

L^AT_EX-workshop (Handleiding)

De Leidsche Flesch

8 september 2021

1 Inleiding

Dit is de handleiding horende bij de L^AT_EX-workshop van De Leidsche Flesch. Kijk allereerst deze handleiding door. Ga vervolgens aan de slag met de opdrachten van het werkblad. Het is de bedoeling dat je bij het maken van deze opdrachten dingen opzoekt in deze handleiding en op internet.

Je zult waarschijnlijk een aantal keer iets fout doen, omdat je *ergens in je code* iets net verkeerd doet. Dit is bewust de opzet van deze workshop, omdat je later bij het gebruik van L^AT_EX hier gegarandeerd mee te maken gaat krijgen en het dus belangrijk is om dit op te kunnen lossen. Onthoud verder ook dat er meestal meerdere manieren zijn om iets weer te geven, met telkens net een iets andere lay-out. Het is vaak een kwestie van wat je fijner vindt.

Mocht het maken van documenten in L^AT_EX je moeilijk afgaan, probeer dan gebruik te maken van online hulpmiddelen. De wikibook over L^AT_EX, te vinden op <https://en.wikibooks.org/wiki/LaTeX>, is een zeer toegankelijk hulpmiddel waar de meest voorkomende situaties uitgebreid in beschreven staan. Een ander nuttig hulpmiddel is de informatiepagina van de online editor Overleaf, te vinden op <https://www.overleaf.com/>.

Mocht je het commando van een bepaald symbool niet kunnen vinden, probeer dan <http://detexify.kirelabs.org>, waar je het symbool dat je zoek kan tekenen, en het bijbehorende commando dan voor je gevonden wordt¹.

1.1 Beginnen

Voor deze workshop raden we aan de online editor <https://www.overleaf.com/> te gebruiken, waarmee in in the cloud kan werken. Het werkt in principe zoals Google Docs, aangezien je op elke computer er bij kan en alle bestanden die je maakt ook online opgeslagen worden. Zo kan je ook nooit je huiswerk meer vergeten. Ook kun je met meerdere personen aan hetzelfde document werken.

Daarnaast zijn er verschillende offline editors, zoals MiKTeX en TeXmaker. Op de unicompsters is MiKTeX te vinden via startmenu → all programs → MiKTeX 2.9 → TeXworks. Je kunt op je eigen computer MiKTeX downloaden via <http://miktex.org/download>.

¹Voor android is er ook een app beschikbaar: <https://play.google.com/store/apps/details?id=website.marty.detexify&hl=nl>.

2 Nieuw document

Een \LaTeX document bestaat uit twee delen. Het eerste deel is de preamble. De dingen die je hier neerzet gelden voor het hele document, hier kun je bijvoorbeeld de juiste pakketten laden en het lettertype instellen. Het tweede deel begint bij `\begin{document}` en eindigt bij `\end{document}`. Hier tussenin staat de inhoud van het document.

Je kunt ook commentaar in je `.tex` bestand zetten, zonder dat dit wordt weergegeven in je PDF. Doe dit door `%` voor je regel met commentaar te zetten.

In Overleaf geeft een nieuw document al een opzet voor je bestand. Van de onderstaande commando's wordt er al een aantal in het bestand gezet.

Het Werkblad dat bij de workshop hoort bevat alle commando's uit dit hoofdstuk al. Indien je dat gebruikt, hoef je dus de commando's die hieronder staan niet meer zelf in te voegen. Zorg wel dat je dit hoofdstuk doorleest, zodat je ongeveer weet wat alles doet.

2.1 Preamble

1. Zet bovenaan `\documentclass{article}`. Dit commando geeft aan wat voor soort tekst je gaat schrijven, in dit geval dus `article`. Daarnaast bepaalt het ook de lettergrootte. Een `article` is het beste voor bijna elke situatie.
2. Direct daaronder zet je de packages die je wilt gebruiken. Deze maken het mogelijk om een aantal extra functies toe te voegen aan \LaTeX die er niet standaard in zitten. Voeg een package toe aan je document door het commando `\usepackage{packagenaam}` bovenaan het document te typen.

