



# **L<sup>A</sup>T<sub>E</sub>X— del II**

## **Dagens opplegg**

- L<sup>A</sup>T<sub>E</sub>X-kommandoer
- Lister
- Matematiske formler
- Tabeller
- Litt typografi
- Egendefinerte kommandoer

# Navnet T<sub>E</sub>X

Navnet T<sub>E</sub>X kommer fra det greske

*technologia* (τεχνολογια)

som betyr noe à la «systematisk gjennomgang av en kunst».

T<sub>E</sub>X er altså en kombinasjon av kunst og vitenskap, slik også typografien er det.

# L<sup>A</sup>T<sub>E</sub>X-kommandoene

L<sup>A</sup>T<sub>E</sub>X-kommandoene ser alltid slik ut:

`\navn {parameter} ...`

Om det ikke er noen parametre, kan man skrive det slik:

`\navn {}`

eller som

`\navn`

**Men** i det siste tilfellet vil blanke etter kommandoen bli spist!

Skriv	Resultat
<code>\LaTeX{} er bra.</code>	L <sup>A</sup> T <sub>E</sub> X er bra.
<code>\LaTeX er bra.</code>	L <sup>A</sup> T <sub>E</sub> Xer bra.

## Kommandoen `\verb`

Kommandoen `\verb` er av og til nyttig til å angi tekst med mange rare tegn:

Skriv	Resultat
<code>\verb:\dag{} og \S.:</code>	<code>\dag{} og \S.</code>

(Resultatet kommer alltid i skrivemaskinskrift.)

**Men** denne kommandoen kan *aldri* være parameter til andre kommandoer (dvs stå inni {}-er). Det er kanskje best å la den være?!

## Automatisk genererte lister

L<sup>A</sup>T<sub>E</sub>X kan gjøre en del automatisk for oss:

**`\tableofcontents`** lager automatisk en innholdsfortegnelse.

**`\listoffigures`** gjør det samme med figurer (i `\begin{figure}... \end{figure}`).

**`\listoftables`** gjør det samme med tabeller (i `\begin{table}... \end{table}`).

# Lister

Lister er usedvanlig nyttige i vitenskapelig arbeider:

- De hjelper forfatteren å strukturere ideene
- De gjør det lettere for leseren å følge forfatterens tanker.
- De bryter opp en lang uoversiktlig tekst.

L<sup>A</sup>T<sub>E</sub>X har tre typer lister:

**description** er «leksikonaktig» (som denne).

**itemize** er punktvis (som den forrige).

**enumerate** nummererer punktene slik:

- ① Hvert nytt punkt startes med `\item`.
- ② I en *description* må vi angi beskrivelsesordet: `\item[ord]`
- ③ Man kan godt ha lister inni lister.

# Matematikk

Det virker uvant å «programmere» formler, men det går greit med trening. Følgende formel laget jeg på 2 minutter og 3 sekunder:

$$\pi(n) = \sum_{m=2}^n \left[ \left( \sum_{k=1}^{m-1} \lfloor (m(k) / \lceil m/k \rceil) \rfloor \right)^{-1} \right]$$

- Det går faktisk forttere med slike kommandoer enn med pek-og-klikk.
- Man kan lett klippe og lime.
- Enkelt å endre formlene.
- Man kan programmere formler eller deler av formler.

Formler finnes i to varianter: *små formler* som  $\sum_{k=1} 2^{-k}$  som står sammen med den vanlige teksten, eller *store formler* som

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

som kommer på en linje for seg. Kommandoene for formlene er de samme, men  $\LaTeX$  vil formatere dem ulikt.

## Omgivelser for matematiske formler

I  $\LaTeX$  skrives små formler som  $\dots$ , mens  $\left[ \dots \right]$  benyttes til store formler. Alternativt finnes  $\begin{equation} \dots \end{equation}$  som gir en stor formel med eget nummer:

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



Typesetting av matematiske formler følger spesielle regler:

- Kun få av de vanlige  $\text{\LaTeX}$ -kommandoene fungerer; i stedet finnes et stort utvalgte egne kommandoer.
- I matematiske formler *ignoreres alle blanke*.

## Matematiske symboler

I matematiske formler inngår diverse symboler; her er en liten liste over de viktigste av dem.

- Bokstaver og tall skrives som normalt:

Skriv	Resultat
-------	----------

e	$e$
---	-----

122	122
-----	-----

- Greske bokstaver angis med kommandoer:

Skriv	Resultat
-------	----------

$\backslash\alpha$ , $\backslash\gamma$ , $\backslash\omega$	$\alpha$ , $\gamma$ , $\omega$
--	--------------------------------

A, $\backslash\Gamma$ , $\backslash\Omega$	A, $\Gamma$ , $\Omega$
--	------------------------

- De fleste vanlige matematiske tegn som finnes på tastaturet, kan brukes direkte:

Skriv	Resultat
+ , - , / , < , > , = , ' ,	+ , - , / , < , > , = , ' ,

- De andre matematiske symbolene må derimot angis med kommandoer:

