

EIF-e.Book Editore — *Software*

Paolo Biffis

10 settembre 2017

Indice

Premessa	1
1 Dattilografia, tipografia e scrittura visuale	2
2 Caratteri <i>vs font</i>	2
3 Software sincrono e asincrono	6
4 Programmi sincroni o a scrittura visuale	7
5 $\text{\LaTeX} 2_{\epsilon}$	7
6 Imparare \LaTeX e usare <i>Share\LaTeX</i>	8
7 <i>TrueType, Type 1... embedded vs subset</i>	8
Riferimenti bibliografici	9

Premessa

Comporre professionalmente un libro è questione diversa dal comporre un paper: quest'ultimo in genere consiste in poche decine di cartelle¹, di solito

¹Per misurare la dimensione del testo, si parte dalla dimensione standard della riga composta di 60 battute (spazi inclusi) per giungere alla dimensione di una cartella. La cartella standard si compone di 1500 battute suddivise in 25 righe; la cartella editoriale si compone di 1800 battute suddivise in 30 righe. La pagina, invece, dipende dalla sua formattazione per cui non vi è corrispondenza fra numero di pagine e numero di cartelle; il numero di battute di un rigo, però, dovrebbe essere contenuto fra le 60 e le 65, in modo da non affaticare il lettore.

spillate e spesso stampate fronte-retro su fogli A4. Un libro, invece, consiste in un numero di pagine maggiore, appaiate e rilegate, sulla quali vengono stampati, oltre che un testo, le tabelle e i piè di pagina, anche indici, tabelle, figure, grafici, ecc.

L'Autore è così chiamato a prestare attenzione, oltre che al contenuto del testo, anche alla sua formattazione. La necessità di ridurre l'attenzione sulla formattazione per concentrarla sui contenuti ha spinto a inventare dei sw che vi provvedessero automaticamente, gestendo anche gli oggetti flottanti (formule, tabelle, grafici, note a piè di pagina ecc.) senza che l'Autore perdesse molto tempo nell'inserire questi oggetti nella pagina desiderata e al punto giusto.

Il prezzo che si paga per risparmiare del tempo prezioso consiste nel far proprio un sw in grado di automatizzare i processi di formattazione. In questo appunto si cerca di spiegare perché lo Stampatore (Lulu.com/) spesso non accetta i file .pdf generati dai wp più diffusi.

1 Dattilografia, tipografia e scrittura visuale

Comporre tipograficamente è ben diverso dal dattilografare un testo, di conseguenza è anche ben diverso dal dattilografare un testo con i moderni sostituti della macchina da scrivere, i *word processor* (wp). Il risultato che emerge usando questi ultimi dipende dall'abilità del tastierista e dalle decisioni in materia di margini, di spazi, di modalità di inserimento delle note ecc. che gli vengono delegate.

A parte il testo, la composizione tipografica professionale prevede che il materiale da leggere faciliti il lettore attraverso l'uso di caratteri, di spazi, di disposizione del testo che ne agevoli la lettura, il che può rappresentare anche una questione di buona educazione. La scelta dei *font*, la loro grandezza, il loro stile, la loro nerezza, la scelta della distanza fra le righe, i margini, gli spazi bianchi lasciati attorno agli oggetti non testuali, sono tutte cose di competenza del 'disegno editoriale': \LaTeX sostituisce sia il disegnatore editoriale sia il tipografo, anche se non sostituisce il tastierista. Se lo si lascia fare, \LaTeX darà un risultato sicuramente migliore dei nostri tentativi dilettanteschi di occuparci di una questione molto complessa qual è la tipografia.