Een aantal commands dat vaak in bestanden wordt gebruikt zijn:

- 1) `\usepackage[a4paper]{geometry}` Deze zorgt dat de afmetingen van je bestand op a4 grootte zijn.
- 2) `\usepackage[english, dutch]{babel}` Deze zorgt voor een goede regelafbreking en goede vertalingen. Hierbij moet je denken aan 'Hoofdstuk' in plaats van 'Chapter'. Hier wordt ook Engels geladen, omdat je bijvoorbeeld Engelse citaten kunt gebruiken. De laatst genoemde taal tussen de blokhaken is de hoofdtaal.
- 3) `\usepackage{parskip}` Deze zorgt voor een mooie weergave van alinea's.
- 4) `\usepackage{amsmath, amssymb}` Dit zijn pakketten die ervoor zorgen dat formules en andere wiskundige dingen mooier worden weergegeven.
- 5) `\usepackage{graphicx}` Dit is een pakket om losse plaatjes toe te voegen.
- 6) `\usepackage{enumerate}` Hiermee kan je makkelijk genummerde lijsten maken.
- 7) `\usepackage{url}` Dit zorgt ervoor dat urls leesbaar worden weergegeven.
- 8) `\usepackage{float}` Dit gebruik je om plaatjes op een goede manier in te kunnen voegen.
- 9) `\usepackage[hidelinks]{hyperref}` Om een inhoudsopgave te maken waarbij je als je klikt op het punt waar je heen wilt, er vanzelf heen springt.
- 10) `\usepackage{color}` Om tekst in **kleur** te kunnen typen.

3. Zet in je preamble de volgende commands:

```
\title{titel}
\author{A. Uthor}
\date{\today}, het commando \today voegt automatisch de goede datum toe.
```

4. Als laatste zet je `\begin{document}` en `\end{document}` neer. Tussen deze commando's zet je de tekst in je document.

5. Compileer het document

2.2 Tekst in document

2.2.1 Titel en auteur

Zet bovenin je document `\maketitle`. Dit zorgt ervoor dat de gegevens die je in de preamble hebt ingevuld als titel op de goede plek in je document komen te staan. Dus er komt in dit geval een titel, auteur en datum te staan.

2.2.2 Inhoudsopgave

Een inhoudsopgave maak je simpel door het commando `\tableofcontents` te typen. Daarna wil je beginnen met een nieuwe pagina, dus daaronder type je `\newpage`.

2.2.3 Kopjes

Een article document in \LaTeX maakt gebruik van sections, subsections, subsubsections en paragrafen. Zo is bijvoorbeeld in deze handleiding het kopje 'Nieuw document' een section. Deze maak je door te typen `\section{Naam kopje}`.

Het kopje 'Tekst in document' is een subsection. Deze maak je door te typen `\subsection{Naam kopje}`.

Het kopje 'Kopjes' is een subsubsection. Deze maak je door te typen `\subsubsection{Naam kopje}`.

Paragraaf Dan heb je ook nog een paragraaf, deze krijg je door het commando `\paragraph{Naam paragraaf}` te typen. Dit kopje is een paragraaf.

2.2.4 Alinea

Je kunt alinea's maken door een lege regel tussen je tekst te maken.

2.2.5 Opsommingen en lijsten

Je kunt een niet-genummerde opsomming maken door het commando `\begin{itemize}` te typen. Eén punt in je opsomming maak je dan door `\item` te typen, achter dit commando type je een spatie en daarachter de tekst die je in dat punt wil hebben. Aan het einde van je opsomming zet je `\end{itemize}`.

- Dit is het eerste punt van de niet-genummerde opsomming
- Dit het tweede
- Dit het laatste

Een genummerde opmaak, zoals eerdere opsommingen in deze handleiding, kun je maken door het commando `\begin{enumerate}` te doen, dan maak je weer nummers door `\item` te typen. Weer eindig je met `\end{enumerate}`. Zonder toevoegingen krijg je dan

1. voorbeeld 1
2. voorbeeld 2

Wil je een andere opmaak of letters die oplopen dan type je `\begin{enumerate}[a.]`. Dan krijg je dit:

- a. voorbeeld 1
- b. voorbeeld 2

Op dezelfde manier kan je ook een opsomming met i'tjes maken door `\begin{enumerate}[i.]`

- i. voorbeeld 1
- ii. voorbeeld 2

3 Tekst

Net zoals met Microsoft Word heb je in L^AT_EX ook de mogelijkheid tekst in verschillende vormen te presenteren, hieronder een paar voorbeelden.

