

Een L^AT_EX-Handleiding voor beginners

Roelof Ruules*

7^e editie, maart 2005[†]

“So we’re absolute beginners.”

DAVID BOWIE

Inleiding

Deze handleiding is geschreven met het doel om in kort bestek het één en ander uit te leggen over L^AT_EX. Met nadruk zij gesteld dat het hier alleen maar gaat om een *beperkte* handleiding; wie meer, of alles, wil weten van L^AT_EX verwijzen we naar de L^AT_EX-boeken van Leslie Lamport [Lam94], door ons aangeduid als ‘het L^AT_EX-boek’, en het duo Kopka & Daley [KD99]. De doelgroep van deze handleiding wordt gevormd door beginnende gebruikers die een eenvoudig document willen maken.

Nota bene: omdat L^AT_EX op verschillende computersystemen verschillend is geïnstalleerd, kan het voorkomen dat sommige zaken niet overal precies hetzelfde zijn. Dit geldt voornamelijk voor de commando’s waarmee een document wordt bekeken op het scherm of wordt geprint. De L^AT_EX-commando’s die wij hier behandelen zijn overal hetzelfde.

We nodigen iedereen uit om te reageren op deze handleiding. Mis je elementaire zaken, vind je domme taalfouten, heb je anderszins vragen: zeg het ons.

Voor wie is deze handleiding?

1. De **absolute leek met haast**, die alleen nog maar van L^AT_EX heeft gehoord, en die bovendien geen tijd heeft om het uitgebreide L^AT_EX-boek van Leslie Lamport door te spitten, omdat er op korte termijn een document, zoals een verslag, moet worden geproduceerd.
2. De **leek zonder haast**, die wèl tijd heeft gehad om het allesomvattende L^AT_EX-boek van Leslie Lamport te bestuderen, maar nog even op een rijtje wil hebben wat hij in ieder geval moet weten om L^AT_EX te kunnen gebruiken.
3. De **nu-en-dan gebruiker**, die wel al de elementaire handelingen van L^AT_EX beheerst, maar die nog even snel wil opzoeken ‘hoe het ook al weer zat’, zonder eerst het onoverzichtelijke L^AT_EX-boek van Leslie Lamport te moeten spellen. Met name voor deze gebruikersgroep is de volgende paragraaf nuttig ter oriëntatie.

Wat moet je zeker lezen?

De handleiding is ingedeeld in een aantal secties, die voor de verschillende gebruikers van meer of minder belang zullen zijn. Daarom geven we hier een korte inhoud van elke sectie, met aanwijzingen wat voor wie

*©Roelof Ruules, R.Ruules@phys.uu.nl

[†]Deze handleiding is ontstaan als een onderdeel van het project ‘T_EX2L_AT_EX’ van de T_EXnicie, en is een bewerking van een handleiding die werd geschreven voor A–Eskwadraat. De vijfde editie is ontdaan van verwijzingen naar A–Eskwadraat en *up-to-date* gemaakt; de zesde editie was verder uitgebreid met het pakket marvosym, maar omdat dit pakket fonts herdefinieert is dit weer verwijderd in de zevende editie. Dank aan alle leken die deze handleiding in verschillende stadia hebben willen lezen, en aan Jeroen van Dijk die verantwoordelijk was voor de oorspronkelijke tekst van sectie 5.

bedoeld is. Wat wij ‘nuttig’ noemen kun je beter wel lezen. Wat wij ‘aardig’ noemen mag je overslaan, maar het is aardig, ook tegenover de auteur, om het tenminste één keer wèl te lezen.

Sectie 1 ‘Wat is L^AT_EX?’ Een korte introductie over wat L^AT_EX is, en wat het *niet* is. Nuttig voor de leken, aardig voor de rest.

Sectie 2 ‘De meest eenvoudige documenten’ Hier vind je alles wat je absoluut moet weten om een L^AT_EX-document te kunnen maken. Nuttig voor de leken, aardig voor de rest.

Sectie 3 ‘Uitgebreidere documenten’ Hier behandelen we een aantal technieken waarmee je documenten nèt dat ietsje meer geeft, waardoor het al echt op een document gaat lijken. Ook brieven worden in dit gedeelte behandeld. Nuttig voor de leken, maar ook voor de gebruikers die het even vergeten waren.

Sectie 4 ‘De eerste stunts’ Hier gaan we dieper in op het gebruik van die commando’s die L^AT_EX maken tot één van de mooiste programma’s ter wereld. Met name de commando’s die het mogelijk maken wetenschappelijke teksten te fabriceren worden hier behandeld. Nuttig voor alle lezers van deze handleiding.

Sectie 5 ‘Het afmaken van een document’ Een overzicht van wat kleine dingen die je document een beter aanzien geven, plus tips over hoe je grotere documenten kunt aanpakken. Nuttig voor alle lezers van deze handleiding.

Appendix A ‘Alles op een rij’ Een overzicht van de meest gebruikelijke L^AT_EX-structuren en commando’s. Nuttig voor iedereen, vooral omdat hier ook commando’s worden genoemd die *niet* in het hoofdgedeelte van de handleiding voorkomen.

Appendix B ‘Begrippen’ Een korte verzameling van begrippen die in deze handleiding voorkomen en misschien niet altijd even duidelijk zijn uitgelegd. Nuttig voor iedereen die zich op zijn hoofd krabt als hij de handleiding aan het lezen is.

‘Bestanden’ en ‘documenten’, en het verschil daartussen

We hebben geprobeerd onderscheid te maken tussen ‘bestand’ en ‘document’. Met het ‘bestand’ bedoelen we steeds het tekstbestand, of delen daarvan, dat je zelf schrijft, en waarin L^AT_EX-commando’s zijn opgenomen. Met het ‘document’ bedoelen we het eindresultaat dat je krijgt als je het ‘bestand’ hebt geL^AT_EXd. Dit is een file die je naar het scherm kan sturen om te bekijken, of naar de printer. Ook deze definitieve, afgedrukte, versie noemen we het ‘document’. Om het ‘bestand’ te maken heb je alleen een editor nodig. Om het ‘document’ te maken heb je het ‘bestand’ nodig, plus de L^AT_EX-programmatuur.

De ‘oude’ L^AT_EX 2.09

De versie van L^AT_EX die momenteel beschikbaar is heet officieel L^AT_EX 2_ε. Deze versie is de opvolger van de ‘oude’ L^AT_EX 2.09. Wij spreken in deze handleiding steeds van L^AT_EX, en bedoelen daarmee L^AT_EX 2_ε.¹ Bestanden die nog onder L^AT_EX 2.09 zijn geschreven worden door L^AT_EX 2_ε herkend maar niet noodzakelijk-kerwijs goed verwerkt.

De voornaamste uitbreiding naar L^AT_EX 2_ε bestaat uit de toevoeging van een groot aantal modules, *packages* genaamd, aan de kale L^AT_EX. Zowel het L^AT_EX-boek [Lam94] als het boek van Kopka & Daley [KD99] behandelen L^AT_EX 2_ε.

Je zult in het algemeen een ernstige waarschuwing krijgen als je een oude variant probeert te L^AT_EXen. Het is dan ook verstandig oude bestanden om te zetten naar L^AT_EX 2_ε.

¹De eerste drie edities van deze handleiding behandelden de twee versies naast elkaar. We zijn daar met ingang van de vierde editie van afgestapt, omdat L^AT_EX 2_ε inmiddels de algemeen gangbare standaard is. De laatste versie 2.09 dateert van 1998.

1 Wat is L^AT_EX?

Het programma L^AT_EX werd ontwikkeld door Leslie Lamport en is sinds 1985 beschikbaar voor de gebruiker. Zelf noemt Lamport het een *document preparation system*, en dat geeft inderdaad heel aardig weer wat het is.

Heb je de inleiding al gelezen? Zonee, doe dat dan eerst maar.

L^AT_EX is géén tekstverwerker, zoals WORD dat is. Je kunt dat verschil als volgt zien. WORD is een *werkomgeving*, het is een programma dat je opstart en van waaruit je vervolgens documenten maakt. Die documenten bewerk je binnen WORD, je print ze met WORD, enzovoort. Feitelijk is de naam tekstverwerker misleidend, het zou tekstbewerker moeten zijn.

L^AT_EX daarentegen kun je het best vergelijken met een programmeertaal zoals PASCAL. Het is een programmeertaal waarmee je je tekst klaar maakt voor de drukker. Je maakt de tekst aan in een gewone editor (sommigen doen dat met WORD!), en het resultaat is een kale *sourcefile*, zo iets als een PASCAL-programma, waarin naast de echte tekst ook nog commando's staan waarmee wordt aangegeven hoe het uiteindelijke document er moet gaan uitzien. Het is deze *sourcefile* die wij het 'bestand' noemen. Vervolgens roep je L^AT_EX aan, waarna de tekst wordt omgezet in een file die naar de printer kan worden gestuurd — vergelijkbaar met het compileren van een PASCAL-programma. Deze file wordt de DVI-file genoemd. Het is deze DVI-file die wordt afgedrukt, op een printer maar ook op het beeldscherm. Wij noemen dat het 'document'.

Dat L^AT_EX een programmeertaal is, wordt extra duidelijk als je weet dat L^AT_EX zelf een uitbouwsel is van een andere programmeertaal: T_EX. Met T_EX kan de gebruiker bijna alles doen wat hij zou willen, maar het is niet zo eenvoudig om met T_EX te leren werken. L^AT_EX is een zogenaamd macropakket van T_EX. De dingen die je nu eenmaal altijd moet doen om een goed document te maken, zijn in L^AT_EX ondergebracht in een aantal overzichtelijke commando's. Met L^AT_EX kun je minder doen dan met T_EX², maar wat je kunt doen is veel gemakkelijker en gestructureerder.

Het programmeer karakter van L^AT_EX heeft nadelen. Een kleine fout in het bestand kan vreselijke gevolgen hebben tijdens het L^AT_EX-en. Een vergeten accolade { kan aanleiding zijn voor een schier oneindige rij foutmeldingen, of zelfs voor het staken van iedere verdere activiteit. En het terugvinden van de fout valt ook niet altijd mee. Net als bij een PASCAL-programma hoeft de plaats waar de fout wordt gemeld helemaal niet de plaats te zijn waar de fout is gemaakt.

Maar er zijn ook voordelen. Een WORD-document kan alleen maar worden bewerkt met WORD. Een L^AT_EX-bestand kan echter op de meest eenvoudige editor worden aangemaakt. Pas in een later stadium wordt het geheel geL^AT_EXd, en ontdaan van fouten en dergelijke. Een beetje L^AT_EX-gebruiker heeft er geen moeite mee thuis zijn bestand in te typen om het pas later om te zetten in een fraai document (maar het bezit van een eigen L^AT_EX is wel zeer aan te bevelen, dat willen wij ook wel toegeven).

Het programmeer karakter wordt ook nog op een andere manier duidelijk. Er zijn meerdere versies van L^AT_EX in omloop. Het verschil zit hem meestal in de hoeveelheid uitbreidingspakketten die je tot je beschikking hebt, en de programmatuur die er omheen zit (zoals viewers en fonts). Een goede distributie heeft uiteraard alles in huis wat in het L^AT_EX-boek wordt behandeld.

