

L. Battaia

L^AT_EX,
naturalmente

Indice



Pag. 1 di 45

L^AT_EX, naturalmente!

Luciano Battaia

Una miniguia di avvio per principianti assoluti

Versione del 9 febbraio 2007— Work-in-progress!

Sommario

In questa miniguia sono raccolti i materiali e le indicazioni fornite durante i corsi di introduzione a L^AT_EX, tenuti, come attività curricolare, agli studenti del Liceo Scientifico Grigoletti di Pordenone. L'impostazione è dunque quella di un manuale scolastico senza, in particolare, la pretesa di costruire una guida sistematica all'uso di L^AT_EX, ma solo un aiuto per entrare con tranquillità nel campo della videoscrittura professionale. Numerosi link rimandano a pubblicazioni specifiche per chi volesse approfondire i temi trattati.

Ho anche inserito alcune note con le motivazioni didattiche che mi hanno condotto ad inserire questo software nei curricoli scolastici, in particolar modo nei corsi ad indirizzo scientifico.

Indice

Premessa	3
1 Introduzione	5
1.1 Che cosa si può fare con L ^A T _E X	5
1.2 Che cosa non si può (o non conviene) fare con L ^A T _E X	5
1.3 Che cosa si deve fare con L ^A T _E X	6
1.4 L ^A T _E X a scuola	7
2 Installazione	9
2.1 Che cosa serve	9
2.2 Download dei pacchetti	10
2.3 Installazione vera e propria	10
2.4 Pronti, via!	11
3 Primi documenti	12
3.1 Qualche dettaglio su TeXnicCenter e WinEdt	12
3.2 Finalmente si scrive	12
3.3 Caccia all'errore	14
4 Allunghiamo un po' il passo	15
4.1 Un titolo per i documenti	15
4.2 Suddivisioni in parti	16
4.3 Sezioni numerate e non numerate	16
4.4 Creiamo l'Indice	17
5 Documenti pdf e link	19
5.1 Semplici documenti	19

5.2	Documenti con segnalibri	19
5.3	Link esterni	20
6	I packages	21
6.1	fancyhdr	22
7	Alcuni comandi importanti	24
7.1	Considerazioni generali	24
7.2	Caratteri speciali	24
7.3	Accenti e virgolette	25
7.4	Qualche comando di uso frequente	26
8	Liste	28
8.1	Tipi di liste	28
8.2	Liste annidate	29
9	Immagini e tabelle	31
9.1	Il formato delle immagini	31
9.2	Programmi di conversione	32
9.3	L’inserimento vero e proprio	32
9.4	Testo e immagini affiancate	33
9.5	Tabelle	34
10	Qualche cenno alle formule matematiche	36
10.1	Regole di base per le formule	36
10.2	Scrittura esplicita delle formule	37
11	Tips	39
11.1	Documenti lunghi	39
11.2	Classi di documenti	41
12	Link per ulteriori letture	42
13	Conclusione	44

Premessa

La premessa più adeguata ad una piccola guida di introduzione a \LaTeX ci è parsa la seguente citazione, presa da *Peter Flynn*, Formatting Information, Silmaril Consultants, reperibile anche in rete all'indirizzo <http://www.silmaril.ie/downloads/>.

No, the harder part might be *writing*, period. \TeX/\LaTeX is actually easy, once you relax and stop worrying about appearance as a be-all-and-end-all. Many people have become 'Word Processing Junkies' and no longer 'write' documents, they 'draw' them, almost at the same level as a pre-literate 3-year old child might pretend to 'write' a story, but is just creating a sequence of pictures with a pad of paper and box of Crayolas — this is perfectly normal and healthy in a 3-year old child who is being creative, but is of questionable usefulness for, say, a grad student writing a Master's or PhD thesis or a business person writing a white paper, etc. For this reason, *I* strongly recommend *not* using any sort of fancy GUI 'crutch'. Use a plain vanilla text editor and treat it like an old-fashioned typewriter. Don't waste time playing with your mouse.

Note: I am not saying that you should have no concerns about the appearance of your document, just that you should write the document (completely) first and tweak the appearance later ... *not* [spend time on] lots of random editing in the bulk of the document itself. [Heller, *New To \LaTeX ... Unlearning Bad Habits* (11 marzo 2003)]

No, la cosa più difficile dovrebbe essere scrivere, punto. \TeX/\LaTeX è in verità semplice, se ti rilassi e smetti di preoccuparti dell'aspetto come la cosa più importante. Molti sono diventati 'Drogati del Word Processing' e non scrivono documenti, li disegnano, quasi nello stesso modo in cui un bambino di tre anni, in età prescolare, potrebbe fingere di 'scrivere' una storia, mentre sta solo creando una sequenza di simboli con un blocco di carta e una scatola di pennarelli — questo è perfettamente normale e soddisfacente in un bambino di tre anni che sta diventando creativo, ma è di discutibile utilità, per esempio, per uno studente laureato che deve scrivere la propria tesi per il Master o PhD o per un uomo d'affari che deve scrivere un rapporto ufficiale, ecc. Per questa ragione invito caldamente a non usare alcun tipo di elaborata interfaccia grafica. Usa un text editor senza fronzoli e consideralo come una vecchia macchina da scrivere. Non perdere il tuo tempo a giocare con il mouse.

L. Battaia

L^AT_EX,
naturalmente

Indice



Pag. 4 di 45

Nota: non sto dicendo che non devi avere alcun interesse per l'aspetto del tuo documento, solo che dovresti prima scriverlo (per intero) e ritoccarne l'aspetto successivamente ...non [perdere tempo] a fare una gran mole di editing casuale nella maggior parte del documento stesso.

1. Introduzione

1.1. *Che cosa si può fare con \LaTeX*

Tra le tante cose che si possono fare con \LaTeX cito solamente alcune tra le più comuni.

- Produzione di documenti destinati alla stampa, da semplici articoli di una o due pagine, a complessi progetti editoriali di testi anche divisi in tomi.
- Produzione di documenti destinati alla lettura a video, anche con impaginazione adatta ad evitare lo scrolling delle pagine.
- Realizzazione di lucidi per conferenze o lezioni, riutilizzando in maniera molto efficace materiale adatto alla pubblicazione.
- Realizzazione di presentazioni mediante slide da utilizzare per conferenze o lezioni, in particolare, ma non solo, contenenti parecchie formule matematiche.
- Realizzazione di sofisticati grafici per illustrare testi scientifici.

Tutto questo si può fare con risultati di qualità molto elevata, con uno sforzo minimo, a mio avviso addirittura inferiore, a parità di risultati, a quello richiesto per i più diffusi sistemi di videoscrittura.

All'indirizzo <http://www.tug.org/texshowcase/> si possono vedere alcuni spettacolari esempi di realizzazioni con \LaTeX , tra cui, per esempio, complesse pagine di spartiti musicali.

1.2. *Che cosa non si può (o non conviene) fare con \LaTeX*

- Schifezze, ovvero documenti in cui contano solo il numero di font diversi utilizzati e le variazioni di stile dei caratteri, più che il contenuto e la leggibilità.
- Volantini o manifesti che richiedano una composizione complessa aggiustabile solo con continui interventi manuali.
- Le patatine fritte (che tra l'altro provocano anche un aumento del colesterolo e quindi sono da evitare...).

1.3. Che cosa si deve fare con L^AT_EX

- Testi a carattere scientifico o comunque contenenti parecchia matematica.
- Documenti redatti a più mani, in cui si vuole avere la assoluta certezza di uniformità di stile, senza bisogno di ricontrollare minuziosamente le varie parti.
- In generale tutti i documenti che debbano essere *ben formati* e gradevoli da consultare.

Per quanto riguarda i testi contenenti matematica segnaliamo in particolare tre fatti:

Formule complesse Nonostante i comuni software in commercio (es. MathType) consentano la realizzazione di formule abbastanza complesse, solamente con L^AT_EX si ha a disposizione una libreria completa di simboli oltre alla possibilità di un controllo minuzioso del look delle formule. La numerazione automatica delle formule è poi un'opzione a cui non si può rinunciare in un testo scientifico.

Problema del cambio di dimensione Se in un documento scritto con un Word Processor si devono cambiare le dimensioni del font utilizzato, occorre procedere ad un cambio manuale delle dimensioni in ciascuna formula, richiamando ogni volta il programma con cui esse sono state scritte. Con L^AT_EX questo problema non si pone, in quanto le formule matematiche sono parte del testo.

Formule in linea e centrate Le formule matematiche inserite in un word processor hanno normalmente le stesse dimensioni e lo stesso stile sia che siano in linea con il testo, sia che siano centrate in una riga a parte. L^AT_EX usa invece, di default, un diverso stile per i due casi, ottenendo così una spaziatura tra le righe molto più uniforme e un aspetto complessivo decisamente più gradevole. Non è poi da sottovalutare, anche in casi non complessi, il diverso look delle formule scritte con L^AT_EX rispetto a quelle scritte con un Word Processor supportato, per esempio, da MathType. Si vedano i due esempi di seguito, il primo costruito con Word e MathType (e qui importato come immagine), il secondo con L^AT_EX.

Questa formula è in linea col testo:

$$\int \frac{x + \ln\left(\frac{\sin x}{x}\right)}{\sqrt{1+x}} dx$$

Questa formula è in linea col testo: $\int \frac{x + \ln\left(\frac{\sin x}{x}\right)}{\sqrt{1+x}} dx$

1.4. L^AT_EX a scuola

L'informatica, come è ben noto, è una delle nuove priorità nei curricula scolastici. Che cosa si debba però insegnare sotto l'etichetta *informatica* è assolutamente poco chiaro, in particolare quando l'informatica non è una materia a sé, ma è una specie di appendice dell'insegnamento della matematica. Le scelte fatte nelle varie realtà sono molto diverse e vanno dallo studio sistematico di un linguaggio di programmazione (Pascal, Visual Basic, Javascript, C++, . . .), alla introduzione di alcuni pacchetti applicativi di uso comune (Fogli elettronici, Data Base, Software di calcolo simbolico, Software di geometria dinamica, Strumenti di presentazione), alla produzione di pagine web.

