

ConTEXt **de handleiding**

Hans Hagen
12 november 2001

This document is typeset using PDF \TeX . The body font is Lucida Bright at 10 points. The layout is based on a 24 cm square paper size, scaled down .875 to fit nicely on an A4.

\TeX and $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ are trademarks of the American Mathematical Society; METAFONT is a trademark of Addison-Wesley Publishing Company; PostScript, Portable Document Format and Acrobat are trademarks of Adobe Systems Incorporated; DVIPSONE and DVIWINDO are trademarks of Y&Y Incorporated; IBM is a trademark of International Business Machines Corporation; MSDOS is a trademark of MicroSoft Corporation; all other product names are trademarks of their producers.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the publisher.

© 1991-2001 PRAGMA ADE, Ridderstraat 27, 8061GH Hasselt, The Netherlands, pragma@wxs.nl

Inhoud

Inhoud

	Woord vooraf	3
1	Inleiding	5
	1 T _E X 5 2 CON _T E _X T 5 3 Commando's 6 4 Programma's 10 5 Files 10 6 Teksten 11 7 Foutmeldingen 12 8 Versienummers 12 9 Top 10 13 10 Waarschuwing 13	
2	Documenten	15
	1 Inleiding 15 2 Starten en stoppen 15 3 Structuur 16 4 Directories 20 5 Versies 21 6 Modes 21	
3	Vlakverdeling	23
	1 Inleiding 23 2 Papierformaat 23 3 Bladindeling 24 4 Grids 36 5 Printen 39 6 Arrangeren 42 7 Beeldmerken 51	
4	Layout	55
	1 Inleiding 55 2 Alineas 55 3 Interlinie 55 4 Spatiëring 64 5 Struts 64 6 Margeteksten 65 7 Verhogen en verlagen 68 8 Kolommen 69 9 Alinea's in kolommen 73 10 Tabulatie 76 11 Uitlijnen 77 12 Regelovergangen 80 13 Pagineren 82 14 Paginanummers 83 15 Hoofd- en voetteksten 85 16 Voetnoten 89 17 Uitgelijnde boxen 92 18 Opmaken 94	
5	Typografie	97
	1 Inleiding 97 2 Het mechanisme 99 3 Omschakelen 100 4 Letters 102 5 Beschikbare alternatieven 102 6 Benadrukken 103 7 Kapitalen 104 8 Getypte tekst 106 9 Wiskunde 110 10 Em en Ex 112 11 Definities 112 12 Omhullende tekst 119 13 Files 119 14 Figuren 119	
6	Kleur en achtergrond	123
	1 Inleiding 123 2 Kleur 123 3 Grijswaarden 126 4 Kleurgroepen en paletten 127 5 Tekst-achtergronden 131 6 Layout-achtergronden 133 7 Overlays 134 8 META-POST 135	
7	Taalspecifieke zaken	137
	1 Inleiding 137 2 Automatisch afbreken 137 3 Definities en instellingen 139 4 Datum 141 5 Labels en koppen 142 6 Taal gebonden commando's 143 7 Automatisch vertalen 143 8 Samengestelde woorden 144	
8	Tekstonderdelen	145
	1 Inleiding 145 2 Onderverdeling 146 3 Variaties in koppen 149 4 Meta-structuur 154 5 Alternatieve mechanismen 155	

9 Verwijzingen 161

1 Inhoudsopgave 161 2 Synoniemen 172 3 Sorteren 175 4 Markeringen 176
5 Terugverwijzen (refereren) 180 6 Regelnummers 184 7 Voorgedefinieerde verwij-
zingen 185 8 Registers 186

10 Beschrijvingen 193

1 Inleiding 193 2 Definities 193 3 Doornummeren 196 4 Doorspringen 200
5 Doorlabelen 201 6 Opsommingen 202 7 Items 210 8 Citaten 212

11 Lijnen en kaders 215

1 Inleiding 215 2 Losse lijnen 215 3 Invullijnen 217 4 Tekstlijnen 219 5 Onder-
strepen 220 6 Omlijnen 222 7 Kaderteksten 228 8 Kantlijnen 231 9 Blokjes 232
10 Roosters 233

A Definities 235**B Index 265****C Commando's 269**

**Woord
vooraf**

Woord vooraf

Deze handleiding gaat over $\text{CON}\text{T}\text{E}\text{X}\text{T}$, een systeem voor het opmaken van teksten. Centraal element in de naam is het woordje TEX . De typografische programmeeromgeving TEX vormt namelijk de kern van $\text{CON}\text{T}\text{E}\text{X}\text{T}$.

Een kenner zal waarschijnlijk aan deze handleiding zien dat ze is opgemaakt met behulp van TEX . Hem of haar valt direct het gebruik op van de voor dit systeem zo karakteristieke \backslash . Verder zijn de regels over het algemeen beter afgebroken dan in het gemiddelde opmaakstelsel.

We gaan in deze handleiding niet al te diep in op TEX zelf, omdat er voldoende goede boeken op dit terrein voorhanden zijn, waaronder:

1. het onovertroffen *The TEX Book* van Donald E. Knuth, de bron van alle kennis en TEX nische inspiratie
2. het overzichtelijke *TEX by Topic* van Victor Eijkhout, hét naslagwerk bij uitstek voor TEX -programmeurs
3. de aanrader *The Beginners Book of TEX* van Silvio Levy en Raymond Seroul, het boek dat van iedere beginner een expert maakt

Nieuwelingen raden we (3) aan, nieuwsgierigen (1) en de ongeduldigen (2). Gebruikers van $\text{CON}\text{T}\text{E}\text{X}\text{T}$ kunnen het in principe stellen zonder aanvullende literatuur, tenzij men zelf in TEX wil programmeren, gebruik maakt van bijzondere karakters, of wiskundige formules wil zetten. In dat geval raden we wederom (3) aan.

Waarom zou men TEX gebruiken? Laten we het erop houden dat TEX over zeer goede typografische routines beschikt en dat bovendien TEX niet alleen programmeerbaar is, en dus flexibel, maar dat het ons ook voldoende mogelijkheden biedt om met de tijd mee te gaan. TEX is echter geen systeem waarmee men als beginner even snel een tekst in elkaar zet. Maar, als men er eenmaal mee heeft leren werken, wil men zelden nog wat anders.

In deze handleiding wordt de basisfunctionaliteit beschreven. Voor een wat eenvoudiger inleiding op $\text{CON}\text{T}\text{E}\text{X}\text{T}$ verwijzen we naar $\text{CON}\text{T}\text{E}\text{X}\text{T}$, *een excursie*. Modules, zoals die voor het zetten van chemische structuurformules en SI-eenheden, worden beschreven in aparte handleidingen. Meer voorbeelden, achtergrondinformatie en aanvullende functionaliteit zijn te vinden in de sourcecode. Verder is er een Quick Reference Guide beschikbaar en een meer uitgebreide functionele specificatie in ontwikkeling. De laatste ontwikkelingen en speciale onderwerpen worden behandeld in de $\text{CON}\text{T}\text{E}\text{X}\text{T}$ Up-To-Date documenten. Er zijn een aantal gedocumenteerde stijlen en meer zijn in voorbereiding. Er is kortom naast deze handleiding nog meer leesvoer.

Hans Hagen, 1996–2001

1

Inleiding

Inleiding

1.1 T_EX

T_EX is begin jaren '80 ontwikkeld aan de Stanford University in de USA. De geestelijk vader, ontwerper en ontwikkelaar van T_EX is Donald E. Knuth. T_EX is door Knuth ontwikkeld om zijn eigen publicaties vorm te geven en een voorbeeld te geven van een systematisch ontwikkeld en beschreven programma. Het project waarin T_EX is ontwikkeld, is gedragen door de American Mathematical Society en heeft geresulteerd in de typografische programmeertaal T_EX en het programma T_EX, de grafische programmeertaal METAFONT en het programma METAFONT, de Computer Modern Typefaces en een aantal hier niet te noemen tools en publicaties. In het verlengde van METAFONT heeft John Hobby METAPOST geschreven, een onmisbaar en minstens zo leuk programma als T_EX.

Hoewel T_EX verbazingwekkend flexibel is gebleken, is toch besloten tot uitbreidingen. Onder de naam ϵ -T_EX zijn en worden uitbreidingen op T_EX geboden, en PDF T_EX biedt naast DVI uitvoer ook PDF. Beiden komen samen in PDF- ϵ -T_EX.

T_EX wordt over de hele wereld toegepast, ondersteunt erg veel talen, draait op vrijwel alle systemen, is 'public domain' en is stabiel, iets dat in de wereld van computers uniek is! T_EX is een zetsysteem dat een tekst in batch verwerkt. Dat wil zeggen dat een ruwe tekst met opmaakcommando's van voor naar achter wordt doorlopen en verwerkt. Omdat de opmaakcommando's aangeven wat we willen zetten, spreken we wel over een intentioneel zetsysteem. Naast dergelijke systemen onderscheiden we pagina-opmaaksystemen. Hierbij worden (meestal) interactief op een pagina de verschillende tekstelementen geplaatst.

Waar een pagina-opmaakstelsel zijn sterke punten toont bij het opmaken van documenten van geringe omvang en complexe layout, blinkt een intentioneel systeem uit in het verwerken van omvangrijke, aan wijzigingen onderhevige documenten. Bovendien ondersteunt een intentioneel systeem consistent gebruik van (snel te wijzigen) layoutkenmerken en documentoverstijgende elementen, zoals een index en een inhoudsopgave. Tot slot zijn intentionele systemen zeer flexibel en kunnen zij goed interacteren met hulp-systemen.

1.2 CON_TE_XT

De ontwikkeling van het macro-pakket CON_TE_XT is in 1990 gestart. De functionaliteit van CON_TE_XT is grotendeels bepaald door het gebruik van T_EX voor opleidingsmateriaal en (werkplek)manuals. In de loop der jaren is de functionaliteit sterk uitgebreid om in 1994 steeds meer te stabiliseren. Dit betekent niet dat de ontwikkeling stil staat. Er ligt nog een hele verlanglijst! De meeste uitbreidingen zullen overigens plaatsvinden binnen bestaande commando's. Op dit moment biedt CON_TE_XT als alternatief op de nederlandstalige ook een engelse en duitse gebruikersinterface, terwijl andere interfaces in voorbereiding zijn.

Het fileformaat PDF heeft voordeel dat in een file alle typografische informatie, font, afbeeldingen enz. zijn opgenomen. Bovendien kan men zeer geavanceerde interactiviteit inbouwen.

Vanzelfsprekend ondersteunt `CONTEXT` dan ook PDF als uitvoer formaat, ofwel als nabewerking van DVI, ofwel direct door middel van `PDFTEX`.

Daarnaast wordt bij gebruik van ϵ -`TEX`, teruggevallen op een aantal mogelijkheden die in dit programma worden geboden. De gebruiker hoeft hier overigens niets voor te doen, het gaat vanzelf. De auteurs van deze handleiding gebruiken dan ook `PDF- ϵ -TEX`.

In tegenstelling tot andere macropaketten, is `CONTEXT` een geïntegreerd systeem: de hele functionaliteit is altijd beschikbaar en men hoeft zich dus niet te verdiepen in allerlei modules die dit of dat bieden. Een direct gevolg is dat de bediening van `CONTEXT` relatief consistent is.

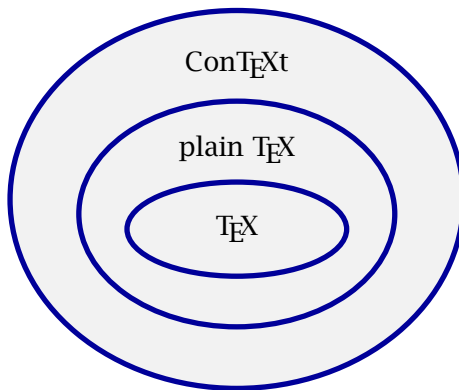
Dit wil niet zeggen dat er geen aanvullende functionaliteit is. Zo is er het macropakket `PPCHTEX`, waarmee men chemische structuurformules kan zetten. Daarnaast zijn er aparte wiskundige en natuurkundige modules. Deze bieden mogelijkheden die zeer specifiek zijn voor een vakgebied. Het aantal modules is en wordt echter beperkt tot zeer gerichte functionaliteit.

Daarnaast zijn er stijlen. In principe bevat een stijl alleen layout definities en eventueel enkele macro definities. Als het goed is vindt men in zo'n file alleen macros die inhaken op bestaande macros, bijvoorbeeld een macro die koppen op een bijzondere manier weergeeft en die wordt gekoppeld aan het mechanisme dat koppen plaatst.

1.3 Commando's

Een `TEX`-tekst is een tekst waarin commando's zijn opgenomen die vertellen hoe de tekst eruit dient te zien. Een voorbeeld van een commando is `\s1`. Een `TEX`-commando begint met een backslash (`\`) en doet meestal iets met de tekst die erop volgt. Zo wordt na het commando `\s1` de tekst schuin gedrukt ofwel: *slanted*.

Als we commando's als `\s1` gebruiken zijn we bezig met vormgeven. Als degene die de tekst invoert in de computer ook de auteur is, leiden dergelijke opmaakhandelingen af van het schrijven zelf. Beter is het in dat geval commando's te gebruiken die aangeven dat iets in de tekst moet opvallen. Een voorbeeld van zo'n commando is `\em` (*emphasis*). Voordeel van een benaming die het doel aangeeft is, dat we later eenvoudig de betekenis van een commando als `\em` kunnen wijzigen, zonder dat we in de tekst hoeven in te grijpen. `TEX` is namelijk programmeerbaar! We zullen zien dat het merendeel van de in deze handleiding beschreven commando's dan ook niet zozeer een vormgeving aanduiden maar eerder het doel van een stuk tekst.



Figuur 1.1

TeX-gebruikers spreken meestal niet over commando's maar over macro's. Een macro vat een aantal zetopdrachten samen. Hoewel we soms de termen door elkaar gebruiken, hebben we een voorkeur voor het woord commando als we ons in de rol van gebruiker bevinden en voor het woord macro als we programmeur zijn. Een verzameling macro's duiden we aan als macro-pakket. Het hier beschreven macro-pakket CONTEXt is een van de meest uitgebreide en volledige pakketten. Binnen CONTEXt zijn de door Knuth geschreven plain TeX macro's grotendeels beschikbaar.

Bij veel commando's is wat er op volgt van wezenlijk belang. Dit kunnen instellingen zijn of delen van de tekst waarop het commando betrekking heeft. Instellingen worden tussen [] opgegeven. De (tekstuele) informatie waartoe het commando zich beperkt, wordt tussen opgegeven, bijvoorbeeld:

```
\omlijnd[breedte=2cm,hoogte=1cm]{ziezo}
```

Dergelijke commando's maken deel uit van de tekst. Zouden we zo'n tekst letterlijk afdrukken, dan krijgen we naast de tekst zelf, de commando's te zien. Om het gewenste resultaat te bereiken, moeten we een tekst dus laten bewerken, in dit geval door TeX. Het bovenstaande levert op:

ziezo

Instellingen vinden in CONTEXt vrijwel altijd plaats door middel van commando's met de vorm:

```
\stelietsin[variabele=waarde,variabele=waarde,...]
```

of

```
\stelietsin[optie,optie,...]
```

In beide gevallen staan de instellingen tussen []. Er kunnen meerdere instellingen tegelijk, gescheiden door een comma, worden opgegeven. Enkele voorbeelden van instellingen zijn:

```
\stelwitruimtein[groot]
```

```
\stelopsommingin[opeelkaar,kolommen]
```

```
\stellayoutin[rugwit=4cm,kopwit=2.5cm]
```

Daarnaast kennen we definities. Deze hebben als vorm:

```
\definieeriets[naam]
```

Sommige definities bouwen voort op bestaande functionaliteit. In dat geval zien we definities als:

```
\definieeriets[kind][ouder]
```

In een aantal gevallen kunnen we instellingen meegeven:

```
\definieeriets[naam][variabele=waarde,...]
```

Vrijwel altijd kunnen de instellingen ook in een later stadium plaatsvinden met:

```
\stelietsin[naam][variabele=waarde,...]
```

Een voorbeeld van zo'n aan een 'naam' gekoppelde instelling is:

```
\stelkopin[hoofdstuk][letter=vet]
```

Hoewel hiermee de belangrijkste varianten van de interface gegeven zijn, gebiedt de eerlijkheid te zeggen dat er op deze regels uitzonderingen zijn. Bijvoorbeeld:

```
\doornummeren[Vraag][plaats=inmarge]
\gebruikexternfiguur[Logo][FIG-0001][breedte=4em]
\definieerkop[Rubriek][paragraaf]
\stelkopin[Rubriek][letter=schuin]
```

Na het eerste commando is (onder andere) het commando `\Vraag` beschikbaar waarmee vragen kunnen worden genummerd, waarbij het nummer in de marge wordt geplaatst. Met het tweede commando wordt een buiten \TeX aangemaakt figuur gedefinieerd, waarbij `breedte` betrekking heeft op de breedte in de tekst, in ons geval 4cm. Na het derde commando is het commando `\Rubriek` beschikbaar, dat zijn eigenschappen erft van het commando `\paragraaf`. Met het laatste commando wordt een van de eigenschappen van de zojuist gedefinieerde kop veranderd.

Veel typografische wensen zijn alleen te realiseren als gebruik wordt gemaakt van `\start-``\stop-`constructies:

```
\startiets
.....
\stopiets
```

Vaak kunnen ook hier instellingen worden meegegeven:

```
\startsmaller[2*links,rechts]
.....
\stopsmaller
```

of

```
\startopsomming[n,ruim,opeelkaar]
\som .....
\som .....
\stopopsomming
```

Daarnaast is er een `\begin-``\eind-`constructie, die wordt gebruikt voor het markeren van tekstblokken. Gemarkeerde tekstblokken kunnen worden gezet, verborgen, gezet maar verborgen, verplaatst en/of elders worden opgeroepen.

```
\beginvaniets
.....
\eindvaniets
```

Dergelijke blokken dient men eerst te definiëren:

```
\definieerblok[antwoord]
\stelblokin[antwoord][korps=klein]
\verbergblokken[antwoord]
.....
\hoofdstuk{.....}
.....
\beginvanantwoord
.....
\eindvanantwoord
.....
```

Commando's als de bovenstaande maken het mogelijk antwoorden in de ruwe tekst op te nemen en op de gewenste plaats op te roepen. Zo kunnen de antwoorden aan het eind van het hoofdstuk worden opgeroepen met:

```
\selecteerblokken[antwoord][criterium=hoofdstuk]
```

Er zijn nog vele andere vormen waarin commando's kunnen worden (en ook zijn) gegoten. Neem bijvoorbeeld:

```
\plaatsfiguur
  [links]
  [fig:logo]
  {Dit is een voorbeeld van een logo.}
  {\externfiguur[Logo]}
```

Dit commando plaatst een figuur links naast de tekst, waarbij de tekst rond de figuur loopt. Deze figuur heeft als referentie `fig:logo`. Het derde argument is de titel. Als hier geen wordt gegeven, blijft de titel achterwege. Het laatste argument is de figuur zelf, in dit geval het eerder gedefinieerde logo. Figuur 1.1 is op een dergelijke manier geplaatst.

In het laatstgenoemde commando zijn argumenten tussen `[]` optioneel. Dit is het geval bij meer commando's. Zo kan overal waar dat relevant is een referentie tussen `[]` worden meegegeven. Bovendien is een ruime layout van de commando's toegestaan.

In deze handleiding wordt de vorm van commando's op de volgende manier weergegeven:

```
\stelvoettekstenin[.1.][.2.][.3.]
```

85

- .1. tekst marge rand
- .2. tekst sectie datum *markering* paginanummer
- .3. tekst sectie datum *markering* paginanummer

Het commando `\stelvoettekstenin` heeft drie argumenten, waarvan de eerste optioneel is. Wanneer we alleen de laatste twee argumenten opgeven, gaat `CONTEX`T ervanuit dat het eerste argument `[tekst]` is. Optionele argumenten staan schuin gedrukt. Standaardinstellingen zijn onderstreept en variabele instellingen schuin gedrukt. In dit voorbeeld betekent *tekst* dat er een willekeurige tekst mag worden opgegeven en *markering* dat een (eerder gedefinieerde)

markering wordt geaccepteerd (bijvoorbeeld hoofdstuk of paragraaf). De instelling datum resulteert in de huidige datum.

Bij sommige commando's zien we een ►◀ in de rechterbovenhoek. Hiermee geven we aan dat het commando een speciale betekenis heeft binnen interactieve, dat wil zeggen voor het beeldscherm geschikte, teksten. Bij commando's die alleen betekenis hebben in interactieve teksten zijn zwarte pijltjes geplaatst, wanneer het slechts aanvullende functionaliteit betreft gebruiken we gekleurde (grijze) pijltjes.

1.4 Programma's

Hoewel in \TeX vrij veel mogelijk is, worden bepaalde manipulaties met tekst uitgevoerd met behulp van programma's. Het PERL script \TeX UTIL ondersteunt het aanmaken van indexen, gesorteerde lijsten, inhoudsopgaven, overzichten van figuren, tabellen en dergelijke en het integreren van externe figuren.

Het verwerken van files wordt ondersteund door het PERL script \TeX EXEC. Hiermee kan men formats genereren en files verwerken. Daarbij kunnen een aantal opties worden opgegeven, bijvoorbeeld of men PDF uitvoer wil hebben, een boekje wil maken, of extra snel wil verwerken.

Hoewel het gebruik van $\text{CON}\TeX\text{T}$ niet is gebonden aan deze programma's, hebben zij wel mede de functionaliteit bepaald. Een deel daarvan komt aan de orde in deze handleiding, de rest laten we aan de fantasie van de lezer over.

1.5 Files

\TeX gaat standaard uit van zogeheten ASCII-files. ASCII is een internationaal gestandaardiseerd computeralfabet. Een ASCII-file, meestal met de extensie `tex`, wordt door \TeX omgezet in een file met grafische commando's. Deze file, die de extensie `dvi` heeft, wordt vervolgens omgezet in een file met voor een fotozetter of printer begrijpelijke stuurcommando's, bijvoorbeeld PostScript (met de extensie `ps`).

Een DVI-file kan met behulp van daartoe geschikte programma's op het scherm worden weergegeven. Wil men teksten ook elektronisch verspreiden, dan kan een PostScript-file worden omgezet in een PDF-file (extensie: `pdf`). Dergelijke files zijn niet alleen van hoogwaardige grafische kwaliteit, ze zijn ook interactief. Als we bijvoorbeeld met de muis klikken op een ingang in de index, gaan we automatisch naar de juiste plaats in de tekst. Overigens kan $\text{CON}\TeX\text{T}$ met behulp van $\text{PDF}\TeX$ ook direct PDF code genereren.

In tegenstelling tot andere macro-pakketten maakt $\text{CON}\TeX\text{T}$ slechts enkele hulpfiles aan. Deze zijn te herkennen aan hun extensie. De zogenaamde `tui` file bevat informatie die voor een tweede verwerkingslag van belang is, en de `tub` file bevat (verplaatsbare) blokken. Het script \TeX UTIL zet een `tui` file om in een geordende `tuo` file.

1.6 Teksten

1.6.1 Karakters

Een \TeX -tekst is opgebouwd uit ASCII-karakters. Omdat in deze set geen karakters met accenten voorkomen, moeten we een andere manier gebruiken om deze te zetten. De meest gebruikelijke manier is het te accentueren karakter te laten voorafgaan door bijvoorbeeld `\`, `'` of `^`, met andere woorden `\"e` wordt `ë`.

Het is technisch mogelijk om andere invoer-coderingen te gebruiken. De beslissing om dit al dan niet te doen moet niet ondoordacht worden genomen, vooral omdat het de uitwisseling van bestanden in de weg kan staan.

Sommige karakters spelen binnen \TeX een bijzondere rol. Deze karakters kunnen worden gezet door er een `\` voor te zetten. Zo krijgt men een procent-teken met `\%`. Alleen een `%` geeft ongewenste resultaten omdat \TeX alles na dit teken als commentaar ziet. Evenzo kunnen we een `$` zetten met `\$`. Bij een `$` zonder `\` gaat \TeX over in de wiskundige mode.

1.6.2 Paragrafen

Binnen \TeX is de eenheid van informatie meestal een paragraaf (alinea). Een paragraaf wordt afgesloten met het commando `\par` of met een lege regel. Het laatste verdient de voorkeur. Door een ASCII-tekst ruim op te zetten, wordt namelijk voorkomen dat \TeX overgangen naar een nieuwe paragraaf 'niet ziet'. Een krappe layout (weinig lege regels) leidt snel tot ongewenste resultaten.

1.6.3 Boxen

\TeX bouwt een pagina op in boxen (rechthoeken). Daarbij wordt onderscheid gemaakt in horizontale en verticale boxen. Elke letter is een box, een woord is een uit boxen opgebouwde box, een regel is `...` enz. Veel van de tijdens het omzetten van een \TeX -file in een `dvi`-file getoonde (fout)meldingen hebben betrekking op te volle of te lege boxen. Horizontale en verticale boxen kunnen worden gezet met de \TeX -commando's: `\hbox` en `\vbox`.

1.6.4 Lettertypen

\TeX is een van de weinige zetsystemen die goed uit de voeten kan met wiskundig zetwerk. Hiervoor zijn echter wel 'volledige' karaktersets nodig, dat wil zeggen sets waarin naast letters, cijfers en leestekens ook wiskundige symbolen beschikbaar zijn. Aan deze eis voldoen onder meer de Computer Modern Roman en Lucida Bright letterfamilies. Beide bieden zowel schreef- als schreefloze letters, aangevuld met een mono gespatieerde typeletter. Natuurlijk kan men ook andere lettertypen gebruiken.

1.6.5 Maten

Letters hebben bepaalde afmetingen. Hetzelfde geldt voor spatiëring. Soms moeten maten worden opgegeven. Binnen \TeX kan men maten opgeven in een van de in tabel 1.1 getoonde eenheden. De afstand tussen twee regels van deze tekst is 14.83998pt.

\TeX kent naast de bovenstaande maten ook nog `em` en `ex`. Beide zijn afhankelijk van het lettertype: een `ex` is gelijk aan de hoogte van een `x`, en een `em` de breedte van een `—` (---). In de Computer Modern Roman fonts hebben cijfers een breedte van `1/2em`.

maat	betekenis	equivalent
pt	point	72.27pt = 1in
pc	pica	1pc = 12pt
in	inch	1in = 2.54cm
bp	big point	72bp = 1in
cm	centimeter	2.54cm = 1in
mm	millimeter	10mm = 1cm
dd	didot point	1157dd = 1238pt
cc	cicero	1cc = 12dd
sp	scaled point	65536sp = 1pt

Tabel 1.1 Maten in $\text{T}_{\text{E}}\text{X}$.

In de meeste voorbeelden zullen we maten specificeren in punten, centimeters of millimeters. Ene punt is ongeveer .35mm. De meeste maten hebben een Amerikaanse oorsprong. De European Didot punt is equivalent aan $1/2660\text{m} = 3.759398496\text{mm}$.

1.7 Foutmeldingen

Een van de nadelen van $\text{T}_{\text{E}}\text{X}$ is dat de foutmeldingen niet altijd voor een ieder begrijpelijk zijn. Meestal is een fout een verkeerd geschreven commando of een vergeten haakje. Het symmetrisch gebruik van haakjes en accoladen kan in $\text{T}_{\text{E}}\text{XEDIT}$ worden gecontroleerd.

Bij elke fout wordt een regelnummer gemeld. Als op het moment dat $\text{T}_{\text{E}}\text{X}$ met een ? komt een e wordt ingetypt, dan wordt in $\text{T}_{\text{E}}\text{XEDIT}$ automatisch naar de laatste fout gesprongen. Soms is het echter ‘onmogelijk’ $\text{T}_{\text{E}}\text{X}$ te verlaten, omdat een filenaam wordt verwacht. Het intypen van `ctrl-z` of `ctrl-c` doet vaak wonderen.

1.8 Versienummers

$\text{T}_{\text{E}}\text{X}$ was in 1982 in principe af. Dat wil zeggen dat aan de functionaliteit niets meer zou worden toegevoegd. Een uitzondering is echter gemaakt voor het ondersteunen van meerdere talen naast elkaar, het gebruik van 8-bits invoer en samengestelde karakters. Bovendien zijn er in de loop der jaren enkele ‘bugs’ gevonden en verwijderd. Op dit moment gebruiken we versie 3.1415926 (bij iedere structurele verbetering wordt een decimaal van π toegevoegd).

We gaven reeds aan dat $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ overweg kan met $\varepsilon\text{-T}_{\text{E}}\text{X}$ en $\text{PDF}_{\text{T}}\text{E}_{\text{X}}$. Beiden zijn in ontwikkeling. Gebruik dus altijd de laatste versies. Bij deze handleiding is gebruik gemaakt van $\text{PDF-}\varepsilon\text{-T}_{\text{E}}\text{X}$, met $\varepsilon\text{-T}_{\text{E}}\text{X}$ versie 2.1 en $\text{PDF}_{\text{T}}\text{E}_{\text{X}}$ versie 14h.

In tegenstelling tot $\text{T}_{\text{E}}\text{X}$ is $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ nog niet af. Hoewel de functionaliteit grotendeels is uitgekristalliseerd, worden de macro’s continu verbeterd, zowel in termen van kwaliteit als verwerkingssnelheid. Uitbreidingen vinden vooral plaats op basis van de wensen van gebruikers.

Sommige mogelijkheden zal men missen, van andere zal men zich wellicht in eerste instantie afvragen waarvoor ze zijn bedoeld. Hoe dan ook, deze handleiding is gezet met versie 2001.11.5 (de eerste versie van deze handleiding kwam beschikbaar in 1992).

Aanpassingen vinden plaats binnen de beschikbare interface. Zo heeft het ondersteunen van interactieve teksten geleid tot grondige aanpassingen in de onderliggende macro's, terwijl er voor de gebruiker weinig is veranderd. `CONTEXT` wordt niet wezenlijk anders, maar hopelijk wel steeds beter en flexibeler.

1.9 Top 10

Deze handleiding is dik. Omdat een beginnend gebruiker van `CONTEXT` wellicht wordt afgeschrikt door de hoeveelheid commando's volgen hier de tien meest gebruikte groepen commando's:

1. `\starttekst`, `\stoptekst`
2. `\hoofdstuk`, `\paragraaf`, `\titel`, `\onderwerp`, `\stelkopin`, `\volledigeinhoud`
3. `\em`, `\bf`, `\kap`
4. `\startopsomming`, `\stopopsomming`, `\som`, `\kop`
5. `\afkorting`, `\voluit`, `\volledigelijstmetafkortingen`
6. `\plaatsfiguur`, `\gebruikexternfiguur`, `\externfiguur`
7. `\plaatstabel`, `\starttabel`, `\stoptabel`
8. `\doordefinieren`, `\doornummeren`
9. `\index`, `\volledigeindex`
10. `\stellayoutin`, `\stelvoettekstenin`, `\stelhoofdtekstenin`

1.10 Waarschuwing

Gebruikers kunnen hun eigen commando's definiëren. Omdat het gevaar bestaat dat eigen commando's conflicteren met die van plain `TEX` en `CONTEXT`, raden we een ieder dan ook aan hoofdletters te gebruiken, bijvoorbeeld:

```
\def\DitIsEenHeelHandigCommando{, nietwaar?}
```


2

Documenten

Documenten

2.1 Inleiding

Waarom gaat men $\text{T}_{\text{E}}\text{X}$ gebruiken? De een komt bij $\text{T}_{\text{E}}\text{X}$ terecht omdat hij of zij wiskundig zetwerk moet leveren, een ander valt voor de charme van scheiding tussen inhoud en opmaak. Weer een ander ziet voordelen in de programmeerbaarheid en natuurlijk kan ook in kwaliteit een argument voor gebruik van $\text{T}_{\text{E}}\text{X}$ worden gevonden.

Men loopt bij $\text{T}_{\text{E}}\text{X}$ niet snel aan tegen capaciteitproblemen. Het zetten van omvangrijke documenten van vele honderden pagina's is dan ook bij uitstek een klusje voor $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$. Bij het coderen van een document dient men als het even mogelijk is verder te kijken dan het huidige document. Wellicht zal het document dat nu voor drukken op papier wordt gezet, over enige tijd elektronisch beschikbaar moeten komen. Of misschien dat naast de volledige versie van 700 pagina's er een korte versie met pakweg 200 pagina's moet komen. Een zo groot mogelijke scheiding van vorm en inhoud enerzijds en een acceptabele redundantie in structuur anderzijds staan vaak al garant voor dergelijk hergebruik.

Een systeem als $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ is bedoeld om het leven gemakkelijker te maken. Bij het coderen van een document kan al gauw de indruk ontstaan dat “dit of dat toch eenvoudiger zou moeten kunnen”. Vaak is dat ook zo en het antwoord is in veel gevallen te vinden in deze handleiding, zij het soms tussen de regels. Ook kost het wat tijd om te leren denken in structuur en inhoud, zeker als men gewend is aan het werken met muisgestuurde opmaaksystemen. In dit hoofdstuk staan we daarom nader stil bij de structuur waarbinnen een document of collectie documenten kan worden opgezet.

2.2 Starten en stoppen

Als een tekst een zelfstandig document is, dan gebruiken we de volgende commando's om het begin en eind van zo'n tekst te markeren:

```
\starttekst  
\stoptekst
```

Het eerste commando voert een aantal noodzakelijke initialisaties uit en het laatste commando vertelt $\text{T}_{\text{E}}\text{X}$ dat er geen commando's meer volgen en dat de tekst wordt afgesloten. Als dit commando wordt weggelaten toont $\text{T}_{\text{E}}\text{X}$, na het verwerken van de file, een * (sterretje). $\text{T}_{\text{E}}\text{X}$ verwacht dan een commando, bijvoorbeeld `\end`.

Hoewel dit niet verplicht is, raden we aan instellingen op te nemen voor het `\starttekst`-commando. Op deze manier kan een slimme tekstverwerker vaststellen welke instellingen moeten worden meegenomen als slechts een deel van het document wordt verwerkt. Tekst en instellingen na het `\stoptekst`-commando worden niet gebruikt.

Hieronder is een voorbeeld gegeven van een zeer eenvoudige opmaak.

```
\starttekst
```

```
\onderwerp{Klinkers en medeklinkers}
```

Als we het Nederlands op Arabische wijze zouden spellen, konden we alleen uit de context afleiden of *bs* stond voor *bas*, *baas*, *bes*, *bis*, *ibis*, *bos*, *boos*, *bus*, *buis*, *abuis* of *base* en of *pn* *pan*, *pen*, *peen*, *pin*, *puin*, *poen*, *apen*, *iepen* of *open* betekende. Paradoxaal genoeg zouden de meeste spellingsruzies dan onmogelijk zijn, want die hangen nu eenmaal vooral samen met het willen schrijven van klinkers en tweeklanken.

```
\blanko
```

```
{\bf G.C. Molewijk, Spellingsverandering van zin naar onzin (1992).}
```

```
\stoptekst
```

De commando's `\starttekst...``\stoptekst` mogen worden genest. Binnen een tekst mag een file worden geladen die zelf ook een `\starttekst` en `\stoptekst` bevat.

2.3 Structuur

In deze paragraaf gaan we nader in op de mogelijkheid het beheer van teksten te structureren. Lezers die slechts een beperkt aantal teksten per jaar produceren kunnen de rest van deze paragraaf rustig overslaan. Voor eenvoudige, losse teksten kan men namelijk volstaan met de opzet:

```
\omgeving ditjes
```

```
\omgeving datjes
```

```
\starttekst
```

```
... ditjes en datjes ...
```

```
\stoptekst
```

Als echter een tekst deel uitmaakt van een groter geheel, kan men overwegen een project-georiënteerde structuur op te zetten. Hierbij gelden de volgende uitgangspunten:

- Samenhangende teksten moeten in hun samenhang kunnen worden onderhouden. Een dergelijk geheel noemen we een *project*.
- Op overstijgend niveau moeten layout-kenmerken kunnen worden vastgelegd. Hiervoor wordt de term *omgeving* gereserveerd.
- Samenhangende teksten in een project noemen we *produkten*.
- Een produkt moet kunnen worden opgedeeld in, eventueel met andere produkten te delen, *onderdelen*. Bovendien moeten deze onderdelen zelfstandig kunnen worden verwerkt en eventueel onderscheiden in een tekstverwerker.

Een *project*, *omgeving*, *produkt* en *onderdeel* worden gestart en gestopt met een van de volgende commando's:

```
\startproject ... .. \stopproject
...   file
```

```
\startprodukt ... .. \stopprodukt
... file
```

```
\startomgeving ... .. \stopomgeving
... file
```

```
\startonderdeel ... .. \stoponderdeel
... file
```

Vóór een `\start`-`\stop`-paar kunnen eventueel commando's worden opgenomen. Als een file niet op het actuele gebied staat, wordt op lagere niveaus gezocht. Dit maakt het mogelijk een of meer omgevingen te gebruiken voor meerdere, op directories geordende, files.

commando	project	omgeving	produkt	onderdeel
<code>\project naam</code>			*	*
<code>\omgeving naam *</code>	*	*	*	
<code>\produkt naam</code>	*		*	
<code>\onderdeel naam</code>		*	*	

Tabel 2.1 Structuur-commando's.

Om producten en onderdelen min of meer als zelfstandig geheel te kunnen behandelen, worden de bovenstaande commando's gebruikt. In tabel 2.1 zijn de met * gemarkeerde commando's verplicht en de met [*] aangegeven commando's facultatief. Vóór de `\stop`-commando's wordt de inhoudelijke tekst geplaatst.

```
\startproject alles
\omgeving layout
\produkt docent
\produkt cursist
\produkt stramien
\stopproject
```

Hiernaast is een voorbeeld opgenomen van een project-file.

```
\startprodukt docent
\project alles
\onderdeel docent1
\onderdeel docent2
\stopprodukt
```

Het produkt `docent.tex` (een docenthandleiding) kan bijvoorbeeld zijn gedefinieerd als hiernaast.

```

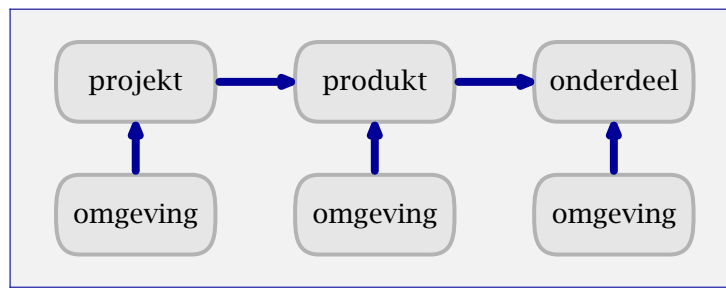
\startonderdeel docent2
\project alles
\produkt docent
... tekst ...
\stoponderdeel

```

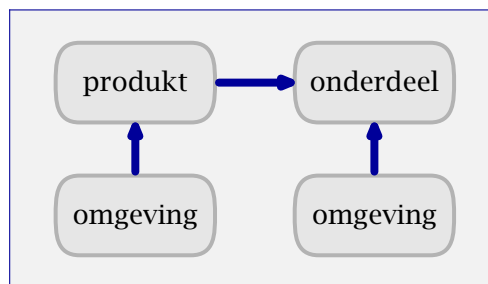
Links zien we het bijbehorende onderdeel.

Het is mogelijk alle vier structurerende elementen afzonderlijk te verwerken (zetten). Verwerking ligt echter alleen voor de hand bij *produkten* en *onderdelen*, omdat daarin de te zetten teksten zijn opgenomen. Een projectfile bevat in principe alleen een lijst met produkten en omgevingen. Bij het verwerken van een projectfile vormen alle produkten een document. Dit zal zelden de bedoeling zijn. Deze handleiding is opgezet als project: ieder deel is een produkt en ieder hoofdstuk van een deel is een onderdeel. Er is een omgeving, die in de projectfile wordt aangeroepen.

Schematisch kan de samenhang tussen files als volgt worden weergegeven:



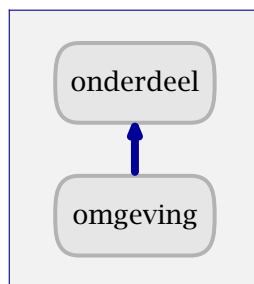
Figuur 2.1 Een uitgebreide projectopzet.



Figuur 2.2 Een opzet met alleen produkten.

De bovenstaande structuur biedt alleen voordelen bij veel samenhangende teksten, bijvoorbeeld opleidingsmateriaal. In veel gevallen kan men prima uit de voeten met `\starttekst` en `\stoptekst`, eventueel in combinatie met `\input` of `\omgeving` voor in andere files ondergebrachte tekst en/of commando's.

Het verdient voorkeur alle gemeenschappelijke instellingen in één omgeving onder te brengen. Als een onderdeel of produkt echter een afwijkende vormgeving of afwijkende commando's kent, dan kunnen *deelomgevingen* worden gedefinieerd:



Figuur 2.3 Een opzet met alleen een onderdeel.

```
\startdeelomgeving[namen]
... instellingen ...
\stopdeelomgeving
```

Een deelomgeving kan worden opgenomen in een omgeving-file of in een aparte file. In het laatste geval kan de deelomgeving worden geladen met het commando:

```
\deelomgeving naam
```

Hieronder is een voorbeeld gegeven van een opzet.

```
\startproject demos
\omgeving omgeving
\produkt voorbeeld
\stopproject
```

file: `demos.tex`

In deze file wordt vastgelegd uit welke produkten het project bestaat. Daarnaast worden eventuele omgevingen gedefinieerd.

```
\startomgeving omgeving
\stelwitruintein[groot]
\stelvoettekstenin[deel][hoofdstuk]
\stopomgeving
```

file: `omgeving.tex`

In de omgeving vinden de instellingen plaats die betrekking hebben op alle produkten. Eventueel kunnen meerdere omgevingen worden gemaakt of binnen deze omgeving deelomgevingen per produkt worden gedefinieerd.

```
\startprodukt voorbeeld
\project demos
\startinleidingen
  \volledigeinhoud
\stopinleidingen
\starthoofsteksten
  \onderdeel eerste
  \onderdeel tweede
\stophoofsteksten
\startuitleidingen
  \volledigeindex
\stopuitleidingen
\stopprodukt
```

file: `voorbeeld.tex`

In deze produkt-file wordt de structuur van het produkt vastgelegd. Omdat inhoudsopgaven en indexen met slechts enkele commando's kunnen worden opgeroepen, maken we hiervoor geen aparte files aan.

```

\startonderdeel eerste
\deel{Een}
\volledigeinhoud
\hoofdstuk{Eerste}
..... tekst .....
\hoofdstuk{Tweede}
..... tekst .....
\volledigeindex
\stoponderdeel

```

file: eerste.tex

In de onderdelen van een produkt plaatsen we de tekst, figuren enz. Hier kunnen we ook eventuele inhoudsopgaven en indexen per produkt definiëren. Er is niets op tegen onderdelen verder onder te verdelen.

```

\startonderdeel tweede
\deel{Twee}
\volledigeinhoud
\hoofdstuk{Alfa}
..... tekst .....
\hoofdstuk{Beta}
..... tekst .....
\volledigeindex
\stoponderdeel

```

file: tweede.tex

Het produkt bestaat in dit geval dus uit meerdere onderdelen. We hadden een deel ook als produkt kunnen definiëren en de hoofdstukken als onderdelen. Het onderscheid tussen produkten en onderdelen kan per toepassing verschillen.

De files `eerste.tex`, `tweede.tex` en `voorbeeld.tex` kunnen afzonderlijk worden verwerkt. Zolang er nog maar één produkt in dit project zit, kan ook `project.tex` worden verwerkt. De omgeving kan apart worden verwerkt (getest) maar levert geen output.

2.4 Directories

Veel \TeX implementaties doorzoeken een hele directory structuur als ze de opgegeven file niet aantreffen op het werkgebied. Dit kost niet alleen tijd, maar kan ook verwarrend zijn, zeker als een verkeerde file wordt geladen.

Om die reden hanteert $\text{CON}\TeX$ een eigen zoekmethode. In grote lijnen komt het er op neer dat altijd eerst op het werkgebied wordt gezocht, dat eventueel tot drie gebieden onder het werkgebied wordt gekeken, dat zonodig het figurengebied wordt bezocht, en dat in een enkel geval op alle gebieden die de betreffende implementatie kan vinden wordt gezocht.

Dit betekent dat men bijvoorbeeld omgevingen een of meer gebieden lager in de hiërarchie kan zetten, wat soms handig is als men files heeft geordend op afzonderlijke gebieden, bijvoorbeeld:

```

/texfiles/cursus/layout.tex
/texfiles/cursus/docent/handleiding.tex
/texfiles/cursus/cursist/cursusmap.tex
/texfiles/cursus/vanalles/sheets.tex

```

In de drie laatste files kan men met `\omgeving layout.tex` de betreffende layout file laden, zonder bang te hoeven zijn dat \TeX een elders in de structuur aanwezige file met dezelfde naam laadt.

Het is dus mogelijk files in verschillende directories te plaatsen. als een project goed is opgezet, kan men de losse componenten verwerken. De belangrijkste randvoorwaarde is dat de projectfile en de eventuele omgevingen moeten kunnen worden gevonden.

2.5 Versies

Het is mogelijk een voorlopige versie te genereren van een tekst. In dat geval wordt informatie in de tekst opgenomen met betrekking tot referenties en zetfouten. Het te gebruiken commando luidt:

```
\versie[...]
...   definitief concept voorlopig
```

Standaard wordt een definitieve versie uitgedraaid. Het uitdraaien van een voorlopige versie heeft alleen zin als er controles moeten worden uitgevoerd ten aanzien van het zetwerk en de verwijzingen. In geval van een concept, wordt onderaan iedere bladzijde vermeld dat het een concept betreft.

De meldingen in de marge worden aangemaakt tijdens het zetten van een bladzijde. Omdat \TeX altijd wat verder vooruit kijkt, kunnen meldingen te vroeg komen.

2.6 Modes

De uitvoer van \TeX kan DVI of PDF zijn. Een document kan voor papier of scherm worden vormgegeven, de laatste met aanvullende functionaliteit. Van eenzelfde document kunnen meerdere versies worden gegenereerd, verschillend in omvang en vormgeving. Kortom: uit een bron kunnen meerdere produkten worden afgeleid.

Het verwerken van een file komt in de praktijk neer op het opstarten van \TeX waarbij de filenaam wordt meegegeven van de te verwerken file. Stel dat standaard DVI uitvoer wordt gegenereerd. Omschakelen naar PDF kan door in de file zelf of in een configuratiefile aan te geven dat een andere uitvoer gewenst is, maar echt comfortabel is dat niet:

```
\steluitvoerin[pdftex]
```

voor directe PDF uitvoer, of in geval van de indirecte weg:

```
\steluitvoerin[dvips,acrobat]
```

Gelukkig is er \TeX EXEC. Dit PERL script voorziet CONTEXT van een zogenaamde command-line-interface. Willen we PDF in plaats van DVI, dan roepen we gewoon \TeX EXEC aan met:

```
texexec --pdf filenaam
```

Er zijn nog meer opties, zoals het maken van A5-boekjes; meer hierover is te vinden in de handleiding bij \TeX EXEC. Bij een optie staan we hier wat langer stil: modes.

```
texexec --pdf --mode=scherf filenaam
```

De idee hierachter is dat binnen een stijldefinitie, op elk moment kan worden opgevraagd in welke mode het document wordt verwerkt. Men kan bijvoorbeeld in een layoutdefinitie de volgende regels opnemen:

```
\startmode[scherm]
  \stelinteractiein[status=start]
  \stelkleurenin[status=start]
\stopmode
```

en wellicht:

```
\startnotmode[scherm]
  \stelkleurenin[status=start,conversie=altijd]
\stopnotmode
```

Men kan ook een serie modes opgeven. Naast deze twee mode commando's zijn er nog wat meer primaire macros beschikbaar. Uitgaande van de mode `scherm` kunnen we bijvoorbeeld zeggen:

```
\doifmodeelse {scherm} {doe dit} {en niet dat}
\doifmode      {scherm} {doe iets}
\doifnotmode   {scherm} {doe wat anders}
```

Een mode kan worden ge(de)activeerd met:

```
\enablemode[scherm]
\disablemode[scherm]
```

Ook hier kan men meerdere modes meegeven, bijvoorbeeld:

```
\enablemode[papier,A4]
```

De kracht van `TEXEXEC` is echter dat men het activeren niet in de file zelf doet, maar daarbuiten. Net als bij het kiezen van het uitvoerformaat, geldt hier: hoe neutraler de broncode, des te beter!

Om het ontwikkelen van algemene stijlen te vergemakkelijken, biedt `CONTEXT` zelf ook modes aan. Op dit moment zijn dat:

```
*lijst      de lijst waarom met zojuist vroeg is geplaatst
*register   het register waarom met zojuist vroeg is geplaatst
*interactie interactie (hyperlinks enz) zijn beschikbaar
*sectieblok het genoemde sectieblok in actief
```

Systeem modes worden voorafgenaam door een `*`, zodat ze niet conflicteren met gebruikers modes. Een voorbeeld van een sectieblok mode is `*inleiding`. Een voorbeeld van het gebruik van een systeem mode is:

```
\startmode[*interactie]
  \setuppapersize[S6][S6]
\stopmode
```

3

Vlakverdeling

Vlakverdeling

3.1 Inleiding

Bij het bewerken van een tekst houdt $\text{T}_{\text{E}}\text{X}$ rekening met (onder andere) de actuele `\hsize` (breedte) en `\vsize` (hoogte). Zodra de ingestelde `\vsize` wordt overschreden, roept $\text{T}_{\text{E}}\text{X}$ de zogenaamde output-routine aan. Deze handelt vervolgens het gezette deel, meestal een bladzijde, af. Dit afhandelen bestaat uit onder meer het plaatsen van hoofd- en voetregels, het zetten van het paginanummer, het aanbrengen van achtergronden en navigatiemiddelen en het plaatsen van voetnoten, verplaatste tabellen en figuren. Er zijn dan ook meer maten in het geding dan alleen de hoogte en de breedte van de te zetten tekst.

3.2 Papierformaat

Met het commando `\stelpapierformaat` worden de afmetingen van het papier ingesteld. We maken daarbij onderscheid tussen het formaat waarop we zetten en het formaat waarop we afdrukken (printen).

```
\stelpapierformaat[...1,...][...2,...]
.1.   A3 A4 A5 A6 letter ... CD naam liggend gespiegeld geroteerd 90 180 270
.2.   A3 A4 A5 A6 letter ... naam liggend gespiegeld geroteerd negatief 90 180 270
```

De afmetingen van de DIN-formaten zijn in tabel 3.1 weergegeven.

formaat	afmetingen in mm	formaat	afmetingen in mm
A0	841 × 1189	A5	148 × 210
A1	594 × 841	A6	105 × 148
A2	420 × 594	A7	74 × 105
A3	297 × 420	A8	52 × 74
A4	210 × 297	A9	37 × 52

Tabel 3.1 Standaard papierformaten.

Naast deze afmetingen zijn ook B0–B9 en C0–C9 beschikbaar, en verder: letter, legal, folio en executive, envelop 9–14, monarch, check, DL en CD.

Men kan een nieuw formaat definiëren met:

```
\definieerpapierformaat[...] [.,.,.=.,.]
```

```
...      naam
breedte  maat
hoogte   maat
offset   maat
schaal   getal
```

Zo is bijvoorbeeld CD gedefinieerd als:

```
\definieerpapierformaat[CD] [breedte=12cm, hoogte=12cm]
```

We kunnen nu bijvoorbeeld zeggen:

```
\stelpapierformaatin[CD] [A4]
```

Wat zoveel betekent als: gebruik voor het zetwerk CD en voor het printen A4. We komen later op die tweede instelling terug.

Naast de op papier gerichte formaten zijn er enkele meer op het beeldscherm afgestelde varianten: S3–S6. Deze definiëren schermen met breedtes oplopend van 300 tot 600 pt en een hoogte van 3/4 maal de breedte. Deze S-formaten sluiten dus aan bij de aspect ratio van het beeldscherm.

Bij het instellen van een papierformaat worden de standaardinstellingen automatisch meege-schaald. Zo heeft men in ieder geval een vertrekpunt.

3.3 Bladindeling

We maken om praktische redenen onderscheid tussen wat we zullen noemen de zetspiegel en de marges. De zetspiegel is dat deel van de bladzijde waar de lopende tekst wordt geplaatst. In dit gedeelte worden standaard alle tekstelementen geplaatst (zie figuur 3.1).

Het hoofd bevindt zich *tussen* het kopwit en de zetspiegel. In het hoofd en de voet worden lopende titels en paginanummers geplaatst. Links en/of rechts van de tekst kunnen we structureel of incidenteel bepaalde informatie kwijt. Bijvoorbeeld nummers van hoofdstukken en paragrafen of trefwoorden. De marges maken deel uit van het rug- en snijwit. De breedte van de marges heeft *geen* invloed op de plaats van de zetspiegel, de hoogte van het hoofd en de voet bepalen *wel* mede de hoogte van het tekstdeel.

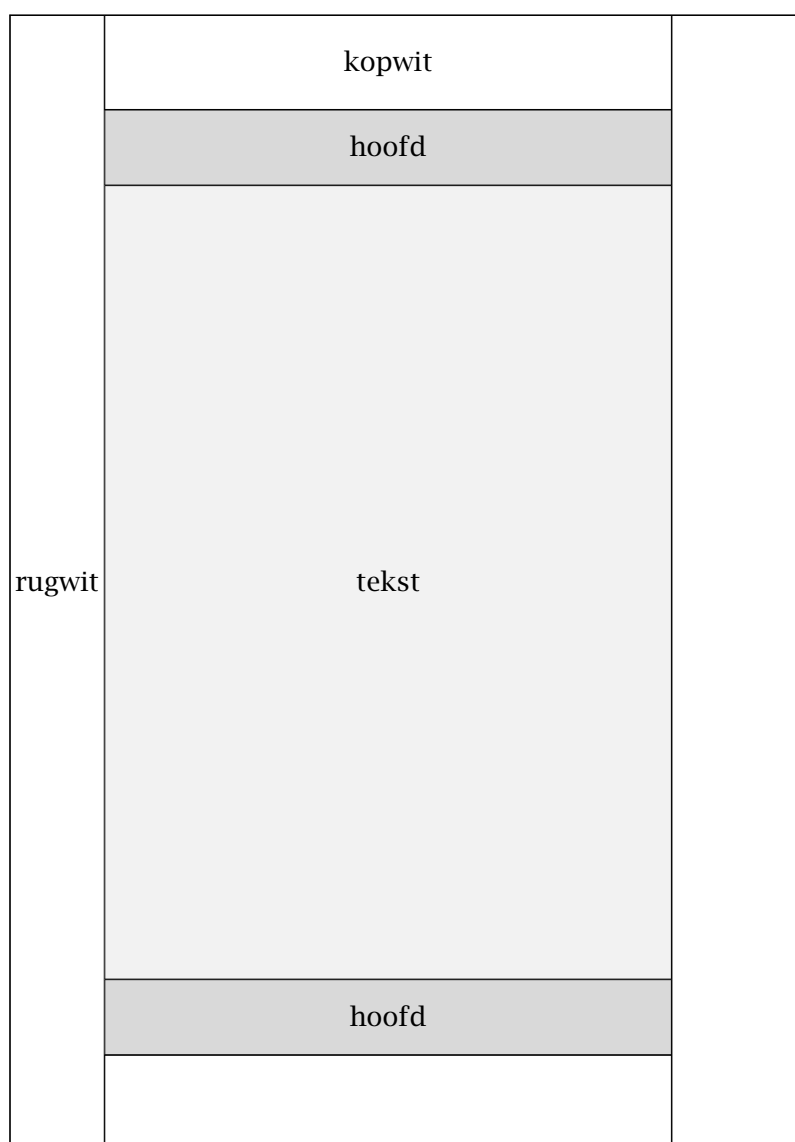
links

rechts

De zetspiegel en de marges kunnen worden ingesteld met het commando `\stellooyoutin`. Onder de zetspiegel verstaan we (hier) het gedeelte van de bladzijde dat overblijft als we de witte randen weglaten. In figuur 3.1 beslaat de zetspiegel het hoofd, het middendeel en de voet (het grijze gedeelte).

Het instellen van de marge heeft bij een papieren document geen gevolgen voor de zetspiegel. Deze instelling speelt alleen een rol bij het plaatsen (en afbreken) van tekst in de linker- of rechtermarge.

In papieren documenten kunnen we meestal volstaan met hoofd- en voetregels. In elektronische documenten daarentegen hebben we ook ruimte nodig voor sturende elementen (meer



| marge |

| marge |

Figuur 3.1 De A4 zetspiegel en marges (hoogte = hoofd + tekst + voet).

```
\stelloayoutin[...]=...]
```

breedte	<i>maat</i> passend midden
hoogte	<i>maat</i> passend midden
rugwit	<i>maat</i>
kopwit	<i>maat</i>
marge	<i>maat</i>
linkermarge	<i>maat</i>
rechtermarge	<i>maat</i>
hoofd	<i>maat</i>
voet	<i>maat</i>
boven	<i>maat</i>
onder	<i>maat</i>
linkerrand	<i>maat</i>
rechterrand	<i>maat</i>
hoofdafstand	<i>maat</i>
voetafstand	<i>maat</i>
bovenafstand	<i>maat</i>
onderafstand	<i>maat</i>
linkermargeafstand	<i>maat</i>
rechtermargeafstand	<i>maat</i>
linkerrandafstand	<i>maat</i>
rechterrandaafstand	<i>maat</i>
rugoffset	<i>maat</i>
kopoffset	<i>maat</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
markering	aan <u>uit</u> kleur
plaats	links midden rechts onder boven <u>enkelzijdig</u> dubbelzijdig
schaal	<i>maat</i>
nx	<i>getal</i>
ny	<i>getal</i>
dx	<i>maat</i>
dy	<i>maat</i>
regels	<i>getal</i>
grid	ja <u>nee</u>
bodemwit	<i>getal</i>
snijwit	<i>getal</i>

daarover in hoofdstuk ??). Omdat bij elektronische documenten het gebruik van achtergronden wat meer voor de hand ligt —door middel van een achtergrond kunnen we de verschillende delen van het scherm benadrukken— zijn ook de afstanden tussen bijvoorbeeld tekst en hoofd en tekst en voet instelbaar.

Het is mogelijk de zetspiegel zichtbaar te maken in de tekst. Hiervoor kunnen de volgende commando's worden gebruikt:

```
\toonkader[...]
```

```
... tekst marge rand
```

De instellingen kunnen zichtbaar worden gemaakt met:


```
\tooninstellingen
```

Een combinatie van beiden roepen we op met:

```
\toonlayout
```

De breedte van de tekst is meestal beschikbaar in `\hsize` en de hoogte in `\vsize`. Willen we echter op safe spelen dan kan men beter gebruik maken van de `\dimen`-registers `\tekstbreedte` en `\teksthoogte`, `\zetbreedte` en `\zethoogte`. Waar de zetbreedte constant is, geeft de tekstbreedte de breedte van de kolom tekst weer. Bij het zetten in twee kolommen is de tekstbreedte bijvoorbeeld iets minder dan de helft van de zetbreedte. De teksthoogte is dat wat overblijft als we de zethoogte verminderen met de hoogte van het hoofd en de voet.

variabele	betekenis
<code>\zetbreedte</code>	breedte van de tekst
<code>\zethoogte</code>	hoogte van de tekst
<code>\tekstbreedte</code>	breedte van een kolom tekst
<code>\teksthoogte</code>	hoogte van de tekst – hoofd – voet

Tabel 3.2 Enkele `\dimen`-variabelen.

Ook de andere afstanden en maten zijn beschikbaar, zoals `\linkermargebreedte` en `\voethoogte`, maar pas op: deze waarden kan men alleen gebruiken, niet instellen.

In principe wordt een tekst volledig automatisch opgemaakt. Het kan echter voorkomen dat het al dan niet verplaatsen van een regel aanzienlijk fraaier zetwerk oplevert. In dat geval kan men ter plaatse de hoogte van de zetspiegel wat aanpassen met:

```
\paslayoutaan[...][...=...]  
...      getal  
hoogte   maat max  
regels   getal
```

Dergelijke commando's kunnen echter ook negatieve gevolgen hebben, bijvoorbeeld wanneer we de tekst hebben gewijzigd en de aanpassing eigenlijk niet meer nodig is. Het is daarom verstandiger de aanpassingen (zichtbaar) bovenaan de tekst te definiëren. Een voorbeeld van zo'n aanpassing is:

```
\paslayoutaan[21,38][hoogte=+.5cm]
```

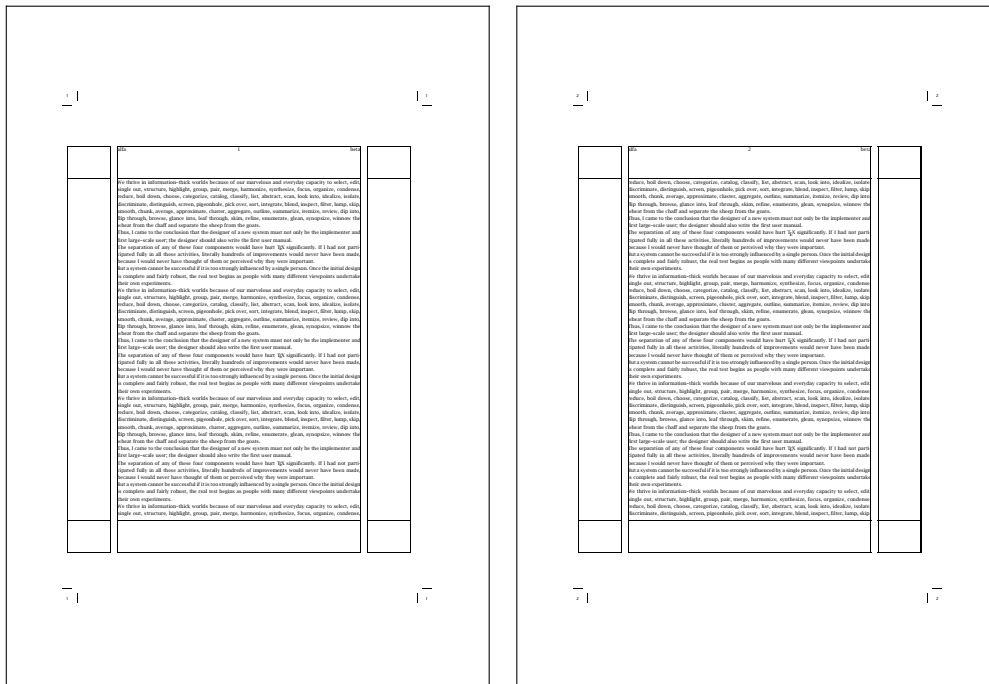
In dit geval wordt op de pagina's 21 en 38 de hoogte van de tekst tijdelijk 0.5 cm verhoogd, waarbij de voetregel netjes op de goede hoogte blijft staan. De opgegeven nummers zijn de volgnummers in de uitvoer-file.

Mocht onverhoopt de layout zijn verstoord, dan kan gebruik worden gemaakt van de volgende aanroep:

```
\stelloayoutin[reset]
```

De oplettende lezer zal gezien hebben dat bij het instellen van de breedte en hoogte het trefwoord `passend` kan worden gebruikt. In dat geval worden de breedte en hoogte automatisch berekend, iets dat vooral handig is bij schermlayouts, waar men vaak symmetrisch kop-, voet-, rug- en snijwit nastreeft. Het trefwoord `midden` stelt de breedte zo in dat het rug- en snijwit gelijk is en houdt geen rekening met randen en marges.

