

# TEX

Roland Heynkes

10. Januar 2005, Aachen

Manchmal ist eben nicht das Bessere, sondern das Schlechtere der Feind des Guten. Das gute alte TEX ist dem sehr viel verbreiteteren MS-Word in fast allen Belangen klar überlegen, wird aber durch die reine Marktmacht von Microsoft in die Nischen der Anwendungsbereiche gedrängt, in denen Word total versagt. Wer TEX überhaupt kennt, fürchtet wohl oft, die Arbeit damit sei zu kompliziert und unbequem. Das stimmt auch, solange man nur einfache Briefe und andere höchstens inhaltlich anspruchsvolle Texte verfasst. Aber je komplexer die Werke im formalen Sinne werden und je besser lesbar sie sein sollen, umso mehr gewinnt das nebenbei auch noch kostenlose TEX gegenüber dem nicht gerade billigen MS-Word.

## Inhaltsverzeichnis

<b>1</b>	<b>Wozu braucht man eigentlich TEX/L<sup>A</sup>TEX?</b>	<b>1</b>
1.1	Textverarbeitungsprogramme überfordern Autoren . . . . .	1
1.2	Hilfe und Bevormundung kommen sich oft zu nahe . . . . .	3
1.3	TEX ist extrem stabil und läuft auf jedem Rechner . . . . .	4
1.4	Außerdem kann TEX einfach mehr . . . . .	5
1.5	TEX, L <sup>A</sup> TEX, Zusatzmodule und sogar Hilfe sind kostenlos . . . . .	6
<b>2</b>	<b>Wie kommt man zu einem TEX/L<sup>A</sup>TEX-System unter Windows?</b>	<b>6</b>
	<b>Quellenverzeichnis</b>	<b>7</b>
	<b>Index</b>	<b>8</b>

# 1 Wozu braucht man eigentlich T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X?

## 1.1 Textverarbeitungsprogramme überfordern Autoren

Bei der Publikation eines wissenschaftlichen oder literarischen Werkes sind prinzipiell vier Arbeitsschritte zu unterscheiden, die jeweils umfangreiches Wissen und Erfahrung erfordern, wenn der Text lesenswert und lesbar werden soll.

Zunächst einmal bedarf es natürlich eines fachkundigen oder zumindest inspirierten Autors, der seine Gedanken sowie vielleicht auch gesammelte Fakten sinnvoll und verständlich zu formulieren versteht. Das klassische Ergebnis solcher Autorenarbeit war früher ein mit einer Schreibmaschine oder gar handgeschriebenenes Manuskript, welches der Autor einem Verlag seines Vertrauens übergab. Der Vorteil dieser Arbeitsteilung bestand darin, daß Autoren sich auf das konzentrieren konnten, wovon sie - hoffentlich - etwas verstanden.

Gute Verlage ließen jedes Manuskript zunächst von mindestens einem Lektor einer kritischen Prüfung im Hinblick auf inhaltliche Relevanz und Korrektheit, sprachliche Qualität und Verkaufbarkeit unterziehen sowie nebenbei auch noch orthographische und grammatikalische Fehler korrigieren. Die meisten Werke profitierten vom oft zähen Ringen zwischen Autor und Lektor um jede Formulierung und unzählige mangelhafte oder überflüssige Werke blieben dem Publikum erspart. Heute wird unsere geistige Umwelt massenhaft mit Belanglosigkeiten und Unwahrheiten in beklagenswerter sprachlicher Qualität verschmutzt, weil man den unterschätzten Beruf des hochqualifizierten Lektors durch vergleichsweise primitive Korrekturprogramme ersetzen zu können glaubt.

Nach bestandener inhaltlicher Qualitätskontrolle wurden überarbeitete Manuskripte an Setzer übergeben, die aufgrund einer langen Ausbildung genau wussten, wie man Buchstaben, Worte, Sätze, Absätze, Seiten und ganze Texte, Tabellen und Graphiken nach bestimmten Regeln so auf zu bedruckenden Flächen verteilt, gliedert, plaziert und darstellt, daß das Interesse des Betrachters geweckt und erhalten, sein Lesen und Verstehen sinnvoll gelenkt sowie seine Augen geschont werden. Natürlich mussten sich die Setzer zusätzlich nach den Vorgaben von Designern richten, die oft verlagstypisch das Layout eines Buches oder einer Zeitschrift entwerfen.