In tutte le realtà comunque l'introduzione della videoscrittura è considerata un requisito minimo e assolutamente irrinunciabile. Videoscrittura significa, nella stragrande maggioranza dei casi, lo studio di un word processor WYSIWYG (del tipo di Word, per intenderci). Questa scelta è avallata anche dalle indicazioni contenute nel Syllabus per l'ottenimento della patente europea ECDL, dove, pur senza riferimento ad alcun software specifico, si evince chiaramente che l'indicazione è per un software del tipo citato.

Del resto, per la grande maggioranza degli utilizzatori della videoscrittura, questa è sicuramente la scelta più semplice (in particolare da quando sono disponibili pacchetti free, come quelli della suite OpenOffice). La situazione cambia radicalmente quando si considerano gli ambiti accademici del settore matematico, fisico, informatico e, parzialmente, linguistico. In questi casi L^AT_EX costituisce lo standard *de facto*, tanto che in molti casi vi è addirittura l'obbligo di scrivere documenti ufficiali (es. tesi di laurea), utilizzando L^AT_EX, magari con particolari implementazioni messe a disposizione dalle stesse università.

Questa situazione mi ha fatto pensare alla opportunità di proporre almeno una introduzione dei fondamenti di L^AT_EX agli studenti del Liceo Scientifico, visto che molto probabilmente lo ritroveranno tra i piedi nel loro futuro.

Ci sono poi altri aspetti non secondari da tenere presente. L^AT_EX richiede all'utente una certa abitudine all'astrazione: non essendo WYSIWYG c'è differenza tra il "codice sorgente" e il "prodotto finito". Inoltre chi redige un testo deve impiegare un po' di tempo (non molto per la verità!) a pensare alla struttura logica del documento: non ci si può mettere alla tastiera e costruire le pagine, modificandole e aggiustandole pian piano, con l'evidente rischio di produrre la prima pagina con

uno stile e l'ultima con un altro stile. Tutto questo, secondo me, depone grandemente a favore di L^AT_EX proprio dal punto di vista della valenza didattica dell'apprendimento di un software.

Per chi poi ha interesse alla programmazione vera e propria, direi che L^AT_EX è l'ambiente naturale: un documento L^AT_EX è costruito con la logica con cui si costruisce un programma informatico, dove, dopo aver prodotto il codice si deve *compilare* il tutto per ottenere il prodotto finito. Se poi si ha la voglia (o la necessità) di produrre documenti complessi, ancora meglio: L^AT_EX (e il suo motore sottostante, T_EX) sono dei veri e propri linguaggi di programmazione (finalizzati alla produzione di documenti), per cui ci si può sbizzarrire a volontà.

É opportuno segnalare anche che un documento L^AT_EX ha una struttura non molto dissimile da una pagina web: chi ha familiarità con la costruzione di siti web troverà naturale l'uso di questo software. Assolutamente da non sottovalutare, ancora, il fatto che la traduzione di un documento L^AT_EX in una pagina web può avvenire in maniera molto naturale e senza l'introduzione di codice complicato come quando l'analogo lavoro si fa a partire da un documento prodotto con un software WYSIWYG.

A proposito dei vantaggi dell'uso di L^AT_EX rispetto ai normali programmi di videoscrittura si possono leggere, tra l'altro

- un articolo di Allin Cottrell in <http://ricardo.ecn.wfu.edu/~cottrell/wp.html>;
- una serie di pagine di Skylar Thompson, in http://www.earlham.edu/~thompsk/final_project/latex, che descrivono in dettaglio i problemi con i software WYSIWYG.

2. Installazione

Vengono qui fornite solo le indicazioni per una installazione di base, in particolare per la piattaforma Windows. Per maggiori dettagli si possono consultare i siti che verranno successivamente indicati, oppure visitare il sito ufficiale degli utilizzatori italiani di T_EX, il GUIT e andare alla apposita pagina di Installazione. Naturalmente questo sito è il miglior punto di partenza per una qualunque ricerca concernente L^AT_EX, e ne consigliamo una attenta ed accurata visita. L'indirizzo web è il seguente: <http://www.guit.sssup.it>.

2.1. Che cosa serve

Un sistema L^AT_EX funzionante deve comprendere almeno:

1. un pacchetto, adeguato alle diverse piattaforme (Windows, Linux, Macintosh, ecc.), contenente il L^AT_EX vero e proprio (con il sottostante motore T_EX);
2. un visualizzatore, per mostrare a video il “prodotto finito”;
3. un “editor” per scrivere il codice sorgente.

In seguito sarà necessario anche installare Ghostscript e Ghostview per la visualizzazione dei file *postscript*. Per la visualizzazione dei file pdf, il formato più comune per l'output dei file prodotti con L^AT_EX, occorrerà infine Acrobat Reader, che comunque dovrebbe essere presente in qualsiasi computer e sul quale dunque non vale la pena di fare ulteriori considerazioni.

Il pacchetto MiKTeX contiene la più comune installazione per Windows e comprende anche il visualizzatore più utilizzato, Yap: si tratta di una installazione assolutamente freeware. Come editor si potrebbe utilizzare anche il Blocco Note di Windows, ma è meglio servirsi di uno degli editor dedicati. Io uso WinEdt (attenzione a non confondersi con WinEdit) che mi pare estremamente efficiente e facilissimo da usare. Non è freeware, ma shareware: dopo 31 giorni di uso occorre registrarsi, per una cifra assolutamente irrisoria viste le potenzialità e visto che è praticamente l'unica cosa da pagare (30\$ per la licenza studente). Chi non volesse spendere nemmeno questa piccola cifra può servirsi, per esempio, di TeXnicCenter, anche se l'editor più efficiente (e free!) è senza dubbio GNU Emacs, che comunque non è facile da usare.

2.2. Download dei pacchetti

Occorre dunque innanzitutto scaricare i software necessari dalla rete. Iniziamo da MiKTeX: aprite la pagina <http://www.miktex.org/Setup.apx> e predisponetevi a scaricare una delle due versioni presenti “Basic MiKTeX Installer” oppure “MiKTeX Net Installer”. Per chi non ha problemi di spazio consiglio l’ultima opzione che installa il L^AT_EX completo di tutti i pacchetti aggiuntivi, tenendo comunque conto che alla fine dell’installazione lo spazio occupato sarà vicino al giga. Se scegliete la versione completa, vi verrà proposto di scaricare un piccolo file di circa 800k, lanciando il quale potrete poi scaricare il pacchetto completo, salvandolo sul vostro computer per la successiva installazione. Nell’apposita finestra che si aprirà scegliete una località vicina a voi per il download.

Per WinEdt (l’editor che consigliamo) bisogna aprire la pagina <http://www.winedt.com/> e scaricare l’ultima versione. Per TeXnicCenter fate la stessa cosa aprendo la pagina www.toolscenter.org. Infine per Ghostscript e Ghostview

2.3. Installazione vera e propria

Installate MiKTeX, seguendo le normali procedure di tutti i programmi Windows e accettando tutte le opzioni predefinite, in particolare consiglio di non cambiare i nomi che vi verranno proposti per le cartelle che MiKTeX usa. Di norma l’installazione avviene nella cartella “MiKTeX 2.5”, posta all’interno di “Programmi”. Il nome della cartella è poco felice, perchè contiene uno spazio, e sarebbe invece opportuno evitare gli spazi nelle cartelle che si riferiscono a L^AT_EX; comunque pare che a partire dalla versione 2.5 di MiKTeX ciò non costituisca più un problema. In ogni caso, per salvare i file che produrrete, vi sconsiglio di usare la cartella Documenti: è meglio creare una cartella direttamente in C:, con un nome che non abbia spazi.

Successivamente installate Ghostscript e Ghostview, nell’ordine e, se già non è presente sul vostro computer, AdobeReader.

Installate infine l’editor che avete scelto, TeXnicCenter o WinEdt, anche qui accettando tutte le opzioni di default. Nell’installazione di TeXnicCenter vi verrà chiesto di indicare il percorso per trovare L^AT_EX: dopo aver cliccato sul pulsante “Sfoglia” sarà sufficiente ricercare la cartella C:\Programmi\MiKTeX 2.5\miktex\bin e poi fare OK. Se avete scelto di installare WinEdt, al termine si lancia automaticamente il programma WinEdt, con una Configuration Wizard in primo piano: si può tranquillamente chiuderla, dopo aver tolto la spunta in basso a sinistra (per evitare

che la finestra compaia ogni volta: quando si diventerà esperti si potrà di nuovo aprirla). Chiudete anche WinEdt.

Ci restano da installare i dizionari italiani per l'ortografia, per WinEdt e TeXnicCenter. Per entrambi andate alla pagina http://www.guit.sssup.it/installazione/win_editor.html del GUIT, sotto la voce rispettiva e scaricate i dizionari. Per WinEdt il dizionario va inserito nella cartella "WinEdt Team\WinEdt\Dict". Successivamente aprite il menu Options, Dictionary Manager, scegliete la voce "Your Custom Dictionaries" nella casella "Dictionaries" e, nella casella "Definition" inserite %B\Dict\It.dic, sostituendo tutto quanto c'è già scritto. Attivate le voci Enabled, Load on Start, Save on Exit, Add New Words, Use for Completion, Compounded Words, e premete OK. Per TeXnicCenter aprite la cartella "Programmi", poi "TeXnicCenter" e infine "Language". In quest'ultima cartella copiate la cartella col dizionario. Chiudete tutte le cartelle e lanciate TeXnicCenter. Aprite il menu Tools-Options e andate alla scheda "Spelling". Alla voce Language selezionate "it". Mettete la spunta su "Check spelling while typing" (se non c'è già) e andate in fondo alla voce "Personal dictionary". Cliccando sul solito pulsante "Browse" ricercate la cartella dove si trova il dizionario italiano che avete appena inserito e selezionatelo. In ogni caso alla citata pagina di GUIT trovate tutte le informazioni che vi servono.