Op de volgende bladzijden tonen we enkele A5 bladspiegels gecentreerd op A4. De standaard instellingen (maten) leveren een soort compromis die goed bruikbaar is voor verslagen en notities. De instellingen passen zich automatisch aan het papierformaat aan. Let op het gebruik van het trefwoord `midden` bij het instellen van de breedte en hoogte. De overige commando's komen later aan de orde.



rechts

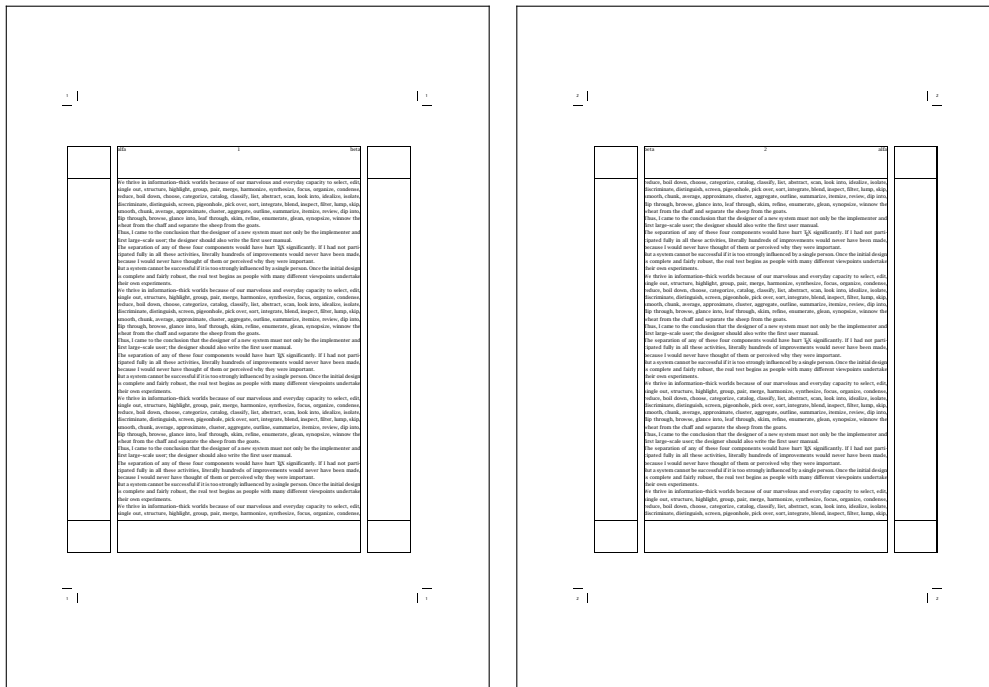
links

Figuur 3.2 De standaard verhoudingen (enkelzijdig).

```

\stelpapierformaatin [A5][A4]
\stellayoutin [plaats=midden,markering=aan]
\stelnummeringin [variant=enkelzijdig]
\stelkorpsin [lbr,6pt]
\stelhoofdtekstenin [alfa][beta]
\toonkader
\starttekst
\dorecurse{10}{\input tufte \par \input knuth \par}
\stoptekst

```



rechts

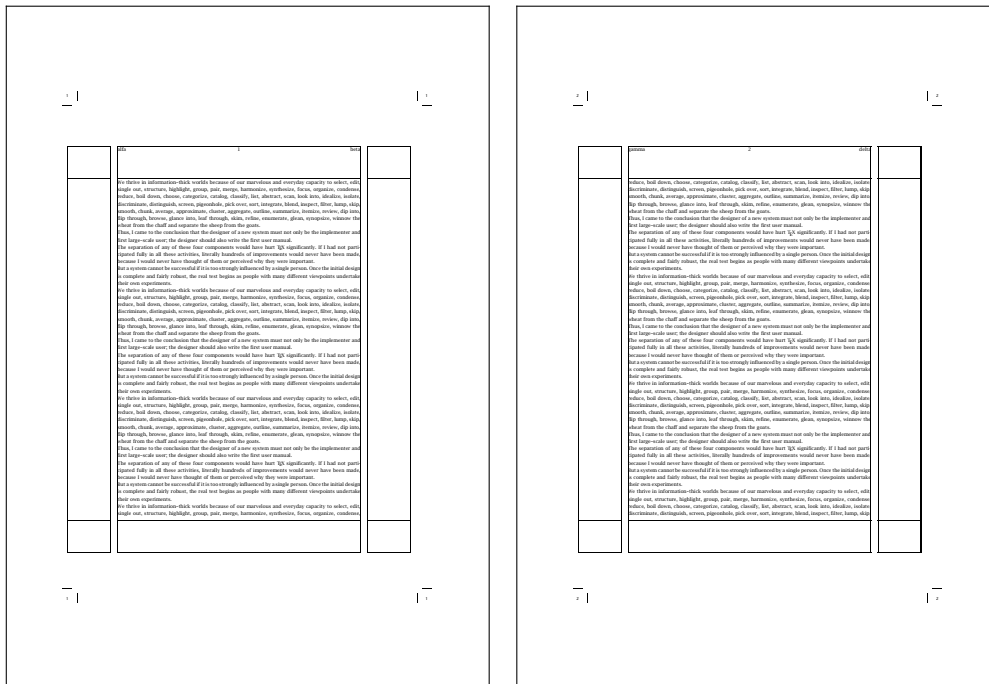
links

Figuur 3.3 De standaard verhoudingen (dubbelzijdig).

```

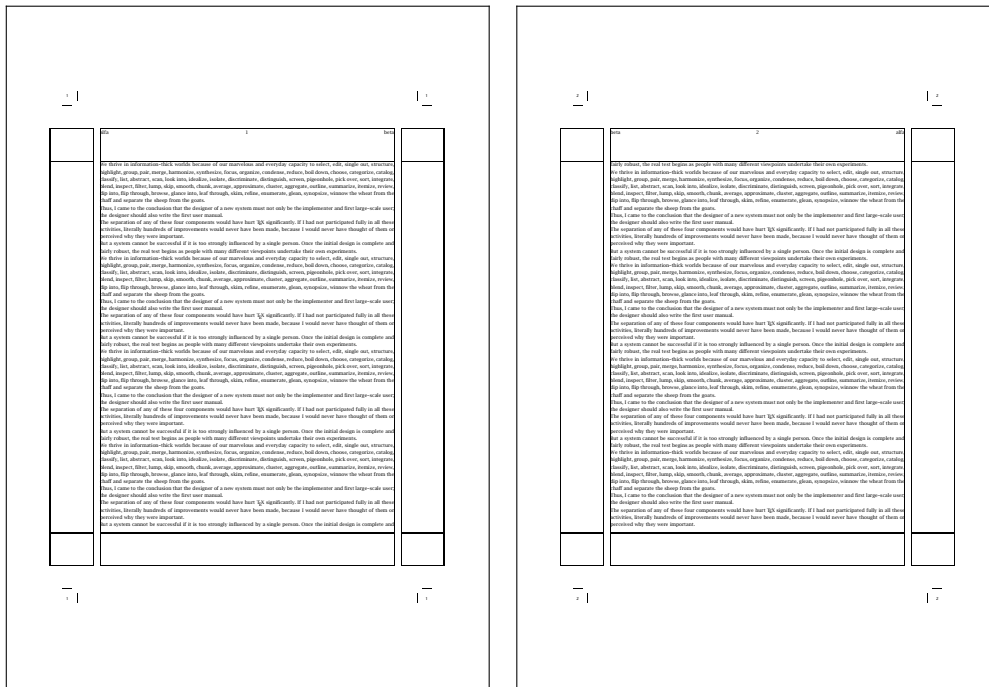
\stelpapierformaatin [A5][A4]
\stellayoutin [plaats=midden,markering=aan]
\stelnummeringin [variant=dubbelzijdig]
\stelkorpsin [lbr,6pt]
\stelhoofdtekstenin [alfa][beta]
\toonkader
\starttekst
\dorecurse{10}{\input tufte \par \input knuth \par}
\stoptekst

```



rechts links
Figuur 3.4 De standaard verhoudingen (enkel-/dubbelzijdig).

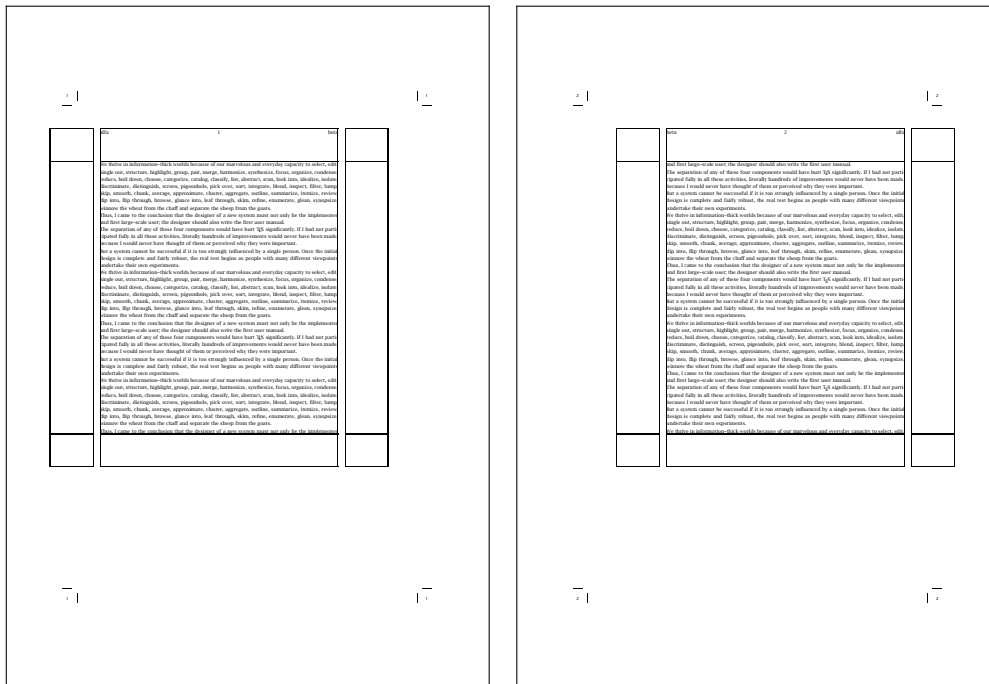
```
\stelpapierformaatin [A5][A4]
\stellayoutin [plaats=midden,markering=aan]
\stelnummeringin [variant={enkelzijdig,dubbelzijdig}]
\stelkorpsin [lbr,6pt]
\stelhoofdstekstenin [alfa][beta][gamma][delta]
\toonkader
\starttekst
\dorecurse{10}{\input tufte \par \input knuth \par}
\stoptekst
```



rechts links
Figuur 3.5 Een automatisch gecentreerde bladspiegel.

```
\stelpapierformaatin [A5][A4]
\stellayoutin [rugwit=1cm,breedte=midden,
               kopwit=1cm,hoogte=midden,
               plaats=midden,markering=aan]
\stelnummering [variant=dubbelzijdig]
\stelkorspsin [lbr,6pt]
\stelhoofdttekstenin [alfa][beta]

\toonkader
\starttekst
  \dorecuse{10}{\input tufte \par \input knuth \par}
\stoptekst
```



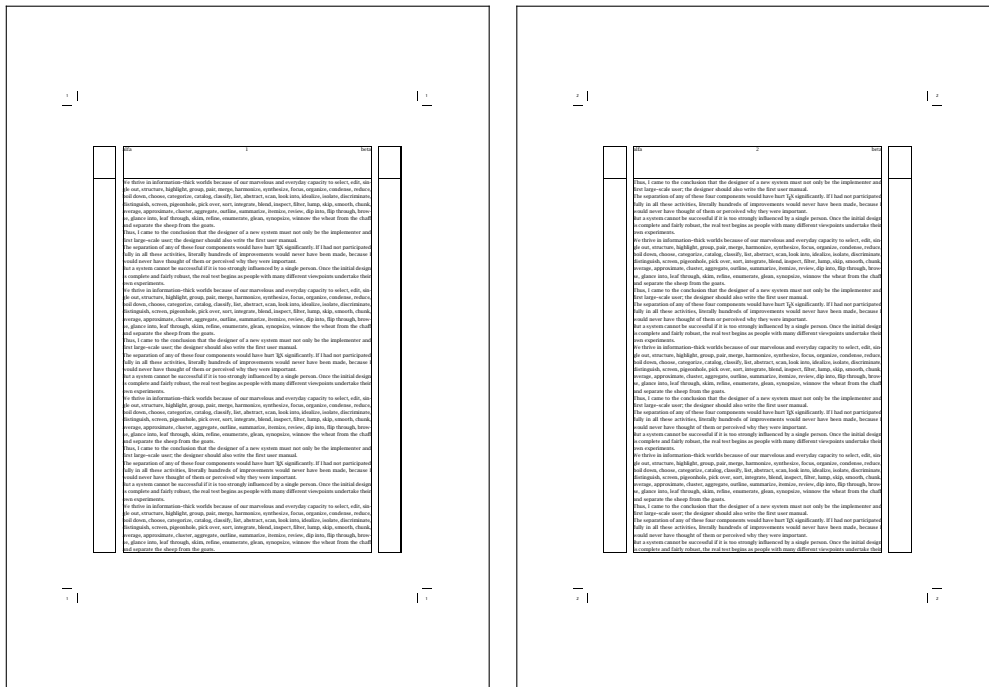
rechts

links

Figuur 3.6 Een a-symmetrische bladspiegel.

```
\stelpapierformaat in [A5][A4]
\stellayout in [rugwit=1cm,breedte=.7\papierbreedte,
kopwit=1cm,hoogte=.7\papierhoogte,
plaats=midden,markering=aan]
\stelnummering in [variant=dubbelzijdig]
\stelkorsps in [lbr,6pt]
\stelhoofdteksten in [alfa][beta]

\toonkader
\starttekst
\dorecurse{10}{\input tufte \par \input knuth \par}
\stoptekst
```



rechts

links

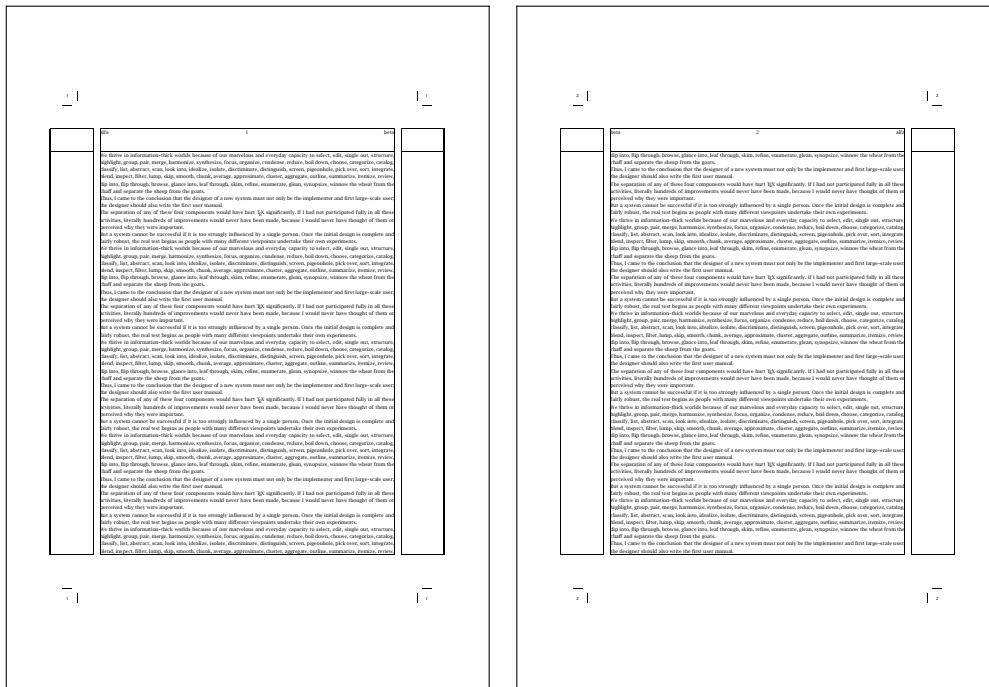
Figuur 3.7 Een bladspiegel zonder voet.

```

\stelpapierformaatin [A5][A4]
\stellayoutin [rugwit=2cm,breedte=midden,
voet=0cm,marge=1cm,
plaats=midden,markering=aan]
\stelnummeringin [variant=enkelzijdig]
\stelkorspin [lbr,6pt]
\stelhoofdtekstenin [alfa][beta]

\toonkader
\starttekst
\dorecurse{10}{\input tufte \par \input knuth \par}
\stoptekst

```

rechts

links

Figuur 3.8 Een bladspiegel gebaseerd op een grid.

```

\stelpapierformaatin [A5][A4]
\stellayoutin [plaats=midden,markering=aan]
\stelnummeringin [variant=dubbelzijdig]
\stelkorpstin [lbr,6pt]
\stelhoofdtekstenin [alfa][beta]
\stellayoutin [kopwit=1cm,regels=56,hoofd=1cm,voet=0cm,
               rugwit=1cm,breedte=midden,grid=ja]
\toonkader
\starttekst
  \dorecurse{10}{\input tufte \par \input knuth \par}
\stoptekst

```

3.4 Grids

Er zijn verschillende manieren om een bladzijde uit te lijnen, bijvoorbeeld:

alfa	alfa	alfa	alfa
beta	beta	beta	beta gamma
gamma	gamma	gamma	

De eerste varianten zijn in de regel ongewenst en kunnen worden bereikt door wat rek in de interlinie in te bouwen. De laatste variant heeft echter als nadeel dat bladzijden een verschillende lengte kunnen hebben. Om die reden voorzien we meestal het tussenwit van wat rek.¹

alfa beta gamma delta	alfa beta gamma delta	alfa beta gamma	alfa beta gamma
--------------------------------	--------------------------------	-----------------------	-----------------------

Een nadeel van deze benadering is dat de naast elkaar getoonde bladzijden of kolommen op een bladzijde zelden uitlijnen, wat vooral bij een wat grotere letter storend kan zijn. Bovendien kan bij dun papier een storend doorschijnend effect optreden.²

In dergelijke situaties gaat de voorkeur uit ² O la la, nog een voetnoot!
³ Ziezo, een laatste voetnoot!

Bij het zetten op een grid worden koppen, figuren, formules en natuurlijk de lopende tekst op een vaste regelafstand gezet. Mocht onverhoopt een typografische component verstorend werken, dan kan men zo'n component op het grid plaatsen (engels: snap) met:

```
\plaatsopgrid{\omlijnd{Snapt u hier wat van?}}
```

Dit levert:

Snapt u hier wat van?

Men kan het mechanisme beïnvloeden door een argument mee te geven:

¹ Oeps, een voetnoot!

```
\plaatsopgrid[onder]{\omlijnd{Snapt u het nu?}}
```

waarna onder de omlijnde tekst een extra lege regel wordt geplaatst. Andere opties zijn: `boven` en `beide`. In het laatste geval wordt de extra regel verdeeld:

```
Nu snapt u het zeker?
```

Beide omlijnde voorbeelden zijn geen schoolvoorbeelden van schoonheid. Dit komt omdat `\omlijnd` standaard geen diepte heeft. Omdat \TeX lijnen anders behandelt dan tekst, biedt `CON \TeX T` de volgende oplossing:

```
\startregelcorrectie
\omlijnd{Dit is iets voor gevorderden.}
\stopregelcorrectie
```

Dit commando probeert zo goed en kwaad als het kan een en ander te plaatsen en houdt daarbij zonnodig rekening met het grid.

```
Dit is iets voor gevorderden.
```

Omdat de regelcorrectie al rekening houdt met het grid, moeten we een ander commando gebruiken om de omkaderde tekst op te rekken:

```
\verplaatsopgrid[beide]
\startregelcorrectie
\omlijnd{Maar het kan geen kwaad een en ander te weten.}
\stopregelcorrectie
```

Inderdaad krijgen we nu wat meer ruimte:

```
Maar het kan geen kwaad een en ander te weten.
```

Men kan voor testdoeleinden het grid zichtbaar maken met het commando `\toongrid`. Samen-
gevat hebben we dus:

```
\plaatsopgrid[.1.]{.2.}
.1. zie p 38: \verplaatsopgrid
```

```
\verplaatsopgrid[...]
```

```
... boven beide onder
```

```
\toongrid
```

Als het grid getoond wordt, is in de linker marge te zien op welke wijze correctie plaatsvindt. Van links naar rechts zien we: de witruimte boven het element, de ruimte boven de onderste baseline van het element en de ruimte eronder. De volgende twee posities geven een eventuele extra verplaatsing aan en de laatste positie het volgnummer van de correctie. Dit soort informatie is eigenlijk alleen zinvol om fouten in het mechanisme op te sporen en/of wanneer men zelf macros schrijft waarvan het resultaat op het grid moet worden geplaatst.

3.5 Printen

In een eerdere paragraaf hebben we onderscheid gemaakt tussen pagina- en papier-afmetingen. We zullen nu wat dieper ingaan op de wijze waarop deze twee kunnen worden gemanipuleerd.

In figuur 3.10 en 3.11 zien we enkele mogelijkheden om de bladspiegel te manipuleren met behulp van instellingen van de eerder besproken commando's `\stelpapierformaat` en `\stellayout`. Het is dus mogelijk een pagina in een hoek of het midden van het papier te plaatsen, te kopiëren en te voorzien van markeringen.

Wanneer we bij het papierformaat `liggend` opgeven, dan worden breedte en hoogte omgewisseld. Er wordt dus niet geroteerd! Dit laatste doen we met de instellingen 90, 180 en 270.

```
\stelpapierformaat[A5,liggend][A4]
```

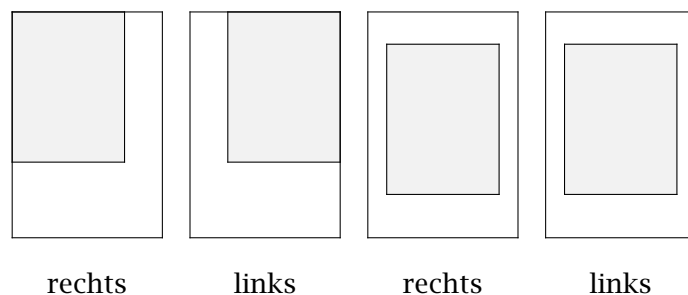
Wat de voorbeelden niet laten zien, is dat we ook kunnen corrigeren voor dubbelzijdig printen. Wanneer we zeggen:

```
\stelpapierformaat[A5][A4]
\stellayout[plaats=midden,markering=aan]
```

dan komt op voor- én achterkant de tekst netjes in het midden te staan. De markering maakt het mogelijk het juiste formaat af te snijden. Als we echter maar twee keer willen snijden, dan kunnen we ook opgeven:

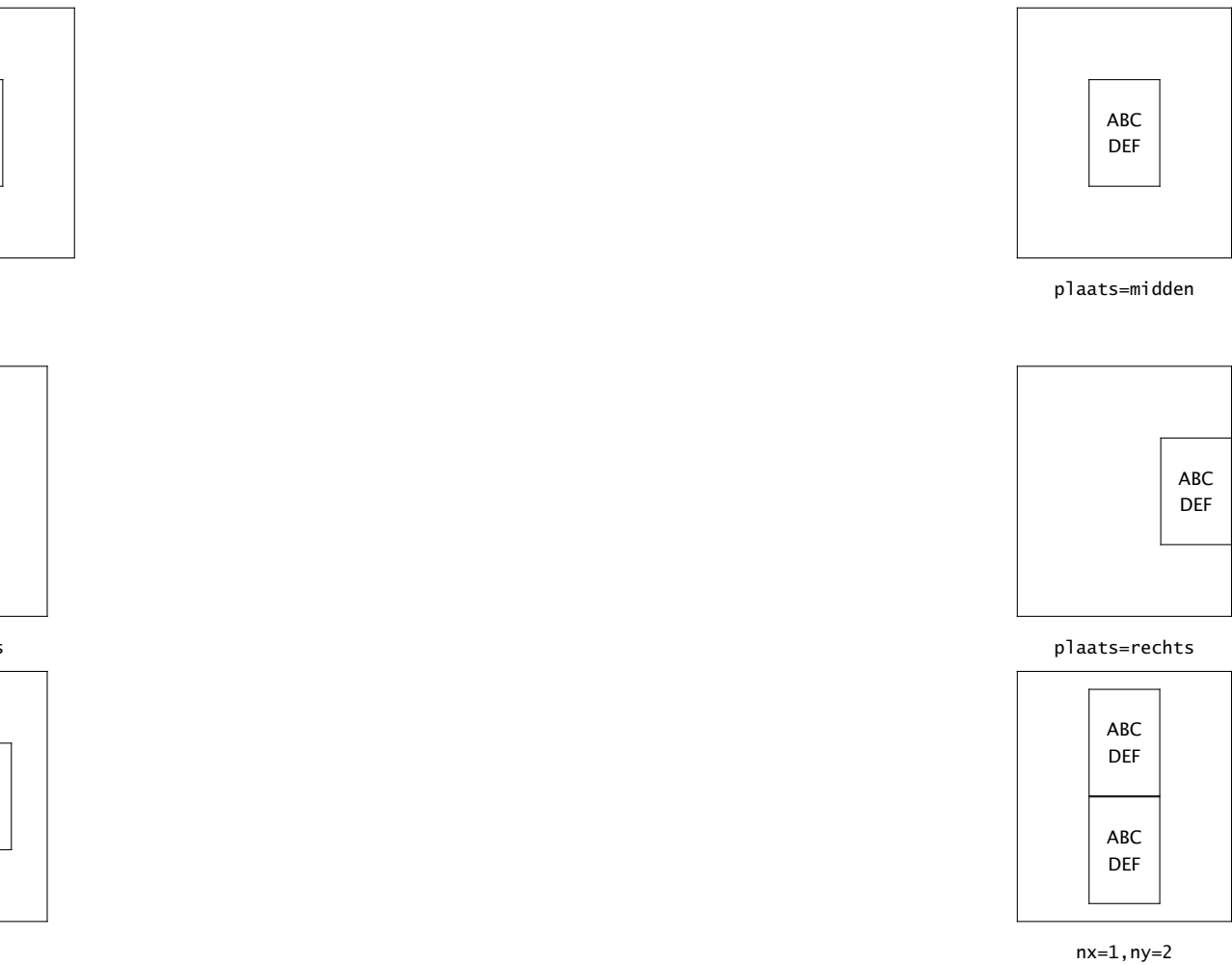
```
\stelpapierformaat[A5][A4]
\stellayout[plaats=dubbelzijdig]
```

Dit komt overeen met `{dubbelzijdig,links}`. Bij deze instelling zorgt `CONTEXT` er voor dat de achterzijde automatisch wordt verschoven naar de juiste hoek. In figuur 3.9 laten we twee varianten zien.

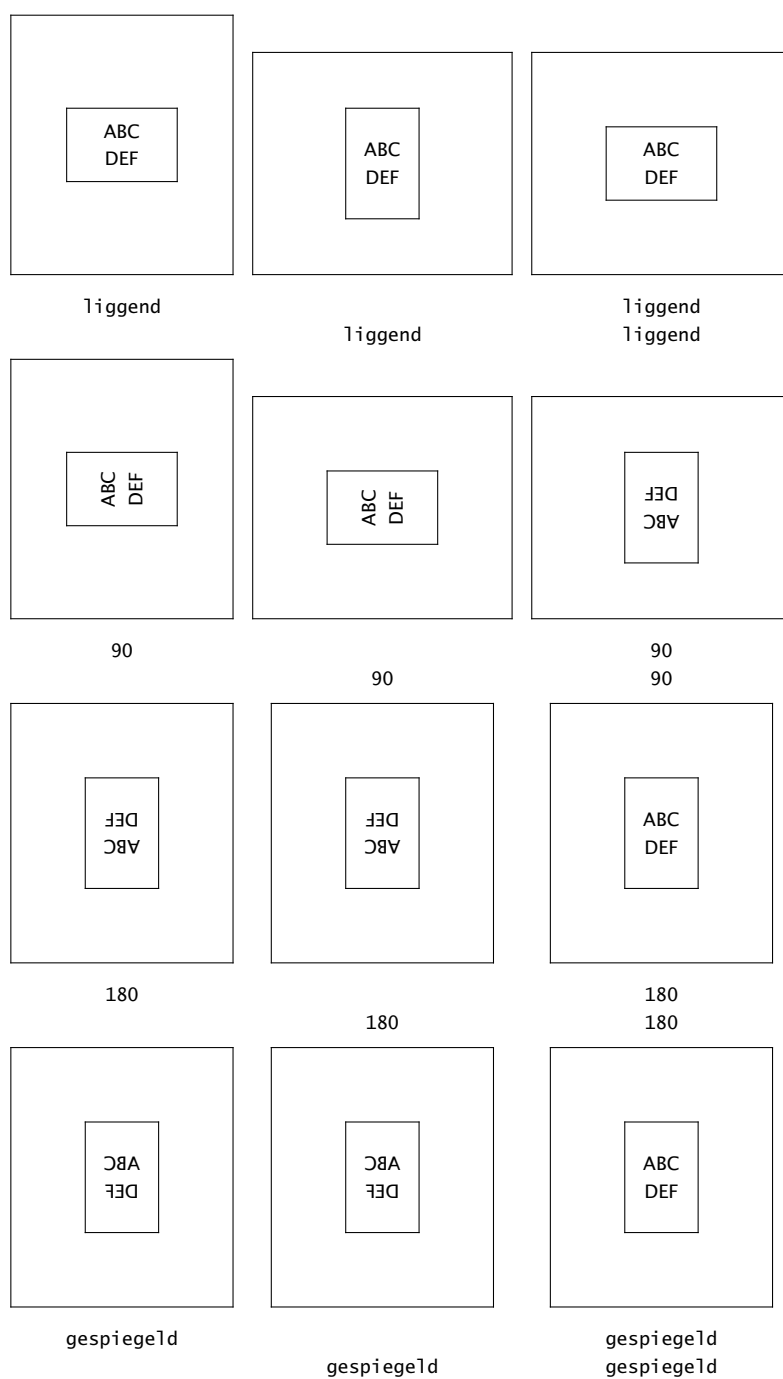


Figuur 3.9 Het positioneren van de bladspiegel ten behoeve van het afsnijden.

Roteren, spiegelen, schalen, dupliceren en plaatsen zijn onafhankelijke operaties. Door ze slim te combineren kunnen echter vrij veel effecten worden bereikt. Roteren en spiegelen worden ingesteld tegelijk met het pagina- en papierformaat. De overige operaties worden ingesteld als layout opties.



Figuur 3.10 Het manipuleren van de bladspiegel met `\stllayoutin`.



Figuur 3.11 Het manipuleren van de bladspiegel met `\stelpapierformaat`.

```
\toonprint[...1,...][...2,...][...=...]
```

```
..=.. zie p 23: \stelpapierformaat  
..=.. zie p 23: \stelpapierformaat  
..=.. zie p 26: \stellayoutin
```

Met `\toonprint` kunnen we een en ander uitproberen. Let wel, dit commando toont niet de echte pagina maar een dummy, zoals in de eerdere voorbeelden. Het meegeven van voorgedefinieerde formaten heeft geen zin.

```
\toonprint[gespiegeld][90][plaats=midden]
```

3.6 Arrangeren

Met behulp van `\stellayoutin` kan men eenzelfde pagina meerdere malen op een blad papier weergeven. Wellicht interessanter is de mogelijkheid om de bladzijden te ordenen in katernen.

```
\stelarrangerenin[.,.,.,.,.]
...   b1okkeer 2*16 2*8 2*4 2*2 2**2 2UP 2DOWN gespiegeld geroteerd dubbelzijdig negatief 90
      180 270
```

Dit commando is er een voor gevorderden. Op de volgende bladzijden laten we wat arrangementen zien. Het meeste inzicht verwerft men als men zelf de mogelijkheden onderzoekt. We laten eerst een aantal alternatieve schikkingen zien.

De voorbeelden op de volgende bladzijden tonen op welke wijze de commando's `\stelpapierformatin`, `\stellayoutin` en `\stelarrangerenin` samenwerken. We laten meteen zien hoe deze testfiles zijn gegenereerd.

8	9	12	5	6	11	10	7
1	9I	8I	7	8	7I	6I	2

Figuur 3.12 De 2*8 ordening.

4	5	3	6
1	8	2	7

Figuur 3.13 De 2*4 ordening.

1	4	3	2
---	---	---	---

Figuur 3.14 De 2*2 ordening.

1	8	2	7	3	6	4	5
---	---	---	---	---	---	---	---

Figuur 3.15 De 2UP ordening.

8	7	6	5
1	2	3	4

Figuur 3.16 De 2DOWN ordening.

```

\stelpapierformaatin [A7][A3]
\stelarrangerenin    [2*8,geroteerd,dubbelzijdig]
\stelnummeringin     [variant=dubbelzijdig]
\stellayoutin        [marge=0pt,breedte=passend]
\stelachtergrondenin [tekst][tekst][achtergrond=raster]
\stelkleurenin       [status=start]
\stellayoutin        [plaats=midden,markering=kleur]
\steltolerantiein    [soepel]
\stelkorpsin         [lbr,6pt]

```

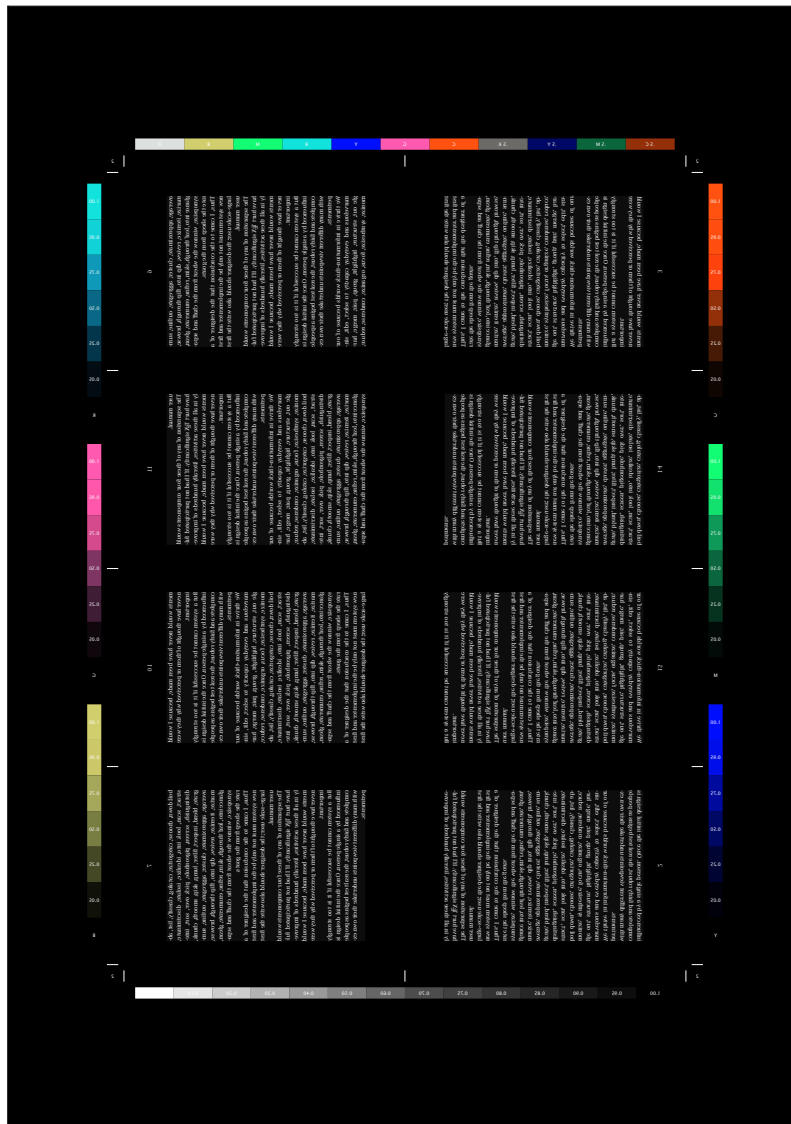


Figuur 3.17 Arrangeren: 16.

```

\starttekst
  \dorecurse{30}{\input tufte \par \input knuth \par}
\stoptekst

```



Figuur 3.18 Arrangeren: negatief gespiegeld 16.

```

\stelpapierformaat [A7] [A3,negatief,gespiegeld]
\stelarrangerenin [2*8,geroteerd,dubbelzijdig]
\stelnummeringin [variant=dubbelzijdig]
\stellayoutin [marge=0pt,breedte=passend]
\stelachtergrondenin [tekst][tekst][achtergrond=raster]
\stelkleurenin [status=start]
\stellayoutin [plaats=midden,markering=kleur]
\steltolerantiein [soepel]
\stelkorpsin [lbr,6pt]
\starttekst
\dorecurse{30}{\input tufte \par \input knuth \par}
\stoptekst
    
```

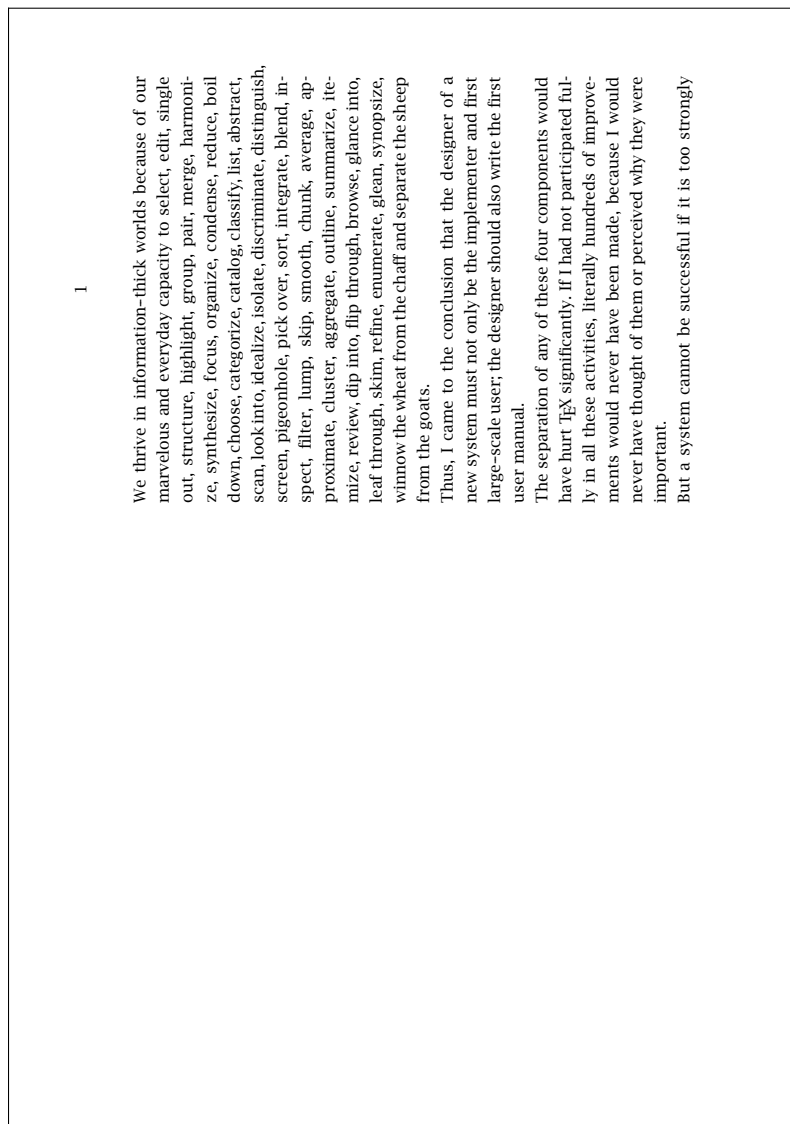


Figuur 3.19 Arrangeren: 8.

```

\stelpapierformaatin [A6][A3]
\stelarrangerenin [2*4,dubbelzijdig]
\stelnummeringin [variant=dubbelzijdig]
\stellayoutin [marge=0pt,breedte=passend]
\stelachtergrondenin [tekst][tekst][achtergrond=raster]
\stelkleurenin [status=start]
\stellayoutin [plaats=midden,markering=kleur]
\steltolerantiein [soepel]
\stelkorpsin [lbr,7pt]
\starttekst
\dorecurse{30}{\input tufte \par \input knuth \par}
\stoptekst

```

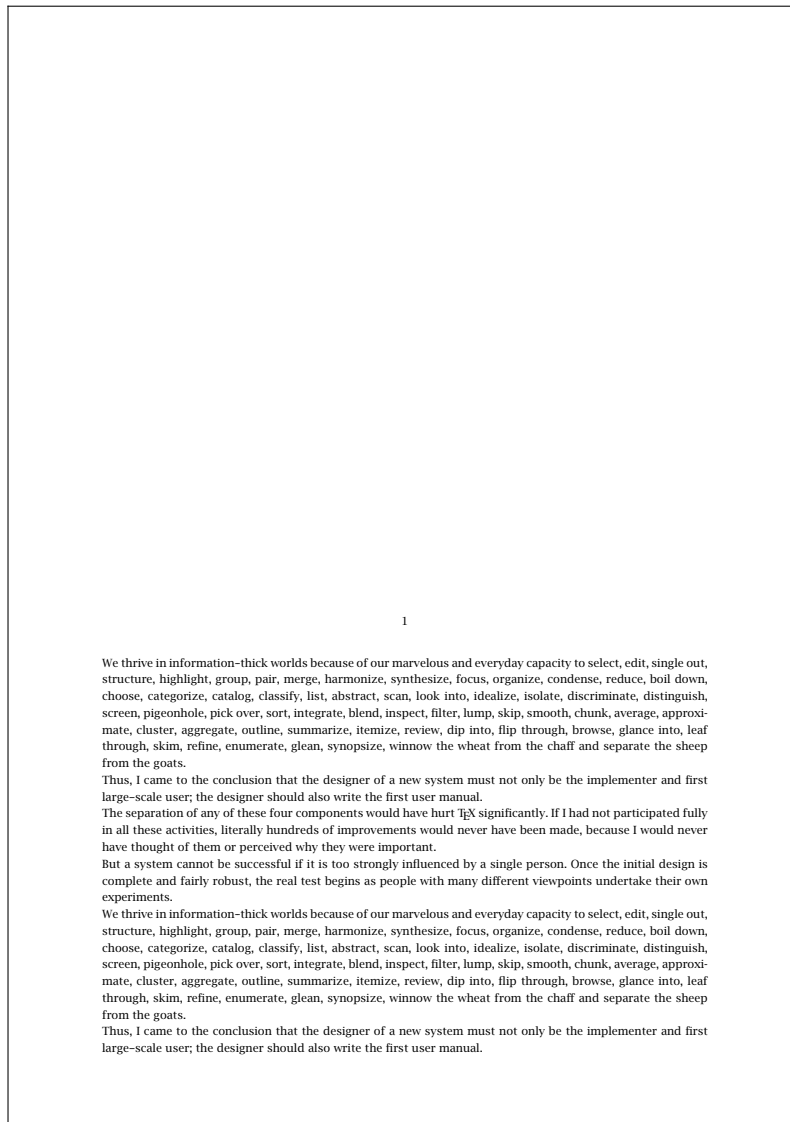
Figuur 3.22 Arrangeren: 2UP (2).

```

\stelpapierformaat in [A5][A4]
\stelarrangeren in   [2UP,geroteerd,dubbelzijdig]
\stelnummering in   [variant=dubbelzijdig]
\stellayout in      [marge=0pt,breedte=passend]
\stelkors in        [lbr,12pt]

\starttekst
  \dorecurse{30}{\input tufte \par \input knuth \par}
\stoptekst

```



Figuur 3.23 Arrangeren: 2DOWN.

```

\stelpapierformaat in [A4,liggend][A3]
\stelarrangerenin    [2DOWN,dubbelzijdig]
\stelnummeringin     [variant=dubbelzijdig]
\stellayoutin        [marge=0pt, breedte=passend]
\stelkorpsin         [1br,12pt]

\starttekst
  \dorecurse{30}{\input tufte \par \input knuth \par}
\stoptekst

```


3.7

Beeldmerken

Het is mogelijk onder- of bovenaan de bladzijde beeldmerken op te nemen. We zullen daar op de volgende bladzijden enkele voorbeelden van laten zien. Aangeraden wordt voor het zetten van een beeldmerk een commando te definiëren.

De (plaats van) de beeldmerken wordt vastgelegd met het commando:

```
\definieerbeeldmerk[.1.][.2.][.3.][...=...]
```

.1.	<i>naam</i>
.2.	boven hoofd voet onder
.3.	geen pagina linkerrand linkermarge links midden rechts rechtermarge rechterrand
commando	<i>commando tekst</i>
status	start <u>stop</u>

Alle beeldmerken met status=start worden automatisch geplaatst. Beeldmerken kunnen echter ook worden opgeroepen:

```
\plaatsbeeldmerken[...,...]
```

... *naam*

In dat geval worden alleen de beeldmerken geplaatst die in de lijst zijn opgenomen, ongeacht status.

Op deze bladzijde zijn enkele plaatsen van een beeldmerk aangegeven. We hebben hier tijdelijk de hoofd- en voetregels onderdrukt. De linker beeldmerken zijn bijvoorbeeld als volgt gedefinieerd:

```
\definieerbeeldmerk  
  [logo a] [onder] [links]  
  [commando=links onder]  
\definieerbeeldmerk  
  [logo d] [boven] [links]  
  [commando=links boven]  
\definieerbeeldmerk  
  [logo g] [voet] [links]  
  [commando=links voet]  
\definieerbeeldmerk  
  [logo j] [hoofd] [links]  
  [commando=links hoofd]
```

```
\plaatsbeeldmerken[logo a,logo b,logo c,logo d]
```

In plaats van *commando* hadden we ook voor *tekst* kunnen kiezen. Door te kiezen voor *commando* geven we de richting aan waarin gedacht moet worden bij beeldmerken. Omdat een beeldmerk meestal meerdere malen wordt gebruikt, ligt het namelijk voor de hand een *commando* te definiëren. We werken dit in een voorbeeld uit.



Allereerst definiëren we een commando dat een klein beeldmerk zet.

```
\def\ContextLogo%  
  {\externfiguur[mp-cont.502][hoogte=24pt]}
```

Als we dit beeldmerk midden onderaan elke bladzijde willen zetten, dan geven we dat aan met:

```
\definieerbeeldmerk  
  [klein logo] [onder] [midden]  
  [commando=\ContextLogo,status=start]
```

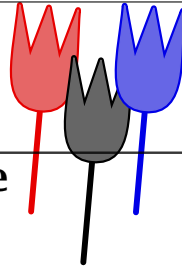
Dit beeldmerk komt op elke (!) bladzijde te staan. Bij briefpapier hebben we echter vaak te maken met een wat anders weergegeven beeldmerk, op een wat andere plaats. We definiëren eerst het (grotere) beeldmerk, ditmaal inclusief adressering. Dit voorbeeld laat tevens zien dat we `\omlijnd` kunnen gebruiken voor meer dan omlijnde teksten. Dergelijke toepassingen vergen echter enige oefening.

```
\def\ContextBriefhoofd%  
  {\hbox  
    {\definieerfont[ContextFont][RegularBold sa 1.5]%  
    \ContextFont \stelinterliniein  
    \stelomlijndin  
    [uitlijnen=midden,boven=\vfill,onder=\vfill,  
    hoogte=10\korpsgrootte,offset=overlay,kader=uit]%  
    \omlijnd  
    {The\ConTeX t\Chronicle}%  
    \externfiguur  
    [mp-cont.502][hoogte=10\korpsgrootte]%  
    \omlijnd  
    {Ridderstraat 27\8061GH Hasselt NL\pragma@wxs.nl}}}
```

Ook hier leggen we de plaats vast:



The
ConT_EXt
Chronicle



Ridderstraat 27
8061GH Hasselt NL
pragma@wxs.nl

`\definieerbeeldmerk`

`[groot logo] [hoofd] [rechts]`
`[commando=\ContextBriefhoofd]`

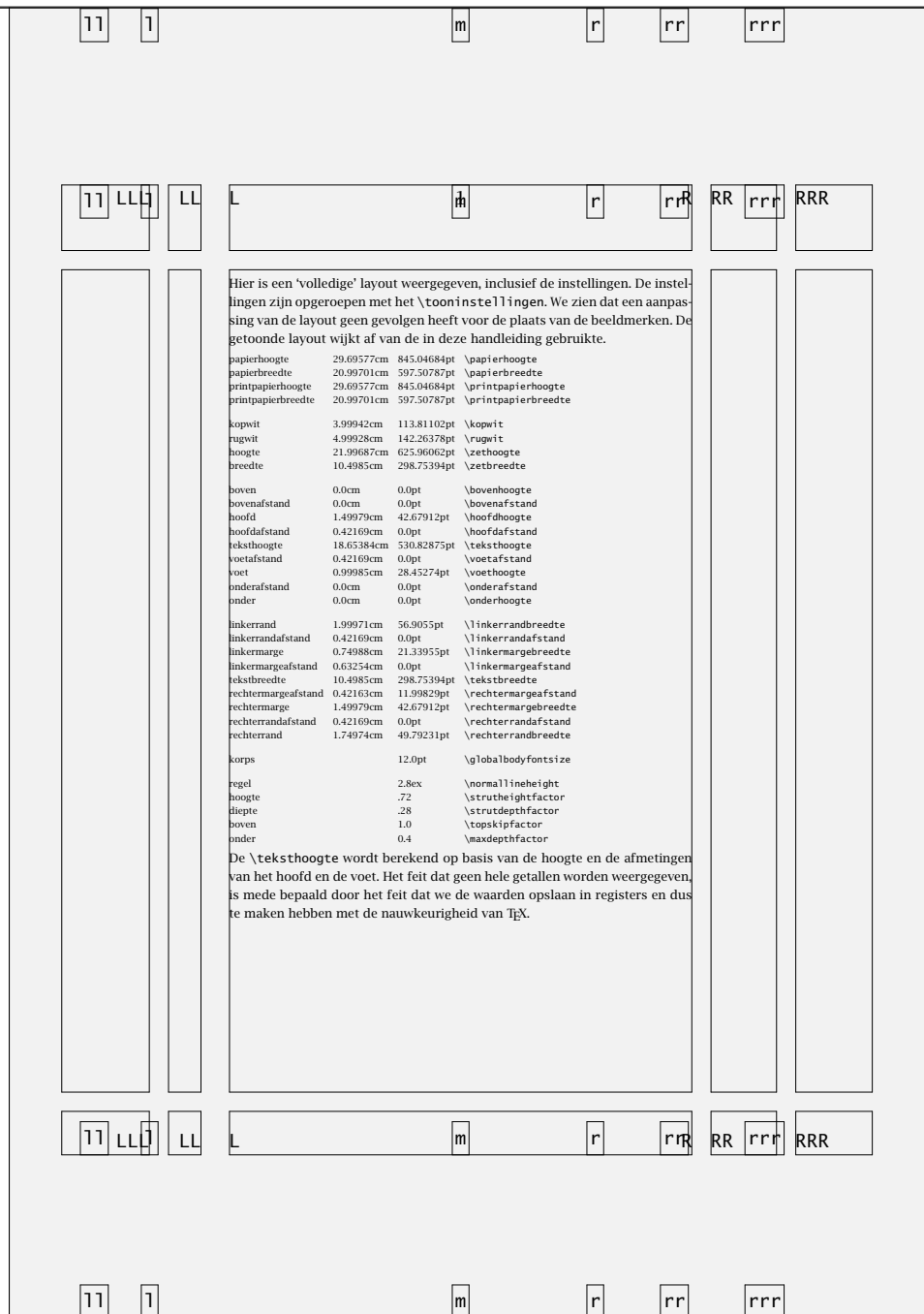
In tegenstelling tot het vervolgvel, kennen we hier aan status *niet* de waarde start toe. We willen immers niet op iedere bladzijde een groot beeldmerk. Als we dit beeldmerk eenmalig willen oproepen, dan doen we dat met:

`\plaatsbeeldmerken[groot logo]`

We zien dat het logo onderaan niet wordt geplaatst. Dit komt omdat het commando `\plaatsbeeldmerken` alle niet opgegeven beeldmerken onderdrukt.

Normaal gesproken dient de tekst wat lager te beginnen, dit kan bijvoorbeeld worden bewerkstelligd door het commando:

`\blanko[forceer,8*rege]`



Figuur 3.24 De plaats van hoofd-, voet-, boven- en onderteksten en beeldmerken in de layout.

4

Layout

Layout

4.1 Inleiding

Het voorkomen van een publicatie wordt in grote mate bepaald door de vlakverdeling, de keuze voor lettertypes en zaken als spatiëring. In dit hoofdstuk besteden we aandacht aan het laatste aspect. Er worden soms vrij veel details vermeld. Een beginnend gebruiker heeft eigenlijk weinig te maken met dit soort dingen, omdat de standaardinstellingen redelijk neutraal zijn en zich aanpassen aan wisselende omstandigheden. Voor de ongeduldige lezer zijn hier vast wat instellingen. Witruimte tussen alinea's krijgt men met:

```
\stelwitruimtein[groot]
```

Wil men bij elke alinea een ingesprongen eerste regel, zeg dan:

```
\stelinspringenin[middel]
```

Een dubbelzijdige layout tenslotte is te realiseren met:

```
\stelnummeringin[variant=dubbelzijdig]
```

Overigens, gebruik tussen alinea's een lege regel. Wie nu nog meer wil weten, moet doorlezen.

4.2 Alineas

De (binnen $\text{T}_\text{E}\text{X}$) belangrijkste eenheid van tekst is de paragraaf. Een overgang naar een nieuwe paragraaf (alinea) wordt op een van de volgende manieren afgedwongen:

1. een lege regel
2. het $\text{T}_\text{E}\text{X}$ -commando `\par` of `\endgraf`
3. het $\text{CON}\text{T}_\text{E}\text{X}\text{T}$ -commando `\alinea`

Het verdient aanbeveling in de ASCII-file een ruime layout te kiezen, niet zozeer omdat dit moet, maar eerder omdat het de inzichtelijkheid vergroot. Bovendien voorkomen we zo fouten. Het eerste alternatief (lege regels) zal dan ook het meest worden gebruikt. Voor het tweede alternatief zal met name worden gekozen wanneer een commando expliciet moet worden afgesloten met `\par`. Dit is het geval bij definities (zie 10.2).

4.3 Interlinie

Binnen $\text{T}_\text{E}\text{X}$ wordt de ruimte tussen de regels, de interlinie, bepaald door een aantal instelbare dimensie variabelen, waaronder `\topskip`, `\parskip` en `\baselineskip`. Al dit soort instellingen zijn binnen $\text{CON}\text{T}_\text{E}\text{X}\text{T}$ gerelateerd aan de korpsgrootte.

Een regel heeft een hoogte en een diepte. De afstand tussen twee regels is in de normale situatie gelijk aan de som van de maximale hoogte en maximale diepte:

$$\blacksquare + \blacksquare = \blacksquare$$

Deze som is binnen $\text{CON}\text{T}\text{E}\text{X}\text{T}$ gelijk aan $2.8ex$, dus bijna drie maal de hoogte van een x , wat vrijwel overeenkomt met $1.2\times$ de korpsgrootte. De verhouding tussen de maximale hoogte en diepte is standaard $.72 : .28$. Hoewel een ex (■) afhangt van het gebruikte lettertype, schaalt de regelafstand alleen mee bij een echte wisseling van korps, of wanneer men expliciet de regelafstand opnieuw instelt:

```
\bfd \stelinterliniein Nu is de regelafstand wat groter ... \par
```

Men dient in het oog te houden dat TEX op paragrafen werkt, vandaar dat men, als men binnen een groep werkt, altijd de alinea moet afsluiten met een lege regel of met `\par`. In veel gevallen regelt $\text{CON}\text{T}\text{E}\text{X}\text{T}$ dit overigens zelf.

Vaak heeft een regel niet de maximale hoogte of diepte, zoals uit het volgende voorbeeld blijkt:

■ De hoogte en diepte van regels verschillen.

Hier staat:

De hoogte en diepte van regels verschillen.

Als we twee van deze regels onder elkaar plaatsen krijgen we:

■ De hoogte en diepte van regels verschillen.
■ De hoogte en diepte van regels verschillen.

We zien dat de afstand wat hoger is dan de som van de hoogte en diepte van een individuele regel. Deze afstand noemen we de regelafstand (`\baselineskip` is hier 14.83998pt). Als we aan deze regel een extra hoog karakter toevoegen zien we het volgende:

■ De hoogte en diepte van regels verschillen. ■
■ De hoogte en diepte van regels verschillen.

Om te voorkomen dat de regels op elkaar staan, voegt TEX een `\lineskip` tussen, in ons geval 1.0pt (afgezien van schaling). Op vergelijkbare wijze zorgt TEX er voor dat de eerste regel tenminste de afstand `\topskip` (hier 10.6848pt) tot de bovenkant van de zetspiegel heeft.

De interlinie kan worden ingesteld met behulp van het commando:

```
\stelinterliniein[...]| [...,...=...,...]  
...      reset klein middel groot aan uit  
hoogte   getal  
diepte   getal  
regel    maat  
boven    getal  
onder    getal
```

Daarbij past de interlinie zich aan het actuele korps aan. In plaats van een factor kan ook een van de vaste instellingen `klein` (1.0), `middel` (1.25) of `groot` (1.5) worden meegegeven. Hieronder is een voorbeeld gegeven van een tekst met een interlinie van 1.3. Ook de witruimte wordt groter!

Uitgaande van het gegeven dat “waar veel van is, ook wel iets zal overblijven”, bekruipt mij soms het angstige gevoel dat men over vele eeuwen tot de bedroevende conclusie zal komen dat het grote aantal door de computer ge(re)produceerde pop-hits het toppunt van kunnen vertegenwoordigde in de 20^e eeuw. Wordt er immers ook niet wild gespeculeerd bij een enkele oude munt uit de eerste eeuw? Aan de andere kant illustreert een dergelijke constatering wel goed het onvermogen van veel luisteraars om kwaliteit te onderscheiden en het gebrek aan zelfkritiek of juist te veel aan eigendunk van de ‘uitvoerende musici’. Dat tijdsbeeld zal dus kloppen.

Als overgegaan wordt op een ander formaat letter, dan kan (tussen paragrafen!) de interlinie worden aangepast door het commando `\stelinterliniein` zonder instellingen te geven of door de instelling te beginnen met `reset`, bijvoorbeeld: `[reset,midde]`.

Meestal gebruikt men in kinderboeken een wat grotere letter, bijvoorbeeld omdat men ervan overtuigd is dat kinderen het boek dan zelf (beter) kunnen lezen. Misschien gebeurt het ook wel automatisch. Helaas zag ik nog vrij recent een voorbeeld van de gebrekkige kwaliteit van de hierbij gebruikte zetsystemen en/of het beperkte inzicht van de gebruiker hiervan. De interlinie wisselde per regel en hing, zo leek het tenminste, onder andere af van de **hoogte** van de regel. Denk er daarom altijd aan om bij het veranderen van een letterformaat, met als doel een stuk tekst te zetten, de interlinie aan te passen.

De bovenstaande tekst is gezet in formaat `\tfa` wat hier gelijk is aan `\rma` (meer hierover in hoofdstuk 5).

```
\start \tfa \stelinterliniein Meestal gebruikt men in kinderboeken een
wat grotere ... aan te passen \par \stop
```

In dit voorbeeld is de `\par` nodig omdat T_EX anders de alinea na de `\stop` afsluit, en op dat moment gelden de oude waarden weer; T_EX handelt immers hele alinea's af!

In plaats van een trefwoord kan men ook instellingen meegeven die de kenmerken van een regel bepalen. De standaard instellingen zijn:

```
\stelinterliniein
  [height=.72,
   depth=.28,
   top=1.0,
   bottom=0.4,
   line=2.8ex]
\stoptyping
```

De `\type {hoogte}` en `\type {diepte}` bepalen de verhouding tussen de hoogte en diepte van een regel. De regelafstand is ingesteld op 2.8ex. De parameters `\type {boven}` en `\type`

{onder} bepalen de relatie tussen de korpsgrootte en de hoogte van de eerst regel en diepte van de laatste. De beide laatste instellingen zijn gerelateerd aan \TEX's \type {\topskip} en \type {\maxdepth}.

```
\paragraaf[inspringen]{Inspringen}
\index{inspringen}
\index{smaller zetten}
\index{paragrafen+inspringen}
\macro{\tex{stelinspringenin}}
\macro{\tex{inspringen}}
\macro{\tex{nietinspringen}}
\macro{\tex{startsmaller}}
\macro{\tex{stelsmallerin}}
```

Wanneer een tekst weinig onderbrekingen kent, iets dat bijvoorbeeld in een roman het geval is, dan wordt meestal de eerste regel van een alinea ingesprongen. Het inspringen van een alinea wordt ingesteld met:

```
\toonsetup{stelinspringenin}
```

Standaard wordt \citeer {nooit} ingesprongen. Wanneer wordt ingesprongen, bepalen de meeste commando's, voor zover dat mogelijk is, zelf of ingesprongen moet worden of niet. Zo zal er bij voorkeur niet worden ingesprongen na verticale witruimte. Het gedrag van \CONTEXT\ zal op dit punt nog worden geoptimaliseerd. In een aantal gevallen, bijvoorbeeld bij opsommingen of koppen, kan men het inspringen van de volgende alinea instellen met de instelling \type {springvolgendein}.

Deze handleiding is gezet zonder inspringen. Bij de veelheid aan korte regels en witruimte tussen de alineas zouden ingesprongen alineas een rommelig beeld opleveren. Wanneer we een document zetten met inspringen, zullen we in een aantal gevallen nadrukkelijk moeten aangeven wanneer we {\em niet} willen inspringen. Dit doen we met:

```
\toonsetup{nietinspringen}
```

We kunnen ter plekke het inspringen anders instellen met:

```
\toonsetup{inspringen}
```

De betekenis van de instellingen is in \in {tabel} [tab:inspringen] weergegeven. Naast deze twee commando's kunnen we vaak ook de \TEX|commando's \type{\indent} en \type{\noindent} gebruiken.

```

\startbuffer
\starttabel[|1|1|]
\HL
\VL \bf instelling \VL \bf gevolg \VL\SR
\HL
\VL \type{nee}
\type{niet} \VL eenmalig niet inspringen \VL\FR
\VL \type{ja}
\type{altijd} \VL inspringen aanzetten \VL\MR
\VL \type{nooit} \VL inspringen uitzetten \VL\MR
\VL \type{eerste} \VL eerste alinea's inspringen \VL\MR
\VL \type{volgende} \VL eerste alinea's niet inspringen \VL\LR
\HL
\stoptabel
\stopbuffer
\plaatstabel
[hier][tab:inspringen]
{De wijze van inspringen.}
{\haalbuffer}

```

De instellingen `\type {eerste}` en `\type {volgende}` hebben betrekking op het al dan niet inspringen van een alinea na witruimte. Het is gebruikelijk dat niet te doen.

Een tekst kan smaller gezet worden dan standaard is ingesteld. In dat geval wordt zowel aan de linker- als aan de rechterkant ingesprongen.

```
\toonsetup{startsmaller}
```

Zo levert bijvoorbeeld:

```

\startbuffer
\startsmaller[3*links,2*rechts]
De relatief kleinschalige revoluties in Rusland in 1917 hebben grote
gevolgen gehad voor dit land en de rest van de wereld. Het is interessant
te kunnen concluderen dat er een evenzo kleinschalige revolutie voor nodig
was om een en ander weer ongedaan te maken. Bovendien lag de aanleiding voor
de omwentelingen in beide gevallen in angst voor een opkomende democratie.
\stopsmaller
\stopbuffer
\startvoorbeeld
\typebuffer
\stopvoorbeeld

```

het volgende op:

```
\startrealiteit
\haalbuffer
\stoprealiteit
```

Naast de instellingen `\type {links}`, `\type {rechts}` en `\type {midden}` zijn ook combinaties en/of veelvoudigen hiervan mogelijk. Het inspringen hierboven had ook kunnen worden bewerkstelligd door de instelling `\type {2*midden,links}`.

De afstand waarover wordt ingesprongen wordt ingesteld met het commando:

```
\toonsetup{stelsmallerin}
\paragraaf[witruimte]{Witruimte}
\index{witruimte}
\index{paragrafen+witruimte}
\index{spati\"ering}
\macro{\tex{blanko}}
\macro{\tex{stelblankoin}}
\macro{\tex{omlaag}}
\macro{\tex{startopelkaar}}
\macro{\tex{stelwitruimtein}}
\macro{\tex{geenwitruimte}}
\macro{\tex{witruimte}}
\macro{\tex{startregelcorrectie}}
\macro{\tex{corrigeerwitruimte}}
```

Het is mogelijk automatisch tussen elke paragraaf witruimte op te nemen. Deze witruimte wordt ingesteld met:

```
\toonsetup{stelwitruimtein}
```

In plaats van een maat kan (beter) een van de vaste instellingen worden meegegeven. Standaard wordt geen witruimte opgenomen tussen alinea's. Als dit commando wordt gegeven zonder argument, dan past de ingestelde witruimte zich aan het formaat van het actuele lettertype aan!

Met de volgende twee commando's kan witruimte worden geforceerd dan wel opgeheven. Beide commando's hebben alleen effect als de witruimte is ingesteld. Beide commando's zijn als het goed is zelden nodig.

```
\toonsetup{witruimte}
\toonsetup{geenwitruimte}
```

`\TEX\` handelt de witruimte rond horizontale lijnen op een afwijkende manier af. In deze en andere problematische situaties kan men de volgende commando's gebruiken. Een

aantal commando's, bijvoorbeeld die bij tabellen, regelen dit zelf.

```
\toonsetup{startregelcorrectie}
```

Bijvoorbeeld:

```
\starttypen
```

```
\startregelcorrectie
```

```
\omlijnd{To boxit or not, that's a delicate question!}
```

```
\stopregelcorrectie
```

De instelling `blanko` zorgt ervoor dat de standaard ingestelde `\blanko` wordt uitgevoerd. Gebruik deze commando's alleen als de spatiering inderdaad te wensen overlaat, anders doen ze meer kwaad dan goed.

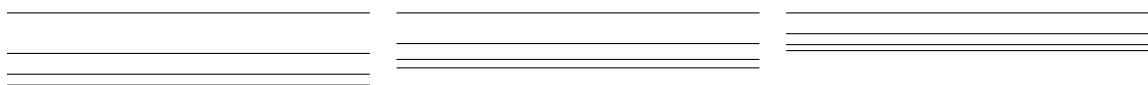
Men kan witruimte tussenvoegen met `\vskip`, maar we raden iedereen aan waar nodig het onderstaande commando te gebruiken.

```
\blanko[.,.,.,.,.]
```

```
...      n*klein n*middel n*groot geenwit terug wit blokkeer forceer reset regel halverege1
         formule vast flexibel
```

We kunnen een maat meegeven, maar ook een van de trefwoorden `klein`, `middel` of `groot`. Hierbij is een grote sprong twee maal een middelgrote sprong en viermaal een kleine sprong. Eventueel kan de maat achterwege blijven (`\blanko`), in dat geval wordt de standaard ingestelde sprong gemaakt. Veel commando's bieden de mogelijkheid de omringende witruimte in te stellen, dit verdient de voorkeur boven het (steeds weer) ter plekke afdwingen van witruimte.

Er zit normaal gesproken wat rek in de sprong. Dit maakt het \TeX mogelijk een bladzijde zo goed mogelijk te vullen. Deze rek kan overigens ongedaan worden gemaakt. In het volgende voorbeelden zien we wat er gebeurt als we rek toevoegen aan witruimte. Van boven naar onder zien we een `\blanko` met de waarde `groot`, `middel` and `klein`. De linker en rechter voorbeelden tonen het bereik van de rek. Rechts zien we dat de rek kan uitmonden in krimp.



maximale rek

geen rek

minimale rek

Een vorige sprong kan ongedaan worden gemaakt door terug op te geven en een volgende `blanko` kan worden geblokkeerd met `blokkeer`. Met `reset` wordt een `blokkeer` ongedaan gemaakt.

Het commando `\blanko` is een van de geavanceerdere commando's. Zo is de volgende aanroep toegestaan:

```
\blanko[2*groot,middel,blokkeer]
```

Omdat een sprong van `middel` gelijk is aan een halve sprong `groot`, resulteert dit in het tussenvoegen van precies 2.5 grote sprongen. De vorige sprongen worden automatisch ongedaan gemaakt en een eventuele volgende sprong wordt onderdrukt door `disable`.

Een vorige sprong kan worden gehandhaafd door `forceer`. Als men bovenaan een pagina witruimte wil hebben, moet men in de regel `forceer` toevoegen.

De standaard sprongen kunnen worden ingesteld met het commando:

```
\stelblankoin[...]
...   normaal standaard regel maat groot middel klein vast flexibel
```

Een voorbeeld van zo'n definitie is:

```
\stelblankoin[groot]
```

De sprongen worden automatisch aangepast aan de grootte van het korps en zijn zoals gezegd flexibel. Het veranderen van de standaard-instellingen is dan ook af te raden. Als dit commando wordt gegeven zonder argument, dan past de sprong zich aan het formaat van de actuele letter aan!

De flexibiliteit kan worden opgeheven met het trefwoord `vast` en weer worden ingesteld met `flexibel`. Het is bijvoorbeeld aan te bevelen binnen kolommen de instelling `[vast,regel]` te gebruiken. Daarbuiten kan vaak beter tegenhanger van deze instelling worden gebruikt: `[flexibel,standaard]`.

Deze tekst is gezet in een korps van 10pt en wordt als geheel iets omlaag geschaald. De instellingen bij deze tekst zijn weergegeven in tabel 4.1. We zien dat er enige rek en krimp in de witruimte zit. Rek en krimp geeft \TeX de mogelijkheid een bladzijde beter uit te lijnen. Standaard is de grootste sprong gelijk aan 75% van de regelafstand, en de rek maximaal 25%.

instelling	waarde
klein	2.78249pt plus 0.92749pt minus 0.92749pt
middel	5.56499pt plus 1.855pt minus 1.855pt
groot	11.12997pt plus 3.70999pt minus 3.70999pt
regel	14.83998pt

Tabel 4.1 De witruimtes bij een 10pt korps.

Zoals reeds in paragraaf 4.2 is beschreven, kan de afstand tussen alinea's worden ingesteld met het commando `\stelwitruimtein`. Standaard wordt geen witruimte opgenomen tussen alinea's. De instellingen van witruimte en blanco sprongen zijn (standaard) aan elkaar gerelateerd.