Auch der Beruf des Setzers wurde offenbar massiv unterschätzt, denn heute versuchen sich insbesondere im Internet die meisten Autoren unbekümmert und mit entsprechend mäßigem Erfolg selbst an der Formatierung ihrer Werke, während der Setzer weitgehend ausgestorben ist. Nur weil man es rein technisch kann oder über entsprechende Programme verfügt, traktiert man häufig den Betrachter mit einer sinnlos verwirrenden Panscherei verschiedenster Schrifttypen, -größen sowie -auszeichnungen. Zahllose Autoren quälen die Augen ihrer Leser mit geschmacklosen oder aufgrund ungenügender Kontraste kaum brauchbaren Farbkombinationen. Dem künstlerischen Ausdrucksdrang mancher Autoren mag dieser normal gewordene Dilettantismus entgegenkommen, ihren Werken verhelfen aber die von keinerlei Sachkenntnis gelenkten Bemühungen der heute meistens selbst setzenden Autoren selten zu befriedigender Lesbarkeit.

Wenn in den in dieser Hinsicht tatsächlich besseren alten Zeiten Designer und Setzer fertig waren mit einem Werk, dann kam es in die Druckerei, wo man noch wusste,

wie Texte und Bilder in hoher Qualität und Auflage zu Papier gebracht werden. Auch dieses Handwerk erfordert umfangreiches Wissen, über welches Autoren im allgemeinen nicht verfügen. Allerdings benötigen Autoren heute auch weder Druckereien, noch die Vertriebswege klassischer Verlage, wenn sie ihre Werke im Internet publizieren. Man muß dazu nur noch wissen, wie und mit welchen Programmen man seine Werke in geeigneten Formaten abspeichert und auf einen Internet-Server überträgt.

Hauptverantwortlich für die sich gerade in Zeiten der Reiz- und Informationsüberflutung bitter rächende Einsparung von Lektoren und Setzern sind natürlich durch Unwissenheit enthemmte Rationalisierer vornehmlich in Chefetagen. Nicht ganz unwesentlich zur Misere beigetragen haben aber auch Textverarbeitungsprogramme wie MS-Word, welche Autoren Formatierungsmöglichkeiten an die Hand geben, von denen diese besser die Finger ließen, weil sie wenigstens in dieser Hinsicht selten wissen, was sie tun. Das ist so ein Fall, wo die Weisheit in der Selbstbeschränkung läge. Leider ist es Autoren mit Textverarbeitungsprogrammen aber auch gar nicht möglich, sich auf Inhalt und logische Gliederung ihrer Werke zu konzentrieren. MS-Word<sup>1</sup>, OpenOffice-Writer<sup>2</sup> und Consorten nehmen nämlich dem Autor die Layout- und setztechnische Arbeit nicht ab, weil ihnen niemand das dafür notwendige Wissen einprogrammiert hat.

Ganz anders ist dies bei Textsatzsystemen wie  $\text{T}_{\text{E}}\text{X}$ <sup>3</sup> (gesprochen Tech), in welches sein Erfinder *Donald Knuth*<sup>4</sup> in zehnjähriger Arbeit bis zur ersten stabilen Version im Jahre 1983 das über Jahrhunderte gewachsene Wissen des Setzerberufes einfließen ließ [1]. Mit  $\text{T}_{\text{E}}\text{X}$  (vom griechischen Wort für Kunst und Technologie  $\tau\epsilon\chi\nu\eta$ ) kann man Briefe, Artikel oder Bücher nach allen Regeln der Kunst setzen, sofern man  $\text{T}_{\text{E}}\text{X}$  beherrscht [1]. Genau wie viele andere mächtige Werkzeuge, lässt sich aber auch  $\text{T}_{\text{E}}\text{X}$  mit seinen rund 900 Befehlen (davon ein Drittel primäre und zwei Drittel daraus abgeleitete sogenannte Makros) [1] nur mit Wissen und Erfahrung wirklich beherrschen und sinnvoll nutzen, über die normale Autoren kaum verfügen.