Dovete poi attivare il pacchetto per la sillabazione in lingua italiana: tramite il pulsante "Start", cercate la voce "MiKTeX 2.5" e lanciate "Settings". Andate alla scheda "Languages" e mettete una spunta su "italian". Premete "OK" e chiudete.

2.4. Pronti, via!

A questo punto dovremmo poter cominciare a scrivere i primi documenti. Come vedremo, per le opzioni di base non servirà altro. In questa fase non useremo programmi come Ghostscript e Ghostview, che saranno utili in seguito. Useremo invece Acrobat Reader: una delle peculiarità più importanti di L^AT_EX è infatti quella di produrre in maniera automatica file di tipo pdf, che possono essere visualizzati al meglio con Acrobat Reader. Credo inopportuno insistere su questo argomento, in quanto questo software fa ormai parte della dotazione di qualsiasi pc che si connette ad Internet. A proposito: una grande percentuale dei file pdf che si trovano in rete (e quasi tutti quelli a contenuto matematico) sono scritti proprio in L^AT_EX.

3. Primi documenti

Prima di cominciare a scrivere creiamo una cartella, direttamente in C:, dove metteremo i nostri file: è meglio se usiamo nomi senza spazi. Lanciamo poi l'editor che abbiamo scelto, e apriamo un nuovo documento con l'apposito pulsante o la voce di menu.

3.1. Qualche dettaglio su TeXnicCenter e WinEdt

Come tutti i programmi anche questi si imparano ad usare . . . usandoli. In questa guida indicheremo solo i comandi più urgenti, alcuni dei quali non hanno bisogno di commenti, visto che la loro grafica è molto simile a quella usata in tutti gli analoghi programmi Windows.

In questo momento ci interessa solo il pulsante di compilazione e visualizzazione: in TeXnicCenter il pulsante si chiama 'Build and view current file' (controllate prima che nella finestra 'Output Profile' ci sia la voce LaTeX=>DVI), in WinEdt si chiama 'TeXify' (come al solito basta passare sopra i pulsanti per vederne la descrizione).

3.2. Finalmente si scrive

Un documento L^AT_EX contiene il testo che vogliamo stampare, inframezzato dai comandi di formattazione e impaginazione. I comandi iniziano tutti con il \, e contengono o solo una lettera, o una parola, magari seguite da argomenti tra parentesi quadre o graffe. Il testo si scrive normalmente, come nel Blocco Note di Windows. Non preoccupatevi se scrivete più di uno spazio tra le parole: L^AT_EX ne mette sempre solo uno (si tratta di una facilitazione enorme rispetto ai programmi tradizionali di videoscrittura, dove capita spesso di vedere le parole distanziate in maniera disuniforme proprio per il diverso numero di spazi che, battendo sulla tastiera, si lasciano tra una parola e l'altra).

In un documento vuoto inserite esattamente le righe seguenti:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
Buongiorno mondo!
\end{document}
```

Clickate sul pulsante di compilazione e visualizzazione e, in WinEdt, aspettate che si apra e chiuda la finestra nera di compilazione (se non ci sono errori sarà così veloce che non riuscirete

nemmeno a vederla, altrimenti non si chiuderà: chiudetela con la ‘x’ solita e ricontrollate di aver scritto tutto giusto). In TeXnicCenter la finestra di compilazione è posta in fondo allo schermo. Se tutto è andato a buon fine si aprirà il visualizzatore Yap: il vostro documento è lì che vi aspetta!

Tornate a WinEdt o a TeXnicCenter, tramite la barra delle applicazioni di Windows, e modificate leggermente il documento scrivendo:

```
\documentclass[a4paper,12pt,italian]{article}
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
Buongiorno mondo!
\end{document}
```

Rifate la procedura di compilazione e visualizzazione: non è cambiato nulla, almeno a prima vista. In realtà abbiamo istruito L^AT_EX ad usare la sillabazione propria della lingua italiana, a scrivere *Capitolo* (e non *Chapter*), *Indice* (e non *Table of Contents*), ecc., e ad accettare i caratteri accentati (é, è, ì, ò, à, ù, e simili) che non sono previsti nella codifica di base. Siccome nel nostro esempio tutto questo non è stato usato, non ci siamo accorti di nulla.

Proviamo a scrivere un documento più complesso, stando attenti a non usare, per il momento, i seguenti caratteri:

$$$, \&, \%, \#, \-, \{, \}, \backslash$$

Se volete andare a capo lasciate una riga vuota (altrimenti L^AT_EX non andrà a capo). Nel modello che segue inserite quello che volete al posto degli asterischi, cercando di utilizzare anche lettere accentate per vedere l'effetto. Le parole con lettere accentate saranno sottolineate, in WinEdt, in rosso (come se fossero errate): non preoccupatevi; questo problema non c'è in TeXnicCenter. Astenetevi, per il momento, anche da simboli come <, > e altri simili.

```
\documentclass[a4paper,12pt,italian]{article}
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
*****
\end{document}
```

Provate a cancellare la riga `\usepackage[latin1]{inputenc}`, e controllate che le lettere accentate non compaiono nel prodotto finito.

Per chiudere questi primi esperimenti proviamo a mettere alcune parole in grassetto o corsivo. Selezioniamo le parole volute e clicchiamo sulla **B** o sulla **I** (in WinEdt) oppure sulla **F** o sulla **H!** (in TeXnicCenter) (questi pulsanti sono presenti sulle barre) e vedremo che la (o le) parole che abbiamo scelto compaiono sia in grassetto o corsivo, ma racchiuse in un comando (rispettivamente `\textbf{parola}`, e `\emph{parola}`). Naturalmente, compilando e visualizzando, i comandi non saranno stampati, ma avremo ottenuto l'effetto voluto!

3.3. Caccia all'errore

Prima di procedere oltre ci resta ancora un piccolo aggiustamento da fare. Se scriviamo un documento molto lungo e, visualizzandolo, ci accorgiamo di un errore, come faremo a risalire nel codice al punto dove si trova l'errore per correggerlo? Come vedrete, con documenti complessi la questione non è di secondaria importanza. Per fortuna Yap ci viene in aiuto. Nella finestra di Yap andate alla voce 'View' del menu, scegliete la voce 'Options' e poi la scheda 'Inverse Search'; sotto la voce 'Program' scegliete WinEdt o TeXnicCenter e cliccate su OK. Se ora farete doppio click su un punto qualunque del testo, si aprirà la finestra di WinEdt o di TeXnicCenter, con evidenziata, all'incirca, la zona nella quale avete fatto click. È questo il metodo usato per cercare e correggere gli errori.

4. Allunghiamo un po' il passo

Con le poche cose viste nella sezione 3 si potrebbe scrivere un buon 50% dei documenti che servono nelle applicazioni. Di norma quello che serve subito dopo è aggiungere un titolo al documento e suddividerlo opportunamente in parti. La cosa potrebbe essere fatta manualmente, mettendo magari i titoli su una riga da soli e in grassetto (come si fa in un normale programma di videoscrittura), ma, *ovviamente, non* si deve fare così, altrimenti che L^AT_EX sarebbe?

Riguardando gli esempi proposti nella sezione 3, vediamo che i documenti sono costituiti da due parti:

1. una prima parte, detta *preambolo*, che comincia con `\documentclass` e termina prima di `\begin{document}`;
2. una seconda parte, detta *corpo*, che comincia con `\begin{document}` e contiene tutto il testo fino a terminare con `\end{document}`.

Questa distinzione è molto importante ed è da tenere sempre presente. Corrisponde alle sezioni `<head>` e `<body>` di un documento html.

4.1. Un titolo per i documenti

Per inserire un titolo al vostro documento modificate l'ultimo codice proposto così:

```
\documentclass[a4paper,12pt,italian]{article}
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\title{Il titolo che volete voi}
\author{L'autore dell'articolo}
\date{}
\begin{document}
\maketitle
*****
\end{document}
```

La riga `\author{}` è facoltativa; la riga `\date{}` è obbligatoria (se volete un titolo); se le parentesi graffe dopo `\date` sono vuote non verrà stampato nulla, altrimenti verrà stampata la data che ci inserirete. La riga `\maketitle` va messa subito dopo l'inizio del documento (è lì che vogliamo il titolo!).

4.2. *Suddivisioni in parti*

In questa mini introduzione parliamo solo della suddivisione di un documento in sezioni e sottosezioni: L^AT_EX prevede un maggior numero di livelli, ma non ci interesseranno qui.

La cosa interessante di L^AT_EX è che se decidiamo di suddividere il nostro documento in parti, cosa quasi obbligatoria a meno che non si tratti di una paginetta o due, e se vogliamo che le parti siano numerate in progressione, non dovremo fare assolutamente nulla: L^AT_EX farà tutto da solo, aggiornandosi automaticamente ad ogni modifica (aggiunta o eliminazione di suddivisioni).

4.3. *Sezioni numerate e non numerate*

Per inserire una sezione che sarà automaticamente numerata in ordine progressivo basta mettere il comando `\section{Titolo sezione}`; analogamente per una sottosezione numerata il comando sarà `\subsection{Titolo sottosezione}`.

Se vogliamo invece che le sezioni (o sottosezioni) non siano numerate, basterà inserire il seguente codice modificato: `\section*{Titolo sezione}` o `\subsection*{Titolo sottosezione}`.

Le sezioni non numerate possono essere l'Introduzione, o la Premessa. È anche possibile inserire un 'abstract' (di solito subito dopo il titolo), con il comando `\begin{abstract}`, seguito dal testo dell'abstract stesso, scritto come un normale testo, con le stesse opzioni. L'abstract termina con `\end{abstract}`.

Si deve ricordare che una sezione, o una sottosezione, termina quando comincia la successiva.

Un esempio:

