

Scrivere, un'arte . . . grafica?

Un breve saggio su come si dovrebbe scrivere al computer, con propaganda delle possibilità tipografiche di T_EX & co. Dedicato a coloro che *non* devono scrivere testi scientifici.

Elaborazione testi

Supponiamo che dobbiate scrivere qualcosa—qualsiasi cosa—al computer; lancerete allora il vostro *word processor* preferito e batterete il testo che volete scrivere, vedendo su schermo una rappresentazione più o meno precisa¹ di quello che otterrete stampando; se decide di modificare l'aspetto di ciò che scrivete, aggiungerete appropriati codici²; l'approccio è comunque per lo più visuale.

Questo approccio presenta dei problemi. Innanzi tutto sposta la concentrazione su *come* far apparire il testo piuttosto su *cosa* mettere nel testo; ovviamente, si può evitare di preoccuparsi dell'aspetto finché non si è completato il testo. Ma supponiamo di aver deciso di formattare tutti i titoli di paragrafo in un certo modo, e di averlo fatto manualmente; se decidessimo in un secondo tempo di cambiare il

¹ a seconda del *word processor* usato.

² operazione che in genere comporta la selezione del testo a cui si vuole applicare la modifica, e successivamente la selezione della modifica da apportare: o da menù, o attraverso scorciatoie da tastiera, oppure attraverso opportune barre di pulsanti.

formato, dovremmo andare a cercare tutti i titoli di paragrafo e sostituire. Un bel lavoro.

Gli utenti più evoluti, se il *word processor* che usano lo permette, usano allora gli *stili*, la cui caratteristica è di separare la *struttura* del documento dal suo *aspetto*; così, si marcano appropriate parti di testo come titoli di capitolo, titoli di paragrafo, etc, e se si decide di cambiare l'aspetto lo si cambia dallo stile e non andando a cercare tutti i capitoli, paragrafi etc.

I *word processor* possono andar bene a due categorie di persone:

- coloro che non ^{devono}_{vogliono} scrivere testi estesi;
- coloro che non ^{vogliono}_{devono} scrivere testi di qualità.

Nelle prossime pagine cercherò di spiegare il perché della mia precedente affermazione, e proporrò un programma, alternativo ai *word processor*, per la composizione di documenti di qualità.

Esistono regole tipografiche su come dovrebbe apparire un testo, regole mirate soprattutto a rendere più facile la lettura del testo, affaticando il meno possibile l'occhio del lettore. Una di queste regole impone ad esempio che il testo dei capoversi venga allineato in modo da occupare l'intera area di stampa, da sinistra a destra, poiché i margini frastagliati affaticano lo sguardo (come potete vedere leggendo questi capoversi).

Esistono regole tipografiche su come dovrebbe apparire un testo, regole mirate soprattutto a rendere più facile la lettura del testo, affaticando il meno possibile l'occhio del lettore. Una di queste regole impone ad esempio che il testo dei

capoversi venga allineato in modo da occupare l'intera area di stampa, da sinistra a destra, poiché i margini frastagliati affaticano lo sguardo (come potete vedere leggendo questi capoversi).

Esistono regole tipografiche su come dovrebbe apparire un testo, regole mirate soprattutto a rendere più facile la lettura del testo, affaticando il meno possibile l'occhio del lettore. Una di queste regole impone ad esempio che il testo dei capoversi venga allineato in modo da occupare l'intera area di stampa, da sinistra a destra, poiché i margini frastagliati affaticano lo sguardo (come potete vedere leggendo questi capoversi).

Avrò sprecato un sacco di carta, ma spero di aver espresso il concetto. Vi sono ovviamente molte altre regole tipografiche; una di esse impone ad esempio che le righe di testo abbiano tra i 60 e i 70 caratteri ognuna, mai di più (questo è il motivo per cui i giornali sono scritti in colonne, e non tutti di fila).

Altre regole riguardano ad esempio la formattazione dei capoversi. Ogni capoverso dovrebbe rappresentare una (sotto)unità di pensiero, e pertanto non dovrebbe essere troppo lungo (mediamente meno di una decina di righe); inoltre il lettore dovrebbe essere in grado di capire quando finisce un capoverso e ne comincia un altro; ad esempio, se un punto fermo capita alla fine di una riga in mezzo ad un capoverso, il lettore deve sapere che la frase successiva fa ancora parte dello stesso capoverso.