L^AT_EX werd geschreven vanuit het standpunt van de redacteur. Er kan veel met L^AT_EX, maar het programma is zo opgezet dat het resultaat toch altijd een zekere onderliggende structuur vertoont. Vandaar dat L^AT_EX-documenten altijd zo herkenbaar zijn. Je zou denken dat dat de gebruiker in zijn vrijheid beknot. In WORD kun je tenslotte helemaal zelf bepalen hoe je document er komt uit te zien — helaas is dat aan het resultaat meestal goed te zien. In de bibliografie van het L^AT_EX-boek vind je dat driekwart van de geciteerde boeken gaat over stijl, lay-out, etcetera. Met andere woorden: er is heel zorgvuldig over nagedacht.³ Wie met L^AT_EX werkt en iets maakt dat ingewikkelder is dan 'gewoon een stukkie tekst', wordt gedwongen om daarover een klein beetje na te denken. Dat maakt het leven voor de redacteur een stuk eenvoudiger, en zo hoort het ook.

L^AT_EX werd ook geschreven met de gedachte dat een zetter de documenten van de redacteur zonder al te veel moeite moet kunnen omzetten in gedrukte werken. Het gevolg daarvan is dat L^AT_EX soms dingen doet die de gebruiker niet verwacht en die nogal pietluttig lijken. Een voorbeeld daarvan is het aanhouden van

²Dit is niet helemaal waar, want veel T_EX-commando's kun je ook in een L^AT_EX-bestand verwerken.

³Weliswaar komen er regelmatig nieuwe *updates* van L^AT_EX uit, maar voor de gemiddelde gebruiker verandert er in de praktijk vrijwel niets. Kom daar eens om bij WORD! Het programma T_EX is zelfs helemaal uitgeëvolueerd — er wordt niets meer aan veranderd.

de kantlijn. Soms vindt L^AT_EX dat een woord niet kan worden afgebroken, maar dat het ook niet precies past op een regel. Het programma meldt dan dat de tekst, bijvoorbeeld, 0.2 punt uitsteekt. Dat is minder dan een tiende millimeter, en zeker in een gedrukte tekst met het blote oog nauwelijks zichtbaar. Maar een zetter zou er over vallen, en dus L^AT_EX ook, onder het motto ‘zachte heelmeeesters maken stinkende wonden’.

Tenslotte werd L^AT_EX opgezet om wetenschappelijke documenten mee te maken. Het is met L^AT_EX relatief eenvoudig om ingewikkelde formules te produceren. L^AT_EX is minder geschikt voor het maken van glossy uitziende werkjes. Een ingewikkeld plaatje opnemen in een L^AT_EX-document kost relatief veel moeite, en vaak is het makkelijker om L^AT_EX een ruimte te laten reserveren voor een later tussen te voegen afbeelding. Wie de show wil stelen bij een publiek van yuppen kan beter geen L^AT_EX gebruiken. Maar wie wil publiceren bij SPRINGER VERLAG heeft zelfs een SPRINGER L^AT_EX-stijl tot zijn beschikking — ze hoeven er bij SPRINGER alleen nog maar een kaftje om te doen.

Zoals gezegd zijn er tegenwoordig allerhande distributies van L^AT_EX beschikbaar. Hoe uitgebreid de L^AT_EX is die jij tot je beschikking hebt, is dus niet te zeggen. Wat wij in deze handleiding presenteren, zou in het algemeen niet tot foutmeldingen mogen leiden. Maar het kan zijn dat de pakketten `txfonts` en `pxfonts`. Je kunt ze altijd van het web plukken, via de website van de T_EX Users Group: www.tug.org.

2 De meest eenvoudige documenten

Deze sectie is opgedeeld in drie onderdelen. Eerst leggen we uit hoe je een bestand in L^AT_EX maakt. Daarna vertellen we hoe dat bestand moet worden omgezet in een L^AT_EX-document. Tenslotte laten we zien hoe je het document naar het beeldscherm of de printer stuurt.

2.1 Fase 1: het maken van een L^AT_EX-bestand

Een L^AT_EX-bestand wordt gemaakt met behulp van een editor. Je kunt de editor gebruiken die jou het beste ligt. De enige voorwaarde is dat je je bestand als ‘kale ASCII-tekst’ kunt opslaan. Er mogen dus geen code’s instaan van een tekstverwerker!

Het bestand zal voor het grootste deel bestaan uit de tekst die in het document komt, maar tussen die tekst staan L^AT_EX-commando’s. De naam van het bestand doet niet zoveel ter zake, maar moet altijd eindigen op `.tex` — wij zullen als voorbeeld steeds `verslag.tex` gebruiken.

De meeste commando’s beginnen met een zogenaamde backslash `\`, maar er zijn een paar uitzonderingen. Veel commando’s krijgen een extra argument mee, dat staat tussen accolade’s `{}`. Ook als je niets wilt meegeven, *moet* je de accolade’s neerzetten. Dus `\section{Fase 1}` is de aanduiding voor een sectie met als titel ‘Fase 1’, en `\section{}` is een aanduiding voor een sectie zonder een titel.

Aan sommige commando’s kun je opties meegeven. Die opties komen tussen rechte haken `[]` te staan. Als je geen opties meegeeft, *moet je de rechte haken weglaten*. Dus `\section{Fase 1}` is een sectie met de titel ‘Fase 1’, en `\section[fase een]{Fase 1}` is een sectie met de titel ‘Fase 1’, maar in de inhoudsopgave zou de tekst ‘fase een’ verschijnen.

L^AT_EX maakt onderscheid tussen hoofdletters en kleine letters. Het commando `\large` is dus anders dan het commando `\Large`. Dat betekent dat L^AT_EX-commando’s *exact* moeten worden overgenomen, inclusief de hoofdletters en kleine letters.

2.1.1 Het minimale bestand

Een L^AT_EX-bestand bestaat tenminste uit de volgende onderdelen:

```
\documentclass[<s-opties>]{<stijl>}
\usepackage[<p-opties>]{<pakket>}
<preamble>
\begin{document}
<tekst>
\end{document}
```

De verschillende woorden die hier met een `\` beginnen zijn commando's. De `<stijl>` geeft aan wat voor soort document er volgt. De vier standaard stijlen zijn `article`, `report`, `book` en `letter`. Het verschil tussen de eerste drie zit hem er voornamelijk in hoe hoofdstukken worden ingedeeld en hoe de lay-out van de pagina er uitziet; de stijl `letter` vormt een apart geval, dat ook apart zal worden behandeld.

De `<s-opties>` geven extra aanwijzingen met betrekking tot de `<stijl>`. Als je *geen* opties wilt gebruiken, moet je dus ook de rechte haken weglaten!

De regel `\usepackage` wordt gebruikt om pakketten in te laden. Sommige pakketten kennen ook weer opties, die je in de `<p-opties>` aangeeft.

Nota bene: als je een pakket inlaadt met opties, kun je *niet* met hetzelfde commando ook een ander pakket inladen. Vergelijk de volgende twee situaties:

```
\usepackage{babel,amsmath}           \usepackage[dutch]{babel}
\usepackage{amsmath}
```

In het linker voorbeeld worden twee pakketten zonder opties ingeladen. In het rechter wordt het pakket `babel` ingeladen met de optie `dutch`; het pakket `amsmath` moet daarom apart worden ingeladen.

In de `<preamble>` staan aanwijzingen voor de verwerking van het bestand, bijvoorbeeld de lengte van de pagina. Vaak zul je hier helemaal niets neerzetten, omdat alles standaard al goed is ingesteld. Maar zaken als de titel en auteur van een document vind je hier.

De `<tekst>` is de tekst. Daar kunnen L^AT_EX-commando's in staan, maar het hoeft niet. Alles wat na het commando `\end{document}` komt wordt door L^AT_EX genegeerd. Je kunt hier bijvoorbeeld commentaar kwijt.

2.1.2 Verschillende stijlen en opties

In de tabellen 1 en 2 vind je de diverse stijlen die je kunt gebruiken, alsmede de opties die je kunt toevoegen. In tabel 3 noemen we een paar pakketten. **Nota bene:** niet alle opties zijn op ieder systeem beschikbaar!

Tabel 1: Mogelijke stijlen

Stijl:	Toelichting:
<code>article</code>	Meest gebruikelijke stijl voor een kort document, zoals een practicum-verslag.
<code>report</code>	Vooral geschikt voor uitgebreide verslagen, zoals een scriptie.
<code>book</code>	Voor echte boeken.
<code>letter</code>	Speciaal voor brieven.

2.1.3 Indeling van de tekst, spaties, alinea's, en afbreekregels

Omdat het bestand door L^AT_EX verder wordt bewerkt, maakt het niet zoveel uit hoe je de tekst in het bestand precies neerzet. Zo maakt L^AT_EX geen onderscheid tussen één spatie of honderd spaties; in het document ziet dat er hetzelfde uit. Opeenvolgende regels tekst in het bestand worden door L^AT_EX aan elkaar geplakt. Om een nieuwe alinea te beginnen, moet je in het bestand tenminste één lege regel tussen de twee alinea's laten — meer regels mag ook, maar in het document ziet dat er allemaal hetzelfde uit.

Tabel 2: Mogelijke stijl-opties

Optie:	Toelichting:
11pt	Geeft de lettergrootte weer. De standaard lettergrootte is 10 punts, zoals in deze tekst, de 11 punts letter is 10% groter. De uitdrukking ‘punt’ komt uit de zettwereld en is een maat voor de hoogte van een letter (zonder stokken).
12pt	Als 11pt, maar deze letter is 20% groter dan de standaard.
a4paper	Zorgt er voor dat de tekst in het document wordt gezet in de juiste grootte voor een vel A4. Standaard wordt de tekst wel voor A4 voorbereid, maar met teveel wit rond de tekst.

Tabel 3: Mogelijke pakketten

babel	Zorgt ervoor dat de juiste afbreekregels worden gebruikt en dat aanduidingen als ‘Hoofdstuk’ en ‘figuur’ in de juiste taal worden gebruikt. De gewenste taal wordt als optie meegegeven: <code>\usepackage[dutch]{babel}</code>
amsmath	Geeft een aanzienlijke uitbreiding op de standaard wiskundige symbolen en structuren. Zie [KD99], pagina 405 en verder, voor een bespreking hiervan.
txfonts	Zet het hele document, dus ook wiskundige symbolen, in het lettertype Times Roman.
pxfonts	Als txfonts, maar dan met Palatino.
eurosym	Allerhande €-symbolen: gebruik <code>\euro</code> voor het standaardstijlsymbool.

Nota bene: \LaTeX kent de afbreekregels voor het Nederlands⁴, maar in het bestand mogen de woorden *niet* worden afgebroken. Sommige tekstverwerkers, zoals WORD, willen nog wel eens afbreekstreepjes invoegen. Als je zo’n tekstverwerker gebruikt om je bestand te maken, zorg er dan voor dat dergelijke afbreekfaciliteiten uitstaan!

2.1.4 Verboden lettertekens

Sommige lettertekens hebben voor \LaTeX een speciale betekenis. Hoewel er geen `\` voor staat, zijn het toch commando’s. Om de gewenste tekens in het document te krijgen moet je speciale commando’s gebruiken. In tabel 4 staan die tekens, en de commando’s om ze te krijgen.