```
\documentclass[a4paper,12pt,italian]{article}
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\title{Il titolo che volete voi}
\author{L'autore dell'articolo}
```

```
\date{}
\begin{document}
\maketitle
\begin{abstract}
  In questa zona inseriamo il sommario del nostro documento.
\end{abstract}
\section*{Introduzione}
  Questa è l'introduzione.
\section{Sezione}
  Testo della sezione
\section{Sezione}
\subsection{Sottosezione}
  Testo della sottosezione.
\subsection{Sottosezione}
  Testo della sottosezione.
\end{document}
```

4.4. Creiamo l'Indice

Forse il titolo più appropriato per questa sezione era: 'Facciamo creare l'indice a L^AT_EX': noi infatti non dobbiamo fare praticamente nulla, se non inserire un apposito comando al posto giusto, di solito subito dopo l'abstract e prima dell'introduzione, e un'istruzione nel preambolo. Il comando è semplicissimo, `\tableofcontents`, e lo mettiamo subito dopo la fine dell'abstract. L'istruzione è altrettanto semplice, `\setcounter{tocdepth}{1}`, e la mettiamo nel preambolo. Questa istruzione serve a far capire a L^AT_EX fino a che livello di dettaglio costruire l'indice. Con 1 abbiamo solo le sezioni, se sostituiamo 1 con 2, tra le parentesi graffe, avremo anche le sottosezioni.

Solo un'osservazione relativa a TeXnicCenter: perchè l'indice venga effettivamente creato dobbiamo compilare due volte (il pulsante TeXify di WinEdt lo fa automaticamente): col passare del tempo capiremo il perchè di questa doppia compilazione.

Le sezioni non numerate non vengono inserite nell'indice. Se si vuole forzare l'inserimento anche di queste sezioni si può aggiungere, subito dopo l'intestazione della sezione, una apposita istruzione, come nell'esempio che segue:

```
\section*{Titolo della sezione}
```

L. Battaia

L^AT_EX,
naturalmente

Indice



Pag. 18 di 48

```
\addcontentsline{toc}{section}{Titolo della sezione}  
\subsection*{Titolo della sottosezione}  
\addcontentsline{toc}{subsection}{Titolo della sottosezione}
```

5. Documenti pdf e link

Nel comporre un documento da distribuire in formato elettronico la scelta più naturale è quella di utilizzare il formato pdf, che è ormai uno standard consolidato.

5.1. Semplici documenti

La creazione di documenti pdf è assolutamente automatica con L^AT_EX, senza bisogno di alcuna modifica al testo che abbiamo costruito con l'editor e visualizzato con Yap. Invece di usare la compilazione L^AT_EX, dovremo usare la compilazione 'PDF L^AT_EX': in WinEdt si usa il pulsante dal nome 'PDF TeXify' che ha come immagine il logo di Acrobat con sovrapposto il lupo; in TeXnicCenter è lo stesso pulsante di prima, ma dovremo cambiare la voce nella finestra 'Output Profile', scegliendo LaTeX=>PDF (come prima, in TeXnicCenter se dobbiamo far comparire l'indice bisognerà compilare due volte). In ogni caso, se avremo installato Acrobat Reader, potremo immediatamente visualizzare il nostro file in un 'pdf decente' (decisamente migliore e più 'leggero' di un pdf che si produce a partire da un qualsiasi programma WYSIWYG).

Vi chiederete perché non si usa direttamente questo sistema invece della compilazione L^AT_EX (magari doppia) e della successiva visualizzazione con Yap. Ci sono diverse ragioni, su cui non possiamo qui entrare; quello che ci interessa è che con Yap è possibile fare in maniera semplice la 'caccia all'errore', cosa non facile con Acrobat Reader.

5.2. Documenti con segnalibri

Per documenti complessi è molto utile la barra di navigazione che contiene l'indice del documento e permette di raggiungere rapidamente la posizione desiderata. La creazione di documenti completi di barra (e anche di Thumbnails, ovvero miniature delle pagine che si possono sempre vedere nella barra di navigazione) richiede solo l'aggiunta di due righe nel preambolo, precisamente:

```
\usepackage{hyperref}  
\hypersetup{colorlinks=true,linkcolor=blue}
```

Il 'linkcolor' influenza il colore delle voci nell'indice. Al posto di blue si può mettere anche un altro colore (per esempio black se si vuole che non ci siano differenze tra l'indice e il resto del testo). Si otterrà anche l'effetto di rendere l'indice cliccabile, come in una normale pagina web!

5.3. *Link esterni*

Con la stessa facilità con cui si inserisce la barra di navigazione, si possono inserire link tra le varie parti del documento (quelli che in `html` si chiamano segnalibri) e link esterni, ad altri file o a pagine web su siti internet.

Limitiamoci a considerare link a siti esterni sul web (per link a segnalibri bisogna prima creare i segnalibri, cosa comunque facile). La sintassi per questo è molto simile a quella che si usa nelle pagine web. Basta inserire il comando `\href{url}{testo del link}`, dove `url` è l'indirizzo web a cui vogliamo puntare (`http://www.etc.etc`), mentre `testo del link` è il testo che diventerà cliccabile per accedere all'indirizzo scelto. Il testo di `url` può contenere tutti i caratteri normalmente usati per gli indirizzi web, con l'eccezione dell'underscore, che, se presente, deve essere preceduto da un backslash, così: `_`.

Se il testo cliccabile può essere uguale al link vero e proprio, la cosa è ancora più semplice: basta usare il comando `\url{url}`, mettendo al posto di `url` tra parentesi graffe l'indirizzo effettivo.

6. I packages

L^AT_EX è costruito in modo da comprendere la maggior parte delle funzioni di base per la composizione di documenti. È ovvio comunque che l'inclusione di tutte le possibili opzioni avrebbe reso il software stesso ingestibile. Inoltre quasi certamente non sarebbe nemmeno possibile prevederle tutte a priori. Per questo L^AT_EX è estensibile (anche se non in maniera elementare!), e chiunque ne abbia le capacità può aggiungere le funzionalità che servono per scopi particolari: è così che sono nate, per esempio, le estensioni per scrivere musica, o quelle per creare presentazioni tipo Power Point (anzi decisamente più efficienti, e non solo per l'uso scientifico, secondo me).

Le estensioni sono contenute in 'packages' aggiuntivi che devono essere richiamati e caricati nel preambolo di un documento che li usa: è questo il motivo del comando `\usepackage` che abbiamo già incontrato.

I pacchetti prodotti al momento dell'installazione di MiKTeX vengono caricati al momento dell'installazione di cui abbiamo parlato. Altri pacchetti prodotti successivamente, o aggiornamenti di vecchi pacchetti, possono venire caricati al bisogno utilizzando una particolare funzione di MiKTeX, il MiKTeX Package Manager, reperibile dal menu Start, sotto la voce MiKTeX, Browse Packages. Per caricare nuovi package dovete essere collegati ad Internet (se avete installato il MiKTeX completo vengono già installati tutti i package, per un totale di circa 900 Mb di spazio disco): aperto il Package Manager attendete che si aggiorni il database (qualche secondo) e poi, dall'elenco che compare, scegliete il package che vi serve; se non è installato, in altro a sinistra si attiverà il segno '+', cliccando sul quale si accederà alla procedura di installazione che farete seguendo le istruzioni a video (andando in Internet quando il software ve lo chiederà, e scegliendo un sito a voi vicino per il download). Accettate pure tutte le opzioni che vi vengono proposte. Per fare l'update di pacchetti già installati dovete invece usare la voce Update di MiKTeX, sempre nel pulsante Start.

Nello stesso Package Manager, se un pacchetto è già installato, una volta selezionato si attiverà anche un'icona (in parte al segno meno in alto), cliccando sulla quale potrete vedere quali sono i file sul vostro pc che si riferiscono a questo pacchetto: è un modo per rintracciare eventuali documentazioni che non riuscite a trovare in altro modo.

Per sapere quali package servono, invece, bisogna fare un po' di esperienza . . .

Tra i package di base, di cui non abbiamo ancora parlato, segnaliamo il 'Geometry package', che serve a modificare i margini delle pagine e alcune altre impostazioni 'geometriche' di base. Un uso classico di questo package è dato nel seguente esempio:

```
\usepackage[margin=3cm]{geometry}
```

Usate comunque con parsimonia la possibilità offerta da questi package di modificare lo stile predefinito di L^AT_EX, e chiedetevi sempre prima se ne vale la pena e se proprio non potete farne a meno.

Per imparare ad usare i vari package si possono leggere le guide (alcune anche di centinaia di pagine!) che vengono installate nella sottocartella `doc` della cartella di MiKTeX; di solito si tratta di guide in formato pdf, ovviamente prodotte con L^AT_EX.

6.1. *fancyhdr*

A mo' d'esempio di un package di uso non standard, ma fortemente consigliato, citiamo `fancyhdr`, ottimo, e molto usato, per la personalizzazione delle intestazioni e piè di pagina.

Come già detto, il modo migliore per imparare ad usare un pacchetto è quello di leggere il manuale e poi di . . . usarlo. Qui forniamo un esempio semplice tanto per partire, anche se, adattando opportunamente questo esempio, molto probabilmente non avrete bisogno d'altro!

Dopo aver installato il package, inserite nel preambolo del documento il seguente codice, seguendo le indicazioni riportate subito dopo:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\lhead{*1*}
\chead{*2*}
\rhead{*3*}
\lfoot{*4*}
\cfoot{*5*}
\rfoot{*6*}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

- `\lhead` è la scritta che comparirà sul margine sinistro dell'intestazione;
- `\chead` è la scritta che comparirà al centro dell'intestazione;
- `\rhead` è la scritta che comparirà sul margine destro dell'intestazione;
- `\lfoot` è la scritta che comparirà sul margine sinistro del piè di pagina;

- `\tfoot` è la scritta che comparirà al centro del piè di pagina;
- `\rfoot` è la scritta che comparirà sul margine destro del piè di pagina;
- `\renewcommand{\headrulewidth}{0.4pt}` serve a tracciare una linea orizzontale, di spessore 0.4pt, sotto l'intestazione;
- `\renewcommand{\footrulewidth}{0.4pt}` serve a tracciare una linea orizzontale, di spessore 0.4pt, sopra il piè di pagina;

Le cose che potete inserire nell'intestazione o nel piè di pagina vanno al posto di `*1*`, oppure `*2*`, oppure `*3*`, oppure `*4*`, oppure `*5*`, oppure `*6*`. Esempi di cose che di solito si inseriscono sono: il titolo del documento, l'autore, la data (metteteci la data che volete, oppure `\today` se volete che compaia la data di compilazione), il numero di pagina (basta metterci `\thepage`). Se lasciate vuoto lo spazio tra le parentesi graffe non verrà inserito nulla. Potreste anche inserire numero e titolo delle sezioni, ma questo esula dallo scopo di questa breve introduzione. Naturalmente tutti i formati, compreso lo stile dei numeri di pagina, possono essere modificati a piacere.

7. Alcuni comandi importanti

7.1. Considerazioni generali

I comandi di formattazione, sparsi nel documento, iniziano sempre con un `\` e sono composti o da un solo carattere, o da più caratteri. Il comando termina con il primo spazio dopo il nome del comando stesso. In questo c'è una differenza con `html`, dove i `tag` iniziano con `<` e terminano con `>`. Se i comandi sono seguiti da un argomento tra parentesi graffe, non ci sono problemi, altrimenti lo spazio dopo il comando sarà considerato come carattere di fine comando e non sarà stampato. Per ottenere lo spazio dopo un comando senza argomenti bisogna obbligare L^AT_EX a inserire uno spazio, mettendo un `\` seguito da un spazio: `_` è uno spazio forzato. Vediamo un esempio, con il comando `\LaTeX`.

- Se scrivete `\LaTeX mi piace` otterrete: L^AT_EXmi piace.
- Se scrivete `\LaTeX_mi piace` otterrete: L^AT_EX mi piace.

Per esercizio (ma promettete di non farlo mai, mai più!!) provate ad imbrogliare L^AT_EX mettendo da qualche parte del vostro documento una cosa del tipo `_ _ _`: otterrete tre spazi vuoti di fila, una cosa che L^AT_EX odia!

7.2. Caratteri speciali

Alcuni caratteri hanno, in L^AT_EX, un significato speciale, in quanto vengono usati nella formattazione e non possono essere utilizzati così come sono. Si tratta dei caratteri

$$$, \&, \% , \# , _ , \{ , \} , \^ , \sim , \backslash$$

Per ottenere i primi sette, occorre farli precedere da un backslash, così:

$$\backslash$, \&, \% , \#, _ , \{ , \}$$

Il segno di `~` serve per inserire uno *spazio insecabile*, ovvero uno spazio tra due parole che non verranno mai separate, nemmeno in fondo riga. L'accento circonflesso si usa come accento, ma, in questo caso, va ovviamente messo sopra un'altro carattere (e vedremo come). Se, per qualche motivo, si vogliono scrivere questi tre caratteri così come sono si possono usare i seguenti comandi:

1. `\textasciicircum`
2. `\textasciitilde`
3. `\textbackslash`

In WinEdt per ottenere alcuni caratteri che non sono presenti sulla tastiera basta aprire l'apposita barra con il pulsante \ddot{E} . In TeXnicCenter la cosa è più complessa: conviene farsi una specie di “tastiera virtuale” con il Blocco Note di Windows e poi fare un copia e incolla, anche se ci sono scorciatoie per alcuni casi.

7.3. *Accenti e virgolette*

Per ottenere le normali lettere accentate, se si è caricato il package `inputenc`, si può usare la tastiera. Altri caratteri si trovano, in WinEdt, nella barra che si apre con il pulsante \ddot{E} . In ogni caso si può mettere qualunque accento su qualunque lettera, facendo precedere la lettera che deve essere sormontata da un accento dal comando `\accento`, dove `accento` è l'accento scelto. Per esempio, per ottenere \hat{A} , basta scrivere `\^A`.

La virgoletta aperta si ottiene mediante il carattere ```, cioè il carattere ASCII di codice 096, (eventualmente ripetuto due volte se si vogliono le virgolette doppie); in WinEdt questo carattere è presente nella già citata barra appena prima della ‘a’ minuscola. La virgoletta chiusa si ottiene con il solito apostrofo (`'`) presente sulla tastiera italiana (anch'esso ripetuto due volte se si vogliono le virgolette doppie). Non si usano le doppie virgolette presenti sulla tastiera italiana, che invece servono a mettere la dieresi sopra qualunque altro carattere: per esempio se si volesse \ddot{c} , basterebbe scrivere `\"c`. In TeXnicCenter per ottenere la doppia virgoletta aperta si può fare un copia e incolla dalla “tastiera virtuale” di cui ho parlato prima, oppure andare nel menu “Tools”, “Options”, scheda “General” e mettere la spunta accanto a “Replace quotation marks”, scegliendo “English” nella casella “Predefined sets”. A questo punto il tasto con le doppie virgolette, mette automaticamente il doppio carattere ```` davanti ad una parola e il doppio carattere `''` alla fine di una parola. (Tip: se volete inserire un solo carattere ``` potete inserirne due col metodo appena descritto e poi cancellarne uno).

Si noti che per scrivere la lettera `i` con gli accenti, bisogna toglierci il puntino, sostituendo la `i` con `\i`, ottenendo `\i`. Si otterrà allora, per esempio, \hat{i} , scrivendo `\^{\i}`, \tilde{i} , scrivendo `\~{\i}`. Analogo discorso per il puntino su `j`.

Anche se a prima vista tutte queste particolarità possono sembrare complicate, col passare del tempo ci si accorge che, invece, si tratta di uno dei punti di forza cruciali del L^AT_EX, uno tra i tanti fatti che lo rendono così potente e flessibile.

7.4. Qualche comando di uso frequente

I comandi L^AT_EX sono centinaia, ma quelli veramente importanti sono pochi e sono costruiti con una sintassi molto ben strutturata, tanto che si imparano senza grandi fatiche. Ne richiamo alcuni, senza alcuna pretesa di essere esaustivo.

- `\emph{testo}` serve a enfaticizzare `testo` (in generale lo rende corsivo, ma se il contesto è corsivo lo rende invece normale: `\emph{Luciano}` produce *Luciano*);
- `\textbf{testo}` serve a rendere `testo` in grassetto: `\textbf{Luciano}` produce **Luciano**;
- `\textsl{testo}` serve a rendere `testo` in forma inclinata (è il corsivo usato per la matematica): `\textsl{Luciano}` produce *Luciano*;
- `\textsc{testo}` serve a rendere `testo` in ‘small caps’: `\textsc{Luciano}` produce LUCIANO; si noti che diventano ‘small caps’ i caratteri scritti in minuscolo;
- `\verb*qualunque cosa*`, produce `qualunque cosa`, esattamente come l’abbiamo digitata da tastiera, compresi gli spazi e ogni altro simbolo, usando un font tipo macchina da scrivere (la coppia di caratteri `*`, che delimitano la scritta `qualunque cosa`, deve essere una coppia di caratteri qualunque, *non* contenuta in `qualunque cosa`; fare attenzione che sul contenuto del comando `\verb`, L^AT_EX non va a capo e, se giunge a fondo riga, non rispetta l’allineamento destro; si tratta pertanto di un comando da usare con moderazione ed è indispensabile controllare i risultati ottenuti ed eventualmente fare qualche aggiustamento manuale;
- `\footnote{testo}` (comando da mettere subito dopo la parola a cui vogliamo collegare una nota a piè di pagina), produce una nota a piè di pagina, separata dal testo da una mezza linea orizzontale e, naturalmente, numerata in modo automatico; il contenuto della nota va scritto, per intero, al posto di `testo`;
- `\ldots` produce i tre puntini di sospensione: ...;

- \cdots produce i tre puntini centrati sulla riga (servono nelle formule matematiche): \cdots ; i due segni di dollaro sono una caratteristica delle formule matematiche, come vedremo nella sezione 10.

8. Liste

L'uso di liste (elenchi puntati o numerati) è una tecnica molto importante nella composizione di testi. A mio modo di vedere l'implementazione di queste tecniche nei word processor WYSIWYG lascia molto a desiderare ed è tutt'altro che facile da usare, tranne nel caso di liste semplici. La creazione di liste con struttura comporta sempre delicati problemi di allineamento ed è molto facile in un documento complesso utilizzare stili diversi nelle varie parti, a scapito della uniformità ed eleganza del documento.

Gli strumenti messi a disposizione da L^AT_EX consentono un controllo semplice e molto efficiente delle liste e, soprattutto, rendono automatico il mantenimento di uno stile uniforme anche in documenti molto lunghi e complessi.

WinEdt prevede, nella voce di menu *Insert*, la possibilità di creare automaticamente il codice per l'inserimento dei tre tipi di liste previsti. TeXnicCenter prevede due pulsanti (identici a quelli di Word) per gli elenchi puntati e quelli numerati.

8.1. Tipi di liste

1. Il primo tipo è costituito dagli elenchi numerati, e naturalmente la numerazione è automatica. La struttura è molto semplice:

```
\begin{enumerate}
  \item Testo del primo item
  \item Testo del secondo item
  \item Testo del terzo item
\end{enumerate}
```

Il testo da inserire può essere di qualunque tipo, comprese altre liste dei tre tipi previsti.

2. Il secondo tipo è costituito dagli elenchi puntati (il *punto-elenco* previsto di default può naturalmente essere modificato) e la struttura è quasi identica a quella degli elenchi numerati:

```
\begin{itemize}
  \item Testo del primo item
  \item Testo del secondo item
```

```
\item Testo del terzo item
\end{itemize}
```

3. Il terzo tipo di liste è quello delle descrizioni (tipo le voci di un dizionario). Ancora una volta la struttura di base rimane la stessa, ma questa volta è previsto un argomento da mettere tra parentesi quadre prima del testo, in sostituzione del numero presente nel primo tipo e del punto-elenco presente nel secondo, proprio come le voci di un dizionario.