- **dik**, typ `\textbf{tekst}`
- *cursief*, typ `\textit{tekst}`
- **kleur**, typ `\textcolor{kleur}{tekst}` of `{\color{kleur}tekst}`
- klein, typ `\small` en de tekst erachter
- groot, typ `\large`
- groter, typ `\Large`
- nog groter, typ `\huge`
- allergrootst, typ `\Huge` .

Typ hierna wel weer `\normalsize` om verder te gaan in normale tekstgrootte.

4 Wiskundige omgeving

4.1 Vergelijkingen weergeven

Wil je formules in je verslag dan moet je gaan werken in een “wiskundige omgeving”. Er zijn meerdere manieren om deze aan te roepen, de drie vaakst gebruikte zijn *math*, *displaymath* en *align*, waarin je wiskundige formules kunt typen. Bij *math* komen de wiskundige symbolen in de regel² te staan, zoals $a^n + b^n = c^n$. Deze omgeving kun je op verschillende manieren aanroepen.

²Om bepaalde redenen werkt dit soms niet in koppen. Je zult het package *fixltx2e* moeten toevoegen om het te laten werken

De bekendste is het zetten van de wiskundige formules tussen $\$$ -tekens.

De *displaymath* omgeving wordt gebruikt op de wiskundige symbolen te centreren, zoals

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi = -i\hbar \frac{\partial \Psi}{\partial t}$$

Je kunt deze omgeving aanroepen door de het tussen de volgende haken te plaatsen: `\[...]`. Tot slot hebben we nog *align*, bij deze omgeving worden de wiskundige symbolen gecentreerd en krijgen ze een verwijzing achter de vergelijking, zoals

$$\int_{-\infty}^{\infty} \frac{\cos x}{x^2 + 1} = \frac{\pi}{e} \tag{1}$$

$$\int_{-\infty}^{\infty} \frac{\cos x}{x^2 + 1} = \frac{\pi}{e} \tag{2}$$

Om een vergelijking zo weer te geven gebruik je `\begin{align}` en `\end{align}`. Wil je dit zonder verwijzingen gebruiken, dan type je `\begin{align*}` en `\end{align*}`. Om meerdere vergelijkingen onder elkaar te zetten, type je achter een vergelijking `\\`. Gebruik het $\&$ -teken om de formules goed onder elkaar te zetten, meestal gebruik je het voor het $=$ -teken. Als je zin hebt om te experimenteren, probeer dan uit te zoeken wat er gebeurt als je meerdere $\&$'s in dezelfde regel gebruikt.

4.2 Superscript en subscript

Bij het werken in een wiskunde omgeving worden subscript ($_$) en superscript ($^$) veel gebruikt. Zo kunnen we bijvoorbeeld de rij a_1, a_2, \dots, a_{n-1} weergeven met de code `a_1, a_2, \dots, a_{n-1}` en de vergelijking $a^b \cdot a^c = a^{b+c}$ weergeven met de code `a^b \cdot a^c = a^{b + c}`. Let wel op dat bij meer dan één teken in super- of subscript dat er `{}` en `}` om heen staat.

4.3 Haakjes

Als we haken om grote symbolen in *displaymath* zetten, zoals bijvoorbeeld om een breuk heen, is het belangrijk de haakjes te schalen naar het symbool. Dit kan worden gedaan door de `\left` en `\right` commandos toe te voegen aan de haakjes:

$$\left| -\frac{1}{2} \right|^3 = \left(\frac{1}{8} \right)$$

wordt geproduceerd met de volgende code:

```
\[\left| -\frac{1}{2} \right|^3 = \left( \frac{1}{8} \right)\]
```

Als je een linkerhaakje wilt laten matchen met een rechterhaakje op een andere regel kan je de onzichtbare `\right.` gebruiken. Dit komt bijvoorbeeld voor bij lange definities van verzamelingen:

$$\text{s.t. } f^2(x) < \frac{\epsilon}{2} \}$$

wat wordt geproduceerd door de code

```
\[\left. \text{s.t. } f^2(x) < \frac{\epsilon}{2} \right. \]
```

(Let op het escapen van de accolade).