Skriv	Resultat
<code>\times</code> , <code>\cdot</code> , <code>\leq</code> ,	$\times$ , $\cdot$ , $\leq$
<code>\geq</code> , <code>\neq</code> , <code>\land</code>	$\geq$ , $\neq$ , $\wedge$
<code>\lor</code> , <code>\in</code> , <code>\approx</code>	$\vee$ , $\in$ , $\approx$
<code>\forall</code> , <code>\exists</code> , <code>\notin</code>	$\forall$ , $\exists$ , $\notin$

- Det finnes også diverse piler:

Skriv	Resultat
<code>\leftarrow</code> , <code>\leftrightarrow</code>	$\leftarrow$ , $\leftrightarrow$
<code>\rightarrow</code> , <code>\uparrow</code>	$\Rightarrow$ , $\uparrow$

- Man kan angi en nedre indeks med operatoren `_`:

Skriv	Resultat
<code>x_{1}</code> , <code>\pi_{a}</code> , <code>H_{2}O</code>	$x_1$ , $\pi_a$ , $H_2O$

- En øvre indeks får man ved å bruke operatoren  $\wedge$ :

Skriv	Resultat
$n^{\wedge\{2\}}, \Theta^{\wedge\{x\}}$	$n^2, \Theta^x$

Indekser kan kombineres

Skriv	Resultat
$x_{\{1\}}^{\wedge\{2\}}, \Psi_{\xi}^{\wedge\{t+1\}}$	$x_1^2, \Psi_{\xi}^{t+1}$

- Brøker angis med  $\backslash\text{frac}$ :

Skriv	Resultat
$\backslash\text{frac}\{1\}\{2\} + \backslash\text{frac}\{1\}\{a+b+c+d\} + \backslash\text{frac}\{22-b\}\{n\}$	$\frac{1}{2} + \frac{1}{a+b+c+d} + \frac{22-b}{n}$

- Kvadratrøtter skrives med  $\backslash\text{sqrt}$ :

Skriv	Resultat
$\backslash\text{sqrt}\{2\} < \backslash\text{sqrt}[n]\{b+4c\}$	$\sqrt{2} < \sqrt[n]{b+4c}$

- Integraler og summetegn lages med `\int` og `\sum`:

**Skriv**

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

**Resultat**

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

- Parenteser er også greie (med unntak av «{» og «}»):

**Skriv**

$$f(x) \rightarrow A[x] \\ \{x \mid x > 0\}$$

**Resultat**

$$f(x) \rightarrow A[x] \\ \{x \mid x > 0\}$$

$$\left( \sum_{0 < k < n} \frac{2^k}{k!} \right)$$

$$\left( \sum_{0 < k < n} \frac{2^k}{k!} \right)$$

$$\left\{ \int_{\alpha}^{\beta} \right\}_{\alpha + \beta = n}$$

$$\left\{ \int_{\alpha}^{\beta} \right\}_{\alpha + \beta = n}$$

# Tabeller

De fleste tabeller er greie å lage i  $\LaTeX$ , om man jobber strukturert og stadig kjører  $\LaTeX$  etterhvert som tabellen skrives.

```
\begin{tabular}{oppsett}
  rad\\
  rad\\
  :
\end{tabular}
```

Oftest vil man ha en «flytende tabell» og skriver da

```
\begin{table}
  \begin{center}
    \begin{tabular}{oppsett}
      rad\\
      rad\\
      :
    \end{tabular}
    \caption{En passende tittel\label{ref-navn}}
  \end{center}
\end{table}
```

## Tabelloppsettet

Oppsettet forteller hvor mange kolonner vi har og hvorledes de skal justeres:

**l** venstrejustert kolonne

**c** sentrert kolonne

**r** høyrejustert kolonne

**|** vertikal skillelinje

## Eksempel

Denne koden

```
\begin{tabular}{lc|r}  
  rader  
\end{tabular}
```

gir

a	b		c
venstre	sentrert		høyre

## Tabellradene

Hver rad skrives slik:

*data& data& ... \\*

Horisontale linjer legges inn med `\hline`.

## Eksempel

Denne koden

```
\begin{tabular}{|c||}
```

```
\hline
```

```
\textbf{Notasjon}& \textbf{Betydning}\\
```

```
\hline\hline
```

```
\texttt{l}& Venstrejustert\\ \hline
```

```
\texttt{c}& Sentrert\\ \hline
```

```
\texttt{r}& Høyrejustert\\ \hline
```

```
\end{tabular}
```

gir

<b>Notasjon</b>	<b>Betydning</b>
l	Venstrejustert
c	Sentrert
r	Høyrejustert

## Vanlige feil

### Feilmeldingen

! Extra alignment tab has been changed to \cr.

betyr at man har flere dataelementer i en rad enn det er kolonner (eller man har glemt en \\).