In plaats van directe instellingen, kan men ook indirecte gebruiken. Dit heeft als voordeel dat een layout eenvoudiger is aan te passen. We gebruiken in dat geval:

```
\definieerblanko[.1.][.2.]
.1.   naam
.2.   zie p 62: \stelblankoin
```

Als we bijvoorbeeld zeggen:

```
\definieerblanko[rondomtypen][middel]
```

dan is vervolgens `rondomtypen` gelijk aan `middel`, dat te gebruiken is in bijvoorbeeld:

```
\steltypenin
  [voor={\blanko[rondomtypen]}],
  [na={\blanko[rondomtypen]}]
```

Willen we toch meer witruimte, dan hoeven we slechts één instelling aan te passen:

```
\definieerblanko[rondomtypen][groot]
```

De witruimte tussen twee regels kan worden onderdrukt met het commando:

```
\opelkaar
```

Witruimte tussen meerdere regels wordt onderdrukt met:

```
\startopelkaar[...] ... \stopopelkaar
...   blanko
```

De witruimte boven en onder de ‘op elkaar geplaatste’ tekst wordt automatisch gecorrigeerd.

De tegenhanger van dit commando is:

```
\startvanelkaar ... \stopvanelkaar
```

Omlaag springen over een bepaalde afstand doet men met:

```
\omlaag[...]
...   maat
```

Gebruik dus in de lopende tekst als het even kan geen `\vskip` of andere harde sprongen omlaag. Dit kan namelijk vervelend interfereren met de (voor de `CONTEX`T gebruiker grotendeels) verborgen mechanismes.

Soms is `TEX` niet in staat zelfstandig vast te stellen of de witruimte in orde is. In dergelijke situaties kan men het volgende commando tussenvoegen op de problematische plek.

```
\corrigeerwitruimte{...}
```

Normaal gesproken is dit commando zelden of nooit nodig, hoewel het bij het schrijven van macros enige zekerheid kan brengen over de correcte afhandeling van witruimte. Gebruik dit commando alleen als het echt niet anders kan.

4.4 Spatiëring

Achter de punt wordt standaard een spatie gezet. In andere landen is het gebruikelijk de ruimte achter de punt te benutten voor het uitlijnen van de regel. Bij smalle kolommen levert dit vaak een fraaiere resultaat. Deze vorm van spatiëring is taalgebonden, en wordt dan ook automatisch aangepast. Zonodig kan (lokaal) een ander gedrag worden ingesteld met:

```
\stelspatieringin[...]
...      ruim opelkaar
```

Niet zelden komen we in een tekst combinaties van woorden en/of getallen tegen die we liever niet gescheiden over meerdere regels zien, bijvoorbeeld: nummer 12. Dergelijke woorden kunnen we verbinden door middel van een harde spatie: nummer~12. Op deze plaats wordt dus nooit afgebroken. Een zachte spatie kan worden opgeroepen met:

```
\spatie
```

Ongewenste spaties kunnen worden onderdrukt met:

```
\geenspatie
```

Bij het uitlijnen van getallen kan men werken met vaste spaties, ter breedte van een cijfer. Het activeren van vaste spaties doen we met:

```
\vastespaties
```

Na dit commando genereert ~ een spatie ter breedte van een cijfer.

4.5 Struts

Een strut is een onzichtbaar blokje zonder breedte, maar met de maximale hoogte en diepte van een karakter, en dan meestal een regel. Als men deze maximale maten wil forceren, bijvoorbeeld als men in eigen commando's boxen gebruikt, dan kan men gebruik maken van het commando `\strut`:

```
\hbox{\strut test}
```

Als we in het bovenstaande voorbeeld de strut weglaten, dan heeft de box geen diepte. De letters in het woordje test komen immers niet onder de baseline. Vergelijk `test` (met strut) eens met `test`.

Veel commando's plaatsen automatisch struts. Als men om een of andere reden geen strut wil, dan kan men proberen de strut te onderdrukken met `\setnostrut`. Zorg er echter voor dat dit commando lokaal werkt en gebruik dus eerder te veel dan te weinig accolades. Een strut kan worden 'geset' met `\setstrut`.

De (door CONTEXT geplaatste) struts kunnen zichtbaar worden gemaakt met het commando:

```
\toonstruts
```

4.6 Margeteksten

In de marge kunnen teksten worden opgenomen. Het commando luidt:

```
\inmarge[.1.][ref]{.2.}
.1.    + - laag
.2.    tekst
```

Het is mogelijk binnen een margetekst een overgang naar een volgende regel af te dwingen met: \\. Een voorbeeld van een margetekst is:

```
\inmarge{marginale\\invloed van\\reclame}Het zou mooi zijn geweest als de
afname van de hoeveelheid waspoeder die nodig is om een was vlekkeloos
schoon te krijgen ook had geresulteerd in een afname van de tijd die nodig
is om dergelijke produkten aan de man te brengen.
```

ofwel:

Het zou mooi zijn geweest als de afname van de hoeveelheid waspoeder die nodig is om een was vlekkeloos schoon te krijgen ook had geresulteerd in een afname van de tijd die nodig is om dergelijke produkten aan de man te brengen.

Wanneer het commando midden in een paragraaf wordt gebruikt, dan zal de tekst op dezelfde regel in de margekomt. Het commando `\inmarge` plaatst de tekst in de linker- of rechtermarge, afhankelijk van het al of niet dubbelzijdig zetten. Een plaats kan echter ook worden afgedwongen met:

```
\inlinker[.1.][ref]{.2.}
.1.    + - laag
.2.    tekst
```

```
\inrechter[.1.][ref]{.2.}
.1.    + - laag
.2.    tekst
```

Bovendien is er:

```
\inanderemarge[.1.][ref]{.2.}
.1.    + - laag
.2.    tekst
```

Hieronder zijn enkele voorbeelden opgenomen:

```
\startregels
\inlinker{eerste}De eerste regel is wat kort,
\inrechter{tweede}de tweede nog wat korter,
\inmarge{derde}de derde echter heeft de lengte van een regel Gorter
en is dan ook hoe kan het anders echt zo droog als gort.
\stopregels
```

Dit wordt:

eerste De eerste regel is wat kort,
de tweede nog wat korter,

tweede

derde de derde echter heeft de lengte van een regel Gorter
en is dan ook hoe kan het anders echt zo droog als gort.

123 Het mechanisme achter het plaatsen van margeteksten is, als men de achterliggende macro's zou bestuderen, vrij gecompliceerd. Er zijn wat truukjes voor nodig om (meer-regelige) margeteksten netjes uit te lijnen met de lopende tekst. Het getal 123 naast deze alinea lijkt echter niet uit maar is wat lager geplaatst. Dit kan men bereiken door het trefwoord laag mee te geven:

```
\inmarge[laag]{\ssd 123}Het mechanisme achter het plaatsen ...
```

Het is mogelijk de aan `\inmarge` gekoppelde plaats af te dwingen met behulp van het commando:

```
\stelinmargein[...][...,...=...,...]
```

...	links rechts <i>getal</i>
plaats	links rechts <u>beide</u>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
voor	<i>commando</i>
na	<i>commando</i>
uitlijnen	<u>binnen</u> buiten links rechts midden normaal nee ja
regel	<i>getal</i>
afstand	<i>maat</i>
scheider	<i>tekst</i>
..=..	zie p 226: <code>\stelomlijndin</code>

een rand-verschijnsel Met uitlijnen geven we aan of we tegen de tekst of tegen de rand van het papier willen uitlijnen. Standaard wordt tegen de rand van de tekst uitgelijnd. Hier zien we de instelling midden in actie.

We kunnen ook automatisch aan de binnenkant of buitenkant van de bladspiegel uitlijnen. Bij een dubbelzijdige layout vindt, net als bij het plaatsen zelf, een optimalisatie plaats; het kan dus zijn dat een extra verwerkingsslag nodig is. Hieronder zijn de verschillende mogelijkheden weergegeven.

links Dit is links uitgelijnd

en dit in het midden. Vergeet niet dat

midden

rechts uitlijnen betekent een raffeelige rechtermarge.

rechts

Voor de volledigheid zijn er ja

ja

en nee.

nee

De buitenbeentjes binnen en

binnen

buiten passen zich aan dubbelzijdig zetten aan.

buiten

Men kan de linker- en rechtermarge apart instellen door als eerste argument [*links*] of [*rechts*] mee te geven.

zo kunnen we nog aardig wat kwijt in de marge

Met voor en na kunnen we de vorm van de margeteksten beïnvloeden. Zo wordt standaard dezelfde interlinie gebruikt als in de tekst. Als we een kleinere letter gebruiken, kunnen we desgewenst de interlinie aanpassen, bijvoorbeeld:

```
\stelinmargein
  [letter=\bfx\stelinterliniein]
```

Het afdwingen van een plaats kan bij overgangen naar een nieuwe bladzijde soms mis gaan. Dit komt omdat T_EX de beste plaats om af te breken pas na het verzamelen van een volle bladzijde tekst vaststelt, terwijl de margetekst dan al links of rechts is geplaatst. Omdat de juiste plaats wel wordt bijgehouden, staan de margewoorden bij een volgende verwerking van de tekst wel op de juiste plaats. Als tekst is toegevoegd, dan kan bij een eerstvolgende verwerkingsslag alles weer verkeerd staan. Wanhoop echter niet en verwerk de tekst nogmaals. Als men desondanks plaatsing in de andere marge wil afdwingen, dan kan dat met `\inmarge[+]`.

Het onderstaande commando is te vergelijken met commando's als `\paragraaf`. Voordat de opgegeven tekst in de marge wordt geplaatst, wordt eerst gekeken of de tekst nog wel op de bladzijde past. Zo niet, dan wordt overgegaan naar een nieuwe bladzijde.

```
\margetekst[.1.][ref]{.2.}
.1.    +- laag
.2.    tekst
```

Dit commando ondersteunt een ruime layout van de ASCII-file en het is dan ook mogelijk indexingangen op te nemen voordat met de tekst wordt begonnen:

```
\margetekst{tekst in de marge}
\index{margeteksten}
```

Na lang experimenteren is het dan toch gelukt om niet alleen de (optionele) lege regel weg te filteren, maar ook commando's als `\index`. Dergelijke commando's worden echter nog wel uitgevoerd!

Omdat er een nauwe relatie bestaat met de rest van de zetspiegel, wordt de breedte van de marge ingesteld met behulp van `\stetlayoutin` (zie paragraaf 3.3).

a's
toch
aardig Het commando `\margetekst` heeft nog een aardige verrassing in petto. Het is namelijk mogelijk meerdere teksten in de marge te plaatsen, elk met eigen kenmerken, zoals een lettertype, uitlijnen, en een offset, uitgedrukt in regels.

```
\margetekst{da's}
\margetekst{toch}
\margetekst{aardig}
```

We zien bij de instellingen het tweede, optionele, argument om de hoek komen. Het nummer wordt bepaald door de volgorde waarin de teksten worden gedefinieerd.

```
\stelinmargein[1][uitlijnen=rechts,regel=1,letter=schuin]
\stelinmargein[2][uitlijnen=midden,regel=2,letter=schuinvet]
\stelinmargein[3][uitlijnen=links,regel=3,letter=vet]
```

Dit betekent dat de tweede margetekst op de tweede regel wordt geplaatst in een schuin vet lettertype. Men kan een margetekst naar een ingestelde positie dwingen door een aanroep als:

```
\margetekst[2]{dit is de tweede tekst}
```

4.7 Verhogen en verlagen

Met de volgende drie commando's kunnen letters of woorden worden verhoogd en verlaagd:

```
\hoog{...}
... tekst
```

```
\laag{...}
... tekst
```

```
\laho[.1.]{.2.}{.3.}
.1. laag
.2. tekst
.3. tekst
```

Het volgende voorbeeld illustreert deze commando's:

Iemand kan weliswaar bij `\hoog{hoog}` en bij `\laag{laag}` beweren dat hij in de 17^e eeuw geboren is, bewijzen is wat anders.

Wat resulteert in:

Iemand kan weliswaar bij ^{hoog} en bij _{laag} beweren dat hij in de 17^e eeuw geboren is, bewijzen is wat anders.

Deze commando's passen zich zo goed en kwaad als dat gaat aan bij [^] en _{_} in de mathematische mode. In geval van grotere letters, zoals `\tfc`, leveren [^] en _{_} echter niet het gewenste resultaat. Bovendien moet men bij deze commando's expliciet naar een bepaald soort letter overgaan. Vergelijk maar eens:

```
test\hoog{test} test test$^{\rm test}$ test
{\bf test\hoog{test} test test$^{\bf test}$ test}
{\tfb test\hoog{test} test test$^{\tfb test}$ test}
```

Dit wordt:

```
testtest test testtest test
testtest test testtest test
testtest test testtest test
```

4.8 Kolommen

Het zetten in kolommen is geen eenvoudige zaak, althans voor de macro schrijver. Gelukkig heeft de gebruiker daar weinig mee van doen.

Tekst kan dus in kolommen worden gezet. De meeste commando's kunnen zonder problemen binnen kolommen worden gebruikt. De mogelijkheden van verplaatsbare blokken zijn echter wat beperkt. Dit is een direct gevolg van het feit dat \TeX standaard beperkte mogelijkheden heeft om in kolommen te zet-

ten. Voor de insiders: kolommen worden gezet door middel van een nauwe samenwerking tussen de primitieven: `\output` en `\vsplit`.

Het aantal kolommen is in principe onbeperkt, hoewel in de praktijk het geheugen van \TeX slechts ruimte biedt voor enkele tientallen kolom-

men naast elkaar, zal men eerder een 'tekort' aan papier hebben dan aan geheugen.

Het aantal kolommen en het al dan niet opnemen van een verticale lijn kan worden ingesteld met het commando:

```
\stelkolommenin[...=...]
```

<code>n</code>	<code>getal</code>
<code>nboven</code>	<code>getal</code>
<code>lijn</code>	aan <u>uit</u>
<code>hoogte</code>	<code>maat</code>
<code>tolerantie</code>	zeerstreng streng <u>soepe</u> l zeersoepe] rek
<code>afstand</code>	<code>maat</code>
<code>balanceren</code>	<u>ja</u> nee
<code>uitlijnen</code>	ja nee <u>tekst</u>
<code>blanko</code>	<u>vast</u> halveregel <u>regel</u> flexibel groot middel klein
<code>optie</code>	achtergrond
<code>richting</code>	links <u>rechts</u>
<code>..=..</code>	zie p 226: <code>\stelomlijndin</code>

Hierbij staat `n` voor het aantal kolommen. De in kolommen te zetten tekst wordt voorafgegaan en afgesloten met de commando's:

```
\startkolommen[.,.,.=.,.,.] ... \stopkolommen
..=.. zie p 69: \stelkolommenin
```

Achter dit commando mogen direkt (lokale) instellingen worden meegegeven. Er wordt naar een nieuwe kolom gesprongen met:

```
\kolom
```

De volgende tekst wordt in twee kolommen gezet. Daarbij wordt `zeersoepel` uitgelijnd.

```
\startkolommen[lijn=aan,n=2,tolerantie=zeersoepel]
Thus, I came to the conclusion that the designer of a new
system must not only be the implementer and first
.
.
\bf D.E. Knuth
\stopkolommen
```

Thus, I came to the conclusion that the designer of a new system must not only be the implementer and first large-scale user; the designer should also write the first user manual.

The seperation of any of these four components would have hurt \TeX significantly. If I had not participated fully in all these activities, literally hundreds of improvements would never have been made, because I would

never have thought of them or percieved why they were important.

But a system cannot be succesful if it is too strongly influenced by a single person. Once the initial design is complete and fairly robust, the real test begins as people with many different viewpoints undertake their own experiments.

D.E. Knuth

Dit voorbeeld laat (op pijnlijke wijze) zien dat de witruimte niet gelijk is aan de afstand tussen twee regels. Standaard is de witruimte in deze tekst groot, wat neer komt op $.75 \times \text{\reglafstand}$. Bovendien zit er een klein beetje rek in deze afstand, zodat uitlijnen, zeker bij ongelijke kolommen, gegarandeerd mislukt.

Om deze reden wordt standaard de witruimte gelijk gemaakt aan de regelafstand en wordt bovendien de rek eruit gehaald. Als men een kleiner lettertype wil, dan kan men dat instellen voordat gestart wordt met kolommen. Doet men dit binnen kolommen, dan zijn de resultaten vaak teleurstellend. De volgende variant toont ons het effect van witruimte die gelijk is aan de regelafstand:

Thus, I came to the conclusion that the designer of a new system must not only be the implementer and first large-scale user; the designer should also write the first user manual.

The seperation of any of these four components would have hurt \TeX significantly. If I had not participated fully in all these activities, literally hundreds of improvements would never have been made, because I would

never have thought of them or perceived why they were important.

But a system cannot be successful if it is too strongly influenced by a single person. Once the initial design is complete and fairly ro-

bust, the real test begins as people with many different viewpoints undertake their own experiments.

D.E. Knuth

Dit effect is bereikt door de onderstaande (default)instelling:

```
\stelkolommenin[blanko={vast,regel}]
```

In paragraaf 3.4 wordt ingegaan op het zetten op grid, wat wil zeggen dat de regels op een linker en rechter pagina, of wanneer we in kolommen zetten, met elkaar uitlijnen. Hoewel dit mechanisme niet helemaal waterdicht is — \TeX kent namelijk niet zoiets als een grid— zijn in kolommen heel aardige resultaten te bereiken.

\TeX is geen eenvoudig programma. Het probleem schuilt mede in het gegeven dat “weten dat alles kan” ook leidt tot “alles willen doen”. En dat kost leertijd, in sommige gevallen niet weinig ook. Hetzelfde geldt voor CONTEXT . De

echte doe-het-zelvers kunnen veel aanvullende informatie in de broncode vinden. Gebruik echter alleen die dingen, die ook bedoeld zijn om te gebruiken.

Dit stukje tekst toont dat men met kolommen wat grappen uit kan halen. Het omlijnen is een gevolg van de instelling:

```
\def\OmkaderKolom#1{\rule{edhbox}{\box#1}}
```

```
\stelkolommenin[commando=\OmkaderKolom]
```

Een wellicht meer zinvolle weergave is:

```
\def\OmkaderKolom#1%
  {\hbox to \hsize
   {\ifodd\currentcolumn\hss\unhbox#1\else\unhbox#1\hss\fi}}
```

Dit keer zien de kolommen er uit als:

\TeX is geen eenvoudig programma. Het probleem schuilt mede in het gegeven dat “weten dat alles kan” ook leidt tot “alles willen doen”. En dat kost leertijd, in sommige gevallen niet weinig ook. Hetzelfde geldt voor CONTEXT . De

echte doe-het-zelvers kunnen veel aanvullende informatie in de broncode vinden. Gebruik echter alleen die dingen, die ook bedoeld zijn om te gebruiken.

Naast de genoemde manipulatie kan ook een kolom als geheel worden gemanipuleerd. Een bijzonder geval is het aanbrengen van een achtergrond:

```
\stelvoetnotenin
  [plaats=kolommen,
   achtergrond=kleur,
   achtergrondkleur=wit]
```

```
\stellayoutin
  [grid=ja]
```

Dit keer krijgen we op grid gezet:

TeX is geen eenvoudig programma. Het probleem schuilt mede in het gegeven dat “weten dat alles kan” ook leidt tot “alles willen doen”. En dat kost leertijd, in sommige gevallen niet weinig ook. Hetzelfde geldt voor CONTEX. De echte doe-het-zelvers kunnen veel aanvullen-

de informatie in de broncode⁴ vinden. Gebruik echter alleen die dingen, die ook bedoeld zijn om te gebruiken.

⁴ De source.

4.9 Alinea's in kolommen

Het werken met kolommen verloopt nog niet altijd vlekkeloos maar de macro's worden regelmatig verbeterd. In aanvulling op het bovenstaande is daarom het volgende commando beschikbaar:

```
\definieeralineas[...][...,...=...,...]
...      naam
n        getal
lijn     aan uit
hoogte   passend maat
voor     commando
na       commando
binnen   commando
afstand  maat
tolerantie  zeerstreng streng soepe] zeersoepe] rek
uitlijnen links rechts midden
```

Met behulp van dit commando definieert men een kolom-layout die op naam oproepbaar is. Dit commando kan gebruikt worden voor tabellen in kolomvorm.

```
\startalinea ... \stopalinea
```

De layout is nader te specificeren met:

```
\stelalinea[in[.1.][.2.][...,...=...,...]]
.1.      naam
.2.      getal elk
letter   normaal vet schuin vetschuin type kap klein... commando
breedte  maat
hoogte   maat
uitlijnen links rechts midden breedte breedte
tolerantie  zeerstreng streng soepe] zeersoepe] rek
afstand  maat
voor     commando
na       commando
binnen   commando
commando commando
lijn     aan uit
```

De breedte van de niet-gespecificeerde kolommen wordt automatisch berekend. De afstand heeft betrekking op de ruimte vóór een kolom. Een volgende kolom wordt gestart met:

```
\alinea
```

Hieronder geven we een eenvoudig voorbeeld van het zetten in kolommen.

```
\definieeralineas[TweeKolommen][n=2]
\stelaalinasin[TweeKolommen][1][breedte=5cm]
\startTweeKolommen
  Het woord linksboven bevat relatief veel medeklinkers.
\TweeKolommen
  Hetzelfde geldt voor het woord rechtsboven, maar hier vinden
  we ook nog vrij veel medeklinkers achter elkaar.
\stopTweeKolommen
\startTweeKolommen
  In het woord linksonder wisselen klinkers en medeklinkers
  elkaar beter af.
\TweeKolommen
  Het woord rechtsonder biedt een aardige mix van de eerder
  genoemde kenmerken.
\stopTweeKolommen
```

Het commando `\TweeKolommen` scheidt de kolommen. Dit levert bij de standaardinstellingen het volgende op:

Het woord linksboven bevat relatief veel medeklinkers. Hetzelfde geldt voor het woord rechtsboven, maar hier vinden we ook nog vrij veel medeklinkers achter elkaar.

In het woord linksonder wisselen klinkers en medeklinkers elkaar beter af. Het woord rechtsonder biedt een aardige mix van de eerder genoemde kenmerken.

We hadden in dit geval in plaats van `\TweeKolommen` ook `\volgendeTweeKolommen` mogen opgeven, soms is deze variant wat leesbaarder. Als dat beter uitkomt kan ook de wat meer compacte maar minder robuuste, constructie worden gebruikt:

```
\TweeKolommen eerste tekst \ \ tweede tekst \ \
```

Aan boven en onder kunnen commando's worden toegekend. Deze worden voor en na de te zetten tekst uitgevoerd. Zo kan bijvoorbeeld een kolom naar onder worden gedrukt door `[boven=\vfill]`.

Dit is een aardig moment om een wat geavanceerder voorbeeld uit te werken. Net als in het vorige voorbeeld, gebruiken we ook hier hoofdletters om de commando's te onderscheiden van de standaard `CONTEXT` comando's. Stel dat we een aantal genummerde items moeten beoordelen op meerdere schalen. Het weergeven in kolommen ligt dan voor de hand. Hoewel

het kolommen-mechanisme kan worden gebruikt, is het beter in dergelijke situaties gebruik te maken van alineas. Deze worden namelijk netjes bij elkaar gehouden.

Als we een item op twee schalen willen beoordelen, hebben we drie kolommen nodig:

```
\definieeralineas [SamengesteldItem] [n=3,lijn=aan]
\stelalineasin [SamengesteldItem] [2] [breedte=3em]
\stelalineasin [SamengesteldItem] [3] [breedte=7em]
```

Het item zelf definiëren we met `\doornummeren` (zie paragraaf 10.3):

```
\doornummeren
[EenItem]
[plaats=links,tekst=,breedte=3em,voor=,na=]
```

De beoordelingsschaal stellen we samen met een opsomming (zie paragraaf 10.6), zodat een item er als volgt uitziet:

```
\startSamengesteldItem
\startEenItem
De cursist is in staat zelfstandig een plan van aanpak te
schrijven voor de bouw van een speelwaterzuivering.
\stopEenItem
\volgendeSamengesteldItem
\startopsomming[5,opelkaar]
\som ja \som nee
\stopopsomming
\volgendeSamengesteldItem
\startopsomming[5,opelkaar]
\som zelfstudie \som bijeenkomst \som opdracht
\stopopsomming
\stopSamengesteldItem
```

Of in gezette vorm:

<p>1 De cursist is in staat zelfstandig een plan van aanpak te schrijven voor de bouw van een speelwaterzuivering.</p>	<ul style="list-style-type: none"> ◦ ja ◦ nee 	<ul style="list-style-type: none"> ◦ zelfstudie ◦ bijeenkomst ◦ opdracht
---	---	---

Wanneer de beoordelingsschalen steeds hetzelfde zijn, kan een en ander wat ‘mooier’ door zelf een kleine macro te definiëren:

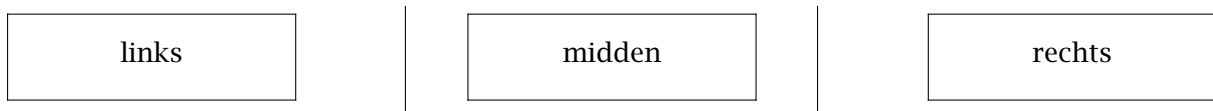
```
\def\EersteSchaal%
{\startopsomming[5,opelkaar]
\som ja \som nee
\stopopsomming}
\def\TweedeSchaal%
{\startopsomming[5,opelkaar]
```

```

\som zelfstudie \som bijeenkomst \som opdracht
\stopopsomming}
\startSamengesteldItem
\startEenItem
  De cursist is in staat zelfstandig een plan van aanpak te
  schrijven voor de bouw van een speelwaterzuivering. \par
\stopEenItem
\volgendeSamengesteldItem
\EersteSchaal
\volgendeSamengesteldItem
\TweedeSchaal
\stopSamengesteldItem
Het kan nog fraaier:
\def\StartItem%
  {\startSamengesteldItem
  \startEenItem}
\def\StopItem%
  {\stopEenItem
  \volgendeSamengesteldItem \EersteSchaal
  \volgendeSamengesteldItem \TweedeSchaal
  \stopSamengesteldItem}
\StartItem
  De cursist is in staat zelfstandig een plan van aanpak te
  schrijven voor de bouw van een speelwaterzuivering.
\StopItem

```

Een commando als hierboven besproken stelt ons soms voor verrassingen. Het is namelijk altijd weer de vraag hoe combinaties van constructies zich gedragen. Binnen dergelijke commando's kunnen namelijk `\vtop`'s, die uitlijnen op de baseline boven, `\vbox`'en, die juist onder uitlijnen, vervelend interfereren. We zagen reeds dat opsommingen en definities goed gaan. Het onderstaande voorbeeld toont aan dat het ook goed gaat met kaderteksten.



4.10 Tabulatie

Hoewel we in een later hoofdstuk dieper ingaan op tabellen, staan we hier alvast kort stil bij tabuleren. We spreken van tabuleren als een tabel deel uitmaakt van de lopende tekst. Hieronder is een voorbeeld opgenomen van een eenvoudige tabulatie.

```
\starttabulatie[|p|]
```

```
\NC vraagje \NC Het is soms aardig te mogen concluderen dat schrijvers
een zelfde thema, langs dezelfde lijnen, op een eigen karakteristieke
manier boeiend weten uit te werken. Dat geldt ook voor drie van de vier
auteurs. Welke auteur (welk boek) hoort in het rijtje niet thuis? \NC \NR
\stoptabulatie
```

```
\starttabulatie[|1|1|1|]
```

```
\NC A. \NC This Perfect Day           \NC Ira Levin           \NC \NR
\NC B. \NC Opstaan op Zaterdag        \NC Jan Gerhart Toonder \NC \NR
\NC C. \NC Tot waar zal ik je brengen \NC Anton Koolhaas      \NC \NR
\NC D. \NC The City And The Stars     \NC Arthur Clarke       \NC \NR
\stoptabulatie
```

Dit wordt:

vraagje Het is soms aardig te mogen concluderen dat schrijvers een zelfde thema, langs dezelfde lijnen, op een eigen karakteristieke manier boeiend weten uit te werken. Dat geldt ook voor drie van de vier auteurs. Welke auteur (welk boek) hoort in het rijtje niet thuis?

- A. This Perfect Day Ira Levin
- B. Opstaan op Zaterdag Jan Gerhart Toonder
- C. Tot waar zal ik je brengen Anton Koolhaas
- D. The City And The Stars Arthur Clarke

Met het commando `\NC` gaan we naar de volgende kolom, en met `\NR` naar de volgende rij. De definities `[|p|]` en `[|1|1|1|]` noemen we een template. De mogelijke instellingen komen grotendeels overeen met die van `\starttabel` (zie tabel ??).

Standaard wordt uitgegaan van twee kolommen, waarbij de tweede een volledige alinea kan zijn: `[|p|]`. Die alinea neemt de rest van de beschikbare breedte in beslag.

```
\starttabulatie
```

```
\NC d: \NC avond, afond, avend, afend \NC \NR
\NC t: \NC avont, afont, avent, afent \NC \NR
\stoptabulatie
```

Dit uit “Spellingsverandering van zin naar onzin” van G.C. Molewijk (1992) ontleende voorbeeld ziet er gezet dus uit als:

```
d: avond, afond, avend, afend
t: avont, afont, avent, afent
```

4.11 Uitlijnen

Het horizontaal en vertikaal uitlijnen van de tekst wordt ingesteld met het commando:

```
\steluitlijnenin[...]
```

```
...    breedte links rechts midden binnen buiten breed ruim hoogte onder regel reset hangend
        niethangend afgebroken nietafgebroken
```

Hierbij hebben *links*, *midden* en *rechts*, *binnen* en *buiten* betrekking op horizontaal uitlijnen en *onder*, *hoogte* en *regel* op vertikaal uitlijnen.

Het trefwoord *rechts* resulteert in een ruffelige rechterkantlijn. Het trefwoord *ruim* is te combineren met *links*, *midden* en *rechts*, en levert dan wat grover uitlijnen op.

De optie *regel* zorgt ervoor dat *op* de onderkant wordt uitgelijnd, terwijl bij *hoogte* de staart (diepte) *onder* de onderkant komt. De verschillende effecten kunnen het best met wat eenvoudige testen worden uitgetest.

Tijdens het doorrekenen van een alinea, zal \TeX verschillende alternatieven proberen om vervolgens de beste te kiezen op basis van toegekende strafpunten. Normaal is \TeX vrij streng, maar we kunnen \TeX vertellen wat toleranter te zijn. Dit betekent dat \TeX , in plaats van problemen onopgelost te laten —i.e. woorden die niet passen in de marge te laten steken— er meer rek wordt toegestaan en de strafpunten wat anders worden toegekend.

Het is mogelijk individuele regels uit te lijnen met:

```
\regellefts{...}
```

```
...    tekst
```

```
\regelmiddle{...}
```

```
...    tekst
```

```
\regelright{...}
```

```
...    tekst
```

Uitlijnen over meerdere regels kan met het commando:

```
\startuitlijnen ... \stopuitlijnen[...]
```

```
..=.. zie p 78: \steluitlijnenin
```

De onderstaande tekst toont enkele voorbeelden van horizontaal uitlijnen.

In Engeland blijft men hardnekkig de voorkeur geven aan links rijden.

Dit is eigenlijk een vorm van conservatisme,
of in onvervalst nederlands: rechts denken.

Rijden op het midden van de weg leidt
echter hoe dan ook tot brokken.

Dit hebben we bereikt met:

```
\regellinks{In Engeland blijft men hardnekkig de}
\regellinks{voorkeur geven aan links rijden.}
\blanko[middel]
\regelrechts{Dit is eigenlijk een vorm van conservatisme,}
\regelrechts{of in onvervalst nederlands: rechts denken.}
\blanko[middel]
\regelmidden{Rijden op het midden van de weg leidt}
\regelmidden{echter hoe dan ook tot brokken.}
```

De laatste woorden van een alinea kunnen rechts worden geplaatst met het commando `\woordrechts`, **dus met:**

```
\woordrechts{...}
... tekst
```

De bij het rechts uitlijnen gehanteerde criteria kunnen worden ingesteld met:

```
\steltolerantiein[...,...,...]
... horizontaal vertikaal rek spatie zeerstreng streng soepel zeersoepel
```

Standaard zijn `[horizontaal,zeerstreng]` en `[vertikaal,streng]` ingesteld. Als men echt iets lelijks wil zien, dan moet men de \TeX -variabele `\pretolerance` eens de waarde 10.000 geven. Je moet er maar van houden. Een beter alternatief, echter alleen te gebruiken in uiterste nood, is het instellen van de tolerantie op rek. Dit levert als redmiddel uitgerekte spaties op.

Het is, bij dubbelzijdig zetten, ook mogelijk het uitlijnen te koppelen aan de plaats van de bladzijde (links of rechts). Het resultaat van de commando's:

```
\startuitlijnen[binnen]
\citaat {Ik ben binnen} is zo'n uitdrukking waar geen tegenpool van
bestaat. Niemand zou begrijpen wat je bedoelt met \citaat {Ik ben buiten},
zeker niet als je achter de open haard in de woonkamer zit.
\stopuitlijnen

\startuitlijnen[buiten]
\citaat {Iets van buiten kennen} is er ook zo een. Het zou trouwens nog
niet zo eenvoudig zijn om te toetsen of iemand \citaat {iets van binnen}
ook inderdaad kent.
\stopuitlijnen
```

hangt dus af van de plaats van de bladzijde. Hier leiden de commando's tot:

“Ik ben binnen” is zo’n uitdrukking waar geen tegenpool van bestaat. Niemand zou begrijpen wat je bedoelt met “Ik ben buiten”, zeker niet als je achter de open haard in de woonkamer zit.

“Iets van buiten kennen” is er ook zo een. Het zou trouwens nog niet zo eenvoudig zijn om te toetsen of iemand “iets van binnen” ook inderdaad kent.

4.12 Regelovergangen

Een overgang naar een nieuwe regel kan worden afgedwongen met:⁵

```
\crlf
```

Het is mogelijk de indeling in regels zoals die in de ruwe tekst wordt gehanteerd af te dwingen. Er wordt in dit geval standaard *niet* ingesprongen. De regels worden gezet tussen de twee commando’s:

```
\startregels ... \stopregels
```

Rond regels kan een en ander worden ingesteld met:

```
\stelregelsin[...]=...]
```

voor	<i>commando</i>
na	<i>commando</i>
tussen	<i>commando</i>
inspringen	ja <u>nee</u> even oneven

Als we bijvoorbeeld `inspringen=oneven` instellen, dan krijgen we, mits natuurlijk inspringen is ingesteld:

```
Ziezo dacht Frans, dat fix ik even
  TeX geeft mij toch veel meer kans
Dan Word mij ooit zal geven
```

In de file was dit:

```
\stelinspringenin[middel]
\stelregelsin[inspringen=even]
\startregels
Ziezo dacht Frans, dat fix ik even
\TeX\ geeft mij toch veel meer kans
Dan Word mij ooit zal geven
\stopregels
```

Het is mogelijk regels te nummeren met behulp van het commando:

⁵ In titels en margeteksten is `\` beschikbaar om een regelovergang af te dwingen.


```
\startregelnummeren[...] ... \stopregelnummeren
... verder
```

In plaats van een conversie kan ook een eigen karakter worden opgegeven, bijvoorbeeld een — (---). In dat geval wordt dit karakter voor de regel geplaatst.

In hoofdstuk 9 zullen we nader ingaan op het verwijzen naar regelnummers. De parameters `prefix` en `refereren` beïnvloeden dat proces.

Hieronder is eerst een voorbeeld gegeven van genummerde regels:

```
\startregelnummeren
Het illegaal copi\''eren van geleende \kap{cd}'s en platen mag best
strafbaar worden gesteld. Stel dan echter tegelijkertijd ook het
illegaal jatten van thema's, teksten en harmonisaties strafbaar.
\stopregelnummeren
```

wordt:

- 1 Het illegaal kopiëren van geleende CD's en platen mag best strafbaar worden gesteld. Stel dan
- 2 echter tegelijkertijd ook het illegaal jatten van thema's, teksten en harmonisaties strafbaar.

We kunnen de wijze van nummeren beïnvloeden met het commando:

```
\stelregelnummerenin[...==...]
```

conversie	<u>cijfers</u> letters Letters romeins Romeins tekst
start	<u>getal</u>
stap	<u>getal</u>
breedte	<u>maat</u>
plaats	intekst <u>inmarge</u>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
prefix	<u>tekst</u>
refereren	<u>aan</u> uit

In het onderstaande voorbeeld handhaven we de regels en gebruiken we achtereenvolgens de instellingen:

```
\stelregelnummerenin[conversie=cijfers,stap=2,plaats=intekst]
```

en:

```
\stelregelnummerenin[conversie=letters,stap=1,plaats=intekst]
```

- | | | |
|------------------------------|---|----------------------------------|
| minimale variatie | a | maximale inspiratie |
| 2 kenmerkt soms zijn spel | b | kenmerkt heel zijn leven |
| geleidelijk verandert ritme | c | immers in die baan van hem |
| 4 je verliest de tel | d | valt heel wat te beleven |
| klank verandert, escaleert, | e | veel op pad, veel vreemde mensen |
| 6 keert terug naar het begin | f | soms apart en eigenaardig |
| goed doordachte intuïtie | g | 't wereldje van de muziek |
| 8 geeft muziek zijn zin | h | is toch wel wat merkwaardig |

Het is mogelijk regels te markeren en zo te verwijzen naar regelnummers. Dit komt in hoofdstuk 9 aan de orde.

4.13 Pagineren

De overgang naar een nieuwe pagina kan worden afgedwongen, geblokkeerd of gestimuleerd met:

```
\pagina[.,.,.,.,.]
...   ja opmaak nee voorkeur grotevoorkeur links rechts blokkeer laatste viertal even oneven
      blanco leeg reset
```

De mogelijke instellingen worden in tabel 4.2 uitgelegd. Als geen instelling wordt meegegeven (`\pagina`), wordt een overgang geforceerd.

instelling	gevolg
ja	forceer een pagina-overgang
opmaak	idem, zonder uitvullen
nee	bij voorkeur geen pagina-overgang
voorkeur	bij voorkeur pagina-overgang hier
grotevoorkeur	bij grote voorkeur pagina-overgang hier
links	ga naar een linker pagina
rechts	ga naar een rechter pagina
blokkeer	volgende commando's hebben geen effect
laatste	aanvullen tot een even aantal pagina's
viertal	aanvullen tot een viertal pagina's
even	doorgaan tot de volgende even pagina
oneven	doorgaan tot de volgende oneven pagina
blanco	geen ornamenten op de pagina plaatsen
leeg	voeg een lege pagina tussen
reset	volgende commando's hebben weer effect

Tabel 4.2 Instellingen bij `\pagina`.

De instellingen `laatste` en `viertal` kan men gebruiken bij dubbelzijdig (verkleind) afdrukken. De eerste instelling vult bij dubbelzijdig zetten aan tot een even aantal, de tweede instelling vult aan tot een viertal. Als een *nummer* wordt meegegeven, dan wordt naar deze pagina gegaan:

```
\pagina[25]
```

Men kan ook een relatief nummer opgeven, bijvoorbeeld `[+4]`. Vaak is dit wat veiliger, omdat we niet op voorhand weten op welke bladzijde we ons bevinden.

Bij het genereren van lege bladzijden moeten we rekening houden met dubbelzijdig zetten, bijvoorbeeld:

```
\pagina[rechts,leeg,rechts]
```

4.14 Paginanummers

Het paginanummer kan overal in de tekst worden ingesteld met het commando:

```
\stelpaginanummerin[...=...]
```

```
nummer   getal
status   start stop handhaaf
```

De plaats van het paginanummer wordt gedefinieerd met:

```
\stelnummeringin[...=...]
```

```
variant      enkelzijdig dubbelzijdig
plaats       hoofd voet links rechts midden marge kantlijn inlinker inrechter
conversie    cijfers letters Letters romeins Romeins
letter       normaal vet schuin vetschuin type kap klein... commando
links        tekst
rechts       tekst
wijze        pertekst persectie perdee]
tekst        tekst
nummerscheider tekst
tekstscheider tekst
sectienummer ja nee
scheider     tekst
strut        ja nee
status       start stop
commando     \commando#1
```

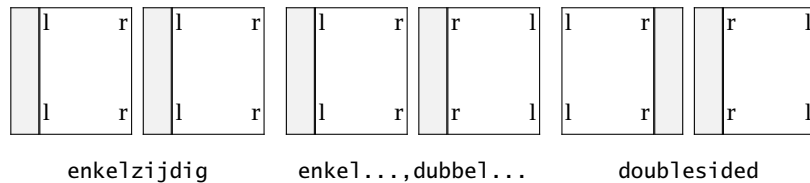
De plaats van het nummer hangt af van de variant. Met `conversie` geven we aan hoe we een nummer willen weergeven. Met `plaats` geeft men aan of het nummer boven- dan wel onderaan de bladzijde, links, rechts of in de marge komt te staan. Er mogen combinaties worden opgegeven, bijvoorbeeld:

```
\stelnummeringin[plaats={hoofd,inmarge}]
```

variant=enkelzijdig	variant=dubbelzijdig
links, rechts	kantlijn
midden	midden
marge	marge

Tabel 4.3 Instellingen bij `\stelnummeringin`.

Voor de liefhebbers is ook een hybride variant `{enkelzijdig,dubbelzijdig}` beschikbaar. Deze variant komt overeen met de dubbelzijdige en spiegelt dus de hoofd- en voetregels. Het rugwit blijft echter ongespiegeld (zie figuur 4.1).



Figuur 4.1 De drie manieren om te spiegelen.

Met `\stelnummeringin` kan men met `links` en `rechts` aangeven wat er voor en na het nummer komt. De instelling:

```
\stelnummeringin[conversie=romeins,links={--~},rechts={~--}]
```

leidt bijvoorbeeld tot: - viii -. Met `letter` wordt een lettertype ingesteld en met `status` regelt men het plaatsen: aan of uit.

Men kan een nummer van toeters en bellen voorzien als men aan `commando` een `commando` toekent. Dit `commando`, dat een argument heeft, wordt iedere keer aangeroepen als het nummer wordt geplaatst. We kunnen bijvoorbeeld een nummer omlijnd weergeven:

```
\stelnummeringin[commando=\inlijnd]
```

of gedeeltelijk omlijnd door:

```
\def\mijnpaginanummer#1%
  {\inlijnd[kader=uit,linkerkader=aan,rechterkader=aan]{#1}}
```

```
\stelnummeringin[commando=\mijnpaginanummer]
```

We gebruiken in dit geval `\inlijnd` in plaats van `\omlijnd`, omdat we het nummer op dezelfde hoogte willen hebben als de hoofd- of voetteksten.

Met `tekstscheider` stelt men in wat er tussen een deelnummer en een paginanummer wordt geplaatst, standaard een -. Als het nummer tegen de kantlijn wordt geplaatst, wordt `nummerscheider` tussen het nummer en de voetekst geplaatst. Standaard is dit een witruimte van 1em.

Er kan, wat bij interactieve teksten soms handig is, met subnummers worden gewerkt. Als bijvoorbeeld iedere paragraaf op een aparte bladzijde staat, kan een voetekst worden ingesteld als:

```
\stelsubpaginanummerin
  [wijze=perparagraaf]
\stelvoettekstenin
  [scherm {\subpaginanummer} van {\aantalsubpaginas}] []
```

De instellingen vinden dus plaats met:

```
\stelsubpaginanummerin[.,.,.=.,.,.]
```

```
wijze      pertekst persectie perdee]
status     start stop geen
```

en de waarden zijn oproepbaar met `\subpaginanummer` en `\aantalsubpaginas`. Deze getallen zijn alleen (betrouwbaar) op te roepen in de hoofd- en voetregels. In geval van interactieve teksten kan een wat meer abstracte wijze van weergeven worden gebruikt:

```
\stelvoettekstenin[][\interactieebalk[variant=d]]
```

Er kan in dat geval naar vorige of volgende subpagina's worden gesprongen. Bij grote aantallen subpagina's kan men beter variant `e` gebruiken. Hoewel het achterliggende mechanisme verschilt, zijn op analoge wijze de paginanummers beschikbaar in `\paginanummer` en `\totaalaantalpaginas`. Men kan subpaginanummers op 0 zetten door in plaats van de instellingen `[reset]` mee te geven.

4.15 Hoofd- en voetteksten

Hoofd- en voetregels kunnen worden ingesteld met de volgende commando's:

```
\stelhoofdttekstenin[.1.][.2.][.3.]
```

```
.1.   tekst marge rand
.2.   tekst sectie datum markering paginanummer
.3.   tekst sectie datum markering paginanummer
```

```
\stelvoettekstenin[.1.][.2.][.3.]
```

```
.1.   tekst marge rand
.2.   tekst sectie datum markering paginanummer
.3.   tekst sectie datum markering paginanummer
```

Hier zijn nogal wat argumenten optioneel. Als de plaatsaanduiding wordt weggelaten, dan wordt uitgegaan van de tekst boven de lopende tekst (`tekst`). De rand bevindt zich naast de marge en wordt eigenlijk alleen gebruikt bij interactieve documenten.

Zoals uit het trefwoord al blijkt, genereert `datum` een datum en `paginanummer` het nummer van de bladzijde. Omdat koppen standaard een `marker`ing hebben, kunnen ook de namen van koppen worden opgegeven: `deel`, `hoofdstuk`, `paragraaf` enz. Ook kan men om een sectienummer vragen: `hoofdstuknummer` enz.

De status kan alleen worden ingesteld voor de hele hoofdregel, zoals:

```
\stelhoofdin[status=hoog]
```

Zij die meer variatie in de hoofd- en voetteksten wensen, kunnen vier in plaats van twee argumenten meegeven. Dit heeft alleen zin bij dubbelzijdig of gecombineerd dubbel- en enkelzijdig zetten.

```
\stelvoettekstenin
  [even links][even rechts]
  [oneven links][oneven rechts]
```

Er zijn dus verschillende combinaties van argumenten mogelijk:

```
\stelhoofdtekstenin
\stelhoofdtekstenin[midden]
\stelhoofdtekstenin[linker tekst][rechter tekst]
\stelhoofdtekstenin[linker tekst][rechter tekst][linker .][rechter .]
\stelhoofdtekstenin[plaats][linker tekst][rechter tekst]
\stelhoofdtekstenin[plaats][linker .][rechter .][linker .][rechter .]
```

In plaats van een tekst kunnen we ook een trefwoord meegeven, zoals hoofdstuk, datum of paginanummer. Wanneer we een paginanummer op deze manier plaatsen, moeten tegelijk de volgende commando worden gegeven:

```
\stelnummeringin[plaats=]
```

Als we helemaal geen instellingen meegeven, dan worden de huidige instellingen ongedaan gemaakt. Problemen krijgt men als men [] toepast. Deze moeten worden omringd door accolades:

```
\stelvoettekstenin[hoofdstuk][{\huidigedatum[maand,jaar]}]
```

Als in de hoofd- en/of voetregels de naam van een hoofdstuk, paragraaf enz. wordt geplaatst, dan kunnen lange titels problemen opleveren.

Het zetten van hoofd- en voetregels kan worden beïnvloed met de commando's

```
\stelhoofdin[...][...,...=...,...]
```

...	tekst marge rand
status	<u>normaal</u> stop start leeg hoog geen geenmarkering naam
strut	<u>ja</u> nee
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
linkerletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
rechterletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
linkerbreedte	maat
rechterbreedte	maat
voor	<i>commando</i>
na	<i>commando</i>

en

```
\stelvoetin[...][...,...=...,...]
```

...	zie p 86: \stelhoofd
..=..	zie p 86: \stelhoofd

Evenals bij `\stel...tekstenin` is het eerste argument optioneel. Hierbij hebben `status`, `voor` en `na` betrekking op alle teksten, dus zowel de lopende tekst als de marges en randen.

Als . . .breedte wordt ingesteld, dan wordt indien nodig de in het hoofd of de voet weer te geven tekst ingekort. Hierbij is het eerste argument niet optioneel. De instelling van de `strut` is van belang als men andere dingen dan tekst wil plaatsen. Zet men de `strut` op `nee`, dan wordt niet langer gecorrigeerd voor diepte in de regel. Het commando `\toonstruts` kan in dat kader zeer verhelderend werken.

De instellingen bij `status` zijn in tabel 4.4 weergegeven. Er wordt in alle gevallen wel doorgenummerd, omdat het nummeren van bladzijden los staat van het plaatsen van het nummer.

instelling	gevolg
<code>normaal</code>	zichtbaar
<code>geen</code>	onzichtbaar, geen witruimte
<code>leeg</code>	één pagina onzichtbaar, wel witruimte
<code>hoog</code>	één pagina onzichtbaar, geen witruimte
<code>start</code>	zichtbaar
<code>geenmarkering</code>	laat markeringen weg
<code>stop</code>	onzichtbaar, wel witruimte

Tabel 4.4 Instellingen bij `\stelhoofdin` en `\stelvoetin`.

Als instellingen tussen `\start` en `\stop` plaatsvinden, dan hebben ze een lokaal karakter. Dat wil zeggen dat ze na `\stop` vergeten zijn. Bij de beide bovenstaande commando's kan dat betekenen dat als de bladzijde nog niet vol is, er hoofd- en voetregels verschijnen terwijl men denkt deze te hebben uitgezet. De oplossing hiervoor is eenvoudig: dwing nog voor `\stop` een overgang af naar een nieuwe pagina door `\pagina` te geven.

Het zetten van hoofd- en voetregels kan gedurende een bladzijde worden uitgezet met het commando:

```
\geenhoofdenvoetregels
```

Naast hoofd- en voetregels hebben we ook nog boven- en onderregels. Deze kunnen op vergelijkbare wijze worden ingesteld:

```
\stelboventekstenin[.1.][.2.][.3.]
.1.   tekst marge rand
.2.   tekst sectie datum markering paginanummer
.3.   tekst sectie datum markering paginanummer
```

```
\stelteksttekstenin[.1.][.2.][.3.]
.1.   tekst marge rand
.2.   tekst sectie datum markering paginanummer
.3.   tekst sectie datum markering paginanummer
```

```
\stelondertekstenin[.1.][.2.][.3.]
```

- .1. `tekst` marge rand
- .2. `tekst sectie` datum `markering` paginanummer
- .3. `tekst sectie` datum `markering` paginanummer

```
\stelbovenin[...][...=...]
```

- ... zie p 86: \stelhoofdin
- ..=.. zie p 86: \stelhoofdin

```
\steltekstin[...][...=...]
```

- ... zie p 86: \stelhoofdin
- ..=.. zie p 86: \stelhoofdin

```
\stelonderin[...][...=...]
```

- ... zie p 86: \stelhoofdin
- ..=.. zie p 86: \stelhoofdin

```
\geenbovenenonderregels
```

Wanneer de hoogte van een gebied nul is, wat standaard het geval is bij het boven en ondergebied, dan wordt een tekstregel niet geplaatst.

Het is mogelijk bij het plaatsen van een kop, een hoofd- of voettekst gedurende een pagina anders te behandelen dan standaard is ingesteld. Stel dat de volgende instelling geldt:

```
\stelhoofdtekstenin[paginanummer]
\stelvoettekstenin[hoofdstuk] [paragraaf]
```

Op een bladzijde waar een hoofdstuk begint, kan dat er erg lelijk uitzien. We kunnen daarom bijvoorbeeld zeggen:

```
\stelkopin[hoofdstuk] [hoofd=leeg,voet=leeg]
```

Maar dan raken we het nummer kwijt. Een meer acceptabele oplossing maakt gebruik van de mogelijkheid hoofd- en voetteksten te definiëren, zoals:

```
\definieertekst[hoofdstuk] [voet] [paginanummer]
```

met:

```
\stelkopin[hoofdstuk] [hoofd=hoog,voet=hoofdstuk,pagina=rechts]
```

krijgen we het gewenste effect: het paginanummer komt in de voet te staan, en het hoofd verdwijnt helemaal. Dit soort commando's zijn onmisbaar als men een tekst wil vormgeven zonder dat in de tekst zelf op allerlei plaatsen instellingen plaatsvinden, bijvoorbeeld voor ieder hoofdstuk. Dit mechanisme werkt alleen als overgaan op een nieuwe pagina is geactiveerd.


```
\definieertekst[.1.][.2.][.3.][.4.][.5.]
.1.    naam
.2.    hoofd voet
.3.    tekst
.4.    tekst
.5.    tekst
```

4.16 Voetnoten

In een tekst kunnen voetnoten worden opgenomen. De voetnoot wordt daarbij in de tekst, direct achter het woord of direct achter de punt van de zin waaraan de noot gekoppeld is, geplaatst.

```
\voetnoot[ref]{...}
...    tekst
```

Een voetnootnummer of -symbool is op te roepen met:

```
\noot[ref]
```

Een voorbeeld van het gebruik van voetnoten volgt hieronder.

Waar de eerste werken van Steve Reich nog bedoeld zijn voor de echte `\citeer {liefhebber}` van experimentele minimal||music `\voetnoot {Een goede eigentijdse minimal kenmerkt zich niet zozeer door een minimaal gebruik van instrumenten, maar veeleer door een polyfoon verschuivende ritmiek.}`, zijn zijn latere werken, zoals `\citeer {The Desert Music}` voor volledig orkest geschreven composities, waarbij het orkest wordt aangevuld met een voor Reich karakteristieke ritmesectie `\voetnoot {Deze bestaat voornamelijk uit piano's, marimba's en xylofoons.}` en zang. Samen met John Adams `\voetnoot {Zijn \citeer {Fearful Symmetries} is een perfecte mix van klassiek, jazz, swing en pop.}` is Reich dan ook een van de toonaangevende componisten van deze tijd. Jammer dat ze alleen te zien `\voetnoot {Het aardige van muziek als \citeer {Drumming} of \citeer {Sextet} is de constatering dat \citaat {wat het oor hoort} anders en meer is dan \citaat {wat het oog ziet}.}` en horen zijn bij de matig met zendtijd bedeelde `\kap {VPRO}`. `\voetnoot{Dit is een publieke omroep.}` `\voetnoot {Soms ook bij andere omroepen, bijvoorbeeld op de achtergrond van aangrijpende documentaires doet muziek van Adams het vaak goed.}`

Ongewenste spaties worden weggewerkt. Ook wordt tussen twee voetnoten afdoende spatiering toegepast. Het resultaat is als volgt:

Waar de eerste werken van Steve Reich nog bedoeld zijn voor de echte ‘liefhebber’ van experimentele minimal-music⁶, zijn zijn latere werken, zoals ‘The Desert Music’ voor volledig orkest geschreven composities, waarbij het orkest wordt aangevuld met een voor Reich karakteristieke ritmesectie⁷ en zang. Samen met John Adams⁸ is Reich dan ook een van de toonaangevende componisten van deze tijd. Jammer dat ze alleen te zien⁹ en horen zijn bij de matig met zendtijd bedeelde VPRO.^{10 11}

De wijze waarop de voetnoot wordt weergegeven, kan worden ingesteld met het volgende commando, dat bij voorkeur aan het begin van de tekst wordt opgenomen.

```
\stelvoetnotenin[...]=...]
```

conversie	<u>cijfers</u> letters Letters romeins Romeins
wijze	<u>per</u> tekst <u>per</u> sectie
plaats	<u>pagina</u> tekst kolommen hoog geen
lijn	aan <u>uit</u>
voor	<i>commando</i>
na	<i>commando</i>
breedte	<i>maat</i>
hoogte	<i>maat</i>
korps	5pt ... 12pt <u>klein</u> groot
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
afstand	<i>maat</i>
kolomafstand	<i>maat</i>
margeafstand	<i>maat</i>
n	<i>getal</i>
nummercommando	<i>\commando#1</i>
splitsen	soepel streng zeerstreng <i>getal</i>
..=..	zie p 222: \omlijnd

Standaard worden voetnoten onderaan de bladzijde geplaatst. Als we met kolommen werken, dan forceert men door plaats op kolommen in te stellen dat de voetnoten in de laatste kolom worden geplaatst.

We kunnen kaders om voetnoten plaatsen, voetnoten in kolommen zetten en, wat soms handig is, voetnoten loskoppelen van de pagina. Van dit laatste geven we een voorbeeld.

```
\startlokaalvoetnoten[n=0]
  \plaatstabel
    {Een Tabel}
  \plaatslegenda
    {\starttabel[|r|]}
```

⁶ Een goede eigentijdse minimal kenmerkt zich niet zozeer door een minimaal gebruik van instrumenten, maar veeleer door een polyfoon verschuivende ritmiek.

⁷ Deze bestaat voornamelijk uit piano's, marimba's en xylofoons.

⁸ Zijn 'Fearful Symmetries' is een perfecte mix van klassiek, jazz, swing en pop.

⁹ Het aardige van muziek als 'Drumming' of 'Sextet' is de constatering dat "wat het oor hoort" anders en meer is dan "wat het oog ziet".

¹⁰ Dit is een publieke omroep.

¹¹ Soms ook bij andere omroepen, bijvoorbeeld op de achtergrond van aangrijpende documentaires doet muziek van Adams het vaak goed.

```

\HL
\VL Nota \voetnoot {Bene} \VL Bene \voetnoot {Nota} \VL\FR
\VL Bene \voetnoot {Nota} \VL Nota \voetnoot {Bene} \VL\LR
\HL
\stoptabel}
{\plaatslokalevoetnoten}
\stoplokalevoetnoten

```

Omdat we niet weten op welke bladzijde een tabel terecht zal komen, is het beter de voetnoten aan de tabel te koppelen. We schakelen daartoe tijdelijk over op lokale voetnoten.

Nota ¹	Bene ²
Bene ³	Nota ⁴

Tabel 4.5 Een Tabel

¹Nota

⁴Bene

```

\startlokalevoetnoten ... \stoplokalevoetnoten
..=.. zie p 90: \stelvoetnotenin

```

```

\plaatslokalevoetnoten[...=...=...=...]
..=.. zie p 90: \stelvoetnotenin

```

Ook kunnen voetnoten, in plaats van onderaan een bladzijde, aan het eind van een tekst of hoofdstuk worden geplaatst. Hiervoor zetten we plaats op tekst en gebruiken het volgende commando om de noten te plaatsen:

```

\plaatsvoetnoten[...=...=...=...]
..=.. zie p 90: \stelvoetnotenin

```

Door `n` op bijvoorbeeld 2 te zetten, kan men voetnoten in kolommen weergeven. Dit moet gebeuren voordat de noten zelf worden geplaatst, voornamelijk omdat \TeX de afmetingen van een voetnoot gebruikt bij het vaststellen van een paginaovergang. Meer informatie is te vinden in de broncode van de betreffende CONTEXT module: `core-not.tex`.

Het volgende voorbeeld demonstreert dat in plaats van een nummer ook een symbool kan worden gezet. Hier is conversie ingesteld op set 3.

- een harde noot kraken^{SSS}
- een hoge noot zingen*

^{SSS} Zo'n noot is te eten.

* Zo'n noot kun je horen.

Standaard is het nummercommando gelijk aan `\hoog`, maar een andere instelling mag ook. Bovendien kan men met:

```
\stelvoetnootdefinitiein[...]=...]
```

..=.. zie p 193: \doordefinieren

de exacte wijze van weergeven instellen, dit omdat gebruik gemaakt wordt van het standaard definitiemechanisme (zie paragraaf 10.2).

4.17 Uitgelijnde boxen

\TeX kent twee soorten boxen: `\hbox` en `\vbox`. Een horizontale `\hbox` is als het ware een regel, een `\vbox` een alinea. Alleen in de `\vbox` wordt dus uitgelijnd. Er zijn twee soorten vertikale boxen: een `\vbox` die uitlijnt op de laatste regel, en een `\vtop` die de baseline van de eerste regel als referentiepunt neemt.

```
\hbox{\hbox{een} \vbox{twee\par drie} \vtop{vier\par vijf}}
```

ziet er, als we de kadertjes ook zichtbaar maken, zoals hier met het commando `\showboxes`, uit als:

	twee	
een	drie	vier
		vijf

Aanvullend zijn binnen CONTEXT de drie commando's `\cbox`, `\lbox` en `\rbox` beschikbaar. Deze kunnen bijvoorbeeld worden gebruikt als argument bij bepaalde commando's of bij het zelf definiëren van commando's. Evenals bij `\hbox` en `\vbox` kunnen ook hier afmetingen worden meegegeven. Het betreft in alle drie gevallen de breedte.

```
\cbox{... tekst ...}
\lbox to 4cm{... tekst ...}
```

De gebruiker wordt uitgenodigd zelf wat te experimenteren met deze commando's. Een overgang naar een volgende regel wordt afgedwongen met `\\`.

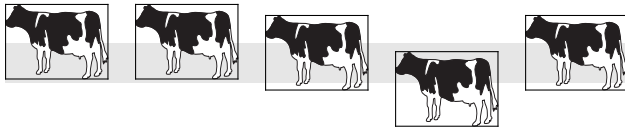
Voor erg gespecialiseerde doeleinden is er ook nog `\sbox`. Dit commando kan worden gebruikt om box de hoogte van een strutbox te geven. Dit is typisch een commando voor de wat meer gevorderde gebruiker, die op het moment dat hij het echt nodig heeft waarschijnlijk al weer vergeten is dat het bestaat.

Tot weer een andere categorie horen `\tbox` en `\bbox`, beiden kunnen worden gebruikt binnen bijvoorbeeld tabellen. In het onderstaande overzicht zien we alle types terug.

	a		a	a	a	a		aa	aa
aa	a	a	a	a	a	a		aa	aa
		a							

`\hbox` `\vbox` `\vtop` `\lbox` `\cbox` `\rbox` `\sbox` `\tbox` `\bbox`

De alternatieven `\tbox` en `\bbox` komen wat beter uit de verf bij figuren:



`\hbox` `\vbox` `\sbox` `\tbox` `\bbox`

Binnen `CONTEX`T is een heel repertoire aan geïnjekteerde macro's beschikbaar. Zo kunnen we aan een willekeurige box een markering toevoegen:

```
\setbox0=\vbox{The Final Cut\par --- \em Pink Floyd}
\makecutbox0 \box0
```

Let op, de markering valt buiten de box:

The Final Cut

— *Pink Floyd*

We kunnen boxen zichtbaar maken door in plaats van `\hbox`, `\vbox` en `\vtop` gebruik te maken van `\ruledhbox`, `\ruledvbox` en `\ruledvtop`. Met `\toonopmaak` kunnen we trouwens van alles automatisch zichtbaar maken en krijgen zo een kijkje achter de schermen van `CONTEX`T en `TEX`.

Het volgende voorbeeld toont aan dat we met `TEX` meer kunnen dan recht-toe-recht-aan teksten zetten. Wil men echter dit soort dingen doen, dan is wel enig inzicht nodig in het manipuleren van boxen. Zodra er tekst bij betrokken is, gebruiken we omwille van de overzichtelijkheid als het even kan buffers.

```
\startbuffer[water]
```

```
Drink geen water \crLf direct uit de kraan! \blanko
```

```
\start
```

```
  \tfx \stelinterliniein Het drinkwater is tijdelijk niet betrouwbaar.
  Kook het water voor consumptie ten minste 2~minuten. Zodra het water
  weer betrouwbaar is, krijgt u bericht. \par
```

```
\stop
```

```
\blanko[2*groot]
```

```
\language[en] Do not drink water \crLf directly from the tap! \blanko
```

```
\start
```

```
  \tfx \stelinterliniein The water is temporarily unfit for drinking.
  Boil the water during at least 2~minutes before consumption. As soon
  as the water is reliable again, you will be notified. \par
```

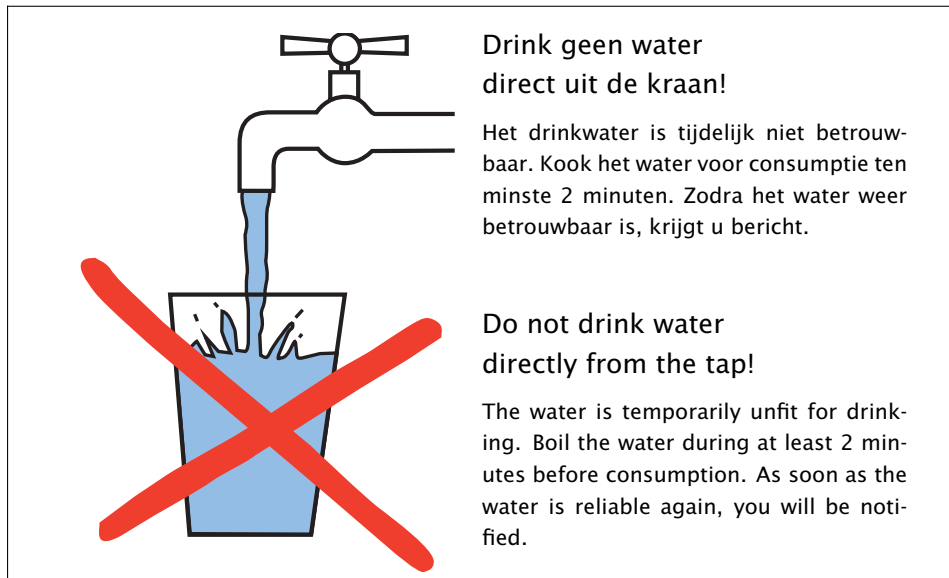
```
\stop
```

```
\stopbuffer
```

Deze tekst wordt gezet in een omliggende box. Daarbij gebruiken we twee kladboxen. De eerste bepaalt de hoogte van de tweede.

```
\omlijnd[offset=\korpsgrootte]
  {\setbox0=\vbox
    {\hsize 16em\switchnaarkorps[schreefloos]\haalbuffer[water]}}
  \setbox2=\vbox to \ht0
    {\vfill\externfiguur[vew1091a][breedte=5cm]\vfill}}
  \hskip1em\box2\hskip1em\box0\hskip1em}
```

Het resultaat van deze definitie, een voorbeeld van een kookadvies, zien we hieronder.



In plaats van `\tfx\stelinterliniein` kan men hier ook met `\switchnaarkorps` naar een kleiner korps omschakelen (`[klein]`). De `\par` blijft echter essentieel!

4.18 Opmaken

Een document heeft meestal een titelpagina, een colofonblad en wat andere bladzijden die geen deel uitmaken van de lopende tekst. Ze worden dan ook niet genummerd en hebben meestal geen hoofd- en voetregels. Omdat ze specifiek worden vormgegeven duiden we dergelijke pagina's aan met 'opmaak' en wanneer de standaard bladspiegel van toepassing is spreken we van de standaard opmaak.

Met de commando's `\startstandaardopmaak` en `\stopstandaardopmaak` kan binnen de standaard zetspiegel worden gewerkt. Hieronder is een eenvoudig voorbeeld van een titelpagina dat deze standaard commando's gebruikt.

```
\startstandaardopmaak
  \tfd Klussen in en rond het huis           \blanko[2*groot]
  \tfb Deel 1: Gas, water en elektra         \vfill
  \tfb J. Hagen \crlf A.F. Otten            \blanko
  \tfb Hasselt \crlf \huidigedatum[maand,jaar]
\stopstandaardopmaak
```

Bij dubbelzijdig zetten wordt standaard een lege achterkant gegenereerd. Het kan echter voorkomen dat de achterkant van zo'n opmaak ook een opmaak moet zijn. Dit gaat als volgt:

```
\startstandaardopmaak[dubbelzijdig=nee]
... de voorkant
\stopstandaardopmaak
\startstandaardopmaak[pagina=nee]
... de achterkant
\stopstandaardopmaak
```

In het eerste geval wordt geen achterkant gegenereerd, in het tweede geval wordt niet overgegaan naar een rechterpagina (wat standaard gebeurt). Dit voorbeeld toont ons tevens dat instellingen direct kunnen worden meegegeven.