Tabel 4: Verboden tekens

Verboden:	#	\$	%	&	~	_	^	\	{	}
Gebruik:	<code>\#</code>	<code>\\$</code>	<code>\%</code>	<code>\&</code>	<code>\~{}</code>	<code>_{}</code>	<code>\^{}</code>	<code>\backslash</code>	<code>\{</code>	<code>\}</code>
Resultaat:	#	\$	%	&	~	_	^	\	{	}

\LaTeX kent nog een boel andere speciale tekens. De meeste zijn opgenomen in Appendix A, en in het \LaTeX -boek staan ze op pagina’s 40–47.

⁴Het kan soms nodig zijn om de juiste talen te activeren; zie daarvoor de handleiding van je lokale \LaTeX .

2.1.5 Commentaar in het bestand

Zoals al gezegd kun je in het bestand commentaar opnemen na het `\end{document}` commando. Maar ook daarvoor is het mogelijk om commentaar op te nemen dat niet in het document verschijnt. L^AT_EX gebruikt daarvoor het commando `%` (vandaar dat je een ander commando moet gebruiken om het teken `%` in je document te krijgen). Als je ergens in een regel van je bestand een `%` gebruikt, wordt alles na dat teken genegeerd en gaat L^AT_EX gewoon verder met de volgende regel. Commentaar kun je dus het beste opnemen door direct aan het begin van een regel een `%` neer te zetten, gevolgd door je commentaar.

2.2 Fase 2: het L^AT_EXen van een bestand

Je bent met het bovenstaande in principe in staat om een bestand te maken, zij het dat dat nog weinig om het lijf heeft. Maar dat komt later. Het wordt tijd om het bestand te L^AT_EXen. Dit is voor ons het moeilijkste gedeelte, want hier ga je de programmatuur van L^AT_EX gebruiken. Die kan per systeem verschillen. Wij geven hier de meest generieke commando's, die op de meeste systemen wel zullen werken. Als dat niet het geval is, zul je daar zelf een oplossing voor moeten vinden⁵.

Je hebt als het goed is een bestand gekregen waarvan de naam eindigt op `.tex`, dus bijvoorbeeld `verslag.tex`. Je kunt het bestand nu eenvoudig L^AT_EXen door te typen `latex verslag`. Je krijgt een aantal mededelingen te zien op het scherm. Het programma is nu bezig het bestand in te lezen en te verwerken. Het resultaat is een file die dezelfde naam heeft als het bestand, maar dan met de extensie `.dvi`, in ons geval wordt dat dus `verslag.dvi`.⁶

De meeste moderne L^AT_EX-distributies beschikken tegenwoordig ook over een variant die pdfLaTeX heet. Daarmee produceer je geen dvi-file maar een kant en klaar pdf-bestand, dat je kunt bekijken en printen met ACROBAT.

Voordat het zover is moet het bestand wel foutloos zijn. Fouten worden door L^AT_EX aangegeven onder vermelding van het regelnummer waar het mis ging — maar dat hoeft niet de plaats te zijn waar de echte fout zit! Daarna volgt er een vraagteken `?`. Je kunt dan doorgaan door `Enter` te geven, of extra uitleg krijgen door `h` te geven gevolgd door `Enter`. Helemaal stoppen kan ook met `x` gevolgd door `Enter`; L^AT_EX produceert dan alleen een document tot aan het punt waar het mis ging.

Het is verstandig om eerst door te gaan, maar dat kan betekenen dat er nog een aantal foutmeldingen volgt, en dat het eindresultaat niet is zoals het zou moeten zijn. Het is echter tijdrovender om steeds één fout te herstellen en weer opnieuw te L^AT_EXen.

Soms is L^AT_EX echt niet in staat om door te gaan. Je krijgt dan iedere keer dezelfde foutmelding te zien. In ieder geval is het raadzaam bij vijf of meer fouten op dezelfde plaats te stoppen en terug te gaan naar het bestand om de boel te repareren. In een enkel geval komt L^AT_EX met een sterretje `*`. Dan zit het echt fout. De enige remedie is dan stoppen en kijken wat er mis kan zijn. Een heel enkele keer loopt L^AT_EX dusdanig vast dat je er ook met `x` niet uit komt; gebruik dan `Ctrl-C`.

Vaak zien de foutmeldingen er heel bedreigend uit, maar zelden is er werkelijk iets vreselijks aan de hand. De meeste fouten worden veroorzaakt door een spelfout in een commando, een ontbrekende accolade `{` of `}`, een missend 'math-mode' teken `$`, of doordat je toch één van de verboden tekens hebt gebruikt.

L^AT_EX geeft ook regelmatig meldingen zonder te stoppen. Meestal gaat het dan om iets in de trant van `overfull \hbox` plus een getal en een stukje van het bestand. De meestvoorkomende woorden zijn `overfull` of `underfull`, die duiden op respectievelijk een overschot of een tekort aan tekst, en `\hbox` of `\vbox` die aangeven dat het gaat om respectievelijk de kantlijn of de onderkant van de pagina.

Alle (fout)meldingen worden, behalve naar het scherm, ook nog weggescheven naar een aparte file: de log-file. Deze file heeft dezelfde naam als het bestand, maar de extensie `.log`, dus `verslag.log`. Je kunt deze log-file dus nog eens op je gemak nakijken, als het op het scherm wat te snel ging.

Als alles goed is gegaan meldt L^AT_EX dat ook. In dat geval kun je door naar de volgende fase.

⁵Een oplossing is om te kijken of mensen in je omgeving wel zonder problemen kunnen werken, en het dan aan hen te vragen. Een andere oplossing is om bij je systeembeheerder te gaan zeuren. Een oplossing die gegarandeerd *niet* werkt is om mij een mailtje te sturen.

⁶L^AT_EX produceert nog meer bestanden, die allemaal dezelfde naam hebben met een andere extensie, maar voorlopig zijn die voor ons niet van belang.

2.3 Fase 3: het bekijken en printen van een document

Hier zul je zelf het meeste veldwerk moeten verrichten, omdat er voor deze programmatuur geen standaard meer is. Er is een grote hoeveelheid viewers beschikbaar. In de documentatie van je lokale L^AT_EX zou je daar meer over moeten kunnen vinden. Op UNIX-systemen vind je meestal wel een programma `xdvi`. Onder windows vind je vaak dingen als `yap`, `windvi` of `dviview`⁷. Als je gebruik hebt gemaakt van pdfL^AT_EX, dan kun je het programma ACROBAT gebruiken om je document te bekijken. Hoe de viewer ook heet, je roept je document tevoorschijn met `<naam-van-viewer> verslag`

Het is verstandig om het document eerst op het scherm helemaal te bestuderen op fouten voordat je het doorstuurt naar de printer.

Met welk commando je het document kan afdrukken is ook zeer afhankelijk van het systeem waarop je werkt. Veel viewers kunnen tegenwoordig ook zonder al te veel problemen printen. Op UNIX-systemen wordt een document vaak eerst geconverteerd naar een PostScript-file, met het commando `dvips`, waarna de postscript-file naar de printer wordt gestuurd.

Ook als je precies weet welke programma's je moet gebruiken voor het bekijken en printen kun je nog op problemen stuiten. De meeste problemen krijg je doordat je lettertypes hebt gebruikt die niet bekend zijn op jouw systeem, of van bekende lettertypes een grootte die weer niet bekend is. Sommige grafische commando's leiden ook tot problemen.

3 Uitgebreidere documenten

In deze sectie gaan we in op de commando's die je het meeste zult nodig hebben wanneer je een document maakt. Eerst behandelen we een paar eenvoudige commando's, daarna vertellen we hoe je lijsten in een mooie lay-out kunt krijgen, en tenslotte laten we zien hoe je de Letter stijl moet gebruiken.

3.1 Begin en einde

Allereerst iets dat je moet begrijpen voor je verder kunt met veel commando's, namelijk het begrip 'scope'. Sommige commando's hebben een beperkte reikwijdte, ze zijn alleen geldig binnen bepaalde grenzen. Dat beperkte gebied wordt de *scope* genoemd.

De *scope* van sommige commando's wordt begrensd door een `\begin{<naam>}` aan het begin en een `\end{<naam>}` aan het eind. De `<naam>` wordt dan de omgeving genoemd, of *environment* in het Engels. Tussen de begin- en eind-commando's is de omgeving in werking, en kun je commando's gebruiken die speciaal voor die omgeving gelden; dezelfde commando's werken buiten de omgeving vaak niet of anders. Het vergeten van een `\begin` of `\end` commando levert meestal meerdere foutmeldingen op. Ook als de `<naam>` in beide gevallen niet hetzelfde is, krijg je dat van L^AT_EX te horen. **Nota bene:** als je binnen een omgeving *X* een omgeving *Y* begint, moet deze *Y* ook worden afgesloten voordat *X* wordt afgesloten!

Een andere manier om de *scope* aan te geven is door accolade's `{ }` om het geldigheidsgebied te zetten. Dat gebeurt vooral bij commando's die tijdelijk iets aan het uiterlijk van de tekst veranderen, zoals commando's die een ander lettertype opleveren. Als je de accolade's vergeet, is het geldigheidsgebied van zo'n commando eenvoudig de rest van het bestand. Ook als dat de bedoeling is, is het toch verstandiger dat wel met accolade's aan te geven. Dat is logischer en bovendien overzichtelijker voor anderen.

3.2 Lettertypen en lettergrootten

Het is in L^AT_EX heel eenvoudig om over te gaan op een ander lettertype, op een andere lettergrootte, of op een combinatie van beide. De commando's worden gebruikt tussen accolade's, dus bijvoorbeeld `{\itshape cursief}` levert het woordje *cursief* op. Alles wat binnen de accolade's staat wordt dan in het betreffende lettertype gezet. Zie ook 3.1 voor meer uitleg. Het is natuurlijk ook mogelijk om binnen de accolade's weer over te gaan op nog een andere lettertype. **Nota bene:** als een lettertype over meerdere alinea's veranderd wordt, moet je binnen de *scope*, dus binnen de accolade's, het einde van de laatste alinea

⁷Ik ben helaas niet thuis op Macs, dus hoe het daar heet weet ik niet...

aangeven; met andere woorden: de lege regel die komt ná de laatste alinea in het andere lettertype staat nog binnen de accolade's.

Tabel 5: Commando's die het lettertype bepalen

Commando:	Toelichting:
<code>\rmfamily</code>	'Roman', een romeins lettertype.
<code>\sffamily</code>	'Sans Serif', een schreefloos lettertype.
<code>\ttfamily</code>	'Teletype', een typemachine lettertype.
<code>\upshape</code>	'Up', een rechtopstaande letter.
<code>\itshape</code>	' <i>Italic</i> ', een cursieve letter.
<code>\slshape</code>	' <i>Slanted</i> ', een andere cursieve letter.
<code>\scshape</code>	'SMALL CAPS', EEN LETTERTYPE IN (KLEINE) HOOFDLETTERS.
<code>\mdseries</code>	'Medium', een gemiddelde letter.
<code>\bfseries</code>	' Bold Face ', een vette letter.

In tabel 5 worden de lettertypen gegeven. Zoals je ziet kennen we *families*, *shapes* en *series*. Deze kun je met elkaar combineren. Een vette schreefloze letter maak je met `{\sffamily\bfseries zo dus}`, **zo dus**. **Nota bene:** niet alle combinaties zijn mogelijk, afhankelijk van welke lettersoorten jouw \LaTeX allemaal kent.

