



L^AT_EX for nybegynnere

Dag Langmyhr

8. august 2003



**Institutt for informatikk
Universitetet i Oslo**

L^AT_EX for nybegynnere

Dag Langmyhr

Dette lille skrivet er en innføring i L^AT_EX beregnet på nybegynnere, dvs brukere som aldri har benyttet dette programmet. Målet er ikke å gjøre leseren til ekspert i L^AT_EX, men gjøre ham eller henne i stand til å bruke L^AT_EX til å skrive for eksempel en rapport eller en hovedoppgave.

Rent konkret er denne innføringen organisert slik at venstre side inneholder *kildekoden* i L^AT_EX (det som brukeren skriver), mens høyre side viser *resultatet* (det som til slutt kommer på det trykte arket). Hensikten er at brukeren på den måten kan se eksempler på bruken av det som omtales. For å få fullt utbytte av innføringen, er det viktig å studere begge sidene parallelt.

Innføringen starter med enkle begreper og kommandoer og innfører nye etter hvert samtidig som effekten av de nye kommandoene demonstreres. Stort sett blir det bare brukt kommandoer som har vært forklart tidligere, men i noen få tilfelle benyttes ting som omtales senere. Noen svært få avanserte kommandoer forklares kun i kommentarer i kildekoden; dette er enda en grunn til å lese kildekoden og resultatteksten samtidig.

```

\documentclass[11pt,norsk,a4paper]{article}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{babel,graphicx,mathppl, textcomp,varioref}

\title{\LaTeX{} for nybegynnere}
\author{Dag Langmyhr\ \ Institutt for informatikk\ \
  Universitetet i Oslo\ \ \texttt{dag@ifi.uio.no}}

\tolerance = 5000      % LaTeX er normalt streng når det gjelder linjebrytingen.
\hbadness = \tolerance % Vi vil være litt mildere, særlig fordi norsk har så
\pretolerance = 2000   % mange lange sammensatte ord.

\begin{document}

\maketitle{}

\section{Bakgrunn}
\LaTeX{} er et meget avansert program for dokumentproduksjon, altså typesetting
av artikler, rapporter, bøker, brev og annet. Som alle andre avanserte
programmer kreves det tid og innsats å lære programmet å kjenne. Dette lille
skrivet er ment å gi en kortfattet innføring slik at nye brukere ledes gjennom
de første vanskelighetene. Det vil imidlertid langfra fortelle den hele og
fulle sannhet om \LaTeX{}, så helt på slutten står en oversikt om hvor man kan
finne ytterligere opplysninger.

\LaTeX{} skiller seg fra de fleste andre programmer for dokumentproduksjon ved
at det ikke er interaktivt. Arbeidsformen ligner mer på den som brukes ved
programmering: Brukeren skriver et «program» som består av tekst ispedd diverse
formateringskommandoer; dette programmet blir så «kompilert» av \LaTeX{} før
det blir «eksekvert» av skriveren eller et program som viser det ferdige
dokumentet på skjermen. Helt sist i dette heftet finnes et diagram som viser
gangen i dette.

\section{Tekst og kommandoer}
Som det er tydelig å se på venstre side, består mesteparten av et
\LaTeX{}-dokument av vanlig tekst. Denne teksten kan stort sett skrives rett
frem, men det er et par ting man bør merke seg.

Det viktigste når man skriver vanlig tekst er å passe seg for visse tegn som
har en spesiell funksjon i \LaTeX{}; nærmere bestemt dreier det seg om tegnene
\verb@\, \verb@{\, \verb@}, \verb@_, \verb@~, \verb@#, \verb@$,
\verb@%, \verb@& og \verb@~. Det beste er å unngå disse tegnene inntil man
vet hvilken effekt de har.

Når man skriver vanlig tekst, behøver man ikke være så nøye med blanke tegn
i teksten. I motsetning til mange andre programmer vil \LaTeX{} betrakte to
eller flere blanke som ekvivalent med \'{e}n blank, så det gjør ikke noe om
det kommer inn noen ekstra blanke her og
der. Det ferdige resultatet
blir det samme, heldigvis.

```

L^AT_EX for nybegynnere

Dag Langmyhr
Institutt for informatikk
Universitetet i Oslo
dag@ifi.uio.no

8. august 2003

1 Bakgrunn

L^AT_EX er et meget avansert program for dokumentproduksjon, altså typesetting av artikler, rapporter, bøker, brev og annet. Som alle andre avanserte programmer kreves det tid og innsats å lære programmet å kjenne. Dette lille skrevet er ment å gi en kortfattet innføring slik at nye brukere ledes gjennom de første vanskelighetene. Det vil imidlertid langfra fortelle den hele og fulle sannhet om L^AT_EX, så helt på slutten står en oversikt om hvor man kan finne ytterligere opplysninger.

L^AT_EX skiller seg fra de fleste andre programmer for dokumentproduksjon ved at det ikke er interaktivt. Arbeidsformen ligner mer på den som brukes ved programmering: Brukeren skriver et «program» som består av tekst ispedd diverse formateringskommandoer; dette programmet blir så «kompilert» av L^AT_EX før det blir «eksekvert» av skriveren eller et program som viser det ferdige dokumentet på skjermen. Helt sist i dette heftet finnes et diagram som viser gangen i dette.

2 Tekst og kommandoer

Som det er tydelig å se på venstre side, består mesteparten av et L^AT_EX-dokument av vanlig tekst. Denne teksten kan stort sett skrives rett frem, men det er et par ting man bør merke seg.

Det viktigste når man skriver vanlig tekst er å passe seg for visse tegn som har en spesiell funksjon i L^AT_EX; nærmere bestemt dreier det seg om tegnene `\`, `{`, `}`, `_`, `^`, `#`, `$`, `%`, `&` og `~`. Det beste er å unngå disse tegnene inntil man vet hvilken effekt de har.

Når man skriver vanlig tekst, behøver man ikke være så nøye med blanke tegn i teksten. I motsetning til mange andre programmer vil L^AT_EX betrakte to eller flere blanke som ekvivalent med én blank, så det gjør

Når man skriver tekst, er det av og til nødvendig å dele ord for å få dem til å passe inn. Korrekt orddeling er langt fra trivielt, men `\LaTeX{}-type\settings\programmet` er ganske flink til slikt; den gjør det oftest riktig, men dessverre ikke alltid. Derfor er det av og til nødvendig å gi den litt hjelp, spesielt i sammensatte ord med bindestrek. Hjelpen består i å sette inn de to tegnene `\verb@-@` der ord kan deles hvis vi ser at `\LaTeX{}` ikke har klart dette skikkelig selv. (Det er liten vits i å gjøre dette hele tiden; det er tidsnok når vi ser at `\LaTeX{}` deler galt eller gir feilmeldinger om «Overfull hbox».)

Som skille mellom avsnittene benyttes blanke linjer, og igjen, det er det samme om det er `\{e}n` eller flere blanke linjer.

Av og til har man behov for å slå av formateringen fordi man ønsker å skrive tegn som egentlig har en spesiell funksjon i `\LaTeX{};` jeg har allerede gjort det flere ganger for å fortelle om spesialtegnene! Da kan man skrive `\verb:\verb@...@:` så vil alt som står mellom de to `\verb:@:-ene` komme nøyaktig som det står; det vil til og med bli skrevet i et eget skrivemaskinlignende skriftsnitt for å gjøre det klart at dette er et eksempel. (Skilletegnene behøver ikke være to `\verb:@:-er;` de fleste tegn kan benyttes til dette, for eksempel to `\verb:@:-er.`)

Helt til sist: Noen ganger ønsker man å skrive et mellomrom mellom to ord, men allikevel hindre `\LaTeX{};` i å skille de to delene på hver sin linje, f.eks.~i store tall som 1~000~000. Dette gjøres ved å benytte en `\verb@~@` i stedet for den blanke.

`\subsection{Kommentarer}`

Ofte kan det være lurt å legge kommentarer inn i `\LaTeX{}-teksten;` dette er tekst som ikke har noen innflytelse på det ferdige dokumentet, men som kan være nyttig for folk som senere skal studere kildefilen.

% Her har jeg lagt en liten kommentar som ikke kommer med i det
% ferdige dokumentet; den bare ignoreres.

I `\LaTeX{};` benyttes `\verb@%@-tegnet` for % Her er noen
kommentarer. Når `\LaTeX{};` oppdager % kommentarer
et `\verb@%@-tegn,` vil resten av linjen % sist på
ignoreres. % linjen.

`\subsection{Enkle kommandoer}`

Det er imidlertid ikke nok å skrive et dokument kun med vanlig tekst; vi ønsker for eksempel å benytte symboler som ikke finnes på tastaturet, eller fortelle `\LaTeX{};` litt om hvorledes formateringen skal foregå. Dette oppnår vi ved å gi kommandoer til `\LaTeX{}.`

I `\LaTeX{};` skrives kommandoer slik: `\verb@xxx{ }@,` der bokstavene mellom `\verb@` og venstre klamme (`\verb@{ }`) er navnet på kommandoen. Tre enkle eksempler på slike kommandoer er `\verb@today{ }@` som gir dagens dato: `\today{ },` `\verb@dots{ }@` som gir tre punktumer med passe avstand (`\dots{ }`), og `\verb@LaTeX{ }@` som skriver `\LaTeX{};` på den «offisielle» måten.

ikke noe om det kommer inn noen ekstra blanke her og der. Det ferdige resultatet blir det samme, heldigvis.

Når man skriver tekst, er det av og til nødvendig å dele ord for å få dem til å passe inn. Korrekt orddeling er langt fra trivielt, men \LaTeX -typesettingsprogrammet er ganske flink til slikt; den gjør det oftest riktig, men dessverre ikke alltid. Derfor er det av og til nødvendig å gi den litt hjelp, spesielt i sammensatte ord med bindestrek. Hjelpen består i å sette inn de to tegnene \- der ord kan deles hvis vi ser at \LaTeX ikke har klart dette skikkelig selv. (Det er liten vits i å gjøre dette hele tiden; det er tidsnok når vi ser at \LaTeX deler galt eller gir feilmeldinger om «Overfull hbox».)

Som skille mellom avsnittene benyttes blanke linjer, og igjen, det er det samme om det er én eller flere blanke linjer.

Av og til har man behov for å slå av formateringen fordi man ønsker å skrive tegn som egentlig har en spesiell funksjon i \LaTeX ; jeg har allerede gjort det flere ganger for å fortelle om spesialtegnene! Da kan man skrive `\verb@...@` så vil alt som står mellom de to @-ene komme nøyaktig som det står; det vil til og med bli skrevet i et eget skrivemaskinlignende skriftsnitt for å gjøre det klart at dette er et eksempel. (Skilletegnene behøver ikke være to @-er; de fleste tegn kan benyttes til dette, for eksempel to :-er.)