Met het commando `\toonkader` kunnen (tijdelijk) hulplijnen worden opgeroepen rond de te zetten tekst. Dit is aan te raden bij het ontwerpen van een layout en/of het opmaken van individuele pagina's.

Naast het genoemde commando `\startstandaardopmaak` kan men zelf een opmaak met andere maatvoering definiëren.

```
\definieeropmaak[...][...,...=...,...]
```

```
...      naam
..=..   zie p 95: \stelopmaakin
```

```
\stelopmaakin[...][...,...=...,...]
```

```
...      naam
breedte   maat
hoogte    maat
voffset   maat
hoffset   maat
pagina    links ja rechts
commandos commando
dubbelzijdig ja nee leeg
hoofdstatus normaal stop start leeg geen geenmarkering
voetstatus  normaal stop start leeg geen geenmarkering
tekststatus  normaal stop start leeg geen geenmarkering
bovenstatus  stop start
onderstatus  stop start
paginastatus stop start
kleur      naam
```

```
\startnaamopmaak ... \stopnaam
```

Het eerste commando genereert een `\start...stop`-paar waartussen zetopdrachten en teksten kunnen worden opgenomen. Een en ander wordt op een lege bladzijde gezet. De pagina wordt gemarkeerd als *naam*, dit ten behoeve van selecteren (zie paragraaf ??).

De achter commandos opgegeven commando's worden direct uitgevoerd. Op deze manier kunnen bijvoorbeeld 'lokale' commando's worden gedefinieerd.

5

Typografie

Typografie

5.1 Inleiding

Door de eeuwen heen zijn methoden ontwikkeld en aangepast om feiten en gedachten aan een medium toe te vertrouwen. Een erg efficiënte methode is het gebruik van logogrammen, zoals bijvoorbeeld Chinezen al sinds mensenheugenis gebruiken. Een andere methode vertaalt elke klank in een symbool. Dit gebeurt bijvoorbeeld in Japan bij het schrijven van telegrammen. De meest bekende manier van opslaan maakt echter gebruik van een beperkte set symbolen die elk een of meer fonemen (klankelementen) verbeelden. We noemen zo'n collectie een alfabet.

Hoewel $\text{T}_{\text{E}}\text{X}$ in eerste instantie bedoeld is voor het zetten van alfabetisch geörienteerde talen, kan het met wat kunstgrepen ook uit de voeten met de andere twee methoden. In deze handleiding zullen we ons echter concentreren op de talen die een alfabet gebruiken.

De symbolen die de karakters in een alfabet representeren zijn min of meer gestandaardiseerd. Hierdoor zijn ze herkenbaar voor lezers, zelfs als ze op details verschillen. Een collectie symbolen, die we in dat verband vaak aanduiden als glyphs, duiden we aan met de naam font.



The image shows the word "gok" repeated four times in a row, each in a different font style. From left to right: Computer Modern (a clean, modern sans-serif), Lucida Bright (a classic serif), Times Roman (a traditional serif), and Antiqua Torunka (a decorative, historical serif). The text is centered within a rectangular box with a thin border.

Van links naar rechts zien we hier een Computer Modern, een Lucida Bright, een Times Roman en een Antiqua Torunka font, allen geschaald naar 60pt. Fonts zijn zo ontworpen dat een bladzijde tekst prettig overkomt en lezen wordt aangemoedigd.



The image shows the word "lap" repeated four times in a row, each in a different font style. From left to right: a standard weight, a bold weight, an italic weight, and a bold italic weight. All are in the Lucida Bright font. The text is centered within a rectangular box with a thin border.

Binnen een bepaald letterontwerp komen in de regel varianten voor die weliswaar dezelfde vorm hebben, maar toch wezenlijk verschillen. Hierboven zien we bijvoorbeeld een standaardvorm, een vette variant, een bijpassende italic en een vette italic, allen Lucida Bright.

We zullen in deze handleiding niet diepgaand ingaan op de wijze waarop letters worden gecombineerd tot woorden, die op hun beurt regels vormen, al dan niet met afbrekingen, samen goed voor een paragraaf. In dit hoofdstuk staat het wisselen van font centraal.

De afstand tussen de verschillende glyphs kan afhangen van de combinatie. Het volgende voorbeeld illustreert dit. Zowel de afstand tussen de b en de o als die tussen de o en de x is aangepast. We noemen dit kerning.

In dit voorbeeld tonen we een Computer Modern, het standaard $\text{T}_{\text{E}}\text{X}$ lettertype. Dit door Knuth ontworpen lettertype is een variant op een Monotype Modern, en kent, in tegenstelling tot de Lucida die in deze handleiding wordt gebruikt, relatief veel kerning paren.

Dit soort micro-typografie valt buiten de invloedssfeer van de gebruiker, het maakt deel uit van het letter ontwerp. De gebruiker kan echter wel invloed uitoefenen op de keuze van een lettertype, de afstand tussen regels, het oogje dat $\text{T}_{\text{E}}\text{X}$ dichtknijpt bij het zetten van een paragraaf, en nog enkele aspecten. In dit hoofdstuk beperken we ons tot de keuze van het lettertype.

Er zijn verschillende manieren om het karakter van een lettertype te benoemen. Men kan bijvoorbeeld klassificeren naar historisch gezien de tijd waarin een letter is ontworpen, de kenmerken van het lettertype, of de gebruikswaarde, bijvoorbeeld een boek of krant.

Hier zien we naast elkaar vijf gedaanten van de Lucida: de Bright, Sans, Typewriter, Handwriting en Calligraphy. Er zijn slechts weinig ontwerpen die zo veel stijlen herbergen. De Computer Modern is ook zo'n ontwerp. Bovendien kent dit ontwerp als een van de weinige nog ontwerp-grootten, iets dat vroeger heel normaal was, maar tegenwoordig een zeldzaamheid. Hieronder zien we, uitvergroet tot 48 punten, het verschil tussen een 5, 7, 9, 12 en 17 punts ontwerp.

We beperken ons hier tot de eerste drie verschijningsvormen Roman, Sans en Typewriter. Als we voor een meer algemene aanduiding kiezen, komen we terecht bij de typering in tabel 5.1.

Serif	Sans	Mono
Regular	Support	Mono
Roman	Sans	Type

Tabel 5.1 Enkele manieren om een style te classificeren.

Hoewel de eerste twee reeksen beter aansluiten bij de typografische wereld, gebruiken we binnen $\text{CON}\text{T}\text{E}\text{X}\text{T}$ vaak de laatste, dit omdat hij beter aansluit bij de traditionele benamingen binnen plain TEX . Zo wordt `\rm` gebruikt om over te gaan op een roman/serif/regular stijl, en `\tt` voor mono gespatieerde of typewriter stijl.

In de volgende paragrafen gaan we in op de wijze waarop men stijl en/of varianten instelt. Een waarschuwing is daarbij op zijn plaats. Gezien de relatie tussen fonts, iets dat met name tot uitdrukking komt in de wiskundige modus (math mode), is het achterliggende mechanisme redelijk complex. Deze situatie wordt nog gecompliceerd door het feit dat zowel de rangschikking van de karakters die men intypt, als de ordening van de karakters in een font file (beiden aangeduid met encoding vector) roet in het eten kunnen gooien. Enige kennis van fonts en hun eigenaardigheden is in dat geval nodig.

5.2 Het mechanisme

Het wisselen van font wordt aangeduid als een font-switch. Omdat zo'n mechanisme onmisbaar is, is het tegelijk een van de oudste onderdelen van $\text{CON}\text{T}\text{E}\text{X}\text{T}$. Wel zijn in de loop der tijd met name in de wijze van definiëren uitbreidingen gerealiseerd. Bij het opzetten van het mechanisme zijn de volgende uitgangspunten gehanteerd.

- Er kan eenvoudig worden gewisseld van *stijl*, dat wil zeggen: roman (serif, regular), sans serif (support), teletype (ook wel monospaced) enz. (`\rm`, `\ss`, `\tt` enz.)
- Er zijn meerdere *varianten* van letters beschikbaar, zoals schuin en vet (`\sl` en `\bf`).
- Er worden verschillende *families* ondersteund, waaronder Computer Modern Roman en Lucida Bright.
- Er kan eenvoudig worden gewisseld van *korps*, waarbij standaard de korpsen tussen 8pt en 12pt beschikbaar zijn.
- Binnen een korps zijn verschillende *formaten* mogelijk, bijvoorbeeld voor sub- en superscripts. De *formaten* kunnen door elkaar worden gebruikt, met behoud van stijl, variant en familie.
- Er wordt rekening gehouden met de specifieke kenmerken van een *font*, waaronder de wijze waarop het font is gedefinieerd, zoals de encoding vector.

Letters, of beter gezegd: de grafische interpretatie van letters, zijn in verschillende korpsen (afmetingen) beschikbaar. Als eenheid wordt de punt gebruikt, de pt. De beschikbaarheid wordt geregeld in definitie-files. In betere tijden werden voor elke afmeting aparte letters ontworpen, een traditie die eigenlijk alleen is voortgezet in de standaard TEX fonts. Bij de meeste fonts zal men het moeten doen met een geschaald 10 punts ontwerp.

Om herhaald definiëren te voorkomen, zijn standaard de meest gebruikelijke afmetingen gedefinieerd: 8–14.4 punten. Wanneer men onverhoopt een nog niet gedefinieerde afmeting instelt, bijvoorbeeld 32 punt op een titelpagina, dan zal $\text{CON}\text{T}\text{E}\text{X}\text{T}$ dit formaat zelf definiëren, binnen de randvoorwaarden van de standaard korpsomgeving. Standaard wordt met een precisie van 1 digit gewerkt (instelbaar op 0, 1 en 2). Dit voorkomt onnodig laden van formaten met slechts kleine verschillen. Uitgangspunt bij de afronding is dat iets kleiner vaak minder erg is dan wat groter.

Het in een tekst te gebruiken korps, de stijl en de familie worden ingesteld met het commando:

```
\stelkorpsin[.,.,.,.,.]
... naam serif regular romaansans support schreefloos mono type teletype handschrift
calligrafie 5pt ... 12pt
```

terwijl lopende de tekst kan worden omgeschakeld met het commando:

```
\switchnaarkorps[.,.,.,.]
... 5pt ... 12pt klein groot globaal
```

Dit laatste commando laat de instellingen van het in hoofd en voet gebruikte korps intact. Met `klein` en `groot` gaat men over op een kleiner korps, zo klein dus, of een groter korps.

Het commando `\stelkorpsin` wordt als het goed is maar eenmaal gebruikt, in de stijldefinitie! Tussentijds wisselen vindt plaats met `\switchnaarkorps`. Haal dit niet door elkaar, anders kunnen vreemde, maar volledig legitieme, nevenverschijnselen optreden. Het eerstgenoemde commando heeft namelijk ook betrekking op de hoofd- en voetregels.

\TeX zoekt de informatie over een bepaald lettertype in een file met de extensie `tfm`. Hoewel het mogelijk is dergelijke files van te voren te laden, stelt `CON \TeX T` dit laden zo lang mogelijk uit. De reden hiervoor is dat dergelijke files per systeem kunnen verschillen.

De voor hoofd- en voetregels en voetnoten gebruikte letterformaten worden na het geven van dit commando automatisch aangepast, evenals de interlinie en de sprongen. De volgende, korte commando's, beperken zich echter tot de tekst: `\vi`, `\vii`, `\viii`, `\ix`, `\x`, `\xi` en `\xii`.

De commando's:

```
{\xii met deze zetomgeving \par}
{\xi is het mogelijk om \par}
{\x een hele redelijke \par}
{\ix ogentest te maken: \par}
{\viii a x c e u i w m q p \par}
```

De interlinie wordt automatisch aangepast, dit wordt links getoond. Rechts zien we wat er gebeurt als de interlinie niet zou zijn aangepast.

met deze zetomgeving	met deze zetomgeving
is het mogelijk om	is het mogelijk om
een hele redelijke	een hele redelijke
ogentest te maken:	ogentest te maken:
a x c e u i w m q p	a x c e u i w m q p

5.3 Omschakelen

Het mechanisme voor het wisselen van korps en lettertype is vrij geavanceerd en daarom niet eenvoudig uit te leggen. Er wordt onderscheid gemaakt in verschillende families. Binnen deze families onderscheiden we een of meer stijlen.

Voorbeelden van stijlen zijn: roman, sans serif en teletype. In enkele gevallen is een handschrift en/of calligrafische letter beschikbaar. Voorbeelden van varianten binnen een stijl zijn: **boldface** en *slanted*.

Er zijn verschillende manieren om van lettertype te wisselen. Zo gaat men met `\ss` over op een sans serif letter. Vanaf dat moment zijn commando's als `\bf` aangepast op deze nieuwe stijl. Als men consequent gebruik maakt van commando's als `\bf` en `\sl`, krijgt men als men bovenaan de tekst van stijl wisselt, automatisch de juiste overgangen op vet en schuin. Een (aanzienlijk) snellere manier van wisselen is ook beschikbaar in de vorm van: `\ssbf`, `\sss\sl` enz. Deze snelle manier maakt echter het als geheel in een andere stijl zetten van een tekst onmogelijk.

Hieronder zijn de varianten weergegeven. De afkorting `\sl` staat voor *slanted*, `\it` staat voor *italic* en `\bf` voor **boldface**. Aanvullend zijn eventueel ook `\bs` en `\bi` beschikbaar, ofwel: ***bold slanted*** en ***bold italic***. Als een variant niet beschikbaar is, dan wordt hij op een redelijk alternatief geprojecteerd.

Met `\os` duiden we aan dat we de mediaeval of old-style cijfers 139 willen in plaats van 139. De aanduiding `\sc` staat voor SMALL CAPS. Met een `x` duiden we een kleiner lettertype aan, met `a`, `b`, `c` en `d` een groter. De actuele stijl wordt aangegeven met `\tf` ofwel `typeface`.

```
\tfa \tfa \tfa \tfa
\tfx \bfx \slx \itx
\bf \sl \it \bs \bi \sc \os
```

Afhankelijk van de volledigheid van de definitie-files zijn ook varianten als `\bfa`, `\bfb` enz. beschikbaar. Als een alternatief niet voorhanden is, wat nog wel eens het geval is voor *italic* of *slanted*, of hun vette companen, dan wordt een redelijk alternatief gekozen. In dat geval zijn bijvoorbeeld *italic* en *schuin* hetzelfde.

Voor de liefhebber is er `\tx` beschikbaar. Dit is equivalent met `\tfx`, `\bfx`, `\slx` enz., afhankelijk van de actuele variant. Nog kleiner kan ook: `\txx`. Binnen `\tx` gaat nogmaals `\tx` over op deze nog kleinere variant.

Het frequent wisselen van letter leidt tot lange verwerkingstijden. Als geen gebruik wordt gemaakt van super- en subscripts en als men echt zeker weet welke letter men wil gebruiken, dan kan men meestal ook een variant op naam oproepen: `\rms\sl`, `\ssbf`, `\tttf` enz.

Het wisselen van stijl vindt plaats met een van de commando's:

```
\rm \ss \tt \hw \cg
```

Als `\rm` gekozen is, leest `CONTEXT` het commando `\tfd` als `\rmd`. Alle standaardinstellingen maken gebruik van `tf`-instellingen en passen zich dus automatisch aan.

De verschillende commando's passen zich steeds zo goed mogelijk aan bij de op dat moment actuele instellingen van het lettertype en formaat. Bijvoorbeeld:

```
{\rm test {\sl test} {\bf test} \tfc test {\tx test} {\bf test}}
{\ss test {\sl test \tx test} {\bf test \tx test}}
```

levert:

test *test* **test** test **test**
 test *test* test **test** test

Als een letter niet voorhanden is, dan wordt er automatisch een acceptabel alternatief gekozen.

Op het typografische enigszins verwerpelijke onderstrepen en doorhalen gaan we hier niet in. Deze worden besproken in 11.5 (“Onderstrepen”).

5.4 Letters

Een aantal commando's heeft een parameter `letter` waarmee een lettertype kan worden ingesteld. In dat geval kunnen commando's als `\sl` of `\rma` worden meegegeven, maar ook een van de volgende aanduidingen:

normaal vet schuin vetschuin italic vetitalic type
 klein kleinvet kleinschuin ... kleinitalic ... kleintype
 kapitaal

Hoewel flexibiliteit zijn grenzen kent, worden bij de parameter `letter` zowel `vet` als `\bf` of `bf` geaccepteerd. Zelfs de achter de schermen opererende naam `12ptrmbf` is toegestaan. Deze laatste variant wordt afgeraden.

5.5 Beschikbare alternatieven

Er zijn slechts weinig font sets (korsen) beschikbaar die ook wiskunde aan kunnen. Natuurlijk is er de Computer Modern Roman, maar de zeer fraaie Lucida Bright en de door uitgevers zeer gewaardeerde Times zijn ook bruikbaar. Al deze fonts beschikken over een volledige set karakters voor wiskundig zetwerk. De Computer Modern Roman onderscheidt zich zowel door volledigheid als door de hoogwaardige kwaliteit van andere fonts. Er zijn op dit ontwerp enkele aanvullingen beschikbaar: Euler en Concrete.¹⁴

De Computer Modern Roman bestaat uit zo'n 70 lettertypes en letterformaten. Omdat een aantal formaten niet standaard aanwezig kunnen worden geacht, zijn onder de optie `cmr` bijvoorbeeld de 11 punts letters gedefinieerd als geschaalde 9 en 10 punts letters. Met `eu1` en `con` krijgen we een variant op de Computer Modern.

```
\toonkorps[.,.,.,.]  
... zie p 100: \stelkorpsin
```

Met behulp van het commando `\toonkorps` kan een overzicht worden gegenereerd van de beschikbare lettertypen. Hieronder is het 12pt-korps Computer Modern Roman (`cmr`) weergegeven. De oplettende lezer zal zien dat niet alle varianten (standaard) beschikbaar zijn.

¹⁴ Zie Concrete Mathematics van Knuth cs., in veel opzichten zowel typografisch als didactisch een perfect boek.

[cmr, 12pt]										\mr : Ag			
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfxx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

Zoals hieronder te zien is, is een 12pt Lucida Bright (lbr) wat groter dan een 12pt Computer Modern Roman. Een x-letter, bijvoorbeeld \bfx is standaard 2pt kleiner dan het op dat moment gebruikte formaat. De grotere formaten zijn met behulp van T_EX's \magstep geschaald.

[lbr, 12pt]										\mr : Ag			
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfxx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

Een opmerking tot slot. Als men een van de grotere letterformaten heeft gekozen, bijvoorbeeld \tfb, dan is op dat moment \tf gelijk aan \tfb, \bf gelijk aan \bfb enz. Deze werkwijze bleek in de praktijk de voorkeur te verdienen boven terugkeer naar het standaard letterformaat.

5.6 Benadrukken

Binnen de meeste macropaketten is wel het commando \em beschikbaar. Dit commando gedraagt zich als een kameleon, dat wil zeggen dat het zich aanpast aan de actuele stijl. Ook binnen CON_TE_XT is \em beschikbaar. Het commando heeft de volgende kenmerken:

- er kan worden overgegaan op *slanted* of *italic*
- er wordt binnen \bf overgegaan op ***bold slanted*** of ***bold italic*** (als beschikbaar)
- er vindt automatisch een zogenaamde ‘italic correction’ plaats (\/)

De vette schuine of italic letter wordt natuurlijk alleen ondersteund als \bs en \bi als zodanig beschikbaar zijn.

De afkorting {\em em} staat voor {\em emphasis}.

{\em De afkorting {\em em} staat voor {\em emphasis}.}

{\bf De afkorting {\em em} staat voor {\em emphasis}.}

{\em \bf De afkorting {\em em} staat voor {\em emphasis}.}

{\it De afkorting em {\em staat \bf voor} emphasis.}

{\sl De afkorting em {\em staat \bf voor} emphasis.}

Het bovenstaande levert: De afkorting *em* staat voor *emphasis*. De afkorting *em* staat voor *emphasis*. **De afkorting *em* staat voor *emphasis*. De afkorting *em* staat voor *emphasis*. De afkorting *em* staat voor *emphasis*. De afkorting *em* staat voor *emphasis*.** De afkorting *em* staat voor *emphasis*. De afkorting *em* staat voor *emphasis*.

Het voordeel van het gebruik van `\em` boven `\it` en/of `\sl` ligt in het feit dat op deze manier consistentie wordt afgedwongen.

Standaard is de *emphasis* ingesteld op *schuin*, maar in deze tekst staat ze op *italic*. Dit is in te stellen met:

```
\stelkorpsomgevingin[default][em=italic]
```

5.7 Kapitalen

Woorden en afkortingen kunnen in kapitalen worden weergegeven. Zowel kleine letters als hoofdletters worden automatisch geconverteerd. Een met `\kap` opgeroepen kapitaal is een x-letter. Bij een overgang naar schuin (`\sl`), vet (`\bf`) enz. verandert de kapitaal mee.

```
\kap{...}
... tekst
```

```
\Kap{...}
... tekst
```

```
\KAP{...}
... tekst
```

```
\Kaps{.. ... ..}
... tekst
```

Het eerste commando converteert alle letters naar hoofdletters. Het verdient aanbeveling zelf geen hoofdletters te gebruiken, dit omdat bij echte klein-kapitalen (small caps) verschil bestaat tussen hoofd- en kleine letters.

Het is, gezien het grote aantal wetten dat wordt uitgevaardigd, zeer waarschijnlijk dat de kapitaal `\kap{w}` het meest wordt gebruikt. Er zijn voorbeelden te over: `\kap{ww}`, `\kap{ww}` en `\kap{bw}`. Niet iedere `\kap{w}` staat overigens voor een wet, neem nu `\kap{www}`.

heeft als resultaat:

Het is, gezien het grote aantal wetten dat wordt uitgevaardigd, zeer waarschijnlijk dat de kapitaal w het meest wordt gebruikt. Er zijn voorbeelden te over: wvw, ww en BW. Niet iedere w staat overigens voor een wet, neem nu www.

Een `\kap` binnen een `\kap` wordt afgevangen en hoeft dus niet tot problemen te leiden:

```
\kap{Kapitaalkrachtige mensen maken niet zelden de \kap{kapitale}
fout hun \kap{kapitaal} exponentieel te doen willen stijgen.
Speculatie kan namelijk tot \nokap{kapitaalvernietiging} leiden.}
```

of:

```
KAPITAALKRACHTIGE MENSEN MAKEN NIET ZELDEN DE KAPITAALKAPITALE FOUT HUN KAPITAALKAPITAAL EXPO-
NENTIEEL TE DOEN WILLEN STIJGEN. SPECULATIE KAN NAMELIJK TOT kapitaalvernietiging LEIDEN.
```

We zien in dit voorbeeld dat binnen een `\kap` tijdelijk het in kapitalen zetten kan worden onderbroken met `\nokap`.

```
\nokap{...}
... tekst
```

Het commando `\Kap` maakt van de eerste letter een hoofdletter en `\KAP` maakt hoofdletters van de letters die worden voorafgegaan door `\`. Met `\Kaps` ten slotte, kan men van meerdere woorden de eerste letter een hoofdletter maken.

```
\stelkapitalenin[...]=...]
titel ja nee
sc ja nee
```

Met dit commando kunnen wat kenmerken van kapitalen worden ingesteld. De instelling `sc=ja` schakelt over op echte SMALL CAPS. Met `titel` bepalen we of in titels kapitalen worden gehonoreerd.

Naast deze `\kap`-commando's zijn er nog:

```
\Woord{...}
... tekst
```

en

```
\Woorden{... ..}
... tekst
```

Deze commando's maken van de beginletter(s) van woorden hoofdletters. Een woord kan in zijn geheel worden omgezet in hoofdletters met:

```
\WOORD{...}
... tekst
```

We sluiten af met een voorbeeld van echte klein-kapitalen. Als deze beschikbaar zijn, dan is het wellicht fraaiër om bij afkortingen en logo's de echte klein-kapitaal `\sc` te gebruiken, dan een pseudo-kapitaal `\kap`. Dit kan men instellen met de parameter `letter`.

Soms verwordt een afkorting tot woord zonder bepaalde betekenis, dit geldt bijvoorbeeld voor de namen van de zendgemachtigden `{\sc vara}` en `{\sc vpro}`. Wat heeft `{\sc Pragma}` eigenlijk ooit betekend?

Levert:

Soms verwordt een afkorting tot woord zonder bepaalde betekenis, dit geldt bijvoorbeeld voor de namen van de zendgemachtigden `VARA` en `VPRO`. Wat heeft `PRAGMA` eigenlijk ooit betekend?

HET IS NATUURLIJK ALTIJD MOGELIJK EEN STUKJE TEKST IN KLEIN-KAPITALEN TE ZETTEN. BESEF ECHTER WEL DAT ONDERKAST LETTERS MEER DISCRIMINEREN EN DUS GEMAKKELIJKER LEZEN.

Een belangrijk verschil tussen `\kap` en `\sc` is dat het laatste commando staat voor een speciaal ontworpen lettertype. Het commando `\kap` past zich daarentegen aan het actuele lettertype aan: *KAP*, **KAP**, **KAP** enz.

Sommige zetpakketten rekken de woorden uit om een acceptabele uitlijning te krijgen. Dit wordt in `CONTEXT` bewust niet ondersteund! Wel kunnen, bijvoorbeeld in titels, woorden worden uitgerekt met:

```
\uitgerekt{...}
... tekst
```

Er wordt bij het uitrekken uitgegaan van de actuele breedte.

```
\hbox to \hsize {\uitgerekt{hier\zit\veel\rek\in ...}}
\hbox to 20em {\uitgerekt{... en\hier\wat\minder}}
```

Met `\` geven we een spatie aan (`{}` mag ook).

```
h i e r z i t v e e l r e k i n . . .
. . . e n h i e r w a t m i n d e r
```

Dergelijke escapades zijn typografisch eigenlijk alleen toegestaan in koppen. De aan het bovenstaande commando ten grondslag liggende macro's lossen dit op door de tekst karakter voor karakter in te lezen en af te handelen.

5.8 Getypte tekst

Tekst kan in getypte vorm worden weergegeven. Daarbij wordt de indeling in regels gehandhaafd. De tekst wordt opgenomen tussen de commando's:

```
\starttypen ... \stoptypen
```

Zoals in:

```
\starttypen
```

In deze tekst zijn voorbeelden genoeg te vinden van getypte tekst.

De commando|definities en voorbeelden worden met behulp van de genoemde commando's weergegeven, zo ook dit voorbeeld.

```
\stoptypen
```

Voor in de tekst opgenomen typewerk is het commando `\type` beschikbaar.

```
\type{...}
... tekst
```

Een file kan in getypte vorm in de tekst worden opgenomen met behulp van het commando:

```
\typefile{.1.}{.2.}
.1. naam
.2. file
```

Instellingen vinden plaats met:

```
\steltypenin[...][...,...=,...]
... file typen naam
spatie aan uit
pagina ja nee
optie schuin normaal commandos kleur geen
tekst ja nee
icommando commando
vcommando commando
ccommando commando
voor commando
na commando
marge maat standaard ja nee
evenmarge maat
onevenmarge maat
blanko maat klein middel groot standaard halveregeel regel
escape /
springvolgendein ja nee
letter normaal vet schuin vetschuin type kap klein... commando
kleur naam
palet naam colorpretty
regels ja nee afgebroken
```

Deze setup beïnvloedt de display verbatim (`\starttypen`) en het verbatim zetten van files (`\typefile`) en buffers (`\typebuffer`). Het eerste, optionele argument kan worden gebruikt om de specifiek verbatim omgeving te specificeren.

```
\steltypenin[file][marge=standaard]
```

Als spatie=aan, dan worden de spatie's zichtbaar gemaakt:

```
Niet-uitlijnen_ verdient_ mijns_ inziens_ de_ voorkeur
boven_ uitlijnen_ met_ behulp_ van_ spaties
en_ Z_ e_ k_ e_ r_ boven_ het_ oprekken_ van_ woorden
```

Een bijzonder geval is het volgende:

```
\definieertypen
[breedtypen]
```

```
\steltypenin
[breedtypen]
[onevenmarge=-1.5cm,evenmarge=-.75cm]
```

Dit kan worden gebruikt als:

```
\startbreedtypen
```

Soms kan een regel verbatim wel erg lang zijn, en omdat we niet afbreken, drukken we hem op de oneven pagina's dus maar in de marge.

```
\stopbreedtypen
```

Op een linker bladzijde wordt de in verbatim gezette tekst in de kantlijn gedrukt.

Soms kan een regel verbatim wel erg lang zijn, en omdat we niet afbreken, drukken we hem op de oneven pagina's dus maar in de marge.

We kunnen in-line verbatim instellen met:

```
\steltypenin[...]=...]
```

spatie	aan <u>uit</u>
optie	schuin normaal <u>geen</u>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
kleur	<i>naam</i>

Als bij de instellingen `optie` de waarde `schuin` heeft, wordt alle tekst tussen `<<` en `>>` in een *schuine typeletter* gezet. Dit kan bij alle hierboven genoemde commando's. Zo levert `\type{aa<<bb>>cc}` op: *aabbcc*.

Omwille van de leesbaarheid kan in plaats van de *buitenste* haakjes ook gebruik worden gemaakt van andere karakters dan `{` en `}`. Er kan een eigen, niet actief, karakter worden gekozen, bijvoorbeeld: `\type+ziezo+` of `\type-ziezo-`. Ook kunnen de reeds genoemde `<<` en `>>` worden gebruikt: `\type<<ziezo>>` of zelfs `\type<ziezo>`.

De instelling `optie=commandos` maakt het mogelijk in een getypte tekst commando's uit te voeren. Bij deze optie wordt `\` vervangen door `/`.

Ook deze optie is met `name` bedoeld voor het opstellen van handleidingen, bijvoorbeeld als een toelichting moet worden opgenomen:

```
\ziezo <</rm : dit commando doet niets>>
```

```
/vdots
```

```
\ozeiz <</sl : en dit commando ook niet>>
```

De dubbele << en >> vervullen hier de functie van {}.

Binnen de type-commando's wordt gebruik gemaakt van \tttf. Als we \tt hadden gebruikt, dan zou \sl een schuine en \bf een vette typeletter opleveren. Nu gebeurt dit dus niet:

```
\ziezo : dit commando doet niets
```

```
\ozeiz : en dit commando ook niet
```

Wellicht de meest interessante optie betreft het verbatim zetten van een programma source. We beperken ons hier tot het verschijnsel, en verwijzen voor meer details naar de documentatie in de files verb-xxx.tex en cont-ver.tex. In de laatst genoemde file vinden onder meer we de regels:

```
\definieertypen [MP] [optie=MP]
\definieertypen [PL] [optie=PL]
\definieertypen [JS] [optie=JS]
\definieertypen [TEX] [optie=TEX]
```

Allereerst zien we dat het mogelijk is een eigen verbatim omgeving te definiëren. Dit gebeurt met het commando:

```
\definieertypen[...][...=...]
```

... file typen *naam*
 ..=.. zie p 107: \steltypenin

De bovenstaande definities koppelen zo'n omgeving aan een optie.

```
\startMP
beginfig (12) ;
  MyScale = 1.23 ;
  draw unitsquare scaled MyScale shifted (10,20) ;
endfig ;
\stopMP
```

Dit ziet er gezet uit als:

```
beginfig (12) ;
  MyScale = 1.23 ;
  draw unitsquare scaled MyScale shifted (10,20) ;
endfig ;
```

Deze opties zorgen er dus voor dat de tekst wordt gezet conform de ingestelde taal. Het is mogelijk meerdere filters te schrijven, standaard worden METAPOST en METAFONT ondersteund, PERL, JAVASCRIPT, en natuurlijk T_EX. De wijze van weergeven is gekoppeld aan het

kleurmechanisme, waarbij gebruik wordt gemaakt van paletten; vandaar de instelling `palet` in `\steltypenin`.

Men kan ook fontwisselingen beïnvloeden of zelfs eigen commando's inhaken, door een waarde toe te kennen aan `icommand` (identifiers), `vcommand` (variabelen) en `ccommand` parameters (de rest). De standaard toekenningen zijn:

```
\steltypenin [icommando=\ttsl, vcommando=, ccommando=\tf]
```

Tot slot bieden we nog wat alternatieven voor `\type`. De met dit commando weergegeven woorden worden niet afgebroken. Wil men dit toch, dat is er:

```
\typ{...}
... tekst
```

Moch men onverhoopt de behoefte hebben een handleiding over \TeX te willen zetten, dan kunnen de volgende twee commando's van pas komen:

```
\tex{...}
... tekst
```

```
\arg{...}
... tekst
```

Het eerste plaatst een `\` voor de getypte tekst, het tweede omringt de tekst met `.`

5.9 Wiskunde

Veel van de \TeX gebruikers hebben voor dit programma gekozen omdat het zo goed is in wiskundig zetwerk. De betrokkenheid van \TeX op wiskundige typografie heeft zijn stempel gedrukt op het font mechanisme. We zullen de lezer niet belasten met details, maar centraal element is de familie. Elke variant vormt een eigen familie. Er is dus een familie voor `\bf`, `\it`, enz. Binnen een familie onderscheiden we drie leden: text, script en scriptscript, ofwel normaal, kleiner, nog kleiner. De normale afmetingen worden gebruikt voor de lopende tekst, de kleinere voor bijvoorbeeld superscripts. Het volgende voorbeeld laat zien wat de verschillende varianten doen.

```


$$\text{\tf } x^2+\text{\bf } x^2+\text{\sl } x^2+\text{\it } x^2+\text{\bs } x^2+ \text{\bi } x^2 =\text{\rm } 6x^2\$$$


$$\text{\tf } x^2+\text{\bf } x^2+\text{\sl } x^2+\text{\it } x^2+\text{\bs } x^2+ \text{\bi } x^2 =\text{\tf } 6x^2\$$$


$$\text{\tf } x^2+\text{\bf } x^2+\text{\sl } x^2+\text{\it } x^2+\text{\bs } x^2+ \text{\bi } x^2 =\text{\bf } 6x^2\$$$


$$\text{\tf } x^2+\text{\bf } x^2+\text{\sl } x^2+\text{\it } x^2+\text{\bs } x^2+ \text{\bi } x^2 =\text{\sl } 6x^2\$$$


```

Gezet wordt dit:

$$x^2 + \mathbf{x}^2 + x^2 + x^2 + \mathbf{x}^2 + \mathbf{x}^2 = 6x^2$$

$$x^2 + \mathbf{x}^2 + x^2 + x^2 + \mathbf{x}^2 + \mathbf{x}^2 = 6x^2$$

$$x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2$$

$$\mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 = 6\mathbf{x}^2$$

We zien dat de karakters zich aanpassen, maar dat de symbolen in hetzelfde font worden gezet. Technisch gezien worden ze gezet in het lettertype dat is toegekent aan familie 0 (er zijn er maximaal 16), en dat is in ons geval standaard `\tf`.

Dat het ook anders kan, toont het volgende voorbeeld. We zien in dit voorbeeld een nieuw commando opduiken: `\mf`, wat staat voor *math font*. Dit commando zorgt ervoor dat de symbolen worden gezet in het laatst gekozen font.

$$x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2$$

$$\mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 = 6\mathbf{x}^2$$

$$x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2$$

$$\mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 = 6\mathbf{x}^2$$

$$x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2$$

$$\mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 + \mathbf{x}^2 = 6\mathbf{x}^2$$

Met moet er overigens rekening mee houden dat in veel gevallen \TeX de formule als geheel zet, en niet per se van voor naar achter werkt. Instellingen aan het eind kunnen dan ook doorwerken naar het begin.

$$\text{\tf\mf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2\$$$

$$\text{\bf\mf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2\$$$

$$\text{\sl\mf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2\$$$

$$\text{\bs\mf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2\$$$

$$\text{\it\mf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2\$$$

$$\text{\bi\mf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = 6x^2\$$$

De exacte positie van `\mf` is dus niet zo belangrijk, we hadden ook kunnen zeggen:

$$\text{\bf } x^2 + x^2 + x^2 + x^2 + x^2 + x^2 = \text{\mf } 6x^2\$$$

Een ander aspect van fonts in wiskundige mode betreft gereserveerde namen als `sin` en `cos`.

$$\text{\bf } x^2 + \text{\hbox{whatever}} + \text{\sin}(2x)\$$$

In tegenstelling tot plain \TeX wordt hier de **sin** ook vet gezet.

$$x^2 + \mathbf{whatever} + \mathbf{sin}(2x)$$

De 12 punts wiskundige (Computer Modern) fonts zijn gedefinieerd met:

```
\definebodyfont [12pt] [mm]
  [ex=cmex10 at 12pt,
   mi=cmmi12,
   sy=cmsy10 at 12pt]
```

Het is mogelijk binnen math mode een andere `tf`, `bf` enz. te gebruiken.

```
\definebodyfont [10pt,11pt,12pt] [mm]
  [tf=Sans          sa 1,
   bf=SansBold     sa 1,
```

```
sl=SansItalic    sa 1,
ex=MathExtension sa 1,
mi=MathItalic    sa 1,
sy=MathSymbol    sa 1]
```

```
\setupbodyfont
```

Het eerdere voorbeeld wordt dan:

$x^2 + \mathbf{whatever} + \mathbf{sin}(2x)$

5.10 Em en Ex

Bij het opgeven van maten maken we onderscheid tussen fysieke eenheden, zoals pt en cm en de interne eenheden em en ex. De laatstgenoemden zijn gerelateerd aan het actuele lettertype. Gebruik van interne eenheden voorkomt vaak veel rekenwerk, omdat \TeX als het ware zelf vaststelt hoe breed of hoog iets moet zijn. Enig inzicht in deze maten kan echter geen kwaad. Zo komt een em niet overeen met de breedte van een M, maar van een — (een em-dash). Als dit karakter niet beschikbaar is, geldt een andere waarde. Tabel 5.2 toont enkele voorbeelden. We zien dat de breedte van een cijfer ongeveer .5em is. In Computer Modern Roman is de breedte exact een halve em.

<code>\tf</code>	<code>\bf</code>	<code>\sl</code>	<code>\tt</code>	<code>\ss</code>	<code>\tfx</code>
12	12	12	12	12	12
M	M	M	M	M	M
—	—	—	---	—	—

Tabel 5.2 De breedte van een em.

Waar de em meestal in de breedte wordt gebruikt, gebruiken we ex in de hoogte. Tabel 5.3 toont enkele voorbeelden. We zien dat 1ex redelijk overeenkomt met de hoogte van een x, in dit geval dus geen kapitaal.

<code>\tf</code>	<code>\bf</code>	<code>\sl</code>	<code>\tt</code>	<code>\ss</code>	<code>\tfx</code>
==x	==x	==x	==x	==x	==x

Tabel 5.3 De hoogte van een ex.

5.11 Definities

Deze paragraaf is alleen bedoeld voor nieuwsgierige gebruikers of gebruikers die willen experimenteren met het instellen van korpsen en lettertypes. Een aantal zaken worden hier niet

besproken, zoals de precieze definities van accenten en encodings. Hiervoor verwijzen we naar de voorbeelden in de broncode en de files font-xxx en enco-xxx.

We hebben reeds gezien dat binnen een gekozen korps allerlei kleinere en grotere afmetingen voorkomen. Deze relaties zijn vastgelegd met:

```
\definieerkorpsomgeving
[12pt]
[      text=12pt,      Wiskunde maten: normale afmetingen,
  script=9pt,         super- en subscripts en
  scriptscript=7pt,   supersuper- en subsubscripts.
      x=10pt,         Pseudo kapitalen en
      xx=8pt,         geneste pseudo kapitalen
  groot=12pt,         Voor geval we overgaan op groot
  klein=10pt]        of klein.
```

Wanneer we om een korpsgrootte vragen die niet op deze manier is voorgedefinieerd, dan gelden verhoudingen die overeenkomen met de bovenstaande. Men kan deze overigens aanpassen door in plaats van bijvoorbeeld een korpsgrootte het trefwoord `default` mee te geven. Als men 'even snel' een groot korps wil definiëren, dan kan worden volstaan met:

```
\definieerkorpsomgeving [24pt]
```

In alle gevallen waarin de gebruiker niets heeft gedefinieerd, wordt teruggevallen op redelijke default waarden. Zo is het mogelijk om te schakelen naar een 12.4 korps(omgeving), zonder dat daar bijzondere voorinstellingen voor nodig zijn. Wanneer dit binnen een groep gebeurt, dan hebben de instellingen een tijdelijk karakter. Wordt een korps vaker gebruikt, dan is het verstandig dit te definiëren aan het begin van de file.

Een overzicht van de verschillende samenhangende formaten binnen een familie kan worden opgevraagd met:

```
\toonkorpsomgeving[...,...,..]
...      zie p 100: \stelkorpsin
```

Voor `\br` levert dit commando het volgende overzicht op:

[1br]							
text	script	scriptscript	x	xx	klein	groot	interlinie
20.7pt	14.4pt	12pt	17.3pt	14.4pt	17.3pt	20.7pt	
17.3pt	12pt	10pt	14.4pt	12pt	14.4pt	20.7pt	
14.4pt	11pt	9pt	12pt	10pt	12pt	17.3pt	
12pt	9pt	7pt	10pt	8pt	10pt	14.4pt	
11pt	8pt	6pt	9pt	7pt	9pt	12pt	
10pt	7pt	5pt	8pt	6pt	8pt	12pt	
9pt	7pt	5pt	7pt	5pt	7pt	11pt	
8pt	6pt	5pt	6pt	5pt	6pt	10pt	
7pt	6pt	5pt	6pt	5pt	5pt	9pt	
6pt	5pt	5pt	5pt	5pt	5pt	8pt	
5pt	5pt	5pt	5pt	5pt	5pt	7pt	
4pt	4pt	4pt	4pt	4pt	4pt	6pt	

Voor alle gangbare formaten zijn omgevingen gedefinieerd die in de praktijk goed voldoen. Een beginnend gebruiker zal hier dus niets mee van doen hebben. Moch men toch wat willen aanpassen, dan is er:

```
\stelkorpssomgevingin[...] [.,.,.,.=.,.,.]
... zie p 100: \stelkorsin
..=.. zie p 100: \stelkorsin
```

Het eigenlijke definiëren, dat wil zeggen, het koppelen van commando's aan font files, kan op verschillende manieren gebeuren. Het meeste inzicht verkrijgt men door te kijken naar een file als font-phv.

```
\definefontsynonym [Sans] [Helvetica]
\definefontsynonym [SansBold] [Helvetica-Bold]
\definefontsynonym [SansItalic] [Helvetica-Oblique]
\definefontsynonym [SansSlanted] [Helvetica-Oblique]
\definefontsynonym [SansBoldItalic] [Helvetica-BoldOblique]
\definefontsynonym [SansBoldSlanted] [Helvetica-BoldOblique]
\definefontsynonym [SansCaps] [Helvetica]
\definebodyfont [14.4pt,12pt,11pt,10pt,9pt,8pt,7pt,6pt,5pt] [ss] [default]
```

Met `\definefontsynonym` koppelen we een logische aanduiding, zoals `SansBold` aan een font naam, zoals `Helvetica-Bold`. De eigenlijke koppeling aan een filenaam vindt elders plaats, standaard in file `font-fil` als:

```
\definefontsynonym [Helvetica-Bold] [hvb] [encoding=texansi]
```

Dit is eigenlijk de enige plek waar een systeemafhankelijke instelling plaatsvindt. Als we onder het Karl Berry regime werken, dan ligt de volgende instelling meer voor de hand (zie `font-ber`).

```
\definefontsynonym [Helvetica-Bold] [phvb] [encoding=ec]
```

Het op die manier aan elkaar koppelen van fonts kent nauwelijks grenzen. Men mag zo vaak afbeelden als nodig. Het is leerzaam eens in font-unk te kijken, waar de verschillende stijlen zo worden afgebeeld dat ze door elkaar te gebruiken zijn.

```
\definefontsynonym [Regular] [Serif]
\definefontsynonym [Roman] [Serif]
```

We zien dat de basisaanduiding `Serif` is. De default serif fonts zijn gedefinieerd met:

```
\definebodyfont [default] [rm]
[ tf=Serif      sa 1,
  tfa=Serif     sa a,
  ...
  sl=SerifSlanted sa 1,
  sla=SerifSlanted sa a,
  ...]
```

Wat hier gebeurt is het volgende. We zagen reeds dat `\tf` het standaard lettertype is. Hier wordt `\tf` gedefinieerd als `Serif sa 1` wat betekent dat het een serif font is, geschaald op de natuurlijke korpsgrootte. Die `Serif` is elders geprojecteerd op bijvoorbeeld `LucidaBright` die op haar beurt wordt afgebeeld op de filenaam `lbr`.

Dit soort in-een-klap definities, met gebruik van `default`, maken het mogelijk snel en comfortabel een korps te definiëren. We gaan daarbij volledig voorbij aan het gegeven dat fonts een ontwerp-grootte hebben, en het geval wil dat \TeX fonts die hebben. Aangezien wij, net als de meeste \TeX gebruikers, begonnen zijn met de \TeX fonts, biedt $\text{CON}\TeX\text{T}$ vanzelfsprekend de mogelijkheid tot zeer nauwkeurige definities. Daar wordt dan ook van gebruik gemaakt in de file `font-cmr`:

```
\definebodyfont [12pt] [rm]
[ tf=cmr12,
  tfa=cmr12 scaled \magstep1,
  tfb=cmr12 scaled \magstep2,
  tfc=cmr12 scaled \magstep3,
  tfd=cmr12 scaled \magstep4,
  bf=cmbx12,
  it=cmti12,
  sl=cmsl12,
  bi=cmbxti10 at 12pt,
  bs=cmbxsl10 at 12pt,
  sc=cmcsc10 at 12pt]
```

We gebruiken hier de standaard \TeX -specificaties `scaled` en `at`, maar we zagen reeds dat $\text{CON}\TeX\text{T}$ als aanvulling een combinatie van beiden biedt: `sa` (`scaled at`). Als we geen gebruik maken van de default definitie, luidt de definitie van de Helvetica bijvoorbeeld:

```
\definebodyfont [12pt,11pt,10pt,9pt,8pt] [ss]
[tf=hv sa 1.000,
 bf=hvb sa 1.000,
```

```

it=hvo sa 1.000,
sl=hvo sa 1.000,
tfa=hv sa 1.200,
tfb=hv sa 1.440,
tfc=hv sa 1.728,
tfd=hv sa 2.074,
sc=hv sa 1.000]

```

We geven in dit geval de schaalwaarde ten opzichte van de korpsgrootte op. Analooq aan \TeX 's `\magstep` kunnen we hier `\magfactor` gebruiken: in plaats van `sa 1.440` kan dus `sa \magfactor2` worden opgegeven. Omdat dit soort getallen niet alleen vervelend is, maar ook onnodig geheugen gebruikt, mag men `1.200` vervangen door `a`, enz. Deze relatie is in te stellen in de korpsomgeving.

```

\definebodyfont [12pt,11pt,10pt,9pt,8pt] [ss]
  [tf=hv sa 1, tfa=hv sa a, tfb=hv sa b, tfc=hv sa c, tfd=hv sa d]

```

Als font files in alle interfaces worden gebruikt, gebruiken we de engelse commando's. De definities vinden plaats in files met de naam `font-???.tex`, zie bijvoorbeeld de file `font-cmr.tex`.

```

\definieerkorps[...1...][.2.][...=... ]
.1.    5pt ... 12pt default
.2.    rm ss tt mm hw cg
tf      file
bf      file
sl      file
it      file
bs      file
bi      file
sc      file
ex      file
mi      file
sy      file
ma      file
mb      file
mc      file

```

De instellingen `ex`, `mi`, `sy`, `ms`, `mb` en `mc` hebben betrekking op wiskundige karaktersets. De eerste drie vinden we ook in Plain \TeX , de laatste drie zijn nodig bij andere families. Zo zijn de binnen $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ te gebruiken letters en symbolen ook binnen $\text{CON}\text{\TeX}\text{T}$ te gebruiken: `\definieerkorps[ams]`. Deze zijn ondergebracht in `ma` en `mb`.

Men is overigens niet gebonden aan a-d. Bij wijze van voorbeeld definiëren we een grote variant van `\tf`:

```

\definieerkorps [10pt,11pt,12pt] [rm] [tfe=ComputerModern at 48pt]
\tfe Grote Woorden.

```

Deze ziet er als volgt uit:

Grote Woorden.

Nu we toch bezig zijn met definiëren, men kan ook onafhankelijk van het besproken mechanisme direct een font definiëren.

```
\definieerkorps[KopLetter][Regular sa 1.2]
```

Hierna kan men `\KopLetter` gebruiken om van font te wisselen. Zo nodig moet men de spatiëring (interlinie) instellen met `\stelinterliniein`, dus:

```
\KopLetter \stelinterliniein tekst \par
```

Voor gevorderde T_EX-gebruikers is het dimensie-register `\korpsgrootte` beschikbaar. Deze variabele kan worden gebruikt om breedtes in te stellen. Het aantal (afgeronde) punten is beschikbaar in `\korpspunten`.

Tot nu toe zijn we er van uitgegaan dat een `a` te voorschijn komt als een `a`. Dit is geen vanzelfsprekendheid in geval van bijvoorbeeld een `ä` of `æ`. Dit karakter is namelijk niet in ieder font aanwezig, zeker niet in de Computer Modern Typefaces. Vaak zal een combinatie van letters, zoals `\"a`, of een commando `\ae` gebruikt worden om zo'n karakter te zetten. In een aantal gevallen zal T_EX uit zichzelf karakters combineren, zoals `f1` tot `fl` en niet `fl`. Een ander probleem vormt het omzetten van hoofd- in kleine letters en omgekeerd. Om met dit laatste te beginnen, hier is een voorbeeld van de `texnansi` mapping:

```
\startmapping[texnansi]
  \definecasemap 228 228 196  \definecasemap 196 228 196
  \definecasemap 235 235 203  \definecasemap 203 235 203
  \definecasemap 239 239 207  \definecasemap 207 239 207
  \definecasemap 246 246 214  \definecasemap 214 246 214
  \definecasemap 252 252 220  \definecasemap 220 252 220
  \definecasemap 255 255 159  \definecasemap 159 255 159
\stopmapping
```

Dit betekent zoveel als: karakter met code 228 wordt in geval van hoofdletters karakter 228 en wordt in kleine letters karakter 196.

Deze definities zijn te vinden in `enco-ans`. Verder vinden we in die file:

```
\startencoding[texnansi]
  \defineaccent " a 228
  \defineaccent " e 235
  \defineaccent " i 239
  \defineaccent " o 246
  \defineaccent " u 252
  \defineaccent " y 255
\stopencoding
```

en wat verderop:

```

\startencoding[texnansi]
  \definecharacter ae 230
  \definecharacter oe 156
  \definecharacter o 248
  \definecharacter AE 198
\stopencoding

```

Zoals gezegd, accenten vormen (niet alleen in T_EX) een geval apart. Dit is mede een gevolg van de wijze waarop accenten worden geplaatst. Er zijn grofweg twee methoden: òf T_EX plaatst de accenten zelf òf er worden karakters gebruikt waarop de accenten al staan. De bovenstaande definities zorgen ervoor dat dit alles goed gaat. Overigens gaan aan dergelijke definities soms andere vooraf, zie daarvoor de file `enco-ini`.

We keren nog even terug naar de fontdefinities. Het vlot wisselen van stijl kan ook met commando's als `\xii` of `\twelvepoint`, waarmee we voortbouwen op plain T_EX. Dergelijke commando's worden gedefinieerd met:

```

\definieerfontsynoniem [twelvepoint] [12pt]
\definieerfontsynoniem [xii]          [12pt]

```

De trefwoorden bij `\stelkorpsin` zijn gedefinieerd in de trant van:

```

\definieeralgemenestijl [rm,romaan,serif,regular] [rm]
\definieeralgemenestijl [ss,schreefloos,sans,support] [ss]
\definieeralgemenestijl [tt,teletype,type,mono] [tt]
\definieeralgemenestijl [hw,handschrift] [hw]
\definieeralgemenestijl [cg,calligrafie] [cg]

```

Bij veel elders beschreven instellingen komen we `letter` of `kopletter` tegen. In die situaties kunnen we een aanduiding meegeven. Met `\definieerletter` zijn deze aanduidingen vastgelegd. Het derde argument betreft de betekenis in titels van hoofdstukken, paragrafen enz. Alleen `\kap` heeft daar betekenis.

```

\definieerletter [normaal]           [\tf] []
\definieerletter [vet]               [\bf] []
\definieerletter [type]              [\tt] []
\definieerletter [italic]            [\it] []
\definieerletter [schuin]            [\sl] []
\definieerletter [vetitalic,italicvet] [\bs] []
\definieerletter [vetschuin,schuinvet] [\bs] []
\definieerletter [klein,kleinnormaal] [\tfx] []

```

We hebben in paragraaf 5.6 al aangegeven hoe benadrukken kan worden ingesteld. Met `oldstyle` getallen ligt dat wat anders. Er is op voorhand niet aan te geven waar die te vinden zijn. Standaard is de instelling:

```

\definefontsynonym [OldStyle] [MathItalic]
\definefontsynonym [OldStyle] [MathItalic]

```

ofwel, ze worden uit hetzelfde font gehaald als de wiskundige italic karakters.

In deze paragraaf zagen we een mengelmoes aan engelse en nederlandse commando's. We raden aan in geval van een internationale stijldefinitie, de engelse commando's te gebruiken. Veel van de voorbeelden zijn ontleend aan files die deel uitmaken van de CONTEX_T distributie, en zijn derhalve achter de schermen engels.

Er zijn naast de hier besproken commando's nog andere beschikbaar, zoals macro's om accenten te manipuleren. Deze worden in de file font-ini besproken. Ook kan men nadere informatie vinden in core-fnt en allerlei geintjes in supp-fun. Genoeg voer dus voor liefhebbers!

5.12 Omhullende tekst

We maken onderscheid tussen lopende en omhullende tekst. Onder de omhullende tekst verstaan we hoofd- en voetregels en interactieve elementen, zoals menu's. Deze zin maakt deel uit van de lopende tekst. Al het voorgaande had voornamelijk betrekking op de lopende tekst. De lettertypen van de omhullende tekst worden ingesteld met verschillende commando's. Meestal zal dit in termen van `letter=vet` gebeuren, maar instellingen als `letter=\ss\bf` zijn ook toegestaan. Instellingen als `letter=\ssbf` liggen minder voor de hand, omdat in dat geval `\kap` en dergelijke niet correct werken.

Het wisselen van stijl (`\ss`) kost tijd. Meestal is dit geen probleem, maar wanneer we bijvoorbeeld interactieve menu's gebruiken met tientallen items, dan heeft het wisselen merkbaar invloed. In dat geval is een meer efficiënte wijze van wisselen mogelijk:

```
\stellayoutin[letter=\ss]
```

Aanvullende instellingen vinden vervolgens plaats met de betreffende commando's en de parameter `letter`, bijvoorbeeld:

```
\stelvoetin[letter=vet]
```

Overigens gelden voor de omhullende tekst altijd de instellingen van `\stelkorpsin`, ook als de lopende tekst daarvan afwijkt.

5.13 Files

In tabel 5.4 zijn enkele van de standaard meegeleverde font definitie files opgenomen. Deze worden dus aangeroepen op de drie laatste letters.

De meest gebruikte encoding vectoren zijn standaard geladen, zoals `ans`, `ec` en `i12`. Men kan een extra encoding laden met `\useencoding`, maar dat is zelden nodig. De twee in tabel 5.5 laatst genoemde files hebben betrekking op het direct ondersteunen van andere dan de standaard toetsenbord karakters. Deze moeten expliciet worden geladen.

5.14 Figuren

Tot slot nog dit. Als in een tekst figuren worden opgenomen, dan ontkomt men er vaak niet aan in die figuren tekst op te nemen. Meestal zijn daarbij de T_EX-fonts niet beschikbaar. Als in de tekst zelf een serif wordt gebruikt, kan bijvoorbeeld in een figuur uitstekend een

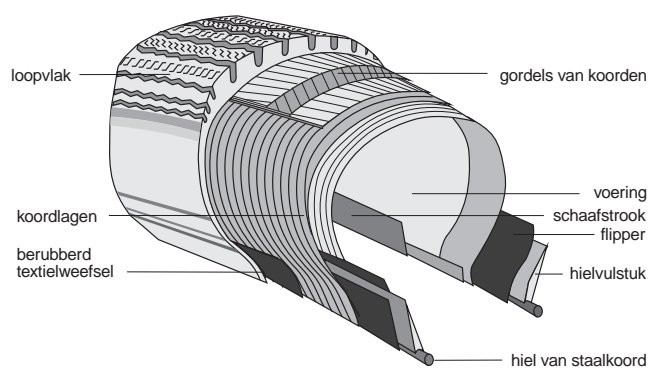
Helvetica worden gebruikt. In figuur 5.1 is bijvoorbeeld gebruik gemaakt van een Helvetica. In het onderschrift gebruiken we Knuth's Sans Serif.

font-cmr	Computer Modern Roman
font-csr	Computer Slavik Roman (?)
font-con	Concrete Roman
font-eul	Euler
font-ams	American Mathematics Society
font-ant	Antykwa Torunska
font-lbr	Lucida Bright
font-pos	Base PostScript Fonts
font-ptm	Times Roman
font-phv	Helvetica
font-pcr	Courier
font-fil	Standard Filenames
font-ber	Karl Berry FileNames

Tabel 5.4 Enkele standaard meegeleverde font definitie files (pos = ptm + phv + pcr).

enco-ans	TeXnansi
enco-ec	European Computer
enco-il2	ISO Latin 2
enco-plr	Polish Roman
enco-ibm	default IBM PC code page
enco-win	default MS Windows code page

Tabel 5.5 Enkele standaard meegeleverde encoding definitie files.



Figuur 5.1 Het gebruik van lettertypes in figuren.

6

Kleur en achtergrond

Kleur en achtergrond

6.1 Inleiding

Bij het vormgeven van een document kan gebruik worden gemaakt van kleur. Voor de eigenlijke tekst is het zetten in kleur in de regel niet aan te bevelen, maar in allerhande ornamenten kan het een ietwat saai document aardig opvrolijken. Bovendien kan in voor het scherm ontworpen documenten kleur de functie overnemen van aan het papier gerelateerde kenmerken. Zo kan kleur worden gebruikt om delen van een tekst af te bakenen.

In dit hoofdstuk staan we stil bij de wijze waarop binnen `CONTEXT` kleur wordt ondersteund. We besteden daarbij ook aandacht aan achtergronden en overlays omdat we zullen zien dat deze aan kleur zijn gerelateerd.

6.2 Kleur

Een van de problemen bij het zetten in kleur is dat verschillende kleuren gelijke grijswaarden kunnen opleveren. We hebben dit indertijd vrij redelijk onderzocht en we zullen dan ook zien dat in `CONTEXT` voorzieningen zijn getroffen om consistent kleurgebruik af te dwingen. Omdat kleurgebruik doordacht moet plaatsvinden, moet men kleur eerst activeren:

```
\stelkleurenin[status=start]
```

Er zijn echter meer instellingen mogelijk:

```
\stelkleurenin[...=...]
```

status	start <u>stop</u> globaal lokaal
conversie	<u>ja</u> nee altijd
reductie	ja <u>nee</u>
rgb	<u>ja</u> nee
cmyk	<u>ja</u> nee
mpcmyk	<u>ja</u> nee

Aan `status` kan naast `start` ook `lokaal` of `globaal` worden toegekend. Bij het passeren van de paginagrens wordt in geval van `start` of `globaal` bijgehouden welke kleur actueel is. Als we, wat in een interactieve tekst vaak het geval is, nooit gekleurd de paginagrens passeren, dan kunnen we de instelling `lokaal` gebruiken. Dit scheelt de nodige overhead en kan, afhankelijk van het kleurgebruik, snelheidswinst opleveren. Dit soort fijnregeling is echter in de praktijk zelden nodig.

Standaard worden zowel RGB als CMYK kleurruimten ondersteund. Zet men echter `cmyk` op `nee`, dan worden CMYK kleurspecificaties automatisch omgezet in RGB. Hetzelfde geldt wanneer `rgb=nee`. Wanneer geen kleur is toegestaan, vindt automatisch omzetting in gewogen grijswaarden plaats. Men kan dit omzetten instellen met `conversie`. Stellen we deze parameter in op `altijd`, dan worden alle kleuren omgezet in een directe grijswaarde, terwijl in geval van `ja` alleen grijze kleuren worden omgezet.

Kleuren moeten eerst worden gedefinieerd. Dit gebeurt in een file met de naam `colo-xxx.tex`, waarna de kleuren kunnen worden opgeroepen onder de naam `xxx`. Standaard is de file `colo-rgb.tex` geladen, waarin definities te vinden zijn als:

```
\definieerkleur [donkerrood] [r=.5, g=.0, b=.0]
\definieerkleur [donkergroen] [r=.0, g=.5, b=.0]
.....
```

Een file met kleurdefinities wordt geladen met (let op het enkelvoud):

```
\stelkleurin[rgb]
```

Kleuren worden geactiveerd zoals hieronder is weergegeven:

```
\startkleur[donkergroen]
```

We kunnen zoveel kleuren gebruiken als we maar willen. We zullen er echter rekening mee moeten houden dat de lezer mogelijk

```
\kleur[donkerrood]{kleurenblind} is. Bovendien is het maar de vraag
of kleurgebruik in de lopende tekst wel zo fraai is.
```

```
\stopkleur
```

Men kan op deze manier ook CMYK kleuren en grijswaarden definiëren:

```
\definieerkleur [cyaan] [c=1,m=0,y=0,k=0]
\definieerkleur [grijs] [s=0.75]
```

We hadden grijs ook kunnen definiëren als:

```
\definieerkleur [grijs] [r=0.75,g=0.75,b=0.75]
```

Wanneer in de instellingen conversie op ja staat, dan worden dergelijke definities automatisch teruggebracht tot de s-vorm: `[s=.75]`. De s staat voor 'screen' ofwel raster. Als reductie op ja staat, dan worden overeenkomstige componenten in een CMYK specificatie vertaald in een k component.

Een aardige eigenschap van het definitie mechanisme is dat men kleuren kan erven:

```
\definieerkleur [opvallend] [rood]
```

Dergelijke definities maken een meer logisch gebruik van kleur mogelijk. Bovendien kan men nu eenvoudig alle **opvallende** zaken een andere kleur geven.

Naast `\stelkleurenin` hebben we dus de volgende commando's tot onze beschikking:

```
\definieerkleur[...] [.,.,.=.,.,.]
...      naam
r        tekst
g        tekst
b        tekst
c        tekst
m        tekst
y        tekst
k        tekst
s        tekst
```


Een kleur-definitie-file laden we met:

```
\stelkleurin[...]
... naam
```

In kleur zetten gebeurt met:

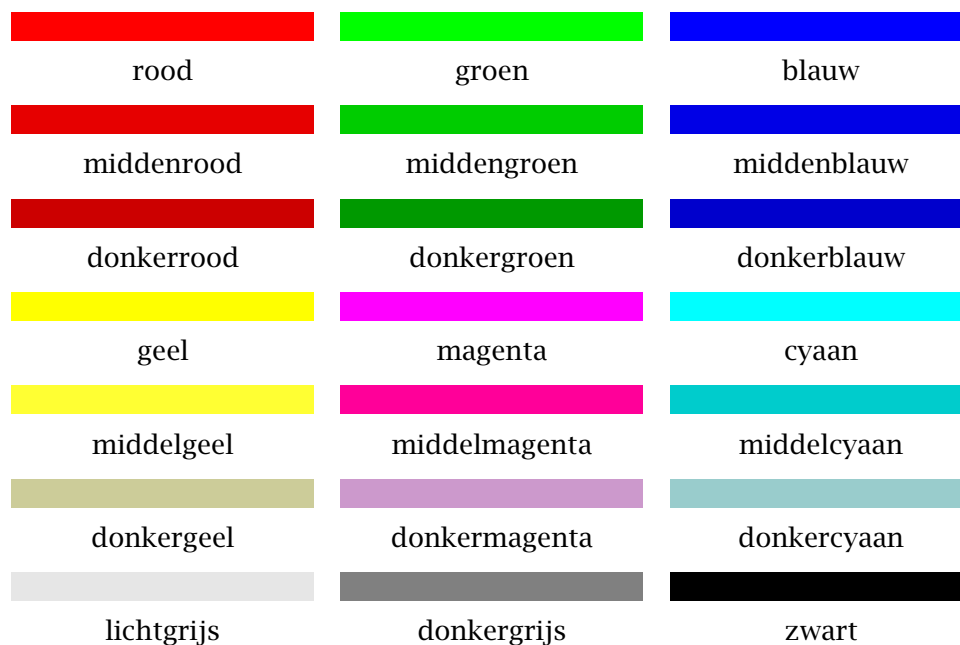
```
\kleur[.1.]{.2.}
.1. tekst
```

```
\startkleur[...] ... \stopkleur
... naam
... tekst
```

Een volledig overzicht van kleuren kan worden gegenereerd met:

```
\toonkleur[...]
... naam
```

In figuur 6.1 zijn een aantal kleuren weergegeven die standaard beschikbaar zijn (zie `color-rgb.tex`).



Figuur 6.1 Enkele voorbeelden van kleuren.

Kleurgebruik is binnen \TeX geen triviale zaak, omdat \TeX geen kleurgebruik ondersteunt. Er wordt in ons geval gebruik gemaakt van de \TeX -commando's `\special` en `\mark` om POST-

SCRIPT of PDF aan te sturen. Aan de meeste gebruikers zullen de beperkingen die aan deze methode kleven voorbijgaan.

Er kan met kleuren over de paginagrens worden gewerkt. Daarbij blijven, als alles tenminste naar behoren werkt, de hoofd- en voetregels en verplaatste figuren in de juiste kleur staan. In hoeverre het onderliggende mechanisme robuust is, blijft een vraag.

Zoals aan deze zin te zien is, kan bij gebruik van kleuren worden **genest**. Het is echter nog maar de vraag of dit esthetisch verantwoord is.

Zodra een kleur wordt gedefinieerd, komt zij ook als commando beschikbaar. Er bestaat dus een commando `\donkerrood`. Dergelijke commando's gehoorzamen groeperen. We kunnen dus zeggen `{\donkerrood dit is gezet in donkerrood}`.

Er zijn nogal wat commando's die als parameter `kleur` hebben. In veel gevallen geldt dat, ook als dat niet wordt besproken, `kleur` is in te stellen. Dat geldt vrijwel zeker als ook het lettertype instelbaar is.