Een speciaal geval is het commando `\emph{}` (van *emphatic*). Dit levert meestal ook een cursief lettertype op, maar niet altijd. Als je dit commando gebruikt binnen een cursieve tekst, is het resultaat juist een rechtopstaande letter. Je kunt dit commando dus beter gebruiken voor cursiveringen: `\emph{cursief}` levert ook *cursief* op.

Op precies dezelfde manier werken de commando's waarmee je de lettergrootte instelt. In tabel 6 worden ze gegeven. De werkelijke dimensies van de verschillende lettergrootten zijn afhankelijk van de

Tabel 6: Commando's die de lettergrootte bepalen

Commando:	Toelichting:
<code>\tiny</code>	Het kleinste lettertype.
<code>\scriptsize</code>	Een iets groter lettertype.
<code>\footnotesize</code>	De lettergrootte voor voetnoten.
<code>\small</code>	De grootste kleine maat.
<code>\normalsize</code>	De standaard lettergrootte.
<code>\large</code>	De kleinste grote maat.
<code>\Large</code>	Ietsje groter dan de vorige.
<code>\LARGE</code>	En weer ietsje groter.
<code>\huge</code>	Al bijna reusachtig.
<code>\Huge</code>	De allergrootste.

grootte die je aan het begin van het bestand hebt gekozen. Als je de optie `[12pt]` hebt gebruikt, dan levert `\tiny` ook een grotere letter dan wanneer je de standaard 10 punts instelling hebt gebruikt. Het is bovendien mogelijk dat de allergrootste of allerkleinste maten niet voorradig zijn. In dat geval neemt \LaTeX de grootte die er het dichtste bij in de buurt komt — dan hebben, bijvoorbeeld, de `\LARGE`, `\huge` en `\Huge` commando's hetzelfde effect. Merk ook nog eens op dat de commando's voor sommige grote typen wel op dezelfde manier worden geschreven, maar met verschillende hoofdletters (bijvoorbeeld `\large`, `\Large` en `\LARGE`). \LaTeX maakt daar dus echt onderscheid in!

3.3 Spaties

Binnen documenten wordt er onderscheid gemaakt tussen enerzijds spaties tussen woorden, en anderzijds spaties tussen hele zinnen. De laatste zijn groter. \LaTeX heeft een hele primitieve manier om te zien wat voor spatie er moet worden gebruikt. Als er ergens een punt staat, en die punt wordt niet vooraf gegaan door een hoofdletter, dan is er sprake van een nieuwe zin, en moet er dus een grotere witruimte worden gebruikt. Dezelfde regel wordt ook gehanteerd voor andere interpunctie zoals een vraagteken of uitroepetekens, en zelfs als voor of na de interpunctie nog een haakje of aanhalingsteken volgt. Het hiervolgende geldt dus ook voor die tekens.

In de meeste gevallen gaat dit goed, maar niet altijd. Een voorbeeld van waar het mis gaat: iemand wordt midden in de zin genoemd met een titel, bijvoorbeeld drs. Huppeldepup. De punt na drs. is niet bedoeld om een zin te eindigen, maar wordt door \LaTeX wel zo gezien. Omgekeerd kan het ook misgaan: een zin kan eindigen op een hoofdletter, zoals wanneer wij het hebben over de geheimzinnige meneer X. In dit geval moet \LaTeX juist wel een grotere spatie gebruiken.

In het eerste geval, de punt midden in een zin, moet je de punt direct laten volgen door een `\`, dus `drs. \ Huppeldepup`. In het tweede geval, het einde van de zin op een hoofdletter, moet je de punt laten voorafgaan door `\@`, dus `meneer X\@`. In . . .

Een apart geval zijn spaties na \LaTeX -commando's. Na een commando binnen een zin zet \LaTeX geen spatie.⁸ Om toch een spatie te krijgen, moet je het commando direct laten volgen door `\`, dus `\LaTeX\` in plaats van alleen `\LaTeX`. **Nota bene:** ná die `\` moet wel weer een spatie komen, want anders denkt \LaTeX dat het gaat om een commando!

Hier is nog een reden om het eerdergenoemde commando `\emph{}` te gebruiken. Als je cursieve tekst hebt gevolgd door normale tekst, is meestal de tussenruimte tussen die verschillende teksten te klein; `\emph{}` corrigeert daarvoor.

In tabel 7 staat alles nog eens opgesomd.

Tabel 7: Spaties in de tekst

- Punt midden in de zin, niet vooraf gegaan door een hoofdletter: `\` direct ná de punt.
- Punt aan het einde van een zin, vooraf gegaan door een hoofdletter: `\@` direct vóór de punt.
- Na een \LaTeX -commando in een zin: `\` direct ná het commando, gevolgd door een spatie.

Een speciaal geval is de ‘onbreekbare’ spatie. Dat is een spatie tussen twee woorden, waardoor die woorden door \LaTeX nooit over twee regels worden opgedeeld. Voorbeelden zijn spaties tussen tussenvoegsel en achternaam, zoals in de naam ‘De Vries’. Gebruik hiervoor het commando `~`, dus `De~Vries`.

3.4 Afbreken

Zoals gezegd kent \LaTeX de Nederlandse afbreekregels, maar het gaat wel eens mis. Zo heeft \LaTeX moeite met het afbreken van woorden waar een koppelteken in voorkomt; die worden alleen afgebroken op het koppelteken, of helemaal niet.

Als de tekst op een bepaald punt niet goed wordt afgebroken, kun je in het bestand aangeven waar je de afbreking het liefste zou zien. Daarvoor kun je het commando `\-` gebruiken: als in je bestand het cruciale woord als `af\bre\ken` wordt geschreven, zal \LaTeX proberen alleen af te breken bij de `\-` commando's. De commando's zelf krijg je uiteraard niet te zien in je document.

Als je een woord gebruikt in je bestand dat dusdanig apart is dat \LaTeX er geen raad mee weet, kun je ook in één keer opgeven hoe dat woord voortaan moet worden afgebroken. Dat kan met het commando `\hyphenation{<lijst>}`, dat moet worden opgenomen in de `<preamble>` (zie 2.1). De `<lijst>` is een lijst van woorden, gescheiden door komma's, waarin met `-` commando's is aangegeven waar er afgebroken mag worden. Dus `\hyphenation{af-bre-ken,af-ge-bro-ken}`.

⁸Er zijn ook commando's, zoals `\section`, die zelf zorgen voor ruimte na het commando.

3.5 Accenten

Accenten krijg je in \LaTeX met speciale commando's, die in tabel 8 staan opgesomd. De commando's werken voor alle letters, dus niet alleen voor de letters die we hier gebruiken. Een speciaal geval is de letter

Tabel 8: Accenten

Commando:	<code>\' {o}</code>	<code>\' {o}</code>	<code>\" {o}</code>	<code>\^ {o}</code>	<code>\~ {o}</code>	<code>\={o}</code>	<code>\. {o}</code>	<code>\r {o}</code>
Werking:	ó	ò	ö	ô	õ	ō	ò	õ
Commando:	<code>\u {o}</code>	<code>\v {o}</code>	<code>\H {o}</code>	<code>\t {oo}</code>	<code>\c {c}</code>	<code>\d {o}</code>	<code>\b {o}</code>	
Werking:	ů	ǒ	Ǔ	ø	ç	ȝ	ǫ	

'i', omdat daar al een puntje op staat. Bij de gangbare accenten gaat dat goed: `\' {i}` levert í op. Maar bij andere accenten kan het fout gaan. Door een `\` voor de 'i' te zetten, krijg je een 'puntloze' letter, dus bijvoorbeeld `\u {i}` om ě te krijgen. Op dezelfde manier levert `\j` een puntloze j op.

3.6 Indeling in ((sub)sub)secties en verder

In de stijlen `article`, `report` en `book` (zie onder 2.1.2) kun je eenvoudig verdelingen maken in secties, subsecties, subsubsecties en nog dieper. De commando's die daarvoor worden gebruikt, maken het bovendien heel gemakkelijk om een inhoudsopgave te genereren. Ze staan in tabel 9 genoemd. De `<tekst>`

Tabel 9: Commando's die secties aangeven

Commando:	Toelichting:
<code>\part{<text>}</code>	Een deel van een document. Wordt romeins genummerd.
<code>\chapter{<text>}</code>	Alleen in <code>report</code> en <code>book</code> , dus <i>niet</i> in <code>article</code> ! Een hoofdstuk. Wordt gewoon doorgenummerd, ook als de hoofdstukken over meerdere delen zijn verspreid.
<code>\section{<text>}</code>	Een sectie. Wordt per hoofdstuk genummerd.
<code>\subsection{<text>}</code>	Een subsectie. Wordt per sectie genummerd.
<code>\subsubsection{<text>}</code>	Wordt normaal niet genummerd.
<code>\paragraph{<text>}</code>	Wordt normaal niet genummerd.
<code>\subparagraph{<text>}</code>	Wordt normaal niet genummerd.

verschijnt steeds als titelkop, eventueel voorafgegaan door een nummer. Alle titelkoppen die zijn genummerd, worden ook opgenomen in een eventuele inhoudsopgave.

Als je wel een titelkop wil, maar geen nummer, kun je dat doen door de commandonaam te laten volgen door een sterretje, dus `\section*{<text>}`. Deze titelkoppen komen niet in een inhoudsopgave. Dit is met name handig als je in een tekst iets wil laten uitspringen zonder dat je daarvoor ingewikkelde dingen wilt doen.

Als je een inhoudsopgave wilt maken, moet je in het bestand op de plek waar de inhoudsopgave moet komen te staan, het commando `\tableofcontents` opnemen. Om de inhoudsopgave te krijgen, moet het bestand tenminste twee keer zonder onderbreking ge \LaTeX d worden. Dat is nodig omdat \LaTeX de informatie voor de inhoudsopgave in een andere file opslaat (bij de eerste keer \LaTeX en), die vervolgens door `\tableofcontents` wordt gelezen (bij de tweede keer \LaTeX en). Als je een inhoudsopgave maakt, is het wel noodzakelijk het pakket `babel` met de optie `dutch` te laden (zie 2.1.2), omdat je anders boven de inhoudsopgave de tekst 'Contents' krijgt.

Je kunt een document voorzien van een vermelding van titel en auteur met behulp van het commando `\maketitle`, dat direct na het commando `\begin{document}` in het bestand moet worden opgenomen. In de `preamble` moeten in ieder geval de commando's `\title{<titel>}` en `\author{<auteur>}` voorkomen, waarbij `<titel>` en `<auteur>` uiteraard staan voor titel en auteur. Je kunt ook nog het commando

`\date{<datum>}` gebruiken, maar als je dat weglaat genereert `\maketitle` automatisch de datum op dat moment. Als je dus helemaal geen datum vermeld wilt zien, moet je `\date{}` gebruiken.

3.7 Lijsten

\LaTeX kent verschillende commando's waarmee je lijsten kunt maken. Lijsten zijn gewoon opsommingen van verschillende items, en het verschil tussen de drie standaard lijsten zit hem voornamelijk in hoe de items binnen de lijst worden weergegeven. De lijsten worden allemaal aangegeven met `\begin{<naam>}` en afgesloten met `\end{<naam>}`, waarbij `<naam>` aangeeft om wat voor soort lijst het gaat.

Binnen een lijst geef je met behulp van een `\item` commando aan dat er een nieuw item begint. Het eerste woord binnen een lijst is dus altijd `\item`. Na het `\item` commando volgt de tekst van het item. Binnen een item in een lijst kun je een tweede lijst maken, een geneste lijst dus.

