UNIVERSITETET I OSLO

IN1150-forkurs

Lær deg å bruke LaTeX i Overleaf for å lage fine innleveringer

Luna Artemis Midttun Hukkelberg
arthuk@uio.no

14. januar 2026



LaTeX?

- LaTeX er en utvidelse av TeX
- TeX kommer av det greske $\tau\varepsilon\chi\nu\eta$
- X'en uttales altså med en kj-lyd

Hva er LaTeX?

- LaTeX er en form for programmeringsspråk, men det er ganske forskjellig fra f.eks. Python

Hva er LaTeX?

- LaTeX er en form for programmeringsspråk, men det er ganske forskjellig fra f.eks. Python
- I Python skriver du kode som kjøres for å utføre handlinger (veldig fleksibelt!)
- I LaTeX skriver du kode som brukes til å formatere/lage dokumenter (mer spesialisert)

Hva er LaTeX?

- LaTeX er en form for programmeringsspråk, men det er ganske forskjellig fra f.eks. Python
- I Python skriver du kode som kjøres for å utføre handlinger (veldig fleksibelt!)
- I LaTeX skriver du kode som brukes til å formatere/lage dokumenter (mer spesialisert)
- Dette kalles også et *markup*-språk
- I denne kategorien finner du også HTML og MarkDown

LaTeX' syntaks

- I LaTeX kan du skrive tekst som vanlig, og det vil dukke opp mer eller mindre slik du skrev det
  - En forskjell er at alle linjeskift og mellomrom mellom ord gjøres om til ett enkelt mellomrom i resultatet
  - Dette er blant annet for at du skal kunne skrive lange avsnitt uten å måtte stappe alt på én linje i kildekoden

LaTeX' syntaks

- I LaTeX kan du skrive tekst som vanlig, og det vil dukke opp mer eller mindre slik du skrev det
  - En forskjell er at alle linjeskift og mellomrom mellom ord gjøres om til ett enkelt mellomrom i resultatet
  - Dette er blant annet for at du skal kunne skrive lange avsnitt uten å måtte stappe alt på én linje i kildekoden
- For eksempel, hvis man skriver følgende:

```
Dette er en setning
som står på to linjer.
```

- Blir det til:
- Dette er en setning som står på to linjer.

Kommandoer

- Kommandoer lar deg sette inn spesielle symboler eller lage komplekse uttrykk
- En kommando består av \, etterfulgt av navnet på kommandoen, og eventuelt argumenter i {}.
- For eksempel:
  - `\textbf{tekst}` blir til **tekst** (ett argument)
  - `\cdots` blir til $\cdots$ (ingen argumenter)
  - `\frac{1 + 3}{64}` blir til $\frac{1+3}{64}$ (to argumenter)

Math mode

- Math mode brukes til å skrive matematiske uttrykk
- Brukes ved å putte `$` på hver side av et uttrykk
- For å sentrere som egen blokk, brukes `\[` og `\]` på hver side av uttrykket i stedet
- For eksempel:
  - `$5 + 3$` blir til $5 + 3$
  - `\[ \frac{50}{100 + 200} \]` blir til

$$\frac{50}{100 + 200}$$

Kommandoer i math mode

- Det er mange kommandoer som ikke fungerer utenfor math mode, blant annet `\frac{}{}` og `\cdots`
- Noen kommandoer fungerer utenfor, men kan ende opp med rare mellomrom, som for eksempel `\lor` ("logical or"):
  - `(A \lor B)` blir til (A ∨B)
  - `$(A \lor B)$` blir til $(A \lor B)$

Kommandoer i math mode

- Det er mange kommandoer som ikke fungerer utenfor math mode, blant annet `\frac{}{}` og `\cdots`
- Noen kommandoer fungerer utenfor, men kan ende opp med rare mellomrom, som for eksempel `\lor` ("logical or"):
  - `(A \lor B)` blir til (A ∨B)
  - `$(A \lor B)$` blir til $(A \lor B)$
- **Hovedregel: Bruk math mode når du skal skrive noe som ikke er setninger/vanlig ren tekst**