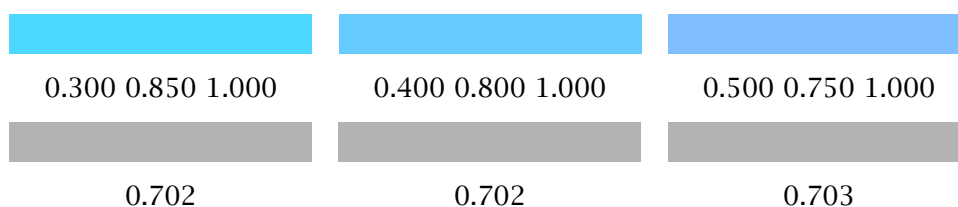
De standaard kleur-instellingen zijn:

```
\stelkleurenin [conversie=ja, reductie=nee, rgb=ja, cmyk=ja]
```

Dit betekent dat beide kleursystemen worden ondersteund, en dat de *k*-component in CMYK kleuren worden gehandhaafd. Als `reductie=ja`, dan wordt de *k*-component 'weggewerkt'. Met `conversie=ja` bepaalt men dat bij gelijke kleurcomponenten, rasters worden gebruikt, wat in sommige gevallen een meer efficiënte en betrouwbare uitvoer oplevert.

6.3 Grijswaarden

Als we een gekleurde tekst op een zwart-wit printer afdrucken, blijkt dat het onderscheid tussen sommige kleuren wegvalt. Figuur 6.2 illustreert dit.



Figuur 6.2 Drie varianten cyaan met gelijke grijswaarden.

Voor de zwart-wit-kijker zien alle blokjes er hetzelfde uit. Toch zijn de drie bovenste blokjes in verschillende kleuren (cyaan) gezet. De onderste blokjes simuleren de grijswaarden. Hierbij is gebruik gemaakt van de volgende conversieformule:

$$\text{grijs} = .30 \times \text{rood} + .59 \times \text{groen} + .11 \times \text{blauw}$$

Een kleur kan in grijs worden weergegeven met het commando

```
\grijskleur[...]  
... tekst
```

De actuele getalswaarden van een kleur zijn zo nodig op te roepen met de commando's `\kleurwaarde{naam}` en `\grijswaarde{naam}`.

We kunnen automatisch alle gebruikte kleuren in gewogen grijswaarden omzetten:

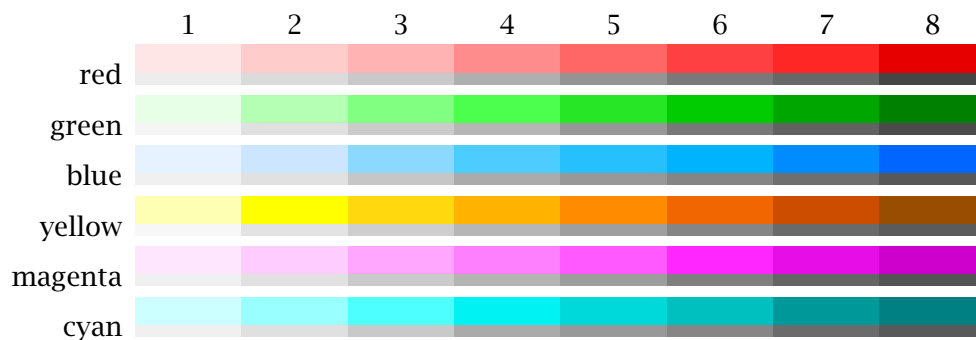
```
\stelkleurenin [conversie=altijd]
```

6.4 Kleurgroepen en paletten

\TeX zelf heeft nauwelijks grafische mogelijkheden in zich. Desalniettemin is het binnen $\text{CON}\text{T}\text{E}\text{X}\text{T}$ beschikbare mechanisme om in kleur te zetten in belangrijke mate bepaald door kleurgebruik in figuren. Een van de problemen waar men mee te maken krijgt is namelijk het feit dat een figuur die in kleur fraai oogt, er in zwart-wit beroerd kan uitzien. We bespreken dit verschijnsel in de vorige paragraaf.

Omdat \TeX nu eenmaal meer mogelijkheden biedt dan tekenpakketten om dit soort problemen systematisch te lijf te gaan, is een kleursysteem ontwikkeld dat zowel in tekst als in grafische pakketten kan worden gebruikt.

We onderscheiden naast de reeds besproken individuele kleuren ook kleurgroepen. Een kleurgroep bevat een aantal gradaties van een kleur. Standaard zijn de volgende kleurgroepen gedefinieerd:



De verschillende gradaties binnen een kleur worden weergegeven door een getal. Een kleurgroep wordt gedefinieerd met:

```
\definieerkleurgroep[.1.][.2.][x:y:z=, ..]
.1.   naam
.2.   rgb cmyk grijs s
```

Een voorbeeld van een deel van een RGB definitie is:

```
\definieerkleurgroep
[blauw][rgb]
[1.00:1.00:1.00,
 0.90:0.90:1.00,
 .....,
```

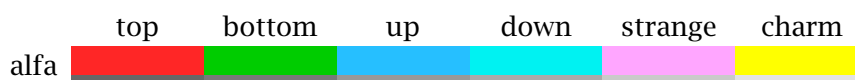
```
0.40:0.40:1.00,
0.30:0.30:1.00]
```

De [rgb] is hier overigens niet verplicht, omdat standaard RGB wordt verwacht. Dit commando kan, afgezien van wat er achter de schermen plaatsvindt, worden beschouwd als een reeks definities van kleuren:

```
\definieerkleur [blauw:1] [r=1.00, g=1.00, b=1.00]
\definieerkleur [blauw:2] [r=0.90, g=0.90, b=1.00]
.....
\definieerkleur [blauw:7] [r=0.40, g=0.40, b=1.00]
\definieerkleur [blauw:8] [r=0.30, g=0.30, b=1.00]
```

Een in een kleurgroep gedefinieerde kleur is dan ook op te roepen met *naam:nummer*, bijvoorbeeld: blauw:4.

Hoewel er geen maximum is aan het aantal gradaties binnen een kleurgroep, blijkt in de praktijk dat men niet verder moet gaan dan 6 tot 8 gradaties. Dit wordt mede ingegeven door het volgende. Naast kleurgroepen onderscheiden we paletten. Een palet bestaat uit een beperkt aantal logische kleuren. Logisch wil zeggen dat we de kleur met een naam aanduiden. Een voorbeeld van een palet is:



De idee achter paletten is dat vermeden moet worden dat in een tekst of figuur kleuren worden gebruikt die in zwart-wit niet te onderscheiden zijn. Een palet wordt dan ook als volgt gedefinieerd:

```
\definieerpalet
  [voorbeeld]
  [strange=rood:3,
   top=groen:1,
   .....
   bottom=geel:8]
```

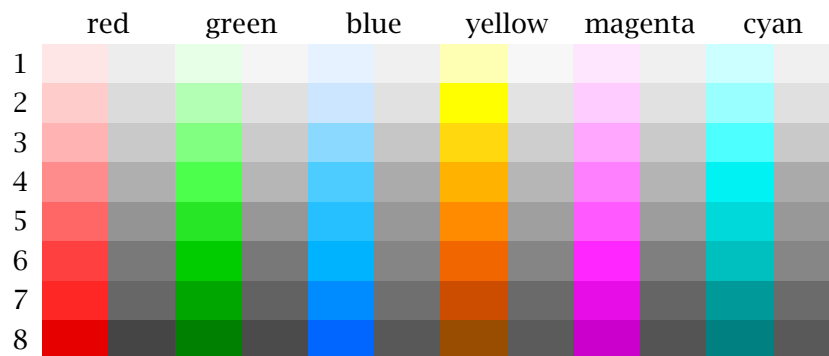
We definiëren een palet dus met het commando:

```
\definieerpalet[...][...]=...
...      naam
naam     naam
```

Binnen `CONTEXT` zijn enkele paletten voorgedefinieerd. Daarbij worden binnen een palet de wat abstracte namen van quarks gebruikt: *top*, *bottom*, *up*, *down*, *strange* en *charm*. Bij gebrek aan meer namen hebben we daarnaast nog *friend* en *rude*. Let wel, dit zijn slechts voorbeelden, gedefinieerd in the RGB definitie-file, gebaseerd op onze eigen experimenten.

De idee achter het systeem van kleurgroepen en paletten is dat als we een palet samenstellen uit elementen van een kleurgroep met verschillende nummers, we in zwart-wit weergave goed

te onderscheiden kleuren houden. Daarbij moet wel aan de randvoorwaarde zijn voldaan dat vergelijkbare nummers in verschillende groepen een vergelijkbare grijswaarde hebben.



Hebben we eenmaal een palet gedefinieerd, dan kunnen we gebruik maken van het commando:

```
\stelpaletin[...]  
... naam
```

Vervolgens kunnen we de kleuren van het gekozen palet op naam oproepen, bijvoorbeeld `\kleur[strange]{is dit niet wat vreemd}`.

Een voorbeeld van het gebruik van paletten is te vinden in het verbatim zetten van \TeX code. Binnen dit mechanisme worden kleuren gebruikt met namen als `prettyone`, `prettytwo`, enz. Er zijn twee reeksen gedefinieerd, kleur en grijs, in termen van:

```
\definieerkleur [colorprettyone] [r=.9, g=.0, b=.0]  
\definieerkleur [grayprettyone] [s=.3]
```

Deze reeksen zijn gecombineerd tot een palet met:

```
\definieerpalet  
[colorpretty]  
[ prettyone=colorprettyone, prettytwo=colorprettytwo,  
prettythree=colorprettythree, prettyfour=colorprettyfour]  
  
\definieerpalet  
[graypretty]  
[ prettyone=grayprettyone, prettytwo=grayprettytwo,  
prettythree=grayprettythree, prettyfour=grayprettyfour]
```

We kunnen nu in een klap alle kleuren veranderen door het palet in te stellen:

```
\steltypenin[palet=colorpretty]
```

Overigens kan dat per filter anders worden ingesteld:

```
\definepalet [MPcolorpretty] [colorpretty]  
\definepalet [MPgraypretty] [graypretty]
```

We zien hier overigens dat we net als bij kleuren, paletten kunnen erven. Dit, wellicht niet eenvoudige voorbeeld, toont iets van de achterliggende filosofie: beschouw kleuren als abstracties, groepeer ze zo mogelijk in paletten en wissel een palet in plaats van een hele reeks.

Ten behoeve van het samenstellen van kleurgroepen en paletten zijn enkele commando's beschikbaar om het onderscheidend vermogen te testen. In de overzichten zijn steeds onder de baseline de grijswaarden weergegeven (dit is alleen in kleur goed te zien). Bij de vergelijkingen worden links de kleuren en rechts de grijswaarden vergeleken.

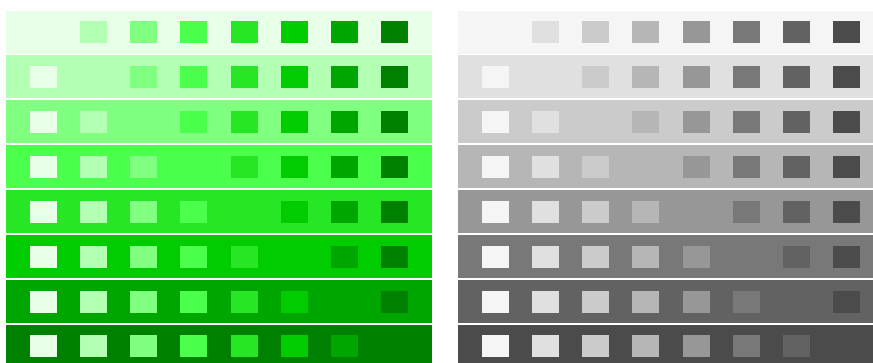
```
\toonkleurgroep[.1.][...2,...]
.1. naam
.2. horizontaal vertikaal naam waarde nummer
```

```
\toonpalet[.1.][...2,...]
.1. naam
.2. horizontaal vertikaal naam waarde
```

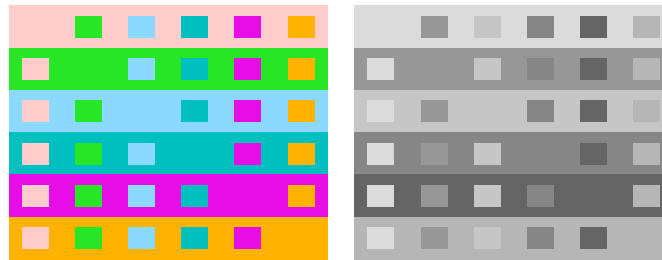
```
\vergelijkkleurgroep[...]
... naam
```

```
\vergelijkpalet[...]
... naam
```

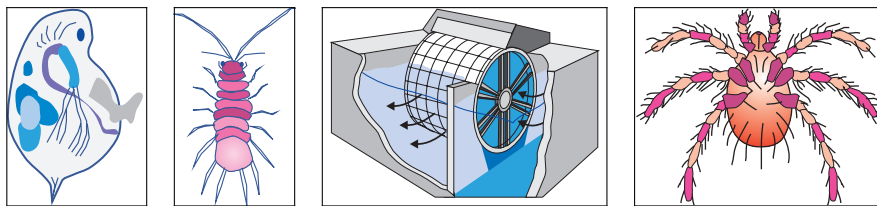
De eerder getoonde overzichten zijn gegenereerd met de eerste twee commando's, de onderstaande overzichten met de laatste twee.



Het bovenstaande overzicht is opgeroepen met `\vergelijkkleurgroep[groen]` en het onderstaande met `\vergelijkpalet[gamma]`.



De standaard beschikbare kleurgroepen en paletten zijn met zorg samengesteld en de gevonden waarden worden binnen onze organisatie systematisch gebruikt bij het inkleuren van figuren. Deze zijn zowel in kleur als in zwart-wit goed weer te geven.



Figuur 6.3 Enkele voorbeelden van het gebruik van kleurcombinaties.

6.5 Tekst-achtergronden

Bij een aantal commando's, bijvoorbeeld `\omlijnd`, kan een achtergrond worden geplaatst. Een achtergrond kan een kleur hebben maar ook grijs zijn (gerasterd). Standaard is daarbij achtergrondraster ingesteld op 0.95. Bruikbare waarden liggen echter tussen 0.70 en 1.00.¹⁵

Het door $\text{T}_{\text{E}}\text{X}$ laten opbouwen van rasters kost erg veel geheugen en kan dan ook leiden tot een foutmelding. Standaard worden daarom rasters extern opgebouwd met behulp van bijvoorbeeld POSTSCRIPT of PDF instructies. Dit is ingesteld met het commando:

```
\stelrastersin[...]=...]
methode      punt lijn extern
resolutie    getal
factor       getal
raster       getal
```

De factor heeft alleen betekenis als de methode `lijn` of `punt` wordt gekozen. De instelling `raster` bepaalt de 'korrel' van het raster. Tekst is over een raster van 0.95 nog goed leesbaar.

Visueel komen de met $\text{T}_{\text{E}}\text{X}$ gemaakte rasters redelijk overeen met de POSTSCRIPT rasters. Als $\text{T}_{\text{E}}\text{X}$ genoeg geheugen heeft en tijd geen rol speelt, zijn de $\text{T}_{\text{E}}\text{X}$ -rasters soms zelfs mooier dan de standaard postscript-rasters. Hoewel in principe meer methoden mogelijk zijn, hebben we voornamelijk alleen de genoemde methoden geïmplementeerd.

¹⁵ Hoewel de achterliggende mechanismen anders zijn, komen grijswaarden overeen met kleuren waarbij de drie componenten dezelfde waarde hebben.

Ook achter lopende tekst kan een achtergrond worden geplaatst. Dit doet men door het betreffende deel van de tekst te omringen met de commando's:

```
\startachtergrond
\stopachtergrond
```

Bij dit stukje tekst is dat dan ook gebeurd. Achtergronden kunnen over meerdere bladzijden lopen. Er wordt extra witruimte (grijsruimte) toegevoegd rond de tekst.

```
\startachtergrond ... \stopachtergrond
```

Zowel de achtergrond als de omringende ruimte kan worden ingesteld met:

```
\stelachtergrondin[...=...]
```

linkeroffset	<i>maat</i>
rechteroffset	<i>maat</i>
bovenoffset	<i>maat</i>
onderoffset	<i>maat</i>
voor	<i>commando</i>
na	<i>commando</i>
status	<u>start</u> stop
..=..	zie p 226: \stelomlijndin

Het commando `\achtergrond` kan worden gebruikt in combinatie met bijvoorbeeld plaatsblokken:

```
\plaatstabel
  {Zomaar een tabel.}
  \achtergrond
  \starttabel[|c|c|c|]
  \HL
  \VL rood \VL groen \VL blauw \VL \AR
  \VL cyaan \VL magenta \VL geel \VL \AR
  \HL
  \stoptabel
```

Het commando `\achtergrond` verwacht een argument. Omdat de tabel 'grouped' is, dat wil zeggen zelf genereert, gaat het bovenstaande goed en zijn geen extra haakjes nodig.

```
\achtergrond
```

Een nogal fundamenteel verschil tussen kleur en rasters, is dat rasters nooit worden geconverteerd. Er is een commando `\startraster` beschikbaar, maar in tegenstelling tot het kleur alternatief, houdt `CONTEX` hier geen rekening met pagina-overgangen. In de regel worden rasters namelijk gebruikt voor objecten met een vaste omvang.

6.6 Layout-achtergronden

Omdat bij interactieve teksten de opbouw cq. indeling van een scherm nogal gecompliceerd kan uitvallen, ligt daar het systematisch gebruik van een achtergrond voor de hand. Het is dan ook mogelijk alle in de layout te benoemen ‘velden’ van een achtergrond te voorzien. Dit gebeurt met het commando (niet te verwarren met `\stelachtergronدين`):

```
\stelachtergrondenin[.1.][...2...][...=...]
```

.1. boven hoofd tekst voet onder pagina papier linkerpagina rechterpagina
 .2. linkerrand linkermarge tekst rechtermarge rechterrand
 status start stop herhaal
 ..=.. zie p 226: \stelomlijndin

In tegenstelling tot de achtergrond van de hele pagina, worden de achtergronden van de 25 layoutvlakken uit oogpunt van efficiëncy, alleen berekend als er iets in de instellingen is veranderd. Wil men bij elke pagina de achtergrond opnieuw genereren, dan dient men aan status de waarde `herhaal` toe te kennen. Ook na simpelweg `stelachtergrondenin`, zonder argumenten worden de achtergronden opnieuw gebruikt.

Een specifiek deel van de layout wordt aangegeven in een soort coördinatenstelsel (zie figuur 6.4).

	linkerrand	linkermarge	tekst	rechtermarge	rechterrand
boven					
hoofd					
tekst					
voet					
onder					

Figuur 6.4 De coördinaten bij `\stelachtergrondenin`.

Er mogen meerdere coördinaten tegelijk worden opgegeven, bijvoorbeeld:

```
\stelachtergrondenin
  [hoofd,tekst,voet]
  [tekst]
  [achtergrond=raster]
```

of

```
\stelachtergrondenin
  [tekst]
  [tekst,rechterrand]
  [achtergrond=kleur,achtergrondkleur=MijnKleur]
```

Een aantal instellingen bij pagina, zoals offset en hoek gelden standaard ook voor de rest van de vakken, bijvoorbeeld:

```
\stelachtergrondenin
  [pagina]
  [offset=.5\korpsgrootte,
   diepte=.5\korpsgrootte]
```

Wanneer in een interactieve tekst menu's worden gebruikt, dan wordt bij het uitlijnen automatisch gecorrigeerd voor de offset en/of diepte zoals hierboven ingesteld. Het is overigens ook mogelijk onder `pagina` de standaard kleuren en rasters in te stellen.

Mocht onverhoopt een aanpassing niet doorkomen, dan kan met `\stelachtergrondenin` (zonder argument) aan `CONTEXT` worden doorgegeven dat een nieuwe achtergrond moet worden berekend.

6.7 Overlays

Eerlijk gezegd zijn de mogelijkheden die `TEX` ons biedt om aardigheidjes in de layout in te bouwen vrij beperkt. In `CONTEXT` is het daarom op vrij veel plaatsen mogelijk 'ergens iets achter te hangen', bijvoorbeeld een in een ander pakket gemaakt beeld, een tekening of wat ornamenten. Omdat er technisch gesproken sprake is van het over elkaar heen leggen, spreken we van 'overlays'.

We zagen reeds dat we bij achtergronden de trefwoorden `kleur` en `raster` konden meegeven. Eigenlijk zijn beiden een voorbeeld van een overlay. Het is daarnaast mogelijk een eigen achtergrond te definiëren:

```
\definieeroverlay[geintje][\groen een groene tekst op de achtergrond]
\omlijnd
  [hoogte=2cm,achtergrond=geintje,uitlijnen=midden]
  {\op\de\voorgond}
```

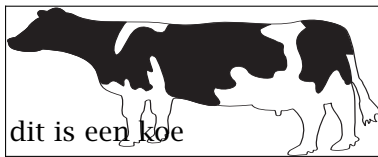
Dit ziet er uit als:



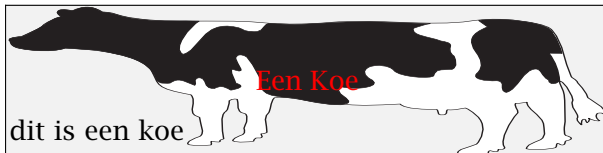
Zo'n overlay kan van alles zijn:

```
\definieeroverlay
  [geintje]
  [{\externfiguur[koe][breedte=\overlaywidth,hoogte=\overlayheight]}]
\omlijnd
  [hoogte=2cm,breedte=5cm,achtergrond=geintje,uitlijnen=rechts]
  {\vfill dit is een koe}
```

We zien dat bij het maken van een overlay de breedte en hoogte beschikbaar zijn in macros. Dit maakt het mogelijk de figuur te schalen.



Het is mogelijk verschillende overlays te combineren, met elkaar, maar ook met een raster of kleur.



Dit ziet er in \TeX definities als volgt uit:

```
\definieeroverlay
  [geintje]
  [{\externfiguur[koe][breedte=\overlaywidth,hoogte=\overlayheight]}]
\definieeroverlay
  [grapje]
  [\rood Een Koe]
\omlijnd
  [hoogte=2cm,breedte=.5\tekstbreedte,
  achtergrond={raster,geintje,grapje},uitlijnen=rechts]
  {\vfill dit is een koe}
```

6.8 METAPOST

In $\text{CON}\TeX$ T kan men in een document METAPOST code opnemen, bijvoorbeeld:

```
\startMPgraphic
  fill unitsquare scaled 100 withcolor (.2,.3,.4) ;
\stopMPgraphic
```

Een koppeling met het $\text{CON}\TeX$ T kleurmechanisme ligt echter meer voor de hand:

```
\startMPgraphic
  fill unitsquare scaled 100 withcolor \MPcolor{mark} ;
\stopMPgraphic
```

METAPOST ondersteuning gaat vrij ver, in de zin dat men definities kan bewaren en op naam kan oproepen. Zo mogelijk worden reeds berekende afbeeldingen hergebruikt.

Een gedetailleerde discussie rond het opnemen van METAPOST figuren gaat hier te ver en vindt elders plaats. Voor het moment is het voldoende de basisprincipes te weten van het plaatsen van figuren in de achtergrond. In het volgende voorbeeld wordt elke keer opnieuw de figuur berekend:

```
\startuseMPgraphic{test a}
  fill unitsquare xscaled \overlaywidth yscaled \overlayheight ;
\stopuseMPgraphic
```

```
\defineoverlay[A Nice Rectangle][\useMPgraphic{test a}]
```

```
\setupbackgrounds[page][background=A Nice Rectangle]
```

Wanneer een figuur hetzelfde blijft, kunnen we hem als volgt opnieuw gebruiken:

```
\startreusableMPgraphic{test b}
```

```
  fill unitsquare xscaled \overlaywidth yscaled \overlayheight ;
```

```
\stopreusableMPgraphic
```

```
\defineoverlay[A Nice Rectangle][\reuseMPgraphic{test b}]
```

```
\setupbackgrounds[page][background=A Nice Rectangle]
```

Als men de `CONTEXT` command line interface `TEXEXEC` gebruikt, dan worden de figuren automatisch verwerkt. , Na het wijzigen van een figuur is tweede verwerkingsslag nodig, tenzij men `METAPOST direct` tijdens het processen van een tekst aanroept.

7

Taalspecifieke zaken

Taalspecifieke zaken

7.1 Inleiding

Iedere taal heeft zijn eigen regels voor het afbreken van woorden, het weergeven van verbindingstreepjes, het omgaan met samengestelde woorden, het gebruik van accenten, enz.

Van nature is $\text{T}_{\text{E}}\text{X}$ erg engels georiënteerd, hoewel in $\text{T}_{\text{E}}\text{X}$ versie 3 wat meer mogelijkheden beschikbaar kwamen voor het (gelijktijdig) ondersteunen van meerdere talen. Er kan veilig worden geconcludeerd dat voor talen met veel accenten, soms zelfs met meerdere per ‘karakter’, de invoer en er vaak ingewikkelder uitziet, dan bijvoorbeeld voor engels.

Een en ander wordt nog gecompliceerd door het feit dat een specifieke invoer niet per se één-op-één overeenkomt met de positionering van glyphs in een font. We stonden daar al bij stil in paragraaf 5.11. Het is daarom erg belangrijk de juiste keuzes voor de fontmatrix en de invoer encoding te maken, hoewel die in veel gevallen zich de afgelopen jaren al redelijk hebben uitgekristalliseerd. De lezer is gewaarschuwd.

7.2 Automatisch afbreken

Er kunnen verschillende afbreekregels worden gehanteerd, afhankelijk van de taal waarin wordt geschreven, bijvoorbeeld: nederlands ($\backslash\text{n1}$), engels ($\backslash\text{en}$), Duits ($\backslash\text{de}$) en Frans ($\backslash\text{fr}$). Een taal wordt ingesteld met het commando:¹⁶

```
 $\backslash\text{taal}[\dots]$ 
...    $\underline{\text{n1}}$  fr en uk de es cz ..
```

Hiernaast zijn ook de onderstaande korte commando's beschikbaar, al dan niet te gebruiken binnen accolades:

```
 $\backslash\text{n1}$   $\backslash\text{en}$   $\backslash\text{de}$   $\backslash\text{fr}$   $\backslash\text{sp}$   $\backslash\text{uk}$   $\backslash\text{pl}$   $\backslash\text{cz}$  ...
```

Het commando $\backslash\text{taal}[\text{n1}]$ komt dus overeen met $\backslash\text{n1}$, waarbij het eerste commando wat duidelijker is. Omdat de mogelijke lijst met twee-letterige codes vrij groot is, is de kans op conflicten met bestaande commando's vrij groot, neem bijvoorbeeld $\backslash\text{it}$, de afkorting voor Italiaans en het commando voor *italic*. Om die reden wordt, wanneer er een potentieel conflict is, geen commando aangemaakt. We gebruiken dan hoofdletters. In de nog te bespreken commando's kan men overigens nog wel de twee letterige code gebruiken.

Er kan op ieder moment in de tekst worden overgegaan op een andere taal. In het onderstaande voorbeeld wordt bijvoorbeeld heen en weer gesprongen tussen Nederlands en Frans.

```
[file cont-n1-messian.tmp bestaat niet]
```

¹⁶ Ga in geval van twijfel na of de afbreekpatronen inderdaad zijn opgenomen in de *fmt*-file. De beschikbaarheid hangt mede af van de capaciteit van de gebruikte $\text{T}_{\text{E}}\text{X}$ -versie.

Het blijkt niet echt nodig te zijn hier te wisselen van taal, omdat we niet over de regelgrens gaan:

Woorden als *perestrojka* en *glasnost* worden in de regel goed afgebroken, het zijn immers in Nederland gebezigde russische termen. Mocht onverhoopt een afbreking niet op de juiste manier plaatsvinden, dan kan het afbreekpatroon worden vastgelegd met het \TeX -commando:

```
\hyphenation{af-kor-ting}
```

Ook in een tekst kan worden aangegeven waar afbreken is toegestaan: op de gewenste plaats kan het commando `\-` worden toegevoegd: `toege\-`staan.

7.3 Definities en instellingen

Aangezien een \TeX format op het moment dat het wordt aangemaakt, de afbreekpatronen ingebakken krijgt, valt het definiëren en installeren van een taal buiten het blikveld van een gebruiker. Dat wil niet zeggen dat een gebruiker het niet kan en mag doen, het moet alleen wel op de juiste manier gebeuren. We laten daarom het proces zien, al is het maar om inzicht in het mechanisme te verschaffen. Neem bijvoorbeeld:¹⁷

```
\installlanguage
[en]
[spatiering=ruim,
linkerzin=---,
rechterzin=---,
linkersubzin=---,
rechtersubzin=---,
linkerciteer=\upperleftsinglesixquote,
rechterciteer=\upperrightsingleninequote,
linkercitaat=\upperlefthoublesixquote,
rechtercitaat=\upperrightdoubleninequote,
datum={maand,\ ,dag,{,\ },jaar},
default=en,
status=stop]
```

en:

```
\installlanguage
[uk]
[default=en,
status=stop]
```

De eerste definitie definieert de engelse taalcomponent. Deze definitie is te vinden in de file `lang-ger.tex`, de germaanse talen. Talen zijn dus geordend in taalgroepen. Dit heeft vooralsnog geen bijzondere betekenis, omdat standaard alle beschikbare definities worden geladen.

De tweede definitie erft zijn instellingen grotendeels van engels. Bij beide definities is `status` ingesteld op `stop`. Dit betekent dat nog geen patronen zijn geladen. Dat gebeurt in de files

¹⁷ De wat vreemde naam `\upperleftsinglesixquote` geeft in ieder geval aan hoe de betreffende quote er ongeveer uitziet.

`cont-xx`, de taal- en/of interface specifieke `CONTEXT` versies. Zodra `status` op start wordt gezet, wordt een patroon geladen, iets dat alleen kan tijdens het genereren van een format.

Vooralsnog gelden enkele conventies rond de namen van files met patronen `lang-xx.pat` en uitzonderingen `lang-xx.hyp`. In de regel wordt een taal geïnstalleerd onder een twee letterige code. Er zijn echter ook drie letterige afkortingen: bijvoorbeeld `deo` voor deutch oude spelling, en `n1x` voor nederlands extended character set, ofwel 8-bit encoding.

Na het installeren is men overigens niet gebonden aan de twee-letterige aanduidingen. Standaard zijn namelijk de (engelse) lange equivalenten gedefinieerd:

```
\installlanguage[german][de]
```

```
\installeertaal[...][...=...]
```

...	<i>naam</i>
spatiering	<u>o</u> pe <u>l</u> kaar ruim
lefthyphenmin	<i>maat</i>
righthyphenmin	<i>maat</i>
status	start <u>s</u> top
linkerzin	<i>commando</i>
rechterzin	<i>commando</i>
linkersubzin	<i>commando</i>
rechtersubzin	<i>commando</i>
linkerciteer	<i>commando</i>
rechterciteer	<i>commando</i>
linkercitaat	<i>commando</i>
rechtercitaat	<i>commando</i>
default	<i>naam</i>

```
\steltaal in[...][...=...]
```

```
...   n] fr en uk de es cz ..
..=.. zie p 140: \installeertaal
```

Het is overigens ook mogelijk taal synoniemen te creëren, bijvoorbeeld:

```
\installlanguage[us][en]
```

De instellingen in de genoemde commando's hebben betrekking op situaties zoals hieronder:

```
\huidigedatum
|<|ziezo daar gaan we|>| | | | |
|<| |<|ziezo|>| daar gaan we|>|
|<|ziezo |<|daar|>| gaan we|>|
\citeer{ziezo daar gaan we}
\citaat{ziezo daar gaan we}
\citaat{\citeer{ziezo} daar gaan we}
\citaat{ziezo \citeer{daar} gaan we}
```

Dit wordt:

12 november 2001—ziezo daar gaan we— — —ziezo— daar gaan we— —ziezo —daar— gaan we— ‘ziezo daar gaan we’ “ziezo daar gaan we” “‘ziezo’ daar gaan we” “ziezo ‘daar’ gaan we”
 Op || gaan we in een van de volgende paragrafen nader in.

7.4 Datum

De wijze waarop de datum wordt weergegeven is taalspecifiek, vandaar dat we er in dit hoofdstuk bij stilstaan. De systeemdatum kan, ervan uitgaande dat de klok van de computer gelijk loopt, worden opgeroepen met:

```
\huidigedatum[.,.,.,.]
... zie p 141: \datum
```

De volgorde waarin dag, maand en jaar worden opgegeven ligt niet vast. Het patroon [dag,maand,jaar] levert bijvoorbeeld als datum op: 12 november 2001. Als we ook nog weekdag toevoegen, krijgen we maandag 12 november 2001.

Als men een kortere aanduiding wil dan kan dat ook: [dd,mm,jj] levert 121101, [d,m,j] levert het wat minder duidelijke 12112001 en met [kenmerk] krijgt men een kenmerkdatum: 20011112. Combinaties zijn ook mogelijk. Er kunnen in de datum karakters worden opgenomen, zo is de datum 12-11-01 opgeroepen door het patroon [dd,--,mm,--,jj].

Een willekeurige datum kan worden gezet met het commando:

```
\datum[.,.,.=.,.,.][.,.,.,.]
d      getal
m      getal
y      getal
...    dag maand weekdag jaar dd mm jj yy d m j y kenmerk
```

Het eerste (optionele) argument wordt gebruikt om de datum te specificeren:

```
\datum[d=10,m=3,y=1996][weekdag,dag,maand,jaar]
```

Als geen argument wordt gegeven, krijgen we de huidige datum, en bij gebrek aan tweede argument, is de weergave gelijk aan die van \huidigedatum. Het voorbeeld resulteert in:

```
zondag 10 maart 1996
```

Bij de conversie naar dagen wordt rekening gehouden met schrikkeljaren. Als men echter terug in de tijd gaat, dan wordt deze conversie soms wat onbetrouwbaar. Zo zijn in 1582 onder invloed van paus Gregorius XIII tussen 4 en 15 oktober zo'n 10 dagen verdwenen en is vanaf dat moment het schrikkeljaar ingevoerd. In Engeland is men pas in 1752 overgegaan op de nieuwe wijze van datumbepaling en in Frankrijk is als nasleep van de Franse revolutie tussen 1792 en 1806 een alternatieve telling gehanteerd met maanden van 30 dagen en weken van 10 dagen verdeeld in 10 uren van 100 minuten. In Rusland zijn tussen 1929 en 1940 weken

van vijf werk- en één rustdag gehanteerd. Het leek ons niet echt verstandig in `CONTEXT` een taalafhankelijke datumbepaling in te bouwen.

7.5 Labels en koppen

In een aantal gevallen plaatst `TEX` automatisch een woord, bijvoorbeeld het woord ‘figuur’ onder een figuur. Dergelijke woorden noemen we labels. Labels kunnen worden ingesteld met het volgende commando:

```
\stellabeltekstin[...][..=..]
...   n] fr en uk de es cz ..
naam  tekst
```

Relevante labels zijn: `tabel`, `figuur`, `hoofdstuk`, `bijlage` en vergelijkbare elementen. Een voorbeeld van een instelling is:

```
\stellabeltekstin[en][hoofdstuk=Chapter ]
\stellabeltekstin[nl][hoofdstuk=Hoofdstuk ]
```

De spatie achter `Chapter` is essentieel, omdat anders het hoofdstuknummer aan `Chapter` vast komt te staan (`Chapter1` in plaats van `Chapter 1`). Een labeltekst kan worden opgevraagd met:

```
\labeltekst{...}
...   tekst
```

Sommige talen, zoals Chinees, maken gebruik van gesplitste labels. Dergelijke labels worden gedefinieerd met behulp van een `commalist`, zoals `chapter={left,right}`.

Koppen kunnen worden ingesteld met:

```
\stelkoptekstin[...][..=..]
...   n] fr en uk de es cz ..
naam  tekst
```

Relevante koppen zijn: `inhoud`, `tabellen`, `figuren`, `afkortingen`, `index` enz. Een voorbeeld van een instelling is:

```
\stelkoptekstin[inhoud=Inhoudsopgave]
```

Een koptekst kan worden opgevraagd met:

```
\koptekst{...}
...   tekst
```

Labels en kopteksten worden in de `lang-xxx` files ingesteld. Het kan beslist geen kwaad daarin eens een kijkje te nemen.

De taal waarin op een bepaald moment wordt gezet, hoeft niet dezelfde te zijn als die van de labels. Naast `\taal` hebben we daarom:

```
\hoofdtaal[...]
...   n] fr en uk de es cz ..
```

Normaal gesproken wordt een document in een taal gezet, de hoofdtaal, bijvoorbeeld `\hoofdtaal[en]`. Een tijdelijke overgang naar een andere taal wordt dan gerealiseerd met het commando `\taal[n]`, omdat dat de labels en kopteksten ongemoeid laat.

7.6 Taal gebonden commando's

Duitstalige \TeX gebruikers zijn gewend "e in te voeren en ë gezet terug te krijgen. Deze combinatie en nog meer twee en drietallen worden gedefinieerd in `lang-ger`. Afgezien van comfort, zorgt dit mechanisme ook voor afbreken en nauwkeurig positioneren van accenten. Schrik niet van het volgende (deel)voorbeeld, want hier worden wat macro's gebruikt die niet worden toegelicht. Dit soort definities vallen buiten de verantwoordelijkheid van de gebruiker.

```
\startlanguagespecifics[de]
  \installcompoundcharacter "a  {\moveaccent{-.1ex}\a\midworddiscretionary}
  \installcompoundcharacter "s  {\SS}
  .....
  \installcompoundcharacter "U  {\smashaccent\"U}
  \installcompoundcharacter "Z  {SZ}
  .....
  \installcompoundcharacter "ck {\discretionary {k-}{k}{ck}}
  \installcompoundcharacter "TT {\discretionary{TT-}{T}{TT}}
  .....
  \installcompoundcharacter "'  {\handlequotation\c!leftquotation}
\stoplanguagespecifics
```

Het commando `\installcompoundcharacter` zorgt ervoor dat in geval van Duits zetten, "a wordt omgezet in ä, "U in Ü, "ck op de juiste wijze wordt afgebroken, enz.

Men kan zelf meer definities toevoegen, maar dat komt de overdraagbaarheid van teksten niet ten goede. In een Poolse versie van $\text{CON}\text{T}\text{E}\text{X}\text{T}$ wordt overigens een / gebruikt in plaats van een ".

7.7 Automatisch vertalen

Het is mogelijk een tekst automatisch in de actuele taal te laten zetten. Dit kan bijvoorbeeld handig zijn bij standaard briefhoofden. Het onderstaande voorbeeld illustreert dit:

```
\vertaal[.,.,.=.,.,.]
naam  tekst
```

Afhankelijk van de actuele taal wordt de toegekende tekst gezet:

```
{\en this is an \vertaal [nl=voorbeeld, en=example], \nl of in goed
nederlands: \vertaal}.
```

De tweede aanroep `\vertaal` maakt gebruik van de eerder toegekende waarden.

Afhankelijk van de actuele taal wordt de toegekende tekst gezet: `this is an example`, of in goed nederlands: `voorbeeld`.

7.8 Samengestelde woorden

Lange woorden kunnen omwille van de leesbaarheid worden opgedeeld in meerdere delen. Deze delen worden gescheiden door een verbindingsstreepje, zoals in x-as. Dit streepje kan worden opgegeven tussen `|`, bijvoorbeeld `--|`. Als alleen `|` wordt gegeven, wordt het standaard ingestelde koppelteken gebruikt. Dit koppelteken, dat standaard is ingesteld op `--`, kan worden ingesteld met het volgende commando.

```
\stelkoppeltekenin[.=..]
teken  -- --- - ) (= /
```

Het commando kan ook worden gebruikt bij andere vormen van koppelen. Het mechanisme is nog niet helemaal waterdicht. Mocht het woord toch nog over de kantlijn heen gaan, dan kunnen in het eerste deel van het woord `\-` worden geplaatst, bijvoorbeeld: `drink\water|` (`voorziening`).

invoer	normaal	afgebroken
<code>polymeer chemie</code>	<code>polymeer-chemie</code>	<code>po-ly-meer-che-mie</code>
<code>polymeer - chemie</code>	<code>polymeer-chemie</code>	<code>po-ly-meer-che-mie</code>
<code>polymeer (chemie)</code>	<code>polymeer(chemie)</code>	<code>po-ly-meer(-che-mie)</code>
<code>(polymeer) chemie</code>	<code>(polymeer)chemie</code>	<code>(po-ly-meer-)che-mie</code>
<code>polymeer -- chemie</code>	<code>polymeer-chemie</code>	<code>po-ly-meer-che-mie</code>
<code>polymeer ~ chemie</code>	<code>polymeer chemie</code>	<code>po-ly-meer-che-mie</code>

Tabel 7.1 Afbreken van gecombineerde woorden.

We zien dat dit commando, dat bij wijze van uitzondering niet begint met een `\`, meerdere doelen dient. Om te beginnen weet $\text{T}_{\text{E}}\text{X}$ niet goed raad met het afbreken van gecombineerde woorden. Vaak verdient afbreken achter achter streepje, of voor een haakje, de voorkeur. Dit commando helpt $\text{T}_{\text{E}}\text{X}$ bij het bepalen van het juiste punt om af te breken.

8

Tekstonderdele

Tekstonderdelen

8.1 Inleiding

De kern van `CONTEXT` wordt gevormd door de structuurcommando's. De meest voor de hand liggende structurerende elementen zijn hoofdstukken en paragrafen. Structuur wordt zichtbaar gemaakt door middel van de koppen en samengevat in de inhoudsopgave.

Een tekst kan op verschillende manieren worden ingedeeld. Ter inleiding vermelden we hier enkele mogelijke indelingen, achtereenvolgens van H. van Krimpen, K. Treebus en het Collectief Gaade:

- | | | |
|-----------------------------|---------------------------|----------------------|
| 1. franse titel | 6. ... | 11. noten |
| 2. titel | 7. lijst van illustraties | 12. literatuuropgave |
| 3. geschiedenis & copyright | 8. verantwoording | 13. register(s) |
| 4. opdracht | 9. corrigenda/errata | 14. colofon |
| 5. voorrede/ten geleide | 10. de eigenlijke inhoud | |

De franse titel vinden we (in boeken) tegenover het schutblad. In het colofon is plaats voor de gebruikte letter, de namen van de typograaf, zetter en illustrator, de oplaag, de drukkerij, het soort papier, de bindwijze enz. De indeling van Treebus is als volgt:

- | | | |
|------------------|----------------------------|------------------------------|
| 1. franse titel | 8. lijst van illustraties | 15. literatuuropgaven |
| 2. titelpagina | 9. inleiding/voorwoord (2) | 16. verklarende woordenlijst |
| 3. colofon | 10. ... | 17. addenda |
| 4. copyright | 11. nawoord/epiloog | 18. index |
| 5. opdracht | 12. appendices | 19. verantwoording foto's |
| 6. voorwoord (1) | 13. samenvattingen | 20. (colofon) |
| 7. inhoudsopgave | 14. noten | |

Bij deze indeling staat het colofon op de achterzijde van de titelpagina. Het eerste voorwoord is van anderen dan de auteur.

De laatste indeling is die van het Collectief Gaade:

- | | | |
|-----------------|----------------|----------------------------|
| 1. franse titel | 7. voorwoord | 13. lijst van illustraties |
| 2. serietitel | 8. inhoud | 14. woordenlijst |
| 3. titel | 9. inleiding | 15. bibliografie |
| 4. copyright | 10. ... | 16. colofon |
| 5. opdracht | 11. appendices | 17. index |
| 6. blanco | 12. noten | |

Omdat er dus geen eenduidige indeling is, is het mechanisme om structuur aan te brengen in `CONTEXT` zo opgezet, dat meerdere alternatieven mogelijk zijn. Daarbij is getracht aan de volgende voorwaarden is voldoen:

1. Hoewel een tekst zelden meer dan vier niveaus kent, kunnen in met name handboeken aanzienlijk meer niveaus nodig zijn. Waar in papieren documenten een grote diepgang verwarrend kan werken, geldt bij interactieve documenten dat een grote diepgang in structuur

vaak noodzakelijk is. Omdat de gebruiker nauwelijks een fysieke voorstelling van het document heeft, zijn structurerende elementen nodig om zinvolle informatie-eenheden te kunnen aanbieden.

2. Hoewel niet ieder niveau een nummer hoeft te hebben, is toch vaak op de achtergrond een nummer nodig, bijvoorbeeld om uniek te kunnen verwijzen.
3. De naamgeving van structurerende elementen moet aansluiten bij het doel.
4. Inhoudsopgaven en registers moeten op ieder niveau kunnen worden opgeroepen en moeten geavanceerde interactieve documenten ondersteunen.
5. Men moet een document kunnen opdelen in functionele onderdelen, bijvoorbeeld inleidingen of bijlagen, met specifieke kenmerken.
6. Het mechanisme moet het afbreken tussen koppen zo goed mogelijk afhandelen.
7. Hoofd- en voetteksten moeten worden ondersteund, uitgaande van document-specifieke namen, zoals hoofdstuk in een tekst of procedure in een handboek.
8. Een gebruiker moet zelf een eigen vormgeving van koppen kunnen afdwingen, zonder zich zorgen te hoeven maken over spatiering, verwijzingen en synchronisatie.

Deze eisen hebben geresulteerd in een mechanisme dat achter de schermen veel werk verzet. De in de volgende paragrafen besproken commando's zijn een voorbeeld van een implementatie, in dit geval gericht op teksten. Aan het eind van dit hoofdstuk laten we zien hoe zelf aanvullende definities mogelijk zijn. We zullen de termen *kop* en *sectie* door elkaar gebruiken, maar bedoelen vaak hetzelfde.

8.2 Onderverdeling

Een tekst wordt in hoofdstukken, paragrafen enz. ingedeeld met de commando's:

```
\deel[ref,..]{...}
... tekst
```

```
\hoofdstuk[ref,..]{...}
... tekst
```

```
\paragraaf[ref,..]{...}
... tekst
```

```
\subparagraaf[ref,..]{...}
... tekst
```

```
\subsubparagraaf[ref,...]{...}
... tekst
```

en

```
\titel[ref,...]{...}
... tekst
```

```
\onderwerp[ref,...]{...}
... tekst
```

```
\subonderwerp[ref,...]{...}
... tekst
```

```
\subsubonderwerp[ref,...]{...}
... tekst
```

De eerste serie commando's (`\hoofdstuk ...`) genereren een genummerde kop, bij de tweede serie blijft het kopnummer achterwege. Overigens zijn standaard enkele niveaus meer beschikbaar; de namen laten zich raden.

niveau	genummerde kop	nummerloze kop
1	<code>\deel</code>	
2	<code>\hoofdstuk</code>	<code>\titel</code>
3	<code>\paragraaf</code>	<code>\onderwerp</code>
4	<code>\subparagraaf</code>	<code>\subonderwerp</code>
5	<code>\subsubparagraaf</code>	<code>\subsubonderwerp</code>

Tabel 8.1 Een overzicht van de structurerende elementen.

Standaard genereert het commando `\deel` geen kop omdat op deelniveau vaak bijzondere koppen worden gebruikt. Wel hoort dit commando de noodzakelijke teller op en voert het de nodige initialisaties uit. Het commando wordt net zo gebruikt als de overige commando's en krijgt dus een argument mee. Het al dan niet plaatsen van een kop en/of het nummer van een kop is instelbaar.

Een structurerend element heeft twee argumenten. Het eerste argument, de referentie, maakt het mogelijk naar bijvoorbeeld een hoofdstuk of bladzijde waarop een hoofdstuk begint te

verwijzen. In paragraaf 9.5 wordt dit verder uitgewerkt. Een referentie kan worden weggelaten, bijvoorbeeld:

```
\paragraaf{Onderverdeling}
```

Hoewel `CONTEXT` uitstekend in staat is zelf de nummers van hoofdstukken en paragrafen te genereren, kan het soms voorkomen dat men de in een (meestal bestaand) document al beschikbare nummers met de hand wil afdwingen. Dit kan dan ook.

```
\stelkopin[subparagraaf][eigennummer=ja]
```

```
\subparagraaf{399}{Het houdt niet op.}
```

```
\subparagraaf[eenref]{400}{Nooit dus.}
```

We zien hier dus een extra argument opduiken. Op de achtergrond genereert `CONTEXT` nog steeds een eigen nummer, zodat men er van verzekerd kan zijn dat de mechanismen die van de nummering afhankelijk zijn goed werken. Overigens hoeft het eigennummer helemaal geen nummer te zijn, `ABC-123` is ook goed.

399 **Het houdt niet op.**

400 **Nooit dus.**

In hoofd- en voetteksten kan (automatisch) de titel van een hoofdstuk, paragraaf enz. worden geplaatst. Als deze titel te lang is, kan ze worden begrensd met:

```
\geenmarkering{...}
... tekst
```

bijvoorbeeld:

```
\hoofdstuk{Stromingen \geenmarkering{in de 20e eeuw:} een vogelvlucht.}
```

De bij `\geenmarkering` opgegeven tekst wordt in dit geval in de hoofd- en/of voetregels vervangen door puntjes. Men kan overigens ook op een andere en meestal meer comfortabele manier te lange titels afvangen. Het volgende commando zorgt ervoor dat de hoofdstuktitel links, en de paragraaftitel rechts in de kopregel komen te staan. Beiden worden daarbij begrensd in lengte.

```
\stelhoofdtekstenin[hoofdstuk][paragraaf]
```

```
\stelhoofdin[linkerbreedte=.4\hsize,rechterbreedte=.5\hsize]
```

Een soortgelijk probleem kan zich voortdoen bij de inhoudsopgave; in dat geval gebruiken we `\geenlijst`:

```
\hoofdstuk{Stromingen in de 20e eeuw: \geenlijst{een vogelvlucht.}}
```

Als men in een titel het commando `\\` opneemt, wordt in de titel zelf overgegaan naar een nieuwe regel. Met `\crlf` gaat men alleen in de inhoudsopgave over op een nieuwe regel. Zo levert:

```
\hoofdstuk
```

```
{Stromingen in de 20e eeuw:\crlf een vogelvlucht.}
```

in de inhoudsopgave twee regels op, terwijl in de tekst de titel op één regel staat. Het omgekeerde zou het geval zijn als we `\` hadden gebruikt.

Het is mogelijk eigen koppen te definiëren. Een ‘eigen’ kop wordt daarbij gekoppeld aan een eerder gedefinieerde kop.

```
\definieerkop[.1.][.2.]
.1.    naam
.2.    sectie
```

Een voorbeeld van een definitie is:

```
\definieerkop[rubriek][subonderwerp]
```

Het vanaf dat moment beschikbare commando `\rubriek` gedraagt zich net als `\subonderwerp`. Rubrieken worden dus niet genummerd.

Een aantal kenmerken van koppen zijn instelbaar met het in paragraaf 8.3 beschreven commando `\stelkopin`. Ook zelf gedefinieerde koppen zijn instelbaar. Hoewel bepaalde kenmerken behouden blijven, zoals bijvoorbeeld nummeren, kunnen andere kenmerken worden ingesteld. De hierboven genoemde (zelf gedefinieerde) rubrieken kunnen dus nog nader worden ingesteld.

Het is mogelijk het weergeven van de sectienummers te blokkeren met de instelling `\stelkoppenin[sectienummer=nee]`. Er wordt in dat geval achter de schermen wel door genummerd. Dit is nodig met het oog op lokale handelingen, zoals het plaatsen lokale inhoudsopgaven.

Omdat het gevaar bestaat dat een naam reeds wordt gebruikt voor andere doeleinden, raden we het gebruik van hoofdletters aan. We zien dan meteen dat het om een eigen commando gaat:

```
\definieerkop[WerkVoorschrift][paragraaf]
```

8.3 Variaties in koppen

De wijze waarop hoofdstukken, paragrafen en subparagrafen worden genummerd en getiteld, kan worden ingesteld met de commando's:

```
\stelkoppenin[...=...]
```

sectienummer	<u>ja</u> nummer nee
variant	<u>normaal</u> marge midden tekst alinea
scheider	<u>tekst</u>
commando	<u>\commando#1#2</u>

en

```

\stelkopin[...][...=...,...]
...      sectie
letter   normaal vet schuin vetschuin type kap klein... commando
tekstletter   normaal vet schuin vetschuin type kap klein... commando
nummerletter   normaal vet schuin vetschuin type kap klein... commando
nummer        ja nee
eigennummer   ja nee
pagina        links rechts ja
doorgaan      ja nee
hoofd         geen leeg hoog geenmarkering
tekst         geen leeg hoog geenmarkering
voet         geen leeg hoog geenmarkering
voor         commando
tussen       commando
na           commando
variant      normaal inmarge midden tekst
commando     \commando#1#2
nummercommando   \commando#1
tekstcommando   \commando#1
prefix       + - tekst
plaatskop    ja nee
verhoognummer   ja nee file
uitlijnen     links rechts normaal ruim
tolerantie    zeerstreng streng soepel zeersoepe] rek
springvolgendein   ja nee
file         naam
expansie     ja commando nee

```

Het nummer van een kop kan handmatig worden ingesteld met:

```

\stelkopnummerin[.1.][.2.]
.1.    sectie
.2.    getal +getal -getal

```

Dit commando accepteert absolute en relatieve nummers, dus [12], [+2] en [+]. De relatieve methode verdient de voorkeur, zoals:

```
\stelkopin[hoofdstuk][+1]
```

Dit commando wordt alleen gebruikt als men in macros listige dingen met koppen doet. Een nummer kan worden opgevraagd met:

```

\kopnummer[...]
...    sectie

```

en/of:

```
\huidigekopnummer
```

Bijvoorbeeld:

```
\huidigekopnummer : 3
\kopnummer[hoofdstuk] : 8
\kopnummer[paragraaf] : 8.3
```

Als men het nummer in berekeningen wil gebruiken, dan komt alleen `\huidigekopnummer` in aanmerking. Men kan dit nummer berekenen met:

```
\bepaalkopnummer[...]
... sectie
```

Met paragrafen en hoofdstukken meelopende hoofd- en voetregels worden automatisch aangepast bij het overgaan naar nieuwe pagina's. Het volgende resulteert in het steeds op een rechter pagina starten van een nieuw hoofdstuk.

```
\stelkopin
[hoofdstuk]
[pagina=rechts,
na={\blanko[2*groot]}]
```

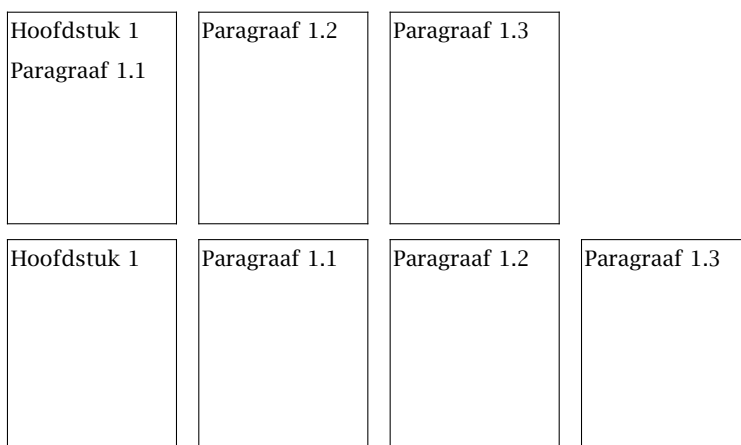
Bij omvangrijke teksten kan men ervoor kiezen de paragrafen op een nieuwe bladzijde te laten beginnen. Doorgaans komt het echter beter uit als de eerste paragraaf op dezelfde bladzijde begint als het hoofdstuk. Het kan echter voorkomen dat men bewust ook de eerste paragraaf op een nieuwe bladzijde wil laten beginnen. In dat geval kan men aan doorgaan de waarde `nee` toekennen.

```
\stelkopin
[paragraaf]
[pagina=ja,doorgaan=nee,
na=\blanko]
```

Figuur 8.1 toont het verschil tussen beide alternatieven. In het eerste geval wordt de eerste paragraaf op dezelfde bladzijde geplaatst (doorgaan), in de tweede situatie begint ook de eerste paragraaf op een nieuwe bladzijde.

Evenzo kan het voorkomen dat men op de bladzijde waarop bijvoorbeeld een hoofdstuk begint *geen* hoofdteksten wil plaatsen. In dat geval moet men aan hoofd de waarde `leeg`, `hoog`, `geenmarkering` of een identificatie van een zelf gedefinieerde hoofdtekst toekennen (zie 4.15).

Standaard worden koppen in een wat groter formaat letter weergegeven dan normaal. De gebruiker kan echter zelf het lettertype bepalen. Wanneer de koppen dezelfde stijl (serif, sans, enz.) moeten hebben als de lopende tekst, dan moet men zo mogelijk gebruik maken van neutrale aanduidingen, dus `\tfb` in plaats van `\rmb`, tenzij men natuurlijk zeker weet dat men een serif letter wil. Zo is ook het gebruik van snelle wisselingen af te raden. Als we bijvoorbeeld `\ssbf` gebruiken in plaats van `\ss\bf`, dan lopen we kans dat kapitalen, synoniemen enz. niet goed worden weergegeven. Dit is overigens geen fout of tekortkoming, maar een gevolg van het feit een snelle wisseling niet voor niets snel is. In het eerste geval schakelen we namelijk



Figuur 8.1 Afwijkend gedrag van eerste subkop.

expliciet over naar een vette sans serif letter, in het tweede geval gaan we over op een sans serif stijl en vervolgens op een vette letter. Commando's als `\kap` passen hun gedrag aan dergelijke wisselingen aan.

Een kop bestaat uit een nummer en een tekst. Het is mogelijk zelf een commando te definiëren dat beide op een andere manier plaatst. Beginners wordt aangeraden gebruik te maken van de standaard weergave van koppen.

8.3.1 Kop variant normaal

8.3.2 Kop variant inmarge

Kop variant midden

Deze koppen zijn gegenereerd met:

```
\stelkoppenin[variant=normaal]
\subparagraaf{Kop variant normaal}
\stelkoppenin[variant=inmarge]
\subparagraaf{Kop variant inmarge}
\stelkoppenin[variant=midden]
\subonderwerp{Kop variant midden}
```

In deze handleiding is een wat afwijkende weergave gebruikt. Het ontwikkelen van zo'n afwijkende kop kost vaak wat tijd. In de onderstaande voorbeelden volgen we dit proces.

```
\def\koptitel#1#2%
  {\hbox to \hsize \bgroup
   \hfill
   \omlijnd[hoogte=1cm,breedte=2cm,uitlijnen=links]{#1}%
   \omlijnd[hoogte=1cm,breedte=4cm,uitlijnen=rechts]{#2}%
   \egroup}

\stelkopin[subparagraaf][commando=\koptitel]
```


8.3.3	Titel
--------------	--------------

Meestal verwacht de lezer het hoofd van een paragraaf aan de linkerkant van de bladzijde, maar hier zien we voor de verandering de kop rechts staan. Met `kader=aan` kan men meer inzicht krijgen in het proces.

8.3.4 | Nog een titel

Deze variant is volgens mij wat aardiger dan de vorige. De eerdere definitie hoeft op slechts enkele punten te worden aangepast. Dit voorbeeld toont tevens wat er allemaal mogelijk is met `\omlijnd`.

```
\def\koptitel#1#2%
  {\hbox to \hsize \bgroup
   \hfill
   \stelomlijndin[hoogte=1cm,offset=.5em,kader=uit]
   \omlijnd[breedte=2cm,uitlijnen=links]{#1}%
   \omlijnd[breedte=4cm,uitlijnen=rechts,linkerkader=aan]{#2}%
   \egroup}

\stelkopin
  [subparagraaf]
  [commando=\koptitel,
   letter=\tfb]
```

We zien dat het lettertype wordt ingesteld met `\stelkopin`. Het heeft dan ook geen zin instellingen in het commando `\koptitel` op te nemen. Het is de vraag wat er gebeurt als we een lange titel meegeven. We laten dit hieronder zien:

8.3.5 | Een wat lange-

Omdat dit duidelijk niet is wat we wensen, passen we de eerdere definitie van `\koptitel` aan. We lossen hierbij meteen het probleem van ontbrekende witruimte onder de kop op.

```
\def\koptitel#1#2%
  {\hbox to \hsize \bgroup
   \hfill
   \stelomlijndin[offset=.5em,kader=uit]
   \tbox{\omlijnd[breedte=3cm,uitlijnen=links]{#1}}%
   \tbox{\omlijnd[breedte=4cm,uitlijnen=rechts,linkerkader=aan]{#2}}%
   \egroup}

\stelkopin
  [subparagraaf]
  [commando=\koptitel]
```

Deze definitie resulteert in een kop waarbij het nummer op de eerste regel uitlijnen (een gevolg van `\tbox`).

8.3.6 Een heel wat langere titel

Hoe fraaier men een kop wil weergeven, des te meer moet men van T_EX weten. Niet alles valt nu eenmaal met algemene macro's op te lossen. Een ogenschijnlijk eenvoudige, maar achter de schermen toch nog wel ingewikkelde variant is die waar de kop in de tekst staat.

```
\stelkopin[subonderwerp] [variant=tekst,letter=vet]
\stelkopin[subsubonderwerp][variant=tekst,letter=schuinvet]
```

Tekstkoppen *Waarom tekstkoppen wat lastiger zijn dan we op het eerste gezicht zouden vermoeden.* Omdat in de regel een beeldscherm breder is dan hoog en papier hoger is dan breed, ligt het voor de hand in geval van elektronische verspreiding van documenten een speciale versie te genereren ten behoeve van printen. In dat geval is het handig als men snel en comfortabel van het ene naar het andere document kan springen. We kunnen daarom documenten koppelen:

```
\koppeldocument
  [schermversie]
  [repman-e]
  [hoofdstuk,paragraaf,subparagraaf,subsubparagraaf,deel,bijlage]
  [Het Reporting Manual]
\stelkopin
  [hoofdstuk,paragraaf,subparagraaf,subsubparagraaf,deel,bijlage]
  [file=schermversie]
```

Het eerste argument van `\koppeldocument` betreft de identificatie van het schermdocument en het tweede argument de filenaam. Het derde argument specificeert de koppelingen, en het vierde is een omschrijving. Dit commando laad alleen de verwijzingen, de daadwerkelijke koppeling wordt vervolgens gerealiseerd met `\stelkopin`. Na dit commando kan men op bijvoorbeeld paragraaftitels klikken om naar dezelfde plaats in het andere document te gaan. Dit werkt voor alle varianten, dus ook voor de tekstvariant. Omdat tekstkoppen over meerdere regels kunnen lopen is achter de schermen nogal wat manipulatie nodig.

8.4 Meta-structuur

Een tekst is in te delen in functionele eenheden. De kenmerken van koppen kunnen afhangen van de eenheden. Standaard onderscheiden we de volgende functionele eenheden:

- inleidingen
- hoofdteksten
- bijlagen
- uitleidingen

In- en uitleidingen kunnen worden omringd door de onderstaande commando's. In dat geval krijgen koppen geen nummers als hoofdstuk, maar worden wel in de inhoudsopgave vermeld. Binnen de functionele eenheid 'hoofdteksten' gebeurt niets bijzonders, maar binnen 'bijlagen' worden de koppen in letters genummerd.

```
\startinleidingen
\volledigeinhoud
```

<code>\hoofdstuk{Inleiding}</code>	in inhoud, geen nummer
<code>\stopinleidingen</code>	
<code>\starthoofdteksten</code>	
<code>\hoofdstuk{Eerste}</code>	nummer 1, in inhoud
<code>\paragraaf{Alfa}</code>	nummer 1.1, in inhoud
<code>\paragraaf{Beta}</code>	nummer 1.2, in inhoud
<code>\hoofdstuk{Tweede}</code>	nummer 2, in inhoud
<code>\onderwerp{Blabla}</code>	geen nummer, niet in inhoud
<code>\stophoofdteksten</code>	
<code>\startbijlagen</code>	
<code>\hoofdstuk{Index}</code>	letter A, in inhoud
<code>\hoofdstuk{Afkortingen}</code>	letter B, in inhoud
<code>\stopbijlagen</code>	
<code>\startuitleidingen</code>	
<code>\hoofdstuk{Verantwoording}</code>	geen nummer, in inhoud
<code>\titel{Colofon}</code>	geen nummer, niet in inhoud
<code>\stopuitleidingen</code>	

Wanneer deze code wordt verwerkt, zullen we zien dat commando's als `\titel` en `\onderwerp` nooit in de inhoud komen en nooit een nummer krijgen. Hun gedrag wordt niet beïnvloed door de functionele inhoud. Het gedrag van de andere commando's hangt af van de instellingen van een functionele eenheid. Het is dus mogelijk 'in één klap' de wijze van nummering van alle genummerde koppen in een functionele eenheid te veranderen.

8.5 Alternatieve mechanismen

Niet ieder document laat zich in een structuur persen met hoofdstukken en paragrafen. Men kan bijvoorbeeld afzien van nummers en een andere aanduiding van niveau gebruiken. Het mechanisme om koppen te plaatsen is hierop afgestemd.

De onderliggende macro's hebben in principe geen weet van hoofdstukken, paragrafen enz. Hoewel ook deze keuze arbitrair is, wordt diep in het binnenste van `CONTEXT` gesproken van secties en sectieblokken. Beide worden gedefinieerd en ingesteld met de commando's:

```
\definieersectieblok[...][...,...=...,...]  
...   naam  
..=.. zie p 156: \stelsectieblokin
```

```
\stelsectieblokin[...] [..., ...=..., ...]
```

```
...      naam
nummer  ja nee
pagina  ja rechts
voor    commando
na      commando
```

```
\definieersectie[...]
```

```
...      naam
```

```
\stelsectiein[.1.][.2.][..., ...=..., ...]
```

```
.1.      naam
.2.      naam
conversie cijfers letters Letters romeins Romeins
vorigenummer ja nee
```

Er zijn standaard vier sectieblokken gedefinieerd:

```
\definieersectieblok [hoofdstuk] [kopnummer=ja]
\definieersectieblok [bijlagen] [kopnummer=ja]
\definieersectieblok [inleidingen] [kopnummer=nee]
\definieersectieblok [uitleidingen] [kopnummer=nee]
```

We zien dat het al dan niet nummeren van een kop hier wordt ingesteld. Het uitzetten van de nummering heeft als gevolg dat er geen lokale inhoudsopgaven meer kunnen worden gegenereerd. Wil men geen nummers zetten, dan kan dat worden ingesteld met `\stelkoppenin[sectienummer=nee]`.

Standaard wordt voor en na een sectieblok overgegaan op een nieuwe (rechter) bladzijde. Dit voorkomt dat markeringen voortijdig worden gereset. Deze paginaovergang is instelbaar met `pagina`.

Binnen `CONTEXT` zijn standaard zeven niveaus beschikbaar, maar meer zijn mogelijk.

```
\definieersectie [sectie-1]
\definieersectie [sectie-2]
.....
\definieersectie [sectie-7]
```

Vervolgens zijn een aantal koppen gedefinieerd met het eerder besproken commando `\definieerkop`. We laten hier enkele van die zeven definities zien:

```
\definieerkop [deel] [sectie=sectie-1]
\definieerkop [hoofdstuk] [sectie=sectie-2]
\definieerkop [paragraaf] [sectie=sectie-3]
```

De definitie van een subparagraaf wijkt wat af. Standaard erven de sub's namelijk de eigenschappen van een paragraaf:

```
\definieerkop
  [subparagraaf]
  [sectie=sectie-4,
   default=paragraaf]
```

De definities van titels en (sub)onderwerpen is weer wat anders, omdat we hier geen nummers willen:

```
\definieerkop
  [titel]
  [koppeling=hoofdstuk,
   default=hoofdstuk,
   verhoognummer=nee]
```

Deze definitie laat zich als volgt verwoorden. Een titel is gekoppeld aan hoofdstuk. Dat wil zeggen dat waar het gaat om het afhandelen van bepaalde kritische handelingen in het binnenste van `CONTEXT`, het commando zich hetzelfde gedraagt als `\hoofdstuk`, bijvoorbeeld bij het forceren cq. blokkeren van een overgang naar een nieuwe bladzijde. Bij het afhandelen van meerdere koppen direct na elkaar speelt het niveau namelijk een rol. Kenmerken als het gebruikte lettertype erft het commando `\titel` van `\hoofdstuk`, tenzij later anders ingesteld. Tot slot geven we aan dat het nummer niet mag worden verhoogd.