Come si separano i capoversi? Due sono i modi più usati; uno (preferibile) consiste nel rientrare la prima riga di ogni capoverso; l'altro consiste nello spaziare i capoversi con righe vuote; l'unico caso in cui questo secondo metodo è consigliabile rispetto al

primo è quando il testo sia costituito per lo più da capoversi più corti di tre righe, poiché in tal caso il primo metodo dà alla pagina un aspetto molto frastagliato (ricordate quanto detto prima circa l'allineamento dei capoversi).

Proprio perché il rientro iniziale ha come ragion d'essere la separazione visiva dei capoversi, esso può essere soppresso quando è ben definito l'inizio di un capoverso; i casi più comuni in cui viene soppresso il rientro sono ad esempio: subito dopo un titolo, o quando si inserisce uno spazio vuoto per separare blocchi logici del discorso senza però iniziare un nuovo paragrafo.

Altre regole tipografiche hanno invece a che fare con l'aspetto dei caratteri.

Ad esempio, certe coppie di lettere hanno un aspetto orribile quando accostate: ad esempio, fi, ff, ffi, ffl; in casi come i precedenti certe caratteristiche di caratteri adiacenti sono in conflitto tra loro, o fastidiosamente vicine; i buoni sistemi tipografici provvedono quindi a sostituire i suddetti gruppi di lettere con delle *legature*: come in fi, ff, ffi, ffl.

Oppure, è preferibile non usare la sottolineatura, e rivolgersi invece all'*inclinato* o al *corsivo*. E non bisognerebbe eccedere con gli attributi dei caratteri, usando ad esempio **grassetto** e *corsivo* **assieme**, a meno che non sia indispensabile.

Non è mia intenzione scrivere un manuale di stile tipografico, essendo io stesso alquanto ignorante in materia. Vorrei solo far notare che far osservare anche solo le regole su descritte ai *word processor* più comuni è cosa non sempre facile. Pertanto, coloro che vorrebbero documenti di alta qualità farebbero meglio a cercare qualche strumento più adatto.

Tipografia

Quando si decide di ottenere documenti di qualità, si abbandona l'elaborazione testi (*word processing*) per passare alla tipografia.

Un buon sistema tipografico cercherà di ridurre al minimo indispensabile le preoccupazioni dell'utente per le regole tipografiche, occupandosi di tutto ciò che può essere reso automatico (come ad esempio la sostituzione delle legature, o la giustificazione dei capoversi).

Ovviamente, si dovrà permettere all'utente qualsiasi fantasia tipografica che egli voglia permettersi, ma poiché queste fantasie (od orrori) richiedono spesso l'infrazione di regole tipografiche, esse potrebbero non essere facili da eseguire.

Un buon sistema tipografico cercherà inoltre di rendere il più semplice possi-

bile cose come le tabelle,

| Programma | Utenze | Qualità |
|------------------|--------|---------|
| Word | 88% | 2/10 |
| WordPerfect | 4% | 6/10 |
| T _E X | 5% | 9/10 |
| Altro | 3% | 4/10 |

le formule,

$$\sin^2 x + \cos^2 x = 1, \quad \int_0^1 \sin x \, dx = 0,$$

le colonne, le note a piè di pagina³ anche in posti strani, e tutte queste cose delicate.

Viene da chiedersi se esistono sistemi tipografici sufficientemente buoni, di non difficile utilizzo e di facile accesso⁴.

La risposta è, felicemente, sì.

³ come questa, ad esempio

⁴ ovvero gratuiti.

T_EX . . .

T_EX: il gruppo di simboli va letto *tech*, con il *ch* tedesco; infatti l’ultima lettera non è una ics maiuscola, ma la lettera greca χ (*chi*). La parola dovrebbe ricordare, a chi ha fatto studi classici, la parola $\tau\acute{\epsilon}\chi\nu\eta$, con i suoi significati di *arte* ed *abilità*.

L’autore del T_EX⁵ ha definito il proprio programma come un sistema tipografico “*mirato alla creazione di libri stupendi—e soprattutto di libri che contengono molta matematica*”⁶.