Helt til sist: Noen ganger ønsker man å skrive et mellomrom mellom to ord, men allikevel hindre \LaTeX i å skille de to delene på hver sin linje, f.eks. i store tall som 1 000 000. Dette gjøres ved å benytte en ~ i stedet for den blanke.

2.1 Kommentarer

Ofte kan det være lurt å legge kommentarer inn i \LaTeX -teksten; dette er tekst som ikke har noen innflytelse på det ferdige dokumentet, men som kan være nyttig for folk som senere skal studere kildefilen.

I \LaTeX benyttes %-tegnet for kommentarer. Når \LaTeX oppdager et %-tegn, vil resten av linjen ignoreres.

2.2 Enkle kommandoer

Det er imidlertid ikke nok å skrive et dokument kun med vanlig tekst; vi ønsker for eksempel å benytte symboler som ikke finnes på tastaturet, eller fortelle \LaTeX litt om hvorledes formateringen skal foregå. Dette oppnår vi ved å gi kommandoer til \LaTeX .

I \LaTeX skrives kommandoer slik: `\xxx{}`, der bokstavene mellom \ og venstre klamme ({}) er navnet på kommandoen. Tre enkle eksempler på slike kommandoer er `\today{}` som gir dagens dato: 8. august 2003, `\dots{}` som gir tre punktumer med passe avstand (...), og `\LaTeX{}` som skriver \LaTeX på den «offisielle» måten.

En `\LaTeX{}`-kommando kan også ha et navn som er ett enkelt ikke-alfa\bet\isk tegn; eksempler på dette er kommandoene som lager tegnene `\%`, `\$`, `\#`, `\{`, `\}`, `_` og `\&`. (Disse tegnene kunne ikke skrives direkte siden de har spesielle oppgaver i `\LaTeX{}`, som vi skal se siden.) Her trenger vi ikke ha `\verb@{}@` etter kommandoen.

Andre symboler som man får tak i ved egne kommandoer, er `\pounds{}`, `\dag{}`, `\ddag{}`, `\copyright{}`, `\textasciicircum{}`, `\textasciitilde{}`, `\textbackslash{}`, `\S{}` og `\ss{}`. I avsnittet om typesetting av matematiske formler kommer det mange flere slike kommandoer.

(Egentlig er regelen om at det alltid skal være klammer etter et kommandoenavn, noe for stiv. Klammene kan ofte sløyfes, men dette kan gi uønskete effekter. Inntil man er fortrolig med `\LaTeX{}` bør man altså benytte disse klammene.)

`\subsection{Kommandoer med parametre}`

Av og til er det ikke nok med enkle kommandoer, vi kan ønske kommandoer med parametre også. Et eksempel jeg allerede har brukt, er `\verb@'\{e}@` som gir en 'e' med aksent: `\'\{e}`. Her angir kommandoen hvilken aksent som skal brukes, og parameteren, som plasseres inni klammene, gir bokstaven som aksenten skal plasseres på.

`\LaTeX{}` har et godt utvalg aksenter og andre vedheng på bokstaver: `\'\{e}`, `\'\{a}`, `\~\{o}`, `\"\{o}`, `\~\{n}`, `\c{c}` og enda flere. Man kan selvfølgelig også skrive disse tegnene ved å taste for eksempel é direkte, men det er ikke alltid like lett å finne ut hvorledes man gjør dette på et tastatur.

`\subsubsection{Kommandoer med opsjoner}`

Hvis en kommando først har parametre, skal disse alltid være med, og alltid det samme antallet. I noen tilfelle, derimot, ønsker vi å angi opsjoner, det vil si parametre som kan være med eller utelates etter ønske. I `\LaTeX{}` angis opsjoner med hakeparenteser (`\verb@[...]@`). Et eksempel er bruken av `\verb@\documentclass@` på den aller første linjen i denne filen.

`\subsection{Omgivelser}`

En siste variant av kommandoer er omgivelser, hvor vi angir starten og slutten på en del av teksten hvor noe spesielt skal gjelde. Starten på omgivelsen `\verb@navn@` angis med `\verb@\begin{navn}@` og slutten med `\verb@\end{navn}@`.

`\subsubsection{Lister}`

Et meget nyttig og ofte brukt eksempel på en omgivelse er en liste:

`\begin{itemize}`

`\item{}` Den vanligste formen for liste heter `\emph{itemize}` og er en slags «punktvis liste».

En \LaTeX -kommando kan også ha et navn som er ett enkelt ikke-alfabetisk tegn; eksempler på dette er kommandoene som lager tegnene %, \$, #, {, }, _ og &. (Disse tegnene kunne ikke skrives direkte siden de har spesielle oppgaver i \LaTeX , som vi skal se siden.) Her trenger vi ikke ha {} etter kommandoen.

Andre symboler som man får tak i ved egne kommandoer, er £, †, ‡, ©, ^, ~, \, § og ß. I avsnittet om typesetting av matematiske formler kommer det mange flere slike kommandoer.

(Egentlig er regelen om at det alltid skal være klammer etter et kommando navn, noe for stiv. Klammene kan ofte sløyfes, men dette kan gi uønskete effekter. Inntil man er fortrolig med \LaTeX bør man altså benytte disse klammene.)

2.3 Kommandoer med parametre

Av og til er det ikke nok med enkle kommandoer, vi kan ønske kommandoer med parametre også. Et eksempel jeg allerede har brukt, er `\'e` som gir en 'e' med aksent: é. Her angir kommandoen hvilken aksent som skal brukes, og parameteren, som plasseres inni klammene, gir bokstaven som aksenten skal plasseres på.

\LaTeX har et godt utvalg aksenter og andre vedheng på bokstaver: é, à, ô, ö, ñ, ç og enda flere. Man kan selvfølgelig også skrive disse tegnene ved å taste for eksempel é direkte, men det er ikke alltid like lett å finne ut hvorledes man gjør dette på et tastatur.

2.3.1 Kommandoer med opsjoner

Hvis en kommando først har parametre, skal disse alltid være med, og alltid det samme antallet. I noen tilfelle, derimot, ønsker vi å angi opsjoner, det vil si parametre som kan være med eller utelates etter ønske. I \LaTeX angis opsjoner med hakeparenteser ([...]). Et eksempel er bruken av `\documentclass` på den aller første linjen i denne filen.

2.4 Omgivelser

En siste variant av kommandoer er omgivelser, hvor vi angir starten og slutten på en del av teksten hvor noe spesielt skal gjelde. Starten på omgivelsen navn angis med `\begin{navn}` og slutten med `\end{navn}`.

2.4.1 Lister

Et meget nyttig og ofte brukt eksempel på en omgivelse er en liste:

- Den vanligste formen for liste heter *itemize* og er en slags «punktvis liste».

`\item{}` Listen innledes med at det står `\verb@\begin{itemize}@` og avsluttes med `\verb@\end{itemize}@`.

Vanligvis står det ingen blank linje etter `\verb@\end{itemize}@` (eller tilsvarende `\verb@\end@` som avslutter andre typer lister) fordi vi ikke ønsker innrykk av den etterfølgende linjen.

`\item{}` Hvert nytt punkt i listen innledes med et kall på kommandoen `\verb@\item{}`.

`\item{}` Vi kan se at innholdet i listen er rykket litt inn i forhold til resten av teksten, og at hvert punkt angis med en svart prikk.

`\end{itemize}`

En annen form for liste kalles `\emph{enumerate}`, og den inneholder fortløpende nummerering av punktene:

`\begin{enumerate}`

`\item{}` Første punktet er alltid nr.~1.

`\item{}` `\LaTeX{}` vil automatisk sørge for å nummerere punktene riktig.

`\end{enumerate}`

Den siste formen for liste kalles `\emph{description}`, og den lager noe som minner om et leksikon der hvert punkt starter med et navn i fet skrift:

`\begin{description}`

`\item[Tittel]` Hvert punkt starter med et ord, for eksempel navnet på noe som skal beskrives.

`\item[Notasjon]` Innledningsordet angis som opsjon til `\verb@\item@`.

`\item[Flere nivåer]` Noe av det fine med lister er at de uten videre kan brukes i flere nivåer; man kan altså ha lister inni lister.

`\begin{itemize}`

`\item{}` Denne punktvisen listen befinner seg inni beskrivelseslisten.

`\item{}` Vi ser at `\LaTeX{}` selv klarer å holde orden på innrykk og markering av punktene i den enkelte listen.

`\item{}` Det er mulig å bruke listene utenpå hverandre i en dybde på inntil fire nivåer.

`\end{itemize}`

`\end{description}`

`\subsubsection{Andre omgivelser}\label{Andre-omgivelser}`

Det finnes også andre nyttige former for omgivelser:

`\begin{description}`

`\item[center]` gir sentrert tekst:

`\begin{center}`

Denne teksten skulle komme sentrert\\

på siden;\\

hvis vi ønsker linjedeling på bestemte steder,\\

kan vi angi dette med `\verb@\\@`.

`\end{center}`

- Listen innledes med at det står `\begin{itemize}` og avsluttes med `\end{itemize}`.
Vanligvis står det ingen blank linje etter `\end{itemize}` (eller tilsvarende `\end` som avslutter andre typer lister) fordi vi ikke ønsker innrykk av den etterfølgende linjen.
- Hvert nytt punkt i listen innledes med et kall på kommandoen `\item{}`.
- Vi kan se at innholdet i listen er rykket litt inn i forhold til resten av teksten, og at hvert punkt angis med en svart prikk.

En annen form for liste kalles *enumerate*, og den inneholder fortløpende nummerering av punktene:

1. Første punktet er alltid nr. 1.
2. \LaTeX vil automatisk sørge for å nummerere punktene riktig.

Den siste formen for liste kalles *description*, og den lager noe som minner om et leksikon der hvert punkt starter med et navn i fet skrift:

Tittel Hvert punkt starter med et ord, for eksempel navnet på noe som skal beskrives.

Notasjon Innledningsordet angis som opsjon til `\item`.

Flere nivåer Noe av det fine med lister er at de uten videre kan brukes i flere nivåer; man kan altså ha lister inni lister.

- Denne punktvisen listen befinner seg inni beskrivelseslisten.
- Vi ser at \LaTeX selv klarer å holde orden på innrykk og markering av punktene i den enkelte listen.
- Det er mulig å bruke listene utenpå hverandre i en dybde på inntil fire nivåer.