### En skjev tabell

Før	Etter
a	

skyldes at det er for få dataelementer:

```
\begin{tabular}{|c|c|}
\hline
\textbf{Før}&\ \textbf{Etter}\\ \hline
a\\ \hline
\end{tabular}
```



## Mer avanserte tabeller

Med kommandoen `\multicolumn` kan man

- slå sammen kolonner i en rad
- endre oppsettet for en celle

og med `\cline` kan man tilpasse horisontale linjer.

```
\begin{tabular}{r|c|c|c|c|}
\cline{2-5}
& \multicolumn{2}{c|}{\textbf{Opp}} & \multicolumn{2}{c|}{\textbf{Ned}} \\
\cline{2-5}
& \textbf{H} & \textbf{V} & \textbf{H} & \textbf{V} \\
1. & 7,2 & 8,4 & 0,1 & 9,1 \\
2. & 7,1 & 8,3 & 0,0 & 8,9 \\
\end{tabular}
```

	<b>Opp</b>		<b>Ned</b>	
	<b>H</b>	<b>V</b>	<b>H</b>	<b>V</b>
1.	7,2	8,4	0,1	9,1
2.	7,1	8,3	0,0	8,9

# Litt typografi

Når man har muligheten, bør man bruke korrekt typografi.

## Anførelstegn

De riktige norske anførelstegnene er « og ». De kan skrives rett inn<sup>†</sup> eller som «<<» og «>>».

På engelsk er det riktig å bruke

## Amerikansk engelsk

“HAL said, ‘Good morning, Dave,’”  
recalled Frank.

## Britisk engelsk

‘HAL said, “Good morning, Dave,”’  
recalled Frank.

Doble anførelstegn skrives «“» og «”».

<sup>†</sup> På en Windows-maskin kan du bruke Alt+0171 og Alt+187 på det numeriske tastaturet.

## Ulike streker

Det finnes fire typer korte horisontale streker:

- er en bindestrek.
- er en intervallstrek i «4-8» eller «Oslo-Bergen»; den skrives «--».
- er en tankestrek — når man gjør et større opphold. Den skrives «---».
- er et minustegn. Det får man automatisk i formler.

## Usynlige bånd

Noen ganger trenger man et «ubrytelig mellomrom»:

1 000 000 kr      nr 4

Da skriver vi «~».

## Orddelinger

L<sup>A</sup>T<sub>E</sub>X har en rimelig avansert algoritme for orddeling og er ganske flink til det, men noen ganger trenger den hjelp.

pils-piss urin-divided bidrag-syter

Vi kan angi unntaksord med

```
\hyphenation{pil-spiss ur-indi-videt bidrags-yter}
```

Vi kan også angi delingspunkter i det enkelte ordet:

```
Simu\ -la-tanke\ -gangen  
Stol\ -ten\ -berg-regje\ -ringen
```

Dette er nødvendig i ord med bindestrek.  
(Men ikke gjør dette før helt til slutt.)

I noen få helt spesielle ord kan vi trenge

```
tall\discretionary{-}{l}{inje  
trafikk\discretionary{-}{k}{onstabel
```

## Egne kommandoer

Erfarne brukere definerer etter hvert mange nye  $\text{\LaTeX}$ -kommandoer:

- Man kan spare skrivearbeid.
- Det er lett å endre utseendet av konstruksjoner man bruker.

En ny kommando med  $n$  parametre defineres slik:

```
\newcommand{\navn}[n]{definisjon}
```

I definisjonen omtales parametrene som «#1», «#2» osv.

## Endre utseendet

Når man plutselig lurer på hvordan noe skal se ut, bør man definere et kommando for det.

## Eksempel

Skal nye begrep skrives med kursiv eller fet skrift?

```
\newcommand{\NyttBegrep}[1]{\textit{#1}}
```

En *font* i  $\LaTeX$  angir en implementasjon av et typesnitt i form av kode eller et raster.

```
\newcommand{\NyttBegrep}[1]{\textbf{#1}}
```

En **font** i  $\LaTeX$  angir en implementasjon av et typesnitt i form av kode eller et raster.

## Spare skriving

Når man har skrevet den samme koden flere ganger, bør man vurdere å skrive en kommando for den.

### Eksempel

I dette kurset snakker jeg ofte om  $\LaTeX$ -kommandoer som `\documentclass` og `\maketitle`.

```
\newcommand{\p}[1]{\textsf{#1}}  
\newcommand{\lacmd}[1]{\p{\textbackslash}#1}  
  
\lacmd{maketitle}
```

### Eksempel

Ende mer jobb er det å skrive `\begin{figure}... \end{figure}`.

```
\newcommand{\l apar}[1]{\p{\{#1\}}}  
\newcommand{\laenv}[1]{\lacmd{begin}\l apar{#1}\dots  
  \hspace{-0.1em}\lacmd{end}\l apar{#1}}
```

## Oppgave

På <http://www.nik.no/2005/Nes.pdf> ligger en artikkel om analyse av skihopp på grunnlag av videoopptak.

På <http://www.ifi.uio.no/~dag/Nes.txt> ligger artikkelteksten og på <http://www.ifi.uio.no/~dag/Nes-1.eps> til -4.eps ligger de fire illustrasjonene.

Prøv å få skikk på artikkelen. (Det er ikke nødvendig å ta med bibliografien på det nåværende tidspunkt.)