

# Scrivere con Heirloom TROFF

Antonio Frecentese

2-6-2023

Un buon programma per impaginare testi, escludendo l'ottimo ma mastodontico L<sup>A</sup>T<sub>E</sub>X e l'ancora piuttosto acerbo SILE, è il vecchio ma ancora utile TROFF. Nella sua variante Heirloom, TROFF è in grado di leggere direttamente i *font* TrueType e OpenType e accetta sorgenti con codifica Unicode, e soprattutto è velocissimo anche su PC molto vecchi.

# Le basi

Innanzitutto indichiamo all'inizio del *file* sorgente `.do xflag 3`. Questo comando fa sì che TROFF non abbia compatibilità con l'altro programma simile e ben noto GROFF ma sfrutti tutte le caratteristiche della versione di Heirloom.

Decidiamo poi le dimensioni della pagina, scrivendo sulla riga successiva (ricordate: sempre con il punto a inizio riga!) ad esempio `.mediasize a5` affinché il documento venga impaginato su fogli in formato A5 e per poter poi aggiungere in seguito alcuni campi appositi per i lettori PDF.

I valori che si possono usare per questo comando sono: **executive**, **letter**, **legal**, **ledger**, **tabloid**, **aN**, **bN**, **cN** oppure due numeri che indicano rispettivamente la larghezza e la l'altezza della pagina.

È bene poi indicare `.lc_type it-IT.UTF-8` perché il programma sappia che deve aspettarsi un sorgente scritto in lingua italiana con la codifica Unicode (UTF-8).

Ora si può cominciare a scrivere il testo vero e proprio. Non preoccupatevi di inserire vocali accentate o caratteri “strani” come ç, € o ß: Heirloom TROFF li userà comunque automaticamente, senza bisogno di usare complicate sequenze di caratteri.

Scrivendo il vostro capolavoro, è possibile andare a capo anche a metà paragrafo: il programma raccoglierà tutte le righe del file sorgente in un paragrafo unico. Per separare un paragrafo dall'altro, è necessario infatti che tra i due vi sia, nel *file* sorgente, una riga vuota a dividerli.

Si può anche indicare a inizio riga `.br` per interrompere la riga corrente e iniziare un nuovo blocco di testo; con `.bp`, invece, si forza l'interruzione della pagina corrente e si passa a una pagina nuova.

Terminate poi il file sorgente con una linea vuota, per evitare che il programma chiuda il file con alcuni errori strani.

Scrivete dunque il vostro primo testo, salvatelo ad esempio nel *file* `prova.tr` e lanciate i comandi

```
$ troff prova.tr | dpost > prova.ps  
$ ps2pdf prova.ps
```

Ora avrete il vostro capolavoro nel *file* `prova.pdf`, pronto da visualizzare e stampare.

Bene, adesso che avete visto proprio il minimo, possiamo cominciare ad arricchire il nostro testo.

## 1. Stili del testo

TROFF usa solitamente il *font* Roman, ovvero il Times New Roman. Questo *font* è salvato come `R` per indicare il carattere “normale”, `I` per indicare quello in *corsivo*, `B` per il **grassetto** e `BI` per il ***grassetto-corsivo***.

Quindi se in una frase si vuole scrivere una parola in corsivo si può cambiare temporaneamente il *font* da `R` a `I` inserendo nel testo `\fI`, scrivere la parola e poi ritornare al testo in `R` inserendo `\fR`: `\fIparole in corsivo\fR` diventa “*parole in corsivo*”.

Questo vale per `R`, `I` e `B`, ma `BI` è composto di due caratteri anziché uno, quindi bisognerà scrivere `\f(BI` invece di `\fBI`.

Se il cambio di stile non va fatto *inline* per una sola parola bensì per un intero blocco di testo, allora è più comodo ricorrere al comando `.ft B`, che imposta in grassetto tutto

il testo successivo fino a un nuovo cambio. Allo stesso modo, `.ft I` lo imposta in corsivo. `.ft R` riporta il testo in stile “normale”.

