> H. C. Andersen sagde »At rejse er at leve«.

Generelt vil LaTeX samle gentagne blanktegn til et enkelt og betragte linjeskift på lige fod med blanktegn. Dog indikerer to eller flere linjeskift efter hinanden, at et nyt tekstafsnit begynder. F.eks. giver teksten

```
Det   var   en mørk og
stormfuld
   nat.


Men intet
  interessant
skete.
```

outputtet

> Det var en mørk og stormfuld nat.
>
> Men intet interessant skete.

Hvis man ønsker flere blanktegn efter hinanden i en tekst, kan man bruge tegnet ~ det nødvendige antal gange. Man kan tvinge et linjeskift med kommandoen `\newline`.

Fordi linjeskift (enkeltvis) har samme betydning som blanktegn, anbefaler vi, at man i Emacs bruger auto-fill-mode, når man skriver LaTeX dokumenter.

1.3 Et struktureret dokument

Som nævnt, fokuserer vi her på rapporter, indikeret af dokumentklassen `report`. Det er denne dokumentklasse, der er brugt til at lave det dokument, du netop er i gang med at læse.

En rapport har en forside med en titel, en forfatter og en dato. Forsiden laves ved at angive de relevante informationer med kommandoer, og derefter lave forsiden med kommandoen `\maketitle`. Titelsiden på indeværende dokument er lavet med kommandoerne

```
\title{En superkort introduktion til \LaTeX}
\author{Torben Mogensen}
\date{\today}
\maketitle
```

der alle står lige efter `\begin{document}`. Kommandoerne `\title`, `\author` og `\date` siger sig selv. Kommandoen `\LaTeX` genererer LaTeX-logoet, og `\today` indsætter dags dato i dokumentet. Det er en god måde at sikre sig, at datoen på forsiden faktisk svarer til den seneste udgave af dokumentet.

En rapport består af kapitler (indledt med `\chapter{overskrift}`), der er opdelt i nummererede afsnit (indledt med `\section{overskrift}`), og underafsnit (indledt med `\subsection{overskrift}`). LaTeX finder selv ud af nummereringen, man skal blot angive titlerne på kapitler og afsnit. I rapporter starter kapitler på nye sider, så LaTeX-dokumentet vist i figur 1.1 vil generere to korte kapitler startende på hver sin side. Prøv selv at kopiere teksten ind i Emacs, gem den under navnet `simple.tex`, og kør kommandoen `pdflatex simple.tex`.

Bemærk, at de første tekstafsnit i hvert kapitel eller afsnit ikke er indrykket, mens de efterfølgende er. Det ses i starten af kapitel 2. Hvis man ikke ønsker, at et tekstafsnit skal indrykkes, kan man indsætte kommandoen `\noindent` på linjen lige inden tekstafsnittet. Det bruger man f.eks. i tekstafsnit, der står lige efter diagrammer eller formler, der vises på linjer for sig med blanke linjer imellem. Vi har i denne rapport blandt andet brugt det efter de eksempler på input og output til LaTeX, der står i `skrivemaskineskrift`.

Til kortere dokumenter kan man bruge dokumentklassen `article`. Her kan man ikke bruge kapitler (kommandoen `\chapter`) er udefineret, så man bruger `\section` som hovedstruktur. Der laves ikke en separat titelside, så titel osv. placeres blot øverst på første side. Emacs-introduktionen er lavet med dokumentklassen `article`.

1.4 Fed tekst, kursiv, m.m.

LaTeX finder selv ud af at skrive overskrifter på kapitler og afsnit med fed og stor tekst, men man kan også selv fremhæve tekst ved at gøre den kursiv eller fed. Den anbefalede måde at fremhæve noget som kursiv er med kommandoen `\emph`, der er kort for *emphasize*. Det sidste ord i den foregående sætning er f.eks. lavet med kommandoen `\emph{emphasize}`. Fed tekst laves med kommandoen `\textbf` (hvor `bf` står for **boldface**). F.eks. er sidste ord i parentesen lavet med kommandoen `\textbf{boldface}`. Skrivemaskinetekst laves med kommandoen `\texttt` (hvor `tt` står for `teletype`, som er betegnelsen for en skrivemaskine, der styres af en computer). Du kan nok gætte, hvordan denne kommando bruges.