Detto da un altro punto di vista, il T_EX è fondamentalmente un buon motore tipografico accompagnato da un potentissimo linguaggio di programmazione—ma non preoccupatevi: non occorre essere programmatori per usare il T_EX felicemente. Anzi, la potenza di programmazione del T_EX permette ad esperti programmatori di costruire estensioni (dette *formati*) del T_EX che a loro volta permettono alla gente comune come noi di usare il T_EX senza preoccupazioni.

Forse a causa della definizione data dall’autore, o forse perché il T_EX è nato e si è diffuso in ambienti prettamente scientifici, la fama del T_EX è fortemente legata alla sua capacità di composizione delle formule matematiche; la gente si avvicina quindi al T_EX per lo più quando si trova a dover scrivere documenti ricchi di formule (probabilmente il 75% degli utenti T_EX ha conosciuto il programma al momento della scrittura della tesi di laurea in qualche facoltà scientifica).

⁵ Donald Ervin Knuth, professore emerito dell’Arte della Programmazione alla Stanford University.

⁶ “*intended for the creation of beautiful books—and especially for books that contain a lot of mathematics*” (T_EXbook, Preface, page v).

Eppure, le capacità tipografiche del \TeX vanno ben oltre la facile composizione di formule e la loro buona qualità tipografica. In effetti, non c'è nulla che impedisca l'utilizzo del \TeX anche da parte di gente che la matematica non sa neppure cos'è, o come funziona.

Anzi, benché la maggior parte dei formati \TeX siano orientati ad una utenza tecnico-scientifica, esistono formati (ad esempio il \CONTeXT , quello che ho usato per comporre queste pagine) in cui la composizione delle formule passa in secondo piano, mentre vengono spinte al massimo le capacità tipografiche generali del \TeX .

... and friends

Quando si parla di \TeX ci si riferisce spesso in realtà a tutta una raccolta di programmi che accompagnano il \TeX ; tra questi, i più importanti sono probabilmente \METAFONT e \METAPOST .

\METAFONT è un programma ideato da Knuth stesso, la cui funzione è quella di creare font che possano essere utilizzati dal \TeX ; questi font vengono creati mediante descrizioni grafiche; è quindi abbastanza ovvio che gli algoritmi del \METAFONT siano poi stati utilizzati in un programma in cui la creazione di grafici fosse l'obiettivo principale; tale programma è \METAPOST , e la parte finale (POST) del nome serve a ricordare che i grafici prodotti dal \METAPOST sono in linguaggio POSTSCRIPT (più precisamente, in formato EPS).

Ovviamente, il fatto che \METAPOST sia considerato “amico” del \TeX suggerisce che i disegni prodotti da \METAPOST possano essere inclusi nei documenti \TeX ; la cosa è ovviamente vera, e questo non implica in alcun modo limitazioni: né il

METAPOST può essere usato solo con T_EX, né, viceversa, il T_EX accetta solo grafici METAPOST; infatti, poiché il METAPOST produce EPS, i suoi grafici possono essere inclusi in qualsiasi programma che accetti EPS, ed inoltre il T_EX può includere disegni in qualsiasi formato⁷.

Tuttavia, ovviamente, il fatto che T_EX e METAPOST siano stati sviluppati da una stessa mente⁸ garantisce tra essi una forte cooperazione; per dirne una, nei grafici METAPOST è possibile inserire testo T_EX (garantendo in tal modo una uniformità di rappresentazione dei simboli); viceversa, alcuni formati (in particolare il CON_TE_XT) permettono una fortissima integrazione del METAPOST nei documenti T_EX, integrazione che va ben oltre l’inserzione di codice METAPOST in documenti T_EX; in [figura 1](#) potete vederne qualche esempio.

Documenti elettronici ... e non

Non tutti i documenti vengono creati per una fruizione cartacea. Alcuni vengono “pubblicati” solo elettronicamente, per diverse ragioni; ad esempio:

- è più facile spostare un file da un computer all’altro, piuttosto che fare centinaia di fotocopie;
- il documento potrà avvantaggiarsi di caratteristiche prettamente elettroniche, come la possibilità di riprodurre filmati o spezzoni sonori;

⁷ in realtà il problema della relazione tra T_EX e grafica è alquanto più complesso, vedi “[T_EX e le immagini](#)” in “[Usare il T_EX](#)”.

