



**AALBORG UNIVERSITET**

P1-PROJEKT  
MATEMATIK-ØKONOMI

# **L<sup>A</sup>T<sub>E</sub>X Skabelon**

**Kom Godt I Gang Med Projektskrivning**

*Forfattere:*

Janus S. Valberg-Madsen  
Janus S. Valberg-Madsen

*Vejledere:*

Janus S. Valberg-Madsen

10. oktober 2018



Institut for Matematiske Fag  
Skjernvej 4A  
DK-9220 Aalborg Ø  
<http://math.aau.dk>

## AALBORG UNIVERSITET

### STUDENTERRAPPORT

**Titel:**

L<sup>A</sup>T<sub>E</sub>X Skabelon

**Tema:**

Projektskrivning

**Projektperiode:**

Efterårssemesteret 2018

**Projektgruppe:**

Gruppe XYZ

**Deltager(e):**

Janus S. Valberg-Madsen

Janus S. Valberg-Madsen

**Vejleder(e):**

Janus S. Valberg-Madsen

**Oplagstal:** 0**Sidetæl:** 19**Afleveringsdato:**

10. oktober 2018

**Abstract:**

Dette dokument fungerer som en L<sup>A</sup>T<sub>E</sub>X skabelon for førsteårsstuderende på Aalborg Universitet, Institut for Matematiske Fag. Eksempler på almindelige kommandoer og environments er inkluderet for at vise, hvordan de bruges.

Skabelonen kan bruges som den er, så studerende udelukkende skal fokusere på selve indholdet, men interesserede studerende opfordres til at bruge skabelonen som inspiration og til at modificere den til deres behov.

# Indhold

|   |           |
|---|-----------|
| <b>1 Første Eksempel</b>                                  | <b>1</b>  |
| 1.1 Environments . . . . .                                | 1         |
| 1.1.1 Lister . . . . .                                    | 2         |
| 1.1.2 Ligninger . . . . .                                 | 2         |
| 1.1.3 Figurer . . . . .                                   | 3         |
| 1.1.4 Tabeller . . . . .                                  | 5         |
| <b>2 Andet Eksempel</b>                                   | <b>7</b>  |
| 2.1 Brugerdefinerede Environments og Kommandoer . . . . . | 7         |
| 2.1.1 Definition, Sætning, Bevis . . . . .                | 7         |
| 2.2 Kildehenvisninger . . . . .                           | 8         |
| <b>Appendicer</b>   | <b>9</b>  |
| <b>A Pseudokode</b>                                       | <b>11</b> |
| <b>B Python Kode</b>                                      | <b>13</b> |



# 1 | Første Eksempel

Dette er et eksempel på et kapital med indhold. Udover `\chapter` findes der adskillige kommandoer for at niveauinddele ens tekst:

- `part`
- `chapter`
- `section`
- `subsection`
- `subsubsection`
- `paragraph`
- `subparagraph`

Hvert niveau er en underafsnit af niveauet over det. Overskrifter tilføjes automatisk til indholdsfortegnelsen. Se mere på [https://en.wikibooks.org/wiki/LaTeX/Document\\_Structure#Sectioning\\_commands](https://en.wikibooks.org/wiki/LaTeX/Document_Structure#Sectioning_commands).

## 1.1 Environments

I L<sup>A</sup>T<sub>E</sub>X vil I komme til at skulle bruge mange forskellige slags **environments**. De åbnes med `\begin{...}` og lukkes med `\end{...}` og har specifikke formål, såsom at lave lister, figurer, ligninger, osv. I Tabel 1.1 er nogle af de mest almindelige environments.

| Environment              | Formål                    |
|--------------------------|---------------------------|
| <code>document</code>    | Selve dokumentets indhold |
| <code>table</code>       | Tabeller ligesom denne    |
| <code>figure</code>      | Figurer                   |
| <code>equation</code>    | Nummererede ligninger     |
| <code>align</code>       | Justerede ligninger       |
| <code>itemize</code>     | Punktliste                |
| <code>enumerate</code>   | Nummereret liste          |
| <code>description</code> | Liste med beskrivelser    |

Tabel 1.1: Almindelige L<sup>A</sup>T<sub>E</sub>X environments og deres formål

### 1.1.1 Lister

Der er tre essentielle slags lister: `itemize`, `enumerate`, and `description`. Varianten `itemize` producerer en simpel punktliste. Hvert punkt angives med kommandoen `\item`.

```
\begin{itemize}
  \item Første punkt
  \item Andet punkt
  \item Tredje punkt
\end{itemize}
```

- Første punkt
- Andet punkt
- Tredje punkt

Varianten `enumerate` bruger samme syntaks som `itemize`, men producerer en nummereret liste.

```
\begin{enumerate}
  \item Første punkt
  \item Andet punkt
  \item Tredje punkt
\end{enumerate}
```