Når man vil vise programtekster, er det irriterende, at LaTeX ignorerer linjeskift og gentagne blanktegn. Derfor bruger man f.eks. `verbatim` omgivelserne. Omgivelser er en særlig slags kommandoer, der typisk bruges til ting, der står på linjer for sig selv i stedet for midt i en tekst. I stedet for at skrive en enkelt kommando med et argument i krølleparenteser, bruger man kommandoen `\begin` til at starte omgivelsen og `\end` til at afslutte den. Vi har allerede set `\begin{document}` og `\end{document}`, der er en en omgivelse for hele dokumentets indhold. Til programtekst kan vi skrive for eksempel

```
\documentclass{report}

\usepackage[utf8x]{inputenc}
\usepackage{latexsym}
\usepackage[danish]{babel}
\usepackage{graphicx}
\usepackage{hyperref}
\usepackage[all]{hypcap}

\begin{document}

\title{En kort rapport}
\author{Torben Mogensen}
\date{\today}
\maketitle

\chapter{Introduktion}

Dette er et eksempel på en simpel rapport.

\section{Formål}

Formålet er at vise, hvordan en simpel rapport struktureres i \LaTeX.

\section{Målgruppe}

Dokumentet retter sig mod studerende på kurset PoP.

\subsection{Hvad er PoP?}

PoP er en forkortelse for "Programmering og problemløsning".

\chapter{Konklusion}

Det er ikke svært, at få rapporter til at se professionelle ud, når
man bruger \LaTeX.

Ting lavet i Microsoft Word ser amatøragtigt ud i sammenligning.

\section{Videre arbejde}

Vi har kun dækket en brøkdel af, hvad man kan i \LaTeX, så vi
fortsætter straks med flere detaljer.

\end{document}
```

Figur 1.1: Eksempel på simpel rapport i LaTeX

```
\begin{verbatim}
let rec factorial n =
    if n = 0
    then 1
    else n * factorial (n - 1)
\end{verbatim}
```

Som vises som

```
let rec factorial n =
    if n = 0
    then 1
    else n * factorial (n - 1)
```

Alle kommandoer mellem `\begin{verbatim}` og `\end{verbatim}` ignoreres og vises blot som tekst. Af denne grund er `verbatim` omgivelserne brugt til at vise input og output fra LaTeX i dette dokument.

Der findes pakker til LaTeX, der kan lave mere avanceret formatering af programtekst, men dem gemmer vi til senere.

1.5 Krydsreferencer og figurer

En af de stærke sider ved LaTeX er håndteringen af krydsreferencer.

Med kommandoen `\tableofcontents` kan man generere en indholdsfortegnelse. LaTeX finder selv ud af at udfylde indholdsfortegnelsen med de kapitler og afsnit, man har brugt i rapporten. Ved skærmlæsning, kan man følge link fra indholdsfortegnelsen til de angivne kapitler og afsnit. Hvis man bruger `\tableofcontents`, vil `pdflatex` oprette filer med endelserne `.out` og `.toc`. Lige som `.aux`-filer, behøver man ikke at læse disse – de bliver blot brugt af `pdflatex` til at holde styr på tingene.

Hvis man i sin tekst vil lave en krydsreference til et andet kapitel eller afsnit, bruger man kommandoerne `\label` og `\ref`. `\label` placeres ved den `\chapter`, `\section` eller lignende kommando, der definerer det sted, man vil krydsreferere til, og gives et navn som argument. Jeg har f.eks. defineret afsnit 1.11 med kommandoerne

```
\section{Supplerende litteratur}\label{supplerendeLatex}
```

Dette gør, at jeg ved at skrive `afsnit~\ref{supplerendeLatex}` får krydsreferencen "afsnit 1.11". Hvis den genererede PDF fil læses på en skærm i stedet for papir, vil krydsreferencen endvidere vises i en rød kasse, som er et link, man kan følge med et museklik. Hvis man bruger `\ref` med et navn, der ikke er defineret med en `\label`-kommando, skriver LaTeX `??` i stedet for krydsreferencen, og `pdflatex` skriver

```
LaTeX Warning: There were undefined references.
```

Dette kan også ske, hvis man refererer til en label længere fremme i teksten, men det løses ved at køre `pdflatex` en gang til.

Krydsreferencer bruges også til at henvise til figurer, f.eks. figur 1.1. En figur defineres med omgivelsen `figure` og gives en titel med kommandoen `\caption`. En figur kan f.eks. defineres med kommandoerne

Dette er indholdet af figuren.

