

En superkort introduktion til L^AT_EX

Torben Mogensen og Jon Sparring

7. september 2022

Resumé

I kurset “Programmering og problemløsning” (PoP) bruger vi værktøjerne Overleaf og L^AT_EX.

Overleaf er et online tekstredigeringsværktøj (<https://overleaf.com>), der specialdesignet til at køre L^AT_EX og til at samarbejde om at skrive L^AT_EX kode. Ulig f.eks. Microsoft Word og Libre Office Writer redigerer man med Overleaf “ren” tekst: Der er ikke formatering såsom fed skrift, kursiv, osv. Til dette bruger vi L^AT_EX.

L^AT_EX (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 L^AT_EX-dokumenter til PDF ved brug af værktøjet `pdflatex`.

Indhold

1	L^AT_EX	2
1.1	Brug af L ^A T _E X	2
1.2	Syntaks	4
1.3	Et struktureret dokument	6
1.4	Fed tekst, kursiv, m.m.	6
1.5	Krydsreferencer og figurer	8
1.6	Formler	9
1.7	Billeder	10
1.8	Udvidelsespakker	11
1.9	Finpudsning	12
1.10	Supplerende litteratur	13
1.11	Fortvivl ikke!	14

Kapitel 1

L^AT_EX

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

L^AT_EX udtales “lahtek” og ikke “lateks”, da de tre bogstaver i T_EX faktisk er de græske bogstaver tau, epsilon og ksi, så denne del af ordet udtales “tech” eller “tek”. L^AT_EX er en videreudvikling af Donald Knuths T_EX 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 L^AT_EX og f.eks. Microsoft Word er, at man forbereder sine L^AT_EX-dokumenter i Overleaf og derefter konverterer dem til PDF ved at trykke på Compile knappen. Det er i princippet ikke anderledes end den måde, et HTML-dokument konverteres til tekst og billeder i et vindue i en browser. Et L^AT_EX-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 L^AT_EX-programmets opgave at fortolke disse koder og bruge dem til at formatere dokumentets tekst. Endvidere opdeler L^AT_EX teksten i sider, som er egnede til udskrivning, visning på en projektor, eller lignende, og indsætter selv sidenumre osv.

1.1 Brug af L^AT_EX

For at afprøve installeringen af L^AT_EX, anbefaler vi, at du starter med et meget simpelt dokument. Log in på Overleaf, tryk på “New Project”, vælg en skabelon som f.eks. “Blank project”, giv dit projekt et navn, og tryk på “Create”. Nu skal du se et næsten tomt dokument i midten af din skærm med rå L^AT_EX kode med det vigtigste elementer forudindsat såsom:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\title{test}
\author{sporrings }
\date{August 2022}
```

```

\begin{document}
\maketitle
\section{Introduction}
\end{document}

```

Allerede nu burde dokumentet være oversat, og i højre side af din skærm skulle du se en Pdf af resultatet. I venstre vil du typisk se en liste af filer, samt et overblik over dit dokument. Vi kommer tilbage til, hvad de enkelte dele af filen betyder senere.

Prøv at opdatere L^AT_EX koden ved at erstatte `\section{Introduction}` med `Hello World!`, evt. rette dit navn og titel, og oversæt det.

```

\documentclass{article}
\usepackage[utf8]{inputenc}

\title{My first LATEX document}
\author{Jon Sparring}
\date{August 2022}

```

```

\begin{document}

```

```

\maketitle

```

```

Hello World!

```

```

\end{document}

```

L^AT_EX kommandoer begynder med `'\'` som vi på Dansk bare kalder for 'backslash', og nogle tager argumenter som enten er indrammet af `'[]'` eller `'{}'` parenteser. Hvis man kommer til at skrive en kommando, som L^AT_EX ikke forstår, vil den give en fejlmeddelelse, og dem kan man se ved at trykke på det lille ikon til højre for "Compile" knappen. F.eks. hvis man kommer til at skrive `\end{dokument}` med 'k' istedet for 'c'