2 Caratteri *vs font*

La scrittura serve a trasmettere e a conservare nel tempo i concetti espressi con il linguaggio parlato. Da almeno sei millenni l'uomo ha cercato

di scrivere, adottando segni che rappresentassero univocamente le parole e quindi i concetti: dapprima attraverso immagini, successivamente attraverso la scrittura cuneiforme, poi con un insieme di immagini pittoriche a rappresentare parole, sillabe e lettere (i geroglifici), fino a giungere ad un numero ridotto di simboli rappresentati dall'alfabeto nell'area della Mezzaluna Fertile. In Cina e in Giappone la quantità di simboli ammonta a diverse migliaia (nel 1946, in Giappone i simboli vennero ridotti a 1.850 dei quali 881 venivano insegnati nelle scuole elementari) ([4]: 11) mentre la lingua inglese si esprime avvalendosi di 26 lettere, facilitando l'apprendimento anche alle persone periferiche rispetto ai poteri costituiti. Ogni Paese, tuttavia, ha sempre cercato una via autonoma di scrittura, facendo così nascere una quantità molto elevata di simboli. Anche attualmente si possono notare non solo i diversi simboli delle lingue morte, ma anche una quantità notevole di simboli diversi, utilizzati dalle lingue vive.

Da sempre si pone anche il problema di standardizzare i segni che hanno l'ambizione di rappresentare un determinato linguaggio parlato e di trovare un modo per rendere comprensibili fra loro i diversi linguaggi così da rendere possibile il confronto fra diversi concetti: il mito della Torre di Babele ne è tuttora testimonianza; la Stele di Rosetta (British Museum) è lo strumento che ci ha consentito di decodificare il linguaggio degli antichi egizi espresso in tre diverse grafie (geroglifico, demotico e greco); il Codice di Hammurabi (Louvre) è scritto in caratteri cuneiformi, decodificati dall'assirologo Jean-Vincent Scheil tradotto e pubblicato nel 1904.

Le persone oggi più anziane, come l'estensore di queste note, che scrivevano prima degli anni Ottanta sono passate attraverso l'esperienza di scrivere un testo a mano con un lapis, per poter correggere gli errori o le frasi, e dattilografarlo successivamente in più copie fruendo della carta carbone. Chi scrive queste note fu molto soddisfatto a suo tempo quando riuscì a dattilografare direttamente il testo, senza dover passare per una stesura manuale e, successivamente, di dover fruire del costoso lavoro di una dattilografa.

I caratteri della macchina per scrivere erano 'scolpiti' su 'martelletti' azionati da tasti che consentivano al martelletto di colpire un nastro inchiostrato che a sua volta lasciava sulla carta il simbolo del carattere scolpito (diverso da quello che si vedeva sul tasto) e faceva avanzare il carrello di uno spazio che avrebbe contenuto il carattere successivo. A fine riga, inoltre, si spostava manualmente il carrello verso destra per scrivere la riga successiva, ottenendo nel contempo un avanzamento di riga (interlinea). I caratteri erano pochi e imm modificabili, l'interlinea era fissa.

Le innovazioni tecnologiche che si susseguirono negli anni consentirono, oltre che di velocizzare il lavoro delle macchine meccaniche, diventate elettriche e poi elettroniche, anche di disporre di diversi *set* di caratteri, semplicemente sostituendo una 'testina' con un'altra, o una 'margherita' con un'altra: si

trattava di 'palline' di metallo, o di supporti di plastica, ove erano 'scolpiti' i *set* di caratteri. Ma si trattava sempre di un numero di caratteri limitato dalla dimensione fisica del supporto di metallo o di plastica e dal disegno dei caratteri che vi erano scolpiti. Uno dei vantaggi delle macchine elettriche era anche quello di non far dipendere la pressione dei tasti dal tastierista, uniformando così l'intensità dell'inchiostro sulla carta. Inoltre, l'avanzamento di riga rimaneva fisso ma era stato automatizzato.