2.4.2 Andre omgivelser

Det finnes også andre nyttige former for omgivelser:

center gir sentrert tekst:

Denne teksten skulle komme sentrert
på siden;
hvis vi ønsker linjedeling på bestemte steder,
kan vi angi dette med `\`.

```

\item[flushleft] gir venstrejustert tekst:
  \begin{flushleft}
    Denne teksten utmerker seg ved at den er «presset» mot den venstre margen;
    den har altså ikke rette marger slik det er vanlig.
  \end{flushleft}

\item[flushright] gir høyrejustert tekst:
  \begin{flushright}
    Denne teksten er i motsetning til den forrige «presset» mot høyre
    marg. Dette er også av og til nyttig; se for eksempel på sitatet
    nedenfor.
  \end{flushright}

\item[quote] egner seg for sitater:
  \begin{quote}
    She looked as if she had been poured into her clothes and had
    forgotten to say ‘‘when.’’ % ‘‘...’’ gir korrekte engelske anførselstegn.
  \begin{flushright}
    \textit{--- P. G. Wodehouse} % --- gir en tankestrek.
  \end{flushright}
  \end{quote}

\item[verse] passer til dikt; linjene må skilles med \verb@\@:
  \begin{verse}
    There was a young curate of Kew,\\
    Who kept a tom cat in a pew;\\
    He taught it to speak\\ \quad Alphabetical Greek,\\
    But it never got farther than $\mu\upsilon$psilon$.
  \end{verse}

\item[verbatim] gjør det samme som \verb@\verb@-kommandoen, altså slår av
  den vanlige tekstbehandlingen. Den egner seg fint hvis teksten går over
  flere linjer:
  \begin{verbatim}
Dette er tekst hvor vanlig redigering er slått av.
«Verbatim» egner seg fint der hvor man
for eksempel skal vise programkode.
  \end{verbatim}
\end{description}

```

\section{Dokumentets struktur}\label{Struktur}

Alle dokumenter har struktur, selv om denne strukturen er tydeligere i noen former for dokumenter enn i andre. Bokstaver settes sammen til ord, ord til linjer, linjer til tekstavsnitt (som dette), tekstavsnitt til underavsnitt, og underavsnitt til avsnitt og hovedavsnitt. I større dokumenter vil man kanskje også ha kapitler og hoveddeler. Innholdsfortegnelsen sist i dette dokumentet viser klart denne strukturen.

Det er mye lettere å lese et dokument hvor strukturen er klar, så det benyttes diverse typografiske hjelpemidler for å gjøre denne lettfattelig for leseren. De viktigste av disse hjelpemidlene er skriftstørrelse, ekstra avstand, \textbf{fete typer} og nummerering. Ved å titte litt i dette dokumentet er det lett å se at alle disse er benyttet her.

flushleft gir venstrejustert tekst:

Denne teksten utmerker seg ved at den er «presset» mot den venstre margin; den har altså ikke rette marger slik det er vanlig.

flushright gir høyrejustert tekst:

Denne teksten er i motsetning til den forrige «presset» mot høyre marg. Dette er også av og til nyttig; se for eksempel på sitatet nedenfor.

quote egner seg for sitater:

She looked as if she had been poured into her clothes and had forgotten to say “when.”

— *P. G. Wodehouse*

verse passer til dikt; linjene må skilles med `\\`:

There was a young curate of Kew,
Who kept a tom cat in a pew;
He taught it to speak
Alphabetical Greek,
But it never got farther than $\mu\nu$.

verbatim gjør det samme som `\verb`-kommandoen, altså slår av den vanlige tekstbehandlingen. Den egner seg fint hvis teksten går over flere linjer:

Dette er tekst hvor vanlig redigering er slått av.
«Verbatim» egner seg fint der hvor man
for eksempel skal vise programkode.

3 Dokumentets struktur

Alle dokumenter har struktur, selv om denne strukturen er tydeligere i noen former for dokumenter enn i andre. Bokstaver settes sammen til ord, ord til linjer, linjer til tekstavsnitt (som dette), tekstavsnitt til underavsnitt, og underavsnitt til avsnitt og hovedavsnitt. I større dokumenter vil man kanskje også ha kapitler og hoveddeler. Innholdsfortegnelsen sist i dette dokumentet viser klart denne strukturen.

Det er mye lettere å lese et dokument hvor strukturen er klar, så det benyttes diverse typografiske hjelpemidler for å gjøre denne lettfattelig

```

\begin{quote}
  \emph{En av hovedtankene bak \LaTeX{}} er at brukeren skal kunne arbeide
  direkte med dokumentets struktur uten å måtte bry seg med hvilke visuelle
  hjelpemidler som benyttes til å vise denne strukturen. Dette oppnås ved å
  bruke kommandoer som \texttt{\textbackslash}section} og
  \texttt{\textbackslash}subsection} og la den aktuelle dokumentstilen (se
  nedenfor) beskrive selve formateringen.}
\end{quote}
%% Her bruker jeg kommandoen \textbackslash{} for å få frem tegnet '\'.
%% Grunnen til dette er at det ikke er lov å ha \verb i en
%% parameterliste. Det er noen få slike sære regler i LaTeX...

```

```

\subsection{Hovedstrukturen}
Alle \LaTeX{-dokumenter har en hovedstruktur som er slik:
\begin{verbatim}
\documentclass[«opsjoner»]{«dokumentklasse»}
«pakker»
\begin{document}
«Teksten i dokumentet»
\end{document}
\end{verbatim}
Parameteren til \verb@\documentclass@ forteller hva slags stil som skal
benyttes i dokumentet; denne kommandoen kan også ha opsjoner (se nedenfor).

```

```

\subsection{Dokumentklasser}
\LaTeX{} kan benyttes til å lage diverse ulike typer dokumenter, og tanken er
at det skal finnes en eller flere dokumentklasser til hvert bruksområde.
De viktigste dokumentklassene er vist i tabell\href{stiler}; de som står over
streken er standard i \LaTeX{} mens de nedenfor er spesielt laget ved Institutt
for informatikk.
Dokumentet du leser nå er laget med klassen \texttt{article} (se parameteren
til \verb@\documentclass@ i aller første linje av filen).

```

```

\begin{table}
  \begin{center}
    \begin{tabular}{|l|l|} \hline
      \textbf{Dokumentklasse}& \textbf{Passer for \dots{}}\ \hline
      \texttt{article}& kortere artikler\ \hline
      \texttt{report}& rapporter og masteroppgaver\ \hline
      \texttt{book}& bøker\ \hline
      \texttt{slides}& lysark\ \hline
      \texttt{letter}& brev\ \hline
      \texttt{uiosloforside}& offisielle forsider\ \hline
      \texttt{privatbrev}& private brev\ \hline
      \texttt{uioexam}& eksamensoppgaver\ \hline
      \texttt{psfoils}& lysark\ \hline
    \end{tabular}
    \caption{Oversikt over dokumentklasser}
    \label{stiler}
  \end{center}
\end{table}

```

Dokumentklasse	Passer for ...
article	kortere artikler
report	rapporter og masteroppgaver
book	bøker
slides	lysark
letter	brev
uiosloforside	offisielle forsider
privatbrev	private brev
uioexam	eksamensoppgaver
psfoils	lysark

Tabell 1: Oversikt over dokumentklasser

for leseren. De viktigste av disse hjelpemidlene er skriftstørrelse, ekstra avstand, **fete typer** og nummerering. Ved å tittle litt i dette dokumentet er det lett å se at alle disse er benyttet her.

En av hovedtankene bak L^AT_EX er at brukeren skal kunne arbeide direkte med dokumentets struktur uten å måtte bry seg med hvilke visuelle hjelpemidler som benyttes til å vise denne strukturen. Dette oppnås ved å bruke kommandoer som `\section` og `\subsection` og la den aktuelle dokumentstilen (se nedenfor) beskrive selve formateringen.

3.1 Hovedstrukturen

Alle L^AT_EX-dokumenter har en hovedstruktur som er slik:

```
\documentclass[«opsjoner»]{«dokumentklasse»}
«pakker»
\begin{document}
«Teksten i dokumentet»
\end{document}
```

Parameteren til `\documentclass` forteller hva slags stil som skal benyttes i dokumentet; denne kommandoen kan også ha opsjoner (se nedenfor).

3.2 Dokumentklasser

L^AT_EX kan benyttes til å lage diverse ulike typer dokumenter, og tanken er at det skal finnes en eller flere dokumentklasser til hvert bruksområde. De viktigste dokumentklassene er vist i tabell 1; de som står over streken er standard i L^AT_EX mens de nedenfor er spesielt laget ved Institutt for informatikk. Dokumentet du leser nå er laget med klassen `article` (se parameteren til `\documentclass` i aller første linje av filen).

```

\subsubsection{Opsjoner til dokumentklassen}
Dokumentklassen definerer som sagt dokumentets utseende, men vi ønsker ofte
å angi små modifikasjoner. Dette oppnår vi ved å bruke opsjoner til
\verb@\documentclass@; i dette dokumentet benyttes de tre opsjonene
\texttt{11pt}, \texttt{norsk} og \texttt{a4paper}. Disse opsjonene har
følgende effekt:
\begin{description}
\item[\texttt{11pt}] angir at det skal benyttes en skriftstørrelse på
11~punkter for den vanlige teksten i dokumentet. En annen aktuell opsjon
er \texttt{12pt} for 12~punktters skrift
({\large % Kommandoen \large benyttes for å gi litt større skrift.
% Det finnes også \Large, \LARGE, \huge og \Huge.
som er så stor}).

(Punkter er den vanligste typografiske måleenheten; 1~punkt er omtrent
 $\frac{1}{3}$ ~mm.) %%  $\$...\$$  angir en formel; mer om dette siden.

Hvis man ikke angir noen størrelse, får man 10~punkter
({\small % Kommandoen \small gir litt mindre skrift.
% Det finnes også \footnotesize, \scriptsize og \tiny.
som blir noe mindre, nemlig så liten}).

\item[\texttt{norsk}] angir at dokumentet er på norsk. Se beskrivelsen av
pakken \texttt{babel} nedenfor for mer informasjon.

\item[\texttt{a4paper}] angir at utskriften skal tilpasses A4-ark (som er
det vanlige i Norge).
\end{description}
Andre nyttige opsjoner som ikke er benyttet her, er disse:
\begin{description}
\item[\texttt{twocolumn}] gjør at dokumentet settes i to spalter.

\item[\texttt{twoside}] gir et oppsett som passer for tosidig trykk. Her vil
høyre og venstre oppslag bli litt forskjellige.
\end{description}

\subsection{Forfatter og tittel}
Dokumentets forfatter og tittel angis henholdsvis med kommandoene
\verb@\author@ og \verb@\title@; om man ønsker det, kan man angi linjedeling
med \verb@\\@. Du kan se hvorledes dette er skrevet helt i begynnelsen av dette
dokumentet.

I tråd med vanlig \LaTeX{-filosofi er \verb@\author@ og \verb@\title@ kun
\emph{spesifikasjoner}; de produserer ingen utskrift men gir \LaTeX{
data. Kommandoen \verb@\maketitle@ derimot skriver ut den aktuelle
informasjonen på en passende form.

\subsection{Pakker}\label{pakker}
De innebygde mekanismene i \LaTeX{ kan utvides ved å hente inn diverse pakker.
Det finnes pakker for svært mange ulike formål; i dette dokumentet er det
benyttet syv stykker:
\begin{description}
\item[\texttt{babel}] styrer tilpasningen til det språk dokumentet er skrevet
i; dette influerer på datoer, diverse faste tekster (som \emph{tabell} og
\emph{figur}) samt ordellingen. De vanligste språkene her ved UiO er
\texttt{norsk}, \texttt{nynorsk}, \texttt{american} (amerikansk engelsk)
og \texttt{english} (britisk engelsk). Hvis man ikke benytter \texttt{babel}
i det hele tatt, får man amerikansk engelsk.