4.4 Wortels en breuken

Bij het gebruik van wiskunde zul je ook veel breuken of wortels gebruiken. Voor het weergeven van de breuk $\frac{a}{b}$ typ je in je code `\frac{a}{b}` en voor het weergeven van de wortel \sqrt{a} gebruiken we `\sqrt{a}` of voor $\sqrt[n]{a}$ gebruiken we `\sqrt[n]{a}`.

4.5 Speciale tekens

Er zijn veel symbolen die bij wiskunde gebruikt kunnen worden. In deze paragraaf zullen een aantal worden genoemd, maar er zijn er veel meer. Om snel op te kunnen zoeken welke code bij een teken hoort kun je de website of app Detexify³ raadplegen. Hierbij kun je denken aan integralen, maar ook aan bijvoorbeeld α en β . Hieronder staan de meest gebruikte symbolen uit het Griekse alfabet.

	code		code		code		code		code		code
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>	ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
ζ	<code>\zeta</code>	θ	<code>\theta</code>	λ	<code>\lambda</code>	μ	<code>\mu</code>	ξ	<code>\xi</code>	π	<code>\pi</code>
ρ	<code>\rho</code>	σ	<code>\sigma</code>	τ	<code>\tau</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>
ψ	<code>\psi</code>	Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Λ	<code>\Lambda</code>	Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>

Naast tekens uit het Griekse alfabet hebben we ook nog standaard functies, waarvoor commando's in \LaTeX bestaan. Deze standaard functies worden op de goede manier aangegeven als het commando wordt gebruikt. De meest gebruikte hiervan zijn:

	code		code		code		code		code		code
\arccos	<code>\arccos</code>	\arcsin	<code>\arcsin</code>	\arctan	<code>\arctan</code>	\arg	<code>\arg</code>	\sinh	<code>\sinh</code>	\cos	<code>\cos</code>
\cosh	<code>\cosh</code>	\deg	<code>\deg</code>	\det	<code>\det</code>	\dim	<code>\dim</code>	\tan	<code>\tan</code>	\ker	<code>\ker</code>
\lim	<code>\lim</code>	\log	<code>\log</code>	\max	<code>\max</code>	\min	<code>\min</code>	\sin	<code>\sin</code>		

Naast de tekens uit het Griekse alfabet en de standaard functies hebben we ook nog enkele sommen, producten, integralen etc. Deze symbolen kunnen zich aanpassen naar de omgeving waarin zij zich bevinden. Zo worden zij groter weergegeven in *displaymath* dan in *math*. De symbolen hieronder worden het meest gebruikt:

	code		code		code		code
\sum	<code>\sum</code>	\bigcap	<code>\bigcap</code>	\iint	<code>\iint</code>	\bigcup	<code>\bigcup</code>
\prod	<code>\prod</code>	\int	<code>\int</code>	\iiint	<code>\iiint</code>	\oint	<code>\oint</code>

Bij zowel \sum als \prod als \lim is het netjes als in *displaymath* omgeving de boven- en ondergrenzen direct boven of onder de kwantor staat. Het gebruik van subscript en superscript. Zo krijg je bijvoorbeeld

$$\sum_{n=1}^{\infty} \frac{1}{n}$$

door de volgende code `\sum_{n=1}^{\infty} \frac{1}{n}` te gebruiken. Bij bijvoorbeeld dubbele integralen is het netjes om het commando `\limits` te gebruiken, dit zorgt er voor dat de grenzen op de juiste plek komen te staan. Vergelijk bijvoorbeeld:

$$\iint_D \text{met} \iint_D$$

³<http://detexify.kirelabs.org/>

verkregen door de code `\[\iint_D \text{ met } \iint \limits_D \]`, hierbij is de tweede correct. Tot slot hebben we nog een aantal tekens die handig zijn om te weten.

