

fonts: context

Fonts in ConT_EXt

Hans Hagen
PRAGMA ADE
Ridderstraat 27
8061GH Hasselt NL
pragma@wxs.nl
ntg-context@ntg.nl

abstract

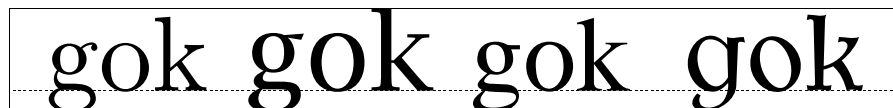
This article is the revised chapter 5 of the ConT_EXt reference manual, typeset in the Maps layout. We will pay attention to all kind of font switching, fine tuning, and also provide some background information. Special attention is paid to the mapping of font names to files and some words are spent on encodings. This text is typeset in Lucida Bright but within the layout specification of the Maps.

keywords

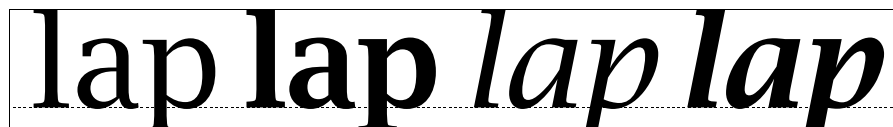
ConT_EXt, fonts, encodings, typography

1 Inleiding

We denken in veel gevallen in woorden, verbaal. Gedurende duizenden jaren heeft men getracht deze gedachten vast te leggen op een medium, bijvoorbeeld steen of papier. Klanken en grammaticale constructies worden vastgelegd in series karakters, die op hun beurt, al dan niet gecombineerd, worden weergegeven in plaatjes: glyphs. Deze afbeeldingen hebben zich niet alleen steeds verder verfijnd, ze zijn ook buitengewoon herkenbaar, zelfs als verschillende vormgevers er hun interpretatie aan geven.



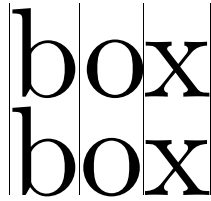
Van links naar rechts zien we hier een Computer Modern, een Lucida Bright, een Times Roman en een Antiqua Torunka font, allen geschaald naar 40pt.



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.



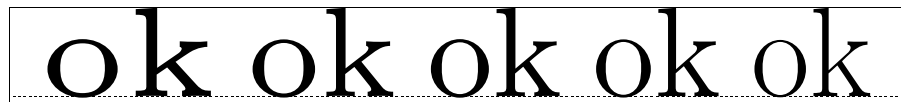
In dit voorbeeld tonen we een Computer Modern, het standaard $\text{T}_\text{E}\text{X}$ lettertype. Dit door Knuth ontworpen lettertype is een variant op een Monotype Times, en kent, in tegenstelling tot de Lucida die in deze handleiding wordt gebruikt, relatief veel kerning paren.

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



We beperken ons hier tot de eerste drie verschijningsvormen. Als we voor een meer algemene aanduiding kiezen, komen we terecht bij de typering in tabel 1.

Serif	Sans	Mono
□□□□□	□□□□□	□□□□□
Regular	Support	Mono
□□□□□□□	□□□□□□□	□□□□□
Roman	Sans	Type
□□□□□	□□□□□	□□□□□

Tabel 1 Een globale indeling in klassen, gedemonstreerd aan een Computer Modern.

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 T_EX. 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.

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 CON_TE_XT. 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 beter tijden werden voor elke afmeting aparte letters ontworpen, een traditie die eigenlijk alleen is voortgezet in de standaard T_EX 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 CON_TE_XT 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 romaan sans support schreefloos mono type teletype
        handschrift calligrafie 5pt ... 12pt
```

terwijl lopende de tekst kan worden omgeschakeld met het commando:

```
\switchnaarkorps[...,...,...]
...      5pt ... 12pt klein groot g1obaa1
```

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 CONTEXT 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

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`, `\ssl` 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 medæval 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 \tfb \tfc \tfd
\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.

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: `\rmsl`, `\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 CON_TE_XT 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 test
```

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

Op het typografische enigzins verwerpelijke onderstrepen en doorhalen gaan we hier niet in. Deze worden elders in de handleiding beschreven.

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, vereist inzicht in onderliggende macros, maar is wel lekker snel.

5 Beschikbare alternatieven

Er zijn slechts weinig font sets (korpsen) 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.¹