```
\begin{description}
\item[Titolo del primo item] Testo del primo item
\item[Titolo del secondo item] Testo del secondo item
\item[Titolo del terzo item] Testo del terzo item
\end{description}
```

8.2. Liste annidate

La cosa interessante di L^AT_EX è però la possibilità di un semplice controllo delle liste annidate. Proponiamo un esempio per chiarire.

Codice necessario:

```
\begin{enumerate}
\item Con \LaTeX si può:
\begin{itemize}
\item Comporre documenti:
\begin{description}
\item[Per la stampa] In una grande varietà di formati
\item[Per lo schermo] Molto simili a Power Point
\end{description}
\item Creare presentazioni
\end{itemize}
\item Con \LaTeX\ non si può:
\begin{itemize}
```

```
\item Creare database tipo:  
\begin{enumerate}  
  \item Microsoft Access  
  \item SQL  
\end{enumerate}  
\item Creare fogli elettronici  
\end{itemize}  
\end{enumerate}
```

Output ottenuto:

1. Con L^AT_EX_{si} può:

- Comporre documenti:

Per la stampa In una grande varietà di formati

Per lo schermo Molto simili a Power Point

- Creare presentazioni

2. Con L^AT_EX non si può:

- Creare database tipo:
 - (a) Microsoft Access
 - (b) SQL
- Creare fogli elettronici

9. Immagini e tabelle

L'inserimento di immagini e tabelle in un documento è una delle opzioni più richieste: fin dal primo giorno di lezione su L^AT_EX gli allievi si sono preoccupati di questo aspetto, in particolare delle immagini.

In L^AT_EX ci sono alcune particolarità che, a prima vista possono sembrare complicazioni e che, invece, sono proprio legate al fatto che L^AT_EX è un software per videoscrittura professionale e non per giocare con il mouse.

Anche in questa sezione, come in tutto il resto di questi appunti, fornisco solo alcune semplici indicazioni di base, rimandando a letture specializzate per realizzazioni più complesse.

9.1. Il formato delle immagini

Non tutti i formati immagine possono essere inseriti in un documento L^AT_EX perchè L^AT_EX pretende di produrre documenti di elevata qualità e quindi non gli si può dare in pasto qualunque cosa; è un po' come le mucche dal cui latte si produce il *Parmigiano Reggiano*: poichè quel formaggio è di elevatissima qualità, esse possono mangiare solo foraggi di prima scelta.

Per un documento da compilare con L^AT_EX e visualizzare con Yap o tradurre in postscript il formato ammesso è solo `.eps`, ovvero `ENCAPSULATEDPOSTSCRIPT`. Per un documento da compilare con PDFL^AT_EX e visualizzare con Acrobat, i formati ammessi sono `.pdf`, `.jpg`, `.png`.

Segnalo in particolare che un'immagine `.eps` è in realtà un file di testo (potete provare ad aprirlo con un comune text editor, come WinEdt) e quindi può essere agevolmente trattato da un software come L^AT_EX che è un text-editor. La possibilità di eseguire in maniera automatica e con risultati di alta qualità rotazioni, cambiamenti di scala, ecc., è uno dei motivi che sono alla base delle limitazioni ai tipi di file inseribili in un documento L^AT_EX.

Di norma quando si lavora ad un documento si usa sia la compilazione L^AT_EX che quella PDFL^AT_EX. È dunque conveniente produrre due diversi formati per ogni immagine da inserire: `.eps` e `.pdf` (o `.jpg` o ancora `.png`).

La maggior parte dei software grafici di qualità possono salvare direttamente le immagini prodotte sia in formato `.eps` che in uno dei formati per il `.pdf`. Se avete però proprio bisogno di inserire immagini prodotte con altri software, è possibile convertirle in uno dei formati citati: il risultato non è sempre ottimale, ma in molti casi è accettabile. Uso normalmente questo sistema

per le immagini prodotte con Cabri, che copio in formato `.bmp` in un programma di grafica, salvo in formato `.jpg` e successivamente converto in formato `.eps`.

9.2. Programmi di conversione

Tra i numerosi sistemi possibili per la conversione ho optato per ImageMagick, che ha il vantaggio di essere free. Il programma può essere reperito sul sito ufficiale <http://www.imagemagick.org>: nel menu di sinistra, alla voce *Binary Releases*, scegliete Windows e, nella finestra che si aprirà scaricate la versione “Dynamic at 16 bits-per-pixel”.

Installate normalmente, prestando attenzione che la cartella di installazione non contenga spazi, nemmeno nel percorso (una sottocartella di `C:\Programmi` va bene).

Il modo più semplice per usare questo programma è dal prompt di DOS. Mettete il file da convertire (che supponiamo si chiami `nomefile.jpg` in una cartella (sempre senza spazi nel nome), supponiamo `C:\img`. Aprite il prompt dei comandi di DOS e cambiate la `dir` di lavoro (digitando `cd img`). A questo punto dovreste trovarvi in `C:\img` (come potete vedere dal prompt) e basterà scrivere `convert nomefile.jpg nomefile.eps`, premendo Invio. Potete usare lo stesso programma per moltissime conversioni di file grafici da un formato all'altro.

Visto che avete installato ImageMagick, segnalo che esso può anche visualizzare (tramite ImageMagick Display che viene inserito nel menu Start), immagini di decine di formati diversi.

9.3. L'inserimento vero e proprio

Una volta che avete la vostra immagine in *due* formati diversi piazzateli entrambi nella cartella dove avete salvato il vostro documento L^AT_EX.

Nel preambolo del documento dovete inserire il comando

```
\usepackage{graphicx}
```

Ci sono due possibilità di base per inserire le immagini:

- semplici immagini senza didascalia e non numerate;
- immagini dotate di didascalia e numerate (automaticamente!).

Nel primo caso è sufficiente il comando `\includegraphics[opzioni]{nome}`. Al posto di *opzioni* potete mettere una o più delle seguenti (o altre di cui non parlo qui), separate da una virgola e senza spazi tra l'una e l'altra:

- `width=5cm` (o le dimensioni che volete voi: l'immagine sarà automaticamente ridimensionata, con buoni risultati qualitativi, se l'immagine di partenza era ben realizzata);
- `height=5cm` (o altre dimensioni).

Al posto di *nome* dovete mettere il nome del file (comprensivo della sottocartella, se l'avete posto in una sottocartella di quella dove si trova il documento), senza l'estensione: se compilate in L^AT_EX sarà caricata la versione `.eps`, se compilate in PDFL^AT_EX sarà caricata l'altra versione.

Nel secondo caso occorre racchiudere il comando in una struttura del tipo:

```
\begin{figure}
\caption{La caption scelta}
\begin{center}
\includegraphics[opzioni]{nome}
\end{center}
\end{figure}
```

Attenzione perché L^AT_EX inserirà l'immagine nel posto che riterrà più opportuno dal punto di vista estetico (anche se ovviamente ci sono possibilità di forzare l'inserimento nel posto che ci piace, ma, come al solito, sconsiglio di forzare L^AT_EX a fare cose che piacciono a noi, invece di quelle che piacciono a lui, almeno fin quando non diventeremo esperti tipografi ...). Non intendo entrare qui in ulteriori dettagli: nei documenti di base quanto finora detto è ampiamente sufficiente.

9.4. Testo e immagini affiancate

Ci sono vari metodi e pacchetti per ottenere l'effetto di immagine contornata dal testo. Riporto qui un esempio semplicissimo, che non richiede alcun pacchetto aggiuntivo e che può andare bene per il 90% (o forse il 99%) dei casi comuni. Ad ogni modo, come al solito, *non* cercate effetti speciali: un buon documento non deve gridare al lettore “guardami” (vedi la Sezione 13, Conclusione, di questa guida).

Supponiamo che abbiate un'immagine, dal nome `immagine.eps` o `immagine.pdf` nella cartella dove si trova il documento che state componendo. Dovete scrivere il seguente codice:

```
\begin{center} (oppure \begin{flushleft}, oppure \begin{flushright})
\begin{minipage}[c]{.40\textwidth}
Qualunque testo volete mettere
\end{minipage}
\begin{minipage}[c]{.40\textwidth}
\includegraphics[opzioni]{immagine}
\end{minipage}
\end{center} (oppure \end{flushleft}, oppure \end{flushright})
```

La `[c]` sta per *centrato*, e può essere sostituita da `[t]` o `[b]`, per *top* o *bottom*. La parte `.40\textwidth` indica che questa ‘minipagina’ ha dimensioni del 40% dell’ampiezza del testo. Per capire come funziona e scoprire altre opzioni basta solo provare a giocare, cambiando qualche cosa nel codice citato (e successivamente andando a leggere manuali più completi ...).

9.5. Tabelle

L’inserimento di tabelle in L^AT_EX non è semplicissimo, a meno che non ci si voglia limitare a casi elementari.

Cominciamo col proporre un esempio di base, che produce una tabella centrata sulla riga, con tre colonne e con allineamento centrato nelle tre colonne.