$\text{T}_{\text{E}}\text{X}$  ist aber nicht nur ein Formatierungsprogramm, sondern gleichzeitig eine Makrosprache, mit der sich auf einer höheren Abstraktionsebene mächtige Funktionssammlungen programmieren lassen, welche auch Formatpakete genannt werden und funktionell ungefähr den Formatvorlagen von MS-Word entsprechen [3]. So konnte *Leslie Lamport*<sup>5</sup> mit  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ <sup>6</sup> eine mächtige Sammlung von Makros schaffen, welche eine überschaubare Anzahl verständlicher Angaben zum Typ (Dokumentklasse), zur Sprache und zur Struktur eines Textes in korrekte  $\text{T}_{\text{E}}\text{X}$ -Befehle übersetzen [1]. Das bis zur Version 2.09 von *Leslie Lamport* weiterentwickelte und seit der Version  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\epsilon$  ohne ihn vom  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  Project Team<sup>7</sup> betreute Formatpaket  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  bietet Formatierungsbeschreibungen für typische Dokumentkategorien (Artikel, Report, Buch) sowie einen Basissatz an Formatierungsmöglichkeiten und ist immer noch die populärste Benutzerschnittstelle für  $\text{T}_{\text{E}}\text{X}$  [3].

---

<sup>1</sup><http://office.microsoft.com/de-de/FX010857991031.aspx>

<sup>2</sup><http://de.openoffice.org/nonav/doc/einfuehrungen/writer/start.htm>

<sup>3</sup><http://www.ctan.org/>

<sup>4</sup><http://www-cs-faculty.stanford.edu/~knuth/>

<sup>5</sup><http://research.microsoft.com/users/lamport/>

<sup>6</sup><http://www.latex-project.org/>

<sup>7</sup><http://www.latex-project.org/latex3.html>

So müssen sich seit der Freigabe von  $\LaTeX$  Mitte 1985 auch am Computer schreibende Autoren um die richtige Formatierung und Gestaltung des Textes im Grunde gar nicht kümmern, sondern in  $\LaTeX$  lediglich angeben, ob es sich bei einem Textabschnitt um eine Überschrift, eine Tabelle, eine mathematische Formel oder einen normalen Abschnitt handelt [1, 2, 3]. Beim Compilieren eines Textes erzeugt dann  $\LaTeX$  das Layout und  $\TeX$  setzt den Text [1] in einer Qualität, wie man sie mit Textverarbeitungsprogrammen kaum erreichen kann [2, 3].  $\LaTeX$  stellt viele Standardlayouts zur Verfügung, erlaubt aber auch die Gestaltung eigener Layouts [1, 2, 3], so wie es generell aufgrund der offenen Quellcodes an  $\TeX$  und  $\LaTeX$  nichts gibt, was Experten nicht ändern könnten [2, 3]. Besonders einfache Anweisungen für im gesamten Dokument wirksame Layoutentscheidungen sorgen für eine Harmonisierung des Erscheinungsbildes [1].

Gerade für nicht US-amerikanische Benutzer eignen sich die im  $\LaTeX$ -Kern definierten Standardklassen aber nicht optimal [3]. Man ersetzt sie besser durch an die eigene Sprache und landesübliche Normen angepasste Dokumentklassen und ergänzt diese wiederum durch zusätzliche Pakete, welche einzelne Eigenschaften einer Dokumentklasse modifizieren, verbessern oder ergänzen [3]. Da dieses Nebeneinander von alternativen Formatpaketen und zusätzlichen Paketen verwirrend sein kann, trifft es sich gut, daß gelegentlich die Verbesserungen bestimmter Pakete im Rahmen der jährlichen Aktualisierungen in den  $\LaTeX$ -Kern übernommen werden [3].