Krøllparenteser, mellomrom og linjeskift

- Siden krøllparenteser har en spesiell betyding i LaTeX (brukes f.eks. rundt argumenter til kommandoer, som vi har sett), så må man "escape" de med en \
- For å skrive { og } må man altså bruke \{ og \}
- For eksempel, $\{1, 2, 3\}$ skrives som `$ \{ 1, 2, 3 \} $`

Krøllparenteser, mellomrom og linjeskift

- Siden krøllparenteser har en spesiell betyding i LaTeX (brukes f.eks. rundt argumenter til kommandoer, som vi har sett), så må man "escape" de med en \
- For å skrive { og } må man altså bruke \{ og \}
- For eksempel, $\{1, 2, 3\}$ skrives som `$ \{ 1, 2, 3 \} $`
  - Kan også skrives uten mellomrom: `$\{1,2,3\}$`

Krøllparenteser, mellomrom og linjeskift

- Siden krøllparenteser har en spesiell betyding i LaTeX (brukes f.eks. rundt argumenter til kommandoer, som vi har sett), så må man "escape" de med en \
- For å skrive { og } må man altså bruke \{ og \}
- For eksempel, $\{1, 2, 3\}$ skrives som `$ \{ 1, 2, 3 \} $`
  - Kan også skrives uten mellomrom: `$\{1,2,3\}$`
- Andre parenteser som ( ) og [ ] kan brukes vanlig

Krøllparenteser, mellomrom og linjeskift

- Siden krøllparenteser har en spesiell betyding i LaTeX (brukes f.eks. rundt argumenter til kommandoer, som vi har sett), så må man "escape" de med en \
- For å skrive { og } må man altså bruke \{ og \}
- For eksempel, $\{1, 2, 3\}$ skrives som `$ \{ 1, 2, 3 \} $`
  - Kan også skrives uten mellomrom: `$\{1,2,3\}$`
- Andre parenteser som ( ) og [ ] kan brukes vanlig

- Siden LaTeX ikke bryr seg så mye om mellomrom og linjeskift, er det noen ganger vi har lyst til å sette det inn manuelt
- \\ setter inn et linjeskift
- \ (backslash + mellomrom) setter inn et ekstra mellomrom

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)
- Denne presentasjonen er laget med `beamer`-klassen

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)
- Denne presentasjonen er laget med `beamer`-klassen
- I IN1150 har vi en `innlevering`-klasse

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)
- Denne presentasjonen er laget med `beamer`-klassen
- I IN1150 har vi en `innlevering`-klasse
- Aller først i dokumentet må man ha en linje som sier hvilken klasse man bruker: `\documentclass{innlevering}`

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)
- Denne presentasjonen er laget med `beamer`-klassen
- I IN1150 har vi en `innlevering`-klasse
- Aller først i dokumentet må man ha en linje som sier hvilken klasse man bruker: `\documentclass{innlevering}`
- Etter det har man eventuelt kommandoer for å sette opp dokumentet, blant annet for å importere pakker (vi ser et eksempel på det senere)

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)
- Denne presentasjonen er laget med `beamer`-klassen
- I IN1150 har vi en `innlevering`-klasse
- Aller først i dokumentet må man ha en linje som sier hvilken klasse man bruker: `\documentclass{innlevering}`
- Etter det har man eventuelt kommandoer for å sette opp dokumentet, blant annet for å importere pakker (vi ser et eksempel på det senere)
- Og så har vi `\begin{document}`, og etter denne kommer all teksten som skal i dokumentet

Strukturen til et LaTeX-dokument

- Alle LaTeX-dokumenter har en "klasse", som kan påvirke hvordan dokumentet ser ut (dette er ikke det samme som en klasse i Python eller Java)
- Denne presentasjonen er laget med `beamer`-klassen
- I IN1150 har vi en `innlevering`-klasse
- Aller først i dokumentet må man ha en linje som sier hvilken klasse man bruker: `\documentclass{innlevering}`
- Etter det har man eventuelt kommandoer for å sette opp dokumentet, blant annet for å importere pakker (vi ser et eksempel på det senere)
- Og så har vi `\begin{document}`, og etter denne kommer all teksten som skal i dokumentet
- Og helt til slutt har vi `\end{document}`