```


3.2.1 Opsjoner til dokumentklassen

Dokumentklassen definerer som sagt dokumentets utseende, men vi ønsker ofte å angi små modifikasjoner. Dette oppnår vi ved å bruke opsjoner til `\documentclass`; i dette dokumentet benyttes de tre opsjonene `11pt`, `norsk` og `a4paper`. Disse opsjonene har følgende effekt:

`11pt` angir at det skal benyttes en skriftstørrelse på 11 punkter for den vanlige teksten i dokumentet. En annen aktuell opsjon er `12pt` for 12 punkters skrift (som er så stor).

(Punkter er den vanligste typografiske måleenheten; 1 punkt er omtrent $\frac{1}{3}$ mm.)

Hvis man ikke angir noen størrelse, får man 10 punkter (som blir noe mindre, nemlig så liten).

`norsk` angir at dokumentet er på norsk. Se beskrivelsen av pakken `babel` nedenfor for mer informasjon.

`a4paper` angir at utskriften skal tilpasses A4-ark (som er det vanlige i Norge).

Andre nyttige opsjoner som ikke er benyttet her, er disse:

`twocolumn` gjør at dokumentet settes i to spalter.

`twoside` gir et oppsett som passer for tosidig trykk. Her vil høyre og venstre oppslag bli litt forskjellige.

3.3 Forfatter og tittel

Dokumentets forfatter og tittel angis henholdsvis med kommandoene `\author` og `\title`; om man ønsker det, kan man angi linjedeling med `\\`. Du kan se hvorledes dette er skrevet helt i begynnelsen av dette dokumentet.

I tråd med vanlig L^AT_EX-filosofi er `\author` og `\title` kun *spesifikasjoner*; de produserer ingen utskrift men gir L^AT_EX data. Kommandoen `\maketitle` derimot skriver ut den aktuelle informasjonen på en passende form.

3.4 Pakker

De innebygde mekanismene i L^AT_EX kan utvides ved å hente inn diverse pakker. Det finnes pakker for svært mange ulike formål; i dette dokumentet er det benyttet syv stykker:

`babel` styrer tilpasningen til det språk dokumentet er skrevet i; dette influerer på datoer, diverse faste tekster (som *tabell* og *figur*) samt

`\item[\texttt{fontenc}]` angir hvilken koding som skal benyttes for font-filene hvor typesnittene er lagret. Vi bør alltid bruke `\texttt{T1}` som gir oss de nyeste 8-bits fontene.

`\item[\texttt{graphicx}]` definerer kommandoen `\verb@\includegraphics@` som brukes til å legge inn PostScript-illustrasjoner i dokumentet; dette omtales nærmere i avsnitt [\vref{Figurer}](#).

(Det litt merkelige navnet `\texttt{graphicx}` kommer av at pakken er en utvidelse («extension») av `\texttt{graphics}`-pakken.)

`\item[\texttt{inputenc}]` angir hvilken koding vi benytter i vår `\LaTeX{}`-fil. Det vanlige for oss er `\texttt{latin1}` som angir at vi bruker `\textsc{iso}~8859-1` (ofte kalt `\textsc{iso} Latin-1`).

`\item[\texttt{mathppl}]` angir at dokumentet skal settes med typesnittet `\emph{Palatino}`; se forøvrig avsnitt [\vref{typer-i-dok}](#).

`\item[\texttt{textcomp}]` gir tilgang til flere nyttige symboler; den bør alltid være med.

`\item[\texttt{varioref}]` definerer kommandoen `\verb@\vref@` som benyttes til kryssreferanser; se avsnitt [\vref{X-ref}](#).

`\end{description}`

Som vist i de første linjene i dette dokumentet, kan også pakker ha opsjoner. Dessuten gjelder at alle opsjonene til `\verb@\documentclass@` vil være opsjoner til alle pakkene; derfor er `\texttt{norsk}` også automatisk opsjon til `\texttt{babel}` og `\texttt{varioref}` i dette dokumentet.

Andre aktuelle pakker er følgende:

`\begin{description}`

`\item[\texttt{color}]` henter inn kommandoer for å bruke farger i dokumentet.

`\item[\texttt{fancyhdr}]` benyttes til å angi toppteksten og bunnteksten (med sidenummer og annen informasjon) i et dokument.

`\item[\texttt{fancyvrb}]` gjør det enkelt å inkludere uformattert tekst som programkode.

`\item[\texttt{rotating}]` henter inn kommandoer for å rotere tekst, tabeller og figurer.

`\end{description}`

Les boken til Leslie Lamport og `\emph{Lokal guide til \LaTeX{}}` for mer informasjon om disse pakkene.

`\subsection{Et lite eksempel}`

I figur [\vref{fig:mini-dok}](#) er vist et helt minimalt `\LaTeX{}`-dokument slik det bør se ut når man skriver dokumenter på norsk. Alle de angitte pakkene og opsjonene bør være med! (Det kan virke strevsomt alltid å huske på dette, men man er da sikret at dokumentet vil kunne kompileres likt uansett hvor det skal være i verden.)

orddelingen. De vanligste språkene her ved UiO er norsk, nynorsk, american (amerikansk engelsk) og english (britisk engelsk). Hvis man ikke benytter babel i det hele tatt, får man amerikansk engelsk.

fontenc angir hvilken koding som skal benyttes for font-filene hvor typesnittene er lagret. Vi bør alltid bruke T1 som gir oss de nyeste 8-bits fontene.

graphicx definerer kommandoen `\includegraphics` som brukes til å legge inn PostScript-illustrasjoner i dokumentet; dette omtales nærmere i avsnitt 6.1 på side 14.

(Det litt merkelige navnet `graphicx` kommer av at pakken er en utvidelse («`extension`») av `graphics`-pakken.)

inputenc angir hvilken koding vi benytter i vår \LaTeX -fil. Det vanlige for oss er `latin1` som angir at vi bruker ISO 8859-1 (ofte kalt ISO Latin-1).

mathpple angir at dokumentet skal settes med typesnittet *Palatino*; se forøvrig avsnitt 4.1 på side 11.

textcomp gir tilgang til flere nyttige symboler; den bør alltid være med.

varioref definerer kommandoen `\vref` som benyttes til kryssreferanser; se avsnitt 3.9 på side 10.

Som vist i de første linjene i dette dokumentet, kan også pakker ha opsjoner. Dessuten gjelder at alle opsjonene til `\documentclass` vil være opsjoner til alle pakkene; derfor er norsk også automatisk opsjon til `babel` og `varioref` i dette dokumentet.

Andre aktuelle pakker er følgende:

`color` henter inn kommandoer for å bruke farger i dokumentet.

`fancyhdr` benyttes til å angi topp teksten og bunnteksten (med sidenummer og annen informasjon) i et dokument.

`fancyvrb` gjør det enkelt å inkludere uformattert tekst som programkode.

`rotating` henter inn kommandoer for å rotere tekst, tabeller og figurer.

Les boken til Leslie Lamport og *Lokal guide til \LaTeX* for mer informasjon om disse pakkene.

3.5 Et lite eksempel

I figur 1 på neste side er vist et helt minimalt \LaTeX -dokument slik det bør se ut når man skriver dokumenter på norsk. Alle de angitte pakkene og opsjonene bør være med! (Det kan virke strevsomt alltid å huske på dette, men man er da sikret at dokumentet vil kunne kompileres likt uansett hvor det skal være i verden.)

```

\begin{figure}
  \begin{verbatim}
\documentclass[a4paper,norsk]{article}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{babel,textcomp}

\begin{document}
Hei, alle sammen!
\end{document}
  \end{verbatim}
  \caption{Et minimalt \LaTeX{}-dokument}\label{fig:mini-dok}
\end{figure}

```

```

\subsection{Strukturkommandoene}\label{Strukt-komm}
Teksten i dokumentet kan deles opp i hoveddeler, kapitler, avsnitt,
underavsnitt osv; dette er for eksempel underavsnitt\vref{Strukt-komm} med
tittelen \textbf{Strukturkommandoene}, og det h rer hjemme i
avsnitt\vref{Struktur} med tittelen \textbf{Dokumentets struktur}.

```

Felles for alle strukturkommandoene er at de har en parameter som angir tittelen; dette underavsnittet starter for eksempel med kommandoen `\verb@\subsection{Strukturkommandoene}@`.

```

\subsubsection{Andre strukturkommandoer}
N  har vi startet et «underunderavsnitt» ved   benytte kommandoen
\verb@\subsubsection@, men bunnen er ikke n dd enn .

```

```

\paragraph{Kommando for et enda lavere niv }
Ved   benytte en kommando som heter \verb@\paragraph@ kan vi starte et
«underunderunderavsnitt».

```

```

\subsection{Innholdsfortegnelse}
\LaTeX{} er i stand til selv   lage en innholdsfortegnelse for dokumentet;
det eneste brukeren beh ver   gj re er   legge inn et kall p  kommandoen
\verb@\tableofcontents@ der han eller hun  nsker at innholdsfortegnelsen
skal st . I dette dokumentet st r den til sist.

```

```

\subsection{Fotnoter}
Mange artikler inneholder fotnoter og det er meget enkelt   lage slike i
\LaTeX{}: man bare benytter kommandoen \verb@\footnote@ og gir fotnotens tekst
som parameter.\footnote{Husk at kommandoen \texttt{\textbackslash}verb} og
omgivelsen \texttt{verbatim} \emph{ikke} er lov i parametre og dermed heller
ikke i fotnoter.} Plasseringen p  siden\footnote{Alle fotnoter plasseres
nederst p  den siden hvor referansen er, om mulig. Det kan allikevel av og
til hende at fotnoten \emph{m } deles eller flyttes til neste side.} og
nummereringen skjer automatisk, likeledes skifte av skriftst rrelse.

```

```

\documentclass[a4paper,norsk]{article}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{babel,textcomp}

\begin{document}
Hei, alle sammen!
\end{document}

```

Figur 1: Et minimalt L^AT_EX-dokument

3.6 Strukturkommandoene

Teksten i dokumentet kan deles opp i hoveddeler, kapitler, avsnitt, underavsnitt osv; dette er for eksempel underavsnitt 3.6 med tittelen **Strukturkommandoene**, og det hører hjemme i avsnitt 3 på side 5 med tittelen **Dokumentets struktur**.

Felles for alle strukturkommandoene er at de har en parameter som angir tittelen; dette underavsnittet starter for eksempel med kommandoen `\subsection{Strukturkommandoene}`.

3.6.1 Andre strukturkommandoer

Nå har vi startet et «underunderavsnitt» ved å benytte kommandoen `\subsubsection`, men bunnen er ikke nådd ennå.