L'elettronica e l'informatica, con gli applicativi per i computer all'inizio degli anni Ottanta e con l'adozione di un linguaggio meno criptico (grazie a Mackintosh, a metà degli anni Ottanta), ebbero il notevole pregio di diffondere la video-scrittura presso il grande pubblico. Divenne così possibile scrivere un testo, sempre schiacciando i tasti; in tal modo, tuttavia, per ogni tasto sulla tastiera (lettera, numero, segno di interpunzione, spazio, tabulatore ecc.) si trasmetteva allo schermo apparentemente il carattere digitato, ma in realtà si trasmetteva l'algoritmo che stava sotto il tasto digitato. Si è iniziato così a poter cancellare e riscrivere il testo quante volte si voleva, a salvare copie e diverse stesure diverse dello stesso documento, ad utilizzare *set di caratteri* (i *font*² scelti a piacimento dal tastierista il quale poteva decidere il formato della pagina, i margini, l'interlinea ecc.

Questa distinzione fra *singolo carattere* e *collezione di caratteri* è utile per segnalare che la maggior parte degli utilizzatori dei wp pensa che i caratteri digitati sulla tastiera e che si vedono sullo schermo siano come quelli che si digitavano sulla macchina per scrivere e che si vedevano sulla carta; cioè pensano che i caratteri che si digitano sulla tastiera si trasmettano, *sic et simpliciter*, allo schermo e, da qui, alla stampante di cui hanno esperienza.

Invece se digito la lettera A (che sulla tastiera compare in maiuscolo) si vede una lettera minuscola, a meno che non si attivi un tasto apposito, come accadeva sulla macchina per scrivere, ma bisogna tenere conto che se digito la lettera A, sullo schermo compare una a (minuscola o maiuscola) che appartiene ad un carattere che fa parte della collezione di caratteri che ho prescelto (Times New Roman, Verdana, Arial, Computer Modern, ecc.), un po' come accadeva con le palline di acciaio ove erano scolpiti i caratteri delle macchine per scrivere elettriche.

Le collezioni di caratteri, però, sono oggi potenzialmente illimitate perché l'elettronica consente la possibilità di disegnarne una enorme quantità³; per

²La parola inglese *font* - di derivazione francese, abbreviazione di *fountain*, dapprima *fount* e, infine, entrata nel linguaggio corrente della tipografia digitale come *font* - sta ad indicare due elementi: un *singolo carattere* e una *collezione di caratteri* (un *set*). A volte si usa la/le *font* ma più spesso il/i *font*

³Essi però si distinguono in due grandi gruppi: 'con grazie' (*serif*) e *sans serif* (in italiano: 'senza grazie' o 'lineare' o 'bastoncino'). Quelli del primo gruppo contengono segni

ogni font si può anche decidere quale dimensione (corpo) utilizzare: oggi però il ‘corpo’ non designa più una dimensione fisica del carattere (come era con le lettere metalliche di un tempo), ma designa la sua dimensione virtuale la quale implica anche un’interlineatura, cioè una distanza fra le righe che consenta una lettura agevole.

Digitando i tasti sulla tastiera di un computer, dunque, non si attivano i caratteri che si osservano sulla tastiera e nemmeno quelli che si vedono sullo schermo, ma si attivano i caratteri che appartengono ad una collezione (i font), cioè un insieme di disegni (*glifi*) che descrivono i caratteri (minuscoli e/o maiuscoli per le lettere, i numeri, i segni di interpunzione ecc.) che si vedono sullo schermo. La visualizzazione a schermo dell’algoritmo che sta sotto i tasti, inoltre, dipende sia dallo schermo, sia dal software, sia dalla macchina sulla quale esso è installato; la visualizzazione sulla carta dipende poi dalla stampante che si utilizza per cui accade che i caratteri digitati (*input*) possono avere (*output*) sullo schermo un determinato aspetto e sulla carta un aspetto diverso: è quanto può accadere con i file .pdf generati all’esterno di Adobe (ad es. tramite i normali wp).

Questo problema è stato risolto tramite due innovazioni fondamentali: la codifica dei font e l’invenzione di una stampante virtuale, indipendente dalla stampante fisica di cui tutti abbiamo esperienza.

La codifica risponde all’esigenza di trasmettere i font (cioè l’informazione che essi rappresentano) fra un punto e l’altro della rete così da rendere inequivocabile il trasferimento e la comprensione dell’informazione⁴. Essa avviene assegnando un numero ad ogni carattere⁵, utile anche per la pagine

quasi invisibili che consentono di ‘legare’ i caratteri fra loro per compattare meglio le righe in modo da non affaticare l’occhio del lettore.