We zijn nog niet klaar. De eerder gedefinieerde sectieblokken komen namelijk nog om de hoek kijken:

```
\stelsectiein
  [sectie-2]
  [bijlageconversie=Letter, % Pas op, een hoofdletter!
   vorigennummer=nee]
\stelkopin
  [deel]
  [plaatskop=nee]
\stelkopin
  [hoofdstuk]
  [bijlagelabel=bijlage,
   hoofdtekstlabel=hoofdstuk]
```

Dit betekent dat binnen een `bijlage` conversie naar letters plaatsvindt, echter alleen op niveau van sectie 2. Verder krijgen koppen die zijn gekoppeld aan `sectie-2` geen prefix voor het nummer. Een nummer bestaat namelijk uit de opeenvolgende nummers van de secties:

```
<sectie-1><scheider><sectie-2><scheider><sectie-3> enz
```

Standaard zou sectie 2 dus worden geprefixed door het deelnummer en een scheider (.) en dit is meestal niet wat we wensen, vandaar dat we op dat niveau de prefix blokkeren. Daarmee voorkomen we tevens dat op alle lagere niveaus (sectie 3 en hoger) het deelnummer wordt meegenomen.

We hebben daarnaast ingesteld dat bij delen geen koppen worden geplaatst. Dit kan ongedaan worden gemaakt met:

```
\stelkopin[deel][plaatskop=nee]
```

Hoofdstukken en bijlagen kunnen een label krijgen, dat wil zeggen: worden voorafgegaan door een in te stellen woord. Dit doen we (elders) met `\stellabeltekstin`, bijvoorbeeld:

```
\stellabeltekstin[bijlage=Bijlage~]
```

We zijn er bijna en hoeven alleen nog maar enkele uiterlijke kenmerken van koppen vast te leggen. Omdat alle sub's de instellingen van `\paragraaf` erven en `\titel`, `\onderwerp` en haar sub's die van de genummerde koppen, zijn slechts drie instellingen nodig. We geven deze volledig, omdat ze ons duidelijk maken wat de standaardinstellingen zijn.

```
\stelkopin
  [deel,hoofdstuk]
  [uitlijnen=normaal,
   doorgaan=nee,
   pagina=rechts,
   hoofd=geenmarkering,
   letter=\tfc,
   voor={\blanko[2*groot]},
   na={\blanko[2*groot]}}
```

```
\stelkopin
  [paragraaf]
  [uitlijnen=normaal,
   letter=\tfa,
   voor={\blanko[2*groot]},
   na=\blanko]
```

We gebruiken bewust `\tfc`, zodat de kop zich aanpast aan de actuele stijl. Bij delen en hoofdstukken willen we op de eerste bladzijde geen hoofd geplaatst zien. De `rond \blanko` zijn essentieel, omdat we anders in de knoei komen met [].

Eerder zagen we reeds dat we nieuwe koppen kunnen definiëren met de kenmerken van oude. Meestal voldoet dit:

```
\defineerkop[rubriek][onderwerp][letter=v,t,voor=\blanko]
\defineerkop[subject][paragraaf][letter=v,t,voor=\blanko]
```

Een van de redenen dat het mechanisme dat op de achtergrond werkt vrij ingewikkeld is, ligt in het feit dat we de namen van koppen (secties) tegenkomen als instelling bij andere commando's. Zo komt bij markeringen `rubriek` overeen met `onderwerp`, maar `onderwerp` niet met `paragraaf`. Het eerste ligt voor de hand, het laatste niet. In de loop der jaren is namelijk gebleken dat `onderwerp` te pas en te onpas wordt gebruikt voor tussenkopjes, en deze willen we zelden in een hoofd- of voetregel terugzien.

Bij het instellen van het criterium bij lijsten en registers en de wijze van nummeren kunnen we naast `perparagraaf` ook `persubject` gebruiken. Voor de aanduiding van het niveau mogen we het trefwoord `paragraaf` naast `subject` gebruiken. We kunnen dus zonder problemen de naamgeving van koppen (secties) aanpassen aan het doel, bijvoorbeeld:

```
\definieerkop [handboek]      [sectie=sectie-1]  
\definieerkop [procedure]    [sectie=sectie-2]  
\definieerkop [subprocedure] [sectie=sectie-3]  
\definieerkop [werkinstructie] [procedure]
```

Hierna kunnen we de koppen instellen (of instellingen laten erven van andere koppen). Omdat standaard lijsten worden gedefinieerd, kunnen we ook lijsten van procedures oproepen. We komen hier in de volgende hoofdstukken op terug.

Verwijzingen

Verwijzingen

9.1 Inhoudsopgave

De wellicht meest voorkomende vorm van verwijzen is de inhoudsopgave. Er kan dan ook automatisch een inhoudsopgave worden gegenereerd met het commando:

```
\plaatsinhoud
```

De invulling van de inhoudsopgave wordt bepaald door de plaats waar hij wordt opgeroepen. Als we de inhoudsopgave vòór het eerste hoofdstuk oproepen, krijgen we een volledige inhoudsopgave. Als we dit binnen een hoofdstuk doen, krijgen we alleen het voor dat hoofdstuk relevante deel van de inhoudsopgave. Hetzelfde geldt voor paragrafen en subparagrafen.

```
\hoofdstuk{Gewervelde dieren}
```

```
\plaatsinhoud
```

```
\paragraaf{Paarden}
```

Een inhoudsopgave is een voorbeeld van een samengestelde lijst. Voordat we wat nader op dit soort lijsten ingaan, behandelen we eerst de deellijsten. Een deellijst wordt gedefinieerd met:

```
\definieerlijst[.1.][.2.][.,.,.=.,.,.]
.1.    naam
.2.    naam
..=..  zie p 162: \stellijstin
```

Een voorbeeld van een definitie is:

```
\definieerlijst[eersteniveau]
```

Zo'n lijst kan worden opgeroepen met:

```
\plaatslijst[eersteniveau]
```

Elke lijst kan eigen instellingen hebben:

```
\stellijstin[eersteniveau][breedte=2em]
```

Er kunnen meerdere lijsten tegelijk worden opgeroepen en ingesteld, bijvoorbeeld:

```
\plaatslijst[eersteniveau,tweedeniveau]
```

Een lijst is dus op te roepen met

```
\plaatslijst[.,.,.,.,.] [.,.,.=.,.,.]
...    naam
..=..  zie p 162: \stellijstin
```

De vorm van een lijst wordt met name bepaald door de instellingen van marge, breedte en afstand. De varianten a, b en c werken op regelbasis. Een regel is als volgt opgebouwd:

\stellijstin[.,.,.,.,.][.,.,.=.,.]

...	<i>naam</i>
status	<u>start</u> stop
variant	a <u>b</u> c ... geen commando
koppeling	aan <u>uit</u>
criterium	<i>sectie</i> <u>lokaal</u> vorige huidige alles
paginaovergangen	<i>lijst</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
nummerletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
tekstletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
paginaletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
kleur	<i>naam</i>
commando	\commando#1#2#3
nummercommando	\commando#1
tekstcommando	\commando#1
paginacommando	\commando#1
interactie	<u>sectienummer</u> tekst paginanummer alles
voor	<i>commando</i>
na	<i>commando</i>
tussen	<i>commando</i>
links	<i>tekst</i>
rechts	<i>tekst</i>
label	ja <u>nee</u>
prefix	<u>ja</u> nee
paginanummer	<u>ja</u> nee
<i>sectienummer</i>	ja <u>nee</u>
titeluitlijnen	ja <u>nee</u>
marge	<i>maat</i>
breedte	<i>maat</i> passend
hoogte	<i>maat</i> passend ruim
diepte	<i>maat</i> passend ruim
afstand	<i>maat</i>
scheider	<i>tekst</i>
symbool	geen 1 2 3 ...
expansie	ja <u>nee</u> commando
maxbreedte	<i>maat</i>
..=..	zie p 222: \omlijnd

marge	breedte	afstand	
	kopnummer		kop en paginanummer

In een papieren document kunnen we in principe volstaan met het opgeven van de breedte. In een interactief document bepaalt de breedte echter het aanklikbare deel.¹⁸

Bij variant d loopt de tekst door. In dat geval hebben voor en na geen betekenis. De afstand, die minimaal 1 em is, heeft betrekking op de afstand tot het volgende item. De volgende instellingen

¹⁸ Het exacte gedrag hangt mede af van de waarde die aan interactie is toegekend.

leveren een compacte inhoudsopgave op:

```
\stellijstin
  [hoofdstuk]
  [voor=\blanko,na=\blanko,letter=vet]
\stellijstin
  [paragraaf]
  [variant=d,links=(,rechts=),paginaletter=schuin,prefix=nee]
```

We hoeven hier de lijsten niet te definiëren, omdat dat al is gebeurd tijdens het definiëren van de koppen. De instelling `prefix` heeft betrekking op het al dan niet opnemen van de voorafgaande nummers. Bij deze variant ligt dat minder voor de hand dan bij de andere. Variant `d` ziet er uit als:

(1) Inhoudsopgave 161 (2) Synoniemen 172 (3) Sorteren 175 (4) Markerin-
gen 176 (5) Terugverwijzen (refereren) 180 (6) Regelnummers 184 (7) Voorge-
definieerde verwijzingen 185 (8) Registers 186

Een (deel)regel is als volgt opgebouwd:

links	kopnummer	rechts	kop	pagina	afstand
-------	-----------	--------	-----	--------	---------

Tegelijk met het definiëren van een kop wordt ook een lijst gedefinieerd. Dit betekent dat voor hoofdstukken, paragrafen enz. aparte lijsten worden bijgehouden, die vervolgens afzonderlijk kunnen worden opgeroepen.

Omdat dergelijke koppen bij elkaar horen, ligt het voor de hand de lijsten te combineren. We noemen zo'n gecombineerde lijst een samengestelde lijst. De inhoudsopgave is een voorbeeld van een samengestelde lijst:

```
\definieersamengesteldelijst
  [inhoud]
  [deel,
  hoofdstuk,paragraaf,subparagraaf,subsubparagraaf,
  subsubsubparagraaf,subsubsubsubparagraaf]
  [niveau=subsubsubsubparagraaf,
  criterium=lokaal]
```

De mogelijke instellingen zijn dezelfde als die van deellijsten.

```
\definieersamengesteldelijst[.1.][.2.,..][.3.,..=.,..]
.1.   naam
.2.   lijst
..=.. zie p 164: \stelsamengesteldelijstin
```

```
\stelsamengesteldelijstin[...] [..., ..=.., ..]
```

```
...      naam
niveau  1 2 3 4 sectie huidige
..=..   zie p 162: \stellijstin
```

De lijsten zelf worden gegenereerd met de commando's:

```
\volledigesamengesteldelijst[...] [..., ..=.., ..]
```

```
...      naam
..=..   zie p 164: \stelsamengesteldelijstin
```

```
\plaatssamengesteldelijst[...] [..., ..=.., ..]
```

```
...      naam
..=..   zie p 164: \stelsamengesteldelijstin
```

Het eerste commando plaatst een kop boven de lijst. Omdat in bijvoorbeeld een inhoudsopgave nooit de inhoudsopgave zelf wordt vermeld, gebruikt dit commando `\titel` in plaats van `\hoofdstuk`. Bij de in de volgende paragrafen besproken lijsten wordt daarentegen `\hoofdstuk` gebruikt.

variant	weergave
a	nummer - titel - paginanummer
b	nummer - titel - spaties - paginanummer
c	nummer - titel - punten - paginanummer
d	nummer - titel - paginanummer (doorlopend)
e	titel (omlijnd)
f	titel (links, midden of rechts uitgelijnd)
g	titel (gecentreerd)

Tabel 9.1 Varianten in de vormgeving van lijsten.

Mogelijke varianten in de vormgeving van lijsten zijn in tabel 9.1 weergegeven. Binnen bepaalde grenzen zijn op deze varianten weer variaties mogelijk. Het best kan men hier eens wat mee experimenteren. De drie instellingen `breedte`, `marge` en `letter` worden òf voor alle niveaus tegelijk, òf voor alle vijf niveaus afzonderlijk gespecificeerd.

```
\stelsamengesteldelijstin
[inhoud]
[variant=c,
titeluitlijnen=nee,
breedte=2.5em]
```

De instelling `titeluitlijnen` dwingt ingangen zonder sectienummer (zoals titels, onderwerpen enz) tegen de linker kantlijn; anders wordt een titel uitgelijnd met de genummerde tegenhangers (zoals hoofdstukken, paragrafen enz). Vergelijk:

```
titel
12 hoofdstuk
```

met:

```
titel
12 hoofdstuk
```

Eventueel kunnen achter de `\plaats...-commando's` instellingen worden meegegeven, bijvoorbeeld:

```
\volledigeinhoud[niveau=deel]
```

In dit geval worden alleen de delen vermeld. In plaats van een naam mag ook een nummer worden opgegeven. Dit veronderstelt wel inzicht in de niveaus van de verschillende secties (deel=1, hoofdstuk=2 enz.)

Vooraf bij wat langere inhoudsopgaven kan de plaats waar wordt overgegaan op een nieuwe bladzijde niet de gewenste zijn. Een afdoende automatisch mechanisme is eigenlijk niet te maken. Men kan daarom zelf, indien gewenst, de overgang naar een nieuwe pagina afdwingen, zoals het volgende voorbeeld laat zien.

```
\volledigeinhoud[paginaovergangen={2.2,8.5,12.3.3}]
```

Natuurlijk mag dergelijke ‘fine-tuning’ pas aan het eind van het productieproces plaatsvinden. Bovendien dient bij een revisie van de tekst een heroverweging plaats te vinden. Als reminder wordt daarom tijdens het zetten op het scherm gemeld, dat er overgangen worden afgedwongen.

Een tekst moet ten minste eenmaal zijn verwerkt, wil een lijst kunnen worden geplaatst. Als geen samengestelde lijst wordt geplaatst, terwijl de tekst toch minstens tweemaal is verwerkt, dan is waarschijnlijk een lokale lijst opgevraagd. Binnen een hoofdstuk of paragraaf wordt namelijk *automatisch* een lokale samengestelde lijst gegenereerd.

Er zijn twee commando's om zelf direct iets naar een lijst te schrijven. Het eerste commando kan worden gebruikt om een ingang, het tweede om een commando tussen te voegen:

```
\schrijfnaarlijst[.1.]{.2.}{.3.}
```

```
.1.    sectie naam
```

```
\schrijftussenlijst[.1.]{.2.}
```

```
.1.    sectie naam
```

We volstaan met een eenvoudig voorbeeld:

```
\schrijftussenlijst [paragraaf] {\blanko}
\schrijfnaarlijst [paragraaf] {---} {vanaf hier voorlopig}
\schrijftussenlijst [paragraaf] {\blanko}
```

Het volgende commando heeft alleen zin als dat wat naar de lijst gaat ook elders wordt gebruikt, wat het geval is in bijvoorbeeld koppen:

```
\geenlijst{...}
... tekst
```

Zo levert het volgende voorbeeld in een kop *manager logistiek (vervalt)* op maar in een lijst het kortere *manager logistiek ...*. Vergeet niet dat elke kop een eigen lijst heeft.

```
\definieerkop[functie][eigennummer=ja]
\functie{A-45}{manager logistiek \geenlijst{(vervalt)}}
\plaatslijst[functie][criterium=alles]
```

In een interactief document kan men wat kwistiger zijn met lijsten, niet alleen omdat een scherm niets kost, maar ook omdat bij gebrek aan een fysieke inhoudsopgave, men voor navigatie volledig is aangewezen op wijzen en klikken. Bij het vormgeven van een document kan men bijvoorbeeld het volgende overwegen:

```
\stellayoutin[rechterrands=3cm]
\stelinteractiein[status=start,menu=aan]
\stelinteractiemenuin[rechts][status=start]
\startinteractiemenu[rechts]
  \plaatsinhoud
    [niveau= huidige, criterium=vorige,
     variant=f, uitlijnen=rechts,
     interactie=alles,
     voor=, na=]
\stopinteractiemenu
```

Hier wordt op elke bladzijde een inhoud geplaatst van één niveau diep (namelijk *huidige*) en passend bij het vorige niveau. Als er een paragraaf begint, dan krijgen we de subparagrafen te zien en bij een nieuw hoofdstuk de paragrafen.

Als men in een interactieve tekst op elk niveau een inhoudsopgave oproept, kan men met behulp van de verwijzing [*vorigeinhoud*] terugspringen naar de vorige inhoudsopgave.

Standaard wordt iets dat naar een lijst gaat niet geëxpandeerd, het gaat er letterlijk heen. Wil men wel expanderen, dan kan dat worden ingesteld met *expansie*. In dat geval wordt de betekenis van commando's weggeschreven, iets wat alleen nodig is als er dingen naar een lijst worden geschreven die steeds veranderen, wat in 'normaal' gebruik zelden of nooit voorkomt.

In een wat omvangrijker document, kan het voorkomen dat niet op ieder niveau een inhoud beschikbaar is. Bij wat meer geavanceerde vormgevingen zal daarom behoefte bestaan aan wat meer informatie over een lijst.

\lijstlengte het aantal gevonden elementen
 \lijstbreedte de maximale breedte van een element
 \lijsthoogte de totale hoogte van de lijst

De bovenstaande gegevens zijn van te voren te bepalen met:

```
\bepaallijstkenmerken[.,.,.,.,.] [.,.,.=.,.,.]
...    naam
..=..    zie p 162: \stellijstin
```

We geven hieronder een kort overzicht van de verschillende beschikbare varianten. De eerste drie varianten zijn vooral bedoeld voor papieren documenten. In het volgende voorbeeld geven we met criterium aan dat we de lijst van paragrafen per hoofdstuk willen.

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=a]

9.1 Inhoudsopgave 161
 9.2 Synoniemen 172
 9.3 Sorteren 175
 9.4 Markeringen 176
 9.5 Terugverwijzen (refereren) 180
 9.6 Regelnummers 184
 9.7 Voorgedefinieerde verwijzingen 185
 9.8 Registers 186

\stellabeltekstin[nl][paragraaf={oeps }]

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=a,
 label=ja,breedte=2cm]

oeps 9.1 Inhoudsopgave 161
 oeps 9.2 Synoniemen 172
 oeps 9.3 Sorteren 175
 oeps 9.4 Markeringen 176
 oeps 9.5 Terugverwijzen (refereren) 180
 oeps 9.6 Regelnummers 184
 oeps 9.7 Voorgedefinieerde verwijzingen 185
 oeps 9.8 Registers 186

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=b]

9.1 Inhoudsopgave 161
 9.2 Synoniemen 172
 9.3 Sorteren 175

9.4	Markeringen	176
9.5	Terugverwijzen (refereren)	180
9.6	Regelnummers	184
9.7	Voorgedefinieerde verwijzingen	185
9.8	Registers	186
\plaatslijst		
[paragraaf]		
[criterium=hoofdstuk,variant=b, paginanummer=nee,breedte=passend,afstand=1em]		
9.1	Inhoudsopgave	
9.2	Synoniemen	
9.3	Sorteren	
9.4	Markeringen	
9.5	Terugverwijzen (refereren)	
9.6	Regelnummers	
9.7	Voorgedefinieerde verwijzingen	
9.8	Registers	
\plaatslijst		
[paragraaf]		
[criterium=hoofdstuk,variant=c, hoofdstuknummer=ja,marge=1.5cm]		
9.1	Inhoudsopgave	9-161
9.2	Synoniemen	9-172
9.3	Sorteren	9-175
9.4	Markeringen	9-176
9.5	Terugverwijzen (refereren)	9-180
9.6	Regelnummers	9-184
9.7	Voorgedefinieerde verwijzingen	9-185
9.8	Registers	9-186
\plaatslijst % let op de spaties rond :		
[paragraaf]		
[criterium=hoofdstuk,variant=c, hoofdstuknummer=ja,scheider={ : },breedte=passend]		
9.1	: Inhoudsopgave	9-161
9.2	: Synoniemen	9-172
9.3	: Sorteren	9-175
9.4	: Markeringen	9-176
9.5	: Terugverwijzen (refereren)	9-180
9.6	: Regelnummers	9-184
9.7	: Voorgedefinieerde verwijzingen	9-185
9.8	: Registers	9-186
\plaatslijst		

[paragraaf]

[criterium=hoofdstuk,variant=d]

9.1 Inhoudsopgave 161 9.2 Synoniemen 172 9.3 Sorteren 175 9.4 Markeringen 176
 9.5 Terugverwijzen (refereren) 180 9.6 Regelnummers 184 9.7 Voorgedefinieerde verwijzingen 185 9.8 Registers 186

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=d,
 afstand=2cm]

9.1 Inhoudsopgave 161 9.2 Synoniemen 172 9.3 Sorteren 175
 9.4 Markeringen 176 9.5 Terugverwijzen (refereren) 180 9.6 Regelnummers 184
 9.7 Voorgedefinieerde verwijzingen 185 9.8 Registers 186

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=d,
 links={()},rechts={}]

(9.1) Inhoudsopgave 161 (9.2) Synoniemen 172 (9.3) Sorteren 175 (9.4) Markeringen 176
 (9.5) Terugverwijzen (refereren) 180 (9.6) Regelnummers 184 (9.7) Voorgedefinieerde verwijzingen 185
 (9.8) Registers 186

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=e]

Inhoudsopgave
Synoniemen
Sorteren
Markeringen
Terugverwijzen (refereren)
Regelnummers
Voorgedefinieerde verwijzingen
Registers

\plaatslijst

[paragraaf]

[criterium=hoofdstuk,variant=e,
 breedte=\tekstbreedte,achtergrond=raster]

Inhoudsopgave
Synoniemen

Sorteren
Markeringen
Terugverwijzen (refereren)
Regelnummers
Voorgedefinieerde verwijzingen
Registers

\plaatslijst
 [paragraaf]
 [criterium=hoofdstuk,variant=e,
 breedte=4cm]

Inhoudsopgave
Synoniemen
Sorteren
Markeringen
Terugverwijzen (refereren)
Regelnummers
Voorgedefinieerde verwijzingen
Registers

\plaatslijst
 [paragraaf]
 [criterium=hoofdstuk,variant=f]

Inhoudsopgave
 Synoniemen
 Sorteren
 Markeringen
 Terugverwijzen (refereren)
 Regelnummers
 Voorgedefinieerde verwijzingen
 Registers

\plaatslijst
 [paragraaf]
 [criterium=hoofdstuk,variant=g]

Inhoudsopgave
 Synoniemen
 Sorteren

Markeringen
 Terugverwijzen (refereren)
 Regelnummers
 Voorgedefinieerde verwijzingen
 Registers

Vergeet bij gebruik van kleur niet het kleurmechanisme te activeren. Bij alle lijsten kunnen de individuele lijstelementen interactief zijn, hoewel in de regel bij schermdocumenten de optie alles de meest voor de hand liggende is. Variant e is bij uitstek geschikt voor schermopmaak en accepteert alle instellingen van `\omlijnd`. In het volgende voorbeeld gebruiken we een symbool in plaats van een paragraafnummer. De instelling diepte heeft betrekking op dit specifieke symbool.

```

\plaatslijst
  [paragraaf]
  [criterium=hoofdstuk,variant=a,
   paginanummer=nee,afstand=1em,
   symbool=3,hoogte=1.75ex,diepte=.25ex,nummerkleur=grijs]

```

- 9.1 Inhoudsopgave
- 9.2 Synoniemen
- 9.3 Sorteren
- 9.4 Markeringen
- 9.5 Terugverwijzen (refereren)
- 9.6 Regelnummers
- 9.7 Voorgedefinieerde verwijzingen
- 9.8 Registers

Alle alternatieven bieden de mogelijkheid een eigen commando aan de sectieaanduiding, de tekst en het paginanummer te koppelen. Volledige vrijheid krijgt men echter pas met alternatief geen.

```

\plaatslijst
  [paragraaf]
  [criterium=hoofdstuk,variant=geen,
   nummercommando=\omlijnd,
   tekstcommando=\omlijnd,paginacommando=\omlijnd]

```

9.1	Inhoudsopgave	161
9.2	Synoniemen	172
9.3	Sorteren	175
9.4	Markeringen	176
9.5	Terugverwijzen (refereren)	180
9.6	Regelnummers	184
9.7	Voorgedefinieerde verwijzingen	185
9.8	Registers	186

```
\def\LijstCommando#1#2#3%
  {op blz. {\bf #3} behandelen we {\bf #2}}
```

```
\plaatslijst
[paragraaf]
[criterium=hoofdstuk,variant=geen,
commando=\LijstCommando]
```

op blz. 161 behandelen we **Inhoudsopgave** op blz. 172 behandelen we **Synoniemen** op blz. 175 behandelen we **Sorteren**. Het kan nog kaler. Het alternatief commando laat vrijwel alles aan de gebruiker over.

```
\def\LijstCommando#1#2#3%
  {Op p~#3 behandelen we {\em #2}; }
```

```
\plaatslijst
[paragraaf]
[criterium=hoofdstuk,variant=commando,
commando=\LijstCommando]
```

Op p 161 behandelen we *Inhoudsopgave*; Op p 172 behandelen we *Synoniemen*; Op p 175 behandelen we *Sorteren*; Op p 176 behandelen we *Markeringen*; Op p 180 behandelen we *Terugverwijzen (refereren)*; Op p 184 behandelen we *Regelnummers*; Op p 185 behandelen we *Voorgedefinieerde verwijzingen*; Op p 186 behandelen we *Registers*;

Als alternatief voor `geen` zijn er ook `horizontaal` en `vertikaal`, die voor wat spatiering betreft wat beter geschikt zijn voor het zetten van lijsten in bijvoorbeeld menus.

9.2 Synoniemen

In veel teksten komen afkortingen voor. Een afkorting staat ergens voor. Zowel de afkorting zelf als de betekenis dienen door de hele tekst heen consistent te worden vormgegeven. We willen niet de ene keer ABC zien en vijf bladzijden verderop ABC. Het is daarom mogelijk een lijst aan te maken van afkortingen en hun betekenis. Hierdoor is consistent gebruik van afkortingen gewaarborgd. Bovendien kunnen lijsten van gebruikte afkortingen worden opgeroepen.

Afkortingen zijn voorbeelden van synoniemen. Een categorie synoniemen wordt gedefinieerd met het commando:

```
\definieersynoniemen[.1.][.2.][.3.][.4.]
.1.
.2.   naam meervoud
.3.   commando
.4.   commando
```

De wijze waarop de lijst wordt weergegeven wordt ingesteld met:

```

\stelsynoniemenin[...] [.,.,.=.,.]
...
tekstletter      naam
synoniemletter  normaal vet schuin vetschuin type kap klein... commando
plaats          links rechts boven aaneelkaar inmarge inlinker inrechter
breedte         maat
status          start stop
criterium       alles gebruikt
conversie       ja nee
expansie        ja commando nee
commando        \commando#1#2#3

```

Afkorting^{en} zijn bijvoorbeeld gedefinieerd met het commando:

```
\definieersynoniemen[afkorting][afkorting][\voluit]
```

Op het vierde, optionele, argument komen we later terug. Na het geven van het bovenstaande commando is het commando `\afkorting` beschikbaar. Een voorbeeld van het gebruik van afkorting^{en} is:

```

\afkorting {ANWB} {Algemene Nederlandse Wielrijders Bond}
\afkorting {VVN}  {Veilig Verkeer Nederland}
\afkorting {NS}  {Nederlandse Spoorwegen}

```

Ook kan een betekenis worden opgeroepen, en wel met het commando:

```
\voluit{afkorting}
```

Het is mogelijk in een afkorting commando's op te nemen. In dat geval moet tussen [] de pure vorm worden opgenomen:

```
\afkorting [TEX] {\TeX} {The \TeX\ Typesetting System}
```

De pure vorm bepaalt zowel de plaats in de gesorteerde lijst met synoniemen als de naam van het commando. Oproepen vindt in dit geval dus plaats met `\TeX` en `\voluit{TEX}`. In een lopende tekst gebruiken we overigens `\TeX` en voor interpunctie `\TeX`.

Een synoniem wordt pas in de lijst opgenomen als het ook daadwerkelijk wordt gebruikt. Wil men in de lijst synoniemen alle gedefinieerde synoniemen opnemen, dan dient het criterium gelijk te zijn aan `alles`. Door status gelijk te maken aan `stop`, voorkomt men dat synoniemen in de lijst worden opgenomen, ook al zijn ze gebruikt. Dit kan tijdelijk gebeuren:

```

\stelsynoniemenin[afkorting][status=stop]
\afkorting {NIL} {Niet In Lijst}
\stelsynoniemenin[afkorting][status=start]

```

In dit geval hebben we het eerste argument weggelaten, wat kan omdat de afkorting zelf geen commando is. We roepen deze afkorting dus op met `\NIL`. Beide volgende definities zijn dus equivalent.

```

\afkorting [NIL] {NIL} {Niet In Lijst}
\afkorting {NIL} {Niet In Lijst}

```

De formele definitie van een afkorting is:

```
\synoniem[.1.]{.2.}{.3.}
.1. tekst
.2. tekst
.3. tekst
```

Een lijst met synoniemen wordt gegenereerd met het commando:

```
\plaatslijstmetsynoniemen
```

Het volgende commando genereert een lijst met titel (`\hoofdstuk`):

```
\volledigelijstmetsynoniemen
```

We zien hier het meervoud opduiken in de naam van het commando. Dit meervoud vinden we ook terug in de geplaatste kop, waar bovendien de eerste letter standaard een kapitaal wordt gemaakt. Deze koptekst is aan te passen met `\stelkoptekstin` (zie paragraaf 7.5).

Synoniemen zijn pas beschikbaar als ze worden opgeroepen. Soms kan dit tot conflicten leiden; we besparen de lezer de details. Mocht dit het geval zijn, dan kunnen de betekenissen van de synoniemen worden geladen met het commando:

```
\laadsynoniemen
```

De betekenissen van afkortingen kunnen dus worden geladen met `\laadafkortingen`. De tekst moet natuurlijk wel ten minste een maal zijn verwerkt.

Naast afkortingen zijn ook si-eenheden als synoniemen gedefinieerd. Deze zijn niet standaard beschikbaar, maar worden als aparte module geladen. We gaan hierop dieper in in paragraaf ??.

De oplettende lezer zal hebben gezien dat het commando `\definieersynoniem` vier argumenten heeft. Als vierde argument kan een commando worden meegegeven, waarmee het synoniem kan worden opgeroepen. Op deze manier worden de verschillende commando's zodanig afgeschermd van de rest van `CONTEXT` dat geen conflicten kunnen optreden in het gebruik ervan.

```
\definieersynoniem[Funcctie][Functies][\FuncctieNaam][\FuncctieNummer]
```

Als we vervolgens enkele functies definiëren als:

```
\Funcctie [0001] {0001a} {Drukker}
\Funcctie [0002] {0002x} {Zetter}
```

Dan kunnen we het nummer en de naam oproepen met `\FuncctieNaam` (Drukker en Zetter) en op `\FuncctieNummer` (0001a en 0002x), zoals in:

```
De \FuncctieName{0001} heeft functienummer \FuncctieNummer{0001}.
```


9.3 Sorteren

Een vereenvoudigde variant op lijsten met synoniemen zijn gesorteerde lijsten. Een sorteerlijst wordt gedefinieerd met het commando:

```
\definieersorteren[.1.][.2.][.3.]
.1.
.2.   naam meervoud
.3.   commando
```

De wijze waarop de lijst wordt weergegeven wordt ingesteld met:

```
\stelsorterenin[...][.,.=.,...]
```

...	<i>naam</i>
voor	<i>commando</i>
na	<i>commando</i>
commando	<code>\commando#1</code>
status	<u>start</u> stop
criterium	alles <u>gebruikt</u>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
expansie	ja <i>commando</i> <u>nee</u>

Na het definiëren is het volgende commando beschikbaar. Hierbij staat *sorteer* voor de naam van de lijst.

```
\sorteer[.1.]{.2.}
.1.   tekst
.2.   tekst
```

Aansluitend op andere lijsten en overzichten zijn ook de volgende twee commando's beschikbaar:

```
\plaatslijstmetsorteren
```

en (ook hier is de koptekst in te stellen met `\stelkoptekstenin`):

```
\volledigelijstmetsorteren
```

Een voorbeeld van sorteren is:

```
\definieersorteren[stad][steden]
\stelsorterenin[stad][criterium=alles]

\stad {Rotterdam}
\stad {Utrecht}
```

```
\stad {Amsterdam}
```

```
\stad {Den Haag}
```

```
\plaatslijstmetsteden
```

De definitie moet bovenaan de file of in de omgeving worden opgenomen. De steden mogen op willekeurige plaatsen aan de lijst worden toegevoegd en de lijst kan overal worden opgeroepen.

```
Amsterdam
```

```
Den Haag
```

```
Rotterdam
```

```
Utrecht
```

Er is een pure variant beschikbaar voor het geval commando's worden gebruikt. Deze pure variant bepaalt de plaats in de lijst en de vorm van het commando. Op deze wijze kunnen lijsten met logo's worden aangelegd. Evenals `\afkorting` is `\logo` een voorgedefinieerd commando.

```
\logo [TEX] {\TeX}
```

```
\logo [TABLE] {\TaBlE}
```

Als gebruik wordt gemaakt van de []-variant, dan wordt analoog aan synoniemen een commando gedefinieerd. In het voorbeeld hierboven zijn dus `\TeX` en `\TABLE` beschikbaar; midden in de zin gebruiken we `\TABLE\` en voor interpunctie `\TABLE`. Het verdient aanbeveling hoofdletters te gebruiken omdat anders het gevaar bestaat dat reeds bestaande commando's opnieuw worden gedefinieerd.

Evenals synoniemen, komen ook sorteringen pas beschikbaar als ze worden opgeroepen. Mocht een van de sorteringen problemen geven, gebruik dan:

```
\laadsorteren
```

Wanneer we bij de definitie als derde argument een commando opgeven, dan kunnen de sorteringen met dit commando worden opgeroepen. Op deze manier kunnen ze nooit conflicteren met bestaande commando's. (Zie ook paragraaf 9.2).

9.4 Markeringen

Het is mogelijk in de tekst 'onzichtbare' markeringen aan te brengen, die later kunnen worden opgeroepen. Markeringen kunnen bijvoorbeeld worden gebruikt voor het automatisch laten meelopen van aanduidingen in een hoofd of voet (zoals in een woordenboek). Binnen `CONTEXT` worden markeringen gebruikt om in de hoofd- en voetregels de titels van hoofdstukken en dergelijke te laten meelopen.

Een markering wordt gedefinieerd met:

```
\definieermarkering[.1.][.2.]
```

```
.1. naam
```

```
.2. naam
```

Op het tweede, optionele, argument komen we aan het eind van deze paragraaf terug. Na dit commando kunnen markeringen worden aangebracht met:

```
\markeer[.1.]{.2.}
.1.    naam
.2.    tekst
```

en opgeroepen met:

```
\haalmarkering[.1.][.2.]
.1.    naam
.2.    eerste laatste vorige beide alles huidige
```

Er worden, analoog met het \TeX -commando `\mark`, per soort markering drie markeringen bijgehouden: (zie tabel 9.2).

markering	plaats
vorige	de laatste van de vorige bladzijde
eerste	de eerste van de huidige bladzijde
laatste	de laatste van de huidige bladzijde
beide	eerste — laatste
alles	vorige — eerste — laatste

Tabel 9.2 Bijgehouden markeringen, aangevuld met enkele combinaties.

In geval van een combinatie van markeringen (**beide** en **alles**) worden markeringen standaard gescheiden door `—`. Deze scheiding is in te stellen met `scheider`.

```
\stelmarkeringin[...][. .= .]
...        naam
status     start stop
scheider   commando
expansie   ja nee
```

Het zetten van een markering kan worden geblokkeerd met `status`. De variabele `expansie` heeft betrekking op het al dan niet expanderen van het argument. Standaard wordt niet geëxpandeerd. Dat wil zeggen dat een commando als `commando` wordt bewaard. Dit scheelt aanzienlijk in het gebruik van het geheugen en is vaak ook wenselijk. Als we echter een commando meegeven dat steeds wijzigt, bijvoorbeeld een teller, dan moeten we wel expanderen. Immers, de meest recente waarde van de teller is niet de waarde die we wilden markeren.

Markeringen kunnen worden geïnitieerd met het commando:

```
\resetmarkering[...]
```

```
... naam
```

Bij het begin van een hoofdstuk worden bijvoorbeeld automatisch de markeringen van de paragrafen, subparagrafen enz. gereset. Zouden we dit niet doen, dan zouden de markeringen van het vorige hoofdstuk blijven gelden totdat een nieuwe wordt aangebracht.

Stel dat een woordenlijst als volgt wordt aangemaakt (we forceren hier bewust wat pagina-overgangen):

```
\definieermarkering[woorden]
\markeer[woorden]{eerste}eerste woord ...
\markeer[woorden]{tweede}tweede woord ...
\pagina
\markeer[woorden]{derde}derde woord ...
\markeer[woorden]{vierde}vierde woord ...
\pagina
\markeer[woorden]{vijfde}vijfde woord ...
\pagina
```

De uitkomsten staan in tabel 9.3 vermeld.

pagina	vorige	eerste	laatste
1	—	eerste	tweede
2	tweede	derde	vierde
3	vierde	vijfde	vijfde

Tabel 9.3 Het opschuiven van markeringen.

Bij het plaatsen van de titels van hoofdstukken, paragrafen enz. wordt *eerste* gebruikt. De waarden van markeringen zijn eenvoudig te controleren door de markering in de voet te plaatsen:

```
\stelvoettekstenin
[{\haalmarkering[woorden][eerste]}]
[]
```

of allemaal tegelijk:

```
\stelvoettekstenin
[{\haalmarkering[woorden][vorige]} --
{\haalmarkering[woorden][eerste]} --
{\haalmarkering[woorden][laatste]}]
[]
```

Hetzelfde resultaat kan ook eenvoudiger worden bereikt. De onderstaande methode heeft bovendien als voordeel dat lege markeringen worden afgevangen.

```
\stelvoettekstenin[{\haalmarkering[woorden][alles]}][ ]
```

Commando's als `\hoofdstuk` maken automatisch markeringen aan. Als de titel van een hoofdstuk te lang is om in een hoofd- of voetregel te worden gebruikt, kan gebruik worden gemaakt van het commando `\geenmarkering` (zie paragraaf 8.2). Een snelle manier om voetteksten van markeringen te voorzien:

```
\stelvoettekstenin[hoofdstuk][paragraafnummer]
```

Wanneer we zelf een kop definiëren, erft de nieuwe kop de markering van de bestaande. Stel dat we de kop `\rubriek` als volgt definiëren:

```
\definieerkop[rubriek][subparagraaf]
```

Het is na dit commando om het even of we de markering oproepen met de naam `rubriek` of `subparagraaf`. Dit komt meestal overeen met wat we wensen. We kunnen nu dus de voetregels ook als volgt instellen:

```
\stelvoettekstenin[hoofdstuk][rubriek]
```

Het kan echter voorkomen dat we wel degelijk `rubriek` als aparte markering willen. We zouden in principe kunnen volstaan met het opnieuw definiëren:

```
\definieermarkering[rubriek]
```

Omdat we echter graag zien dat een markering op het juiste moment automatisch wordt gereset, zullen we toch moeten aangeven op welk niveau de markering een rol speelt. De volledige serie commando's ziet er dan ook als volgt uit.

```
\definieerkop[rubriek][subparagraaf]
```

```
\definieermarkering[rubriek]
```

```
\koppelmarkering[rubriek][subparagraaf]
```

Let wel: deze exercitie is alleen nodig als we `rubriek` apart (naast `subparagraaf`) willen gebruiken! Meestal volstaat het eerste commando. Na deze serie commando's is het volgende mogelijk:

```
\stelvoettekstenin[subparagraaf][rubriek]
```

Het laatste van de drie commando's in de bovenstaande serie is nieuw:

```
\koppelmarkering[.1.][.2.]
```

```
.1.    naam
```

```
.2.    naam
```

De tegenhanger van dit commando is:

```
\ontkoppelmarkering[...]
```

```
...    naam
```

Het is trouwens ook mogelijk zelf, analoog aan de standaardkoppeling in koppen, markeringen te koppelen. In dat geval is de nieuwe markering synoniem aan de oude.

```
\definieermarkering[zomaarwat][paragraaf]
```

Dergelijke manipulaties met markeringen veronderstellen van de gebruiker een meer dan gemiddeld inzicht in de werking van `CONTEXT`.

9.5 Terugverwijzen (refereren)

In de tekst kunnen referentiepunten worden vastgelegd, bijvoorbeeld bij hoofdstuktitels, paragraaftitels, figuren en tabellen. Referentiepunten worden in de tekst tussen [] opgenomen. Er mogen meerdere referentiepunten, gescheiden door een comma, worden opgegeven. De (elders) in de tekst opgenomen referenties kunnen worden opgeroepen met de commando's:

```
\in{.1.}{.2.}[ref]
```

```
.1. tekst
```

```
\op{.1.}{.2.}[ref]
```

```
.1. tekst
```

```
\over{...}[ref]
```

```
... tekst
```

Een verwijzing naar een pagina, tekst (nummer) of beide kan worden aangemaakt met:

```
\paginareferentie[ref]
```

```
\tekstreferentie[ref]{...}
```

```
... tekst
```

```
\referentie[ref]{...}
```

```
... tekst
```

Het commando `\in` levert het nummer van het hoofdstuk, de paragraaf, een figuur, een tabel enz. Het commando `\op` levert een paginanummer en `\over` levert een titel, bijvoorbeeld van een hoofdstuk. De eerste twee commando's hebben een optioneel tweede argument, dat achter het paginanummer of the titel wordt geplaatst.

In het onderstaande voorbeeld wordt verwezen naar eerdere paragrafen en bladzijden:




In paragraaf `\in[terugverwijzen]`, getiteld `\over[terugverwijzen]`, wordt aangegeven hoe een referentie kan worden gedefinieerd. Deze paragraaf begint op bladzijde `\op[terugverwijzen]` en maakt deel uit van hoofdstuk `\in[verwijzingen]`.

Dit wordt:

In paragraaf 9.5, getiteld “Terugverwijzen (refereren)”, wordt aangegeven hoe een referentie kan worden gedefinieerd. Deze paragraaf begint op bladzijde 180 en maakt deel uit van hoofdstuk 9.

Het bovenstaande voorbeeld had ook als volgt kunnen worden gemaakt:

In `\in{paragraaf}[terugverwijzen]`, getiteld `\over[terugverwijzen]`, wordt aangegeven hoe een referentie kan worden gedefinieerd. Deze paragraaf begint op `\op{bladzijde}[terugverwijzen]` en maakt deel uit van `\in{hoofdstuk}[verwijzingen]`.

Deze manier van oproepen verdient de voorkeur, zeker bij het zetten van interactieve teksten. In het laatste geval wordt namelijk ook het woord aanklikbaar, terwijl in het eerste geval een symbool  verschijnt waarop kan worden geklikt. Dit symbool geeft tevens de richting van de verwijzing aan: terug  of vooruit .

De richting kan men overigens ook in woorden oproepen met het commando `\ergens`. Zo vinden we het hoofdstuk over tekstonderdelen hiervoor en gaan we later in op beschrijvingen.

```
\ergens{.1.}{.2.}[ref]
.1.   tekst
```

Dit commando krijgt twee teksten mee. Een eerdere alinea was ingevoerd als:

De richting kan men overigens ook in woorden oproepen met het commando `\type {\ergens}`. Zo vinden we het hoofdstuk over tekstonderdelen `\ergens {hiervoor} {hierna} [tekstonderdelen]` en gaan we `\ergens {eerder} {later} [beschrijvingen]` in op beschrijvingen.

Het volgende commando krijgt geen tekst mee maar levert die zelf. De gegenereerde teksten zijn in te stellen met `\stellabeltekstin` (zie bladzijde 142).

```
\oppagina[ref]
```

Op de plaats waar naar wordt verwezen kunnen meerdere referenties worden aangemaakt, bijvoorbeeld:

```
\hoofdstuk[eerste,tweede,derde]{Eerste, tweede en derde}
```

Er kan naar dit hoofdstuk worden terugverwezen met `\in[eerste]`, `\in[tweede]` of `\in[derde]`. Bij een grote tekst bestaat echter het gevaar dat men dezelfde verwijzing op verschillende plaatsen definieert. Het verdient daarom aanbeveling bij grote teksten een ordening aan te brengen, in het voorbeeld: `[fig:eerste]`, `[par:eerste]`, `[tab:eerste]` enz.

Naast een eigen ordening kan ook automatisch een ordening worden aangebracht. Bij `\stelrefererenin` kan daartoe een prefix worden ingesteld. Men kan zelf een prefix definiëren, maar ook kan men `CONTEXT` er zelf een laten genereren. Als bijvoorbeeld `prefix=alfa`, dan wordt vanaf dat moment achter de schermen voor iedere verwijzing het woord `alfa` gezet. Een in termen van geheugengebruik wat meer efficiënte benadering is die waarbij het systeem een prefix genereert. Iedere keer als `prefix=+` wordt gegeven, wordt een nieuwe prefix aangeemaakt. Door `prefix=-` wordt het toevoegen van prefixen stopgezet.

In veel gevallen levert het frequent instellen van een prefix een slordig brondocument op. Het is daarom mogelijk om `CONTEXT` per sectie en nieuwe prefix te laten instellen. Wanneer we bijvoorbeeld voor elk hoofdstuk een nieuwe prefix willen hebben, dan kunnen we met `\stelkopin` een `+` toekennen aan de parameter `prefix`. De hoofdstuk verwijzing zelf wordt niet geprefixed en is dus ondubbelzinnig beschikbaar. De verwijzingen binnen het hoofdstuk worden echter automatisch van een prefix voorzien, en krijgen dus een lokaal karakter. Wanneer het hoofdstuk zelf een label heeft, dan wordt dat de nieuwe prefix, in andere gevallen wordt een nummer gebruikt. Stel dat we het volgende tegenkomen:

```
\stelkopin[hoofdstuk][prefix=+]
\chapter[texwereld]{De wereld van \TeX}
```

In dit hoofdstuk kunnen we nu veilig verwijzingen aanmaken en gebruiken, zonder gevaar van conflicten met verwijzingen in andere hoofdstukken. Neem bijvoorbeeld de volgende figuur:

```
\plaatsfiguur[hier][fig:wereldkaart]{Een kaart van de \TeX\ wereld}{...}
```

In dit hoofdstuk kunnen we naar deze figuur verwijzen met:

```
\in {figuur} [fig:wereldkaart]
```

Buiten dit hoofdstuk is de verwijzing echter:

```
\in {figuur} [texwereld:fig:wereldkaart]
```

Bij het ‘opzoeken’ van een verwijzing met `\in`, `\op` enz., wordt eerst gekeken of er een lokale verwijzing is (dus met prefix) en als deze niet wordt gevonden of er sprake is van een globale verwijzing (dus zonder prefix). Mocht men er onverhoopt niet uitkomen: verwijzingen kunnen in de marge zichtbaar gemaakt worden met het eerder beschreven commando `\versie[voorlopig]`.

Het kan voorkomen dat men midden in een document een globale verwijzing wil aanmaken, bijvoorbeeld naar een inhoudsopgave of een index. In dat geval kan men met een `-:` in de verwijzing aangeven dat de verwijzing *geen* prefix heeft, bijvoorbeeld `[-:inhoud]`. Vooral bij interactieve teksten heeft het prefix-mechanisme zijn nut. Het volgende voorbeeld (zie ook hierboven) krijgen we gegarandeerd een globale verwijzing:

```
\plaatsfiguur[hier][-:fig:wereldkaart]{Een kaart van de \TeX\ wereld}{...}
```

Het genereren van verwijzingen kan worden stopgezet en gestart met het commando:


```
\stelrefererenin[.,.,.=.,.,.]
status          start stop
sectienummer   ja nee
prefix         + - tekst
interactie     label tekst alles symbool
breedte        maat
links          commando
rechts         commando
converteerfile ja nee klein groot
scheider       tekst
autofile       ja nee pagina
globaal        ja nee
```

Hierbij hebben de instellingen `\sectienummer` betrekking op het weergeven van deze nummers bij een verwijzing naar een bladzijde.

In interactieve documenten kunnen we naar andere documenten verwijzen. In dat geval worden, wanneer `converteerfile` de waarde `yes` heeft, de namen van externe files automatisch geconverteerd naar hoofdletters, iets dat soms nodig is voor CDROM producties. We komen daar elders op terug.

Referenties uit een andere file kunnen worden geladen met het commando:

```
\gebruikreferenties[.,.,.,.,.]
...   file
```

Met `links` en `rechts` stelt men in wat rond een met `\over` gegenereerde verwijzing staat, standaard zijn dat quotes. Met `interactie` kan men aangeven of een verwijzing wordt gezet als *paragraaf 1.2*, *paragraaf, 1.2* of een symbool, zoals **■**.

Wat is precies een referentie. We zagen reeds dat wanneer we bij een hoofdstuk een verwijzing definiëren, we op een andere plaats het hoofdstuknummer, de hoofdstuktitel en het paginanummer waarop het hoofdstuk begint kunnen oproepen. Voor interne doeleinden is ook het echte paginanummer beschikbaar; dit volgnummer speelt in interactieve documenten achter de schermen een belangrijke rol. Er is ook nog een extra verwijzing mogelijk, maar die wordt voorsnog niet gebruikt in de standaard-commando's. Hieronder wordt gedemonstreerd hoe men de afzonderlijke elementen van een referentie kan specificeren en oproepen.

```
\referentie[mijn verwijzing]{{Kijk}}{eens}{aan}}
```

De afzonderlijke elementen krijgt men met `\ref`:

<code>p</code>	het gezette paginanummer	<code>\ref[p][mijn verwijzing]</code>	183
<code>t</code>	de tekstreferentie	<code>\ref[t][mijn verwijzing]</code>	Kijk
<code>r</code>	het pagina-volgnummer	<code>\ref[r][mijn verwijzing]</code>	209
<code>s</code>	de subtekstreferentie	<code>\ref[s][mijn verwijzing]</code>	eens
<code>e</code>	de extra tekstreferentie	<code>\ref[e][mijn verwijzing]</code>	aan

In een papieren document is een verwijzing statisch: een nummer of wat tekst. In een interactief document kan achter een verwijzing echter iets verborgen zitten, bijvoorbeeld een sprong naar

de locatie waar naar wordt verwezen. In aanvulling op de besproken commandos' `\in` en `\op` is er het commando `\naar`, waarmee men naar een bepaalde plaats kan springen. Dit commando geeft geen nummer of tekst terug en heeft in die zin nauwelijks een typografische of tekstinhoudelijke betekenis.

Binnen `CONTEXT` is een veelheid aan interactieve mogelijkheden geïntegreerd in het referentie mechanisme. De belangrijkste reden hiervoor ligt in het feit dat op die manier een besparing op het aantal commando's mogelijk is en tegelijk een meer natuurlijke wijze van omgaan met dit soort boven-papierlijke zaken. Er is dan ook geen wezenlijk verschil tussen het gaan naar hoofdstuk 3, het activeren van een `JAVASCRIPT`, het gaan naar een ander document, of het verzenden van een ingevuld formulier.

Een direct gevolg van deze verwevenheid is dat men niet gebonden is aan een referentie, maar zonodig er meerdere kan opgeven. De volgende verwijzing is dan ook volledig legaal:

```
... zie \in{paragraaf}[vliegwiel,StartVideo{vliegwiel 1}] ...
```

Dit commando genereert zoals we verwachten het nummer van de paragraaf. Bovendien kunnen we, wanneer het document als interactief document wordt gezet, op de woorden *paragraaf nr* klikken en zo naar de aangegeven plaats gaan. Daarnaast wordt, zodra die plek is bereikt, een video gestart, getiteld *vliegwiel 1*. Willen we dit soort dingen comfortabel realiseren, dan ontkomen we niet aan een geïntegreerd referentiemechanisme.

Stel dat men in een stijl voor algemeen gebruik iets wil doen met een verwijzing, maar dat (de naam van) die verwijzing zelf nog niet bekend is. In het volgende voorbeeld willen we vanuit een menu een video opstarten:

```
\startinteractiemenu[rechts]
  \but [vorigepagina]   vorige   \\
  \but [volgendepagina] volgende \\
  \but [ToonEenVideo]  video     \\
  \but [SluitDocument] stoppen  \\
\stopinteractiemenu
```

We kunnen nu op elk moment vastleggen welke video we willen activeren door `ToonEenVideo` te definiëren:

```
\definieerreferentie[ToonEenVideo][StartVideo{een hele leuke video}]
```

Deze referentie is op elk moment opnieuw in te stellen, of zelfs te wissen:

```
\definieerreferentie[ToonEenVideo][]
```

```
\definieerreferentie[...][ref,...]
...   naam
```

9.6 Regelnummers

1 Een speciaal geval van verwijzingen zijn die naar regelnummers. Daarbij kunnen meerdere be-
 2 reiken door elkaar lopen. Dit kan een verwarrende invoer opleveren. Nietwaar? Een verwijzing
 3 naar een regel kan resulteren in een enkel nummer of een bereik. Een verwijzing wordt opge-
 4 roepen met `\inregel`, waarbij automatisch het woord *regel(s)* wordt toegevoegd. Hier hebben
 5 we drie verwijzingen: regel 1-2, regel 2, regel 2 en als laatste verwijzing 3

Met `\startregel..\stopregel` krijgt men een bereik als het meerdere regels betreft, in
 geval van `\eenregel` krijgt men alleen het beginnummer. We zien hier dat de automatisch
 gegenereerde tekst kan worden onderdrukt door tussen accolades een eigen tekst mee te
 geven.

`\startregelnummers`

Een speciaal geval van verwijzingen zijn die naar regelnummers. `\startregel`
`[regel:a]` Daarbij kunnen meerdere bereiken door elkaar lopen. `\startregel`
`[regel:b]` Dit kan een verwarrende invoer opleveren. `\stopregel [regel:a]`
`\startregel [regel:c]` Nietwaar? `\stopregel [regel:c]` `\stopregel [regel:b]` Een
 verwijzing naar een regel kan resulteren in een enkel nummer of een bereik.
`\eenregel [regel:d]` {Een verwijzing wordt opgeroepen met `\type {\inregel}`,
 waarbij automatisch het woord `{\em regel(s)}` wordt toegevoegd.} Hier hebben
 we drie verwijzingen: `\inregel [regel:a]`, `\inregel [regel:b]`, `\inregel`
`[regel:c]` en `\inregel {als laatste verwijzing} [regel:d]`

`\stopregelnummers`

```
\startregel[ref] ... \stopregel
```

```
\eenregel[ref]
```

```
\inregel[ref]
```

9.7 Voorgedefinieerde verwijzingen

Het kan saai en zelfs gevaarlijk voor de consistentie zijn als we veel referenties met hetzelfde
 label moeten verwerken, zoals bijvoorbeeld **figuur** in `\in{figuur}[fig]`. Het kan bijvoorbeeld
 wenselijk zijn, naderhand elke **figuur** te vervangen door **Figuur**. In dat geval kan het volgende
 commando uitkomst bieden:

```
\definieerreferentieformaat[...][..,..=..,..]
```

```
...      naam
links   tekst
rechts  tekst
tekst   tekst
label   naam
```

Stel dat we de volgende definities hebben:

```
\definieerreferentieformaat [indemo] [links=(,rechts=),tekst=demo]
\definieerreferentieformaat [indemos] [links=(,rechts=),tekst=demos]
\definieerreferentieformaat [endemo] [links=(,rechts=),tekst=en]
```

We hebben nu de beschikking over drie nieuwe commando's:

```
\indemo [demo:b]
\indemo {wat tekst} [demo:b]
\indemos {wat tekst} [demo:b] \indemo {en wat meer tekst} [demo:c]
\indemos [demo:b] \endemo [demo:c]
```

Dit ziet er gezet uit als:

```
demo (BB)
wat tekst (BB)
wat tekst (BB) en wat meer tekst (CC)
demos (BB) en (CC)
```

In plaats van de parameter `tekst` kunnen we ook `label` gebruiken om een voorgedefinieerd label op te roepen. Aan `commando` kan men een `commando` toevoegen, standaard is dat `\in`.

9.8 Registers

Een naslagwerk zonder register loopt grote kans niet serieus te worden genomen. Het is daarom in `CONTEX` mogelijk een of meer registers te genereren. De ingangen van de registers worden naar een file geschreven. Deze file is voor `TEX` niet zonder meer bruikbaar. Het PERL script `TEXUTIL` zet deze file om in `TEX`-commando's.

Een register wordt gedefinieerd met het commando:

```
\definieerregister[.1.][.2.]
.1.
.2.   naam meervoud
```

Er is een aantal commando's beschikbaar om ingangen te creëren in een register en om een register op te roepen. Er is standaard één register gedefinieerd met:

```
\definieerregister[index][indices]
```

Een ingang wordt gecreëerd met het commando:

```
\register[.1.]{.+ .2.+ .}
.1.   tekst
.2.   tekst
```

Een ingang kan uit maximaal drie niveaus zijn opgebouwd. Dergelijke geneste ingangen worden gescheiden door een `+` of `&`. Het onderstaande voorbeeld illustreert dit.

```
\index{leesplank}
\index{leesplank+aap}
\index{leesplank+noot}
```

Als een ingang op een bepaalde manier moet worden gezet, kunnen problemen ontstaan met het ordenen. \TeX -commandos, zoals `\s1` en `\kap` worden bij het sorteren omzeild. Als in een ingang zetcommando's zijn opgenomen die het woord genereren, dan kan beter het iets uitgebreidere commando worden gebruikt. Tussen [] wordt dan een ASCII-string opgenomen die de alfabetische ordening bepaalt.

Als we bijvoorbeeld de afkortingen of logo's VVN en ANWB hebben gedefinieerd (zie paragraaf 9.2), dan dient een register-ingang er als volgt uit te zien:

```
\index[VVN]{\VVN}
\index[ANWB]{\ANWB}
```

Doen we dit niet, dan zullen VVN en ANWB worden gerangschikt onder \.

Een verwijzing binnen een index wordt gecreëerd met:

```
\ziregister[.1.]{.2.}{..+.3.+..}
.1.   tekst
.2.   tekst
.3.   tekst
```

Ook dit commando kent een 'pure' variant waarbij de commando-loze ingang tussen vierkante haken wordt meegegeven.

Een register wordt opgeroepen met het commando:

```
\plaatsregister[.....=.....]
..=.. zie p 188: \stelregisterin
```

Het volgende commando genereert een register met titel:

```
\volledigeregister[.....=.....]
..=.. zie p 188: \stelregisterin
```

De wijze waarop een register wordt gezet wordt ingesteld met `\stelregisterin`. Als bovenaan de tekst het commando `\versie[voorlopig]` wordt gegeven, dan wordt in de marge aangegeven waar een ingang is opgenomen (zie paragraaf 2.5).

```

\stelregisterin[.1.][.2.][.,.,.=.,.]
.1.
.2.          naam
n            getal
balanceren  ja nee
uitlijnen   ja nee
letter      normaal vet schuin vetschuin type kap klein... commando
paginaletter normaal vet schuin vetschuin type kap klein... commando
tekstletter normaal vet schuin vetschuin type kap klein... commando
aanduiding  ja nee
koppeling   ja nee
sectienummer ja nee
criterium   sectie lokaal alles dee1
afstand     maat
symbool     1 2 ... n a ... geen
interactie  paginanummer tekst
expansie    ja commando nee
refereren   aan uit
commando    \commando#1
plaats      links midden rechts
maxbreedte  maat
onbekendeverwijzing leeg geen

```

Standaard wordt een volledig register gegenereerd. Het is echter mogelijk een deelregister op te vragen. In dat geval dient het `criterium` een waarde te krijgen. Met `aanduiding` legt men vast of een letteraanduiding wordt geplaatst in de index zelf. Als `refereren=aan`, dan wordt voor elk register een pagina verwijzing aangemaakt voor iedere aanduiding. Voor indexen zijn dat verwijzingen als: `index:a` of `index:w`. We kunnen deze automatisch aangemaakte verwijzingen bijvoorbeeld gebruiken om te verwijzen naar de pagina waar de a-ingangen beginnen. De genoemde `commando`'s staan een ruime layout toe. Dit betekent dat het volgende is toegestaan:

```
\hoofdstuk{Ziezo}
```

```
\paragraaf{Hoezo Ziezo}
```

```
\index{hoezo}
```

```
\index{ziezo}
```

Het gebruik van het achtervoegsel `zo ...`

Tussen `\hoofdstuk` en `\paragraaf` mogen we niets plaatsen, omdat anders de verticale spatiering wordt verstoord. De lege regel na `\index` geeft echter geen problemen en wordt netjes afgevangen. Mocht het achterliggende mechanisme problemen geven, bijvoorbeeld binnen zelf gedefinieerde `macro`'s, dan kan ook het wat meer directe `commando` worden gebruikt:

```
[setup schrijfnaarregister* is niet gedefinieerd]
```

Eventueel kan met `expansie` worden ingesteld of men het argument wil expanderen; standaard gebeurt dit namelijk niet (meer hierover op pagina 177).

In deze handleiding is een register opgenomen met `commando`'s. Dit register is gedefinieerd en vervolgens geïnitieerd met:

```
\definieerregister [macro] [macros]
\stelregisterin    [macro] [aanduiding=uit]
```

Waarna we ingangen tegenkomen als:

```
\macro{\tex{hoofdstuk}}
\macro{\tex{paragraaf}}
```

We kunnen het bij dit hoofdstuk behorende deel van dit register oproepen met het volgende commando (het commando `\tex` plaatst een `\` voor het woord, maar wordt bij sorteren weggefilterd):¹⁹

```
\plaatsregister[macro]                % of: \plaatsmacro
  [criterium=hoofdstuk,n=2,voor=,na=]
```

en krijgen dan:

Standaard worden de paginanummers geplaatst. Met `symbool` is echter een alternatieve weergave in te stellen. Met afstand wordt de spatïering tussen het woord en het eerste nummer of symbool ingesteld.

symbool	weergave
a	a b c d
n	1 2 3 4
1	• • • •
2	■ ■ ■ ■

Tabel 9.4 Alternatieven voor paginanummers in registers.

Een waarschuwing is hier op zijn plaats. Het aanmaken van een goed register is een kunst apart. Het is bijvoorbeeld maar de vraag of het register achterin deze handleiding zo'n goed register is.

De tot nu toe besproken index ingangen geven aan op welke bladzijde een onderwerp begint. Naast dergelijke ingangen kunnen we ook een gebied markeren, met:

```
[setup start*register is niet gedefinieerd]
```

Tusseliggende ingangen van dezelfde orde worden niet in het register opgenomen.

```
\startregister[eindeloos]{eindeloos}
..... een eindeloos lang verhaal .....
\stopregister[eindeloos]
```

In de meeste gevallen zal de vormgeving van een register eenvoudig zijn. In een naslagwerk kan echter de behoefte bestaan aan het differentiëren van ingangen. Hoewel het in zo'n geval

¹⁹ Natuurlijk zijn ook `\plaatsmacro` en `\volledigemacros` beschikbaar.

de moeite waard is meerdere registers te maken, kan ook door gerichte vormgeving meer helderheid worden bereikt. We zagen reeds de twee verschijningsvormen:

```
\index          {ingang}
\index [sleutel] {ingang}
```

In beide gevallen vinden we *ingang* terug in de index, maar in het tweede geval gesorteerd onder de *s* van *sleutel*. De vormgeving volgt in dit geval de broodtekst.

Een afwijkende vormgeving is alleen zinvol als ze consistent wordt toegepast. Om die reden ondersteund CONTEXT het de vormgevingsprefix.

```
\index[vorm::]      {ingang}
\index[vorm::steutel]{ingang}
\index              {vorm::ingang}
\index[sleutel]     {vorm::ingang}
\index[vorm::]      {vorm::ingang}
\index[vorm::sleutel]{vorm::ingang}
```

De `vorm::` aanduiding geeft aan op welke wijze een paginanummer of het trefwoord zelf wordt vormgegeven. Daarbij wordt de volgorde waarin de ingangen worden gesorteerd geen geweld aangedaan. De specificatie tussen `[]` heeft betrekking op het paginanummer, die voor de `ingang` op de `ingang` zelf.

```
\stelregisterin[index][vorm][paginaletter=vet,tekstletter=schuin]
```

We kunnen zonder problemen meerdere verschijningsvormen gebruiken, al dan niet hetzelfde voor paginanummer en `ingang`.

```
\stelregisterin[index][nb][paginaletter=vet]
\stelregisterin[index][hm][paginaletter=schuin]
```

Neem bijvoorbeeld:

```
\index[nb::]{vierkantswortel}
\index[hm::wortel]{{\sqrt{2}}}
```

In het eerste geval wordt het paginanummer achter *vierkantswortel* *vet* gedrukt, in het tweede geval *schuin*.

Een beetje grote index, met meerdere verwijzingen per `ingang`, kan erg oncomfortabel zijn, met name in een voor het scherm ontworpen document. Je kunt daar immers niet je vinger bij de indexbladzijde houden. Een bijzondere optie bij registers is daarom het koppelen van ingangen. Het is op deze manier mogelijk door de tekst te springen, van `ingang` naar `ingang`. Een koppeling wordt geactiveerd met:

```
\stelregisterin[index][koppeling=ja]
```

Op deze manier wordt een mechanisme geactiveerd dat zowel in het register (◀▶) als in de tekst (◀ **woord** ▶) verwijzingen plaatst, afhankelijk van de beschikbaarheid van alternatieven. Vanuit het register wordt naar het eerste, de middelste of de laatste voorkomen van de `ingang` gesprongen.

Dit mechanisme werkt alleen op het eerste niveau; subingangen worden overgeslagen. Als op het woord zelf wordt geklikt, gaan we terug naar het register. Omdat in de tekst het woord moet voorkomen, gebruiken we het commando:

```
\gekoppelderegister[.1.]{.2.}
```

```
.1.   tekst
```

```
.2.   tekst
```

dus bijvoorbeeld `\gekoppeldeindex{ziezo}`. De koppelingen moeten van te voren worden geladen met het commando:

```
\koppelregister[...]
```

```
...   naam
```

Dit commando wordt normaal gesproken automatisch uitgevoerd als nodig en is dus alleen nodig in uitzonderlijke situaties.