Eksempel på et dokument

```
\documentclass{innlevering}

% Pakke for å håndtere bilder
\usepackage{graphicx}
\graphicspath{ {./images/} }

\brukernavn{arthuk}

\begin{document}
    ...
\end{document}
```

Omgivelser

- På forrige slide så vi `\begin{document}` og `\end{document}`

Omgivelser

- På forrige slide så vi `\begin{document}` og `\end{document}`
- Men dokumenter er ikke det eneste som kan begynne og slutte

Omgivelser

- På forrige slide så vi `\begin{document}` og `\end{document}`
- Men dokumenter er ikke det eneste som kan begynne og slutte
- For å lage en liste, kan man bruke `\begin{itemize}`, og `\item` for hvert listepunkt

```
\begin{itemize}
    \item Dette er noen
    \item punkter i
    \item en liste
\end{itemize}
```

- Dette er noen
- punkter i
- en liste

Oppgaver og deloppgaver

- I `innlevering`-klassen finnes det omgivelser for oppgaver og deloppgaver
- Disse er allerede fylt ut med riktig struktur i malen dere får
- Eksempelbruk:

```
\kapittel{1}
\begin{oppgave}{1}
    \begin{deloppgaver}
        \item Svar på oppgave 1.1 (a)
        \item Svar på oppgave 1.1 (b)
        \item ...og så videre
    \end{deloppgaver}
\end{oppgave}
\begin{oppgave}{2}
    Svar på oppgave 1.2 (ingen deloppgaver)
\end{oppgave}
```

Tabeller

- I IN1150 bruker vi sannhetsverditabeller, som ser slik ut:

P	Q	(P	$\wedge$	(P	$\vee$	Q))
0	0	0	**0**	0	0	0
0	1	0	**0**	0	1	1
1	0	1	**1**	1	1	0
1	1	1	**1**	1	1	1

Skrive tabeller

- For å lage en tabell, bruker man `table`-omgivelsen:

```
\begin{table}[h!]
    ...
\end{table}
```

  - `[h!]` betyr "plasser tabellen her" (i stedet for f.eks. nederst på siden)

- Inni `table`-omgivelsen starter vi en `tabular`-omgivelse:

```
\begin{tabular}{ c c|c c c c c }
    ...
\end{tabluar}
```

  - Argumentet `{ c c|c c c c c }` definerer kolonnene i tabellen
  - Her har vi to kolonner, så en skillevegg, og så fem til kolonner

Skrive tabeller

- Og så kommer innholdet i tabellen. & brukes til å skille kolonner innad i hver linje, og linjen avsluttes med `\\`
- Tabellen vår har 7 kolonner, så hver linje har 6 &
- Det går ikke an å bruke math mode på tvers av &
- Første linje i tabellen kan da skrives slik:
  `$P$ & $Q$ & $(P$ & $\land$ & $(P$ & $\lor$ & $Q))$ \\`
  - Merk at hver kolonne bruker math mode ($ rundt), og har & mellom seg
- For å få en horisontal skillevegg, kan man skrive `\hline`

Hele tabellen skrives da slik:

```
\begin{table}[h!]
    \begin{tabular}{ c c|c c c c c }
        $P$ & $Q$ & $(P$ & $\land$ & $(P$ & $\lor$ & $Q))$ \\
        \hline
        0 & 0  &  0 & \textbf{0} & 0 & 0 & 0 \\
        0 & 1  &  0 & \textbf{0} & 0 & 1 & 1 \\
        1 & 0  &  1 & \textbf{1} & 1 & 1 & 0 \\
        1 & 1  &  1 & \textbf{1} & 1 & 1 & 1 \\
    \end{tabular}
\end{table}
```