Die besten Textverarbeitungsprogramme ermöglichen immerhin die Verwendung von Formatvorlagen, sodaß man sich im Text auf ein strukturelles Markup, also Angaben zur strukturellen Gliederung des Textes beschränken kann [3]. Dies ist aber schon so kompliziert, daß sicher die weitaus meisten Anwender stattdessen einfach prozedurales Markup verwenden. Sie weisen also jeder Textpassage direkt eine bestimmte Formatierung zu und müssen selbst sehen, wie sie eine Einheitlichkeit der Formatierungen gewährleisten. Mit reinem  $\TeX$  müsste man ebenfalls prozedurales Markup benutzen, aber  $\LaTeX$  ermöglicht ähnlich wie HTML ein relativ einfaches Arbeiten mit strukturellem Markup, welches gleichzeitig ein explizites Markup ist, weil die Formatanweisungen sichtbar sind [3]. Genau wie HTML erlaubt allerdings auch  $\LaTeX$  die zusätzliche Anwendung von prozeduralem Markup [3], welches man möglichst sparsam verwenden sollte.

## 1.2 Hilfe und Bevormundung kommen sich oft zu nahe

Textverarbeitungsprogramme wie MS-Word wurden nicht für Programmierer und Buchautoren, sondern in erster Linie für Sekretärinnen und Privatanwender geschaffen, die Computer wie Schreibmaschinen einfach nur benutzen wollen, ohne sich näher mit deren Technik zu befassen. Dementsprechend sollen sie möglichst leicht zu bedienen sein und die Komplexität der technischen Realisierung vor den Anwendern verbergen. Formatierungen werden deshalb über Menüs ausgewählt und auf dem Bildschirm werden keine Formatierungsbefehle, sondern nur deren Ergebnisse angezeigt [2, 3]. Man nennt dies implizites Markup, weil statt der auch Tags genannten Formatierungsbefehle nur deren Wirkung zu sehen ist [3]. So muß sich der Anwender kaum etwas merken, kann praktisch ohne Vorkenntnisse sofort loslegen und wird nicht durch eine Mischung von eigentlichem Text und Formatierungsbefehlen irritiert. Daher eignen sich Textverarbeitungsprogramme wie MS-Word recht gut für das Schreiben von Briefen und ähnlich kleinen Texten,

besonders wenn deren gewünschte Formatierungen keinem Standard entsprechen [3]. Will man allerdings mit Textverarbeitungsprogrammen wirklich anspruchsvolle Projekte mit möglichst einheitlichem Erscheinungsbild realisieren, dann ist dies eher komplizierter als mit  $\LaTeX$  und die Qualität der Ergebnisse reicht doch nicht an die von  $\TeX$  heran. Darüber hinaus warnt  $\TeX$  sogar vor typographisch unbefriedigenden Ergebnissen [1]. Außerdem ist die WYSIWYG-Technik<sup>8</sup> ziemlich kompliziert und fehleranfällig und macht Textverarbeitungsprogramme langsam [2].

Gelegentlich hat es auch Nachteile, daß Textverarbeitungsprogramme dem Anwender nicht zeigen, was wirklich in einer Textdatei abgespeichert ist. Manchmal schleichen sich in Texten für den Anwender unsichtbare Steuerzeichen ein, die dann beispielsweise die Arbeit eines Druckers ohne erkennbaren Grund abbrechen lassen oder andere schwer behebbare Fehler zur Folge haben. Unangenehm ist auch die besonders von MS-Word bekannte Eigenschaft, mit neueren und teilweise sogar mit älteren Versionen geschriebene Texte nicht korrekt oder auch gar nicht lesen zu können. Was das Leben einfach machen soll, kann es also auch sehr erschweren.