Kommando for et enda lavere nivå Ved å benytte en kommando som heter `\paragraph` kan vi starte et «underunderunderavsnitt».

3.7 Innholdsfortegnelse

L^AT_EX er i stand til selv å lage en innholdsfortegnelse for dokumentet; det eneste brukeren behøver å gjøre er å legge inn et kall på kommandoen `\tableofcontents` der han eller hun ønsker at innholdsfortegnelsen skal stå. I dette dokumentet står den til sist.

3.8 Fotnoter

Mange artikler inneholder fotnoter og det er meget enkelt å lage slike i L^AT_EX: man bare benytter kommandoen `\footnote` og gir fotnotens

```
\subsection{Kryssreferanser}\label{X-ref}
```

Ofte har vi behov for å referere til andre avsnitt i dokumentet; hvis vi snakker om typesnitt, kan vi for eksempel henviser til avsnitt\ref{typesnitt}. For å sikre at disse referansene alltid er korrekte, selv om vi skulle finne på å legge inn nye avsnitt slik at nummereringen endres, er det tryggest å benytte kommandoene som finnes i `\LaTeX{}` for dette.

Mekanismen for kryssreferanser er meget enkel; det er bare å legge en etikett i form av et kall på kommandoen `\verb@label@` de stedene vi ønsker å referere til, slik det for eksempel er gjort rett etter kallet på `\verb@\subsection{Kryssreferanser}@` før forrige avsnitt. Parameteren er et navn vi selv finner på, men det er naturlig å velge noe som er lett å forbinde med stedet det står. Selve kallet på `\verb@label@` gir ingen synlig effekt.

For å referere til en etikett benyttes kommandoen `\verb@\vref@` med etikettens navn som parameter, og dette gir da den riktige referansen:\ref{X-ref}. Hvis referansen og etiketten ikke står på samme side, angis det også hvilken siden det refereres til.\footnote{Legg forøvrig merke til at det `\emph{ikke}` skal stå noen blank foran kallet på `\texttt{\textbackslash\vref}.`}

Etiketter for kryssreferanser kan plasseres nesten overalt i dokumentet: etter strukturkommandoer (som er det mest vanlige), i lister, i fotnoter osv. Det er mulig å referere både forover og bakover i dokumentet.

Av og til kan man etter `\LaTeX{}`-kjøringen oppleve å få se meldingen

```
\begin{verbatim}
```

```
LaTeX Warning: Label(s) may have changed.
```

```
Rerun to get cross-references right.
```

```
\end{verbatim}
```

Dette betyr at `\LaTeX{}` har oppdaget at kryssreferansene ikke er helt

```
\{a} jour, men at neste kjøring vil rette på dette.
```

```
\section{Typesnitt}\label{typesnitt}
```

Nå man skriver tekst, kommer den stort sett i samme typesnitt gjennom hele dokumentet med mindre man ber om noe annet. Kun i noen få tilfelle endres typesnittet automatisk, for eksempel i avsnittsoverskrifter der den kommer med `\textbf{fete typer}`, eller i eksempler laget med `\verb@\verb@`-kommandoen der det benyttes `\texttt{skrivemaskinskrift}`.

tekst som parameter.¹ Plasseringen på siden² og nummereringen skjer automatisk, likeledes skifte av skriftstørrelse.

3.9 Kryssreferanser

Ofte har vi behov for å referere til andre avsnitt i dokumentet; hvis vi snakker om typesnitt, kan vi for eksempel henvise til avsnitt 4. For å sikre at disse referansene alltid er korrekte, selv om vi skulle finne på å legge inn nye avsnitt slik at nummereringen endres, er det tryggest å benytte kommandoene som finnes i L^AT_EX for dette.

Mekanismen for kryssreferanser er meget enkel; det er bare å legge en etikett i form av et kall på kommandoen `\label` de stedene vi ønsker å referere til, slik det for eksempel er gjort rett etter kallet på `\subsection{Kryssreferanser}` før forrige avsnitt. Parameteren er et navn vi selv finner på, men det er naturlig å velge noe som er lett å forbinde med stedet det står. Selve kallet på `\label` gir ingen synlig effekt.

For å referere til en etikett benyttes kommandoen `\vref` med etikettens navn som parameter, og dette gir da den riktige referansen: 3.9. Hvis referansen og etiketten ikke står på samme side, angis det også hvilken siden det refereres til.³

Etiketter for kryssreferanser kan plasseres nesten overalt i dokumentet: etter strukturkommandoer (som er det mest vanlige), i lister, i fotnoter osv. Det er mulig å referere både forover og bakover i dokumentet.

Av og til kan man etter L^AT_EX-kjøringen oppleve å få se meldingen

```
LaTeX Warning: Label(s) may have changed.  
Rerun to get cross-references right.
```

Dette betyr at L^AT_EX har oppdaget at kryssreferansene ikke er helt à jour, men at neste kjøring vil rette på dette.

4 Typesnitt

Nå man skriver tekst, kommer den stort sett i samme typesnitt gjennom hele dokumentet med mindre man ber om noe annet. Kun i noen få tilfelle endres typesnittet automatisk, for eksempel i avsnittsoverskrifter der den kommer med **fete typer**, eller i eksempler laget med `\verb`-kommandoen der det benyttes skrivemaskinskrift.

¹Husk at kommandoen `\verb` og omgivelsen `verbatim` *ikke* er lov i parametre og dermed heller ikke i fotnoter.

²Alle fotnoter plasseres nederst på den siden hvor referansen er, om mulig. Det kan allikevel av og til hende at fotnoten *må* deles eller flyttes til neste side.

³Legg forøvrig merke til at det *ikke* skal stå noen blank foran kallet på `\vref`.

I `\LaTeX{}` kan brukeren skifte typesnittet når han eller hun ønsker det. Dette gjøres med enkle kommandoer, for eksempel `\verb:\textbf{ord}`: som vil sette `\textbf{ord}` med fete typer.

De vanligste kommandoene for endring av typesnittet er `\verb:\textbf:` som vil gi `\textbf{fete typer}`, `\verb:\textit:` som vil gi `\textit{kursiv}`, `\verb:\emph:` som vil gi `\emph{uthevet skrift}` (vanligvis kursiv), `\verb:\textsc:` som gir «små store bokstaver» (såkalte `\textsc{kapiteler}`), `\verb:\textsf:` som gir et såkalt `\textsf{grotesksnitt}` og `\verb:\texttt:` som vil gi `\texttt{skrivemaskinskrift}`. Det er også mulig å kombinere disse kommandoene og få for eksempel `\textbf{\textit{fet kursiv}}`.

```
\subsection{Skifte typesnitt for hele dokumentet}\label{typer-i-dok}
{\renewcommand{\rmdefault}{cmr} % Ved å omdefinere \rmdefault og \ttdefault
 \renewcommand{\ttdefault}{cmtt} % kan man skifte standard typesnitt for
 % vanlig skrift og skrivemaskinskrift.
 \rm % Kommandoen \rm velger standard typesnitt.
I \LaTeX{ er det også mulig å skifte typesnitt for hele dokumentet ved å
hente inn en passende pakke. I dette dokumentet (med unntak av dette
avsnittet) benyttes Palatino (angitt av pakken \texttt{mathppl}); andre
muligheter er
{\fontfamily{pbk}\selectfont Bookman} (pakken \texttt{bookman}),
{\fontfamily{phv}\selectfont Helvetica} (pakken \texttt{helvet}),
{\fontfamily{pnc}\selectfont New Century Schoolbook} (pakken \texttt{newcent}),
{\fontfamily{ptm}\selectfont Times} (pakken \texttt{times})
%% Kommandoene \fontfamily og \selectfont brukes til å skifte typesnitt
%% lokalt; les mer om disse i «Lokal guide til LaTeX».
og mange andre. Hvis man ikke benytter noen
skriftopsjon i det hele tatt, får man standardsnittet \emph{Computer Modern}
som er benyttet i dette avsnittet. Det er tydelig å se at det er et mye
spinklere typesnitt enn Palatino. Enhver får velge hvilket snitt de vil bruke;
slikt er avhengig av personlig smak og hva slags dokument det er snakk om.)

%% Da er vi tilbake til vanlig typesnitt igjen; {}-ene rundt redefinisjonen
%% av \rmdefault og \ttdefault gjorde at endringene ble lokale.
```

```
\section{Matematiske formler}
\LaTeX{ er spesielt sterk når det gjelder å sette matematiske formler.
Foruten at det finnes et meget stort utvalg symboler, har \LaTeX{
innebygget mye kunnskap om hvorledes matematiske formler bør se ut.
```

Dokumenter skrevet i `\LaTeX{}` kan inneholde formler i to varianter, enten `\emph{små formler}` som $\sum_{k=1}^{\infty} 2^{-k}$ som står sammen med den vanlige teksten, eller `\emph{store formler}` som

```
\[
 \sum_{k=1}^{\infty} 2^{-k}
\]
```

som kommer på en linje for seg. Kommandoene for å uttrykke formlene er de samme i begge tilfellene, men `\LaTeX{}` vil formatere dem ulikt. Se for eksempel på plasseringen av summegrensene og hvor høyt indeksen $-k$ er plassert.

```
\subsection{Omgivelser for matematiske formler}
I \LaTeX{ skrives matematiske formler i en egen omgivelse hvor det gjelder
spesielle regler. For små formler angis omgivelsen med \verb@$...$@, mens
\verb@[...@ benyttes til å merke store formler. Man kan alternativt bruke
omgivelsen \texttt{equation} som vil gi en stor formel med et eget nummer:
```


I L^AT_EX kan brukeren skifte typesnittet når han eller hun ønsker det. Dette gjøres med enkle kommandoer, for eksempel `\textbf{ord}` som vil sette **ord** med fete typer.

De vanligste kommandoene for endring av typesnittet er `\textbf` som vil gi **fete typer**, `\textit` som vil gi *kursiv*, `\emph` som vil gi *uthevet skrift* (vanligvis kursiv), `\textsc` som gir «små store bokstaver» (såkalte KAPITÉLER), `\textsf` som gir et såkalt grotesksnitt og `\texttt` som vil gi skrivemaskinskrift. Det er også mulig å kombinere disse kommandoene og få for eksempel *fet kursiv*.

4.1 Skifte typesnitt for hele dokumentet

I L^AT_EX er det også mulig å skifte typesnitt for hele dokumentet ved å hente inn en passende pakke. I dette dokumentet (med unntak av dette avsnittet) benyttes Palatino (angitt av pakken `mathppl`); andre muligheter er **Bookman** (pakken `bookman`), **Helvetica** (pakken `helvet`), **New Century Schoolbook** (pakken `newcent`), **Times** (pakken `times`) og mange andre. Hvis man ikke benytter noen skriftopsjon i det hele tatt, får man standardsnittet *Computer Modern* som er benyttet i dette avsnittet. Det er tydelig å se at det er et mye spinklere typesnitt enn Palatino. Enhver får velge hvilket snitt de vil bruke; slikt er avhengig av personlig smak og hva slags dokument det er snakk om.