1. Første punkt
2. Andet punkt
3. Tredje punkt

I `description`-listen tager `\item` kommandoerne navnet på det, som beskrives, som argument, og resten af linjen udgør selve beskrivelsen. Dette producerer en liste, hvor navnene er skrevet med fed, og deres beskrivelser er normal tekst.

```
\begin{description}
  \item[Første ting] Beskrivelse af den første ting
  \item[Anden ting] Beskrivelse af den anden ting
  \item[Tredje ting] Beskrivelse af den tredje ting
\end{description}
```

**Første ting** Beskrivelse af den første ting

**Anden ting** Beskrivelse af den anden ting

**Tredje ting** Beskrivelse af den tredje ting

### 1.1.2 Ligninger

En af hovedårsagerne bag, hvorfor folk bruger  $\text{\LaTeX}$ , er hvor godt matematiske formler ser ud. Der er mange forskellige matematikrelaterede environments til forskellige formål, og de fleste har både nummererede og unummererede varianter. For eksempel, hvis man skriver

```
\begin{equation}
\label{eq:1}
e^{i\pi} - 1 = 0 ,
\end{equation}
```

så får man følgende output:

$$e^{i\pi} + 1 = 0, \tag{1.1}$$

og fordi den har en label, så kan man referere til den med kommandoen `\eqref{eq:1}`, hvilket producerer en klikbar reference i paranteser, (1.1). Hvis man i stedet for `equation` bruger `equation*`, så får ligninger *ikke* et nummer. Alternativt kan man bruge `\[...\]`, så hvis man skriver `\[ e^{i\pi} + 1 = 0 \]`, så får man

$$e^{i\pi} + 1 = 0.$$

Hvis man skal have flere ligninger i træk, f.eks. til trin-for-trin udregninger, så brug `align`, som får ligninger til at justere sig omkring `&` symboler. For eksempel, hvis man skriver

```
\begin{align*}
(x + y)^2 &= x^2 + xy + yx + y^2 \\
&= x^2 + y^2 + 2xy
\end{align*}
```

så vil = i de to ligninger stå lige over hinanden;

$$\begin{aligned} (x + y)^2 &= x^2 + xy + yx + y^2 \\ &= x^2 + y^2 + 2xy. \end{aligned}$$

Dobbelt backslash indsætter et linjeskift. Bemærk stjernen; ligesom med `equation` så har `align` både en nummereret og unummereret udgave. Den nummererede udgave har et separat nummer for *hver* linje.

For at skrive matematik i brødteksten, f.eks.  $\cos^2 \theta + \sin^2 \theta = 1$ , brug `\(...\)`. Alternativt kan man bruge `$...$`, men `\(...\)` har bedre mellemrum og fejlmeddelelser. Se <https://en.wikibooks.org/wiki/LaTeX/Mathematics> for en god oversigt over kommandoer og symboler.

### 1.1.3 Figurer

Figurer i  $\text{\LaTeX}$  indsættes som såkaldte *floats* ved brug af `figure` environmentet. Et “flydende” objekt kan ikke brydes af et sideskift, så en figur vil flyttes, hvis der ikke er plads nok tilbage på siden. Syntaksen er som følger:

```
\begin{figure}[placement]
\centering
\includegraphics[options]{path/to/image}
\caption{Billedteksten}
\label{fig:label}
\end{figure}
```

Argumentet `placement` kan enten være `h` (her), `t` (toppen af siden), `b` (bunden af siden), eller `p` (på en særlig side kun for floats). Kommandoen `\centering` centrerer billedet på siden. Iblandt de forskellige `options`, man kan give til `\includegraphics`, så er det vigtigste

nok `width`. For at få billedet til at fylde halvdelen af siden (margener ikke inkluderet), så brug `width=0.5\textwidth`. Alle figurer bør have en billedtekst, hvilket sættes med kommandoen `\caption`, og `\label` gør det muligt at referere til den (for eksempel bliver `Figur \ref{fig:me}` til Figur 1.1). For mere info, se [https://en.wikibooks.org/wiki/LaTeX/Floats,\\_Figures\\_and\\_Captions](https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions).

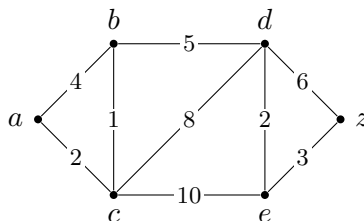


Figur 1.1: Et billede af mig, der svarer på emails fra mine studerende

Figurer behøver ikke nødvendigvis at bestå af billedfiler som JPG, PNG eller PDF, men kan også bestå af  $\text{\LaTeX}$ -kode. Et vigtigt eksempel på dette er *TikZ*, som bruges til at tegne vektorgrafik med  $\text{\LaTeX}$ -kommandoer. Koden til et TikZ-billede indsættes i environmentet `tikzpicture`, som indsættes i et `figure` environment, ligesom billeder:

```
\begin{figure}[placement]
  \centering
  \begin{tikzpicture}
    % indsæt tikz kode her...
  \end{tikzpicture}
  \caption{Billedtekst}
  \label{fig:label}
\end{figure}
```

Et eksempel på en graf tegnet med TikZ-kode er vist på Figur 1.2.