```
\begin{center}
\begin{tabular}{ccc}
cella 1.1 & cella 1.2 & cella 1.3\\
cella 2.1 & cella 2.2 & cella 2.3\\
cella 3.1 & cella 3.2 & cella 3.3\\
cella 4.1 & cella 4.2 & cella 4.3\\
\end{tabular}
\end{center}
```

L’istruzione `ccc` tra parentesi graffe indica che le colonne sono centrate (ciascun ‘c’ può essere sostituito da ‘l’ o ‘r’, con ovvio significato), Il simbolo `&` separa le colonne, il simbolo `\\` termina una riga.

L'output del codice soprastante è il seguente:

cella 1.1	cella 1.2	cella 1.3
cella 2.1	cella 2.2	cella 2.3
cella 3.1	cella 3.2	cella 3.3
cella 4.1	cella 4.2	cella 4.3

Esistono degli help che facilitano la costruzione di tabelle (anzi la rendono quasi automatica) Citiamo `array` e `tablas`. Informazioni nella pagina <http://www.winedt.org>, seguendo la voce `Plug-ins` e `Add-ons`.

Un esempio di tabella prodotta con `tablas` (richiede il package `colortbl`).

	1	2	3
a	Rosso	Giallo	Verde
b	Blu	Arancio	Bianco

Ovviamente il codice, che riportiamo per completezza qui di seguito, potrebbe anche essere inserito manualmente, con parecchie semplificazioni.

```

\begin{center}
\begin{tabular}{c|p{1.63cm}|p{1.63cm}|p{1.63cm}}
  & \multicolumn{1}{p{1.63cm}}{\centering 1} &
  & \multicolumn{1}{p{1.63cm}}{\centering 2} &
  & \multicolumn{1}{p{1.63cm}}{\centering 3} \\ \hline
a & \multicolumn{1}{>{\columncolor{red}}p{1.63cm}}{\centering Rosso} &
  & \multicolumn{1}{>{\columncolor{yellow}}p{1.63cm}}{\centering Giallo} &
  & \multicolumn{1}{>{\columncolor[rgb]{0,1,0}}p{1.63cm}}{\centering Verde} \\
b & \multicolumn{1}{>{\columncolor{blue}}p{1.63cm}}{\centering Blu} &
  & \multicolumn{1}{>{\columncolor[rgb]{1,0.5,0.25}}p{1.63cm}}{\centering Arancio} &
  & \multicolumn{1}{p{1.63cm}}{\centering Bianco} \\
\end{tabular}
\end{center}

```

10. Qualche cenno alle formule matematiche

Potrà sembrare strano che abbia rimandato di parlare di formule matematiche fin quasi alla fine del mio piccolo manuale, in quanto in realtà L^AT_EX è nato principalmente per problemi connessi alla resa tipografica di formule.

Il fatto è che i matematici sono pochi (per fortuna ...!) e forse non varrebbe la pena, in una scuola media superiore, dedicare tempo ad un software, se esso servisse solo per chi in futuro dovrà occuparsi di scrivere testi pieni di formule astruse e complicate. Io credo invece che L^AT_EX sia assolutamente il top di quanto offre il mercato in termini di videoscrittura, anche, e forse soprattutto, per chi deve scrivere normalissimi libri contenenti solo puro e semplice testo. Il codice sorgente di un file L^AT_EX contenente parecchia matematica è infatti leggibile con tranquillità solo dagli addetti al settore, in quanto le istruzioni di formattazione possono occupare anche più del 50% del totale. Per chi deve occuparsi di redigere invece articoli contenenti quasi solo testo, L^AT_EX offre la grande opportunità di un ambiente tipo *macchina da scrivere* che non distoglie l'attenzione dal contenuto e non chiede nessuno spreco di tempo a giocare con il mouse o con la scelta di strani (e inutili) font che servono solo ad affaticare la vista del lettore.

10.1. Regole di base per le formule

Ci occupiamo qui solo di formule inserite in linea con il testo o su righe separate, ma senza la numerazione tipica dei testi di matematica (cosa che comunque si potrebbe realizzare in maniera automatica –come sempre in L^AT_EX– e molto semplice).

Innanzitutto occorre dichiarare nel preambolo l'uso di opportuni package. Consiglio di usare quelli della American Mathematical Society, con le seguenti istruzioni:

```
\usepackage{latexsym}  
\usepackage{amsmath,amssymb,amsfonts}
```

Successivamente, per inserire una formula occorre racchiuderla tra due delimitatori e precisamente due segni di dollaro, $\$formula\$, oppure i segni $\backslash[e \backslash]$, cioè $\backslash[formula\backslash]$: nel primo caso la formula è in linea con il testo, nel secondo caso è su una riga separata e di norma centrata nella riga. La distinzione è molto importante perchè L^AT_EX usa due tecniche diverse per scrivere le formule: le formule in linea sono più compatte, per evitare sgradevoli distanziature tra le righe; le$

formule centrate sono più grandi, per una lettura ottimale e perchè di solito esse devono catturare l'attenzione del lettore. Si può vedere la differenza negli esempi che seguono.

Formula in Linea $\sum_1^{+\infty} (-1)^{n+1} \frac{1}{n} = \ln(2)$

Formula centrata

$$\sum_1^{+\infty} (-1)^{n+1} \frac{1}{n} = \ln(2)$$

Un'altra particolarità delle formule matematiche è che viene usato automaticamente uno speciale corsivo (testo inclinato o 'slanted text') per scrivere le variabili o le lettere, mentre viene usato il carattere normale per i numeri. Anche qui si può vedere un esempio:

$$\sqrt{3}x^3 + \frac{1}{3}x^2 - 2(k+1)x + 3k = 0$$

Si noti che anche MathType fa automaticamente così. La cosa diventerebbe complicata se volessi scrivere invece una formula che non richieda MathType in un programma di videoscrittura. Provate a scrivere, per esempio, $x^3 + 2x^2 - 2(k+1)x + 3k = 0$, direttamente in un programma WYSIWYG e controllate se alla fine vi è venuta o no la 'sindrome carpale' a forza di cliccare sui vari bottoni con il mouse!

Molti "Latexisti" (forse quasi tutti) usano un doppio segno di dollaro sia iniziale che finale, al posto di \lbrack e \rbrack . Si tratta di un tipo di notazione che non dovrebbe essere usato perchè ha dei problemi con le spaziature. Se però non siete dei puristi la cosa non è gravissima, ma in L^AT_EX è meglio essere puristi!

Anche la scrittura delle lettere greche richiede l'uso dei due segni di dollaro: $\$ \alpha \$$ produce α (non preoccupatevi del codice $\backslash alpha$, sia WinEdt che TeXnicCenter lo inseriscono automaticamente con le apposite barre).

10.2. Scrittura esplicita delle formule

Proponiamo solo alcuni comandi base, per gli altri è disponibile un software (gratuito!), praticamente identico a MathType, che si chiama TexAide e si può reperire all'indirizzo del fornitore (<http://www.dessci.com/en/products/texaide/>). Chi ha MathType può usarlo direttamente, andando

alla voce *Translators* del menu, scegliendo *Translation to other language (text)* e poi *Tex-AMS-LaTeX*: con il solito copia e incolla si otterrà direttamente nell'editor il codice per la formula scritta visualmente in MathType. A proposito: togliete le spunte dalle due caselle in basso in questa finestra di MathType!

E ora qualche comando:

Frazioni $\frac{\text{numeratore}}{\text{denominatore}}$: $\frac{\text{numeratore}}{\text{denominatore}}$;

Radici quadrate $\sqrt{\text{radicando}}$: $\sqrt{\text{radicando}}$;

Radici n-esime $\sqrt[n]{\text{radicando}}$: $\sqrt[n]{\text{radicando}}$;

Vettori \vec{v} : \vec{v} ;

Frecce sopra \overrightarrow{AB} : \overrightarrow{AB} ;

Apici e pedici $x^3+x_2+x^3_2$: $x^2 + x_2 + x_2^3$;

...

Un esempio per concludere (usando solo i comandi indicati):

$$\frac{\frac{\sqrt[4]{2} + \sqrt[3]{\frac{1}{2}}}{5}}{\sqrt{2} + \frac{1}{\sqrt{2+\sqrt{2}}} + \sqrt{2 + \sqrt{2 + \sqrt{2}}}}$$

$$\frac{\frac{\sqrt[4]{2} + \sqrt[3]{\frac{1}{2}}}{5}}{\sqrt{2} + \frac{1}{\sqrt{2+\sqrt{2}}} + \sqrt{2 + \sqrt{2 + \sqrt{2}}}}$$

A sinistra abbiamo messo la formula L^AT_EX, a destra la stessa formula scritta in MathType e inserita in un programma tipo Word. Si tenga conto che abbiamo proprio usato MathType per creare il codice L^AT_EX: ma quale differenza di risultato!

11. Tips

In questa sezione sono raccolti alcuni suggerimenti, principalmente derivati da discussioni con gli allievi, in relazione a problemi sorti nell'uso di L^AT_EX, oppure qualche breve istruzione che non ha trovato posto negli altri capitoli.

- Poichè la compilazione di un file L^AT_EX produce parecchi file secondari, conviene inserire ciascun nuovo documento in una apposita cartella, dove saranno anche salvati i file secondari, per non fare confusione. Inoltre, ma questo vale in generale quando si lavora al computer, quando si riapre un file per modificarlo, conviene, prima di procedere, salvarlo con un nuovo nome (io uso sempre lo stesso nome seguito dalla data in formato compresso `-nome.ggmm`) naturalmente senza inserire spazi. In questo modo sarà facile recuperare vecchie versioni in caso di problemi.
- In alcuni rari casi L^AT_EX può dare problemi alla compilazione se sono presenti nella cartella vecchie versioni di alcuni dei file di compilazione: è opportuno, prima di disperarsi, provare a cancellarli. WinEdt fornisce un aiuto per questo nel menu *Tools* alla voce *Erase Output Files*.
- In documenti molto complessi la lettura del codice può diventare complessa, a causa della presenza di un gran numero di comandi che frammentano il testo. È opportuno usare la possibilità offerta da L^AT_EX, come da quasi tutti i linguaggi di programmazione, di inserire commenti, che vengono completamente ignorati dal compilatore. In L^AT_EX i commenti si inseriscono facendoli precedere dal segno `%`: tutto quello che va da un simile segno a fine riga non viene letto da L^AT_EX. Conviene usare questa opportunità, per esempio, per separare con un commento le varie sezioni, oppure per chiarire qual è il livello di annidamento di una lista strutturata, o per ogni altra esigenza.

11.1. Documenti lunghi

In caso di documenti molto lunghi non è opportuno tenere tutto il testo su un unico file: conviene piuttosto spezzarlo in varie parti, creando quello che si chiama un 'main file', con vari allegati; è molto più facile, in questo modo, la ricerca e modifica.

I comandi da usare sono:

- `\input{nomefile.tex}`: in questo caso il contenuto del file `nomefile.tex` viene aggiunto al file principale, come se fosse scritto di seguito;
- `\include{nomefile.tex}`: in questo caso si ha l'effetto di `\input`, con un più un salto pagina prima e dopo il file inserito. Questa opzione conviene usarla quando si ha una divisione in capitoli, dove l'interruzione di pagina è ovviamente richiesta.

Un esempio:

```
\documentclass[opzioni]{tipo_di_classe}
preambolo
\begin{document}
Inizio del documento (titolo, ecc.)
\include{premessa.tex}
\include{capitolo1.tex}
...
\include{capitolo10.tex}
\end{document}
```

Anche WinEdt può venirci incontro in questa gestione. Dal file principale cliccare sul bottone *Set Main File* (il terzo della seconda fila) e poi sul bottone *Project Tree* (il sesto della seconda fila); nella finestra che compare a sinistra cliccare sul primo bottone *Build Tree* e ci comparirà l'elenco di tutti i file collegati al file principale. Questo elenco è cliccabile, nel senso che ci consente di passare rapidamente da un file all'altro. In questo *Tree* compaiono anche la *Table of Contents* e altre cose che si impareranno ad usare con l'esperienza. Tutto questo può essere salvato (e poi recuperato) con un apposito nome usando il menu *Project* e le voci *Save Project*, *Open Project*, *Close Project*. Conviene sperimentare un po' per capire il funzionamento.

Molto simile la situazione con TeXnicCenter. Posizionatevi nel file che deve essere il "Main File", dal menu *Project* scegliete la voce *Create with active file as main file*, scegliete la voce "it" nel *Language* e nel *Dialect* e tutto è fatto: nella finestra a sinistra comparirà l'elenco dei file inseriti man mano nel progetto e che compongono il nostro documento. Le voci *Save Project*, *Open Project*, *Close Project*, si trovano le menu *File*. Come sempre sperimentando un po' si capisce il funzionamento. Per compilare usate il pulsante *Build and view output*.

11.2. Classi di documenti

La prima istruzione di un documento L^AT_EX è `\documentclass`. Noi abbiamo sempre usato, finora, l'opzione `article`. È venuto il momento di chiederci se è l'unica possibilità (ovviamente no, altrimenti non sarebbe stata prevista per naturale economia di pensiero) e a che cosa serve.

Le classi di documento influenzano alcune caratteristiche dei documenti stessi, come la possibilità di suddivisione in parti, alcuni comandi disponibili in una classe e non in un'altra, e così via.

Le classi base che ci interessano sono `article` (quella che abbiamo sempre citato finora) e `report`; la prima serve per piccoli documenti divisi in sezioni, ma non in capitoli, la seconda per documenti più complessi.

Sono previste altre classi standard: `book` per veri e propri libri divisi in capitoli e addirittura in tomi, `slides` per presentazioni a video, `letter` per scrivere lettere. Oltre a queste sono previste altre classi che si possono utilizzare caricando appositi package. Cito, per un esempio eclatante, la class `prospcr` che consente di creare diapositive sofisticate come quelle di PowerPoint, ma con in più tutta la potenza di L^AT_EX.

Non intendo insistere oltre su questo argomento, pur importante: la stragrande maggioranza dei documenti per l'utente base può tranquillamente essere fatta con la classe `article` o al massimo `report`. L'unica cosa degna di nota è la seguente: se usate la classe `report` e non volete usare i capitoli, inserite, nel preambolo, il seguente comando:

```
\renewcommand{\thesection}{\arabic{section}}
```

Serve a impedire che le sezioni vengano pensate come appartenenti al capitolo 0, cosa del tutto poco simpatica.

12. Link per ulteriori letture

Come più volte ripetuto, questa *non* è una guida a L^AT_EX ma solo una raccolta di suggerimenti e di informazioni utili, per entrare senza troppe difficoltà nell'affascinante mondo della videscrittura professionale. Per chi vuole proseguire l'avventura, propongo qui una serie di link a materiale disponibile in rete, segnalando che si tratta solo di una scelta di carattere assolutamente personale tra la sterminata letteratura pubblicata su L^AT_EX. Alcuni tra i materiali segnalati vengono anche automaticamente inseriti nella cartella `C:\texmf`, se si installa la versione completa di MiKTeX. Segnaliamo inoltre che queste guide sono reperibili presso molti server diversi.

- *Una (mica tanto) breve introduzione a L^AT_EX₂ε*, di Tobias Oetiker, Herbert Partl, Irene Hyna e Elisabeth Schlegl: una delle guide introduttive a L^AT_EX più famose, in <http://www.dimi.uniud.it/~gorni/TeX/itTeXdoc/itlshort.pdf>;
- *Impara L^AT_EX (... e mettilo da parte!)*, di Marc Baudoin: guida ottima e abbastanza completa, in http://www.dimi.uniud.it/~gorni/TeX/itTeXdoc/impara_latex.pdf;
- *Introduzione al L^AT_EX*, di Gianluca Gorni: una guida sotto forma di slide, anche come guida in linea, in <http://www.dimi.uniud.it/~gorni/TeX/itTeXdoc/CorsoTeX2003stampa.pdf>. Nel sito di Gianluca Gorni potete trovare molti altri link relativi a L^AT_EX nonché, per esempio, un template per una tesi di laurea, che i più volenterosi possono studiare anche per imparare a costruire realizzazioni personali complesse con L^AT_EX;
- *Formatting Information*, di Peter Flynn: la guida che io utilizzo come manuale di consultazione continua, completa e comunque adatta anche ai principianti, in <http://www.silmaril.ie/downloads/>;
- *L^AT_EX for Complete Novices*, di Nicola Talbot: una guida introduttiva semplice, con numerose informazioni utili per chi inizia in <http://theoval.cmp.uea.ac.uk/~nlct/latex/novices/novices.html>
- *A comprehensive L^AT_EX symbol list*, di Scott Pakin: una specie di manuale dove reperire praticamente tutti i simboli che servono per il L^AT_EX di uso scientifico, in <http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>;

- *Gestione di figure e tabelle con L^AT_EX*, di Luca Caucci e Mariano Spadaccini: il titolo dice già tutto (ottimo), in http://www.guit.sssup.it/downloads/fig_tut.pdf.

Ripetiamo anche l'indirizzo già citato del Gruppo Italiano Utilizzatori di T_EX (GUIT): <http://www.guit.sssup.it/>, dove potrete trovare tutte le notizie che vi serviranno. Fortemente consigliata, per chi ha voglia di proseguire nello studio di L^AT_EX, l'iscrizione al forum attivo su questo sito.

13. Conclusione

Come conclusione di questo breve manualetto mi pare utile riportare quanto scrive Peter Wilson nel suo manuale d'uso di *The Memoir Class for Configurable Typesetting*:

The essence of good typography is that it is not noticeable at first, or even second or later, glances to any without a trained eye. If your initial reaction when glancing through a book is to exclaim about its layout then it is most probably badly designed, if it was designed at all. Good typography is subtle, not strident.

With the advent of desktop publishing many authors are tempted to design their own books. It is seemingly all too easy to do. Just pick a few of the thousands of fonts that are available, use this one for headings, that one for the main text, another one for captions, decide how big the typeblock is to be, and there you are.

However, just as writing is a skill that has to be learned, typography is also an art that has to be learned and practised. There are hundreds of years of experience embodied in the good design of a book. These are not to be cast aside lightly and many authors who design their own books do not know what some of the hard-earned lessons are, let alone that what they are doing may be the very antithesis of these. An expert can break the rules, but then he is aware that he has good reasons for breaking them.

... If a book shouts 'look at me!' then it is an advertisement, and a bad one at that, for the designer.

L'essenza di un libro ben stampato è che non si fa notare al primo, o addirittura al secondo o successivo, sguardo di chiunque non abbia un occhio allenato. Se la vostra prima reazione nello sfogliare un libro è di fare un'esclamazione di meraviglia osservando il layout, allora il libro è molto probabilmente mal progettato, se mai è stato progettato. La stampa di qualità è raffinata, non stridente.

Con l'avvento del desktop publishing molti autori hanno la tentazione di progettare da soli i loro testi. Sembra molto facile farlo. Basta scegliere alcune delle migliaia di font disponibili, usarne uno per i titoli, uno per il testo principale, un altro per le didascalie, decidere le dimensioni dei caratteri, e la cosa è fatta.

Tuttavia, come scrivere è un'abilità che bisogna apprendere, anche comporre tipograficamente un testo è un'arte che si deve apprendere e su cui bisogna esercitarsi. Ci sono centinaia di anni di esperienza racchiusi nel buon design di un libro. Essi non possono essere trascurati con leggerezza e

L. Battaia

L^AT_EX,
naturalmente

Indice



Pag.  di 

molti autori che progettano i loro libri non conoscono alcune delle conquiste più importanti, per non parlare del fatto che quello che fanno è esattamente in antitesi con esse. Un esperto può infrangere le regole, ma allora sa che ha delle buone ragioni per farlo.

... Se un libro grida 'guardami', questo è un avviso, e un pessimo avviso, per chi l'ha progettato.