5 Matematiske formler

L^AT_EX er spesielt sterk når det gjelder å sette matematiske formler. Foruten at det finnes et meget stort utvalg symboler, har L^AT_EX innebygget mye kunnskap om hvorledes matematiske formler bør se ut.

Dokumenter skrevet i L^AT_EX kan inneholde formler i to varianter, enten *små formler* som $\sum_{k=1}^{\infty} 2^{-k}$ som står sammen med den vanlige teksten, eller *store formler* som

$$\sum_{k=1}^{\infty} 2^{-k}$$

som kommer på en linje for seg. Kommandoene for å uttrykke formlene er de samme i begge tilfellene, men L^AT_EX vil formatere dem ulikt. Se for eksempel på plasseringen av summegrensene og hvor høyt indeksen $-k$ er plassert.

5.1 Omgivelser for matematiske formler

I L^AT_EX skrives matematiske formler i en egen omgivelse hvor det gjelder spesielle regler. For små formler angis omgivelsen med `$. . . $`, mens

```

\begin{equation}
  \frac{-b \pm \sqrt{b^2-4ac}}{2a}
\end{equation}
Typesetting av matematiske formler følger som sagt spesielle regler:
\begin{itemize}
\item{} Kun få av de vanlige \LaTeX{}-kommandoene fungerer; i stedet finnes et
  stort utvalgte egne kommandoer.

\item{} I matematiske formler \emph{ignorerer alle blanke}. Det er således
  det samme om man skriver  $1+2+\cdots+n$  eller  $1 + 2 + \cdots + n$ ;
  det blir ingen forskjell i resultatet.

\item{} Siden blanke ignoreres, kan man sløyfe klammene etter kommandoer uten
  parametre (men man må sette en blank bak hvis det følger en bokstav etter).
\end{itemize}

\subsection{Matematiske symboler}
I matematiske formler inngår diverse symboler; her er en liten liste over de
viktigste av dem. Det finnes ganske mange flere, så studer boken til Leslie
Lamport og \emph{Lokal guide til \LaTeX{}} hvis det er noen du savner.
\begin{itemize}
\item{} Bokstaver og tall skrives som normalt:  $e$ ,  $x$ ,  $122$ ,
   $2\{,71828$ . % Fordi komma vanligvis benyttes som skilletegn i formler,
  % f.eks.  $f(a,b,c)$ , må vi ha klammer rundt desimalkomma.
  % Dette influerer på hvor mye luft det blir på hver side.

\item{} Bokstavmarkeringer angis med kommandoer:  $\bar{a}$ ,  $\vec{a}$ ,  $\dot{a}$ ,
   $\tilde{a}$ .

\item{} Fete bokstaver får man med kommandoen \verb@mathbf@:
   $a \not\sim \mathbf{a}$ .

\item{} «Skjønnskriftbokstaver» (kun  $\mathcal{A}$ – $\mathcal{Z}$ ) er
  %% -- gir en intervallstrek som er litt lengre enn en vindestrek.
  tilgjengelig gjennom kommandoen \verb@mathcal@.

\item{} Greske bokstaver angis med kommandoer:  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,
   $\ldots$ ,  $\omega$ . Dette gjelder også for de store bokstavene  $A$ ,  $B$ ,  $\Gamma$ ,
   $\Delta$ ,  $\ldots$ ,  $\Omega$ . (Vi bruker latinske bokstaver der de er de samme
  som de greske.)

\item{} De fleste vanlige matematiske tegn som finnes på tastaturet, kan
  brukes direkte:  $+$ ,  $-$ ,  $/$ ,  $<$ ,  $>$ ,  $=$ ,  $'$ .

\item{} De andre matematiske symbolene må derimot angis med kommandoer:
   $\times$ ,  $\cdot$ ,  $\leq$ ,  $\geq$ ,  $\neq$ ,  $\land$ ,  $\lor$ ,  $\in$ ,  $\infty$ ,  $\forall$ ,
   $\cdots$ ,  $\exists$ ,  $\ldots$ ,  $\equiv$ ,  $\sim$ .

\item{} Det finnes også diverse piler:  $\leftarrow$ ,  $\rightarrow$ ,
   $\leftrightarrow$ ,  $\Leftarrow$ ,  $\Rightarrow$ ,  $\longleftarrow$ ,  $\longrightarrow$ ,
   $\uparrow$ , og mange flere.

\item{} For de fleste symboler kan man bruke kommandoen \verb@not@ foran for
  å få det negerte symbolet:  $\not<$ ,  $\not\rightarrow$ ,  $\notin$ .
  % Det finnes en egen \notin fordi \not\in ikke blir helt pen.

```

`\[...]` benyttes til å merke store formler. Man kan alternativt bruke omgivelsen `equation` som vil gi en stor formel med et eget nummer:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (1)$$

Typesetting av matematiske formler følger som sagt spesielle regler:

- Kun få av de vanlige `LaTeX`-kommandoene fungerer; i stedet finnes et stort utvalgte egne kommandoer.
- I matematiske formler *ignorerer alle blanke*. Det er således det samme om man skriver $1 + 2 + \dots + n$ eller $1 + 2 + \dots + n$; det blir ingen forskjell i resultatet.
- Siden blanke ignorerer, kan man sløyfe klammene etter kommandoer uten parametre (men man må sette en blank bak hvis det følger en bokstav etter).

5.2 Matematiske symboler

I matematiske formler inngår diverse symboler; her er en liten liste over de viktigste av dem. Det finnes ganske mange flere, så studer boken til Leslie Lamport og *Lokal guide til LaTeX* hvis det er noen du savner.

- Bokstaver og tall skrives som normalt: $e, x, 122, 2, 71828$.
- Bokstavmarkeringer angis med kommandoer: $\bar{a}, \vec{a}, \hat{a}, \tilde{a}$.
- Fete bokstaver får man med kommandoen `\mathbf`: $\mathbf{a} \not\sim \mathbf{a}$.
- «Skjønnskriftbokstaver» (kun $\mathcal{A} - \mathcal{Z}$) er tilgjengelig gjennom kommandoen `\mathcal`.
- Greske bokstaver angis med kommandoer: $\alpha, \beta, \gamma, \delta, \dots, \omega$. Dette gjelder også for de store bokstavene $A, B, \Gamma, \Delta, \dots, \Omega$. (Vi bruker latinske bokstaver der de er de samme som de greske.)
- De fleste vanlige matematiske tegn som finnes på tastaturet, kan brukes direkte: $+, -, /, <, >, =, '.$
- De andre matematiske symbolene må derimot angis med kommandoer: $\times, \cdot, \leq, \geq, \neq, \wedge, \vee, \in, \infty, \forall, \dots, \exists, \dots, \equiv, \sim$.
- Det finnes også diverse piler: $\leftarrow, \rightarrow, \leftrightarrow, \Leftrightarrow, \longleftarrow, \longrightarrow, \uparrow$, og mange flere.
- For de fleste symboler kan man bruke kommandoen `\not` foran for å få det negerte symbolet: $\not\sim, \not\rightarrow, \notin$.

`\item{}` Man kan angi en (nedre) indeks for alle symboler ved å bruke operatoren `\verb@_@`: x_{1} , π_{a} , $H_{2}0$.

`\item{}` Man kan også angi en øvre indeks for alle symboler ved å bruke operatoren `\verb@^@`: n^{2} , Θ^{x} . En øvre indeks kan godt kombineres med en nedre indeks: x_{1}^{2} , Ψ_{ξ}^{t+1} .

`\item{}` Brøker angis med kommandoen `\verb@frac@`: $\frac{1}{2} + \frac{1}{a+b+c+d+e} + \frac{22-b}{n}$.

`\item{}` Kvadratrøtter skrives med kommandoen `\verb@\sqrt@`: $\sqrt{2}$. Ved å anvende en opsjon i hakeparenteser, kan man også uttrykke en vilkårlig rot: $\sqrt[n]{b+4c}$.

`\item{}` Integraler og summetegn lages med henholdsvis `\verb@\int@` og `\verb@\sum@`: \int , \sum . Øvre og nedre indekser kan benyttes til å angi grenser:
`\[\int_{0}^{1} t \]; \partial t \approx \sum_{i=1}^{n} x_{i} \]`
% Her var det faktisk nødvendig å hjelpe LaTeX litt: Kommandoen `\;` er brukt
% for å legge inn litt luft mellom `t` og `\partial t`. Det er av og til påkrevet
% å foreta slike små manuelle justeringer, men ikke ofte.)

`\item{}` Vanlige parenteser og hakeparenteser fungerer som man skulle forvente: $f(x) \rightarrow A[x]$. Klammer må angis med kommandoene `\verb@{\@` og `\verb@}\@` som i uttrykket $\{x \mid x > 0\}$.

Ved å sette henholdsvis `\verb@\left@` og `\verb@\right@` foran, vil parentesene justeres til riktig størrelse:

```
\[
  \left( \lambda+2 \right) \times
  \left( \sum_{0 < k < n} \frac{2^{k^4}}{k!} \right) <
  \left\{ \int_{\alpha}^{\beta} \right\}_{\alpha+\beta=n}
\]
```

`\end{itemize}`

`\section{Figurer og tabeller}`

Et dokument inneholder ikke bare tekst, men også figurer, grafer, tabeller og lignende. Dette er det enkelt å få med i et LaTeX-dokument. Det er også trivielt å få LaTeX til å lage oversikter over hvilke figurer og tabeller som finnes i et dokument, det er bare å benytte kommandoene `\verb@\listoffigures@` og `\verb@\listoftables@`; se eksempel bakerst i dette skrevet.

Typisk for figurer og tabeller er at de gjerne er ganske store og at de ikke kan deles hvis de tilfeldigvis havner på et sideskift. Dette ordnes i LaTeX ved at figurer og tabeller «flyter». De havner altså ikke akkurat der de skal stå ifølge resten av teksten, men i nærheten. På den måten er det aldri nødvendig å dele dem. Det er allikevel lett å referere til dem i teksten med kommandoen `\verb@\vref@` (se avsnitt `\vref{X-ref}`).

- Man kan angi en (nedre) indeks for alle symboler ved å bruke operatoren $_$: x_1, π_a, H_2O .
- Man kan også angi en øvre indeks for alle symboler ved å bruke operatoren $^$: n^2, Θ^x . En øvre indeks kan godt kombineres med en nedre indeks: x_1^2, Ψ_ξ^{t+1} .
- Brøker angis med kommandoen $\frac{1}{2} + \frac{1}{a+b+c+d+e} + \frac{22-b}{n}$.
- Kvadratrøtter skrives med kommandoen $\sqrt{2}$. Ved å anvende en opsjon i hakeparenteser, kan man også uttrykke en vilkårlig rot: $\sqrt[n]{b+4c}$.
- Integraler og summetegn lages med henholdsvis \int og \sum . Øvre og nedre indekser kan benyttes til å angi grenser:

$$\int_0^1 t \, dt \approx \sum_{i=1}^n x_i$$

- Vanlige parenteser og hakeparenteser fungerer som man skulle forvente: $f(x) \rightarrow A[x]$. Klammer må angis med kommandoene $\{$ og $\}$ som i uttrykket $\{x \mid x > 0\}$.