Figur 1.2: Graf fra [Rosen, 2013].

For at lære syntaksen, se [Cr  mer, 2011], tutorial-afsnittene af [Tantau, 2015], og <https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>.



### 1.1.4 Tabeller

Ligesom med figurer så indsættes tabeller som floats, og dette gøres med `table` environmentet. Selve indholdet af tabellen indsættes i et `tabular` environment, hvor søjlerne separeres af `&`, og rækkerne separeres af `\\`:

```
\begin{table}[placement]
  \begin{tabular}{columns}
    % indsæt indholdet af tabellen her...
  \end{tabular}
  \caption{Beskrivelse af tabellen}
  \label{tab:label}
\end{table}
```

Argumentet `placement` er det samme som for figurer. Argumentet `columns` er en specifikation af søjlerne og består i sin mest basale form af bogstaverne `l`, `c`, og/eller `r`, muligvis med `|` indsæt i de mellemrum, hvor der skal være lodrette streger. For eksempel ville `\begin{tabular}{r|cc}` fortælle L<sup>A</sup>T<sub>E</sub>X, at tabellen har tre søjler, hvoraf den første er højrejusteret (`r`) og efterfølges af en lodret streg, og de to næste søjler er centrerede uden en streg imellem dem.

For et fuldstændigt eksempel på en tabel, se kildekoden for Tabel 1.1 (kan findes i `fig/tab/my-table.tex`). For mere information og mere avancerede options, se <https://en.wikibooks.org/wiki/LaTeX/Tables>.



## 2 | Andet Eksempel

Her er endnu et eksempel på en inputfil.

### 2.1 Brugerdefinerede Environments og Kommandoer

Selvom L<sup>A</sup>T<sub>E</sub>X har mange nyttige kommandoer indbygget, så vil I ofte have brug for at definere jeres egne. I denne skabelon har jeg inkluderet nogle eksempler på brugerdefinerede environments og kommandoer i preamblet (`preamble.tex`).

Sådanne kommandoer kan spare jer for at skulle gentage jer selv for meget, når I skriver lange dokumenter såsom et semesterprojekt. For eksempel kan man, i stedet for at skrive `\mathbb{N}` hver gang, man skal bruge symbolet for de naturlige tal, definere en kortere kommando som f.eks. `\N`. Syntaksen er som følger:

```
\newcommand{name}[num]{definition}
```

hvor **name** er kommandoens navn, f.eks. `\N`, **num** er antallet af argumenter, kommandoen tager (udelad firkantklammerne, hvis kommandoen ingen argumenter tager), og **definition** er outputtet af kommandoen, f.eks. `\mathbb{N}`.

#### 2.1.1 Definition, Sætning, Bevis

I et matematikprojekt får man brug for at skrive matematiske definitioner, propositioner, lemmaer, sætninger, osv. Pakken `amsthm` giver en simpel måde at definere environments til formålet:

```
\newtheorem{name}{Printed output}[numberby]
```

Et par eksempler er inkluderet i preamblet. Se <https://en.wikibooks.org/wiki/LaTeX/Theorems> for mere information.

**Definition 2.1** (Rationelle tal). A reelt tal  $r$  kaldes *rationelt*, hvis der eksisterer heltal  $p$  og  $q$  med  $q \neq 0$  sådan at  $r = p/q$  (et reelt tal, som ikke er rationelt kaldes *irrationelt*). Mængden af rationelle tal skrives som  $\mathbb{Q}$ .

**Sætning 2.2** (Eksempel på Navngivet Sætning).  $\sqrt{2}$  er *irrationel*.

*Bevis.* Antag  $\sqrt{2} \in \mathbb{Q}$  og lad  $k = \min\{N \in \mathbb{N} : N\sqrt{2} \in \mathbb{N}\}$ . Men så har vi, at  $k(\sqrt{2} - 1) = (k\sqrt{2} - k) \in \mathbb{N}$ , og  $k(\sqrt{2} - 1) < k$ , hvilket er en modstrid. Dermed har vi, at  $\sqrt{2} \notin \mathbb{Q}$ .  $\square$

## 2.2 Kildehenvisninger

Når I bruger andre folks værker, så skal I henvise til dem. Henvisninger i L<sup>A</sup>T<sub>E</sub>X håndteres normal med BibTeX, et program, der identificerer ressourcer fra en bibliografi-fil, I skriver. Literaturhenvisninger skal defineres i en `.bib`-fil og inkluderes med kommandoen `\bibliography`. Hver ressource i denne fil skal have en type og et ID, som bruges til at indsætte henvisninger til den. Syntaksen for at indsætte en henvisning er

```
\citep[scope]{bibkey}
```

hvor `bibkey` er det ID, I giver en ressource i `.bib`-filen, og `scope` kan bruges til at henvise til specifikke afsnit eller sidetal.