```

\documentclass{report}

```

```

\begin{document}

```

```

Hello World

```

```

\end{dokument}

```

får man mange fejl, og den først hedder noget i retningen af "L^AT_EX Error: `\begin{document}` ended by `\end{dokument}`". Dette betyder at `\begin{document}` og `\end{dokument}` hører sammen som et par, og L^AT_EX kan kun finde den ene. Bemærk, at forekomsten af en fejl medfører, at der ikke produceres nogen PDF fil. En anden typisk fejl kan være at skrive `\edn{document}` i stedet for `\end{dokument}` får vi en anden fejl som siger "Undefined control sequence", hvilket betyder, at L^AT_EX ikke kender kommandoen `\edn`.

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/12h/errors.html>.

1.2 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 stavereregler*, 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{cmap}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{amssymb}
\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 `\textbackslash{}`, 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`.

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

```

\documentclass{report}
\title{En superkort introduktion til \LaTeX}
\author{Torben Mogensen}
\date{\today}
\begin{document}
\maketitle
\end{dokument}

```

Kommandoerne `\title`, `\author` og `\date` definerer title, forfatter(e) og dato og skal stå i preamble, altså inden `\begin{document}`. Kommandoen `\maketitle` skal stå i toppen af hovedteksten altså efter `\begin{document}`. Kommandoen `\LaTeX` genererer L^AT_EX-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}`). L^AT_EX finder selv ud af nummereringen, man skal blot angive titlerne på kapitler og afsnit. I rapporter starter kapitler på nye sider, så L^AT_EX-dokumentet vist i figur 1.1 vil generere to korte kapitler startende på hver sin side. Prøv selv at kopiere teksten ind i Overleaf og oversæt den.

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 L^AT_EX, 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.

1.4 Fed tekst, kursiv, m.m.

L^AT_EX 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 parentesens 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 L^AT_EX 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


```

\documentclass{report}
\usepackage{cmap}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{amssymb}
\usepackage[danish]{babel}
\usepackage{graphicx}
\usepackage{hyperref}
\usepackage[all]{hypcap}

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

\begin{document}
\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 L^AT_EX

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 omgivelse for hele dokumentets indhold. Til programtekst kan vi skrive for eksempel

```
\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 L^AT_EX i dette dokument.

Der findes pakker til L^AT_EX, 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 L^AT_EX er håndteringen af krydsreferencer.

Med kommandoen `\tableofcontents` kan man generere en indholdsfortegnelse. L^AT_EX 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 1.10 med kommandoerne

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

Dette gør, at jeg ved at skrive `afsnit~\ref{supplerendeLatex}` får krydsreferencen “afsnit 1.10”. 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 L^AT_EX ?? i stedet for krydsreferencen, og `pdflatex` skriver

LaTeX Warning: There were undefined references.

Dette er indholdet af figuren.

Figur 1.2: Eksempel på figur

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`. Det er vigtigt at placere `\label` *efter* `\caption`. En figur kan f.eks. defineres med kommandoerne

```
\begin{figure}
\begin{center}
Dette er indholdet af figuren.
\end{center}
\caption{Eksempel på figur}
\label{eksempelfigur}
\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 side¹ 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 $\text{BIB}\text{\TeX}$, der gør dette nemt. Vi henviser til den supplerende litteratur i afsnit 1.10 for detaljer om, hvordan man bruger $\text{BIB}\text{\TeX}$. .

1.6 Formler

\LaTeX er specielt 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

¹Ja, denne fodnote

tegn for at få dem alle løftet. For eksempel vil teksten `e^{-x}` give formelen e^{-x} , mens teksten `e^{-x}` giver formelen e^{-x} . Sænk et tegn laves tilsvarende med `_{`, så `x_0` giver formelen x_0 . De kan kombineres, så `x_0^n` giver x_0^n . 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. L^AT_EX 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. L^AT_EX kan også lave mere avancerede formler med integraler, brøker og meget mere, f.eks.

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

Ovennævnte formel laves med

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

Dette er et avanceret emne og vil ikke blive dækket yderligere. Se supplerende litteraturhenvisninger i afsnit 1.10.