I font si distinguono in diversi modi: ad esempio, in ‘proporzionali’ (che destinano un spazio diverso per ogni carattere) e ‘a spaziatura fissa’ che invece destinano ad ogni carattere uno spazio uguale. Per esempio:

«questo è un esempio di spaziatura proporzionale E ANCHE QUESTO» mentre
«questo è un esempio di spaziatura fissa E ANCHE QUESTO».

⁴L’alfabeto Morse, alla metà dell’Ottocento, codificando ogni lettera o numero con una serie di punti e di linee, consentiva di trasmettere e di leggere i messaggi indipendentemente dalla grafia del mittente: chi non conosceva il codice non poteva decodificare l’informazione. Analogamente Enigma, il sistema di codifica dei caratteri utilizzato dai tedeschi durante la II guerra mondiale, non consentì di intercettare i messaggi fino a che A.M. Turing non riuscì a decodificarli.

⁵All’inizio, i caratteri sono stati codificati dallo ASCII (*American Standard Code for Information Interchange, o US-ASCII*) che codificava i caratteri alfabetici latini della lingua inglese senza le lettere accentate, assenti da quella lingua. Successivamente le accentate vennero comprese nella codifica *extended ASCII*: questo passaggio richiese di passare da stringhe di 7 bit ($2^7 = 128$ numeri, corrispondenti a 95 caratteri stampabili, compreso uno spazio, e 33 caratteri non stampabili ma di controllo) a stringhe di 8 bit ($2^8 = 256$ numeri, un byte).

web, ecc. ([5] § 47.7) per cui, in fase di input da tastiera, non vi sono più equivoci.

Rimaneva però il problema della decodifica dei caratteri da parte dell'output, cioè dello schermo e/o della stampante. La 'stampante' virtuale è il linguaggio PostScript⁶. Adobe ha poi sviluppato `ghostscript`, una collezione di programmi liberi basata su un interprete dei linguaggi di descrizione di pagina PostScript e Portable Document Format utilizzabile in vari s.o. (Unix e derivati, Mac OS X, VMS, Windows, OS/2, Mac OS classico e DOS).

Questa sintetica e semplificata digressione sui caratteri, sui font e sulla loro codifica può consentire di rendere edotti gli Autori del motivo per cui lo Stampatore (Lulu.com/) può richiedere interventi manuali per accettare i file .pdf inviati, ma generati al di fuori del Distiller di Adobe o di `ghostscript`.

3 Software sincrono e asincrono

I wp (la 'video composizione') sono programmi per la composizione di testi dove si ha l'impressione che vi sia corrispondenza fra quanto si vede sullo schermo e quanto apparirà sulla stampata perché la relazione fra compositore (il tastierista) e il programma sono interattivi (c.d. 'composizione sincrona'): ogni gesto del tastierista implica l'immediata variazione del testo a video.

I vantaggi della sincronia però vanno a scapito della qualità della composizione, perché quest'ultima dipende da una elaborazione accurata sul testo, che i wp più diffusi non fanno, nonostante il loro sensibile miglioramento avvenuto nel corso degli ultimi anni.

Oggi lo standard **UNICODE**, o ISO 10646, ha l'obiettivo di codificare tutti caratteri di tutte le lingue del mondo (vive e morte), indipendentemente dalle piattaforme e dai sw di utilizzo ed è supportato dai moderni standard della programmazione, del markup e dei sistemi operativi.

Originariamente pensato come una codifica a 16 bit (2^{16} con possibilità di codificare 65.536 caratteri), UNICODE ora prevede una codifica fino a 2^{21} bit che supporta un repertorio di codici numerici che possono rappresentare circa un milione di caratteri. Ciò appare sufficiente a coprire anche i fabbisogni di codifica di scritti del patrimonio storico dell'umanità, nelle diverse lingue e negli svariati sistemi di segni utilizzati.