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 46: \stelkorpsin
```

Met behulp van het commando `\toonkorps` kan een overzicht worden gegenereerd van de beschikbare lettertypen. Hieronder is het 9pt-korps Computer Modern Roman (`cmr`) weergegeven. De oplettende lezer zal zien dat niet alle varianten (standaard) beschikbaar zijn.

[cmr, 9pt]													
	<code>\tf</code>	<code>\sc</code>	<code>\sl</code>	<code>\it</code>	<code>\bf</code>	<code>\bs</code>	<code>\bi</code>	<code>\tfx</code>	<code>\tfxx</code>	<code>\tfa</code>	<code>\tfb</code>	<code>\tfc</code>	<code>\tfd</code>
<code>\rm</code>	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\ss</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\tt</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

¹ Zie Concrete Mathematics van Knuth cs., in veel opzichten zowel typografisch als didactisch een perfect boek.

Zoals hieronder te zien is, is een 9pt Lucida Bright (lbr) wat groter dan een 9pt 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, 9pt]													
	<code>\tf</code>	<code>\sc</code>	<code>\sl</code>	<code>\it</code>	<code>\bf</code>	<code>\bs</code>	<code>\bi</code>	<code>\tfx</code>	<code>\tfxx</code>	<code>\tfa</code>	<code>\tfb</code>	<code>\tfc</code>	<code>\tfd</code>
<code>\rm</code>	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\ss</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\tt</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

Meestal werkt men met eenzelfde familie en korps. Als men echter verschillende families door elkaar gebruikt, kunnen problemen ontstaan. Deze zijn het gevolg van de verschillende manieren waarop leveranciers de karakters coderen. Er kunnen met name problemen ontstaan rond karakters met accenten en bijzondere karakters, bijvoorbeeld wiskundige karakters.

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.

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*.

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]`

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 kleine 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: `ww`, `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 KAPITALE FOUT HUN KAPITAAL EXPONENTIEEL
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 fraaier 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.

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{...}
...      file
```

Instellingen vinden plaats met:

```

\steltypenin[...][...,...=...,...]
...                file typen naam
spatie             aan uit
pagina             ja nee
optie              schuin normaal commandos kleur geen
voor              commando
na                commando
marge              maat standaard
evenmarge         maat
onevenmarge       maat
blanko            maat klein middel groot standaard halveregeel regel
escape            /
springvolgendein ja nee
palet             naam colorpretty

```

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 spaties zichtbaar gemaakt:

```
Niet-uitlijnen, verdient mijns inziens de voorkeur
boven uitlijnen met behulp van spaties
en zekere boven het oprekken van woorden
```

Een bijzonder geval is het volgende:

```

\definieertypen
  [breedtypen]
\steltypenin
  [breedtypen]
  [evenmarge=-1.5cm,onevenmarge=0cm]

```

Dit kan worden gebruikt als:

```

\startbreedtypen
Soms kan een regel verbatim erg lang zijn, en omdat we niet afbreken,
drukken we hem op de even 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 erg lang zijn, en omdat we niet afbreken, drukken we hem op de even pagina's dus maar in de marge.

We kunnen in-line verbatim instellen met:

```

\steltypenin[...=...]
spatie   aan uit
optie    schuin normaal geen

```

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 53: \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`.

Tot slot bieden 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 T_EX 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 `{}`.

9 Wiskunde

Veel van de T_EX gebruikers hebben voor dit programma gekozen omdat het zo goed is in wiskundig zetwerk. De betrokkenheid van T_EX 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:

```


$$x^2 C \mathbf{x}^2 C x^2 C x^2 C \mathbf{x}^2 C x^2 D 6x^2$$


$$x^2 C \mathbf{x}^2 C x^2 C x^2 C \mathbf{x}^2 C x^2 D 6x^2$$


$$x^2 C \mathbf{x}^2 C x^2 C x^2 C \mathbf{x}^2 C x^2 D \mathbf{6}x^2$$


$$x^2 C \mathbf{x}^2 C x^2 C x^2 C \mathbf{x}^2 C x^2 D 6x^2$$


```

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.

```
x2 C x2 C x2 C x2 C x2 C x2 C x2 D 6x2
x2 C x2 C x2 C x2 C x2 C x2 C x2 D 6x2
x2 C x2 C x2 C x2 C x2 C x2 C x2 D 6x2
x2 C x2 C x2 C x2 C x2 C x2 C x2 D 6x2
x2 C x2 C x2 C x2 C x2 C x2 C x2 D 6x2
x2 C x2 C x2 C x2 C x2 C x2 C x2 D 6x2
```

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.

```
\tf\mf x2 + x2 + x2 + x2 + x2 + x2 = 6x2$
\bf\mf x2 + x2 + x2 + x2 + x2 + x2 = 6x2$
\sl\mf x2 + x2 + x2 + x2 + x2 + x2 = 6x2$
\bs\mf x2 + x2 + x2 + x2 + x2 + x2 = 6x2$
\it\mf x2 + x2 + x2 + x2 + x2 + x2 = 6x2$
\bi\mf x2 + x2 + x2 + x2 + x2 + x2 = 6x2$
```

De exacte positie van `\mf` is dus niet zo belangrijk, we hadden ook kunnen zeggen:

```
\bf x2 + x2 + x2 + x2 + x2 + x2 = \mf 6x2$
```

Een ander aspect van fonts in wiskundige mode betreft gereserveerde namen als `sin` en `cos`.

```
\bf x2 + \hbox{whatever} + \sin(2x)$
```

In tegenstelling tot plain \TeX wordt hier de `sin` ook vet gezet.