Ved å sette henholdsvis \left og \right foran, vil parentesene justeres til riktig størrelse:

$$(\lambda + 2) \times \left(\sum_{0 < k < n} \frac{2^{k^4}}{k!} \right) < \left\{ \int_\alpha^\beta \right\}_{\alpha + \beta = n}$$

6 Figurer og tabeller

Et dokument inneholder ikke bare tekst, men også figurer, grafer, tabeller og lignende. Dette er det enkelt å få med i et \LaTeX -dokument. Det er også trivielt å få \LaTeX til å lage oversikter over hvilke figurer og tabeller som finnes i et dokument, det er bare å benytte kommandoene \listoffigures og \listoftables ; se eksempel bakerst i dette skrevet.

Typisk for figurer og tabeller er at de gjerne er ganske store og at de ikke kan deles hvis de tilfeldigvis havner på et sideskift. Dette ordnes i \LaTeX ved at figurer og tabeller «flyter». De havner altså ikke akkurat der de skal stå ifølge resten av teksten, men i nærheten. På den måten er det aldri nødvendig å dele dem. Det er allikevel lett å referere til dem i teksten med kommandoen \vref (se avsnitt 3.9 på side 10).

`\paragraph{Advarsel!}` Enkelte bruker voldsomt mye tid og krefter på å prøve å tvinge `\LaTeX{}` til å plassere en figur eller tabell et fast sted i dokumentet, for eksempel slik at de kan skrive «`\dots{}`» som vi ser i følgende figur:». Slikt blir sjelden vellykket! La heller `\LaTeX{}` få bestemme hvor flytende figurer og tabeller `\emph{bør}` plasseres og bruk kryssreferansemekanismen til å referere til den.

`\subsection{Figurer}\label{Figurer}`
`\LaTeX{}` har ikke mange og avanserte tegneprogrammer for å lage illustrasjoner, og dette er et bevisst valg. Uansett hva man bygger inn, vil brukeren aldri bli fornøyd! I stedet har man i `\LaTeX{}` valgt å la brukeren benytte egne favorittegneprogrammer til å lage illustrasjonene, så kan de etterpå enkelt hentes inn i `\LaTeX{}`-dokumentet; se for eksempel `figur\vref{fig:penn}`.

```
\begin{figure}
  \begin{center}
    \fbox{\includegraphics[width=10cm]
      {/local/lib/clipart/pubpack/misc/penandink.eps}}
    % Kommandoen \fbox tegner en ramme.
  \end{center}
  \caption{Tegning laget med Adobe Illustrator}\label{fig:penn}
\end{figure}
```

Det finnes diverse gode interaktive tegneprogrammer, for eksempel Adobe Illustrator, CorelDraw og `\texttt{xfig}`, samt et meget godt deklarativt tegneprogram: `\texttt{metapost}`. For nybegynnere er nok `\texttt{xfig}` det enkleste å bli kjent med.

Felles for alle disse tegneprogrammene er at de kan produsere såkalt `\emph{innkapslet PostScript-kode}` («`\textsc{eps}`») som enkelt kan hentes inn i `\LaTeX{-do\ -ku\ -men\ -t\ -et}` ved å benytte kommandoen `\verb@\includegraphics@`; denne kommandoen er blitt tilgjengelig fordi vi har hentet inn pakken `\texttt{graphicx}`. Figuren gjøres flytende ved å legge den inn i omgivelsen `\texttt{figure}` og utstyres med en tittel ved å benytte kommandoen `\verb@\caption@`. For mer informasjon om dette, les Lamports bok og `\emph{Lokal guide til \LaTeX{}}`.

`\subsubsection{Omgivelsen \texttt{picture}}`
Selv om `\LaTeX{}` ikke har avanserte innebygde tegneverktoy, har den en egen omgivelse for å lage enkle strektegninger; et eksempel på en slik tegning er `figur\vref{fig:Arbeid}`. Beskrivelse av hvorledes man benytter `\texttt{picture}`-omgivelsen finnes i Leslie Lamports bok.



Figur 2: Tegning laget med Adobe Illustrator

Advarsel! Enkelte bruker voldsomt mye tid og krefter på å prøve å tvinge \LaTeX til å plassere en figur eller tabell et fast sted i dokumentet, for eksempel slik at de kan skrive «... som vi ser i følgende figur:». Slikt blir sjelden vellykket! La heller \LaTeX få bestemme hvor flytende figurer og tabeller *bør* plasseres og bruk kryssreferansemekanismen til å referere til den.

6.1 Figurer

\LaTeX har ikke mange og avanserte tegneprogrammer for å lage illustrasjoner, og dette er et bevisst valg. Uansett hva man bygger inn, vil brukeren aldri bli fornøyd! I stedet har man i \LaTeX valgt å la brukeren benytte egne favorittegneprogrammer til å lage illustrasjonene, så kan de etterpå enkelt hentes inn i \LaTeX -dokumentet; se for eksempel figur 2.

Det finnes diverse gode interaktive tegneprogrammer, for eksempel Adobe Illustrator, CorelDraw og `xfig`, samt et meget godt deklarativt tegneprogram: `metapost`. For nybegynnere er nok `xfig` det enkleste å bli kjent med.

Felles for alle disse tegneprogrammene er at de kan produsere såkalt *innkapslet PostScript-kode* («EPS») som enkelt kan hentes inn i \LaTeX -dokumentet ved å benytte kommandoen `\includegraphics`; denne kommandoen er blitt tilgjengelig fordi vi har hentet inn pakken `graphicx`. Figuren gjøres flytende ved å legge den inn i omgivelsen `figure` og utstyres med en tittel ved å benytte kommandoen `\caption`. For mer informasjon om dette, les Lamports bok og *Lokal guide til \LaTeX* .

6.1.1 Omgivelsen `picture`

Selv om \LaTeX ikke har avanserte innebygde tegneverktøy, har den en egen omgivelse for å lage enkle strektegninger; et eksempel på en slik tegning er figur 4 på side 17. Beskrivelse av hvorledes man benytter `picture`-omgivelsen finnes i Leslie Lamports bok.

```
\subsection{Tabeller}
```

På samme måte som figurer, kan tabeller gjøres flytende ved å legge dem inn i omgivelsen `\texttt{table}`. Selve tabellen lages ved hjelp av omgivelsen `\texttt{tabular}` slik det er vist i figur\href{fig:tabell}. (Omgivelsen `\texttt{center}` er ikke nødvendig, men det tar seg ofte best ut å sentrere tabellen.)

```
\begin{figure}
  \rule{\textwidth}{1pt} % \rule gir her en horisontal strek 1 punkt tykk.
  \begin{verbatim}
\begin{table}
  \begin{center}
    \begin{tabular}{<Definisjon av kolonner>} \hline
      <Data>& <Data> & ... & <Data>\\
      <Data>& <Data> & ... & <Data>\\
      :      :      :      :
      <Data>& <Data> & ... & <Data>\\ \hline
    \end{tabular}
  \end{center}
  \caption{<Tittel på tabellen>}\label{<En etikett>}
\end{table}
  \end{verbatim}
  \vspace{-15pt} % Fjern litt vertikal luft; det var litt for mye.
  \rule{\textwidth}{1pt}
  \caption{Oppsett for tabell}\label{fig:tabell}
\end{figure}
```

Følgende gjelder for `\texttt{tabular}`-omgivelsen:

```
\begin{itemize}
\item{} Omgivelsen \texttt{tabular} har en parameter som definerer kolonnene i tabellen, det vil si antallet kolonner, om de skal være høyre- eller venstrejustert eller sentrert, og hvor det skal være vertikale skillelinjer. De aktuelle spesifikasjonene er vist i tabell\href{tab:Kolonner}.
```

```
\begin{table}
  \begin{center}
    \begin{tabular}{|c|l|} \hline
      \textbf{Kode}& \textbf{Betydning}\\ \hline
      \texttt{||}& Vertikal delelinje\\
      \texttt{l}& Venstrejustert tekst\\
      \texttt{c}& Sentrert tekst\\
      \texttt{r}& Høyrejustert tekst\\ \hline
    \end{tabular}
  \end{center}
  \caption{Spesifikasjon av kolonner for \texttt{tabular}}\label{tab:Kolonner}
\end{table}
```

```
\item{} Kommandoen \verb@hline@ lager en horisontal linje.
```

```
\item{} Data som skal inn i tabellen kommer rad for rad. Kolonnene i raden adskilles med \verb@&&@, og hver rad avsluttes med \verb@\\@.
\end{itemize}
```

```

\begin{table}
  \begin{center}
    \begin{tabular}{«Definisjon av kolonner»} \hline
      «Data»& «Data» & ... & «Data»\\
      «Data»& «Data» & ... & «Data»\\
      :      & :      & :      & :      \\
      «Data»& «Data» & ... & «Data»\\ \hline
    \end{tabular}
  \end{center}
  \caption{«Tittel på tabellen»}\label{«En etikett»}
\end{table}

```

Figur 3: Oppsett for tabell

Kode	Betydning
	Vertikal delelinje
l	Venstrejustert tekst
c	Sentrert tekst
r	Høyrejustert tekst

Tabell 2: Spesifikasjon av kolonner for tabular

6.2 Tabeller

På samme måte som figurer, kan tabeller gjøres flytende ved å legge dem inn i omgivelsen `table`. Selve tabellen lages ved hjelp av omgivelsen `tabular` slik det er vist i figur 3. (Omgivelsen `center` er ikke nødvendig, men det tar seg ofte best ut å sentrere tabellen.)

Følgende gjelder for `tabular`-omgivelsen:

- Omgivelsen `tabular` har en parameter som definerer kolonnene i tabellen, det vil si antallet kolonner, om de skal være høyre- eller venstrejustert eller sentrert, og hvor det skal være vertikale skillelinjer. De aktuelle spesifikasjonene er vist i tabell 2.
- Kommandoen `\hline` lager en horisontal linje.
- Data som skal inn i tabellen kommer rad for rad. Kolonnene i raden adskilles med `&`, og hver rad avsluttes med `\\`.

```
\section{Bruk av \LaTeX{}}
```

Som tidligere sagt, foregår arbeidet med et \LaTeX{}-dokument i tre faser: redigering av kildekoden, kompilering, og utskrift til skjermen eller på papir.

Redigeringen av kildekoden kan skje i hvilket som helst redigeringsprogram, men svært mange foretrekker Emacs.