Per scrivere in corsivo alcune righe, è anche possibile andare a capo nel *file* sorgente, indicare `.ul`, scrivere sotto la riga che dovrà andare in corsivo, andare a capo e proseguire col testo normale. Si può anche far seguire a `.ul` il numero di righe del *file* sorgente che dovranno essere rese in corsivo.

Attenzione: dato che la “riga” è intesa tale finché non c’è un “a capo”, usare `.ul` fa sì che le diverse righe del *file* sorgente non vengano più raccolte l’una dopo l’altra in un unico paragrafo.

Un altro tipo di carattere si può impostare con `.f (CW`, ovvero “*Constant Width*”, cioè a larghezza costante (tipo Courier).

## 2. Allineamento del testo

Il comportamento normale di TROFF è quello di giustificare il testo, cioè fare in modo che esso occupi tutta la larghezza prevista dal documento e venga allineato sia lungo il bordo sinistro che lungo il bordo destro. Per far questo, le righe del documento vengono “riempite” con le varie parole fino a che non si arriva al margine destro, eventualmente aggiungendo qui e lì uno spazio in più, per poi andare automaticamente a capo una volta raggiunto il limite.

È possibile però specificare a inizio riga `.nf` per far sì che da quel punto in poi le righe del file sorgente non vengano più riunite tra loro; questo è utile per esempio in una lettera, per indicare il destinatario o altre informazioni che vanno semplicemente allineate a sinistra, oppure per una poesia. Per

tornare al comportamento di *default* basta usare a inizio riga `.fi`.

Per centrare delle righe, si usa `.ce` specificando subito dopo il numero di righe da centrare. Se non viene indicato alcun numero, solo la riga successiva verrà centrata. È possibile anche indicare un numero molto alto di righe, poi le righe che andranno centrate e poi `.ce 0` per tornare alla normale giustificazione.

Se si scrive `.ad 1`, il testo successivo sarà allineato solo a sinistra, mentre il testo inserito dopo `.ad r` verrà giustificato a destra.

Se dopo aver impostato uno di questi due comandi si centrano alcune righe, poi il testo continuerà con l'allineamento scelto precedentemente.

Per tornare alla normale giustificazione si usa `.ad n` oppure `.ad b`.

### 3. Altri tipi di carattere

È possibile poi scrivere anche scegliere dei *font* diversi dal Times New Roman e scrivere ad esempio con Verdana, Arial, o anche in giapponese. Se nel vostro sistema è installato il *font* Sazanami, per esempio, potreste impostare `.fp 0 SA sazanami-mincho ttf` e indicare quindi:

- che si vuole definire un *font*, col comando `.fp`
- la posizione che deve avere il *font*; lo zero indica “la prima posizione disponibile” (si possono avere fino a 255 tipi di carattere)
- la sigla che verrà poi usata per richiamare il *font*, in questo esempio SA

- il nome del *font*
- il tipo di *font* (Heirloom TROFF supporta anche OTF e TTF).

Dopodiché basta richiamare il *font* voluto ove serve:  
`\f(SA私は\fR` si legge WATASHI WA visualizza “私は  
si legge WATASHI WA”.

Ricordate che dovete impostare le cartelle corrette affinché TROFF trovi i vostri *font*. Io, per esempio, in `.profile`, ho inserito `export TROFFONTS=/usr/local/share/fonts/sazanami`.

In questo modo è anche possibile ridefinire il *font* di base da usare per il documento: la posizione 1 è per il *regular*, la posizione 2 per il corsivo, la numero 3 per il grassetto e la numero 4 per il grassetto-corsivo. Quindi si può ad esempio scrivere:

```
.fp 1 R LiberationSerif-Regular ttf  
.fp 2 I LiberationSerif-Italic ttf  
.fp 3 B LiberationSerif-Bold ttf  
.fp 4 BI LiberationSerif-BoldItalic ttf
```

## 4. Grandezza del testo

Il testo principale di un documento è naturalmente scritto con la stessa grandezza, ma per i titoli o altre necessità particolari è utile variare la grandezza del testo, in modo da aumentare la leggibilità.