Hier volgt een lijst waarin de drie lijsten worden omschreven:

description Geeft een lijst waarbij de items voorzien zijn van een omschrijving. De items worden binnen de lijst aangegeven met `\item[<beschr>]`, waarbij `<beschr>` de omschrijving is van het item. Deze lijst is zelf een `description`, en de `<beschr>` bestond hier uit het woord 'description'.

enumerate Geeft een lijst waarbij de items in oplopende volgorde worden genummerd. Hier begint ieder nieuw item dus met `\item`, waarna de tekst van het item volgt.

itemize Geeft een lijst waarbij de items worden voorzien van een merkteken. Meestal is dat een zogenaamde bullet `•`, maar dat hoeft niet per se.

Hieronder staan twee voorbeelden van lijsten. Merk op dat het `\item` commando in het ene geval geen rechte haken `[]` heeft, en in het andere geval wel.

<code>\begin{itemize}</code>	• Dit is item 1	<code>\begin{description}</code>	Item 1 Tekst
<code>\item Dit is item 1</code>		<code>\item[Item 1] Tekst</code>	
<code>\item Dit is item 2</code>	• Dit is item 2	<code>\item[Item 2] Tekst</code>	Item 2 Tekst
<code>\end{itemize}</code>		<code>\end{description}</code>	

Als de tekst van een item met rechte haken wilt laten beginnen, dan moet je die tekst weer omgeven met accolade's. Dus `\item {[Dit item...]}`

3.8 Brieven

\LaTeX kent een speciale documentstijl `letter`. Wanneer je een brief, of meerdere brieven, wilt schrijven wordt het bestand iets anders opgebouwd. Hieronder volgt het minimale bestand voor een brief:

```
\documentclass[<s-opties>]{letter}
  <preamble>
\signature{<je naam>}
\address{<je adres>}
\begin{document}
\begin{letter}{<naam>\\<adres>\\<plaats>}
  <brieffopties>
\opening{<aanhef>}
  <tekst>
\closing{<groet>}
\end{letter}
\end{document}
```

De `<s-opties>` zijn dezelfde die we al eerder hebben behandeld onder [2.1.2](#).

Het spreekt vanzelf dat `<je naam>` je eigen naam is, en `<je adres>` je eigen adres. Die gegevens komen in principe op iedere brief weer terug, en daarom moet je ze helemaal aan het begin al opgeven. Je kunt `\\` commando's gebruiken om tekst over meerdere regels te splitsen.

De `\signature` is de ondertekening van de brief. Hoewel die pas aan het einde van een brief komt, moet je hem al voor de eigenlijke aanhef van de brief vermeld hebben. Eventueel kun je hier meer informatie kwijt, zoals je functie. Je kunt dan `\\` commando's gebruiken om die teksten op nieuwe regels te krijgen.

De brief zelf begint met een `\begin{letter}` commando, gevolgd door de adressering *tussen nieuwe accolade's*, en afgesloten met een `\end{letter}`. Je kunt in één bestand meerdere brieven opnemen, die allemaal op die manier worden begrensd. Iedere brief wordt op dezelfde manier afgesloten met de tekst die je met het commando `\signature` hebt aangegeven; als je een enkele brief anders wilt beëindigen, kun je een extra `\signature` opnemen direct ná het commando `\begin{letter}` waarmee de betrokken brief begint.

Onder `<naam>`, `<adres>` en `<plaats>` verstaan we het adres van degene aan wie de brief wordt gericht. De aanduidingen worden gescheiden door `\\` commando's om te zorgen dat ze op nieuwe regels komen.

De `<brieffopties>` zijn commando's die extra toeters en bellen aan je brief toevoegen. Je mag ze weglaten. Ze staan vermeld in tabel 10. De accolade's hebben we leeg gelaten, maar normaal gesproken vul je daar natuurlijk de juiste tekst in.

Tabel 10: Optionele commando's binnen de stijl `brief`

<code><brieffopties></code>	Toelichting:
<code>\date{}</code>	Geeft een datumaanduiding. Als je dit commando weglaat, krijg je de courante datum die de computer intern bijhoudt; een brief zonder datum krijg je dus door dit commando aan te roepen zonder tekst tussen de accolade's: <code>\date{}</code> .
<code>\cc{}</code>	Geeft de tekst onder 'cc' (staat voor <i>carbon copy</i>).
<code>\encl{}</code>	Geeft de tekst onder 'encl' (voor <i>enclosure</i>).
<code>\ps{}</code>	Geeft de tekst voor een p.s., maar <i>zonder</i> enige aanduiding — je zult dus zelf de tekst 'p.s.' moeten plaatsen.

De `<aanhef>` en `<groet>` bevatten de opening en afsluiting van de brief, dus teksten zoals 'Geachte meneer...' en 'Met vriendelijke groet.'. Tussen de `<groet>` en de tekst die wordt gegenereerd door `\signature` wordt automatisch ruimte opengelaten voor een handtekening.

De `<tekst>` tenslotte bevat de tekst van de brief. Als die tekst erg lang is en over meerdere pagina's moet worden verdeeld, dan krijgt alleen de eerste pagina het complete briefhoofd, en zijn volgende pagina's leeg. Iedere nieuwe brief in je bestand begint weer op een nieuwe pagina in het document.

Nota bene: je mag in een `letter` *geen* gebruik maken van de sectie-commando's, genoemd in tabel 9.

Nota bene (2): wij beschrijven hier de algemene vorm zoals die in standaard \LaTeX bestaat, maar het is mogelijk dat er op jouw systeem wijzigingen bestaan, of dat er een aparte stijl bestaat voor brieven met een specifiek logo.

4 De eerste stunts

In deze sectie behandelen we twee onderwerpen die een stuk ingewikkelder zijn: wiskundige tekst en tabellen. We behandelen die dan ook niet ten voeten uit. Als je er echt alles van wilt weten, of echt ingewikkelde stunts wilt uithalen, moet je de \LaTeX -boeken bestuderen.

Met name tabellen zijn erg gevoelig voor fouten. Je moet van tevoren vrij goed in je hoofd hebben zitten wat je van plan bent te gaan doen. Maar je moet ook niet vergeten bepaalde commando's op te nemen in je tabel; doe je dat niet of verkeerd, dan is de hoeveelheid foutmeldingen vaak niet te overzien.

4.1 Mathematische tekst

Het zetten van wiskundige tekst is binnen \LaTeX relatief eenvoudig ten opzichte van andere programma's (zoals bijvoorbeeld WORD). Wiskundige tekst wordt op een andere manier gezet dan gewone tekst. Met name de weergave van letters (meestal cursief), en de ruimte tussen afzonderlijke letters is heel anders. Het

woord *raffinaderij* in gewone cursieve tekst ziet er anders uit dan het woord *raffinaderij* in mathematische tekst (let op de combinatie ‘ffi’, ligatuur genoemd, en op de ruimte binnen de ‘lange ij’). Eén van de grootste verschillen is, dat in mathematische tekst geen rekening wordt gehouden met spaties die in het bestand staan; spaties anders dan die welke L^AT_EX zelf maakt, moet je expliciet aangeven.

Let op: L^AT_EX gebruikt een standaardfont om mathematische tekst te zetten, dat kan afwijken van de broodletter van het document. Alleen met pakketten zoals `txfonts` en `pxfonts` (zie tabel 3) wordt de gehele tekst in hetzelfde font gezet.

4.1.1 Soorten formules

We onderscheiden verschillende soorten mathematische tekst: *in-line* formules zijn opgenomen in de lopende tekst, en *displayed* formules staan apart in de tekst, meestal met witregels erboven en eronder, en soms voorzien van een rangnummer. Het verschil tussen de twee zit hem met name in de grootte van verschillende symbolen. Grote formules kun je in ieder geval beter als *displayed* opnemen, al was het alleen maar voor het overzicht. **Nota bene:** Een *displayed* formule mag nooit aan het begin van een alinea komen, dus in het bestand nooit voorafgegaan worden door een lege regel!

In-line formules worden omgeven door twee dollartekens: `<form>`. Een *displayed* formule kun je krijgen door er `[` voor te zetten en `]` achter. Als je er ook nog een rangnummer aan wilt toekennen, moet je in plaats daarvan respectievelijk `\begin{equation}` en `\end{equation}` zetten. Alle tekst tussen deze commando’s wordt als mathematische tekst gezet; men zegt dan dat L^AT_EX in *math mode* werkt.

4.1.2 Eenvoudige structuren

In *math mode* kent L^AT_EX een heleboel speciale tekens, die niet in gewone tekst kunnen worden opgenomen. Deze tekens staan gedeeltelijk in Appendix A, en in het L^AT_EX-boek vanaf pagina 40. Daar vind je ondermeer griekse letters (*ὦ τὸν κύνα*), maar ook allerlei symbolen. De al eerder behandelde ‘bullet’ • krijg je door in je bestand `\bullet` op te nemen.

In tabel 11 geven we een paar van de meest voorkomende structuren. We geven ook expliciete voorbeelden van structuren. We gaan er vanuit dat we al in *math mode* zijn, dus de omgevende commando’s, zoals `$`, hebben we weggelaten. Deze opsomming is uiteraard verre van compleet. In Appendix A vind je

Tabel 11: Veelvoorkomende structuren in *math mode*

Structuur:	Voorbeeld:	Resultaat en toelichting:
<code>^{}x</code>	<code>x^{2}</code>	x^2 , superscript.
<code>_{}x</code>	<code>x_{i}</code>	x_i , subscript.
<code>\frac{}{}</code>	<code>\frac{1}{2}</code>	$\frac{1}{2}$, een deling.
<code>\sqrt{}x</code>	<code>\sqrt{2}</code>	$\sqrt{2}$, een wortelteken.
<code>\sum</code>	<code>\sum^n_{i=1}</code>	$\sum_{i=1}^n$, een sommatie, hier in combinatie met een super- en een subscript.
<code>\int</code>	<code>\int_0^1</code>	\int_0^1 , een integraal, hier in combinatie met een super- en een subscript.
<code>\log</code>	<code>\log n</code>	$\log n$, een zogenaamde <i>log-like</i> functie, waarbij de functienaam anders is gezet.
<code>\sin</code>	<code>\sin \pi</code>	$\sin \pi$, een andere <i>log-like</i> functie.

veel meer commando’s (en in het L^AT_EX-boek, vanaf pagina 43, een volledig overzicht). In combinatie met de bovenstaande voorbeelden zou je dan een heel eind moeten komen. Merk op dat de manier waarop sommige symbolen worden gezet afhankelijk is van de soort. De *in-line* gezette formule $\int_0^\pi \frac{1}{\sin x} dx$ ziet er heel anders uit dan zijn eeneiige *displayed* tweelingbroertje, hoewel de achterliggende code hetzelfde is: `\int^{\pi}_{0} \frac{1}{\sin x} dx`

$$\int_0^\pi \frac{1}{\sin x} dx$$

4.1.3 Tekst in formules

Soms is het nodig om in een formule een tekst op te nemen. Omdat \LaTeX in *math mode* tekst op een andere manier zet dan meestal de bedoeling is, is het wenselijk om \LaTeX tijdelijk te laten denken dat tekst gewoon moet worden aangemaakt. Dat kun je door het commando `\mbox{}` te gebruiken, waarbij de tekst tussen de accolade's komt te staan. De volgende formule is op die manier gemaakt:

$$\sum_{\text{teller=ondergrens}}^{\text{bovengrens}} \text{ te sommeren termen}$$

De tekst in `\mbox{}` wordt gezet zoals gewone tekst, dus niet in *math mode*.