⁸ ovvero da Knuth; infatti METAPOST è basato sul METAFONT, creato sempre da Knuth.

mondo \TeX e \MetaPost . Ecco un esempio della potenza offerta dal \Context , un po' particolare.

L'integrazione tra \TeX e \MetaPost offerta dal \Context permette effetti interessanti; si può ad esempio far seguire al testo un percorso non lineare (come la circonferenza di contorno a questo capoverso), oppure si può scrivere in forme un po' particolari (come questo capoverso circolare).

Contorni ...

Possiamo far seguire al testo anche curve meno regolari della circonferenza, come potete vedere ...

... e strane curve.

Fig. 1 Due esempi delle capacità tipografiche del \TeX , al di là delle formule

- il documento raggiunge il massimo della propria utilità attraverso interazioni con l'utente, ad esempio per
 - ▽ navigazione mediante collegamenti (come con le pagine web);
 - ▽ riempimento di moduli;
 - ▽ (perché no?) storie a finale multiplo;
 - ▽ ... qualsiasi cosa richieda l'intervento dell'utente.

Se si decide di pubblicare un documento elettronicamente, si viene incontro alla grossa domanda: in quale formato? La risposta dipende ovviamente da quale si desidera sia l'utenza finale del documento. La scelta dovrebbe comunque essere orientata verso la massima portabilità, pur mantenendo la fedeltà al documento originario.

Una scelta potrebbe essere ad esempio di pubblicare un documento in formato HTML, quello comunemente usato per le pagine web; ma il formato HTML (ed i suoi fratelli maggior XML e SGML) presenta delle limitazioni circa la fedeltà (non tutti i browser rappresentano le pagine allo stesso modo, e non è possibile garantire che l'utente disponga dei font usati nel documento) e le potenzialità (ad esempio, l'unico modo di rappresentare spezzoni fantasia come quelli [in figura 1](#) è attraverso immagini—mentre altri formati permettono di inserirli come testo; potete vederlo se state leggendo il formato elettronico di questo opuscolo: provate a selezionare il testo, e vedrete che è vero testo).

Esistono degli *standard* per la pubblicazione elettronica (e non solo) che vanno ben oltre le limitazioni dell'HTML; presenterò qui quelli attualmente più diffusi.

- **POSTSCRIPT**

Il POSTSCRIPT è un linguaggio di programmazione ideato dalla Adobe, orientato alla descrizione di pagine di stampa ed in generale di oggetti grafici (vedi anche il formato [EPS](#)); esso è spesso usato in stampanti laser, plotter ed altri utensili di stampa.

I documenti POSTSCRIPT sono quindi per natura dedicati alla stampa; è tuttavia possibile rappresentarli su schermo, usando programmi opportuni (vedi anche [“Oltre al T_EX”](#) nell'opuscolo [“Come e dove procurarsi il T_EX”](#)).

Tuttavia, il POSTSCRIPT non è nato per la fruizione elettronica, e pertanto manca di funzionalità come i collegamenti o la riproduzione di filmati. Esso può però essere convertito in PDF, ed è quindi possibile inserire nei documenti POSTSCRIPT delle speciali istruzioni che, dopo la conversione in PDF, daranno al documento una piena funzionalità elettronica.

- EPS

Il formato EPS (*Encapsulated POSTSCRIPT*) è un sottoinsieme del POSTSCRIPT, mirato alla descrizione di immagini che possano essere incluse in altri documenti (mancano quindi tutte le funzioni dedicate alla pagina).

Una grande qualità del POSTSCRIPT e dell'EPS è che permette descrizioni del contenuto in maniera indipendente dalla risoluzione dell'utensile che le deve rappresentare (ovvero dalla grana di stampa, o dalla dimensione su schermo), permettendo quindi di sfruttare al massimo le capacità di strumenti ad alta qualità senza che appaiano strani artefatti nelle rappresentazioni con strumenti a bassa qualità.

- PDF

Ideato sempre dalla Adobe, il formato PDF (*Portable Document Format*) è mirato alla creazione e rappresentazione di documenti elettronici. Il formato PDF ha (a differenza del POSTSCRIPT) supporto intrinseco per tutte le caratteristiche precedentemente elencate per i documenti elettronici.