Lo codifica standard per le lingue occidentali, oggi il **UTF-8** (*Unicode Transformation Format*, 8 bit) combina i caratteri di controllo US-ASCII con quelli a 8 bit.

⁶PostScript è un linguaggio di programmazione volto alla descrizione di pagine sviluppato da Adobe Systems nel 1976. L'obiettivo principale era lo sviluppo di uno strumento per descrivere pagine di testo e grafica in modo indipendente dalla risoluzione e dal dispositivo di visualizzazione, in modo da trasferire da un computer all'altro le informazioni senza perdita di qualità. Steve Jobs, con il progetto NeXT, pensava di far gestire sia lo schermo sia la stampa con il linguaggio PostScript.

La ‘composizione asincrona’ consiste invece nel digitare il testo senza alcuna formattazione, cioè senza occuparsi del suo aspetto grafico, e senza occuparsi della formattazione del testo nel suo complesso: questi aspetti verranno curati, successivamente e automaticamente da altri programmi che hanno ricevuto determinate istruzioni. Queste ultime poi possono essere anche modificate successivamente, interessando il solo aspetto grafico del testo per migliorarne la resa editoriale.

La composizione avviene dunque in due fasi: l’introduzione di puro testo e l’ottimizzazione della composizione. In questo modo si raggiunge una migliore qualità dato che non si tiene conto della velocità di visualizzazione, ma della qualità della composizione ([1], § 1.1). L’ottimizzazione della composizione e dell’impaginazione, non si realizza dunque con i sw sincroni, ma può realizzarsi con quelli asincroni.

A questo punto però interviene una questione dei costi monetari dei sw: è noto che i sw di Adobe, ad esempio, hanno prezzi relativamente molto elevati ed è quindi conveniente ricorrere a sw gratuiti.

Fra i sw sincroni **gratuiti** si ricordano la *suite* OpenOffice di Oracle, la *suite* LibreOffice che nasce da una ‘costola’ della OpenOffice Community e la *suite* ABIWord. Fra i sw asincroni gratuiti si ricordano Scribus e alcuni derivati dal sistema $\text{T}_{\text{E}}\text{X}$ ⁷. Mentre però Scribus appare più adatto alla stampa pubblicitaria, i derivati di $\text{T}_{\text{E}}\text{X}$ sono maggiormente adatti alla composizione di libri ([1], § 1.3).

4 Programmi sincroni o a scrittura visuale

Questi programmi richiedono dunque all’Autore di occuparsi di definire la maggior parte dei parametri che connotano la pagina, i caratteri del testo, delle note, delle didascalie, i titoli dei capitoli, dei paragrafi, dei sotto-paragrafi, le testatine e i piè di pagina ecc.

Fra quelli gratuiti, per le *suite* LibreOffice e ABIWord, oltre che le guide on line, vi è anche un’interessante guida operativa negli *Appunti di informatica libera* di D. Giacomini ([5]: cap. 59).

⁷Il sistema comprende il sw $\text{T}_{\text{E}}\text{X}$, inventato da D.E. Knuth nel 1978, e i suoi derivati (Plain $\text{T}_{\text{E}}\text{X}$, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, Con $\text{T}_{\text{E}}\text{X}$ e Lua $\text{T}_{\text{E}}\text{X}$. Altri sistemi che si avvalgono di $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ sono LyX e TeXmacs ([1], § 4.8-11))

5 $\text{\LaTeX} 2_{\epsilon}$

Fra i ‘derivati’ di \TeX gratuiti vi è $\text{\LaTeX} 2_{\epsilon}$ (così si indica la versione più aggiornata e stabile di \LaTeX)⁸; come si vede dall’indicazione della versione la sua evoluzione è in pieno svolgimento: oggi (2017) nel web sono reperibili anche diverse app gratuite per smartphone che consentono di rammentare i principali *mark up* abitualmente utilizzati.

La $\text{\TeX}Live$ (oggi 2017) è la distribuzione multi-piattaforma che consente di installare il software abbastanza agevolmente.