4.1.4 Spaties

In *math mode* maakt \LaTeX (vrijwel) geen spaties. Symbolen, wiskundige tekens zoals integralen, of variabelen worden ongeveer even dicht tegen elkaar aan gezet. Als je meer, of misschien juist minder spatiëring wilt, kun je gebruik maken van de in tabel 12 genoemde commando's.

Tabel 12: Spaties in *math mode*

Commando, voorbeeld en toelichting:		
<code>\!</code>	<code>00</code>	Een 'negatieve' spatie, een stapje terug dus.
	<code>00</code>	Ter vergelijking: een gewone spatie in <i>math mode</i> .
<code>\,</code>	<code>00</code>	Een dunne spatie, mag ook buiten <i>math mode</i> worden gebruikt.
<code>\:</code>	<code>00</code>	Een gemiddelde spatie.
<code>\;</code>	<code>00</code>	Een dikke spatie.
	<code>00</code>	Ter vergelijking: een spatie in gewone tekst.
<code>\quad</code>	<code>0 0</code>	Extra veel ruimte, ook buiten <i>math mode</i> te gebruiken.
<code>\qquad</code>	<code>0 0</code>	Dubbel extra veel ruimte, ook buiten <i>math mode</i> te gebruiken.

4.1.5 Kleine haken, grote haken

Wanneer je in een formule haken gebruikt, is het soms nodig om die haken groter te laten worden dan ze standaard zijn. De commando's `\left` en `\right` zorgen voor deze truc. Tegenover iedere `\left` moet een `\right` komen te staan, maar de haken die er op volgen hoeven niet van dezelfde soort te zijn. Als je maar één haak wilt hebben, hetzij aan de rechter- danwel aan de linkerkant, moet je het zustercommando laten volgen door een punt `.`. Dus `\left(...\right.` zorgt voor een aangepaste linkerhaak, terwijl rechts niets te zien is.

4.1.6 Een voorbeeld

We geven tenslotte als voorbeeld een uitgebreide formule. We hebben gewerkt met spaties, met linker- en rechterhaken, en met tekst. De tekst zoals die in het bestand staat, is eronder afgedrukt.

$$\left(\sum_{i=1}^n i^2\right) = \left(\sum_{i=1}^n i^2\right) \left[\neq \left(\sum_{i=1}^n i\right)^2\right] \quad \left. \vphantom{\sum_{i=1}^n i^2}} \right\} \text{een voorbeeld}$$

```
\left.{\sum_{i=1}^n i^2}\right\}\quad = \quad \quad
\left(\sum_{i=1}^n i^2\right)\quad
\left[\neq\quad \left(\sum_{i=1}^n i\right)^2\right];
\right]\}\quad\right\} \mbox{een voorbeeld}
```


4.2 Tabellen

De laatste structuur die we hier geven is die van de tabel. Hieronder volgt de basisopzet van een tabel:

```
\begin{tabular}{<formaat>}
<kol> & <kol> & <kol> \\
\hline
<kol> & <kol> & <kol> \\
\multicolumn{<n>}{<mulform>}{<mulkol>} & <kol> \\
\end{tabular}
```

Het <formaat> geeft aan hoeveel kolommen er zijn, en hoe die er komen uit te zien. Iedere kolom geef je aan door middel van een letter, en iedere letter geeft aan hoe tekst in die kolom moet worden gezet. Een l of r geeft aan dat de tekst respectievelijk tegen de linker- of rechterkant moet worden gezet, terwijl een c staat voor gecentreerde tekst. Als je ‘|’ tekens gebruikt, wordt er een verticale lijn getrokken. Dus de aanduiding { | r | l l | } staat voor drie kolommen, de eerste rechts en de rest links uitgelijnd, met een dubbele lijn aan de buitenkant van de tabel, en een enkele lijn tussen de eerste en de tweede kolom. Het aantal kolommen ligt hiermee voor de gehele tabel vast, hoewel er een klein beetje speelruimte is.

Vervolgens ga je de kolommen vullen, regel voor regel. De tekst voor iedere kolom wordt gescheiden door & commando’s, en de gehele regel wordt afgesloten met een \\ commando. Het aantal & commando’s is dus altijd één minder dan het aantal opgegeven kolommen. Hoewel wij dat in het voorbeeld wel hebben gedaan, hoef je in het bestand geen rekening te houden met hoe de kolommen er precies komen uit te zien. L^AT_EX doet dat zelf; de breedte van een kolom wordt bepaald door de breedste tekst in die kolom, niet door hoe breed jij de kolom hebt gemaakt in het bestand.

Het commando \hline zorgt voor een horizontale lijn over de gehele breedte van de tabel.

Als je tekst over meerdere kolommen wilt uitsmeren, kun je het \multicolumn{}{}{} commando gebruiken. De <n> geeft aan over hoeveel kolommen je de tekst uitsmeert. Dat kan natuurlijk nooit meer zijn dan het aantal kolommen dat je in <formaat> hebt opgegeven. Met <mulform> geef je aan hoe de tekst wordt uitgelijnd, op dezelfde manier als in <formaat>.

Soms is het nodig om in <mulform> ook | commando’s op te nemen. De exacte regels hiervoor zijn hier moeilijk te geven, maar in ieder geval moeten er na de overlappende tekst net zoveel |’s komen als er stonden na de oorspronkelijke laatste kolom die werd overlapt. Ook de eerste | van <formaat> moet worden aangegeven. Het totale aantal kolommen moet nog steeds gelijk zijn in een regel met een \multicolumn commando. Dus extra kolommen moeten nog wel worden toegevoegd als de overlappende kolom niet het totaal aantal kolommen beslaat.

Een voorbeeld tot slot:

```
\begin{tabular}{|r|ll|}
tekst1 & tekst2 & tekst3 \\ \hline
\multicolumn{2}{|c|}{tekst12} & tekst3 \\
tekst1 & \multicolumn{2}{c|}{tekst23} \\
\end{tabular}
```

tekst1	tekst2	tekst3
tekst12		tekst3
tekst1	tekst23	

Let bij het maken van een tabel steeds op de volgende zaken:

- Een tabel wordt gewoon opgenomen op de regel waar hij wordt gedefinieerd. Dat wil zeggen dat tussen tekst en een tabel minstens een lege regel moet zitten om de tabel in een aparte alinea te krijgen. Zet \noindent vóór de tabel om te voorkomen dat deze ‘inspringt’.
- Een tabel wordt niet automatisch in het midden van een bladzijde gezet. Daarvoor moet je vóór de tabel het commando \begin{center} en na de tabel \end{center} zetten.
- Iedere regel moet het juiste aantal kolommen bevatten en worden afgesloten met een \\ . Ook de laatste regel, zelfs als er een \hline commando op volgt.

Er kan veel meer met een tabel dan wij hier nu vertellen. Op pagina 62 van het L^AT_EX-boek, alsmede op pagina 204, beginnen vrij uitgebreide beschrijvingen ervan, die voorzien zijn van uitgewerkte voorbeelden.

5 Het afmaken van een document

Het maken van een groot document vereist nog wat meer kennis van \LaTeX en van *type-setting* dan wat je nu hebt (anders hoeft je deze tiplijst niet te lezen). Daarom volgen hier wat tips om je document er wat mooier uit te laten zien en om het maken ervan wat gemakkelijker te maken.

5.1 Typografie

Kleine dingetjes die vaak mis gaan zijn:

- Een spatie vóór een punt, komma, dubbele punt, puntkomma, uitroepetekens, vraagteken etcetera is fout in het Nederlands en Engels (niet in het Frans).
- Geen spatie na een punt, komma, dubbele punt, puntkomma, uitroepetekens, vraagteken etcetera is eveneens fout.
- In plaats van aanhalingstekens "dezelfde kant op" kun je beter aanhalingstekens "de andere kant op" gebruiken. Dit kan door ‘ ‘ in te typen als je de aanhalingstekens-openen wilt. Dit geldt ook voor enkele aanhalingstekens (‘). Gebruik nooit het symbool " dat op het toetsenbord voorkomt.
- Inspringen aan het begin van een lap tekst is lelijk. Gebruik in dat geval `\noindent` vóór de tekst.

Voorbeeld 1

Hier spring ik dus wel in. Dit is niet echt fout, want de tekst blijft gewoon leesbaar. Het is echter wel lelijk.

De volgende alinea moet je dus wel gewoon inspringen. Kijk eens in gewone boeken, daar gebeurt dit ook.

Voorbeeld 2

Hier spring ik dus niet in. Mooi hè? Bij zo'n klein stukje zie je het verschil niet zo goed, maar toch.

En bij de volgende alinea spring je dus gewoon wel weer in. Zo gaat het in alle boeken die door professionele zettters gemaakt zijn, en dat wil toch wel wat zeggen...

- Let goed op het gebruik van hoofdletters.
- Zorg dat er niet al te lange alinea's komen. Voor \LaTeX betekent een lege regel een nieuwe alinea. Gebruik dus géén dubbele backslashes (`\`), maar gewoon een lege regel. Als je een witregel wil, gebruik dan het commando `\bigskip` aan het einde van de alinea.

De lengte die \LaTeX gebruikt om in te springen heet de `\parindent`. De ruimte tussen twee alinea's heet de `\parskip`. Je kunt ze instellen boven `\begin{document}`. Neem bijvoorbeeld: `\parskip=4mm`.

5.2 Werken met meerdere bestanden

Als je een groot document maakt krijg je waarschijnlijk een heel groot bestand. Het is veel gemakkelijker als je ieder deel in een apart bestand schrijft en die in een totaalbestand weer opvraagt. Als je bijvoorbeeld de bestanden `inleidin.tex`, `begin.tex`, `programm.tex` en `sluiting.tex` hebt, die tezamen je document vormen, maak dan het volgende totaalbestand:

```
...
\begin{document}
\include{inleidin}
\include{begin}
\include{programm}
\include{sluiting}
\end{document}
```

Het commando `\include` zorgt ervoor dat ieder nieuw bestand dat je bijvoegt op een nieuwe pagina begint. Als je dat niet wilt, gebruik dan `\input` i.p.v. `\include`.

5.3 Macro's

Een belangrijk onderdeel bij het maken van een document is het consequent zijn. Dus niet: de ene keer de naam van een auteur in boldface (`\bfseries`) zetten en de andere keer in sans serif (`\sffamily`). Om dit eenvoudig consequent te kunnen doen en het ook nog eenvoudig te kunnen veranderen gebruik je macro's. Een macro definieer je in de `<preamble>`, dus vóór `\begin{document}`.

In het geval van de auteur van hierboven maak je een commando `\auteur`. Dit ziet er bijvoorbeeld zo uit:

```
\newcommand{\auteur}[1]{\medskip\par{\bfseries #1}}
```

De `\newcommand` is het commando om een macro te maken, met als parameter de naam van het nieuwe commando. Met `[1]` geef je de hoeveelheid parameters op (in dit geval één); in de definitie van het commando kun je elke parameter aanspreken met `#` gevolgd door het nummer van de parameter (hier dus `#1`). Als er geen parameters zijn laat je `[1]` dus weg. Daarna komt de definitie van het commando. In het voorbeeld is `\medskip` een extra ruimte tussen de tekst en de naam van de auteur, `\par` dient om de `\medskip` meteen te laten werken en `\bfseries` zet de naam van de auteur in boldface. In je \LaTeX -bestand zet je dan gewoon `\auteur{Jeroen van Dijk}`. Vind je de ruimte tussen de tekst en de auteur te groot? Of moet de naam toch in sans serif? Dat kun je nu zo aanpassen.

