

# Introduksjon til L<sup>A</sup>T<sub>E</sub>X

Johannes Sørby Heines

13. oktober 2023

## Sammendrag

Dette dokumentet er laget for å gi en innføring i L<sup>A</sup>T<sub>E</sub>X, med nok informasjon til å skrive det som forventes av oppgavebesvarelser og rapporter i løpet av bachelorstudiet. Jeg går gjennom overskrifter, henvisninger, likninger, figurer og programkode, samt flere nyttige pakker for skriving i fysikk og generelt. Jeg forsøker å gi en forståelse av prinsippene i L<sup>A</sup>T<sub>E</sub>X som vil gjøre det lettere å lete seg fram til mer informasjon der det trengs. Forslag til forbedringer mottas med takk på <https://github.com/johashei/LaTeXintro>. Der ligger også den nyeste versjonen av dette dokumentet.

Åpne dette dokumentet i en tex-editor slik at du lett kan se både tex-filen og den genererte pdf-en.

## I Dette er en overskrift.

Hvis du bruker `\documentclass[...]{report}` har du også kapitler, som kommer før overskrifter.

### I–1 Dette er en underoverskrift.

Teksten ser lik ut uavhengig av hvor mange overskriftnivåer du har.

#### a) Dette er en underunderoverskrift.

Hvis du trenger tre overskriftnivåer bør du tenke deg om. Blir teksten klarere om den omstruktureres?

Som du ser over kommer det ikke innrykk på første avsnitt etter en overskrift, innrykket kommer derimot når et avsnitt følger et annet.

## II Nummerering

## II-1 Overskrifter

Du kan fritt endre hvordan overskrifter nummereres ved å redefinere kommandoene `\thesection`, `\thesubsection`, og `\thesubsubsection`. Prøv deg fram og finn hva som passer best.

# Overskrift uten nummer

Du kan også lage en overskrift uten nummer ved å sette en \* mellom kommandoen og {}, slik: `\subsection*{Overskrift uten nummer}`. Nest overskrift vil fortsette fra den forrige nummererte overskriften, som dette:

## II-2 Henvisninger

Det kan ofte være hensiktsmessig å vise til en bestemt seksjon i teksten. Hvis du skriver f.eks. “Se avsnitt II–2”, men så bestemmer deg for å flytte denne delen eller legge til en annen del før, må du endre alle henvisninger manuelt. For å slippe det kan du sette merker på overskrifter med kommandoen `\label{}` og henvise til den med `\ref{}` eller `\cref{}`. Den første er innebygd i L<sup>A</sup>T<sub>E</sub>X og gjør ikke annet enn å gi deg nummeret: II–2. Den andre ligger i pakken `cleveref` og vet også hva du refererer til: avsnitt II–2.

### III Matematikk

En av hovedgrunnene til å bruke L<sup>A</sup>T<sub>E</sub>X er å skrive matematikk. Det finner flere måter å gjøre dette på, som passer til forskjellige formål.

Når matematikken skal være integrert i teksten – såkalt “inline” – skrives den mellom  $\backslash \langle \rangle$ . For eksempel  $r \exp(\theta i) = r \cos(\theta) + i r \sin(\theta)$ . For uttrykk som krever litt mer plass brukes  $\backslash [ \ ]$  – kalt “display”, for eksempel

$$f(x) \approx \sum_{n=0}^{\infty} \frac{(x-a)^n}{n!} \frac{d^n f}{dx^n}(a).$$

Husk at også likninger som skrives på denne måten skal kunne leses som en del av teksten, så sett komma eller punktum etter dem.

Ren  $\text{\TeX}$ , som  $\text{\LaTeX}$  er basert på, bruker  $\$$   $\$$  og  $\$$  $\$$   $\$$  $\$$  for henholdsvis inline- og display-matematikk, og dette fungerer også i  $\text{\LaTeX}$ , men kommandoene over er anbefalt.

For lengre utledninger er det fint å kunne sette likninger pent under hverandre, det gjør vi med `align`. Her brukes `&` for å markere referansepunktet, og `\` for å markere linjeskift.

$$\frac{dN}{dt} = -\frac{N}{\tau} \tag{1}$$

$$\frac{1}{N} \frac{dN}{dt} = -\frac{1}{\tau}$$

$$\int \frac{1}{N} dN = \int -\frac{1}{\tau} dt \tag{2}$$

$$\ln(N) = -\frac{t}{\tau}$$

$$N = e^{-t/\tau} \tag{3}$$

Som med overskrifter kan du bruke `\label{}` og `\ref{}` eller `\cref{}` for å enkelt vise til en bestemt likning. For eksempel er likning (3) formelen for radioaktivt henfall. `\nonumber` fjerner nummereringen for én likning, hvis du vil fjerne nummerering for alle kan du bruke `align*`.

## IV Figurer

Figurer i vitenskapelige tekster skal ha nummer og figurtekst under.  $\text{\LaTeX}$  forsøker å plassere figurer slik at det ikke blir for mye tomrom i teksten. Det betyr at figurer ofte ikke ender opp akkurat der du definerer dem i tex-filen. Du kan sende ekstra instruksjoner om hvordan figuren skal plasseres:

- **h**: “here” Finn den beste plasseringen i nærheten av dette punktet i teksten.
- **t**: “top” Plasser figuren øverst på siden.
- **b**: “bottom” Plasser figuren nederst på siden.
- **p**: “page” Lag en egen side med bare figurer.



Figur 1: Fotografi av nordlys sett fra Sognsvann 16. Mars 2023.

På samme måte som for overskrifter og likninger kan du henvise til figurer. Figur 1 er et .jpg bilde, men de fleste formater støttes. Når du lager en figur i f.eks. python er det best å lagre den som pdf: da slipper du pikselering.

## V Kode

Det er ofte nødvendig å ta med programkode i dokumentet. Det er dette som er den viktigste funksjonen til pakken listings. Den gir deg mulighet til å bestemme akkurat hvordan koden skal se ut. I opplisting 1 kan du se et eksempel, merk at du kan referere til kode på samme måte som til figurer, men label og caption settes som argumenter i kommandoen `\listinginputlisting`, ikke med egne kommandoer. Innstillingene kan settes både i dokumentet og i preamble. Se dokumentasjonen for detaljer.

Listing 1: Et enkelt program for å trille terninger.

```
import numpy as np
import sys

def main():
    hjelp = "bruk: python terning.py <antall sider> <antall kast>"
```

```

try:
    antall_sider = int(sys.argv[1])
    antall_kast = int(sys.argv[2])
    if (antall_sider < 1) or (antall_kast < 1):
        raise ValueError
except IndexError:
    print(hjelp)
    sys.exit()
except ValueError:
    print(hjelp)
    print("<antall sider> og <antall kast> må være positive heltall.")
    sys.exit()

terning = Terning(antall_sider)
resultat = terning.kast(antall_kast)
print(f"Kastet {terning} {antall_kast} ganger:\n{resultat}")

class Terning:
    def __init__(self, antall_sider):
        self.rng = np.random.default_rng()
        self.antall_sider = antall_sider

    def __str__(self):
        return f"en {self.antall_sider}-sidet terning"

    def kast(self, antall_kast):
        return self.rng.integers(
            low=1,
            high=self.antall_sider,
            size=antall_kast,
            endpoint=True)

if __name__ == "__main__":
    main()

```