Successivamente, si compone il testo con uno *shell editor*⁹ cui si aggiungono una serie di comandi specifici (i *mark-up* di \LaTeX). Il file digitato che ne risulta è il sorgente con estensione .tex che viene poi ‘compilato’, cioè lo si fa ‘leggere’ e ‘tradurre’ da altri sw ausiliari specifici, che rifiniscono automaticamente il file sorgente sulla base delle indicazioni fornite, e, infine, da $\text{pdf}\text{\LaTeX}$, il cui output è direttamente un file .pdf. Tutti questi sw sono gratuiti e sono compresi nella $\text{\TeX}Live$.

6 Imparare \LaTeX e usare *Share\LaTeX*

Si tratta dunque di imparare i comandi specifici (i *mark-up*) da inserire nel file composto con l’*editor*: i processi successivi sono automatici. Anche se la curva di apprendimento è un po’ ripida, sono tuttavia disponibili molte guide gratuite on line e in diverse lingue¹⁰ che facilitano l’apprendimento e la soluzione di diversi problemi: i vantaggi in termini di *editing* sono incredibili, analogamente al risparmio nei tempi di composizione.

Lo sforzo e il tempo richiesti all’inizio per superare la ripida curva di apprendimento è sicuramente inferiore allo sforzo e al tempo richiesti per apprendere un utilizzo avanzato di un software sincrono. Il rapporto qualità-prezzo di un prodotto sviluppato in \LaTeX rispetto ad un prodotto sviluppato

⁸Creato nei primi anni Ottanta da Leslie Lamport, ampliando i *mark up* di \TeX , ha raggiunto ragguardevoli livelli di precisione grazie all’apporto di molti autori che vi hanno contribuito e che vi contribuiscono tuttora trattandosi di un sw *open source*. Il manuale di Lamport è ovviamente molto tecnico: per una semplificazione si veda [6].

⁹Programmi accessori che consentono di inserire il testo. Gli *shell editor* sono molti e gratuiti, alcuni sono multipiattaforma altri sono adatti alle macchine (Linux, Mac e Windows) sui quali sono installati. Per una disamina circostanziata si veda [1]: § 4.7.

¹⁰Per l’Italiano si veda \GjT . Si può partire dalla guida semplificata di Lamport [6], per passare alla guida per l’*impaziente* [7], poi all’*Arte* di L. Pantieri e T. Gordini [8] e, infine, alla fondamentale di Guida \GjT di C. Beccari [1].

Per gli appassionati: *The Elements of Typographic Style* [2] e *\TeX* [3].

con un wp sincrono è estremamente interessante e professionale. In termini tipografici il risultato è impareggiabile.

Da qualche tempo vi è anche la possibilità di utilizzare L^AT_EX in *sharing*: i vantaggi di questa soluzione sono sostanzialmente due:

- la possibilità che più soggetti possano intervenire direttamente sul medesimo sorgente e di compilarlo on line per metterlo a punto e per vedere il risultato in PDF. Ciò consente, ad esempio all'Editore, di aiutare gli Autori a migliorare l'apprendimento con qualche suggerimento, che deve però innestarsi su di una minima competenza di base già acquisita o nella disponibilità ad acquisirla avvalendosi delle guide appena citate;
- la caduta della necessità di installare la T_EXLive sulla propria macchina. Tuttavia l'esperienza ha dimostrato che disporre del sw in locale consente di provare le soluzioni tipografiche individuate, prima di inserirle nello *sharing*, evitando così di trasferire errori e imprecisioni sul file condiviso e bloccarne la compilazione.