Im Gegensatz zu MS-Word werden in  $\TeX/\LaTeX$  sämtliche Formatierungsbefehle ganz offen sichtbar und relativ leicht verständlich als explizites Markup ähnlich wie inzwischen auch bei HTML oder XML direkt in den Text hinein geschrieben und dem Autor bleibt kein Steuerzeichen verborgen [3]. Selbst wenn einmal ein Text für dessen korrekte Darstellung ein für das eigene Betriebssystem nicht existierendes Formatpaket oder fremde zusätzliche Pakete verlangt und deshalb nicht ganz korrekt interpretiert werden kann, so führt dies lediglich zu Fehlerwarnungen, aber der ganze Text ist trotzdem mit jedem beliebigen Editor vollständig lesbar. Anders als bei MS-Word muß man dann nicht unbedingt eine passende  $\TeX$ - oder  $\LaTeX$ -Version bzw. die entsprechenden Pakete besorgen, sondern kann einfach die in den Text eingestreuten  $\LaTeX$ -Befehle anpassen [3].

Mit Satzsystemen geschriebene Texte sind daher wegen ihrer offenen und leicht verständlichen Codierung in simpelstem ASCII-Code auch zukunftssicherer als Textverarbeitungsprogramme mit ihrem impliziten Markup. Deshalb kann man  $\TeX/\LaTeX$  auch mit jedem beliebigen Editor benutzen [1, 3]. Außerdem kann man  $\TeX$  und  $\LaTeX$  relativ leicht mit selbst definierten Makros an die eigenen Bedürfnisse anpassen oder erweitern [1, 3]. Andererseits zahlt man für die Vorteile natürlich den Preis, daß man als Benutzer von  $\LaTeX$  die für die eigene Arbeit am besten geeigneten Pakete kennen sowie sich zumindest die wichtigsten  $\LaTeX$ -Befehle merken muß und natürlich bei deren Eingabe auch Fehler machen kann [3].

### 1.3 $\TeX$ ist extrem stabil und läuft auf jedem Rechner

Das im Gegensatz zu Word et al. auf allen Rechnertypen und Betriebssystemen laufende Satzsystem  $\TeX$  [3] wurde von Donald Knuth bereits in den 70er Jahren und damit lange vor den heute üblichen Textverarbeitungsprogramme wie MS-Word oder Star Writer entwickelt. Schon 1983 erschien eine stabile Version von  $\TeX$  [1]. Der  $\TeX$ -Aufsatz  $\LaTeX$  zur Vereinfachung der Arbeit mit  $\TeX$  wurde von Leslie Lamport auch schon ab

---

<sup>8</sup>What you see is what you get.

1984 entwickelt [3], als es praktisch noch gar keine Personal Computer gab. Bereits seit 1994 ist  $\text{\LaTeX}2\epsilon$  der aktuelle  $\text{\LaTeX}$ -Standard [1]. Seine Implementationen laufen daher extrem stabil [2] und die von MS-Word bekannten Probleme mit inkompatiblen Dateiformaten existieren bei  $\text{\LaTeX}$  nicht [3]. Es gibt allerdings neben  $\text{\LaTeX}$  noch alternative oder zusätzliche  $\text{\TeX}$ -Aufsätze für spezielle Zwecke. Dazu gehören das in Mik $\text{\TeX}$  schon enthaltene pdf $\text{\LaTeX}$  für die direkte Ausgabe im pdf-Format, Hyper $\text{\LaTeX}$ <sup>9</sup> für ein paralleles Publizieren auf Papier und in HTML, XML $\text{\TeX}$  für XML als Eingabeformat, das moderne Con $\text{\TeX}$ t für die Verwendung aktueller Fähigkeiten des pdf-Formates unter Benutzung von pdf $\text{\TeX}$  und Bio $\text{\TeX}$ <sup>10</sup> für Sequenz-Alignments und Membranprotein-Topologien.