Si può modificare la grandezza del testo col comando `.ps` seguito dal valore in punti: `.ps 12` imposta per esempio il *font* a 12 punti (fino al successivo cambio). 1 punto equivale a 1/72 di 1 pollice, e i valori che è possibile dare al comando `.ps` sono: 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 28 e 36. Un valore diverso da questi viene “arrotondato” al primo valido subito superiore, con 36 come massimo. Se non

viene dato un numero dopo `.ps`, si ritorna alla grandezza precedentemente impostata.

Il cambio di grandezza può esser fatto anche *inline* col comando `\s`, sia con valori assoluti che con valori relativi: Normale, `\s-2più piccolo di 2pt\s+2`, normale dà come risultato “Normale, più piccolo di 2pt, normale”.

Ecco come risultano le varie dimensioni: **36 28**  
**24 22 20 18 16 14 12 11 10 9 8 7 6**

## 5. Spaziamento in verticale

Come avrete notato nella sezione precedente, `.ps` varia la grandezza del testo, ma non la distanza tra le righe. Impostando quindi un *font* molto grande ma una piccola distanza tra le righe, i caratteri grandi andrebbero a sovrapporsi a quelli scritti sopra. Bisogna quindi prestare grande attenzione e impostare un’adeguata distanza tra le righe di testo in modo che questo sia leggibile.

L’interlinea tipica tra le righe è pari al 120% del corpo dei caratteri utilizzati (per esempio, di *default* TROFF usa caratteri di corpo 10 con una distanza di 12 punti).

Per specificare lo spaziamento tra le righe si ricorre a `.vs` con un valore seguito da una delle seguenti unità di misura:

- i    inch (pollice)
- c    centimetri
- p    pica (1/6 di 1 pollice)
- m    em (larghezza della lettera *m*)
- n    en (metà di 1 em)
- p    punti (1/72 di pollice)
- u    unità

Se a `.vs` non viene fatto seguire un numero, si torna al valore precedente.

Il comando `.ls` può essere inserito, seguito da un numero, per indicare il numero di righe vuote tra una riga e l'altra del testo, per esempio `.ls 2`. Il valore di *default* è 1.

È possibile impostare uno spaziamento in verticale *una tantum* tra blocchi di testo col comando `.sp`: per esempio, `.sp 3i` lascia uno spazio di 3 pollici dalla posizione attuale al successivo blocco di testo. Quelli successivi avranno il normale distanziamento.

## 6. Lunghezza delle righe

Normalmente, il testo comincia da poco meno di 1 pollice di distanza dal bordo sinistro del foglio fisico. In altre parole, a questa distanza si trova il margine sinistro del foglio. Di solito non vado a modificare questo valore perché lo trovo comodo, ma è possibile modificarlo col comando `.po`. Per esempio, `.po 0` farebbe coincidere il margine sinistro del testo col bordo sinistro fisico del foglio.

Potrebbe essere necessario modificare questo valore per far sì che il margine sia più all'interno rispetto al normale, per esempio quando si rilega con una spirale di plastica o metallo e si vuole che il margine sinistro del foglio sia diverso tra pagina pari e pagina dispari ed evitare che la spirale vada troppo vicino al testo del documento.

Il comando `.ll` viene fatto seguire dalla lunghezza che le successive righe dovranno avere (di *default* l'unità di misura è in em, ma molti preferiscono ragionare in termini di centimetri o pollici).



È chiaro quindi che con l'accoppiata di questi due comandi si possa determinare anche il margine destro del documento.

Il comando `.in` imposta l'indentazione (quindi il rientro) del testo; `.ti` fa rientrare invece solo la prima riga del paragrafo successivo.

È importante sapere che si può usare anche un valore relativo, usando quindi `+` e `-` per indicare la differenza rispetto al precedente valore. Se invece non viene indicato nessun valore, allora si torna al valore precedentemente utilizzato.

Per esempio posso indicare

```
.in 3i
.ll -0.3i
Padre nostro, che sei nei cieli...
.ll +0.3i
.in -3i
```

per ottenere:

```
Padre  nostro,
che   sei  nei
cieli...
```

## Livello intermedio

Combinando queste informazioni di base possiamo ora definire l'aspetto che dovrà avere il nostro testo. Per fare questo, vedremo ora un argomento nuovo: le *macro*.