Når det gjelder kompileringen, kan denne enten gjøres direkte fra Emacs eller fra et kommandovindu.

```
\subsection{Prosessering av \LaTeX{}}
```

Arbeidsprosessen med å skrive et \LaTeX{}-dokument er vist i figur\vref{fig:Arbeid} og de aktuelle kommandoene er:

```
\begin{description}
```

```
\item[\texttt{latex}] kompilerer en kildefil (med et navn som \texttt{fil.tex} siden det er vanlig at \LaTeX{}-filer har navn som slutter på \texttt{.tex}) til en såkalt \textsc{dvi}-fil (med navn \texttt{fil.dvi}), som er en utstyrsuavhengig representasjon av det ferdige dokumentet.
```

```
\item[\texttt{preview}, \texttt{xdvi}, \texttt{yap} eller andre] viser innholdet av en \textsc{dvi}-fil på skjermen.
```

```
\item[\texttt{print} eller tilsvarende] skriver innholdet av en fil ut på en skriver.
```

```
\end{description}
```

```
\section{Veien videre}
```

Når du nå er ferdig med denne enkle innføringen i \LaTeX{}, skulle du være i stand til å bruke det til enkle rapporter eller kanskje en hovedoppgave.

Det er imidlertid mye mer interessant å vite om \LaTeX; de viktigste kildene til slik informasjon er følgende (nevnt i den rekkefølgen de bør leses):

```
\begin{description}
```

```
\item[\texttt{sample.tex}] er en kort ( $\frac{1}{2}$  side) oversikt over \LaTeX{} skrevet av Leslie Lamport. Ved å kompilere og skrive ut kildekoden og så sammenholde med resultatet, får man en god oversikt over hvorledes de vanligste kommandoene fungerer. Dette gjøres enkelt med følgende
```

```
UNIX-kommandoer:
```

```
\begin{verbatim}
```

```
latex sample2e.tex
```

```
print sample.dvi
```

```
print /usr/local/texmf/tex/latex/base/sample2e.tex
```

```
\end{verbatim}
```

(Det kan variere hvor denne filen ligger.)

7 Bruk av L^AT_EX

Som tidligere sagt, foregår arbeidet med et L^AT_EX-dokument i tre faser: redigering av kildekoden, kompilering, og utskrift til skjermen eller på papir.

Redigeringen av kildekoden kan skje i hvilket som helst redigeringsprogram, men svært mange foretrekker Emacs.

Når det gjelder kompileringen, kan denne enten gjøres direkte fra Emacs eller fra et kommandovindu.

7.1 Prosessering av L^AT_EX

Arbeidsprosessen med å skrive et L^AT_EX-dokument er vist i figur 4 på neste side og de aktuelle kommandoene er:

`latex` kompilerer en kildefil (med et navn som `fil.tex` siden det er vanlig at L^AT_EX-filer har navn som slutter på `.tex`) til en såkalt DVI-fil (med navn `fil.dvi`), som er en utstyrsuavhengig representasjon av det ferdige dokumentet.

`preview`, `xdvi`, `yap` **eller andre** viser innholdet av en DVI-fil på skjermen.

`print` **eller tilsvarende** skriver innholdet av en fil ut på en skriver.

8 Veien videre

Når du nå er ferdig med denne enkle innføringen i L^AT_EX, skulle du være i stand til å bruke det til enkle rapporter eller kanskje en hovedoppgave. Det er imidlertid mye mer interessant å vite om L^AT_EX; de viktigste kildene til slik informasjon er følgende (nevnt i den rekkefølgen de bør leses):

`sample.tex` er en kort ($2\frac{1}{2}$ side) oversikt over L^AT_EX skrevet av Leslie Lamport. Ved å kompilere og skrive ut kildekoden og så sammenholde med resultatet, får man en god oversikt over hvorledes de vanligste kommandoene fungerer. Dette gjøres enkelt med følgende UNIX-kommandoer:

```
latex sample2e.tex
print sample.dvi
print /usr/local/texmf/tex/latex/base/sample2e.tex
```

(Det kan variere hvor denne filen ligger.)

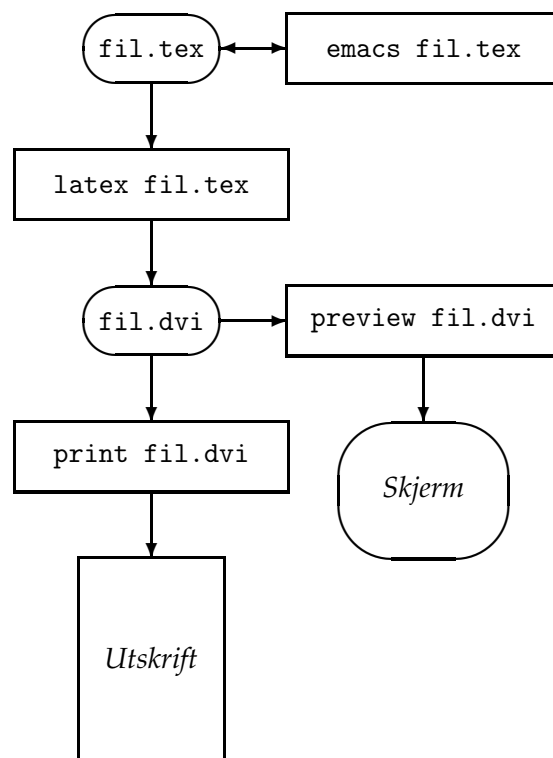
\item[The not so short introduction to \LaTeXe{}] er en 101~sider lang innføring i \LaTeX{} skrevet av \textit{Tobias Oetiger}, \textit{Hubert Partl}, \textit{Irene Hyna} og \textit{Elisabeth Schlegl}. Dette heftet ligger lagret på maskinlesbar form, for eksempel på filen \texttt{/local/texmf/doc/intro/lshort2e.dvi}.

\item[\LaTeX{} user's guide and reference manual] er skrevet av \emph{Leslie Lamport} som er mannen bak \LaTeX{}. Denne boken er en grundig innføring i \LaTeX{} og er absolutt nødvendig for alle som vil bruke \LaTeX{} seriøst. Den er utgitt på forlaget Addison-Wesley og har ISBN~0-201-52983-1. Husk å få annen utgave.

\item[The \LaTeX{} Companion] av \emph{Michel Goossens, Frank Mittelbach og Alexander Samarin} er den store boken for avanserte \LaTeX{}-brukere; den inneholder omtrent alt man trenger å vite om \LaTeX{}. Den er litt gammel nå, men er stadig svært nyttig.

\end{description}

```
\begin{figure}
\begin{center}
\setlength{\unitlength}{9mm}
\begin{picture}(10,11)
\thicklines
\put(1,10){\makebox(2,1){\texttt{fil.tex}}}
\put(2,10.5){\oval(2,1)}
\put(2,10){\vector(0,-1){1}}
\put(3,10.5){\vector(1,0){1}}
\put(3,10.5){\vector(-1,0){0}}
\put(4,10){\framebox(4,1){\texttt{\textbf{emacs fil.tex}}}}
\put(0,8){\framebox(4,1){\texttt{\textbf{latex fil.tex}}}}
\put(2,8){\vector(0,-1){1}}
\put(1,6){\makebox(2,1){\texttt{fil.dvi}}}
\put(2,6.5){\oval(2,1)}
\put(2,6){\vector(0,-1){1}}
\put(3,6.5){\vector(1,0){1}}
\put(4,6){\framebox(4,1){\texttt{\textbf{preview fil.dvi}}}}
\put(6,6){\vector(0,-1){1}}
\put(4.75,3){\makebox(2.5,2){\textit{Skjerm}}}
\put(6,4){\oval(2.5,2)}
\put(0,4){\framebox(4,1){\texttt{\textbf{print fil.dvi}}}}
\put(2,4){\vector(0,-1){1}}
\put(0.94,0){\framebox(2.12,3){\textit{Utskrift}}}
\end{picture}
\end{center}
\caption{Arbeidsprosessen i \LaTeX{}}
\label{fig:Arbeid}
\end{figure}
```



Figur 4: Arbeidsprosessen i \LaTeX

The not so short introduction to $\text{\LaTeX} 2_{\epsilon}$ er en 101 sider lang innføring i \LaTeX skrevet av *Tobias Oetiger, Hubert Partl, Irene Hyna og Elisabeth Schlegl*. Dette heftet ligger lagret på maskinlesbar form, for eksempel på filen `/local/texmf/doc/intro/lshort2e.dvi`.

\LaTeX user's guide and reference manual er skrevet av *Leslie Lamport* som er mannen bak \LaTeX . Denne boken er en grundig innføring i \LaTeX og er absolutt nødvendig for alle som vil bruke \LaTeX seriøst. Den er utgitt på forlaget Addison-Wesley og har ISBN 0-201-52983-1. Husk å få annen utgave.

The \LaTeX Companion av *Michel Goossens, Frank Mittelbach og Alexander Samarin* er den store boken for avanserte \LaTeX -brukere; den inneholder omtrent alt man trenger å vite om \LaTeX . Den er litt gammel nå, men er stadig svært nyttig.

% Her kommer så dokumentets «hale» med innholdsfortegnelse og lignende.

```
\newpage{}          % Jeg vil ha innholdsfortegnelsen på egen side.  
\tableofcontents{}
```

Innhold

1	Bakgrunn	1
2	Tekst og kommandoer	1
2.1	Kommentarer	2
2.2	Enkle kommandoer	2
2.3	Kommandoer med parametre	3
2.3.1	Kommandoer med opsjoner	3
2.4	Omgivelser	3
2.4.1	Lister	3
2.4.2	Andre omgivelser	4
3	Dokumentets struktur	5
3.1	Hovedstrukturen	6
3.2	Dokumentklasser	6
3.2.1	Opsjoner til dokumentklassen	7
3.3	Forfatter og tittel	7
3.4	Pakker	7
3.5	Et lite eksempel	8
3.6	Strukturkommandoene	9
3.6.1	Andre strukturkommandoer	9
3.7	Innholdsfortegnelse	9
3.8	Fotnoter	9
3.9	Kryssreferanser	10
4	Typesnitt	10
4.1	Skifte typesnitt for hele dokumentet	11
5	Matematiske formler	11
5.1	Omgivelser for matematiske formler	11
5.2	Matematiske symboler	12
6	Figurer og tabeller	13
6.1	Figurer	14
6.1.1	Omgivelsen picture	14
6.2	Tabeller	15
7	Bruk av L^AT_EX	16
7.1	Prosessering av L ^A T _E X	16
8	Veien videre	16

```
\newpage{}          % Jeg vil ha figur- og tabelloversiktene på egen side.  
\listoffigures{}   % Gir en oversikt over alle figurene.  
\listoftables{}    % Gir en oversikt over alle tabellene.  
\end{document}
```


Figurer

1	Et minimalt L ^A T _E X-dokument	9
2	Tegning laget med Adobe Illustrator	14
3	Oppsett for tabell	15
4	Arbeidsprosessen i L ^A T _E X	17

Tabeller

1	Oversikt over dokumentklasser	6
2	Spesifikasjon av kolonner for tabular	15