Cosa c'entra il T_EX in tutto questo? Be', ovviamente con T_EX è possibile creare documenti elettronici; benché il T_EX produca normalmente un formato proprietario

portabile sono entro certi limiti, esistono vari strumenti del $\text{T}_{\text{E}}\text{X}$ (ed alcune sue estensioni) che convertono i documenti prodotti dal $\text{T}_{\text{E}}\text{X}$ in formato POSTSCRIPT o PDF .

Usando in particolare il CONTEXT (e possibilmente la variante PDFTE_{X} del $\text{T}_{\text{E}}\text{X}$) è possibile sfruttare appieno le capacità interattive del PDF senza eccessivi affanni. Le potenzialità offerte dal CONTEXT vanno dai riferimenti incrociati all'interno del documento ai collegamenti tra documenti, dall'inclusione di filmati e spezzoni audio all'uso dei *tooltip* (i tooltip sono brevi testi che compaiono quando il cursore del mouse passa su particolari zone “sensibili” dello schermo; la versione elettronica di questo documento ha tooltip dimostrativi in corrispondenza della parola tooltip).

I documenti CONTEXT interattivi possono poi essere creati in modo da tener conto della fruizione su schermo, impostando opportunamente la dimensione della pagina (i formati per le risoluzioni/dimensioni più comuni sono predefinite). Inoltre, con minimi sforzi (e con un unico sorgente), è possibile creare documenti “sincronizzati”, con lo stesso contenuto, l'uno pronto per la fruizione a stampa, l'altro per la fruizione elettronica. Ad esempio, questo opuscolo e gli abbinati “[Usare il \$\text{T}_{\text{E}}\text{X}\$](#) ” e “[Come e dove procurarsi il \$\text{T}_{\text{E}}\text{X}\$](#) ” sono stati creati e compilati in questo modo.

Le brutte notizie

Ci sarebbe da chiedersi perché il $\text{T}_{\text{E}}\text{X}$, nonostante la sua ottima qualità, non sia così diffuso. Uno dei motivi (forse il motivo principale) è che il $\text{T}_{\text{E}}\text{X}$ non brilla per

semplicità d'uso, soprattutto per coloro che sono abituati a lavorare con ambienti ad approccio visuale (comunemente noti come ambienti WYSIWYG⁹).

Tre sono i principali problemi legati ad un ambiente di lavoro non visuale: rappresentazione ritardata, gestione non intuitiva dei font, gestione non intuitiva delle immagini. Per maggiori dettagli su questi aspetti, potete dare un'occhio all'opuscolo “Usare il T_EX”, in cui descrivo aspetti un po' più tecnici dell'uso del T_EX, ed in cui vengono affrontati e discussi i problemi sopra enunciati. (L'opuscolo esiste, ed i riferimenti presenti in questo opuscolo sono collegati ai corrispondenti punti dell'opuscolo “Usare il T_EX”, se consultate la versione elettronica).

Supponiamo che vi abbia convinto ...

Sarò ottimista, ma penso che la cosa potrebbe interessarvi; date allora un'occhio all'opuscolo “Come e dove procurarsi il T_EX”, in cui si trovano i principali punti di riferimento per chi fosse interessato al mondo del T_EX.

⁹ la sigla sta per *What You See Is What You Get*, ovvero “ciò che vedi è ciò che ottieni”, quindi di per sé indica tutti gli ambienti in cui è possibile avere una anteprima fedele di ciò che si otterrà stampando: ogni programma con anteprima è allora WYSIWYG; in quest'ottica si potrebbe il T_EX è forse più WYSIWYG degli altri, poiché la rappresentazione data dai visualizzatori è sempre molto fedele. La sigla viene però comunemente usata per indicare gli ambienti di lavoro in cui si ha una rappresentazione più o meno fedele di ciò che si scrive *mentre* si scrive, quindi gli ambienti ad approccio visuale.

Colofon

Queste pagine sono state composte dal `CONTEXT`; tutti gli errori di battitura ed eventuali errori di altro genere sono da ascrivere all'autore dei fogli, e non al programma di composizione.

bilotta78@hotpop.com