Si tratta semplicemente di comandi che al loro interno racchiudono altri comandi, in maniera tale per cui basta invocare una tale *macro* una singola volta per eseguirli tutti, eventualmente con dei parametri.

Importante notare che potete creare un vostro *file* di *macro* personali e poi includerlo nel vostro documento con un semplice comando `.so`, per esempio `.so mie_macro.tr`. In questo modo potete crearvi un *file* con le impostazioni generali di un dato documento che poi utilizzerete ancora in altri testi, e limitarvi a indicare di volta in volta solo ciò che cambia. Questo permette anche di creare dei pacchetti di *macro* personalizzati a seconda del tipo di documento che volete creare (per esempio un pacchetto per dei tascabili, uno per delle tesi di laurea, uno per dei saggi...).

TROFF viene già dotato di alcuni pacchetti di *macro*, per consentire di essere già produttivi: `me`, `mm`, `ms` e `mom`, per esempio. Le *macro* presenti in uno di questi pacchetti però non sono necessariamente presenti anche negli altri, per cui ad esempio in uno il paragrafo viene introdotto da `.P` e in un altro da `.PP`. Per poter imparare meglio, però, io ho creato le mie *macro*, per le mie esigenze specifiche.

Non esiste quindi un unico modo di impostare una *macro* per i propri scopi, anzi, spesso io cambio le mie per migliorarle o correggere alcuni errori. Quindi quelle che troverete spiegate qui vanno prese solo come esempi di funzionamento, per capire come funziona TROFF.

## 1. Paragrafi

È sufficiente separare nel *file* sorgente le righe di testo con delle righe vuote per fare in modo che TROFF separi i paragrafi l'uno dall'altro. Io però preferisco fare in modo che ogni paragrafo abbia alcune caratteristiche ben definite: una certa grandezza del carattere, una certa interlinea, una piccola indentazione.

Inoltre, in un documento potrei voler definire più tipi di paragrafo: uno “normale” per il testo in genere, uno per una citazione da riportare all'inizio di un nuovo capitolo, uno per una citazione da riportare invece all'interno di un capitolo, uno per le note a piè di pagina, e via dicendo. Si possono quindi creare diverse *macro*, una per ogni tipo di paragrafo, e poi sarà sufficiente far precedere tale paragrafo dalla sua *macro* per potergli assegnare tutti quei comandi di stile che altrimenti dovremmo ripetere ogni volta.

Potrà sembrarvi strano dover separare con questo comando ogni paragrafo, piuttosto che lasciare semplicemente una riga vuota, ma in effetti non è molto diverso da quanto si fa ad esempio scrivendo codice HTML, in cui si usa `<p></p>` invece di .P.

Normalmente TROFF usa “10 su 12” cioè un *font* di 10 punti con una spaziatura tra le righe di 12 punti. Io per questo libretto ho deciso di usare altre caratteristiche, che ho definito

all'interno della *macro* chiamata `.P` (i nomi di *macro* possono essere solamente di uno o due caratteri, e normalmente tali caratteri sono maiuscoli per evitare di interferire coi normali comandi di TROFF):

```
.de P
.sp 0
.ps 12
.vs 15
.ft R
.ne 1+\n( .Vu
.in 0i
.ti +2m
.ad pb
..
```

- La prima riga indica che sto definendo la *macro* che si chiamerà `.P`
- la seconda che non ci dovrà essere spazio verticale tra le righe di questo paragrafo e di quello precedente
- la terza che il carattere dovrà essere di 12 punti
- la quarta che lo spazio tra una riga e l'altra dovrà essere di 15 punti
- la quinta che il *font* dovrà essere Roman, cioè "normale" (non corsivo o in grassetto)
- la sesta riga richiede che il paragrafo contenga almeno una linea di testo
- la settima stabilisce che l'indentazione deve essere di 0 pollici, cioè nulla
- l'ottava riga impone un'indentazione solo per la prima riga del paragrafo
- la riga successiva impone di giustificare il testo sia a destra che a sinistra, tenendo conto dell'intero paragrafo
- l'ultima riga, formata da due caratteri "punto", indica la fine della *macro*.