	code		code		code		code
\pm	<code>\pm</code>	\times	<code>\times</code>	\setminus	<code>\setminus</code>	\cdot	<code>\cdot</code>
\mp	<code>\mp</code>	\gg	<code>\gg</code>	\ll	<code>\ll</code>	\top	<code>\top</code>
\leq	<code>\leq</code>	\in	<code>\in</code>	\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>
\cong	<code>\cong</code>	\geq	<code>\geq</code>	\approx	<code>\approx</code>	\mid	<code>\mid</code>
\equiv	<code>\equiv</code>	\sim	<code>\sim</code>	\neq	<code>\neq</code>	\leftarrow	<code>\leftarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>	\mapsto	<code>\mapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\rightarrow	<code>\rightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Uparrow	<code>\Uparrow</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>	\downarrow	<code>\downarrow</code>
\Downarrow	<code>\Downarrow</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	∞	<code>\infty</code>	∇	<code>\nabla</code>
∂	<code>\partial</code>	\ni	<code>\ni</code>	\supseteq	<code>\supseteq</code>	\Rightarrow	<code>\Rightarrow</code>

Voor vergelijkende operatoren zoals \leq , \in en \subset kun je ook een onkennende operator maken door `\not` voor de operator te zetten. Dan krijg je bijvoorbeeld $\not\leq$, \notin en $\not\subset$.

4.6 Matrices

Er zijn verschillende soorten matrices. De matrix die je het meest zult gebruiken is de matrix die tussen haken staat. Een matrix die je ook veel zult gebruiken is die tussen rechte lijnen, voor het berekenen van een determinant. Verder zijn er ook nog andere soorten matrices, maar deze zul je niet zo vaak gebruiken.

		L ^A T _E X-code
Gewoon	$\begin{matrix} a & b \\ c & d \end{matrix}$	<code>\begin{matrix} a & b \\ c & d \end{matrix}</code>
Tussen haken	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	<code>\begin{pmatrix} a & b \\ c & d \end{pmatrix}</code>
Tussen rechte lijnen	$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	<code>\begin{vmatrix} a & b \\ c & d \end{vmatrix}</code>

5 Tabellen

Er zijn verschillende mogelijkheden om een tabel aan te maken in een L^AT_EX-bestand. De meest gebruikte zijn de tabel (`\begin{tabular}[positie]{opmaak}`) en de array (`\begin{array}[positie]{opmaak}`). Een array kan alleen worden aangemaakt in de wiskunde omgeving.

Net als bij de matrices worden de elementen in een rij met een `&` gescheiden en sluit je een rij af met `\`. Om een horizontale lijn in je tabel te krijgen plaats je `\hline` in de code van je tabel.

Voor het bepalen van de positie kunnen de volgende commando's worden aangeroepen:

Commando	Resultaat
<code>t</code>	De tabel is met zijn top gealigneerd op de onderkant van de regel.
<code>b</code>	De tabel is met zijn onderkant gealigneerd aan de onderkant van de regel.
<code>c</code>	De tabel is gecentreerd rond de onderkant van de regel.

Bij het bepalen van de opmaak, bepaal je ook het aantal kolommen waaruit de tabel bestaat. Om te bepalen hoe deze kolommen eruit zien kunnen de volgende commando's worden aangeroepen:

Commando	Resultaat
<code>l</code>	Voegt een kolom toe waar de tekst links wordt uitgelijnd.
<code>r</code>	Voegt een kolom toe waar de tekst rechts wordt uitgelijnd.
<code>c</code>	Voegt een kolom toe waar de tekst wordt gecentreerd.
<code>p{x}</code>	De tekst wordt uitgevuld over een breedte <code>x</code> .
<code>*{x}{opmaak}</code>	Herhaalt de structuur in <code>opmaak</code> <code>x</code> keer.
<code> </code>	Voegt een verticale lijn toe.
<code> </code>	Voegt een dubbele verticale lijn toe.
<code>@{text}</code>	Gebruikt <code>text</code> als verticale lijn.