## 1.4 Außerdem kann $\text{\TeX}$ einfach mehr

$\text{\TeX}$  ist immer noch das Standardprogramm der Mathematiker und Informatiker sowie der Naturwissenschaftler, die viele mathematische oder chemische Formeln und oder Symbole in ihren Texten verwenden. Das ist nicht verwunderlich, denn die typographische Qualität der von  $\text{\TeX}$  produzierten Ausdrücke ist exzellent und besonders bei mathematischen Ausdrücken unerreicht [3]. Da der Quellcode von  $\text{\TeX}$  im Gegensatz zu dem von MS-Word frei zugänglich ist, konnten von den unter  $\text{\TeX}$ -Anwendern reichlich vorhandenen Programmierern bei Bedarf relativ leicht neue Module zur Erweiterung der Fähigkeiten von  $\text{\TeX}$  geschrieben werden [2, 3]. Darum ist es mit  $\text{\TeX}$  heute kein Problem, sämtliche Unicode-Zeichen zu verwenden, die kompliziertesten mathematischen und chemischen Formeln [2] oder technischen Zeichnungen einzugeben, sich selbst aktualisierende Querverweise einzubauen, automatisch Inhaltsverzeichnisse, Indexe und sogar Literaturverzeichnisse [2] zu erstellen sowie Texte im pdf-Format auszugeben. Man kann mit  $\text{\TeX}/\text{\LaTeX}$  in mehr als 50 Sprachen schreiben [3]. Autoren müssen nicht, können aber  $\text{\TeX}/\text{\LaTeX}$  die Rechtschreibprüfung sowie die optimale Platzierung von graphischen Elementen und Tabellen überlassen [2].

Da mit  $\text{\TeX}$  häufig umfangreiche Bücher geschrieben werden, unterstützt es auf einfachste Weise die Aufteilung langer Texte in handliche Teile, die problemlos in das Hauptdokument eingebunden werden. Dabei funktionieren Numerierungen von Seiten, Kapiteln oder Abbildungen problemlos so, als ob es sich um nur ein großes Dokument handelte. Insgesamt kann es wohl kein Textverarbeitungsprogramm mit der bei  $\text{\TeX}/\text{\LaTeX}$  vorhandenen Fülle von Funktionen aufnehmen [3]. Nebenbei hat die Modularität von  $\text{\TeX}$  und seinen Erweiterungen den Vorteil, daß durch Deklarationen zu Beginn eines Dokuments immer nur die Module eingebunden werden, die auch tatsächlich benötigt werden. Im Gegensatz zu MS-Word bleibt  $\text{\TeX}/\text{\LaTeX}$  aber selbst bei sehr komplexen Dokumenten mit vielen Grafiken und Tabellen belastbar und schnell [2].

---

<sup>9</sup><http://hyperlatex.sourceforge.net/html/hyperlatex.html>

<sup>10</sup><http://homepages.uni-tuebingen.de/beitz/biotex.html>

## 1.5 T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, Zusatzmodule und sogar Hilfe sind kostenlos

T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, fast alle Zusatzmodule und L<sup>A</sup>T<sub>E</sub>X-Alternativen sowie Anleitungen, Listen mit Fragen und Antworten und sogar Mailinglisten sowie Newsgroups für die Kommunikation unter den T<sub>E</sub>X-Anwendern und -entwicklern sind im Internet frei verfügbar [2]. In vielen Städten finden außerdem regelmäßig Treffen von T<sub>E</sub>X-Experten und Anfängern statt, auf denen man sich austauschen und Hilfe bekommen kann. Im Gegensatz zu MS-Word sind Updates nicht erforderlich, aber jederzeit kostenlos machbar.