Figur 1.2: Eksempel på figur

```
\begin{figure}\label{eksempelfigur}

\begin{center}
Dette er indholdet af figuren.
\end{center}

\caption{Eksempel på figur}

\end{figure}
```

som genererer figur 1.2. Figurer nummereres og placeres automatisk. Normalt vil LaTeX placere en figur øverst eller nederst på en side eller på en side for sig i stedet for midt på en side. Den vil aldrig komme på en side før det sted, hvor kommandoen er placeret, men det kan godt være flere sider senere.

Andre former for krydsreferencer er links til websider og fodnoter. Et link til en webside skrives med kommandoen `\url`, f.eks. `\url{http://www.diku.dk}`, der genererer følgende URL: `http://www.diku.dk`. Når dokumentet læses på en skærm, kan man følge linket med et museklik.

En fodnote skrives med kommandoen `\footnote`. Fodnoten nederst på denne side[1] er lavet med kommandoen `\footnote{Ja, denne fodnote}` lige efter ordet "side".

En mere avanceret form for krydsreference er en litterær henvisning. LaTeX har et værktøj kaldet BibTeX, der gør dette nemt. Vi henviser til den supplerende litteratur i afsnit 1.11 for detaljer om, hvordan man bruger BibTeX.
.

1.6 Formler

LaTeX er særlig god til at skrive matematiske formler. Vi har ikke plads her til at dække alle former for formler, så vi holder os til det simple. En formel kan skrives enten inde i et tekstafsnit eller på en linje for sig, hvor den bliver centreret og omgivet af blanke linjer. Se f.eks. forskellen på $ax^2 + bx + c = 0$ og

$$ax^2 + bx + c = 0$$

Formler inde i tekstafsnit omgives af dollartegn, så den viste formel skrives som `$ax^2+bx+c=0$`. For at få formlen på en linje for sig, omgives den af `\[` og `\]`, altså `\[ax^2+bx+c=0\]`.

Som det kan ses, laves hævet tekst i formler ved at sætte tegnet `^` foran. Dette vil kun løfte et enkelt tegn, men man kan sætte krølleklammer rundt om flere tegn for at få dem alle løftet. For eksempel vil teksten `$e^{-x}$` give formlen e^{-x}, mens teksten `$e^-x$` giver formlen e^-x. Sænket tekst laves tilsvarende med `_`, så `$x_0$` giver formlen x_0. De kan kombineres, så `$x_0^n$` giver x_0^n.

[1] Ja, denne fodnote

Generelt bliver blanktegn ignoreret i formler, så `$a bc$` vises som *abc*. Man kan dog sætte et `~` ind, hvis man vil have et blanktegn. Som det ses af eksemplerne, vises bogstaver i kursiv i formler, men matematiske funktioner som log og sin skrives normalt med normal tekst i formler. LaTeX kender de fleste matematiske funktioner, og man kan få dem til at se rigtige ud ved at sætte et backslash foran. Sammenlign `$sin(x)$`, der giver $sin(x)$, med `$\sin(x)$`, der giver $\sin(x)$. Græske bogstaver skrives også med et backslash foran, så `$\alpha\beta\Gamma$` giver $\alpha\beta\Gamma$. Bemærk, at vi fik et stort gamma, da vi skrev `Gamma` med stort begyndelsesbogstav.

LaTeX kan også lave mere avancerede formler med integraler, brøker og meget mere, f.eks.

$$\int_{x=0}^{\infty} \frac{\log(x)}{x^2} dx$$

men vi henviser til supplerende litteratur i afsnit 1.11 for de mere avancerede dele.

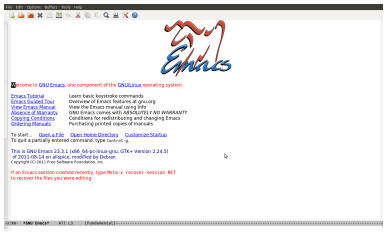
1.7 Billeder

Man kan inkludere billeder i sin tekst med kommandoen `\includegraphics`, For eksempel er billedet af Emacs-skærmen i Emacs-introduktionen lavet med kommandoen `\includegraphics[width=\textwidth]{emacs-welcome.png}`. hvor `emacs-welcome.png` er en fil, der indeholder billedet i billedformatet PNG. Breddeangivelsen skalerer billedet til bredden af teksten, idet `\textwidth` angiver denne bredde. Man kan også angive bredde i f.eks. centimeter og millimeter, så kommandoen `\includegraphics[width=5cm]{emacs-welcome.png}` vil skalere billedet til en bredde på 5 centimeter. Udover PNG kan `includegraphics` også indlæse JPG og PDF.