## 2. Liste non ordinate

È possibile simulare una lista non ordinata con un paragrafo in cui il testo sia rientrato tranne la prima riga, che deve riportare all'inizio un punto grosso (indicato da `\(bu)`). Basta quindi copiare la definizione del paragrafo che abbiamo appena visto e variare i rientri:

```
.de UL
.sp 0
.ps 12
.vs 15
.ft R
.ne 1+\n(.Vu
.in \w'\(bu\0'u
.ti -\w'\(bu\0'u
\ (bu
.ad pb
..
```

Come vedete, qui si è in pratica rovesciato il ragionamento del paragrafo definito prima: il rientro temporaneo della prima riga è in negativo per farlo andare verso sinistra. Di quanto va fatto rientrare il paragrafo? Di una certa distanza data da `\w' 'u`, che equivale alla *larghezza* (ossia *width*, ecco il significato della *w*) occupata dal carattere `\(bu`, ossia •, seguita da `\0`, ovvero lo spazio di una cifra, giusto per separare un po' il • dal resto del testo.

## 3. Liste ordinate

Il funzionamento è simile a quello delle liste non ordinate, ovviamente, ma bisogna tenere traccia del numero e aumentarlo di conseguenza. Ho scelto quindi di impostare `.OL` per creare la lista ordinata e indicare i vari punti della lista con `.LI`. Il

successivo uso di `.OL` riporterà il contatore a 1:

```
.de OL
.nr 0 0 1 \" inizializza 0, contatore
..
.de LI
.sp 0
.ps 12
.vs 15
.ft R
.ne 1+\\n(.Vu
.in \\w'\\n(bu\\0'u
.ti -\\w'\\n(bu\\0'u
\\n+O)
.ad pb
..
```

## 4. Semplici tabelle

Se vogliamo inserire delle tabelle complesse, è consigliabile usare il programma **tbl**, che viene installato insieme a TROFF (potete trovare dei *tutorial* anche in rete, quindi non mi dilungherò qui). Per semplici dati separati da tabulazioni, però, si può usare il semplice TROFF:

```
.nf
.ta 1.5i +li
.tc
Nome [TAB] Età [TAB] Sesso

Tizio [TAB] 45 [TAB] M
Caio [TAB] 40 [TAB] M
Sempronio [TAB] 40 [TAB] M
Maria [TAB] 9 [TAB] S
.fi
```

dà come risultato:

Nome	Età	Sesso
Tizio	45	M

Caio	40	M
Sempronio	40	M
Maria	9	S

La prima riga impone di rispettare le interruzioni di riga, col comando `.nf`.

La cosa più importante, qui, è la distanza tra le varie tabulazioni, indicata nella riga che inizia con `.ta`: nel caso in esempio, la prima tabulazione porterà a 1.5 pollici, la seconda a 1 pollice di distanza dalla prima (+1).

Noterete che alla prima tabulazione, i valori sono allineati a sinistra (infatti il “9” dell’ultima riga è posto sotto le decine; per allineare questo valore a destra, provate con `.ta 1.5iR +1i` (usare C invece di R per centrare il valore).

Dopo `.tc` si può specificare un carattere da usare come riempimento tra una tabulazione e la successiva: provate con `.tc \ (ru`, per esempio. Possiamo scrivere

```
.nf
.tc _
.ta 10c
Nome e cognome [TAB] ,
.ta 2.5c 5c 9c 10c
CAP [TAB] Città [TAB] Via [TAB] N. [TAB] ,
.fi
```

per ottenere

Nome e cognome \_\_\_\_\_,  
CAP \_\_\_\_\_Città \_\_\_\_\_Via\_\_\_\_\_N. \_\_,

# Livello avanzato

## 1. Header e footer

Tralasciando i documenti molto semplici, normalmente un testo scritto ha alcune informazioni nell'intestazione del foglio ("*header*") e a fondo pagina ("*footer*"): il titolo del libro o del capitolo, il numero di versione del documento, il numero della pagina, e così via.

Si può quindi fare in modo che TROFF, una volta raggiunto il limite inferiore del foglio, stampi il numero di pagina, poi crei un foglio nuovo, stampi in alto l'intestazione e poi proceda col resto del testo.