Voorbeeld

Tijd in maanden	Aantal konijnen	Berekening
0	1	1
1	2	1+1
2	3	1+2
3	5	2+3
4	8	3+5
5	13	5+8
6	21	8+13
\vdots	\vdots	\vdots

Deze tabel kan worden aangeroepen met de volgende code:

```
\begin{tabular}[c]{| l | c | r |}
\hline
Tijd in maanden & Aantal konijnen & Berekening \\
\hline
0 & 1 & 1 \\
1 & 2 & 1+1 \\
2 & 3 & 1+2 \\
3 & 5 & 2+3 \\
4 & 8 & 3+5 \\
5 & 13 & 5+8 \\
6 & 21 & 8+13 \\
$\vdots$ & $\vdots$ & $\vdots$ \\
\hline
\end{tabular}
```

6 Afbeeldingen invoegen

Het is ook mogelijk om afbeeldingen in te voegen bij \LaTeX . Hiervoor heb je het moet je bovenaan je code `\usepackage{graphicx}` zetten. Deze package zorgt ervoor dat je afbeeldingen kan invoegen. Vervolgens gebruik je de code `\includegraphics{naam_afbeelding.jpg}` om de afbeelding ook daadwerkelijk in te voegen. Om dit te doen moet de afbeelding die je wilt invoegen ook in de map staan waarin je \LaTeX -code ook staat. Let op: \LaTeX kan alleen `.pdf`-, `.png`- of `.jpg`-bestanden lezen.

Om het formaat van de afbeelding aan te passen kun je de code `\includegraphics[scale=0.5]{naam_afbeelding.jpg}`. Hierbij geef je de schaal van de afbeelding, in plaats hiervan kun je ook `[width=0.9\textwidth]` gebruiken. Zo wordt de afbeelding ook aangepast naar de gewenste grootte.

6.1 Plaatsing van de afbeelding

Het is vaak ingewikkeld om de afbeelding ook daadwerkelijk op de gewenste plek te krijgen. Om dit voor elkaar te krijgen gaan we de afbeelding in een drijvende omgeving zetten. Dit kunnen we doen door

```
\begin{figure}[]
\centering
\caption{Bijschrift bij afbeelding}
\includegraphics[width=0.5\textwidth]{naam_afbeelding.jpg}
\end{figure}
```

Hierbij kunnen tussen de rechte haken bij `\begin{figure}[]` commando's worden gegeven voor de plaatsing van de afbeelding. Hieronder vind je welke commando's handig zijn om te weten.

commando	resultaat
h	Plaats de omgeving hier.
t	Plaats de omgeving aan de bovenkant van een pagina.
b	Plaats de omgeving aan de onderkant van een pagina.
!	Forceer een van vorige opties.

Zoals je in de code hierboven kunt zien, is het ook mogelijk om een bijschrift te hebben bij je afbeelding. Dit kun je doen door `\caption{beschrijving}` in de drijvende omgeving van de afbeelding te plaatsen. Bij deze drijvende omgeving komt de beschrijving onder de afbeelding te staan. Het is ook mogelijk dat de beschrijving naast de afbeelding komt te staan. Dit kun je doen met de volgende code:

```
\begin{SCfigure}
\caption{Korte informatie over de afbeelding}
\includegraphics[width=0.5\textwidth]{naam_afbeelding.jpg}
\end{SCfigure}
```

Verder is het ook mogelijk om de tekst om de afbeelding heen te laten lopen. Zo komt de afbeelding in de tekst te staan. Hiervoor moet je allereerst bovenaan je code het volgende package aanroepen: `\usepackage{wrapfig}`. Om dan vervolgens de foto te plaatsen gebruik je de volgende code:

```
\begin{wrapfigure}{r}{0.5\textwidth}
\includegraphics[width=0.45\textwidth]{naam_afbeelding.jpg}
\caption{Korte informatie over de afbeelding}
\end{wrapfigure}
```

Hierbij kun je tussen de tweede accolades, waar in het voorbeeld een `r` staat, de plaatsing van de afbeelding bepalen. Hieronder vind je de meest handige commando's om te weten voor deze plaatsing:

commando	resultaat
r	Langs de rechterkant.
l	Langs de linkerkant.
R	Langs de rechterkant met een drijvende optie, om splitsing over pagina's tegen te gaan.
L	Langs de linkerkant met drijvende optie.