- Het commando `\newcommand` zorgt ervoor dat je geen bestaande commando's kunt overschrijven. Als je dat wel probeert te doen, krijg je een foutmelding. Als je er helemaal zeker van wilt zijn dat je geen schade aanricht, kun je bovendien als macronaam Nederlandse woorden aanhouden, dus bijvoorbeeld `\auteur` in plaats van `\author`.
- Wees een beetje voorzichtig met macro's die een heleboel tekst als parameter moeten gaan krijgen. Dit soort constructies is niet bedoeld om bijvoorbeeld een heel hoofdstuk in een bepaald lettertype te kunnen weergeven. Als je niet oppast, kan \LaTeX in geheugenproblemen komen.
- Als je veel macro's maakt, of macro's in meerdere bestanden wilt kunnen gebruiken, kun je ze het best in aparte bestanden plaatsen. Je kunt zo'n bestand dan inlezen met een `\input`. Als je het nog fraaiër wilt doen, kun je je eigen *packages* maken — maar tegen die tijd moet je maar eens in een echt \LaTeX -boek kijken.

Referenties

- [KD99] Helmut Kopka and Patrick W. Daley. *A Guide to \LaTeX , third edition*. Addison Wesley Longman Ltd., Harlow, Essex, England, 1999.
- [Lam94] Leslie Lamport. *\LaTeX , a Document Preparation System; User's Guide & Reference Manual, Second Edition*. Addison-Wesley, Reading, Mass., 1994.

\LaTeX op het web

<http://www.tug.org/>

A Alles op een rij

In dit overzicht worden de meest gebruikelijke L^AT_EX-structuren en -commando's gepresenteerd. Het overzicht is niet volledig; voor alle commando's zul je toch in L^AT_EX-boek moeten zijn. Sommige zaken zijn in deze handleiding niet uitgelegd, maar meestal lijken dergelijke commando's dan erg op welgenoemde commando's, en we zullen dan ook steeds op die gelijkenissen wijzen. De commando's zijn min of meer ingedeeld naar volgorde waarin ze in deze handleiding voorkomen.

Document en pagina's

`\documentclass[<s-opties>]{<stijl>}` De <stijl> is article, report, book, of letter. De <s-opties> zijn:

11pt, of 12pt groter lettertype.
a4paper, of a5paper papiergrootte.

`\pagestyle{<stijl>}` bepaalt de lay-out van een pagina; komt direct na `\documentclass`. De <stijl> is:

plain standaard.
empty geen paginanummers, geen kopteksten.
headings paginnummer plus de titel van de lopende sectie in de kop boven de pagina (zoals bij deze handleiding).

`\usepackage[<p-opties>]{<pakket>}` De belangrijkste pakketten zijn:

babel afbreekregels en labels in de juiste taal; de taal wordt in de <p-opties> gegeven, voor het Nederlands gebruik je `dutch`. Als je meerdere talen opgeeft, is de *laatst genoemde* de actieve taal voor het document.
amsmath uitgebreide bibliotheek van wiskundige symbolen en structuren.
txfonts, of pxfonts zet het document in Times of Palatino.
eurosym definieert allerhande €-symbolen, gebruik `\euro` voor het standaardsymbool.

`\maketitle` direct na `\begin{document}`, genereert tekst die in de <preamble> is vastgelegd met de commando's `\title{}`, `\author{}` en `\date{}`.

Speciale lettertekens

\$	<code>\\$</code>	&	<code>\&</code>	%	<code>\%</code>	#	<code>\#</code>
{	<code>\{</code>	}	<code>\}</code>	-	<code>_</code>		
†	<code>\dag</code>	‡	<code>\ddag</code>	£	<code>\pounds</code>		
§	<code>\S</code>	¶	<code>\P</code>	©	<code>\copyright</code>		

Lettertype en -grootte

Gebruik voor cursiveringen binnen een regel het commando `\emph{}`.

<code>\rmfamily</code>	Roman	<code>\upshape</code>	Up	<code>\mdseries</code>	Medium
<code>\sffamily</code>	Sans Serif	<code>\itshape</code>	<i>Italic</i>	<code>\bfseries</code>	Boldface
<code>\ttfamily</code>	Teletype	<code>\slshape</code>	<i>Slanted</i>		
		<code>\scshape</code>	SMALL CAPS		
<code>\tiny</code>		<code>\normalsize</code>	<code>\large</code>		
<code>\scriptsize</code>			<code>\Large</code>		
<code>\footnotesize</code>			<code>\LARGE</code>		
<code>\small</code>			<code>\huge</code>		
			<code>\Huge</code>		

Interpunctie en spaties

L^AT_EX herkent en interpreteert de gebruikelijke interpunctie, eventueel beïnvloed door het pakket babel. De volgende combinaties leveren speciale interpunctie op:

-	-	koppelteken	‘	‘	aanhalingsteken openen
--	-	intervalstreep (als in ‘1–3’)	“	“	idem dubbel
---	—	gedachtestreep	’	’	aanhalingsteken sluiten
\ldots	...	ellipsis	”	”	idem dubbel

Speciale gevallen van spaties:

- Punt midden in de zin, niet vooraf gegaan door een hoofdletter: \direct ná de punt.
- Punt aan het einde van een zin, vooraf gegaan door een hoofdletter: \@direct vóór de punt.
- Na een L^AT_EX-commando in een zin: \direct ná het commando, gevolgd door een spatie.
- Tussen een cursief en een niet-cursief woord, als je geen gebruik maakt van \emph{}: \direct ná het cursieve woord.
- Onbreekbare spatie tussen twee woorden: ~ tussen de twee woorden.

Afbreken

\- gebruikt binnen de tekst om aan te geven waar een woord mag worden afgebroken: af\-bre\-ken.

\hyphenation{<lijst>} wordt gebruikt in de <preamble>. De <lijst> is een lijst van woorden gescheiden door komma’s, waarin met commando’s is aangegeven waar er afgebroken mag worden. Dus \hyphenation{af-bre-ken,af-ge-bro-ken}.

\mbox{<woord>} wordt gebruikt in de lopende tekst om te voorkomen dat <woord> wordt afgebroken; dit kan tot gevolg hebben dat de kantlijn wordt overschreden (‘overfull hbox’).

\\ breekt de lopende regel af.

\nolinebreak voorkomt dat een regel op die plaats wordt afgebroken.

\newpage breekt de lopende pagina af.

\clearpage breekt de lopende pagina af, print alle tabellen en figuren die nog niet geprint waren.

\nopagebreak voorkomt dat de lopende pagina op die regel wordt afgebroken.

Accenten

Gebruik \i of \j voor een i of j zonder punt.

\’{o}	ó	\~{o}	õ	\v{o}	ö	\c{o}	ø
\‘{o}	ò	\={o}	ō	\H{o}	ő	\d{o}	ø
\" {o}	ö	\. {o}	ó	\t{oo}	ö	\b{o}	ø
\^ {o}	ô	\u{o}	ü	\r{o}	ö		

Secties

\part[<toc>]{<titel>} genummerd (boek)deel met <titel> als de titel. De tekst <toc> komt in de inhoudsopgave. Verwante commando’s zijn \chapter (niet in article), \section, \subsection, \subsubsection, \paragraph en \subparagraph.

\part*{<titel>} (boek)deel met <titel> als de titel *maar zonder nummering*. Idem voor alle bovengenoemde sectiecommando’s.

\appendix alle secties na dit commando genummerd als appendices (meestal met letters).

`\tableofcontents` inhoudsopgave; het bestand moet tenminste tweemaal ge \LaTeX d worden voor een correcte inhoudsopgave. Als de inhoudsopgave zelf meer dan twee bladzijden beslaat, moet je nóg een keer \LaTeX en.

Lijsten

Algemene vorm: `\begin{<lijst>} ... \end{<lijst>}`. De `<lijst>`: `description`, `itemize` en `enumerate`. Binnen iedere lijst: items scheiden met `\item`. Zie [3.7](#) voor details.

Speciale opmaak

Algemene vorm: `\begin{<lijst>} <tekst> \end{<lijst>}`. Deze *displayed paragraphs* werken als lijsten, maar zonder items. De `<lijst>` kan zijn:

quote korte citaten.

quotation langere citaten met aparte alinea's.

center gecentreerde regels; iedere gecentreerde regel afsluiten met `\\`.

verse poëzie; iedere regel afsluiten met een `\\`; coupletten scheiden door een lege regel; vóór een lege regel geen `\\`.

Brieven

In de `<preamble>` zet je `\signature{<je naam>}`. De brieven komen ná `\begin{document}`. Algemene vorm van een brief:

```
\begin{letter}{<naam>\\<adres>\\<plaats>}
  <brieffopties>
\opening{<aanhef>}
  <tekst>
\closing{<groet>}
\end{letter}
```

Voor briefopties, zie tabel [10](#)

Tabellen

Algemene vorm: `\begin{tabular}<formaat> ... \end{tabular}`. Het `<formaat>`: `l`, `r` of `c`; `|` voor een verticale lijn. Kolommen scheiden met `&`, regels met `\\`. Gebruik `\hline` voor een horizontale lijn, `\multicolumn{<n>}{<formaat>}{ ... }` voor tekst over n kolommen. Zie [4.2](#).

Math Mode

\$... \$ of \ (... \)	<i>in-line</i> formule.
\ [... \]	<i>displayed</i> formule.
\begin{equation} ... \end{equation}	genummerde <i>displayed</i> formule.
\mbox{<tekst>}	<tekst> correct gezet in een formule.
_ { ... } en ^ { ... }	subscript _{onder} en superscript ^{boven} .
\frac{<n>}{<d>}	deling $\frac{n}{d}$.
\sqrt[<n>]{<arg>}	wortel $\sqrt[<n>]{arg}$.
\overline{<expr>} en \underline{<expr>}	streep boven \overline{expr} en onder \underline{expr} .
\vec{<x>}	vector \vec{x} .
\not<symbool>, bijv. \$x\not<y\$	negatie volgend symbool: $x \not\prec y$.
,	een <i>prime</i> : x' .
\ldots, \cdots, \ddots, \vdots	diverse puntjes: \dots, \cdots, \ddots en \vdots .
Spaties: negatief \! dun \, gemiddeld \: dik \; of \quad of \qquad	

In de volgende tabellen zijn sommige symbolen gemerkt met een *; deze symbolen zijn alleen te gebruiken als je het pakket latexsym of het pakket amsfonts hebt geladen.