Per esempio si può impostare il codice seguente:

```
.de fo
.ev 1
.ft R
.ps 9
.if \\n%<1 \\{\\
'sp |\\n(.pu-0.5i-1
.if o .tl ' '%
.if e .tl ' % ' '
\\}
.ev
'bp
..
.de hd
.ev 1
.if \\n%>1 \\{\\
```



```

'sp |0.5i-1
.lt llc
.ft R
.ps 9
.tl '\\"*(TI'
\}
.ev
'sp 0.3i
..

.wh 0 hd
.wh -1i fo

```

In pratica ho creato la *macro* `fo` che verrà lanciata non appena il testo avrà raggiunto 1 pollice di distanza partendo dal fondo pagina (questo test viene effettuato da `.wh -1i fo`; il segno `-` indica di partire dal fondo della pagina e non dal bordo superiore).

- viene prima impostato il *font* del *footer* di tipo R e con la grandezza di 9 punti
- viene poi indicata una condizione per cui se la pagina è dispari (la “o” sta per “*odd*”) la riga dovrà riportare il numero della pagina (il carattere “%”) sul lato destro del foglio, mentre se la pagina è pari (la “e” sta per “*even*”) allora il numero della pagina dovrà essere messo sul lato sinistro; questo verrà fatto solo se il numero della pagina è diverso da 1
- viene riportato il *font* alla grandezza e allo stile precedente
- di seguito viene impostato il *font* dello *header* di tipo R e con la grandezza di 9 punti
- viene indicato il testo da mettere al centro della pagina (ne parleremo tra poco)
- viene riportato il *font* alla grandezza e allo stile precedente
- dopo la *macro* viene dato indicato appunto il test per poterla lanciare

Avrete notato che due comandi (`bp` e `sp`) avevano un apostrofo all'inizio invece del solito punto. Il motivo è che questi due comandi interrompono il flusso del testo non appena viene raggiunta la nuova pagina, e quindi, inserendoli col punto iniziale, avremmo una parte della riga di testo nella pagina attuale e parte in quella successiva, che ovviamente non è quanto vogliamo. L'uso dell'apostrofo consente invece di specificare a TROFF che l'intera riga debba essere riempita prima di procedere con la nuova pagina. I comandi che provocano questa interruzione nel testo sono `.bp`, `.br`, `.ce`, `.fi`, `.nf`, `.sp`, `.in` e `.ti`.

Avete poi visto il comando `.tl`, che serve per indicare tre porzioni di testo da mettere a sinistra, in centro e a destra nella pagina. Notate la sua sintassi particolare, con l'uso degli apostrofi per delimitare queste sezioni e l'uso di `%` per indicare il numero della pagina corrente.

A questo proposito, notiamo che si può usare `.bp n` per terminare una pagina e iniziarne subito una nuova col numero `n` (si può anche usare ad esempio `.bp +3` per causare un'interruzione e al tempo stesso aumentare di 3 il numero di pagina), così come `.pn n` imposta a `n` il numero della pagina successiva ma non crea una interruzione di pagina.

Per finire, un'altra cosa molto utile: come parametro di `.tl` è stato dato `''\*(TI''`. Si è detto quindi a TROFF di recuperare il contenuto della stringa chiamata `TI`. La doppia barra inversa indica che dobbiamo recuperare il valore della stringa (e non definirla); `\*` indica appunto che devo recuperare il valore di una stringa, e la parentesi aperta che il nome di tale stringa è composto da due caratteri anziché uno.

E la stringa? È definita con

```
.ds TI Strivere con Heirloom TROFF
```

Si tratta di una cosa molto utile perché permette, ad esempio, di definire una certa stringa che viene ripetuta molte volte nel testo, e quindi possiamo individuare un comando molto corto per richiamarla. Non solo: è sufficiente modificare la definizione per poter aggiornare immediatamente tutte le sue occorrenze all'interno del testo.