10

Beschrijvingen

Beschrijvingen

10.1 Inleiding

Afgezien van hoofdstuk- en paragraafkoppen, vinden we in een document —we hebben het hier niet over romans— verschillende elementen met een herkenbare structuur en vormgeving. Zo zijn er bijvoorbeeld definities, al dan niet genummerd, opsommingen en citaten. Een voordeel van $\text{T}_{\text{E}}\text{X}$ en dus $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ is dat het coderen van dergelijke elementen zodanig kan plaatsvinden, dat consistentie min of meer gegarandeerd is. Bovendien hoeven we ons tijdens het invoeren van een tekst niet bezig te houden met de vormgeving, maar kunnen we ons beperken tot de inhoud. Dit neemt niet weg dat we soms wel willen ingrijpen.

In dit hoofdstuk staan we stil bij een aantal van die herkenbare elementen. Het kan geen kwaad eens flink te experimenteren met de hier gepresenteerde commando's, vooral omdat op die manier het denken in structuur wordt geoefend. Mits goed toegepast, zal blijken dat het gebruik van de hier beschreven commando's er toe bijdraagt dat in de tekst niet of nauwelijks opmaakcommando's hoeven te staan.

10.2 Definities

Een lijst met definities kan worden vormgegeven met behulp van `\doordefinieren`. Bij dit commando moet als eerste parameter een naam worden opgegeven.

```
\doordefinieren[...] [.,.,.=.,.,.]
...      naam
..=...   zie p 195: \steldoordefinierenin
```

Een voorbeeld van een definitie is:

```
\doordefinieren[begrip] [plaats=boven,kopletter=vet]
\begrip{ikoon}
```

Sinds de introductie van grafische besturingssystemen zijn ikonen onlosmakelijk verbonden met computers. \par

Hierna zijn de meeste alternatieven weergegeven.

ikoon

Sinds de introductie van grafische besturingssystemen zijn ikonen onlosmakelijk verbonden met computers.

Ze worden door sommige gebruikers van computers met een bijna religieus fanaticisme aanbeden.

ikoon

ikoon Ze worden geacht herkenbaar te zijn voor iedere gebruiker, maar aangezien ze hun beeltenis ontleen aan een snel veranderende werkelijkheid, is deze herkenbaarheid relatief.

ikoon Het 8-bits karakter van computers heeft er lang voor gezorgd dat niet-latijnse schriften nauwelijks werden ondersteund door besturingsystemen. Daar is sinds kort verandering in gekomen.

ikoon Wat eerst een handicap leek —namelijk geen latijnse karakters— wordt nu een kracht. De duizenden karakters van de chinese taal zijn immers even zoveel ikonen. Duizenden woorden en concepten zijn daar reeds in karakters vervat.

ikoon Ligt het niet voor de hand te verwachten dat culturen waar de taal veel beeldender is op termijn hun relatieve achterstand in computergebruik weten om te zetten in een voorsprong, bepaald door hun beeldende taal?

ikoon Welk westers kind zal achter het beeldscherm nog de gedateerde vuilnisbak herkennen waarin men bestanden kan weggooien? Gelukkig staat er vaak nog onder dat het een vuilnisvat is. Een in de taal opgenomen karakter voor vuilnisvat is aanzienlijk minder gevoelig voor veroudering.

In het vijfde voorbeeld is het begrip aaneelkaar geplaatst en gedefinieerd als:

```
\doordefinieren
  [begrip]
  [plaats=aaneelkaar,kopletter=vet,breedte=ruim,monster={ikoon}]
\begrip{ikoon}
```

Wat eerst een handicap leek |<|namelijk geen latijnse karakters|>| wordt nu een kracht. De duizenden karakters van de chinese taal zijn immers even zoveel ikonen. Duizenden woorden en concepten zijn daar reeds in karakters vervat. \par

In het laatste voorbeeld is aan hang de waarde ruim meegegeven. Deze instelling heeft alleen zin bij links of rechts definiëren. Als aan breedte in plaats van een getal passend of ruim wordt meegegeven, dan wordt de breedte van het monster als uitgangspunt genomen. Als geen monster is opgegeven, dan wordt het ‘te definiëren’ woord als monster beschouwd. Stel zonodig afstand in. De instelling uitlijnen heeft betrekking op de tekst. De instelling marge is alleen van toepassing wanneer de definitie in de marge of aan elkaar wordt geplaatst.

Het na de definitie beschikbare commando heeft dus de vorm:

```
\doordefinitie{.1.}.2.\par
.1. tekst
.2. tekst
```

Het is mogelijk enkele algemene kenmerken van het doordefiniëren in te stellen met:

```

\steldoordefinierenin[.,.,.,.,.][.,.=.,.]
...          naam
kopletter    normaal vet schuin vetschuin type kap klein... commando
letter       normaal vet schuin vetschuin type kap klein... commando
kleur        naam
breedte      passend ruim maat
afstand      maat
monster      tekst
tekst        tekst
uitlijnen    links midden rechts
marge        standaard ja nee maat
plaats       links rechts boven aanelkaar inmarge inlinker inrechter hangend
hang         passend ruim getal
voor         commando
tussen       commando
na           commando
springvolgendein ja nee

```

De instellingen van een definitie kunnen worden gewijzigd met het onderstaande *commando*, dat dezelfde vorm heeft als `\doordefinieren`:

```
\steldoordefinierenin[naam][instellingen]
```

Als men meerdere paragrafen in de definitie wil opnemen moet gebruik gemaakt worden van een `\start-``\stop` constructie:

```
\startbegrip{Icoon}
```

Een ikoon is een religieuze afbeelding die in bepaalde religies van grote waarde kan zijn voor de gelovigen.

Om een of andere reden heeft het begrip ikoon ook zijn weg gevonden naar de computerwereld, waar het een eigen leven is gaan leiden.

```
\stopbegrip
```

De *commando's* vangen zo goed en kwaad als dat kan lege regels af.

```

\startdoordefinitie{...} ... \stopdoordefinitie
...      tekst

```

Vaak zal men in een tekst meerdere soorten definities gebruiken. Het volgende voorbeeld laat zien dat we consistentie in de definities kunnen bevorderen door gebruik te maken van overerving.

```

\doordefinieren
[definitie]
[tekst=Definitie, kopletter=\sc,
 plaats=aanelkaar, breedte=ruim,
 wijze=perhoofdstuk, sectienummer=ja]

```

```
\doordefinieren
  [lemma]
  [definitie]
  [tekst=Lemma]
```

10.3 Doornummeren

Het is mogelijk bepaalde aanduidingen automatisch te nummeren, bijvoorbeeld vragen, opmerkingen en definities. Een dergelijke aanduiding wordt gedefinieerd met:

```
\doornummeren[...1...][.2.][...=...]
```

```
.1.   naam
.2.   naam
..=.. zie p 197: \steldoornummerenin
```

De op te geven naam bestaat uit letters. Na het commando `\doornummeren` zijn de volgende commando's beschikbaar:

```
\naam
\subnaam
\subsubnaam
\subsubsubnaam
```

waarbij naam staat voor de opgegeven naam.

```
\doornummering...\par
...   tekst
```

Er kunnen maximaal 4 niveaus worden ingesteld. Conversie heeft betrekking op het laatste niveau. Als een tekst wordt ingesteld, dan gaat deze vooraf aan het nummer.

Het is mogelijk het nummeren in te stellen of opnieuw te starten met het commando:

```
\setdoornummering{getal}
\resetdoornummering
```

Met de `start` parameter kan een startnummer worden ingesteld. Houd er rekening mee, dat de doornummercommando's de teller ophogen, dus om op 5 te starten, moet 4 worden ingesteld. Ook kunnen (sub)nummers worden opgehoogd, en wel met de commando's:

```
\volgendedoornummering
\volgendesubdoornummering
\volgendesubsubdoornummering
```

Het onderstaande voorbeeld illustreert het gebruik van `\doornummeren`. Na het getoonde commando kan een opmerking worden geplaatst na `\opmerking`.

```
\doornummeren
  [opmerking]
```



```
[plaats=boven,
tekst=Opmerking,
tussen=\blanko,
voor=\blanko,
na=\blanko]
```

Enkele voorbeelden van opmerkingen zijn:

Opmerking 1

Een ‘opmerking’ is na het bovenstaande commando op vier niveaus beschikbaar: `\opmerking`, `\subopmerking` en `\subsubopmerking`.

Opmerking 2

Dit commando lijkt op het commando `\doordefinieren`.

De kenmerken van het doornummeren stelt men in met `\steldoornummeren` in. Veel instellingen komen overeen met de eerder beschreven definities omdat doornummeren eigenlijk een speciaal geval is van doordefiniëren.

```
\steldoornummeren[naam][instellingen]
```

```
\steldoornummeren[...][...][...]=...]
```

...	<i>naam</i>
..=..	zie p 193: <code>\doordefinieren</code>
plaats	links rechts <u>boven</u> aaneelkaar inmargin linker inrechter hangend
tekst	<i>tekst</i>
niveaus	<i>getal</i>
conversie	<u>cijfers</u> letters Letters romeins Romeins
wijze	<u>pertekst</u> <i>persectie</i>
blokwijze	<u>ja</u> nee
sectienummer	ja nummer nee
scheider	<i>tekst</i>
afsluiter	<i>tekst</i>
koppeling	<i>naam</i>
koppelwijze	globaal <u>lokaal</u>
nummer	nee <i>naam</i>
titeluitlijnen	nee <u>ja</u>
start	<i>getal</i>

Ook de sub- en subsubnummeringen kunnen worden ingesteld, bijvoorbeeld:

```
\steldoornummeren[voorbeeld][kopletter=vet]
\steldoornummeren[subvoorbeeld][kopletter=schuin]
```

Evenals bij het in de vorige paragraaf besproken commando is ook hier weer een `\start-``\stop-`paar beschikbaar:

```
\startdoornummering ... \stopdoornummering
```

Soms is het nummer overbodig. Bijvoorbeeld als we per hoofdstuk nummeren en in een bepaald hoofdstuk maar één doornummering tegenkomen. In dat geval kan door middel van [-] worden aangegeven dat men geen nummer wil:

Opmerking

Omdat de aanroep hier `\opmerking[-]` was, is er *geen* nummer geplaatst. Evenals bij andere commando's het geval is, kan men ook hier tussen [] een referentie opgeven, zodat naar het nummer kan worden verwezen. Met `nummer=nee` kan het weergeven van het nummer (tijdelijk) worden uitgezet.

Dit commando is (zinvol) te combineren met dat om blokken tekst te verplaatsen. Een voorbeeld hiervan is te vinden in paragraaf ???. In dit voorbeeld wordt ook gedemonstreerd op welke wijze een koppeling met een andere genummerde tekst kan worden gelegd. Dergelijke koppelingen hebben alleen betekenis bij interactieve teksten, waar automatisch over en weer verwijzingen worden gegenereerd.

Het kan voorkomen dat een doornummering in meerdere gedaanten voorkomt. In dat geval kunnen doornummeringen worden gebruikt die de tellers van elkaar erven. We laten dit aan een voorbeeld zien.

```
\doornummeren[eerste]
```

```
\eerste De doornummering \type {eerste} is een uniek geval. We zien dat we met een argument kunnen volstaan. Standaard staan het label en nummer links.
```

```
\doornummeren[tweede][eerste][plaats=rechts]
```

```
\tweede De \type {tweede} erft de tellers van \type {eerste}, maar wordt rechts geplaatst. In geval van drie argumenten is het eerste de kopie en het tweede het origineel.
```

```
\doornummeren[derde,vierde][plaats=inrechter]
```

```
\derde De doornummeringen \type {derde} en \type {vierde} zijn beide uniek en worden in de rechtermarge geplaatst.
```

```
\vierde Beiden zijn in \char 233 {} \char 233 {} n opdracht gedefinieerd maar hebben eigen tellers en zijn op geen enkele manier gekoppeld.
```

```
\doornummeren[vijfde,zesde][eerste]
```

```
\vijfde De doornummeringen \type {vijfde} en \type {zesde} erven beiden de eigenschappen en tellers van \type {eerste}.
```

```
\zesde Pas op: erven van \type {tweede} is niet toegestaan, omdat \type {tweede} geen origineel is! \par
```

Het lijkt wat ingewikkeld, maar hopelijk verduidelijkt het onderstaande een en ander:

eerste 1

De doornummering `eerste` is een uniek geval. We zien dat we met een argument kunnen volstaan. Standaard staan het label en nummer links.

De tweede erft de tellers van eerste, maar wordt rechts geplaatst. In geval **tweede 2** van drie argumenten is het eerste de kopie en het tweede het origineel.

De doornummeringen derde en vierde zijn beide uniek en worden in de rechtermarge geplaatst. **derde 1**

Beiden zijn in één opdracht gedefinieerd maar hebben eigen tellers en zijn op geen enkele manier gekoppeld. **vierde 1**

vijfde 3

De doornummeringen vijfde en zesde erven beiden de eigenschappen en tellers van eerste.

zesde 4

Pas op: erven van tweede is niet toegestaan, omdat tweede geen origineel is!

Het is mogelijk een doornummering te koppelen aan een andere doornummering. We kunnen bijvoorbeeld vragen aan antwoorden koppelen. Klikken op een vraag resulteert in springen naar het antwoord en andersom. De nummers moeten wel synchroon lopen, dat wil zeggen dat na iedere vraag een antwoord moet staan. Eventueel kunnen de antwoorden in blokken worden geplaatst en later worden opgeroepen. Vergeet daarbij zonodig niet te resetten.

```
\doordefinieren [vraag] [koppeling=antwoord]
```

```
\doordefinieren [antwoord] [koppeling=vraag]
```

De wijze van nummeren wordt beïnvloed door het instellen van `wijze` en `sectienummer`. Wil men wat meer controle, dan kan men zelf de prefix regelen. De volgende instellingen gaan bijvoorbeeld om het bestaande prefix mechanisme heen: er wordt per tekst genummerd, maar alleen het paragraafnummer wordt gezet. Met de - geven we aan geen prefix te willen.

```
\doornummeren
```

```
[definitie]
```

```
[tekst=DEFINITIE,
```

```
links={\kopnummer[- ,paragraaf].},
```

```
letter=klein,
```

```
breedte=ruim,
```

```
sectienummer=nee,
```

```
wijze=pertekst]
```

```
\doornummeren [theorema] [definitie] [tekst=THEOREMA]
```

```
\doornummeren [lemma] [definitie] [tekst=LEMMA]
```

Ter illustratie geven we hier drie equivalente oplossingen voor het op deze wijze manipuleren van het paragraafnummer:

```
\stelsectiein[sectie-3][vorigennummer=nee]\kopnummer[paragraaf]
```

```
\bepaalkopnummer[paragraaf]\huidigekopnummer
```

```
\kopnummer[- ,paragraaf]
```

De eerste oplossing vereist nogal wat kennis van de onderliggende mechanismes, de tweede is vooral bedoeld voor situaties waarin men het nummer wil manipuleren, en de derde is bedoeld

voor alledaags gebruik. Overigens geeft `\kopnummer` zonder argument het laatst ingestelde huidige kopnummer.

10.4 Doorspringen

Inspringende opsommingen, bijvoorbeeld dialogen, kunnen worden gezet met het commando:

```
\doorspringen[...][...,...=...,...]
...      naam
..=...   zie p 200: \steldoorspringenin
```

Na dit commando zijn de commando's `\naam`, `\subnaam` en `\subsubnaam` beschikbaar.

```
\doorsprong...\par
...      tekst
```

Standaardwaarden kunnen worden ingesteld met het commando:

```
\steldoorspringenin[.,.,.,.,.][...,...=...,...]
...      naam
letter   normaal vet schuin vetschuin type kap klein... commando
kopletter normaal vet schuin vetschuin type kap klein... commando
breedte  passend maat
tekst    tekst
monster  tekst
voor     commando
na       commando
afstand  maat
scheider tekst
```

Het is mogelijk de bij `\doorspringen` meegegeven instellingen te wijzigen met het commando:

```
\steldoorspringenin[naam][instellingen]
```

Een voorbeeld van het gebruik van `\doorspringen` wordt hieronder gegeven:

```
\steldoorspringenin
  [monster={rijm m},
   scheider={:},
   afstand=.5em]

\doorspringen[ra][tekst=rijm a]
\doorspringen[rb][tekst=rijm b]
\doorspringen[rc][tekst=rijm c]

\startopelkaar
\ra ra, ra, ra
\ra het is het jaar van de \citeer{k}
```

```
\rb koryan en kampioen
\rc kwadraat en korporaāl
\rb waarom zou men dat toch doen
\rc ja, waar is dat normaal
\stopopeelkaar
```

Dit resulteert in:

```
rijm a : ra, ra, ra
rijm a : het is het jaar van de 'k'
rijm b : koryan en kampioen
rijm c : kwadraat en korporaāl
rijm b : waarom zou men dat toch doen
rijm c : ja, waar is dat normaal
```

Een serie doorspring-commando's kan worden omringd met de commando's:

```
\startdoorspringen
\stopdoorspringen
```

10.5 Doorlabelen

In een tekst kunnen genummerde verwijzingen worden opgenomen, bijvoorbeeld verwijzingen naar transparanten of video's. Een verwijzing wordt gedefinieerd met het commando:

```
\doorlabelen[...] [.,.,.=.,.]
...      naam
tekst    tekst
plaats   inmarge intekst
wijze    pertekst persectie perhoofdstuk
blokwijze ja nee
kopletter normaal vet schuin vetschuin type kap klein... commando
kopkleur naam
voor     commando
na       commando
```

Als plaats kan worden meegegeven *intekst* en *inmarge*. Na dit commando zijn een aantal commando's beschikbaar:

```
\resetnaam
\verhoognaam
\volgendenaam
\huidigenaam[referentie]
```

De [referentie] bij \huidigenaam is facultatief. Na

```
\doorlabelen[video] [tekst=video,plaats=inmarge]
```

levert \video een eervolle vermelding in de marge op. Het commando \huidigevideo had **video 10.1** hier volgnummer 1 opgeleverd. Het label kan ook worden opgeroepen met:

```
\doorlabel[ref]
```

In ons geval is dat `\video`. De waarden toegekend aan `aan voor` en `na` worden rond het label uitgevoerd.

10.6 Opsommingen

Opsommingen worden automatisch voorafgegaan door speciale symbolen, oplopende nummers of elkaar opvolgende letters. Het symbool en de wijze van nummeren kunnen worden ingesteld (zie tabel 10.1). Tevens kunnen enkele layout-kenmerken worden ingesteld, wat in de volgende paragraaf wordt beschreven. Opsommingen kunnen tot vier niveaus diep plaatsvinden.

instelling	gevolg	instelling	gevolg
n	1, 2, 3, 4	1	dot (•)
a	a, b, c, d	2	streepje (–)
A	A, B, C, D	3	sterretje (★)
KA	A, B, C, D	4	driehoekje (▷)
r	i, ii, iii, iv	5	rondje (◦)
R	I, II, III, IV	6	groter rondje (○)
KR	I, II, III, IV	7	nog groter rondje (◯)
m	1, 2, 3, 4	8	vierkantje (◻)
g	α, β, γ		
G	A, B, Γ		

Tabel 10.1 Mogelijke markeringen om de items in een opsomming te kenmerken.

Het commando om een opsomming te plaatsen luidt:

```
\startopsomming[instelling]
```

```
\som .....
```

```
\som .....
```

```
\stopopsomming
```

Zodat:

Welke van de twee onderstaande uitspraken is juist?

```
\startopsomming[A]
```

```
\som Zwolle is een stad, want er zijn resten van muren te vinden en er  
zijn verdedigingswerken.
```

```
\som Zwolle is een gemeente want er staat een gemeentehuis.
```

```
\stopopsomming
```

leidt tot:

Welke van de twee onderstaande uitspraken is juist?

- A. Zwolle is een stad, want er zijn resten van muren te vinden en er zijn verdedigingswerken.
- B. Zwolle is een gemeente want er staat een gemeentehuis.

De onder 1 tot en met 8 gebruikte symbolen kunnen worden ingesteld met het commando `\definieersymbool` (zie paragraaf ??) en de conversie van het volgnummer met `\definieerconversie` (zie paragraaf ??). Bijvoorbeeld:

Bevatten de volgende uitspraken een kern van waarheid?

```
\definieersymbool[1][{$\diamond$}
```

```
\startopsomming[1]
```

```
\som De Stille Veerkade in Zwolle heet zo omdat daar vroeger een veerpont
aanlegde.
```

```
\som De Stille Veerkade ontleent haar naam aan het feit dat het vroeger nu
niet bepaald een drukke straat was.
```

```
\stopopsomming
```

levert:

Bevatten de volgende uitspraken een kern van waarheid?

- ◇ De Stille Veerkade in Zwolle heet zo omdat daar vroeger een veerpont aanlegde.
- ◇ De Stille Veerkade ontleent haar naam aan het feit dat het vroeger nu niet bepaald een drukke straat was.

De instellingen n , i , enz. zijn gerelateerd aan conversies. Dat wil zeggen dat alle conversies worden geaccepteerd, ook die hier niet zijn genoemd. Neem bijvoorbeeld:

α . een g voor griekse letters

β . een G voor griekse hoofdletters

Als de instelling, inclusief de vierkante haken wordt weggegelaten, dan wordt het bij het niveau passende symbool geplaatst. In de meeste situaties zal dit de bedoeling zijn. Deze niveauafhankelijke symbolen zijn ingesteld met:

```
\definesymbol [1] [{\symbol[bullet]}]
```

```
\definesymbol [2] [{\symbol[dash]}]
```

met als nadere symbooldefinities:

```
\definesymbol [bullet] [{$\bullet$}
```

```
\definesymbol [dash] [{$-$}]
```

Het volgende voorbeeld toont een andere wijze van instellen. Dit keer koppelen we direct een symbool aan een niveau van opsommen.

```
\definieersymbool [waarom] [?]
```

```
\definieersymbool [daarom] [!]
```

```
\stelopsommingin[1][symbool=waarom]
```

```
\stelopsommingin[2][symbool=daarom]
```

Toegepast op een opsomming krijgen we:

? het voor jou een 'vraag'
! maar voor mij een 'weet'
? nietwaar

Het commando `\stelopsommingin` kan ook worden gebruikt voor het vastleggen van het al dan niet inspringen en de eventuele witruimte tussen de onderdelen:

```
\stelopsommingin[.1.][. . . , .2. , . . .][. . . , . = . . . , . . .]
.1.          getal elk
.2.          standaard n*ruim n*aanelkaar opelkaar vanelkaar afsluiter aansluitend
             opmarge inmarge autointro los paragraaf intekst
marge       nee standaard maat
breedte     maat
afstand     maat
factor      getal
items       getal
start       getal
voor        commando
tussen      commando
na          commando
links       tekst
rechts      tekst
kopvoor     commando
kopna       commando
kopletter   normaal vet schuin vetschuin type kap klein... commando
marletter   normaal vet schuin vetschuin type kap klein... commando
symletter   normaal vet schuin vetschuin type kap klein... commando
afsluiter   tekst
n           getal
symbool     getal
uitlijnen   links rechts normaal
springvolgendein ja nee
```

Deze argumenten kunnen in verschillende combinaties voorkomen, zoals:

Welke van de volgende uitspraken is juist:

```
\startopsomming[a, opelkaar][afsluiter=:]
\som 1991 is een schrikkeljaar
\som 1992 is een schrikkeljaar
\som 1993 is een schrikkeljaar
\som 1994 is een schrikkeljaar
\stopopsomming
```

wat wordt:

Welke van de volgende uitspraken is juist:

- a: 1991 is een schrikkeljaar
- b: 1992 is een schrikkeljaar

c: 1993 is een schrikkeljaar

d: 1994 is een schrikkeljaar

Beide argumenten zijn optioneel. De instelling `opelkaar` is één van de meest voor de hand liggende.

Welke van de volgende uitspraken is juist:

```
\startopsomming[n,opelkaar,inmarge]
\som      1991 is een schrikkeljaar
\som[ok]  1992 is een schrikkeljaar
\som      1993 is een schrikkeljaar
\som      1994 is een schrikkeljaar
\stopopsomming
```

dat resulteert in:

Welke van de volgende uitspraken is juist:

1. 1991 is een schrikkeljaar
2. 1992 is een schrikkeljaar
3. 1993 is een schrikkeljaar
4. 1994 is een schrikkeljaar

Het komt niet zelden voor dat voorafgaand aan een opsomming een zin staat die wordt afgesloten in de trant van “... *zien we hieronder:*”. In dat geval kan de instelling `intro` worden meegegeven, zodat bij paginaovergangen de inleidende zin wordt meegenomen naar de volgende bladzijde. Hierboven had dat er als volgt uitgezien:

```
\startopsomming[n,opelkaar,inmarge,intro]
```

De mogelijke instellingen, naast de in de voorbeelden genoemde, zijn in tabel 10.2 weergegeven.

instelling	gevolg
standaard	standaard instellingen
opelkaar	geen witruimte tussen onderdelen
vanelkaar	wel witruimte tussen onderdelen
aansluitend	geen wit voor en na de opsomming
alinea	geen wit voor de opsomming
n^* aanelkaar	weinig witruimte na het symbool
n^* ruim	extra witruimte na het symbool
inmarge	markering tegen de kantlijn
opmarge	aanduiding in de kantlijn
afsluiter	afsluiter na de aanduiding
intro	niet overgaan op nieuwe bladzijde
kolommen	in twee kolommen

Tabel 10.2 Instellingen bij `\stelopsommingin`.

In het laatste voorbeeld zien we dat achter het commando `\som` een verwijzing kan worden opgenomen, zodat we naar antwoord 2 kunnen verwijzen met het commando `\in[ok]`.

We kunnen, indien gewenst doornummeren door in plaats van een markering het woord verder op te nemen:

```
\startopsomming[verder,intro]
\som 1995 is een schrikkeljaar
\stopopsomming
```

wat de voortgezette (hier zinloze) opsomming levert:

5. 1995 is een schrikkeljaar

Een eveneens zinloos voorbeeld toont ons dat `verder` ook op hogere niveaus tot resultaten leidt:

- **eerste**
 - a. alfa
 - b. beta
- **tweede**
 - c. gamma
 - d. delta

Voor de volledigheid volgen hier de opdrachten (in dit document hebben we `kopletter=vet`):

```
\startopsomming[1,opelkaar]
\kop eerste \par
  \startopsomming[a]
  \som alfa \som beta
  \stopopsomming
\kop tweede \par
  \startopsomming[verder]
  \som gamma \som delta
  \stopopsomming
\stopopsomming
```

Als we het trefwoord `kolommen` meegeven, dan worden de opsommingen in twee of meer kolommen gezet. Het aantal kolommen kan worden ingesteld met de trefwoorden `een`, `twee` (standaard), `drie` of `vier`.

```
\startopsomming[n,kolommen,vier]
\som alfa \som beta \som gamma \som delta \som epsilon
\som zeta \som eta \som theta \som kappa \som lambda
\stopopsomming
```

We zien hier tevens dat het niet per se nodig is de items onder elkaar op te geven.

- | | | | |
|----------|------------|----------|------------|
| 1. alfa | 4. delta | 7. eta | 10. lambda |
| 2. beta | 5. epsilon | 8. theta | |
| 3. gamma | 6. zeta | 9. kappa | |

Wanneer we, zoals hierboven, de items nummeren, kan de daarvoor beschikbare ruimte aan de krappe kant zijn. In dat geval kunnen we `ruim` als trefwoord meegeven.

- | | | | |
|----------|------------|----------|------------|
| 1. alfa | 4. delta | 7. eta | 10. lambda |
| 2. beta | 5. epsilon | 8. theta | |
| 3. gamma | 6. zeta | 9. kappa | |

De tegenhanger van `ruim` is `aaneelkaar`. Ook hier kunnen we een factor meegeven. Hieronder is `2*aaneelkaar` gebruikt.

?Wat gebeurt hier nou weer?

We kunnen de optie `ruim` misbruiken om eenvoudige tabellen te maken. Het blijft daarbij wel gokken naar de noodzakelijke breedte.

Dit levert de wat vreemde keuzemogelijkheid:

```
\startopsomming[4*ruim,opelkaar]
\sym {ja} dit is mooi
\sym {nee} dit is lelijk
\stopopsomming
```

```
ja dit is mooi
nee dit is lelijk
```

Het komt vaak voor dat binnen een opsomming de eerste regels van de onderdelen worden gevolgd door een tekstblok. In dat geval kan het commando `\kop` worden gebruikt. De wijze waarop een kop wordt weergegeven, kan worden ingesteld met het hiervoor besproken `\stelopsommingin`.

```
\stelopsommingin[elk][kopletter=vet]
```

```
\startopsomming[n]
```

```
\kop Koppen in opsommingen
```

```
Na het commando \type{\kop} moet een lege regel worden opgenomen.
Doet men dit niet, dan gaat het mis.
```

```
\stopopsomming
```

Dit wordt:

1. Koppen in opsommingen

Na het commando `\kop` moet een lege regel worden opgenomen. Doet men dit niet, dan gaat het mis.

Als we hier `\som` hadden gebruikt, dan was de kop in een gewone letter gezet. Bovendien had, als dat zo was uitgekomen, een pagina-overgang kunnen ontstaan tussen de kopregel en de tekst, wat bepaald geen fraai gezicht is.

```
\kop[ref,..]
```

Soms bestaat een opsomming uit slechts één item. In dat geval kunnen de commando's `\startopsomming` en `\stopopsomming` worden weggelaten en wordt de bij niveau 1 behorende aanduiding gebruikt.

```
\som Het heeft geen zin hier een beschouwing op te nemen over de
      achtergrond van de naam van dit commando. \par
```

In plaats van `\par` mag natuurlijk ook een lege regel worden opgenomen. Het resultaat is verbluffend:

```
? Het heeft geen zin hier een beschouwing op te nemen over de achtergrond van de naam van
  dit commando.
```

Alleen de tekst, volgend op het commando en afgesloten met een lege regel of `\par`, wordt inspringend gezet.

Als men (tijdelijk) een oude nummering wil handhaven, kan gebruik worden gemaakt van `\sub` in plaats `\som`. Dit kan van pas komen bij discussiestukken, waarbij we in tussentijdse versies de nummering niet al te veel willen verstoren.

1. Deze opsomming wordt voorafgegaan door `\startopsomming[n,opelkaar]`.
- +1. Dit item wordt voorafgegaan door `\sub`, de andere items door `\som`.
2. De opsomming wordt natuurlijk afgesloten met `\stopopsomming`.

Samengevat luiden de belangrijkste commando's:

```
\startopsomming[.,.,.,.,.][.,.,.=.,.,.] ... \stopopsomming
...   a A KA n N m r R KR getal verder standaard n*ruim n*aaneelkaar opelkaar afsluiter
      aansluitend opmarge inmarge intro kolommen
..=.. zie p 204: \stelopsommingin
```

```
\som[ref,..]
```

```
\sub[ref,..]
```

Als aanvulling op `\som` is `\sym` beschikbaar. Met dit commando kan men 'even vlot tussendoor' een ingesprongen tekst achter een eigen symbool plaatsen.

```
\sym{...}
```

Het commando `\mar` komt overeen met `\som`, maar heeft een argument. Dit argument wordt in de marge geplaatst (standaard in een typeletter). Dit commando kan worden gebruikt om opmerkingen bij een item te plaatsen.

```
\mar[ref,...]{...}
```

Een wat vreemd alternatief is:

```
\its[ref,...]
```

```
\ran{...}
```

We illustreren dit aan de hand van een voorbeeld ontleend aan de NTG jubileum enquête:

oneens eens

- ◦ ◦ ◦ Ik kan T_EX niet missen.
- ◦ ◦ ◦ Ik zal altijd T_EX blijven gebruiken.
- ◦ ◦ ◦ Ik verwacht binnen enkele jaren een redelijk alternatief.
- ◦ ◦ ◦ Ik gebruik naast T_EX ook andere zet/opmaakpakketten.
- ◦ ◦ ◦ Ik gebruik T_EX nauwelijks.
- ◦ ◦ ◦ Ik ben op zoek naar een ander systeem.

Dit ziet er in broncode als volgt uit. Let op de instellingen. Het verdient de voorkeur dergelijke instellingen los te koppelen van de vragen en slechts eenmalig te doen.

```
\startopsomming[5,opelkaar][breedte=8em,afstand=2em,items=4]
```

```
\ran {oneens\hss eens}
```

```
\its Ik kan \TeX\ niet missen.
```

```
\its Ik zal altijd \TeX\ blijven gebruiken.
```

```
\its Ik verwacht binnen enkele jaren een redelijk alternatief.
```

```
\its Ik gebruik naast \TeX\ ook andere zet|/|opmaakpakketten.
```

```
\its Ik gebruik \TeX\ nauwelijks.
```

```
\its Ik ben op zoek naar een ander systeem.
```

```
\stopopsomming
```

Voor interactieve toepassingen is er:

```
\but[ref]
```



Dit commando komt overeen met `\som` maar levert een aanklikbare aanduiding. Het argument is dus een verwijzing.

Het onderstaande voorbeeld toont een aantal van de genoemde commando's. We zien bovendien nog een alternatief `\nop`.

```
\nop
```

? kopzorgen

1. van sommen heeft hij
op school noppes begrepen
- ++ 2. omdat slechts marginaal is uitgelegd
- +2. wat al die substantiële doch
- # eigenaardige symbolen betekenen

Dit rijtje is verkregen door:

```

\startopsomming
\kop kopzorgen
    \startopsomming[n,opelkaar]
    \som van sommen heeft hij
    \nop op school noppes begrepen
    \mar{++} omdat slechts marginaal is uitgelegd
    \sub wat al die substantiële doch
    \sym{\#} eigenaardige symbolen betekenen
\stopopsomming
\stopopsomming

```

Tijdens de verwerking van een opsomming wordt het aantal items in de opsomming bijgehouden. Dit geldt voor alle varianten. Bij een volgende verwerking wordt deze informatie gebruikt om zonodig beter te kunnen vaststellen op welk punt in een opsomming mag worden overgegaan op een nieuwe bladzijde. Wanhoop dus niet als een opsomming wat vreemd wordt afgebroken, meestal komt het in een volgende verwerkingslag weer in orde.

Een tweetal tips tot slot. Gebruik bij meerdere alinea's per opsomming `\kop` in plaats van `\som` wanneer de eerste alinea uit slechts één regel bestaat. Het commando `\kop` zorgt er namelijk voor dat niet wordt afgebroken tussen de eerste twee alinea's. Gebruik daarnaast de optie `[intro]` als er slechts één regel boven de opsomming staat. Dit laatste kan worden geautomatiseerd door de instelling:

```
\stelopsommingin[elk][autointro]
```

10.7 Items

Het commando `\items` is een variant op de commando's ten behoeve van opsommingen. Met dit commando kunnen eenvoudig genummerde of anders gemarkeerde lijsten worden gemaakt. De lijsten kunnen zowel horizontaal als vertikaal worden gezet. Het commando heeft de vorm:

```
\items{alternatief 1,alternatief 2,...,alternatief N}
```

In plaats van een alternatief kan ook - worden meegegeven. In dat geval wordt wel ruimte gereserveerd, maar niets gezet.

De vorm van de lijst wordt ingesteld met het commando:

<code>\stelitemsin[...=...]</code>	
plaats	<u>links</u> rechts inmarge boven onder
symbool	1 2 ... n a ... <i>tekst</i> geen
breedte	<i>maat</i>
n	<i>getal</i> onbekend
voor	<i>commando</i>
tussen	<i>commando</i>
uitlijnen	links rechts <u>midden</u> marge
na	<i>commando</i>

Zowel het aantal (n) als de breedte worden automatisch berekend. Ten behoeve van een meer eenduidige vormgeving kan het echter soms wenselijk zijn de breedte van de verschillende items in te stellen. Het aantal is alleen van belang als er geen alternatieven worden meegegeven.

Hieronder worden enkele voorbeelden gegeven. Hieruit blijkt dat de instellingen ook direkt achter `\items` kunnen worden meegegeven.

```
\items[plaats=links]{eerste,tweede,derde}
```

- eerste
- tweede
- derde

```
\items[plaats=onder]{eerste,tweede,derde}
```

eerste	tweede	derde
◦	◦	◦

```
\items[plaats=rechts,breedte=2cm]{eerste,tweede,derde}
```

- | | | |
|--------|---|--|
| eerste | ◦ | |
| tweede | ◦ | |
| derde | ◦ | |

```
\items[plaats=boven,breedte=6cm,uitlijnen=links]{eerste,tweede,derde}
```

- | | | |
|--------|--------|-------|
| ◦ | ◦ | ◦ |
| eerste | tweede | derde |

```
\items[plaats=inmarge]{eerste,tweede,derde}
```

- eerste
- tweede
- derde

```
\items[plaats=links,n=2,symbool=5]{alfa,beta}
```

- alfa
- beta

```
\items[symbool=3,n=6,breedte=\hspace,plaats=boven]{aa,bb,cc,dd,ee,ff}
```

*	*	*	*	*	*
aa	bb	cc	dd	ee	ff

De instellingen achter `\items` komen overeen met die van `\stelitemsin`:

```
\items[...=...]{...}
..=.. zie p 211: \stelitemsin
```

10.8 Citaten

Het gebruik van quotes verschilt per taalgebied: ‚nederlands‘, ‘english’, ‚deutch‘, <francais> Een consistent gebruik van enkele en dubbele quotes wordt ondersteund door een aantal commando's. Een citaat kan tussen de lopende tekst worden opgenomen met:

```
\startcitaat[...,...] ... \stopcitaat
... n*links n*midden n*rechts
```

Dit commando is te vergelijken met `\startsmaller` en kent de zelfde instellingen. Om het citaat worden zo mogelijk (en nodig) dubbele quotes geplaatst, waarbij de quotes buiten de tekst vallen.

„In reclames citeert men graag ‚deskundigen‘. Zo zag ik pas een reclame waarin een wasmiddel wordt aanbevolen door de Nederlandse Vereniging van Huisvrouwen. Wat mij daarbij enigzins bevreemdde was dat de woordvoerder namens deze club een man was. Hij werd daarbij aangeduid als „directeur“. Het zal toch niet waar zijn dat deze vereniging wordt geleid door een man? ”

In dit voorbeeld komen we nog twee commando's tegen:

```
\startcitaat
```

```
In reclames citeert men graag \citeer{deskundigen}. Zo zag ik pas een
reclame waarin een wasmiddel wordt aanbevolen door de Nederlandse
Vereniging van Huisvrouwen. Wat mij daarbij enigzins bevreemdde was dat
de woordvoerder namens deze club een man was. Hij werd daarbij aangeduid
als \citaat{directeur}. Het zal toch niet waar zijn dat deze vereniging
wordt geleid door een man?
```

```
\stopcitaat
```

Hierbij levert `\citaat` dubbele en `\citeer` enkele quotes.

```
\citaat{...}
... tekst
```



```
\citeer{...}
...      tekst
```

Deze commando's passen zich aan de taal aan. Men krijgt dus andere quotes in een engelse tekst dan in een duitse of franse tekst. Het lettertype is in te stellen met:

```
\stelciterenin[...]=...]
```

voor	<i>commando</i>
na	<i>commando</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
kleur	<i>naam</i>
plaats	tekst <u>marge</u>

De plaats van een punt, binnen of buiten een citaat, of twee maal, is wat arbitrair. De meningen (en adviezen) hierover lopen uiteen.

Hij zei: „Dat is een fiets” Waarop zij zei: „Zo heb je niets”.

De quotes in citeren passen zich aan de taal aan. Het is daarom belangrijk dat het wisselen van taal op de juiste wijze plaatsvindt:

```
\citaat {Hij antwoorde: \fr \citaat {Je ne parle pas fran\c cais}.}
\citaat {Hij antwoorde: \citaat {\fr Je ne parle pas fran\c cais}.}
\citaat {\fr Il r'epondait: \citaat{Je ne parle pas fran\c cais}.}
\fr \citaat {Il r'eponde: \citaat{Je ne parle pas fran\c cais}.}
```

Let op het subtiele verschil:

```
„Hij antwoorde: «Je ne parle pas français».”
„Hij antwoorde: „Je ne parle pas français”.”
„Il répondait: «Je ne parle pas français».”
«Il réponde: «Je ne parle pas français».»
```

Als we andere quotes willen, dan is dat in te stellen met bijvoorbeeld:

```
\setuplanguage
[nl]
[leftquote=\upperleftsinglequote,
leftquotation=\upperleftdoublequote]
```

Omwille van de consistentie, kan zo'n instelling het best worden opgenomen in de file `cont-sys.tex`, samen met andere lokale instellingen. Dit document is gezet met hoge quotes.

<code>\lowerleftsinglequote</code>		<code>\lowerrightsinglequote</code>	
<code>\lowerleftdoublequote</code>	„	<code>\lowerrightdoublequote</code>	”
<code>\upperleftsinglequote</code>	’	<code>\upperrightsinglequote</code>	’
<code>\upperleftdoublequote</code>	”	<code>\upperrightdoublequote</code>	”
<code>\upperleftsinglequote</code>	‘	<code>\upperrightsinglequote</code>	‘
<code>\upperleftdoublequote</code>	“	<code>\upperrightdoublequote</code>	“

11

Lijnen en
kaders

Lijnen en kaders

11.1 Inleiding

Zo sterk als \TeX kan omgaan met tekst, zo zwak is zij in grafische zaken. Een uitzondering zijn rechte lijnen, dat wil zeggen, zolang ze horizontaal of vertikaal zijn. Dat wil niet zeggen dat met \TeX niets grafisch kan worden gedaan, integendeel: \TeX kan prettig samenwerken met `METAPOST` en biedt mogelijkheden genoeg om extern aangemaakte afbeeldingen in een tekst op te nemen.

In dit hoofdstuk introduceren we een aantal commando's die betrekking hebben op lijnen. We zullen zien dat met name `\omlijnd` op vele manieren inzetbaar is. De functionaliteit van dit commando is ook in veel andere commando's beschikbaar. Opletten dus.

11.2 Losse lijnen

Verreweg het eenvoudigste commando om lijnen te tekenen is:

```
\haarlijn
```

Bijvoorbeeld:

```
\haarlijn
```

```
Is het nou in \citeer {roodkapje} of in \citeer {de wolf en de zeven  
geitjes} dat de wolf de put in gaat met een maag vol stenen?
```

```
\haarlijn
```

levert als resultaat:

Is het nou in 'roodkapje' of in 'de wolf en de zeven geitjes' dat de wolf de put in gaat met een maag vol stenen?

Echt fraai oogt dit niet. Dit komt omdat een `haarlijn` een eigen regel krijgt. In paragraaf 11.4 is te lezen hoe het echt moet.

De werking van `\haarlijn` laat zich het best begrijpen als we de `\strut`'s zichtbaar maken. We doen dit met het commando `\toonstruts`.

Een strut is een karakter met maximale hoogte en diepte en geen breedte. Hier zien we wat tekst omgeven door twee gestrutte lijnen.

Het is ook mogelijk een lijn over de breedte van de actuele paragraaf te trekken:

```
\dunnelijn
```

of eventueel meerdere lijnen onder elkaar:

```
\dunnelijnen[.=..]
..=.. zie p 217: \steldunnelijnenin
```

Bijvoorbeeld:

```
\startopsomming[n]
\som Mijn eerste voorkeur is: \par \dunnelijnen[n=2]
\som Mijn tweede voorkeur is: \par \dunnelijnen[n=2]
\stopopsomming
```

Als we de `\par` (of lege regel) weglaten, dan sluiten de lijnen aan op de tekst. Vergelijk:

1. Mijn eerste voorkeur is:

2. Mijn tweede voorkeur is:

met:

1. Mijn eerste voorkeur is: _____

2. Mijn tweede voorkeur is: _____

In het tweede voorbeeld lieten we de `\par` weg:

```
\startopsomming[n]
\som Mijn eerste voorkeur is: \dunnelijnen[n=2]
\som Mijn tweede voorkeur is: \dunnelijnen[n=2]
\stopopsomming
```

Instellingen vinden plaats met:

```
\steldunnelijnenin[.=..]
interlinie      klein middel groot
n               getal
voor           commando
tussen         commando
na             commando
kleur          naam
achtergrondkleur naam
hoogte         maat max
diepte        maat max
variant       a b c d
lijndikte     maat
```

Er kunnen in de tekst dunne horizontale of verticale lijnen worden getrokken met de commando's:

```
\v1[...]  
...   getal
```

```
\h1[...]  
...   getal
```

Het argument is facultatief. Bij \v1 (|) kan een getal worden meegegeven dat de hoogte in regels aangeeft; bij \h1 (—) betreft die factor de breedte in em. Een \v1[2] produceert dus een verticale lijn met een hoogte van 2 regels.

11.3 Invullijnen

Ten behoeve van bijvoorbeeld vragenlijsten is het volgende commando beschikbaar:

```
\invullijnen[.,.,.=.,.,.]{.1.}{.2.}  
..=.. zie p 217: \stelinvullijnenin
```

met daarbij het instelcommando:

```
\stelinvullijnenin[.,.,.=.,.,.]
breedte        passend ruim maat
afstand        maat
voor           commando
na             commando
letter         normaal vet schuin vetschuin type kap klein... commando
n              getal
interlinie     klein middel groot
scheider       tekst
```

Zo leidt:

```
\invullijnen[n=2,breedte=2cm]{naam}
\invullijnen[n=2,breedte=2cm]{adres}
```

tot het lijstje:

```
naam _____
_____
_____
adres _____
_____
_____
```

Een variant hierop is het plaatsen van een invulregel aan het eind van een paragraaf. Dit gebeurt met behulp van de commando's:

```
\invulregel[...==...]\par
..==.. zie p 218: \stelinvulregelsin
```

```
\stelinvulregelsin[...==...]
```

```
breedte  maat
marge    maat
afstand  maat
voor     commando
na       commando
```

Het volgende voorbeeld verduidelijkt dit:

```
\invulregel[breedte=3cm] Consumenten krijgen te pas en te onpas
vragenlijsten en enquêtes voorgelegd. Het vermoeden bestaat dat
mensen die dergelijke vrijpostigheid vermoeden, daarvoor graag
een blokje omlopen. Bent u het daarmee eens?
```

Dit is overigens typisch een voorbeeld van een vraag waarbij men de ondervraagde kan laten kiezen uit alternatieven, maar hier laten we het antwoord open.

Consumenten krijgen te pas en te onpas vragenlijsten en enquêtes voorgelegd. Het vermoeden bestaat dat mensen die dergelijke vrijpostigheid vermoeden, daarvoor graag een blokje omlopen. Bent u het daarmee eens? _____

De volgende reeks voorbeelden laat zien wat de invloed is van de verschillende parameters.

```
\invullijnen[n=2,breedte=passend]{eerste}
\invullijnen[n=2,breedte=ruim]{eerste}
\invullijnen[n=2,breedte=3cm]{eerste}
\invullijnen[n=2,breedte=passend,afstand=.5em,scheider=:] {laatste}
\invullijnen[n=2,breedte=ruim,afstand=.5em]{eerste}{laatste}
```

```
eerste_____
```

eerste _____

eerste _____

laatste: _____

eerste _____

_____ laatste

11.4 Tekstlijnen

Een stuk tekst kan worden voorafgegaan en afgesloten met een lijn. In de bovenste lijn kan een tekst worden opgenomen. Het commando luidt:

```
\tekstlijn[.1.]{.2.}
.1.    boven onder
.2.    tekst
```

Een voorbeeld:

```
\tekstlijn[boven]{Instrumenten}
```

Het vermelden van de gebruikte instrumenten kan soms verrassend zijn. Op het meeslepende \citeer {Digging in the dust} gebruikt Peter Gabriel de {\em diembe}, {\em tama} en {\em surdu}. Naast deze instrumenten vinden we op de \kap{cd} waarop dit nummer staat nog de {\em doudouk}. Nog meer \citeer {onbekende} instrumenten worden gebruikt op zijn \kap{cd} \citeer {Passion}. Waarom kan dit alleen maar in Engeland?

```
\tekstlijn
```

Dit wordt:

— **Instrumenten** —

Het vermelden van de gebruikte instrumenten kan soms verrassend zijn. Op het meeslepende ‘Digging in the dust’ gebruikt Peter Gabriel de *diembe*, *tama* en *surdu*. Naast deze instrumenten vinden we op de CD waarop dit nummer staat nog de *doudouk*. Nog meer ‘onbekende’ instrumenten worden gebruikt op zijn CD ‘Passion’. Waarom kan dit alleen maar in Engeland?

Het gedrag van tekstlijnen kan worden ingesteld met het volgende commando. Hierbij heeft breedte betrekking op de lengte van de lijn voor de tekst.

```
\steltekstlijnenin[...]=...]
```

plaats	<code>\links</code> inmargin
voor	<i>commando</i>
na	<i>commando</i>
tussen	<i>commando</i>
breedte	<i>maat</i>
afstand	<i>maat</i>
korps	5pt ... 12pt klein groot
kleur	<i>naam</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
lijnkleur	<i>naam</i>

Er is ook een `\start-\stop` variant beschikbaar. De parameter `korps` wordt door dit alternatief afgehandeld.

```
\starttekstlijn[.1.]{.2.} ... \stoptekstlijn
```

.1.	boven onder
.2.	<i>tekst</i>

11.5 Onderstrepen

Onderstrepen is een verre van ideale manier om een deel van een tekst te laten opvallen. Desalniettemin kunnen we ook in `CONTEXT` onderstrepen. We gebruiken daarvoor:

```
\onderstreep{...}
```

... *tekst*

Een nadeel van dit commando is dat de woorden niet langer worden afgebroken. Dit is een wat minder prettig neveneffect. Als bonus wordt `\genest onderstrepen` ondersteund.

In de vorige alinea zagen we dat ook de spaties worden onderstreept. Willen we dit niet, iets dat in deze alinea het geval is, dan gebruiken we:

```
\onderstrepen{.. ... ..}
```

... *tekst*

De bovenstaande onderstreepte tekst was ingevoegd als:

```
\onderstreep {Een nadeel van dit commando is dat de woorden
\understreep {niet} langer worden afgebroken. Dit is een wat minder
prettig neveneffect. Als bonus wordt \onderstreep {genest onderstrepen}
ondersteund.}
```

`\onderstrepen` {In de vorige alinea zagen we dat ook de spaties worden onderstreept. Willen we dit niet, iets dat in deze alinea het geval is, dan gebruiken we:}

De tegenhanger van deze commando's zijn:

```
\overstreep{...}
...   tekst
```

```
\overstrepen{. . . .}
...   tekst
```

De belangrijkste bestaansreden voor dit commando zijn financiële overzichten, zoals:

```
post 1  1.420
post 2  3.182
totaal  4.602
```

Zo'n overzicht is gemaakt met:

```
\starttabulatie[|l|r|]
\NC post 1 \NC           1.420 \NC \NR
\NC post 2 \NC           3.182 \NC \NR
\NC totaal \NC \overstreep{4.602} \NC \NR
\stoptabulatie
```

Het aantal instellingen bij deze commando's is beperkt.

```
\stelonderstrepenin[.,.=.,.]
variant      a b c
lijndikte    maat
onderoffset  maat
bovenoffset  maat
lijnkleur    naam
```

De varianten zijn: variant a, variant b, variant c terwijl een andere lijndikte resulteert in: 1pt lijndikte, 2pt lijndikte.

Een deel van de tekst kan worden ~~doorgehaald~~ met het commando:

```
\doorstreep{...}
...   tekst
```

Dit commando ondersteund geen nesting. Losse woorden kunnen we ~~doorhalen~~ met:

```
\doorstrepn{.. ... ..}
... tekst
```

11.6 Omlijnen

Een tekst kan worden omlijnd met het commando `\omlijnd`. In de meest eenvoudige vorm is de aanroep:

```
\omlijnd{'In hokjes denken' kan uitmonden in 'in hokjes schrijven'.}
```

Dit geeft:

```
'In hokjes denken' kan uitmonden in 'in hokjes schrijven'.
```

De volledige definitie van het commando is:

```
\omlijnd[.,.,.=.,.,.]{...}
..=.. zie p 226: \stelomlijndin
... tekst
```

Zoals uit het volgende voorbeeld blijkt, mogen de instellingen worden weggelaten.

```
\omlijnd
 [hoogte=ruim]
 {Bestaat naast het woord ruimdenkend het woord ruimschrijvend?}
```

In plaats van een maat kan het woord `ruim` worden meegegeven:

```
Bestaat naast het woord ruimdenkend het woord ruimschrijvend?
```

Voor wat betreft de hoogte, is er geen verschil in de opties `passend` en `ruim`. Zo zal:

```
\hbox
 {\omlijnd[hoogte=ruim] {Is dit nu een passende maatregel?}}
 \hskip1em
 \omlijnd[hoogte=passend]{Of niet soms?}}
```

als uitvoer geven:

```
Is dit nu een passende maatregel? Of niet soms?
```

Om een gelijksoortige vormgeving van gekaderde naast niet gekaderde tekst mogelijk te maken, kan het kader aan of uit worden gezet.

```
welles nietus welles
nietis wellus nietis
```

De dikte van de lijnen kan worden ingesteld met het commando `\stelliijndiktein` (zie paragraaf ??).

Een in de lopende regel opgenomen omlijnde tekst wordt 'op' de regel geplaatst. Als men de tekst wil uitlijnen moet men het commando `\inlijnd` gebruiken.

het is `\omlijnd{erop}` of `\inlijnd{eronder}` met die lijnen

ofwel:

het is `erop` of `eronder` met die lijnen

Het is mogelijk een deel van het kader te tekenen. In dat geval dienen de te tekenen lijnen te worden gespecificeerd met `...kader=aan`.

Hieronder zijn ter vergelijking een aantal varianten weergegeven van het commando `\omlijnd`. Let vooral op de invloed van `offset`. Als geen maat wordt opgegeven, wordt de offset mede bepaald door de hoogte en diepte van de `\strut`, het denkbeeldige blokje met de maximale hoogte en diepte van een karakter dat geen breedte heeft. Als men binnen `\omlijnd` exact wil positioneren, dan dient men als offset geen op te geven (zie ook also tabel 11.1, 11.2 en 11.3). Als we de offset op geen of `overlay` zetten, dan wordt geen strut toegevoegd.

<code>breedte=passend</code>	<code>breedte=ruim</code>
<code>breedte=8cm, hoogte=1.5em</code>	
<code>offset=5pt</code>	
<code>offset=0pt</code>	
<code>offset=geen</code>	
<code>offset=overlay</code>	

De elders besproken commando's `\lbox`, `\cbox` en `\rbox` zijn goed te combineren met `\omlijnd`:

links van het midden	de gulden middenweg	rechts van het midden
<code>\lbox</code>	<code>\cbox</code>	<code>\rbox</code>

De middelste tekst is bijvoorbeeld gezet met het commando:

```
\omlijnd
[breedte=.2\hsize, hoogte=3cm]
{\cbox to 2.5cm{\hsize2.5cm de\\gulden\\middenweg}}
```

De optie `uitlijnen` maakt gebruik van dit mechanisme. Hierbij kan eventueel aan onder en boven een vertikaal uitvul-commando worden toegekend. Hieronder zijn enkele voorbeelden gegeven. Het tweede voorbeeld toont de standaardinstellingen.

```
\stelomlijndin[breedte=.2\hsize,hoogte=3cm,uitlijnen=midden]
\startcombinatie[4]
  {\omlijnd[onder=\vss,boven=\vss]{de\gulden\middenweg}}
  {\type{boven=\vss}\crlf\type{onder=\vss}}
  {\omlijnd[onder=\vss,boven=]{de\gulden\middenweg}}
  {\type{boven=} \crlf\type{onder=\vss}}
  {\omlijnd[onder=,boven=\vss]{de\gulden\middenweg}}
  {\type{boven=\vss}\crlf\type{boven=}}
  {\omlijnd[onder=,boven=]{de\gulden\middenweg}}
  {\type{boven=} \crlf\type{onder=}}
\stopcombinatie
```

de gulden middenweg	de gulden middenweg	de gulden middenweg	de gulden middenweg
boven=\vss onder=\vss	boven= onder=\vss	boven=\vss boven=	boven= onder=

Er kan een kleur of een raster achter de omlijnde tekst worden geplaatst door achtergrond in te stellen op `kleur` of `raster`. Vergeet niet eerst kleuren te activeren met (`\stelkleurenin[status=start]`).

In het	donker zijn
achtergrond=raster	achtergrond=raster achtergrondraster=0.7
alle katjes	grauw.
achtergrond=kleur	achtergrond=kleur achtergrondkleur=rood

Eventueel kan men een kader of achtergrond wat groter maken door het instellen van een `kaderoffset` en/of `achtergrondoffset`. Deze hebben geen invloed op de afmetingen. Naast rasters en kleuren kunnen ook willekeurige achtergronden worden gebruikt. Dit mechanisme wordt beschreven in paragraaf 6.7.

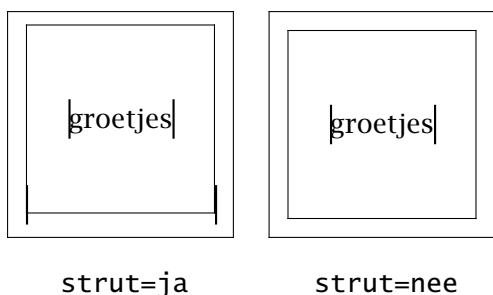
Het commando `\omlijnd` kan zelf ook als argument van `\omlijnd` worden meegegeven. We krijgen dan een kader binnen een kader.

```
\omlijnd
  [breedte=3cm, hoogte=3cm]
  {\omlijnd[breedte=2.5cm, hoogte=2.5cm]{groetjes}}
```

In dat geval staat het tweede kader wat hoger dan we misschien verwachten. Dit is een gevolg van het feit dat er automatisch een `\strut` wordt geplaatst, wat nodig is om bij naast elkaar geplaatste omkaderde teksten de regels op dezelfde hoogte te krijgen. We kunnen een `\strut` als volgt onderdrukken:

```
\omlijnd
  [breedte=3cm, hoogte=3cm, strut=nee]
  {\omlijnd[breedte=2.5cm, hoogte=2.5cm]{groetjes}}
```

Als we beide voorbeelden naast elkaar zetten zien we het verschil:



Een `\haarlijn` wordt normaal gesproken getrokken over de breedte (`\hsize`). Binnen `\omlijnd` wordt een haarlijn echter van de linker naar de rechter lijn getrokken.

```
\omlijnd[breedte=8cm, uitlijnen=midden]
  {hij maakt zich zorgen om haar lijn \haarlijn
   en zij over zijn haarlijn \haarlijn
   daar zit dus geen lijn in}
```

Levert dan ook:

hij maakt zich zorgen om haar lijn
en zij over zijn haarlijn
daar zit dus geen lijn in

Als geen breedte wordt opgegeven, dan worden verticale lijnen getrokken:

hun meningen staan dus	lijnrecht	tegenover elkaar
------------------------	-----------	------------------

Wat eenvoudig is verkregen met:

```
\omlijnd
  {hun meningen staan dus \haarlijn lijnrecht \haarlijn tegenover elkaar}
```

Men kan de vaste instellingen van `\omlijnd` veranderen met het commando:

```

\stelomlijndin[.,.,.=.,.,.]
hoogte           passend ruim maat
breedte          passend ruim maat
offset           geen overlay default maat
plaats           laag diepte
optie            geen leeg
strut            ja nee
uitlijnen        nee links rechts midden normaal hoog laag laho
onder            commando
boven            commando
kader            aan uit overlay
bovenkader       aan uit
onderkader       aan uit
linkerkader      aan uit
rechterkader     aan uit
kaderoffset      maat
kaderdiepte      maat
kaderhoek        rond recht
kaderstraal      maat
kaderkleur       naam
achtergrond      raster kleur geen voorgrond naam
achtergrondraster getal
achtergrondkleur naam
achtergrondoffset kader maat
achtergronddiepte maat
achtergrondhoek  rond recht
achtergrondstraal maat
diepte           maat
hoek             rond recht
straal           maat
leeg             ja nee
voorgroundkleur naam
...             tekst

```

Het commando `\omlijnd` wordt in veel andere commando's gebruikt. Dit is mede een gevolg van het feit dat het een vrij flexibel commando is. Met name het samenspel van `offset` en `strut` kan verwarrend werken. Ook de instellingen van breedte en hoogte zijn bepalend voor de wijze van omlijnen. Om die reden vatten we de gevolgen van hun instellingen nog eens samen in de tabellen 11.1, 11.2 en 11.3.

		offset			
		.25ex	0pt	geen	overlay
strut	ja	⏏		.	
	nee	□	.	.	

Tabel 11.1 De invloed van `strut` en `offset` bij `\omlijnd` (1).

		offset			
		.25ex	0pt	geen	overlay
strut	ja				
	nee				

Tabel 11.2 De invloed van strut en offset bij `\omlijnd` (2).

		breedte	
		passend	ruim (<code>\hsize=4cm</code>)
hoogte	passend		
	ruim		

Tabel 11.3 De invloed van hoogte en breedte bij `\omlijnd`.

iene
miene
mutte

Hoewel dat op het eerste gezicht niet meteen duidelijk zal zijn, kan `\omlijnd` zelf de breedte van een alinea tekst bepalen. Wanneer uitlijnen wordt ingesteld, dan zal achter de schermen eerst de tekst worden gezet, en zullen pas daarna de afmetingen worden ingesteld. Deze optie is vooral handig op bijvoorbeeld titelpagina's. In dit voorbeeld zijn de regelovergangen afgedwongen, maar dat is niet verplicht. Omdat deze optie gebruik maakt van het analyseren van reeds gezette informatie, werkt ze alleen als we te maken hebben met tekst.

```
\plaatsfiguur
[links]
{geen}
{\omlijnd[uitlijnen=midden]{iene\\miene\\mutte}}
```

Tot slot nog een opmerking over de `offset`. Deze heeft standaard de waarde `.25ex` en hangt dus af van het actuele lettertype. De volgende voorbeelden illustreren dit:

```
\hbox{\bf \omlijnd{test} \sl \omlijnd{test} \tfa \omlijnd{test}}
\hbox{\omlijnd{\bf test} \omlijnd{\sl test} \omlijnd{\tfa test}}
```

De waarde van `1ex` buiten `\omlijnd` is bepalend voor de offset. Dit blijkt in de praktijk goed te voldoen.

test	test	test
test	test	test

De verschillen zijn subtiel. De afstand tussen de blokjes wordt onder andere bepaald door het actuele lettertype en de afmetingen van het kader, door de offset en de strut.

T_EX kan alleen maar rechte lijnen trekken. Krommen anders dan karakters kunnen weliswaar worden opgebouwd met puntjes of korte rechte lijnen, maar alleen ten koste van grote DVI-files en lange procestijden. Ronde hoeken zijn vooralsnog dan ook alleen geïmplementeerd met behulp van POSTSCRIPT. Er zijn twee parameters die daarbij een rol spelen: `hoek` en `straal`. Als `hoek` rond is, worden ronde hoeken gezet.

We lopen de kantjes er vanaf.

Het is na de eerder gegeven voorbeelden (en die elders in deze handleiding) niet moeilijk voor te stellen dat er vrij veel mogelijk is (maar ook weer niet alles). Het is bijvoorbeeld mogelijk cirkelbogen te tekenen. In dat geval dient men als `straal` de halve breedte en/of hoogte op te geven.

Technisch gezien zijn de achtergrond, het kader en de tekst aparte onderdelen van een omlijnde tekst. Eerst wordt de achtergrond geplaatst, dan de tekst, en tot slot het kader. Een ronde hoek is een kenmerk van het kader en houdt als zodanig geen rekening met de tekst. Zolang echter de `straal` kleiner is dan de `offset` gaat het wel goed.

11.7 Kaderteksten

Een variant op omlijnen (zie 11.6) is het volgende commando:

```
[setup startkadertekst is niet gedefinieerd]
```

zoals blijkt uit:

```
\startkadertekst[links]
```

Uit onderzoek van C. van Noort (1993) is gebleken dat `intermezzo`'s niet altijd het gewenste effect hebben.

```
\stopkadertekst
```

ofwel:

Uit onderzoek van C. van Noort (1993) is gebleken dat `intermezzo`'s niet altijd het gewenste effect hebben.

Het een en ander is instelbaar met het commando:

```
\stelkaderetekstenin[.,.=.,.]
korps          5pt ... 12pt klein groot
letter         normaal vet schuin vetschuin type klein... commando
links         commando
rechts        commando
voor          commando
na            commando
binnen        commando
regelcorrectie aan uit
dieptecorrectie aan uit
marge         standaard ja nee
..=..         zie p 226: \stelomlijndin
```

Kaderteksten laten zich goed combineren met `commando's` bedoeld voor het plaatsen van blokken.

```
\plaatsintermezzo
[hier][int:demo 1]
{Een voorbeeld van een intermezzo.}
\startkadertekst
  For millions of years mankind lived just like animals. Then
  something happened, which unleashed the power of our imagination.
  We learned to talk.
  \blanko
  \regelrechts{--- The Division Bell / Pink Floyd}
\stopkadertekst
```

In dat geval moet wel de plaats van de kadertekst (tussen []) worden weggelaten.

For millions of years mankind lived just like animals.
 Then something happened, which unleashed the
 power of our imagination. We learned to talk.
 — The Division Bell / Pink Floyd

Intermezzo 11.1 Een voorbeeld van een intermezzo.

Ook is het mogelijk een deel van het kader te tekenen. Zo levert de instelling:

```
\stelkaderetekstenin[kader=uit,bovenkader=aan,linkerkader=aan]
```

het intermezzo 11.2 op.

Het ware te wensen dat krijgslustige staatslieden zich
 net zo gemakkelijk lieten ontroeren als ondergetekende
 door het liedje *Wozu sind Kriege da?* van Udo Lindenberg.

Intermezzo 11.2 Een voorbeeld van een intermezzo.

Ook kan een achtergrond worden ingesteld. In dat geval kan wellicht beter het kader achterwege blijven.

Op deze manier valt een intermezzo wat meer op,
 wat nog niet wil zeggen dat het nu wel gelezen wordt.
 Of las u het toch. Hoe dan ook, de instelling was:
`\stelkaderstekstenin[kader=uit,achtergrond=raster]`

Intermezzo 11.3 Een voorbeeld van een intermezzo met achtergrond.

Tot slot nog eentje in kleur. Met behulp van het overzicht van de instellingen en gewapend met wat kennis uit andere paragrafen moet intermezzo 11.4 kunnen realiseren.

```
\stelkaderstekstenin
  [achtergrond=raster,
   rechterkader=aan,
   kader=uit,
   kaderkleur=donkergroen,
   lijndikte=3pt]
\plaatsintermezzo
  [hier][int:kleur]
  {Een voorbeeld van een intermezzo met een geintje.}
\startkaderstekst
Het truukje dat hier is toegepast is eigenlijk vrij eenvoudig.
Toch zou het jammer zijn als Jan en Alleman dit ging na\"apen.
\stopkaderstekst
```

Het truukje dat hier is toegepast is eigenlijk vrij eenvoudig.
 Toch zou het jammer zijn als Jan en Alleman dit ging naäpen.

Intermezzo 11.4 Een voorbeeld
 van een intermezzo met een geintje.