7 *TrueType, Type 1... embedded vs subset*

Quando si carica su Lulu.com/ un file .pdf generato al di fuori di Adobe, come accade sovente dato l'elevato costo del sw, lo Stampatore spesso non lo accetta perché i font non sono *embedded* (incorporati nel .pdf) e quindi richiede di procedere manualmente per incorporarli (v. *supra*, § 2 e *selfpublishing*). Se invece si carica un file .doc, lo Stampatore restituisce un file .pdf. Ma il nostro .pdf originario, proveniente dalla trasformazione di un file .doc, risulta completamente diverso dal .pdf restituitoci dallo Stampatore: infatti non si è mantenuta la formattazione che avevamo prescelto con tanta fatica, generando un .pdf dal nostro .doc, poi dato in pasto allo Stampatore.

Il che può accadere per due motivi:

- perché quanto si vede sullo schermo di un .pdf prodotto all'esterno di Adobe può essere diverso da quanto si vede sulla carta. Se apriamo infatti il .pdf con Acrobat Reader e osserviamo, fra le Preferenze, i font vediamo un elenco di specifici font (TrueType, Type 1 ecc.) cui segue l'indicazione «*Embedded subset*»: ciò significa che quel font non è stato incorporato del tutto nel file .pdf che osserviamo, ma che ne è stata incorporata una parte, un sottoinsieme (*subset*). Se, dunque, generiamo un .pdf, a partire per esempio dal nostro file .doc, il sottoinsieme incorporato può essere diverso dal sottoinsieme incorporato dallo Stampatore nel .pdf che ci ha restituito.

- perché lo Stampatore si preoccupa di produrre file archiviabili (PDF/A). Se si dà in pasto allo Stampatore un file composto in L^AT_EX sappiamo che il file .tex sorgente, prima di giungere al .pdf, ha subito un trattamento molto più elaborato sui font¹¹ e, soprattutto, ha fruito automaticamente dei sw `ghostscript` e `ghostview` che, unitamente al `pdflatex`, hanno la capacità di produrre, per l'appunto, file archiviabili ([1]: § 22 e 22.2.) per cui è più facile che esso venga immediatamente accettato dallo Stampatore (v. anche *supra* § 2).

Riferimenti bibliografici

- [1] Beccari C. *L'arte della composizione tipografica con L^AT_EX*, G_JT_R Versione B5-0.99.16 — 2017/03/08.
- [2] Bringhurst, *The Elements of Typographic Style*, Hartley & Marks, Seattle (WA) 2012.
- [3] Eijkhout, V., *T_EX by Topic - a T_EXnician's Reference*, Dante - Lehmanns media, Berlino 2014.
- [4] Naveh, J. *Origins of the Alphabets*, The Jerusalem Publishing House Ltd, Jerusalem s.d.
- [5] Giacomini D., *Appunti di informatica libera*, Ver. 03.03.2013, Vol. II.
- [6] Oetiker, T., Partl, H., Hyna, I., e Schlegl, E. *Una (mica tanto) breve introduzione a L^AT_EX 2_ε ovvero L^AT_EX 2_ε in 93 minuti*, qui (Traduzione italiana a cura di G. Agostini, G. Bilotta, F. Casadei Della Chiesa, O. de Bari, G. Delre, L. Ferrante, T. Pecorella, M. Rigido, R. Zanasi), Versione 3.16, 25 settembre 2000.
- [7] Pantieri L., *L^AT_EX per l'impaziente*, G_JT_R 2009.
- [8] Pantieri L., Gordini T., *L'arte di scrivere con L^AT_EX*, G_JT_R 2012

¹¹In estrema sintesi, i font visibili sullo schermo fino ad una trentina di anni fa erano a matrice di punti e sullo schermo apparivano spesso slabbrati o sgranati. Lo sviluppo della tecnologia ha inciso notevolmente sugli schermi e anche sul disegno dei font: D.E. Knut (l'inventore di T_EX) poté così sviluppare caratteri a matrice di punti molto compatti. Ora però i caratteri sono vettoriali: Adobe sviluppò il PostScript Type 1 cui seguì, a seguito di un accordo fra Apple e Microsoft ([1]: § 18.2), lo sviluppo dei font TrueType. Entrambe queste collezioni di caratteri sono ora in parte superate da altre famiglie (OpenType ecc.), ma L^AT_EX utilizza direttamente i caratteri TrueType.