Una stringa può essere lunga anche diverse righe: è sufficiente far seguire tali righe dal carattere \:

```
.ds TI Questo \
è un titolo \
molto lungo
```

Dato che esistono diversi modi di ottenere un dato risultato, potete anche impostare *header* e *footer* separatamente e richiamarli con due test, uno quando si raggiunge una certa distanza dal fondo pagina e uno per la parte superiore della pagina successiva.

## 2. Intestazioni di capitolo e sezione

Vengono qui usate due *macro*. La prima stabilisce le regole per il titolo del capitolo (indicando anche che a ogni suo utilizzo deve essere azzerato il contatore delle sezioni), la seconda quelle del titolo della sezione:

```
.de H1
.ps 16
.sp 0.4
.ne v.2+\n( .Vu
.fi
\fB\\n+Z. \\$1\fR
.nr W 0 1
.ps
..
.de H2
```

```
.ps 14
.sp 0.3
.ne v.2+\n(.Vu
.fi
\f(BI\\nZ.\\n+W. \\$1\fR
.ps
..

.nr Z 0 1 \" inizializza Z
.nr W 0 1 \" inizializza W
    La sintassi da usare è
.H1 "Nome del capitolo"
.H2 "Nome della sezione"
```

Qui troviamo diverse cose interessanti da notare.

Innanzitutto, l'uso di `\\$1` per indicare il primo argomento fornito alla *macro* che stiamo definendo. Questo argomento è normalmente di una sola parola, quindi dato che qui dobbiamo forse ricorrere a più di una parola occorre racchiudere l'argomento tra virgolette.

Poi, *en passant*, l'uso di `\` per indicare (da quel punto fino alla fine della riga) un commento, che non viene interpretato da TROFF.

Soprattutto, prestiamo attenzione al comando `.nr` usato per impostare un registro, chiamato con un nome lungo uno o due caratteri. Ne abbiamo impostati due, Z e W, per poter tenere traccia rispettivamente del numero di capitolo e di sezione. Lo scopo è poter fare dei test: ogni volta che viene creato un nuovo capitolo, il numero delle sezioni deve ritornare a zero. `.nr Z 0 1` crea il registro Z impostato all'inizio a 0 e il cui valore deve essere incrementato di 1 ogni volta che viene usato il comando per cominciare un nuovo capitolo.

La doppia barra inversa viene usata al posto della barra singola per fare in modo che `\n` (oppure `\n(` se il nome del registro è di due caratteri) recuperi il valore di un dato registro

quando viene *usato*, non quando viene *definito*.

Altro dettaglio importante: tutte le specifiche su cosa debba essere visualizzato (quindi escludendo cambi di spaziatura o altro), va messo nella stessa riga.

### 3. Spostamenti locali

Potrebbe esser necessario modificare temporaneamente la posizione in cui scrivere determinati caratteri per ottenere effetti particolari, come apici e pedici.

Questi si ottengono spostandosi di mezza linea verso l'alto mediante `\u` e verso il basso mediante `\d`. Ovviamente questi due comandi andranno usati in coppia: `testo\uapice\d`  
`testo\dpedice\utesto` dà come risultato “testo<sup>apice</sup>  
testo<sub>pedice</sub>testo”.

Se si volesse usare questo sistema per scrivere una formula matematica, bisognerebbe anche predisporre una diminuzione della grandezza del carattere, per esempio `H\s-4\d2\u\s0O` per ottenere  $H_2O$ .

A volte, però, è necessario usare degli spostamenti più complessi, come quelli che ho impostato per poter scrivere L<sup>A</sup>T<sub>E</sub>X:

$$L\backslash v'-0.2m'\backslash h'-0.3m'\backslash s-2A\backslash h'-0.1m'\backslash v'+0.2m'\backslash s0T\backslash h'-0.1m'\backslash v'0.2m'E\backslash v'-0.2m'X$$

Attenzione: tutti questi parametri vanno messi su un'unica riga. Vediamo un po':

- Il comando `\v` imposta uno spostamento locale in verticale; è seguito da due apostrofi tra cui va specificato di quanto ci si vuole spostare: il segno - indica uno spostamento verso l'alto, il segno + verso il basso. È importante notare che, terminato quanto va scritto a questa nuova altezza, bisogna ritornare al punto di partenza; ecco perché abbiamo `\v'-`