Tot slot is het ook mogelijk om verschillende afbeeldingen naast elkaar of onder elkaar te plaatsen. Dit kan fijn zijn om verschillende grafieken de kunnen vergelijken. Hiervoor moet je allereerst bovenaan je code het volgende package aanroepen: `\usepackage{subfig}`. Vervolgens kun je de volgende code gebruiken om een blok van vier afbeeldingen te maken. Hierbij komen twee afbeeldingen rijen van twee afbeeldingen te staan.

```
\begin{figure}
\centering
\subfigure[Beschrijving 1]{\includegraphics[width=0.4\textwidth]{grafiek1}
~
\subfigure[Beschrijving 2]{\includegraphics[width=0.4\textwidth]{grafiek2} \\
\subfigure[Beschrijving 3]{\includegraphics[width=0.4\textwidth]{grafiek3}
~
\subfigure[Beschrijving 4]{\includegraphics[width=0.4\textwidth]{grafiek4}
\caption{Grafieken van ...}
\end{figure}
```

7 Code invoegen

Het invoegen van code gebeurt met het package *listings*. Dit geeft een environment *lstlisting*. Je zet je code dan dus tussen `\begin{lstlisting}` en `\end{lstlisting}`. Wel moet je specificeren welke programmeertaal je gebruikt; als je bijv. C++ gebruikt zet je `\lstset{language=C++}` boven `\begin{document}`.

De L^AT_EX-code

```
\begin{lstlisting}
int main (int argc, char** argv) {
    std::cout << "Hello world" << std::endl;
    return 0;
}
\end{lstlisting}
```

heeft bijvoorbeeld

```
int main (int argc, char** argv) {
    std::cout << "Hello_world" << std::endl;
    return 0;
}
```

als output.

Tevens kun je simpelweg het bestand zelf invoegen door

```
\lstinputlisting{bestandnaam.extensie}
```

te gebruiken.

8 Referenties

Bij eigenlijke alle genummerde dingen kun het commando `\label{HierEenNaam}` gebruiken. Als je `\label{...}` eenmaal op de juiste plaats gezet hebt, kun je het nummer gebruiken met `\ref{HierEenNaam}`. Ook kun je met `\pageref{HierEenNaam}` verwijzen naar de pagina waar de label staat.

Voor een voetnoot in je tekst gebruik je `\footnote{Dit is een voetnoot}`⁴.

Voor uitgebreide referenties gebruik je

```
\begin{thebibliography}{99}
  \bibitem{afbeeldingen} Wikibooks, \emph{LaTeX/Floats, Figures and Captions}
  --- Wikibooks{,} The Free Textbook Project}.
  \url{http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions}
\end{thebibliography}
```

Vervolgens kan je met `\cite{afbeeldingen}` naar bijvoorbeeld de url met meer info over afbeeldingen verwijzen [2].

Voor verwijzingen is het belangrijk om de code 2x te compileren. Anders komen er vraagtekens of verouderde nummering te staan.

9 Overige interessante packages

Verder zijn er nog allerlei verschillende packages om je leven makkelijker te maken. Zoek ze eens op op internet!

- `\usepackage{fancyhdr}` Voor intelligente headers en footers op je pagina.
- `\usepackage[section]{placeins}` Zorg ervoor dat floating objects niet voorbij een sectieovergang gaan. Daarnaast kan je met `\FloatBarrier` handmatig een barrière definiëren voor alle floats.
- `latexmk` Kan het hele compileerproces voor je automatiseren.
- `\usepackage{sidecap}` Voor captions naast je floats in plaats van onder of boven.
- `\usepackage{beamer}` Om beamer-presentaties te maken (powerpoint).
- Wist je dat `\`, het commando voor een regelovergang, een optioneel argument heeft? Probeer bijvoorbeeld eens `\[5cm]` .
- Het LION heeft een eigen \LaTeX -klasse voor het schrijven van templates. [3] Deze doet heel veel voorwerk voor je zoals de layout van de voorpagina etc.

Referenties

[1] <http://en.wikibooks.org/wiki/LaTeX>

[2] Wikibooks, *LaTeX/Floats, Figures and Captions* — Wikibooks, *The Free Textbook Project*.
http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions

⁴Dit is een voetnoot

[3] <https://www.physics.leidenuniv.nl/bachelor/80-education/606-education-thesis-templates>