Man kan centrere billeder (og andet) med `center` omgivelserne, altså som f.eks.

```
\begin{center}
\includegraphics[width=5cm]{emacs-welcome.png}
\end{center}
```

som vil give dette resultat:



1.8 Udvidelsespakker

Vi viste i afsnit 1.2.1 en dokumentindledning med en række `\usepackage` kommandoer. Denne kommando bruges til at tilføje udvidelsespakker til den grundlæggende LaTeX. De viste pakker gør følgende:

`\usepackage[utf8x]{inputenc}` tillader danske bogstaver og andre symboler i input.

`\usepackage{latexsym}` udvider udvalget af matematiske symboler.

`\usepackage[danish]{babel}` betyder, at visse autogenererede tekster (f.eks. "Resumé", "Indhold" og "Kapitel") skrives på dansk, sørger for at ord deles efter (tilnærmede) danske regler, skriver datoer (`\today`) på dansk, og lignende sprogspecifikke ting. Findes til de fleste sprog.

`\usepackage{graphicx}` tillader indsætning af billeder med `\includegraphics` kommandoen.

`\usepackage{hyperref}` **og** `\usepackage[all]{hypcap}` gør krydsreferencer til links.

Der findes tusindvis af pakker til LaTeX. TeX Live sørger for at hente de mest brugte pakker ved installering og henter flere automatisk efter behov. Nogle brugbare pakker er

listings bruges (som alternativ til `verbatim`) til at vise programtekster.

makeidx bruges til at lave stikordslister.

color bruges til at lave farvet tekst.

geometry bruges til at ændre størrelsen af margener, papir, osv.

fontenc bruges til at indkode skriftsnit. `\usepackage[T1]{fontenc}` tillader PostScript type-1 skriftsnit.

cmap Gør teksten i producerede PDF-filer søgbar og kopierbar. Bør komme før `fontenc` og `babel` i indledningen.

Websiden `https://www.ctan.org/` er et arkiv over pakker til LaTeX, og man kan finde beskrivelser og vejledninger til over 5000 pakker på denne side. Man kan søge efter navn eller emne. Hvis man f.eks. går ind i emnelisten under "music", finder man 35 pakker til formatering af noder, guitarakkorder, sanghæfter, osv.

1.9 LaTeX og Emacs

Emacs har en LaTeX-mode, der blandt andet viser kommandonavne i farver og viser kursiv og fed tekst som sådan. Hvis man har LaTeX-mode installeret i sin Emacs, skifter Emacs automatisk til denne mode, når man redigerer i en fil med filendelsen `.tex`.

LaTeX-mode tilføjer et par LaTeX-specifikke kommandoer til Emacs: C-c C-e tilføjer en `\end{}`-kommando svarende til den nærmeste uparrede `\begin{}`-kommando, og C-c C-o tilføjer et par af `\begin{}` og `\end{}`-kommandoer, hvor man kan vælge navnet ved at skrive det i minibufferen.

I LaTeX-mode har Emacs også en TeX-menu, der blandt andet kan køre kommandoen `latex` på den fil, man redigerer i, og vise resultatet i et andet vindue. Denne kommando producerer dog ikke PDF, men et andet format, der hedder DVI, så selv om det er fint at bruge denne menufunktion til at finde

fejl i sit dokument, så bør man bruge kommandoen `pdflatex` til sidst for at få output i PDF. Endvidere kan `latex`-kommandoen ikke håndtere billeder i de samme formater som `pdflatex`.

1.10 Finpudsning

Der er et par småting, men skal være opmærksom på, når man bruger LaTeX, så man får det pænest mulige resultat:

- Blanktegn lige efter en kommando, der ikke har parametre i krølleklammer, ignoreres. For eksempel vil teksten `\LaTeX styrer!` give resultatet LaTeXstyrer! For at få et blanktegn ind alligevel give kommandoen en tom parameter, altså `\LaTeX{} styrer!`, som giver resultatet LaTeX styrer!. Det er ikke nødvendigt, hvis kommandoen følges af andet end blanktegn eller linjeskift, som f.eks. punktum eller komma. Alternativt kan man skrive `\LaTeX\ styrer!`, altså sætte et backslash før blanktegnet, så det ikke bliver undertrykt.