Når I henviser til en ressource, bliver den automatisk tilføjet til literaturlisten, og i PDF-filen bliver henvisninger i teksten til klikbare links til denne liste. For eksempel, ved at henvise til [Edwards and Penney, 2014, s. 104-110], tilføjes denne til listen på sidste side.

Der er mange forskellige typer af ressourcer i BibTeX, og hver type har sit eget sæt af felter, som skal eller kan udfyldes. For mere information, se [https://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management#BibTeX](https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#BibTeX).

# Appendicer



# A | Pseudokode

For at indsætte pseudokode i ens dokument bruges `algorithmic` environmentet, som placeres inde i et `algorithm` environment for at få en figur-agtig float:

```
\begin{algorithm}
  \begin{algorithmic}
    % pseudokode her...
  \end{algorithmic}
  \caption{algoritmens titel}
  \label{alg:label}
\end{algorithm}
```

---

## Algoritme 1 Bubble Sort

---

```
procedure BUBBLESORT( $a_1, \dots, a_n$  : reelle tal med  $n \geq 2$ )
  for  $i := 1$  til  $n - 1$  do
    for  $j := 1$  til  $n - i$  do
      if  $a_j > a_{j+1}$  then ombyt  $a_j$  og  $a_{j+1}$ 
    end for
  end for
 $\triangleright a_1, \dots, a_n$  er nu i voksende orden
```

---

Se [https://en.wikibooks.org/wiki/LaTeX/Algorithms#Typesetting\\_using\\_the\\_algorithmicx\\_package](https://en.wikibooks.org/wiki/LaTeX/Algorithms#Typesetting_using_the_algorithmicx_package) for en oversigt over syntaksen.





## B | Python Kode

Hvis I implementerer algoritmer i et specifikt programmeringssprog, vil I muligvis inkludere kildekoden i et appendiks som dette. Pakken `listings` gør det muligt at inkludere kildekode med syntax highlighting for mange forskellige sprog.

Med environmentet `lstlisting` kan man indsætte kode direkte, men for det meste vil I typisk bruge `\lstinputlisting` til at indsætte kode fra en fil, f.eks.

```
\lstinputlisting[options]{path/to/source_file}
```

Nogle nyttige `options` for denne kommando er:

**caption** sætter caption for koden

**label** sætter label for koden (til `\ref`)

**language** sætter syntaksen til et bestemt sprog lokalt

**style** sætter highlighter stilen lokalt

**firstline** springer linjerne før den angivende linje over

**lastline** stopper med at læse linjer, når den angivende linje nåes

I preamblet kan man også sætte globale options med `\lstset`. For eksempel, hvis al ens kildekode er skrevet i Python, kan man skrive `\lstset{language=Python}`, og så behøver man ikke bruge sætte `language` i hver enkel `\lstinputlisting`.

```
def bubblesort(a):
    for i in range(len(a)-2):
        for j in range(len(a)-i-1):
            if a[j] > a[j+1]:
                tmp = a[j]
                a[j] = a[j+1]
                a[j+1] = tmp

ex = [54, 26, 93, 17, 77, 31, 44, 55, 20]
bubblesort(ex)
print ex
```

Script B.1: Bubble Sort in Python

For mere information, se [https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings).



# Figurer

|     |   |   |
|-----|---|---|
| 1.1 | Et billede af mig, der svarer på emails fra mine studerende . . . . . | 4 |
| 1.2 | Graf fra [Rosen, 2013]. . . . .                                       | 4 |



# Tabeller

|     |  |   |
|-----|--|---|
| 1.1 | Almindelige $\text{\LaTeX}$ environments og deres formål . . . . . | 1 |
|-----|--|---|



# Litteratur

- Crémer, J. (2011). *A very minimal introduction to TikZ*. Toulouse School of Economics. <https://cremeronline.com/LaTeX/minimaltikz.pdf>.
- Edwards, C. H. and Penney, D. E. (2014). *Calculus: Early Transcendentals*. Pearson, 7th edition.
- Rosen, K. H. (2013). *Discrete Mathematics and Its Applications*. McGraw-Hill, 7th edition.
- Tantau, T. (2015). *The TikZ and PGF packages*. Institut für Theoretische Informatik, Universität zu Lübeck. <http://mirrors.dotsrc.org/ctan/graphics/pgf/base/doc/pgfmanual.pdf>.