1.7 Billeder

Man kan inkludere billeder i sin tekst med kommandoen `\includegraphics`, For eksempel hvis man har et billede i PNG format, som hedder `cat.png`, så skal man uploade det til Overleaf og skrive kommanden

```
\includegraphics[width=5cm]{cat.png}
```

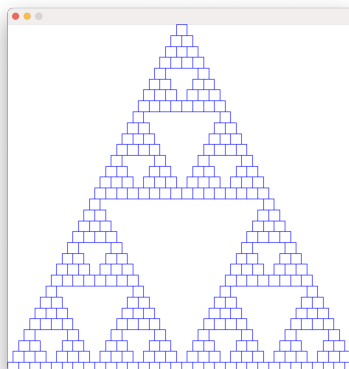
Breddeangivelsen skalerer billedet til bredden af teksten, idet `\textwidth` angiver denne bredde. Man kan også angive bredde i f.eks. centimeter og millimeter, så kommandoen `\includegraphics[width=5cm]{cat.png}` vil skalere billedet til en bredde på 5 centimeter. Udover PNG kan `includegraphics` også indlæse JPG og PDF.

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

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

Figurer placeres ofte i `figure` omgivelser, og det er ofte en god ide at bruge et relativ breddemål, såsom `width=0.45\linewidth`, så figurstørrelserne automatisk indretter sig efter globale formateringskommandoer f.eks. sat med `geometry` pakken, se afsnit 1.8. F.eks. blev figur 1.3 lavet med følgende kode

```
\begin{figure}
\begin{center}
\includegraphics[width=0.45\linewidth]{sierpinski.png}
\end{center}
\caption{Sierpinskis trekant}
\label{fig:sierpinski}
\end{figure}
```



Figur 1.3: Sierpinskis trekant

1.8 Udvidelsespakker

Vi viste i afsnit 1.2 en dokumentindledning med en række `\usepackage` kommandoer. Denne kommando bruges til at tilføje udvidelsespakker til den grundlæggende \LaTeX . De viste pakker gør følgende:

`\usepackage{cmap}` sørger for, at tekst kan klippes fra PDF'en uden, at æ, ø og å og andre mærkelige bogstaver vises forkert.

`\usepackage[T1]{fontenc}` indikerer, at vi bruger PostScript type-1 skriftsnit.

`\usepackage[utf8]{inputenc}` tillader danske bogstaver og andre symboler i input. Hvis du ikke har den nyeste version af \LaTeX , kan du bruge `utf8x` i stedet.

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

`\usepackage[danish]{babel}` betyder, at visse autogeneratede 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.

newunicodechar bruges sammen med utf8-pakken til at udvide mængden af unicodetegn, man kan have i input. f.eks. vil kommandoen

```
\newunicodechar{×}{\ensuremath\times}
```

tillade brugen af ×-tegnet i input, hvor man ellers skulle skrive `\times` i math mode og `\times` i text mode. `\ensuremath` er en kommando, der skriver sit argument i math mode, uanset om man er i math mode eller ej.

Websiden <https://www.ctan.org/> er et arkiv over pakker til L^AT_EX, 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 Finpudsning

Der er et par småting, men skal være opmærksom på, når man bruger L^AT_EX, så man får det pæneste mulige resultat:

- Blanktegn lige efter en kommando, der ikke har parametre i krølleklammer, ignoreres. For eksempel bliver `\LaTeX styrer!` formatteret til L^AT_EX styrer! For at få et blanktegn ind alligevel, kan man give kommandoen en tom parameter, altså `\LaTeX{} styrer!`, som giver resultatet L^AT_EX 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 L^AT_EX 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 *ad-varsel* 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 (`10.22188pt 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 `10.22188pt` er ca. 3.58mm. 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. L^AT_EX 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 L^AT_EX med orddelingen ved at sætte et `\-` det sted, man gerne vil have delt ordet. L^AT_EX vil kun indsætte en bindestreg, der hvor ordet deles (hvis det deles), så man kan sagtens indsætte flere for at give L^AT_EX nogle valgmuligheder, f.eks. `ord\ -de\ -ling`.

Man kan også bruge `\-`, hvis L^AT_EX laver en forkert orddeling. L^AT_EX bruger et antal tommelfingerregler til orddeling, og det går nogle gange galt. F.eks. vil L^AT_EX dele ordet “menupunkt” som “men-upunkt”, hvilket ikke

er særligt kønt (og slet ikke korrekt). For at hindre dette, kan man skrive `menu\pukt`, som fortæller, at ordet kun må deles der. Hvis ordet ikke deles, vises bindestregen ikke.

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\@`.

1.10 Supplerende litteratur

Der findes mange vejledninger til \LaTeX på forskellige niveauer. En almindelig brugt vejledning er “The Not So Short Introduction to \LaTeX 2 ϵ ”, 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/>.

1.11 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.