- Som det ses herover, kan LaTeX nogen gange lade tekst strække sig ud over højre kant af tekst med lige højremargen. Hvis dette sker, vil `pdflatex` give beskeden `Overfull \hbox`, men ikke standse. Denne besked er en *advarsel* og ikke en *fejlmeddelelse*, da den netop ikke hindrer produktion af output. Efter beskeden `Overfull \hbox`, angives hvor meget, teksten strækker sig ud, i dette tilfælde `(24.74678pt too wide)` samt en angivelse af, hvor i teksten, dette forekommer. Forkortelsen `pt` står for "punkt"; som er en typografisk måleenhed, der er ca. 0.35mm, så 24.74678pt er ca. 8.66mm. I det viste tilfælde hjælper det at sætte `{}` efter `\LaTeX`, men i andre tilfælde kan man sætte kommandoen `\newline` før det ord, der er for langt. LaTeX forsøger selv at dele ord efter regler for dansk orddeling, men hvis den er i tvivl, lader den være. Hvis man får et langt ord, der stikker ud til højre, kan man hjælpe LaTeX med orddelingen ved at sætte et `\-` det sted, man gerne vil have delt ordet. LaTeX vil kun indsætte en bindestreg, der hvor ordet deles (hvis det deles), så man kan sagtens indsætte flere for at give LaTeX nogle valgmuligheder, f.eks. `ord\-de\-ling`.

  Man kan også bruge `\-`, hvis LaTeX laver en forkert orddeling. LaTeX bruger et antal tommelfingerregler til orddeling, og det går nogle gange galt. F.eks. vil LaTeX dele ordet "menupunkt" som "men-upunkt", hvilket ikke er særligt kønt. For at hindre dette, kan man skrive `menu\-punkt`, som fortæller, at ordet kun må deles der.

  Lad dog være med at bekymre dig om forkert orddeling eller lige højrekant indtil du er færdig med at skrive rapporten: Det kan sagtens være, at du skriver om i teksten, så det ikke længere er et problem, og så er det spildt arbejde at bruge tid på orddeling eller tvungne linjeskift.

- Der er forskel på bindestreger (-) og tankestreger (–). Bindestreger bruges til at dele ord og til visse ordsammensætninger – f.eks. IT-sikkerhed – og tankestreger bruges til indskud, som vist lige før. En tankestreg skrives i LaTeX som to bindestreger efter hinanden, altså `--`.

- LaTeX skifter aldrig linje mellem ord, der er adskilt kun af `~`. Derfor kan man bruge dette tegn til at forhindre linjeskift mellem to ord. Det kan f.eks. bruges til at forhindre et linjeskift mellem "kapitel" og "2" i "kapitel 2".

- LaTeX laver lidt større mellemrum efter et punktum, der afslutter en sætning, end efter f.eks. et komma. Den måde, LaTeX ser, om et punktum afslutter en sætning, er lidt primitiv: Hvis punktummet står lige efter et stort bogstav, tror LaTeX, at der er tale om et forkortet navn, f.eks. H. C. Andersen, men hvis det kommer efter et lille bogstav, tror LaTeX, at det er slutningen af en sætning. For det meste fungerer dette fint, men efter forkortelser som f.eks. "f.eks." er det forkert. Man kan undgå det lange mellemrum efter en forkortelse ved at skrive en backslash lige efter punktummet, som f.eks. `f.eks.\ H. C. Andersen`. Modsat kan man sætte `\@` mellem et stort bogstav og et punktum, hvis man gerne vil have, at der kommer ekstra plads efter punktummet. Det bruges hvis man afslutter en sætning med en forkortelse, der består af store bogstaver, som f.eks. DVD. Den foregående sætning er f.eks. afsluttet med `f.eks.\ DVD\@.`

1.11 Supplerende litteratur

Der findes mange vejledninger til LaTeX på forskellige niveauer. En almindelig brugt vejledning er "The Not So Short Introduction to LaTeX2ε", som findes på `https://tobi.oetiker.ch/lshort/lshort.pdf`. Den kan virke lidt uoverkommelig ved første øjekast, men den er god som oplagsværk.

Lidt mere tilgængelig er den LaTeX tutorial, der ligger på `http://www.latex-tutorial.com/tutorials/`.

Der findes en meget omfattende bog om LaTeX på dansk, skrevet af Leif Madsen fra Aarhus Universitet. Den kan hentes på
`http://math.au.dk/videnudveksling/latex/bog/`.

1.12 Fortvivl ikke!

LaTeX kan i starten virke uoverskueligt, men man vænner sig hurtigt til at bruge det, og efter nogen tid, vil man sætte pris på dets muligheder. Ældre studerende kan i reglen hjælpe dig, hvis du har spørgsmål om LaTeX, og hvis alt andet fejler, er Google din ven.