```
x2 C whatever C 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]
```

```
\stelkorpsin
```

Het eerdere voorbeeld wordt dan:

```
x2 C whatever C sin(2x)
```

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 T_EX 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 2 toont enkele voorbeelden. We zien dat de breedte van een cijfer .5em is (of omgekeerd: 1em heeft de breedte van twee cijfers).

<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
H	H	H	--	H	H

Tabel 2 De breedte van een em.

Waar de em meestal in de breedte wordt gebruikt, gebruiken we ex in de hoogte. Tabel 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 3 De hoogte van een ex.

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.

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 korps-grootte 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 46: \stelkorpsin
```

Voor `\br` levert dit commando het volgende overzicht op:

[1br]						
text	script	scriptscript	x	xx	klein	groot
14.4pt	11pt	9pt	12pt	10pt	12pt	14.4pt
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
22pt	15.3pt	11pt	17pt	13.2pt	17pt	22pt
17pt	11.8pt	8.5pt	13.6pt	10.2pt	13.6pt	20.3pt

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:

```
\stelkorpsomgevingin[...] [..., ...=..., ...]
... zie p 46: \stelkorpsin
...=... zie p 46: \stelkorpsin
```

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]
```

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 CON_TE_XT 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 `CONTEXT` 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:

```
\definieerkorps [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. Analoog 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.

```
\definieerkorps [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`.

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 $\mathcal{AMS}\text{-}\TeX$ te gebruiken letters en symbolen ook binnen `CONTEXT` te gebruiken: `\definieerkorps[ams]`. Deze zijn ondergebracht in `ma` en `mb`.

```
\definieerkorps[...1...][.2.][...=...]
```

.1.	5pt ... 12pt default
.2.	rm ss tt mm hw cg
tf	<i>file</i>
bf	<i>file</i>
sl	<i>file</i>
it	<i>file</i>
bs	<i>file</i>
bi	<i>file</i>
sc	<i>file</i>
ex	<i>file</i>
mi	<i>file</i>
sy	<i>file</i>
ma	<i>file</i>
mb	<i>file</i>
mc	<i>file</i>

Men is overigens niet gebonden aan a-d. Bij wijze van voorbeeld definiëren we een grote variant van `\tf`:

```
\definieerkorps [9pt,10pt,11pt,12pt] [rm]
  [tfe=ComputerModern at 36pt]
\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 spatiering (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 \TeX) een geval apart. Dit is mede een gevolg van de wijze waarop accenten worden geplaatst. Er zijn grofweg twee methoden: $\cdot f \TeX$ plaatst de accenten zelf $\cdot f$ 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 $\backslash xii$ of $\backslash twelvpoint$, waarmee we voortbouwen op plain \TeX . Dergelijke commando's worden gedefinieerd met:

```

\definieerfontsynoniem [twelvpoint] [12pt]
\definieerfontsynoniem [xii] [12pt]

```

De trefwoorden bij $\backslash stel$ korps in 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 $\backslash letter$ of $\backslash kopletter$ tegen. In die situaties kunnen we een aanduiding meegeven. Met $\backslash definieerletter$ zijn deze aanduidingen vastgelegd. Het derde argument betreft de betekenis in titels van hoofdstukken, paragrafen enz. Alleen $\backslash kap$ 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] []
```

```
\definieerletter [kap,kapitaal] [\kap] [\kap]
```

We hebben in paragraaf 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]
```

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 CON_TE_XT 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!

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:

```
\stelloayoutin[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.

13 Files

In tabel 4 zijn enkele van de standaard meegeleverde font definitie files opgenomen. Deze worden dus aangeroepen op de drie laatste letters.

De `ans` en `i12` encoding files zijn voorgeladen. Men kan een extra encoding laden met `\useencoding`. De twee in tabel 5 laatst genoemde files hebben betrekking op het direct ondersteunen van andere dan de standaard toetsenbord karakters.

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

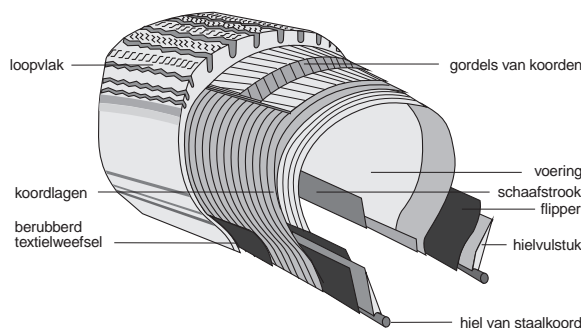
Tabel 4 Enkele standaard meegeleverde font definitie files.

enco-ans	TeXnansi
enco-il2	ISO Latin 2
enco-ibm	default IBM PC code page
enco-win	default MS Windows code page

Tabel 5 Enkele standaard meegeleverde encoding definitie files.

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 \TeX -fonts niet beschikbaar. Als in de tekst zelf een serif wordt gebruikt, kan bijvoorbeeld in een figuur uitstekend een Helvetica worden gebruikt. In figuur 1 is bijvoorbeeld gebruik gemaakt van een Helvetica. In het onderschrift gebruiken we Knuth's Sans Serif.



Figuur 1 Het gebruik van lettertypes in figuren.