Og her er resultatet igjen:

P	Q	(P	$\wedge$	(P	$\vee$	Q))
0	0	0	**0**	0	0	0
0	1	0	**0**	0	1	1
1	0	1	**1**	1	1	0
1	1	1	**1**	1	1	1

Bilder

- LaTeX kan gjøre alt vi trenger til dette emnet (pensumboka og alle innleveringsoppgavene er skrevet i LaTeX!), men i noen tilfeller er det enklere å tegne/skrive for hånd

Bilder

- LaTeX kan gjøre alt vi trenger til dette emnet (pensumboka og alle innleveringsoppgavene er skrevet i LaTeX!), men i noen tilfeller er det enklere å tegne/skrive for hånd
- Heldigvis lar LaTeX oss også sette inn bilder i dokumentet vårt

Bilder

- LaTeX kan gjøre alt vi trenger til dette emnet (pensumboka og alle innleveringsoppgavene er skrevet i LaTeX!), men i noen tilfeller er det enklere å tegne/skrive for hånd
- Heldigvis lar LaTeX oss også sette inn bilder i dokumentet vårt
- Vi må bruke en ekstra pakke, som gjøres ved å skrive `\usepackage{graphicx}` på starten av filen, mellom `\documentclass{...}` og `\begin{document}`
- Her må vi også si hvor bildene våre finnes: `\graphicspath{ {./images/} }`
  - Dette forteller at bildene våre ligger i `images`-mappa, som selv ligger i samme mappe som LaTeX-filen vi redigerer

Bilder

- LaTeX kan gjøre alt vi trenger til dette emnet (pensumboka og alle innleveringsoppgavene er skrevet i LaTeX!), men i noen tilfeller er det enklere å tegne/skrive for hånd
- Heldigvis lar LaTeX oss også sette inn bilder i dokumentet vårt
- Vi må bruke en ekstra pakke, som gjøres ved å skrive `\usepackage{graphicx}` på starten av filen, mellom `\documentclass{...}` og `\begin{document}`
- Her må vi også si hvor bildene våre finnes: `\graphicspath{ {./images/} }`
  - Dette forteller at bildene våre ligger i `images`-mappa, som selv ligger i samme mappe som LaTeX-filen vi redigerer
- Da er det klart for å sette inn bilder: `\includegraphics{et-bilde.jpeg}`
  - Bytt ut `et-bilde.jpeg` med navn+filendelse på bildet innenfor mappen du skrev over

Bilde :)

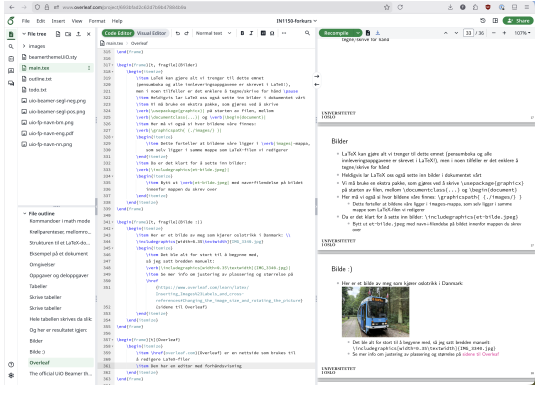
- Her er et bilde av meg som kjører oslotrikk i Danmark:



  - Det ble alt for stort til å begynne med, så jeg satt bredden manuelt:
    `\includegraphics[width=0.35\textwidth]{IMG_3340.jpg}`
  - Se mer info om justering av plassering og størrelse på sidene til Overleaf

Overleaf

- Overleaf er en nettside som brukes til å redigere LaTeX-filer
- Den har en editor med forhåndsvisning:



Overleaf

- Malen for innleveringsoppgavene er tilgjengelig som et overleaf-prosjekt som dere kan duplisere og begynne å jobbe på
- Når innleveringen er ferdig, kan man laste ned resultatet som pdf, som kan leveres i devilry