Griekse letters klein

α	\alpha	θ	\theta	o	o	τ	\tau
β	\beta	ϑ	\vartheta	π	\pi	υ	\upsilon
γ	\gamma	ι	\iota	ϖ	\varpi	ϕ	\phi
δ	\delta	κ	\kappa	ρ	\rho	φ	\varphi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	χ	\chi
ε	\varepsilon	μ	\mu	σ	\sigma	ψ	\psi
ζ	\zeta	ν	\nu	ς	\varsigma	ω	\omega
η	\eta	ξ	\xi				

Griekse letters groot

Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

Binaire operatoren

\pm	\pm	\cap	\cap	\circ	\circ	\bigcirc	\bigcirc
\mp	\mp	\cup	\cup	\bullet	\bullet	\square	\square*
\times	\times	\uplus	\uplus	\diamond	\diamond	\diamond	\Diamond*
\div	\div	\sqcap	\sqcap	\triangleleft	\triangleleft*	\triangle	\bigtriangleup
\cdot	\cdot	\sqcup	\sqcup	\triangleright	\triangleright*	∇	\bigtriangledown
$*$	\ast	\vee	\vee	\trianglelefteq	\trianglelefteq*	\triangleleft	\triangleleft
\star	\star	\wedge	\wedge	\trianglerighteq	\trianglerighteq*	\triangleright	\triangleright
\dagger	\dagger	\oplus	\oplus	\oslash	\oslash	\setminus	\setminus
\ddagger	\ddagger	\ominus	\ominus	\odot	\odot	\wr	\wr
\amalg	\amalg	\otimes	\otimes				

Relaties

\leq	\leq	\geq	\geq	\neq	\neq	\sim	\sim
\ll	\ll	\gg	\gg	\doteq	\doteq	\simeq	\simeq
\subset	\subset	\supset	\supset	\approx	\approx	\asymp	\asymp
\subseteq	\subseteq	\supseteq	\supseteq	\cong	\cong	\smile	\smile
\sqsubset	\sqsubset*	\sqsupset	\sqsupset*	\equiv	\equiv	\frown	\frown
\sqsubseteq	\sqsubseteq	\sqsupseteq	\sqsupseteq	\propto	\propto	\bowtie	\bowtie
\in	\in	\ni	\ni	\prec	\prec	\succ	\succ
\vdash	\vdash	\dashv	\dashv	\preceq	\preceq	\succeq	\succeq
\vDash	\models	\perp	\perp	\parallel	\parallel	\mid	\mid

Pijlen

\leftarrow	<code>\leftarrow</code>	<code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	<code>\to</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>		\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>		\Rrightarrow	<code>\Rrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>		\longleftrightarrow	<code>\longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>		\Llongleftrightarrow	<code>\Llongleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>		\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>		\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>		\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>		\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>		\leadsto	<code>\leadsto*</code>		

Diversen

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>	\Box	<code>\Box*</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\exists	<code>\exists</code>	\Diamond	<code>\Diamond*</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\neg	<code>\neg</code>	\triangle	<code>\triangle</code>
\jmath	<code>\jmath</code>	$\sqrt{\quad}$	<code>\sqrt{\quad}</code>	\flat	<code>\flat</code>	\flat	<code>\flat</code>	\clubsuit	<code>\clubsuit</code>
ℓ	<code>\ell</code>	∂	<code>\partial</code>	\natural	<code>\natural</code>	\natural	<code>\natural</code>	\diamondsuit	<code>\diamondsuit</code>
\wp	<code>\wp</code>	\top	<code>\top</code>	\sharp	<code>\sharp</code>	\sharp	<code>\sharp</code>	\heartsuit	<code>\heartsuit</code>
\Re	<code>\Re</code>	\perp	<code>\perp</code>	$\ $	<code>\ </code>	$\ $	<code>\ </code>	\spadesuit	<code>\spadesuit</code>
\Im	<code>\Im</code>	\vdash	<code>\vdash</code>	\angle	<code>\angle</code>	\angle	<code>\angle</code>	\Join	<code>\Join*</code>
\mho	<code>\mho*</code>	\dashv	<code>\dashv</code>	\backslash	<code>\backslash</code>	\backslash	<code>\backslash</code>	∞	<code>\infty</code>

'log-like' functies

Zie tabel 11 voor voorbeelden.

<code>arccos</code>	<code>\arccos</code>	<code>csc</code>	<code>\csc</code>	<code>ker</code>	<code>\ker</code>	<code>min</code>	<code>\min</code>
<code>arcsin</code>	<code>\arcsin</code>	<code>deg</code>	<code>\deg</code>	<code>lg</code>	<code>\lg</code>	<code>Pr</code>	<code>\Pr</code>
<code>arctan</code>	<code>\arctan</code>	<code>det</code>	<code>\det</code>	<code>lim</code>	<code>\lim</code>	<code>sec</code>	<code>\sec</code>
<code>arg</code>	<code>\arg</code>	<code>dim</code>	<code>\dim</code>	<code>lim inf</code>	<code>\liminf</code>	<code>sin</code>	<code>\sin</code>
<code>cos</code>	<code>\cos</code>	<code>exp</code>	<code>\exp</code>	<code>lim sup</code>	<code>\limsup</code>	<code>sinh</code>	<code>\sinh</code>
<code>cosh</code>	<code>\cosh</code>	<code>gcd</code>	<code>\gcd</code>	<code>ln</code>	<code>\ln</code>	<code>sup</code>	<code>\sup</code>
<code>cot</code>	<code>\cot</code>	<code>hom</code>	<code>\hom</code>	<code>log</code>	<code>\log</code>	<code>tan</code>	<code>\tan</code>
<code>coth</code>	<code>\coth</code>	<code>inf</code>	<code>\inf</code>	<code>max</code>	<code>\max</code>	<code>tanh</code>	<code>\tanh</code>

Schaalbare symbolen

Onderstaande symbolen worden geschaald naar gebruik in *in-line* of *displayed* formule.

Σ	<code>\sum</code>	\cap	<code>\bigcap</code>	\odot	<code>\bigodot</code>
\prod	<code>\prod</code>	\cup	<code>\bigcup</code>	\otimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\sqcup	<code>\bigsqcup</code>	\oplus	<code>\bigoplus</code>
\int	<code>\int</code>	\vee	<code>\bigvee</code>	\uplus	<code>\biguplus</code>
\oint	<code>\oint</code>	\wedge	<code>\bigwedge</code>		

Schaalbare haken

Onderstaande symbolen worden geschaald met `\left` en `\right`; zie 4.1.5.

$($	<code>(</code>	$)$	<code>)</code>	\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>
$[$	<code>[</code>	$]$	<code>]</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$\{$	<code>\{</code>	$\}$	<code>\}</code>	\langle	<code>\langle</code>	\rangle	<code>\rangle</code>
$ $	<code> </code>	$\ $	<code>\ </code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
				\Updownarrow	<code>\Updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>

B Begrippen

ASCII-FILE Een tekst die alleen maar bestaat uit de karakters die op het toetsenbord aanwezig zijn. In een ASCII-file komt geen vette tekst voor, geen onderlijnde tekst, etcetera. Een speciaal geval is de ‘kale ASCII-file’, zie aldaar.

AUXILIARY FILES Worden door \LaTeX aangemaakt naast het eigenlijke document. Ze hebben dezelfde naam als het bestand, maar steeds andere extensies. Ze worden bijvoorbeeld gebruikt voor het bijhouden van een inhoudsopgave of een bibliografie.

BESTAND Met het bestand bedoelen we steeds het tekst-bestand, of delen daarvan, dat je zelf schrijft, en waarin \LaTeX -commando’s zijn opgenomen. Het bestand is een file met een naam en de extensie `.tex`, dus bijvoorbeeld `verslag.tex`.

DISPLAYED FORMULE Een formule die apart is gezet in de tekst, meestal gecentreerd op de pagina, met witregels erboven en eronder, en soms voorzien van een (volg)nummer. De tegenhanger is de ‘in-line formule’, zie ook aldaar.

DISPLAYED PARAGRAPH Tekst die apart is gezet, meestal met witregels erboven en eronder, soms met inspringende kantlijn.

DOCUMENT Met het document bedoelen we het eindresultaat dat je krijgt als je het bestand hebt ge \LaTeX d, en dat je naar de printer stuurt, of naar het beeldscherm. Met het document bedoelen we zowel de DVI-file als de uiteindelijke geprinte versie.

DVI-FILE De DVI-file is wat er ontstaat als je het bestand ge \LaTeX d hebt. DVI staat voor ‘device independent’, wat wil zeggen dat je onafhankelijk van het gebruikte medium (scherm, (soort) printer) toch altijd hetzelfde resultaat krijgt. De DVI-file heeft dezelfde naam als het bestand, maar de extensie `.dvi`. Met pdf \LaTeX (zie aldaar) kun je in plaats van een DVI-file een PDF-file maken.

EDITOR Een eenvoudige tekstverwerker, geschikt voor het maken van ASCII-files (zie aldaar).

ENVIRONMENT Een structuur, begrensd door `\begin{}` en `\end{}` commando’s. Binnen de ‘scope’ (zie aldaar) van een environment gelden soms speciale commando’s die daarbuiten niet bestaan, of zijn bestaande commando’s anders gedefinieerd.

EXTENSIE Het laatste deel van een file-naam. In `verslag.tex` is `.tex` de extensie. Een extensie bestaat uit een punt en maximaal drie karakters.

HULPBESTANDEN Zie bij ‘auxiliary files’.

IN-LINE FORMULE Een formule die is opgenomen in de lopende regel. De mathematische tekst is in deze toestand zoveel mogelijk aangepast aan de grootte van een normale regel tekst. Tegenhanger is de ‘displayed formule’, zie aldaar.

KALE ASCII-FILE Een tekst die alleen maar bestaat uit de karakters die op het toetsenbord aanwezig zijn. Letters die zijn gemaakt in combinatie met een `Alt` toets (zoals `Alt-137` ë) vallen hier niet onder. \LaTeX slaat alle karakters over die niet als kale ASCII-tekst worden gezien.

\LaTeX EN Iedere activiteit, in de ruimste zin van het woord, die te maken heeft met \LaTeX . In meer engere zin: het verwerken van een bestand tot een document, door middel van het aanroepen van het programma `latex`.

LOG-FILE Een file waarin \LaTeX bijhoudt wat er tijdens het maken van het document is gebeurd.

MATH MODE De toestand waarin \LaTeX verkeert als er mathematische tekst wordt gezet. Tekst wordt op een andere manier gezet in math mode dan daarbuiten.

OMGEVING Zie ‘environment’.

PDF \LaTeX Een variant van \LaTeX waarmee je geen DVI-file maakt maar een PDF-file, te bekijken met het programma ACROBAT.

PREAMBLE Het stuk van het bestand, vóór het `\begin{document}` commando, dat bedoeld is voor declaraties. Hier kan bijvoorbeeld worden aangegeven hoe groot de pagina moet worden, of hoeveel een nieuwe alinea inspringt. Niet bedoeld voor gewone tekst.

SCOPE Het ‘bereik’ van een bepaald commando; binnen de scope is het commando werkzaam, daarbuiten niet. De scope wordt aangegeven met accolade’s `{}` of door `\begin{}` en `\end{}` commando’s; in het laatste geval spreekt men van een ‘environment’ (zie aldaar).

SOURCEFILE Een file die als basis dient voor bewerking. Het bestand waarover wij steeds spreken is een *sourcefile*, een PASCAL-programma dat moet worden gecompileerd is ook een *sourcefile*.

TEXEN Jargon. Zoals ‘ \LaTeX en’ staat voor het werken met \LaTeX , zo staat \TeX en voor het werken met \TeX , het programma dat de basis vormt van \LaTeX .

VIEWER Een programma dat een \LaTeX -document op het scherm laat zien, in de vorm waarin het ook uit de printer komt rollen.

WERKOMGEVING Een computerprogramma waarbinnen je diverse taken kunt uitvoeren.