0.2m' che si accompagna al successivo \v'0.2m'

- Il comando \h imposta invece uno spostamento locale in orizzontale; anche qui, tra i due apostrofi va specificato di quanto ci si vuole spostare: il segno - indica uno spostamento verso sinistra, il segno + verso destra

Io ho usato questi comandi anche per creare una *macro* che mi permettesse di inserire i *furigana*, ossia la lettura dei *kanji* nei miei testi in giapponese. Eccola:

```
.de RU
.nr KA \w'\f(SA\\$1'u
.nr FU \w'\s-6\f(SA\\$2'u
\f(SA\\$1\fR\v'-0.9m'\h'-\\n(KAu/2u'\h'-
\\n(FUu/2u'\s-6\f(SA\\$2\fR\s+6\v'0.9m'\h'-
\\n(FUu/2u'\h'\\n(KAu/2u'
..
```

La sintassi da usare è ad esempio .RU 私わたし per ottenere 私

Attenzione: lo spazio tra i due parametri del comando deve essere uno spazio “normale”, non uno ottenuto con l'IME (Input Method Editor) che avete scelto per inserire i caratteri giapponesi, altrimenti il programma non riuscirà a separarli correttamente l'uno dall'altro. Inoltre, questa *macro* dovrà essere usata a inizio riga.

## 4. Campi per file PDF

È possibile impostare i campi che dovranno essere inseriti nei *file* PDF che si andranno a creare. Per esempio, io inserisco \X'PDFMark: Author Antonio Frecentese' per inserire il mio nome nel campo destinato all'autore. Invece \X'PDFMark: Title \\\\*(TI` imposta nel campo destinato al titolo del documento il testo definito con la stringa TI.

Vi sono diverse altre possibilità (come impostare soggetto, parole chiave, *link* a risorse esterne o interne e così via), ma secondo me i più importanti sono quelli che consentono di creare i *bookmarks*, ovvero dei segnalibri che fungono anche da indice di navigazione per i *file* PDF. Ecco allora che, all'interno delle *macro* che definiscono i capitoli e le sezioni, io aggiungo `\X'PDFMark: Bookmark N \\$1'`, in maniera tale da inserire nell'elenco il titolo appunto del capitolo, contenuto nel primo argomento (cioè \$1). Al posto di *N* bisogna indicare 0 se si tratta di un segnalibro di livello 0 (il più alto; il primo segnalibro deve per forza essere di livello 0), 1 se deve essere un sottolivello del segnalibro precedente, e così via.

Potremmo quindi avere una struttura come:

- 0           Capitolo 1
  - 1           Sezione 1.1
  - 1           Sezione 1.2
- 0           Capitolo 2
  - 1           Sezione 2.1
    - 2           Sottosezione 2.1.1
    - 2           Sottosezione 2.1.2
  - 1           Sezione 2.2
    - 2           Sottosezione 2.2.1
    - 2           Sottosezione 2.2.2
- 0           Capitolo 3
- 0           Indice

# Approfondimenti

Questa è solo una guida introduttiva, ci sono ancora molti comandi da usare e come già anticipato vi sono diversi modi di costruire delle *macro* per i vostri scopi.

Tenete inoltre presente che vi sono anche alcuni programmi accessori, come `tbl` e `eqn`, che servono rispettivamente per inserire tabelle complesse ed equazioni all'interno dei vostri documenti, nonché pacchetti di *macro* che potete usare ma anche analizzare per usarli come punto di partenza per le vostre *macro* personali, dato che implementano anche funzioni molto complesse come le note a piè di pagina o la formattazione in più colonne: `ms`, `mm` e `me`.

Per ulteriori informazioni, non posso quindi che rimandarvi ai seguenti testi:

- “A TROFF Tutorial”, di Brian W. Kernighan
- “Nroff/Troff User’s Manual”, di Joseph F. Ossanna, Brian W. Kernighan, Gunnar Ritter (che potete trovare nel sito di Heirloom Troff; attualmente la pagina è <http://heirloom.sourceforge.net/doctools.html>).
- tutti i *tutorial* che riuscite a trovare sui suddetti pacchetti di *macro*

Buon lavoro!