Dus, om een deel van het kader te krijgen, moeten we eerst het hele kader uitzetten met `frame=off`. Trouwens, we kunnen dit soort coderingen ook wat leesbaarder realiseren:

```
\startbuffer
\startkaderstekst ... \stopkaderstekst
\stopbuffer

\plaatsintermezzo
  [hier][int:kleur]
  {Een voorbeeld van een intermezzo met een geintje.}{\haalbuffer}
```

Het steeds weer instellen van een kadertekst is niet echt handig, vandaar dat we de beschikking hebben over:

```
\definieerkadertekst[...] [.,.,.=.,.,.]
...   naam
..=.. zie p 229: \stelkadertekstenin
```

De definitie:

```
\definieerkadertekst
  [muziekfragment]
  [kader=uit, rechterkader=aan, linkerkader=aan]
\plaatsintermezzo
  [hier][ ]
  {Een voorbeeld van een gedefinieerde kadertekst.}
```

```
\startmuziekfragment
```

Stel je voor dat in een document muziekfragmenten zouden staan. Hoe kun je dan nog rustig zitten lezen zonder je omgeving te storen?

```
\stopmuziekfragment
```

levert op:

```

| Stel je voor dat in een document muziekfragmenten zouden staan.
| Hoe kun je dan nog rustig zitten lezen zonder je omgeving te sto-
| ren?
|
```

Intermezzo 11.5 Een voorbeeld van een gedefinieerde kadertekst.

11.8 Kantlijnen

Er kunnen lijnen in de kantlijn worden getrokken, bijvoorbeeld om wijzigingen aan te geven. De te gebruiken commando's zijn:

```
\startkantlijn[...] ... \stopkantlijn
...   getal
```

```
\kantlijn[.1.]{.2.}
.1.   getal
```

Het eerste commando wordt gebruikt rond alinea's, het tweede binnen een alinea.

Door een niveau aan te geven kan men later een kantlijn onderdrukken. Dit doet men door het 'globale' niveau hoger in te stellen dan het 'lokale' niveau:

```
\stelkantlijn[niveau=2]
```

```
niveau  getal
dikte    maat
```

Hieronder is een voorbeeld gegeven van het gebruik van kantlijnen.

```
\startkantlijn
```

Het geluid van eenden demonstreert eveneens hoe moeilijk het is te bepalen wat we eigenlijk horen. Hoewel dit geluid overal in Europa gelijk klinkt, wordt het toch wisselend omschreven als kwaak||kwaak (nederlands), couin||couin (frans), gick||gack (duits), rap||rap (deens) en mech||mech (spaans). Wie deze woorden hardop uitspreekt zal merken dat \kantlijn [4] {ondanks de verschillende} medeklinkers het gekwaak redelijk goed wordt weergegeven. En zegt een koe eigenlijk boe, mboe of mmmmmm?

```
\stopkantlijn
```

Ofwel:²⁰

Het geluid van eenden demonstreert eveneens hoe moeilijk het is te bepalen wat we eigenlijk horen. Hoewel dit geluid overal in Europa gelijk klinkt, wordt het toch wisselend omschreven als kwaak-kwaak (nederlands), couin-couin (frans), gick-gack (duits), rap-rap (deens) en mech-mech (spaans). Wie deze woorden hardop uitspreekt zal merken dat ondanks de verschillende medeklinkers het gekwaak redelijk goed wordt weergegeven. En zegt een koe eigenlijk boe, mboe of mmmmmm?

Als we bijvoorbeeld \stelkantlijn[niveau=2] hadden ingesteld, dan was alleen het deel midden in de tekst van een kantlijn voorzien. We zien in dit voorbeeld ook dat de dikte van de lijn wordt aangepast aan het niveau. We kunnen dit ongedaan maken met \stelkantlijn[dikte=1].

11.9 Blokjes

Er kunnen zwarte blokjes (■) worden getekend met het commando \blokje:

```
\blokje[...=...]
```

```
..=.. zie p 233: \stelblokjesin
```

De instellingen kunnen achterwege worden gelaten. In dat geval worden de standaardinstellingen gebruikt.

²⁰ G.C. Molewijk, Spellingsverandering van zin naar onzin (1992).

```
\stelblokjesin[...]=...]
```

```
breedte  maat max
hoogte   maat max
diepte   maat max
variant  a b
afstand  maat
n        getal
```

De hoogte, diepte en breedte komen overeen met de binnen T_EX gebruikelijke `height`, `depth` en `width`. Als in plaats van een maat het trefwoord `max` wordt gebruikt, worden de maten van de eveneens binnen T_EX gebruikelijke `\strutbox` gebruikt: Wanneer we alle drie de maten op `max` zetten, krijgen we .

- Blokjes kunnen voor verschillende doeleinden worden gebruikt, bijvoorbeeld om een paragraaf te markeren. Deze paragraaf is gemarkeerd met een blokje met de standaardafmetingen: `\inlinker{\blokjes}`.

Een serie blokjes kan worden gezet met het commando `\blokjes`:

```
\blokjes[...]=...]
```

```
..=.. zie p 233: \stelblokjesin
```

- Er zijn twee varianten. Variant `a` zet `n` blokjes naast elkaar, met ieder de ingestelde breedte. Variant `b` verdeelt de ingestelde breedte over het opgegeven aantal blokjes. Deze alinea wordt gemarkeerd door: `\inlinker{\blokjes}`. De instellingen achter `\blokjes` en `\blokjes` zijn dus facultatief.

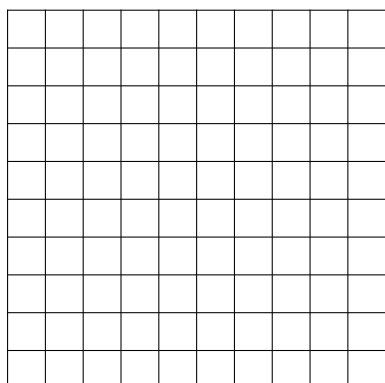
11.10 Roosters

Er kan een rooster (ruitjes) worden gemaakt met het (nog uit te breiden) commando:

```
\rooster[...]=...]
```

```
x        getal
y        getal
nx       getal
ny       getal
dx       getal
dy       getal
xstap    getal
ystap    getal
offset   ja nee
factor   getal
schaal   getal
eenheid  cm pt em mm ex es in
plaats   links midden
```

De standaard-instellingen leveren het volgende rooster op:



Echt zinvol is dit commando niet, maar omdat het achter de schermen wordt gebruikt, is deze functionaliteit toch maar beschikbaar gesteld.

A

Definitives

Definities

<code>\achtergrond</code>	132
<code>\a$\acute{$linea</code>	74
<code>\arg{...}</code> ... <i>tekst</i>	110
<code>\bepaa\lkopnummer[...]</code> ... <i>sectie</i>	151
<code>\bepaallijstkenmerken[...][...=...]</code> ... <i>naam</i> ..=.. zie p 162: <code>\stellijstin</code>	167
<code>\blanko[...]</code> ... <i>n*klein n*middel n*groot geenwit terug wit blokkeer forceer reset regel halverege\l formule vast flexibe\l</i>	61
<code>\blokje[...=...]</code> ..=.. zie p 233: <code>\stelblokjesin</code>	232
<code>\blokjes[...=...]</code> ..=.. zie p 233: <code>\stelblokjesin</code>	233
<code>\but[ref]</code>	▶ ◀ 209
<code>\citaat{...}</code> ... <i>tekst</i>	212
<code>\citeer{...}</code> ... <i>tekst</i>	213
<code>\corrigeerwitruimte{...}</code>	63
<code>\cr\lf</code>	80
<code>\datum[...=...][...]</code> d <i>geta\l</i> m <i>geta\l</i> y <i>geta\l</i> ... <u>dag maand</u> weekdag <u>jaar</u> dd mm jj yy d m j y kenmerk	141

A

`\deel[ref,..]{...}` 146
 ... *tekst*

`\definieeralineas[...][...=...]` 73
 ... *naam*
 n *getal*
 lijn *aan uit*
 hoogte *passend maat*
 voor *commando*
 na *commando*
 binnen *commando*
 afstand *maat*
 tolerantie *zeerstreng streng soepel zeersoepel rek*
 uitlijnen *links rechts midden*

`\definieerbeeldmerk[.1.][.2.][.3.][...=...]` 51
 .1. *naam*
 .2. *boven hoofd voet onder*
 .3. *geen pagina linkerrand linkermarge links midden rechts rechtermarge rechterrind*
 commando *commando tekst*
 status *start stop*

`\definieerblanko[.1.][.2.]` 62
 .1. *naam*
 .2. *zie p 62: \stelblankoin*

`\definieerkadertekst[...][...=...]` 231
 ... *naam*
 ..=.. *zie p 229: \stelkadertekstenin*

`\definieerkleur[...][...=...]` 124
 ... *naam*
 r *tekst*
 g *tekst*
 b *tekst*
 c *tekst*
 m *tekst*
 y *tekst*
 k *tekst*
 s *tekst*

`\definieerkleurgroep[.1.][.2.][x:y:z=, ..]` 127
 .1. *naam*
 .2. *rgb cmyk grijs s*

`\definieerkop[.1.][.2.]` 149
 .1. *naam*
 .2. *sectie*

`\definieerkorps[...1,...][.2.][...=,...]` **116**

.1. 5pt ... 12pt default
 .2. rm ss tt mm hw cg
 tf *file*
 bf *file*
 sl *file*
 it *file*
 bs *file*
 bi *file*
 sc *file*
 ex *file*
 mi *file*
 sy *file*
 ma *file*
 mb *file*
 mc *file*

`\definieerlijst[.1.][.2.][...=,...]` **161**

.1. *naam*
 .2. *naam*
 ..=.. zie p 162: \stellijstin

`\definieermarkering[.1.][.2.]` **176**

.1. *naam*
 .2. *naam*

`\definieeropmaak[...][...=,...]` **95**

... *naam*
 ..=.. zie p 95: \stelopmaakin

`\definieerpalet[...][...=,...]` **128**

... *naam*
naam naam

`\definieerpapierformaat[...][...=,...]` **24**

... *naam*
 breedte *maat*
 hoogte *maat*
 offset *maat*
 schaal *getal*

`\definieerreferentie[...][ref,..]` **184**

... *naam*

`\definieerreferentieformaat[...][...=,...]` **185**

... *naam*
 links *tekst*
 rechts *tekst*
 tekst *tekst*
 label *naam*

`\definieerregister[.1.][.2.]` **186**

.1.
 .2. *naam meervoud*

<code>\definieersamengesteldelijst[.1.][...2...][...=...]</code> .1. <i>naam</i> .2. <i>lijst</i> ..=.. zie p 164: \stelsamengesteldelijstin	163
<code>\definieersectie[...]</code> ... <i>naam</i>	156
<code>\definieersectieblok[...][...=...]</code> ... <i>naam</i> ..=.. zie p 156: \stelsectieblokin	155
<code>\definieersorteren[.1.][.2.][.3.]</code> .1. .2. <i>naam meervoud</i> .3. <i>commando</i>	175
<code>\definieersynoniemen[.1.][.2.][.3.][.4.]</code> .1. .2. <i>naam meervoud</i> .3. <i>commando</i> .4. <i>commando</i>	172
<code>\definieertekst[.1.][.2.][.3.][.4.][.5.]</code> .1. <i>naam</i> .2. <i>hoofd voet</i> .3. <i>tekst</i> .4. <i>tekst</i> .5. <i>tekst</i>	89
<code>\definieertypen[...][...=...]</code> ... file typen <i>naam</i> ..=.. zie p 107: \steltypenin	109
<code>\doordefinieren[...][...=...]</code> ... <i>naam</i> ..=.. zie p 195: \steldoordefinierenin	193
<code>\doordefinitie{.1.}.2.\par</code> .1. <i>tekst</i> .2. <i>tekst</i>	194
<code>\doorlabel[ref]</code>	202

<code>\doorlabelen[...][...=...]</code>	201
<pre> ... naam tekst tekst plaats inmarge <u>intekst</u> wijze pertekst <u>persectie</u> <u>perhoofdstuk</u> blokwijze <u>ja</u> <u>nee</u> kopletter normaal vet schuin vetschuin type kap klein... <i>commando</i> kopkleur naam voor <i>commando</i> na <i>commando</i> </pre>	
<code>\doornummeren[...,.1,...][.2.][...=...]</code>	196
<pre> .1. naam .2. naam ... zie p 197: \steldoornummerenin </pre>	
<code>\doornummering...\par</code>	196
<pre> ... tekst </pre>	
<code>\doorspringen[...][...=...]</code>	200
<pre> ... naam ... zie p 200: \steldoorspringenin </pre>	
<code>\doorsprong...\par</code>	200
<pre> ... tekst </pre>	
<code>\doorstreek{...}</code>	221
<pre> ... tekst </pre>	
<code>\doorstrepen{... ..}</code>	222
<pre> ... tekst </pre>	
<code>\dunnelijn</code>	216
<code>\dunnelijnen[...=...]</code>	216
<pre> ... zie p 217: \steldunnelijnenin </pre>	
<code>\eenregel[ref]</code>	185
<code>\ergens{.1.}{.2.}[ref]</code>	181
<pre> .1. tekst </pre>	
<code>\gebruikreferenties[...,...]</code>	183
<pre> ... file </pre>	
<code>\geenbovenenonderregels</code>	88

<code>\geenhoofdenvoetregels</code>	87
<code>\geenlijst{...}</code> ... <i>tekst</i>	166
<code>\geenmarkering{...}</code> ... <i>tekst</i>	148
<code>\geenspatie</code>	64
<code>\gekoppelderegister[.1.]{.2.}</code> .1. <i>tekst</i> .2. <i>tekst</i>	191
<code>\grijskleur[...]</code> ... <i>tekst</i>	126
<code>\haalmarkering[.1.][.2.]</code> .1. <i>naam</i> .2. eerste laatste vorige beide alles huidige	177
<code>\haarlijn</code>	215
<code>\h1[...]</code> ... <i>getal</i>	217
<code>\hoofdstuk[ref,..]{...}</code> ... <i>tekst</i>	146
<code>\hoofdtaal[...]</code> ... <u>n</u> l fr en uk de es cz ..	143
<code>\hoog{...}</code> ... <i>tekst</i>	68
<code>\huidigedatum[.,.,.,.,.]</code> ... zie p 141: \datum	141
<code>\huidigekopnummer</code>	150
<code>\in{.1.}{.2.}[ref]</code> .1. <i>tekst</i>	▶ ◀ 180

<code>\inanderemarge[.1.][ref]{.2.}</code>	65
.1. +- laag .2. <i>tekst</i>	
<code>\inlinker[.1.][ref]{.2.}</code>	65
.1. +- laag .2. <i>tekst</i>	
<code>\inmarge[.1.][ref]{.2.}</code>	65
.1. +- laag .2. <i>tekst</i>	
<code>\inrechter[.1.][ref]{.2.}</code>	65
.1. +- laag .2. <i>tekst</i>	
<code>\inregel[ref]</code>	185
<code>\installeertaal[...][...=...]</code>	140
... <i>naam</i> spatiering <i>opelkaar ruim</i> lefthyphenmin <i>maat</i> righthyphenmin <i>maat</i> status <i>start stop</i> linkerzin <i>commando</i> rechterzin <i>commando</i> linkersubzin <i>commando</i> rechtersubzin <i>commando</i> linkerciteer <i>commando</i> rechterciteer <i>commando</i> linkercitaat <i>commando</i> rechtercitaat <i>commando</i> default <i>naam</i>	
<code>\invullijnen[...=...]{.1.}{.2.}</code>	217
... zie p 217: <code>\stelinvullijnenin</code>	
<code>\invulregel[...=...]\par</code>	218
... zie p 218: <code>\stelinvulregelsin</code>	
<code>\items[...=...]{...}</code>	212
... zie p 211: <code>\stelitemsin</code>	
<code>\its[ref,..]</code>	209
<code>\kantlijn[.1.]{.2.}</code>	231
.1. <i>getal</i>	

<code>\KAP{...}</code> ... <i>tekst</i>	104
<code>\Kap{...}</code> ... <i>tekst</i>	104
<code>\kap{...}</code> ... <i>tekst</i>	104
<code>\Kaps{..}</code> ... <i>tekst</i>	104
<code>\kleur[.1.]{.2.}</code> .1. <i>tekst</i>	125
<code>\kolom</code>	70
<code>\kop[ref,..]</code>	208
<code>\kopnummer[...]</code> ... <i>sectie</i>	150
<code>\koppelmarkering[.1.][.2.]</code> .1. <i>naam</i> .2. <i>naam</i>	179
<code>\koppelregister[...]</code> ... <i>naam</i>	191
<code>\koptekst{...}</code> ... <i>tekst</i>	142
<code>\laadsorteren</code>	176
<code>\laadsynoniemen</code>	174
<code>\laag{...}</code> ... <i>tekst</i>	68
<code>\labeltekst{...}</code> ... <i>tekst</i>	142

<code>\laho[.1.]{.2.}{.3.}</code>	68
.1. laag .2. tekst .3. tekst	
<code>\mar[ref,..]{...}</code>	209
<code>\margetekst[.1.][ref]{.2.}</code>	67
.1. +- laag .2. tekst	
<code>\markeer[.1.]{.2.}</code>	177
.1. naam .2. tekst	
<code>\nokap{...}</code>	105
... tekst	
<code>\noot[ref]</code>	89
<code>\nop</code>	209
<code>\omlaag[...]</code>	63
... maat	
<code>\omlijnd[...==...]{...}</code>	222
... zie p 226: \stelomlijndin ... tekst	
<code>\onderstreep{...}</code>	220
... tekst	
<code>\onderstrepen{. . . .}</code>	220
... tekst	
<code>\onderwerp[ref,..]{...}</code>	147
... tekst	
<code>\ontkoppelmarkering[...]</code>	179
... naam	
<code>\op{.1.}{.2.}[ref]</code>	180
.1. tekst	

<code>\opelkaar</code>	63
<code>\oppagina[ref]</code>	181
<code>\over{...}[ref]</code> ... tekst	▶ ◀ 180
<code>\overstreep{...}</code> ... tekst	221
<code>\overstrepen{. . . .}</code> ... tekst	221
<code>\pagina[.,.,.,.,.]</code> ... <u>ja</u> opmaak nee voorkeur grotevoorkeur links rechts blokkeer laatste viertal even oneven blanco leeg reset	82
<code>\paginareferentie[ref]</code>	180
<code>\paragraaf[ref,..]{...}</code> ... tekst	146
<code>\paslayoutaan[.,.,.,.,.][.,.,.=.,.,.]</code> ... <i>getal</i> hoogte <i>maat</i> max regels <i>getal</i>	27
<code>\plaatsbeeldmerken[.,.,.,.,.]</code> ... <i>naam</i>	51
<code>\plaatslijst[.,.,.,.,.][.,.,.=.,.,.]</code> ... <i>naam</i> ..=.. zie p 162: \stellijstin	161
<code>\plaatslijstmetsorteren</code>	175
<code>\plaatslijstmetsynoniemen</code>	174
<code>\plaatslokalevoetnoten[.,.,.=.,.,.]</code> ..=.. zie p 90: \stelvoetnotenin	91
<code>\plaatsopgrid[.1.]{.2.}</code> .1. zie p 38: \verplaatsopgrid	37

<code>\plaatsregister[...=...]</code>	▶ ◀ 187
... zie p 188: \stelregisterin	
<code>\plaatsmengesteldelijst[...][...=...]</code>	164
... naam	
... zie p 164: \stelsamengesteldelijstin	
<code>\plaatsvoetnoten[...=...]</code>	91
... zie p 90: \stelvoetnotenin	
<code>\ran{...}</code>	209
<code>\referentie[ref]{...}</code>	180
... tekst	
<code>\regellinks{...}</code>	78
... tekst	
<code>\regelmidden{...}</code>	78
... tekst	
<code>\regelrechts{...}</code>	78
... tekst	
<code>\register[.1.]{..+.2.+..}</code>	186
.1. tekst	
.2. tekst	
<code>\resetmarkering[...]</code>	178
... naam	
<code>\rooster[...=...]</code>	233
x getal	
y getal	
nx getal	
ny getal	
dx getal	
dy getal	
xstap getal	
ystap getal	
offset ja nee	
factor getal	
schaal getal	
eenheid <u>cm</u> pt em mm ex es in	
plaats <u>links</u> midden	
<code>\schrijfnaarlijst[.1.]{.2.}{.3.}</code>	165
.1. sectie naam	

<code>\schrijftussenlijst[.1.]{.2.}</code> .1. <i>sectie naam</i>	165
<code>\som[ref,..]</code>	208
<code>\sorteer[.1.]{.2.}</code> .1. <i>tekst</i> .2. <i>tekst</i>	175
<code>\spatie</code>	64
<code>\startachtergrond ... \stopachtergrond</code>	132
<code>\startalinea ... \stopalinea</code>	73
<code>\startcitaat[.,.,.,.] ... \stopcitaat</code> ... <i>n*links n*midden n*rechts</i>	212
<code>\startdoordefinitie{...} ... \stopdoordefinitie</code> ... <i>tekst</i>	195
<code>\startdoornummering ... \stopdoornummering</code>	197
<code>\startkantlijn[...] ... \stopkantlijn</code> ... <i>getal</i>	231
<code>\startkleur[...] ... \stopkleur</code> ... <i>naam</i> ... <i>tekst</i>	125
<code>\startkolommen[.,.,.=.,.] ... \stopkolommen</code> ... zie p 69: <code>\stelkolommenin</code>	70
<code>\startlokalevoetnoten ... \stoplokalevoetnoten</code> ... zie p 90: <code>\stelvoetnotenin</code>	91
<code>\startnaamopmaak ... \stopnaam</code>	95
<code>\startomgeving \stopomgeving</code> ... <i>file</i>	17

<code>\startonderdeel \stoponderdeel</code> ... <i>file</i>	17
<code>\startopelkaar[...]</code> ... <code>\stopopelkaar</code> ... <i>blanko</i>	63
<code>\startopsomming[...][...=...]</code> ... <code>\stopopsomming</code> ... a A KA n N m r R KR <i>getal</i> verder <u>standaard</u> <i>n*ruim n*aanelkaar</i> opelkaar afsluiter aansluitend opmarge intro kolommen ..=.. zie p 204: <code>\stelopsommingin</code>	208
<code>\startprodukt \stopprodukt</code> ... <i>file</i>	17
<code>\startproject \stopproject</code> ... <i>file</i>	16
<code>\startregel[ref]</code> ... <code>\stopregel</code>	185
<code>\startregelnummers[...]</code> ... <code>\stopregelnummers</code> ... verder	81
<code>\startregels</code> ... <code>\stopregels</code>	80
<code>\starttekstlijn[.1.]{.2.}</code> ... <code>\stoptekstlijn</code> .1. <i>boven onder</i> .2. <i>tekst</i>	220
<code>\starttypen</code> ... <code>\stoptypen</code>	107
<code>\startuitlijnen</code> ... <code>\stopuitlijnen[...]</code> ..=.. zie p 78: <code>\steluitlijnenin</code>	78
<code>\startvanelkaar</code> ... <code>\stopvanelkaar</code>	63
<code>\stelachtergrondenin[.1.][...2...][...=...]</code> .1. <i>boven hoofd tekst voet onder pagina papier linkerpagina rechterpagina</i> .2. <i>linkerrand linkermarge tekst rechtermarge rechterrind</i> status <u>start stop herhaal</u> ..=.. zie p 226: <code>\stelomlijndin</code>	133

\stelachtergrondin[...]=...]

132

linkeroffset *maat*
 rechteroffset *maat*
 bovenoffset *maat*
 onderoffset *maat*
 voor *commando*
 na *commando*
 status start stop
 ... zie p 226: \stelomlijndin

\stelalineasin[.1.][.2.][...]=...]

73

.1. *naam*
 .2. *getal* elk
 letter normaal vet schuin vetschuin type kap klein... *commando*
 breedte *maat*
 hoogte *maat*
 uitlijnen links rechts midden breedte breedte
 tolerantie zeerstreng streng soepel zeersoepel rek
 afstand *maat*
 voor *commando*
 na *commando*
 binnen *commando*
 commando *commando*
 lijn aan uit

\stelarrangerenin[...]=...]

42

... blokkeer 2*16 2*8 2*4 2*2 2**2 2UP 2DOWN gespiegeld geroteerd dubbelzijdig negatief 90 180 270

\stelblankoin[...]

62

... normaal standaard regel *maat* groot middel klein vast flexibel

\stelblokjesin[...]=...]

233

breedte *maat* max
 hoogte *maat* max
 diepte *maat* max
 variant a b
 afstand *maat*
 n *getal*

\stelbovenin[...][...]=...]

88

... zie p 86: \stelhoofdin
 ... zie p 86: \stelhoofdin

\stelboventekstenin[.1.][.2.][.3.]

87

.1. tekst marge rand
 .2. *tekst* *sectie* datum *markering* paginanummer
 .3. *tekst* *sectie* datum *markering* paginanummer

\stelciterenin[...]=...]

213

voor *commando*
 na *commando*
 letter normaal vet schuin vetschuin type kap klein... *commando*
 kleur *naam*
 plaats tekst marge

\steldoordefinierenin[...][...=...] 195

...	<i>naam</i>
kopletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
kleur	<i>naam</i>
breedte	passend ruim <i>maat</i>
afstand	<i>maat</i>
monster	<i>tekst</i>
tekst	<i>tekst</i>
uitlijnen	links midden rechts
marge	standaard ja nee <i>maat</i>
plaats	links rechts boven aaneelkaar inmarge inlinker inrechter hangend
hang	passend ruim <i>getal</i>
voor	<i>commando</i>
tussen	<i>commando</i>
na	<i>commando</i>
springvolgendein	<u>ja</u> nee

\steldoornummerenin[...][...=...] 197

...	<i>naam</i>
..=.	zie p 193: \doordefinieren
plaats	links rechts <u>boven</u> aaneelkaar inmarge inlinker inrechter hangend
tekst	<i>tekst</i>
niveaus	<i>getal</i>
conversie	<u>cijfers</u> letters Letters romeins Romeins
wijze	<u>pertekst</u> <u>persectie</u>
blokwijze	<u>ja</u> nee
sectienummer	ja nummer nee
scheider	<i>tekst</i>
afsluiter	<i>tekst</i>
koppeling	<i>naam</i>
koppelwijze	globaal <u>lokaal</u>
nummer	nee <i>naam</i>
titeluitlijnen	nee <u>ja</u>
start	<i>getal</i>

\steldoorspringenin[...][...=...] 200

...	<i>naam</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
kopletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
breedte	passend <i>maat</i>
tekst	<i>tekst</i>
monster	<i>tekst</i>
voor	<i>commando</i>
na	<i>commando</i>
afstand	<i>maat</i>
scheider	<i>tekst</i>

\steldunnelijnenin[...] 217

interlinie	<u>klein</u> middel groot
n	<i>getal</i>
voor	<i>commando</i>
tussen	<i>commando</i>
na	<i>commando</i>
kleur	<i>naam</i>
achtergrondkleur	<i>naam</i>
hoogte	<i>maat</i> max
diepte	<i>maat</i> max
variant	<u>a</u> b c d
lijndikte	<i>maat</i>

<code>\stelhoofdin[...][...]=...]</code>	86
...	<u>tekst</u> marge rand
status	<u>normaal</u> stop start leeg hoog geen geenmarkering <i>naam</i>
strut	<u>ja</u> nee
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
linkerletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
rechterletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
linkerbreedte	<i>maat</i>
rechterbreedte	<i>maat</i>
voor	<i>commando</i>
na	<i>commando</i>

<code>\stelhoofdtekstenin[.1.][.2.][.3.]</code>	85
.1.	<u>tekst</u> marge rand
.2.	<u>tekst sectie</u> datum <i>markering</i> paginanummer
.3.	<u>tekst sectie</u> datum <i>markering</i> paginanummer

<code>\stelmargin[...][...]=...]</code>	66
...	links rechts <i>getal</i>
plaats	links rechts <u>beide</u>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
voor	<i>commando</i>
na	<i>commando</i>
uitlijnen	<u>binnen</u> buiten links rechts midden normaal nee ja
regel	<i>getal</i>
afstand	<i>maat</i>
scheider	<i>tekst</i>
..=..	zie p 226: \stelomlijndin

<code>\stelinterliniein[...][...]=...]</code>	56
...	reset <u>klein</u> middel groot aan uit
hoogte	<i>getal</i>
diepte	<i>getal</i>
regel	<i>maat</i>
boven	<i>getal</i>
onder	<i>getal</i>

<code>\stelinvullijnenin[...]=...]</code>	217
breedte	passend ruim <i>maat</i>
afstand	<i>maat</i>
voor	<i>commando</i>
na	<i>commando</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
n	<i>getal</i>
interlinie	<u>klein</u> middel groot
scheider	<i>tekst</i>

<code>\stelinvulregelsin[...]=...]</code>	218
breedte	<i>maat</i>
marge	<i>maat</i>
afstand	<i>maat</i>
voor	<i>commando</i>
na	<i>commando</i>

\stelitemsin[...]=...] 211

plaats links rechts inmarge boven onder
 symbool 1 2 ... n a ... tekst geen
 breedte maat
 n getal onbekend
 voor commando
 tussen commando
 uitlijnen links rechts midden marge
 na commando

\stelkadertekstenin[...]=...] 229

korps 5pt ... 12pt klein groot
 letter normaal vet schuin vetschuin type klein... commando
 links commando
 rechts commando
 voor commando
 na commando
 binnen commando
 regelcorrectie aan uit
 dieptecorrectie aan uit
 marge standaard ja nee
 ... zie p 226: \stelomlijndin

\stelkantlijnin[...]=...] 232

niveau getal
 dikte maat

\stelkapitalenin[...]=...] 105

titel ja nee
 sc ja nee

\stelkleurenin[...]=...] 123

status start stop globaal lokaal
 conversie ja nee altijd
 reductie ja nee
 rgb ja nee
 cmyk ja nee
 mpcmyk ja nee

\stelkleurin[...] 125

... naam

\stelkolommenin[...]=...] 69

n getal
 nboven getal
 lijn aan uit
 hoogte maat
 tolerantie zeerstreng streng soepel zeersoepe l rek
 afstand maat
 balanceren ja nee
 uitlijnen ja nee tekst
 blanco vast halverege l regel flexibel groot middel klein
 optie achtergrond
 richting links rechts
 ... zie p 226: \stelomlijndin

<code>\stelkopin[...][...=...]</code>	150
...	<i>sectie</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
tekstletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
nummerletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
nummer	<u>ja</u> nee
eigennummer	ja <u>nee</u>
pagina	links rechts ja
doorgaan	<u>ja</u> nee
hoofd	geen leeg hoog geenmarkering
tekst	geen leeg hoog geenmarkering
voet	geen leeg hoog geenmarkering
voor	<i>commando</i>
tussen	<i>commando</i>
na	<i>commando</i>
variant	<u>normaal</u> inmargin midden tekst
commando	<code>\commando#1#2</code>
nummercommando	<code>\commando#1</code>
tekstcommando	<code>\commando#1</code>
prefix	+ - <i>tekst</i>
plaatskop	<u>ja</u> nee
verhoognummer	<u>ja</u> nee <i>file</i>
uitlijnen	links rechts <u>normaal</u> ruim
tolerantie	zeerstreng streng <u>soepe1</u> zeersoepe1 rek
springvolgendein	ja <u>nee</u>
file	<i>naam</i>
expansie	ja <i>commando</i> <u>nee</u>

<code>\stelkopnummerin[.1.][.2.]</code>	150
.1.	<i>sectie</i>
.2.	<i>getal +getal -getal</i>

<code>\stelkoppeltekenin[...]</code>	144
teken	<code>-- --- -) (= /</code>

<code>\stelkoppnenin[...=...]</code>	149
sectienummer	<u>ja</u> nummer nee
variant	<u>normaal</u> marge midden tekst alinea
scheider	<i>tekst</i>
commando	<code>\commando#1#2</code>

<code>\stelkoptekstin[...][...=...]</code>	142
...	<u>n1</u> fr en uk de es cz ..
<i>naam</i>	<i>tekst</i>

<code>\stelkorpsin[...=...]</code>	100
...	<i>naam</i> <u>serif</u> regular romaan sans support schreefloos mono type teletype handschrift calligrafie 5pt ... <u>12pt</u>

<code>\stelkorpsomgevingin[...][...=...]</code>	114
...	zie p 100: <code>\stelkorpsin</code>
...	zie p 100: <code>\stelkorpsin</code>

<code>\stellabeltekstin[...][...=...]</code>	142
...	<u>n1</u> fr en uk de es cz ..
<i>naam</i>	<i>tekst</i>

`\stelloayoutin[...]=...]`

26

breedte	<i>maat</i> passend midden
hoogte	<i>maat</i> passend midden
rugwit	<i>maat</i>
kopwit	<i>maat</i>
marge	<i>maat</i>
linkermarge	<i>maat</i>
rechtermarge	<i>maat</i>
hoofd	<i>maat</i>
voet	<i>maat</i>
boven	<i>maat</i>
onder	<i>maat</i>
linkerrand	<i>maat</i>
rechterrand	<i>maat</i>
hoofdafstand	<i>maat</i>
voetafstand	<i>maat</i>
bovenafstand	<i>maat</i>
onderafstand	<i>maat</i>
linkermargeafstand	<i>maat</i>
rechtermargeafstand	<i>maat</i>
linkerrandafstand	<i>maat</i>
rechterrandafstand	<i>maat</i>
rugoffset	<i>maat</i>
kopoffset	<i>maat</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
markering	aan <u>uit</u> kleur
plaats	links midden rechts onder boven <u>enkelzijdig</u> <u>dubbelzijdig</u>
schaal	<i>maat</i>
nx	<i>getal</i>
ny	<i>getal</i>
dx	<i>maat</i>
dy	<i>maat</i>
regels	<i>getal</i>
grid	ja <u>nee</u>
bodemwit	<i>getal</i>
snijwit	<i>getal</i>

\stellijstin[...][...=...]

162

...	<i>naam</i>
status	<u>start</u> stop
variant	a b c ... geen commando
koppeling	aan <u>uit</u>
criterium	<i>sectie</i> <u>lokaal</u> vorige huidige alles
paginaovergangen	<i>lijst</i>
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
nummerletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
tekstletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
paginaletter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
kleur	<i>naam</i>
commando	\commando#1#2#3
nummercommando	\commando#1
tekstcommando	\commando#1
paginacommando	\commando#1
interactie	<u>sectienummer</u> tekst paginanummer alles
voor	<i>commando</i>
na	<i>commando</i>
tussen	<i>commando</i>
links	<i>tekst</i>
rechts	<i>tekst</i>
label	ja <u>nee</u>
prefix	<u>ja</u> nee
paginanummer	<u>ja</u> nee
<i>sectienummer</i>	ja nee
titeluitlijnen	ja <u>nee</u>
marge	<i>maat</i>
breedte	<i>maat</i> passend
hoogte	<i>maat</i> passend ruim
diepte	<i>maat</i> passend ruim
afstand	<i>maat</i>
scheider	<i>tekst</i>
symbool	geen 1 2 3 ...
expansie	ja <u>nee</u> commando
maxbreedte	<i>maat</i>
... ..	zie p 222: \omlijnd

\stelmarkeringin[...][..=..]

177

...	<i>naam</i>
status	<u>start</u> stop
scheider	<i>commando</i>
expansie	ja <u>nee</u>

\stelnummeringin[...=...]

83

variant	<u>enkelzijdig</u> dubbelzijdig
plaats	hoofd voet links rechts <u>midden</u> marge kantlijn inlinker inrechter
conversie	<u>cijfers</u> letters Letters romeins Romeins
letter	normaal vet schuin vetschuin type kap klein... <i>commando</i>
links	<i>tekst</i>
rechts	<i>tekst</i>
wijze	pertekst per <i>sectie</i> <u>perdee</u> l
tekst	<i>tekst</i>
nummerscheider	<i>tekst</i>
tekstscheider	<i>tekst</i>
<i>sectienummer</i>	ja nee
scheider	<i>tekst</i>
strut	<u>ja</u> nee
status	<u>start</u> stop
commando	\commando#1

<code>\stelomlijndin[...]=...]</code>	226
hoogte	passend ruim <i>maat</i>
breedte	passend ruim <i>maat</i>
offset	geen overlay default <i>maat</i>
plaats	laag diepte
optie	<u>geen</u> leeg
strut	<u>ja</u> <i>nee</i>
uitlijnen	<i>nee</i> links rechts midden normaal hoog laag laho
onder	<i>commando</i>
boven	<i>commando</i>
kader	<u>aan</u> uit overlay
bovenkader	aan <u>uit</u>
onderkader	aan <u>uit</u>
linkerkader	aan <u>uit</u>
rechterkader	aan <u>uit</u>
kaderoffset	<i>maat</i>
kaderdiepte	<i>maat</i>
kaderhoek	rond <u>recht</u>
kaderstraal	<i>maat</i>
kaderkleur	<i>naam</i>
achtergrond	raster kleur <u>geen</u> voorgrond <i>naam</i>
achtergrondraster	<i>getal</i>
achtergrondkleur	<i>naam</i>
achtergrondoffset	kader <i>maat</i>
achtergronddiepte	<i>maat</i>
achtergrondhoek	rond <u>recht</u>
achtergrondstraal	<i>maat</i>
diepte	<i>maat</i>
hoek	rond <u>recht</u>
straal	<i>maat</i>
leeg	<u>ja</u> <i>nee</i>
voorgndkleur	<i>naam</i>
...	<i>tekst</i>

<code>\stelonderin[...][...]=...]</code>	88
...	zie p 86: \stelhoofdin
.....	zie p 86: \stelhoofdin

<code>\stelonderstrepenin[...]=...]</code>	221
variant	<u>a</u> b c
lijndikte	<i>maat</i>
onderoffset	<i>maat</i>
bovenoffset	<i>maat</i>
lijnkleur	<i>naam</i>

<code>\stelondertekstenin[.1.][.2.][.3.]</code>	88
.1.	<u>tekst</u> marge rand
.2.	<i>tekst sectie</i> datum <i>markering</i> paginanummer
.3.	<i>tekst sectie</i> datum <i>markering</i> paginanummer

`\stelopmaakin[...][...,.=...]`

95

... *naam*
 breedte *maat*
 hoogte *maat*
 voffset *maat*
 hoffset *maat*
 pagina links ja rechts
 commandos *commando*
 dubbelzijdig ja nee leeg
 hoofdstatus normaal stop start leeg geen geenmarkering
 voetstatus normaal stop start leeg geen geenmarkering
 tekststatus normaal stop start leeg geen geenmarkering
 bovenstatus stop start
 onderstatus stop start
 paginastatus stop start
 kleur *naam*

`\stelopsummingin[.1.][...2,...][...=.,...]`

204

.1. *getal* elk
 .2. standaard *n*ruim n*aanelkaar opelkaar vanelkaar afsluiter aansluitend opmarge inmarge autointro los*
 paragraaf intekst
 nee standaard *maat*
 marge
 breedte *maat*
 afstand *maat*
 factor *getal*
 items *getal*
 start *getal*
 voor *commando*
 tussen *commando*
 na *commando*
 links *tekst*
 rechts *tekst*
 kopvoor *commando*
 kopna *commando*
 kopletter normaal vet schuin vetschuin type kap klein... *commando*
 marletter normaal vet schuin vetschuin type kap klein... *commando*
 symletter normaal vet schuin vetschuin type kap klein... *commando*
 afsluiter *tekst*
 n *getal*
 symbool *getal*
 uitlijnen links rechts normaal
 springvolgendein ja nee

`\stelpaginanummerin[...,.=...]`

83

nummer *getal*
 status start stop handhaaf

`\stelpaletin[...]`

129

... *naam*

`\stelpapierformaatin[...1,...][...2,...]`

23

.1. A3 A4 A5 A6 letter ... CD *naam* liggend gespiegeld geroteerd 90 180 270
 .2. A3 A4 A5 A6 letter ... *naam* liggend gespiegeld geroteerd negatief 90 180 270

`\stelrastersin[...,.=...]`

131

methode punt lijn extern
 resolutie *getal*
 factor *getal*
 raster *getal*

\stelrefererenin[...=...] 183

status start stop
sectienummer ja nee
prefix + - tekst
interactie label tekst alles symbool
breedte maat
links *commando*
rechts *commando*
converteerfile ja nee klein groot
scheider tekst
autofile ja nee pagina
globaal ja nee

\stelregelnummerenin[...=...] 81

conversie cijfers letters Letters romeins Romeins tekst
start *getal*
stap *getal*
breedte maat
plaats intekst inmarge
letter normaal vet schuin vetschuin type kap klein... *commando*
prefix tekst
refereren aan uit

\stelregelsin[...=...] 80

voor *commando*
na *commando*
tussen *commando*
inspringen ja nee even oneven

\stelregisterin[.1.][.2.][...=...] 188

.1.
.2. naam
n *getal*
balanceren ja nee
uitlijnen ja nee
letter normaal vet schuin vetschuin type kap klein... *commando*
paginaletter normaal vet schuin vetschuin type kap klein... *commando*
tekstletter normaal vet schuin vetschuin type kap klein... *commando*
aanduiding ja nee
koppeling ja nee
sectienummer ja nee
criterium *sectie* lokaal alles deel
afstand maat
symbool 1 2 ... n a ... geen
interactie paginanummer tekst
expansie ja *commando* nee
refereren aan uit
commando *\commando#1*
plaats links midden rechts
maxbreedte maat
onbekendeverwijzing leeg geen

\stelsamengesteldelijstin[...][...=...] 164

... naam
niveau 1 2 3 4 sectie huidige
... zie p 162: \stellijstin

`\stelsectieblokin[...][...]=...]` **156**

... *naam*
 nummer ja *nee*
 pagina ja *rechts*
 voor *commando*
 na *commando*

`\stelsectiein[.1.][.2.][...]=...]` **156**

.1. *naam*
 .2. *naam*
 conversie cijfers letters Letters romeins Romeins
 vorigennummer ja *nee*

`\stelsorterenin[...][...]=...]` **175**

... *naam*
 voor *commando*
 na *commando*
 commando \commando#1
 status start *stop*
 criterium alles *gebruikt*
 letter normaal vet schuin vetschuin type kap klein... *commando*
 expansie ja *commando* nee

`\stelspatieringin[...]` **64**

... ruim opelkaar

`\stelsubpaginanummerin[...]=...]` **85**

wijze pertekst *persectie* perdee
 status start stop *geen*

`\stelsynoniemenin[...][...]=...]` **173**

... *naam*
 tekstletter normaal vet schuin vetschuin type kap klein... *commando*
 synoniemletter normaal vet schuin vetschuin type kap klein... *commando*
 plaats links rechts boven aaneelkaar inmargin inlinker inrechter
 breedte *maat*
 status start *stop*
 criterium alles *gebruikt*
 conversie ja *nee*
 expansie ja *commando* nee
 commando \commando#1#2#3

`\steltaal[...][...]=...]` **140**

... n1 fr en uk de es cz ..
 zie p 140: \installeertaal

`\steltekstin[...][...]=...]` **88**

... zie p 86: \stelhoofdin
 zie p 86: \stelhoofdin

`\steltekstlijnenin[...]=...]` 220

plaats links inmarge
 voor *commando*
 na *commando*
 tussen *commando*
 breedte *maat*
 afstand *maat*
 korps 5pt ... 12pt klein groot
 kleur *naam*
 letter normaal vet schuin vetschuin type kap klein... *commando*
 lijnkleur *naam*

`\stelteksttekstenin[.1.][.2.][.3.]` 87

.1. tekst marge rand
 .2. *tekst sectie* datum *markering* paginanummer
 .3. *tekst sectie* datum *markering* paginanummer

`\steltolerantiein[.....]` 79

... horizontaal vertikaal rek spatie zeerstreng streng soepel zeersoepel

`\steltypein[...]=...]` 108

spatie aan uit
 optie schuin normaal geen
 letter normaal vet schuin vetschuin type kap klein... *commando*
 kleur *naam*

`\steltypenin[...][.....=...]` 107

... file typen *naam*
 spatie aan uit
 pagina ja nee
 optie schuin normaal *commandos* kleur geen
 tekst ja nee
icommando *commando*
vcommando *commando*
ccommando *commando*
 voor *commando*
 na *commando*
 marge *maat* standaard ja nee
 evenmarge *maat*
 onevenmarge *maat*
 blanco *maat* klein middel groot standaard halverege1 regel
 escape /
 springvolgendein ja nee
 letter normaal vet schuin vetschuin type kap klein... *commando*
 kleur *naam*
 palet *naam* colorpretty
 regels ja nee afgebroken

`\steluitlijnenin[...]` 78

... breedte links rechts midden binnen buiten breed ruim hoogte onder regel reset hangend niethangend afgebroken nietafgebroken

`\stelvoetin[...][.....=...]` 86

... zie p 86: \stelhoofdin
 zie p 86: \stelhoofdin

`\stelvoetnootdefinitiein[...]=...]` **92**
 ..=.. zie p 193: \doordefinieren

`\stelvoetnotenin[...]=...]` **90**

conversie	<code>cijfers</code> letters Letters romeins Romeins
wijze	<code>per</code> tekst <code>persectie</code>
plaats	<code>pagina</code> tekst kolommen hoog geen
lijn	aan uit
voor	<code>commando</code>
na	<code>commando</code>
breedte	<code>maat</code>
hoogte	<code>maat</code>
korps	5pt ... 12pt <code>klein</code> groot
letter	normaal vet schuin vetschuin type kap klein... <code>commando</code>
afstand	<code>maat</code>
kolomafstand	<code>maat</code>
margeafstand	<code>maat</code>
n	<code>getal</code>
nummercommando	<code>\commando#1</code>
splitsen	soepel streng zeerstreng <code>getal</code>
..=..	zie p 222: \omlijnd

`\stelvoettekstenin[.1.][.2.][.3.]` **85**

- .1. `tekst` marge rand
- .2. `tekst` `sectie` datum `markering` paginanummer
- .3. `tekst` `sectie` datum `markering` paginanummer

`\sub[ref,..]` **208**

`\subonderwerp[ref,..]{...}` **147**
 ... `tekst`

`\subparagraaf[ref,..]{...}` **146**
 ... `tekst`

`\subsubonderwerp[ref,..]{...}` **147**
 ... `tekst`

`\subsubparagraaf[ref,..]{...}` **147**
 ... `tekst`

`\switchnaarkorps[...]=...]` **100**
 ... 5pt ... 12pt `klein` groot `globaal`

`\sym{...}` **208**

`\synoniem[.1.]{.2.}{.3.}` **174**

- .1. `tekst`
- .2. `tekst`
- .3. `tekst`

<code>\taal[...]</code> ... <code>n</code> fr en uk de es cz ..	137
<code>\tekstlijn[.1.]{.2.}</code> .1. boven onder .2. <i>tekst</i>	219
<code>\tekstreferentie[ref]{...}</code> ... <i>tekst</i>	180
<code>\tex{...}</code> ... <i>tekst</i>	110
<code>\titel[ref,...]{...}</code> ... <i>tekst</i>	147
<code>\toongrid</code>	38
<code>\tooninstellingen</code>	27
<code>\toonkader[...]</code> ... tekst marge rand	26
<code>\toonkleur[...]</code> ... <i>naam</i>	125
<code>\toonkleurgroep[.1.][...,2,...]</code> .1. <i>naam</i> .2. horizontaal vertikaal naam waarde nummer	130
<code>\toonkorps[.,...,..]</code> ... zie p 100: <code>\stelkorpsin</code>	102
<code>\toonkorpsomgeving[.,...,..]</code> ... zie p 100: <code>\stelkorpsin</code>	113
<code>\toonlayout</code>	27
<code>\toonpalet[.1.][...,2,...]</code> .1. <i>naam</i> .2. horizontaal vertikaal naam waarde	130

<code>\toonprint[...1,...][...2,...][...=,...]</code>	41
... zie p 23: \stel papierformaat in	
... zie p 23: \stel papierformaat in	
... zie p 26: \stellayout in	
<code>\toonstruts</code>	65
<code>\typ{...}</code>	110
... tekst	
<code>\type{...}</code>	107
... tekst	
<code>\typefile{.1.}{.2.}</code>	107
.1. naam	
.2. file	
<code>\uitgerekt{...}</code>	106
... tekst	
<code>\vastespaties</code>	64
<code>\vergelijkkleurgroep[...]</code>	130
... naam	
<code>\vergelijkpalet[...]</code>	130
... naam	
<code>\verplaatsopgrid[...]</code>	38
... boven <u>beide</u> onder	
<code>\versie[...]</code>	21
... <u>definitief</u> concept voorlopig	
<code>\vertaal[...=,...]</code>	143
naam tekst	
<code>\v1[...]</code>	217
... getal	
<code>\voetnoot[ref]{...}</code>	89
... tekst	

<code>\volledigelijstmetssorteren</code>	175
<code>\volledigelijstmetssynoniemen</code>	174
<code>\volledigeregister[...]=...]</code> ... zie p 188: \stelregisterin	▶ ◀ 187
<code>\volledigesamengesteldelijst[...][...]=...]</code> ... naam ... zie p 164: \stelsamengesteldelijstin	164
<code>\Woord{...}</code> ... tekst	105
<code>\WOORD{...}</code> ... tekst	106
<code>\Woorden{... ..}</code> ... tekst	105
<code>\woordrechts{...}</code> ... tekst	79
<code>\zieregister[.1.]{.2.}{...+3.+...}</code> .1. tekst .2. tekst .3. tekst	187

B

Index

Index

De paginanummers verwijzen naar het begin van het hoofdstuk of de paragraaf, waarin het onderwerp wordt besproken.

a

accenten 11
 achtergronden
 layout 133
 tekst 131
 afbreken 137
 afkortingen 172
 alinea's 11, 55, 73
 arrangeren 42
 ASCII 10

b

baselines 55
 beeldmerken 51
 benadrukken 103
 beschrijvingen 196
 blokjes 232
 boldface 100
 boxen 11
 briefpapier 51
 bijlagen 154

c

citeren 212
 cm 11
 cmr 102
 CMYK 123
 commando's 6
 con 102
 CONTEX^T 5

d

datum 141

definities 193
 delen 146
 dik drukken 100
 directories 20
 doorhalen 220
 doorstrepen 220
 dubbelzijdig 83
 Duits 143

e

em 11, 112
 emphasis 103
 enkelzijdig 83
 eu1 102
 ex 11, 112
 extensies 10

f

figuren
 lettertypes 119
 files 10
 directories 20
 font files 119
 foutmeldingen 12
 franse spatiëring 64

g

getypte tekst 106
 grid 36
 grijs conversie 123
 grijswaarden 126

h

B

haakjes 6
 hoofdletters 104
 hoofdstukken 146, 149
 hoofdtteksten 83, 85, 119, 154
 markeringen 146, 176
 huisstijl 51

i

index 186
 controleren 21
 inhoudsopgave 161
 inleidingen 154
 inslagschemas 42
 inspringen 200
 instellingen 24
 interactie
 registers 186
 interlinie 55
 invullijsten 202, 210
 italic 100
 italic zetten 103
 items 202, 210

k

kaders 24, 222, 228
 kapitalen 104
 karakters 11
 klein-kapitalen 104
 kleurgebruik 123
 kleurgroepen 127
 Knuth 5
 kolommen 36, 69, 73
 koppelteken 144
 koppen 142, 146
 alternatieven 155
 kopwit 24
 korps 99
 definiëren 112

l

labelen 201
 labels 142, 185
 layout 24, 55
 lbr 102
 letter 102
 letterformaten 99
 lettertypen 11
 lettertypen 100
 logo's 51, 175
 lijnen 215, 217
 lijsten 76, 161, 210

m

macro's 6
 marges 24
 lijnen 231
 teksten 65
 markeringen 146, 176
 maten 11
 math 110
 medaeval cijfers 100
 menu's 119
 METAPOST 135
 modes 21

n

nieuwe
 pagina 82
 regel 80
 nummeren
 hoofdstukken 146, 149, 155
 labels 201
 opsommingen 202
 pagina's 83
 regels 80
 teksten 196

o

old style 100
 omgevingen 16

omlijnen 222, 228
 onderdelen 16
 onderstrepen 220
 opmaken 94
 opsommingen 196, 202, 210
 overlays 134
 overzichten
 sorteren 175

p

pagina's 82
 paginanummers 83
 paletten 127
 papierformaat 23
 paragrafen 11, 55, 146
 PDF- ϵ - $\text{T}_{\text{E}}\text{X}$ 5
 PDF $\text{T}_{\text{E}}\text{X}$ 5
 printen 39
 produkten 16
 projecten 16
 prtex 10
 pt 11

r

rasters 131, 133
 referenties 180
 regelafstand 36, 55
 regelnummers 184
 regels 80
 regelval 77
 registers 186
 interactie 186
 RGB 123
 roman 99, 100
 roosters 233
 rugwit 24, 83
 ruitjes 233

s

samengestelde lijsten 161

sans serif 99, 100
 scheef drukken 100
 scheef zetten 103
 schermnummers 83
 schreefloze letter 100
 schuin drukken 100
 schuin zetten 103
 selectief zetten 21
 slanted 100
 small-caps 104
 snijwit 24
 sorteren 175
 spatiëring 55, 64
 spatie na punt 64
 specials 21
 spiegelen 83
 staartwit 24
 starten 15
 stellingen 193
 stoppen 12, 15
 structuur 15, 16, 145, 146
 struts 64
 stijlen 21
 subscript 68
 superscript 68
 symbolen 89
 synoniemen 172

t

tabellen 76
 $\text{T}_{\text{A}}\text{B}_{\text{L}}\text{E}$ 10
 tabulatie 69, 200
 tabuleren 76
 talen 137
 quotes 212
 tekstonderdelen 146
 testen 21
 $\text{T}_{\text{E}}\text{X}$ 5
 versie 12
 $\text{T}_{\text{E}}\text{X}_{\text{E}}\text{X}_{\text{E}}\text{C}$ 10
 mode 21
 $\text{T}_{\text{E}}\text{X}_{\text{U}}\text{T}_{\text{I}}\text{L}$ 10

titels *146, 149*

marges *65*

typeletter *99, 100*

typen *106*

typografie *97*

u

uitleidingen *154*

uitlijnen *36, 64, 77, 92*

kolommen *69*

uitvoerformaat *21*

v

verbatim *106*

verbindingsstreepje *144*

verhogen *68*

verlagen *68*

vertalen *143*

verwijzingen *161, 180*

controleren *21*

vet drukken *100*

vlakverdeling *23*

voetnoten *36, 89*

voetteksten *83, 85, 119*

markeringen *146, 176*

vragenlijsten *202, 210, 217*

w

wiskunde *110*

witruimte *55*

z

zetspiegel *24*

C

Commando's

Commando's

De paginanummers verwijzen naar het begin van het hoofdstuk of de paragraaf, waarin het commando wordt besproken.

aantalsubpaginas 83
achtergrond 131, 132, 235
afkorting 172
alinea 73
alinea 55
alinea 74, 235
arg 110, 235

bbox 92
bepaalkopnummer 151, 235
bepaallijstkenmerken 167, 235
blanko 61, 235
blokje 232, 235
blokjes 232, 233, 235
but 202, 209, 235

cbox 92
citaat 212, 235
citeer 212, 213, 235
corrigeerwitruimte 63, 235
crlf 80, 235

datum 141, 235
de 137
deel 146, 236
defineaccent 112
definebodyfont 110, 112
definebodyfontenvironment 112
definecasemap 112
definecharacter 112
definecommand 112
definefontsynonym 112
definieeralgemenestijl 112
definieeralineas 73, 236
definieerbeeldmerk 51, 236
definieerblanko 62, 236
definieerfontsynoniem 112
definieerkadertekst 231, 236
definieerkaderteksten 228
definieerkleur 123, 124, 236
definieerkleurgroep 127, 236
definieerkop 146, 149, 236
definieerkorps 112, 116, 237
definieerkorpsomgeving 112
definieerletter 112
definieerlijst 161, 237
definieermarkering 176, 237
definieeropmaak 94, 95, 237
definieeroverlay 134
definieerpalet 127, 128, 237
definieerpapierformaat 23, 24, 237
definieerreferentie 184, 237
definieerreferentieformaat 185, 237
definieerregister 186, 237
definieersamengesteldelijst 161, 163, 238
definieersectie 156, 238
definieersectieblok 155, 238
definieersorteren 175, 238
definieersynoniemen 172, 238
definieertekst 85, 89, 238
definieertypen 109, 238
disablemode 21
doifmode 21
doifmodeelse 21
doifnotmode 21
doordefinieren 193, 238
doordefinitie 193, 194, 238
doorlabel 201, 202, 238
doorlabelen 201, 239
doornummeren 196, 239
doornummering 196, 239
doorspringen 200, 239
doorsprong 200, 239

C

doorstreep 220, 221, 239
 doorstrepen 220, 222, 239
 dunnelijn 215, 216, 239
 dunnelijnen 215, 216, 239

 eenregel 184, 185, 239
 em 103
 en 137
 enablembox 110
 enablemode 21
 ergens 180, 181, 239

 fr 137

 gebruikreferenties 183, 239
 geenbovenenonderregels 85, 88, 239
 geenhoofdenvoetregels 85, 87, 240
 geenlijst 146, 161, 166, 240
 geenmarkering 146, 148, 240
 geenspatie 64, 240
 geentest 85
 gekoppelderegister 191, 240
 grijskleur 126, 240
 grijswaarde 126

 haalmarkering 176, 177, 240
 haarlijn 215, 240
 hbox 92
 hl 215, 217, 240
 hoofdstuk 146, 240
 hoofdtaal 142, 143, 240
 hoog 68, 240
 huidigedatum 141, 240
 huidigekopnummer 150, 240
 huidigenaam 201

 in 180, 240
 inanderemarge 65, 241
 inlinker 65, 241
 inlijnd 222
 inmarge 65, 241
 inrechter 65, 241
 inregel 184, 185, 241

 installeertaal 139, 140, 241
 invullijnen 217, 241
 invulregel 217, 218, 241
 items 210, 212, 241
 its 202, 209, 241
 ix 99

 kantlijn 231, 241
 KAP 104
 Kap 104
 kap 104
 Kap 104
 KAP 104, 242
 Kap 242
 kap 242
 Kaps 104, 242
 karakters 104
 kleur 123, 125, 242
 kleurwaarde 126
 kolom 69, 70, 242
 kop 202, 208, 242
 kopnummer 149, 150, 242
 koppeldocument 149
 koppelmarkering 176, 179, 242
 koppelregister 186, 191, 242
 koptekst 142, 242

 laadsorteren 175, 176, 242
 laadsynoniemen 172, 174, 242
 laag 68, 242
 labeltekst 142, 242
 laho 68, 243
 lbox 92
 logo 175

 mar 202, 209, 243
 margetekst 65, 67, 243
 markeer 176, 177, 243
 mf 110

 naam 193, 196, 200
 nl 137
 nokap 104, 105, 243

noot 89, 243
nop 202, 209, 243

omgeving 16
omlaag 63, 243
omlijnd 222, 243
onderdeel 16
onderstreep 220, 243
onderstrepen 220, 243
onderwerp 146, 147, 243
ontkoppelmarkering 176, 179, 243
op 180, 243
opelkaar 63, 244
oppagina 180, 181, 244
over 180, 244
overstreep 221, 244
overstrepen 221, 244

pagina 82, 244
paginanummer 83
paginareferentie 180, 244
par 55
paragraaf 146, 244
paslayoutaan 24, 27, 244
plaatsbeeldmerken 51, 244
plaatslokalevoetnoten 89, 91, 244
plaatslijst 161, 244
plaatslijstmetsorteren 175, 244
plaatslijstmetsynoniemen 172, 174, 244
plaatsopgrid 36, 37, 244
plaatsregister 186, 187, 245
plaatssamengesteldelijst 161, 164, 245
plaatsvoetnoten 89, 91, 245
produkt 16
project 16

ran 202, 209, 245
rbox 92
ref 180
referentie 180, 245
regellinks 77, 78, 245
regel midden 77, 78, 245
regelrechts 77, 78, 245

register 186, 245
resetmarkering 176, 178, 245
resetnaam 196, 201
rooster 233, 245

sbox 92
schrijfnaarlijst 161, 165, 245
schrijfnaarregister 186
schrijftussenlijst 161, 165, 246
resetnaam 196
setnostrut 64
setstrut 64
som 202, 208, 246
sorteer 175, 246
sp 137
spatie 64, 246
startachtergrond 131, 132, 246
startalinea 73, 246
startbijlagen 154
startcitaat 212, 246
startdeelomgeving 16
startdoordefinitie 193, 195, 246
startdoornummering 196, 197, 246
startencoding 112
starthoofdteksten 154
startinleidingen 154
startkadertekst 228
startkantlijn 231, 246
startkleur 123, 125, 246
startkolommen 69, 70, 246
startlokalevoetnoten 89, 91, 246
startmapping 112
startmode 21
startnaamopmaak 95, 246, 94
startnotmode 21
startomgeving 16, 17, 246
startonderdeel 16, 17, 247
startopelkaar 63, 247
startopsomming 202, 208, 247
startprodukt 16, 17, 247
startproject 16, 247
startraster 131
startregel 184, 185, 247

startregelnummers 80, 81, 247
startregels 80, 247
startregister 186
startstandaardopmaak 94
starttabulatie 76
starttekst 15
starttekstlijn 220, 247
starttypen 106, 107, 247
startuitleidingen 154
startuitlijnen 77, 78, 247
startvanelkaar 63, 247
stelachtergrondenin 133, 247
stelachtergrondin 131, 132, 248
stelalineasin 73, 248
stelarrangerenin 42, 248
stelblankoin 62, 248
stelblokjesin 232, 233, 248
stelbovenin 85, 88, 248
stelboventekstenin 85, 87, 248
stelciterenin 212, 213, 248
steldoordefinierenin 193, 195, 249
steldoornummersin 196, 197, 249
steldoorspringenin 200, 249
steldunnelijnenin 215, 217, 249
stelhoofdin 85, 86, 250
stelhoofdtekstenin 85, 250
stelinmargein 65, 66, 250
stelinterliniein 55, 56, 250
stelinvullijnenin 217, 250
stelinvulregelin 217
stelinvulregelsin 218, 250
stelitemsin 210, 211, 251
stelkadertekstenin 228, 229, 251
stelkantlijnin 231, 232, 251
stelkapitalenin 104, 105, 251
stelkleurenin 123, 251
stelkleurin 123, 125, 251
stelkolommenin 69, 251
stelkopin 149, 150, 252
stelkopnummerin 149, 150, 252
stelkoppeltekenin 144, 252
stelkoppenin 149, 252
stelkopstekstin 142, 252
stelkorpsin 99, 100, 252
stelkorpsomgevingin 112, 114, 252
stellabeltekstin 142, 252
stellayoutin 24, 26, 253
stellijstin 161, 162, 254
stelmarkeringin 176, 177, 254
stelnummeringin 83, 254
stelomlijndin 222, 226, 255
stelonderin 85, 88, 255
stelonderstrepenin 221, 255
stelondertekstenin 85, 88, 255
stelopmaakin 94, 95, 256
stelopsommingin 202, 204, 256
stelpaginanummerin 83, 256
stelpaginasubnummerin 83
stelpaletin 127, 129, 256
stelpapierformaatin 23, 256
stelrastersin 131, 256
stelrefererenin 180, 183, 257
stelregelnummersin 80, 81, 257
stelregelsin 80, 257
stelregisterin 188, 257
stelregistersin 186
stelsamengesteldelijstin 161, 164, 257
stelsectieblokin 156, 258
stelsectiein 156, 258
stelsorterenin 175, 258
stelspatieringin 64, 258
stelsubpaginanummerin 85, 258
stelsynoniemenin 172, 173, 258
steltaal in 139, 140, 258
steltekstin 85, 88, 258
steltekstlijnenin 219, 220, 259
stelteksttekstenin 85, 87, 259
steltolerantiein 77, 79, 259
steltypein 106, 108, 259
steltypenin 106, 107, 259
steluitlijnenin 77, 78, 259
steluitvoerin 21
stelvoetin 85, 86, 259
stelvoetnootdefinitiein 92, 260
stelvoetnotenin 89, 90, 260
stelvoettekstenin 85, 9, 85, 260

strut 64
sub 202, 208, 260
subnaam 196
subonderwerp 146, 147, 260
subpaginanummer 83
subparagraaf 146, 260
subsubnaam 196
subsubonderwerp 146, 147, 260
subsubparagraaf 146, 147, 260
subsubsubnaam 196
switchnaarkorps 99, 100, 260
sym 202, 208, 260
synoniem 172, 174, 260

taal 137, 261
tbox 92
tekstlijn 219, 261
tekstreferentie 180, 261
tex 106, 110, 261
titel 146, 147, 261
toongrid 36, 38, 261
tooninstellingen 24, 27, 261
toonkader 24, 26, 261
toonkleur 123, 125, 261
toonkleurgroep 127, 130, 261
toonkorps 102, 261
toonkorpsomgeving 112, 113, 261
toonlayout 24, 27, 261
toonpalet 127, 130, 261
toonprint 39, 41, 262
toonstruts 64, 65, 262
totaalaantalpaginas 83
typ 106, 110, 262
type 106, 107, 262
typefile 106, 107, 262

uitgerekt 106, 262

vastespaties 64, 262

vbox 92
vergelijkkleurgroep 127, 130, 262
vergelijkpalet 127, 130, 262
verhoognaam 201
verplaatsopgrid 36, 38, 262
versie 21, 262
vertaal 143, 262
viii 99
v1 215, 217, 262
voetnoot 89, 262
volgendenaam 196, 201
volgenderegister 186
volgendesectie 155
volgendesubnaam 196
volgendesubsubnaam 196
volledigelijstmetsorteren 175, 263
volledigelijstmetsynoniemen 172, 174, 263
volledigeregister 186, 187, 263
volledigesamengesteldelijst 161, 164, 263
vtop 92

Woord 104, 105
WOORD 106
Woord 263
WOORD 263
Woorden 104
WOORDEN 104
Woorden 105, 263
woordrechts 79, 263

x 99
xi 99
xii 99

zieregister 186, 187, 263

CON_TE_XT is een macropakket geschreven rond de typografische programmeertaal T_EX. Het biedt de gebruiker een veelheid aan hulpmiddelen om documenten van vorm te geven. Hoewel oorspronkelijk ontwikkeld voor het vormgeven en ontwikkelen van opleidingsmateriaal, heeft CON_TE_XT zich ontwikkeld tot een systeem om allereerste, eenvoudige, maar vooral ook complexe documenten zo toegankelijk mogelijk vorm te geven.

CON_TE_XT wordt dan ook gebruikt voor het vormgeven van uiteenlopende documenten. Men kan daarbij denken aan boeken, al dan niet wiskundig of technisch van aard, maar ook aan handleidingen bij programma's, machines of producten, kwaliteitshandleidingen, produktiematerialen, overzichtswerken, database-uitvoer en collectiesamenhangende documenten. Hergebruik van bronnen, data-abstractie en structuur zijn dan ook kenmerkend voor CON_TE_XT.

Door de aard van T_EX is CON_TE_XT uitermate geschikt om medium-neutraal gecodeerde bronnen te verwerken tot niet alleen papieren maar ook interactieve documenten. Er is een verregaande ondersteuning voor PDF en XML de daaraan gerelateerde functionaliteit.

CON_TE_XT is beschikbaar in het 'for free'. Naast deze referentie handleiding zijn er handleidingen voor beginners in verschillende talen en zijn er documenten die specifieke functionaliteit beschrijven. Daarnaast zijn er een aantal voorbeeldstijlen beschikbaar en voorbeelden van kant-en-klare documenten.

voor meer informatie:

www.pragma-ade.com