## 2 Wie kommt man zu einem T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X-System unter Windows?

Um die von T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X erzeugten pdf-Dateien ansehen zu können, sollte man zunächst den Acrobat-Reader<sup>11</sup> installieren, wenn dies nicht längst geschehen ist [2]. Anschließend benötigt man für eine Installation auf einem Windows-Rechner die MiK<sub>T</sub>E<sub>X</sub>-Implementation<sup>12</sup> [2, 3] oder fpT<sub>E</sub>X [3]. Mit einem halbwegs aktuellen Internetanschluß kann man sich die 25 Megabyte große Sammlung der jeweils aktuellsten Pakete praktisch kostenlos von der MiK<sub>T</sub>E<sub>X</sub>-Internetseite auf den eigenen Rechner laden. Dort kann man MiK<sub>T</sub>E<sub>X</sub> aber auch auf einer CD bestellen. Die Datei install.html im Hauptverzeichnis erklärt, wie man MiK<sub>T</sub>E<sub>X</sub> danach installiert. Erst nach MiK<sub>T</sub>E<sub>X</sub> soll man Ghostscript<sup>13</sup> installieren [2], und tatsächlich hat bei mir mit einem schon vorher installierten Ghostscript die Einbindung von Graphiken nicht richtig funktioniert. Ghostscript benötigt man für ein funktionierendes L<sup>A</sup>T<sub>E</sub>X-System nur dann, wenn man sogenannte EPS-Grafiken aus Corel Draw, Excel oder Origin einbinden möchte [2]. Zuletzt braucht man noch irgendeinen Editor, wobei natürlich nicht jeder Editor die Arbeit mit T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X gleich gut unterstützt. Besonders einfach ist für den Anfänger das Schreiben mit dem kostenlosen T<sub>E</sub>XnicCenter<sup>14</sup>, welches unbedingt zuletzt installiert werden soll [2]. Bei diesem handelt es sich aber nicht nur einfach um einen Editor, sondern um eine von mehreren alternativen integrierenden Oberflächen (T<sub>E</sub>X-Shells), mit denen man die Texte auf Knopfdruck oder menügesteuert formatieren und anzeigen kann, ohne jedes einzelne Programm auf der Kommandozeilenebene aufrufen zu müssen [3].

Nun hat man einen Editor zum Schreiben und logischen Strukturieren von Texten, das T<sub>E</sub>X-Programm als Formatierer und Preview-Programme sowie T<sub>E</sub>X-Treiber für die Anzeige bzw. zum Ausdrucken von DVI- und PDF-Dateien [3]. Man kann mit dem Editor oder auch maschinell beispielsweise mit Datenbankprogrammen Texte erzeugen, die mit der Dateinamenserweiterung „.tex“ gekennzeichnet sein müssen und aus denen dann T<sub>E</sub>X geräteunabhängige Ausgabedateien mit der Endung „.dvi“ erzeugt [3]. Diese können mit Hilfe von T<sub>E</sub>X-Treibern entweder direkt ausgedruckt oder auf einem Bildschirm angezeigt werden oder in die ebenfalls geräteunabhängigen Ausgabedateiformate „Postscript“ (.ps) oder „Portable Document Format“ (.pdf) von Adobe geschrieben werden, welche

---

<sup>11</sup><http://www.adobe.de/>

<sup>12</sup><http://www.miktex.org/>

<sup>13</sup><http://www.cs.wisc.edu/%7Eghost/doc/AFPL/index.htm>

<sup>14</sup>[http://www.toolscenter.org/front\\_content.php](http://www.toolscenter.org/front_content.php)

schließlich mittels speziell an die eigene Hardware angepasster Treiber des jeweiligen Betriebssystems wahlweise über einen Drucker oder Bildschirm ausgegeben werden können [3]. Teilweise arbeiten auch normale Textverarbeitungsprogramme intern so ähnlich, aber da sie nicht modular sind, kann man sie für eine automatische Textgenerierung nicht gebrauchen, während beispielsweise die deutsche Bahn und das britische Patentamt mit  $\text{T}_{\text{E}}\text{X}$  automatisch Datenbankinhalte formatiert ausgeben lassen [3]. Außerdem kann man sich bei den monolithischen Textverarbeitungsprogrammen natürlich nicht aussuchen, mit welchem Editor man arbeiten will [3]. Auch hinsichtlich der Ausgabeformate sind Textverarbeitungsprogramme normalerweise weniger flexibel.

## Quellenverzeichnis

- [1] Ralf Banning ; Heiko A. Groeneveld: *Textsatz und Layout mit  $\text{E}_{\text{T}}\text{E}_{\text{X}}$* .  
<http://elib.uni-stuttgart.de/opus/doku/gestaltung/l2ebook.pdf>, Letzte Änderung:  
24. September 1998
- [2] Torsten Bronger ; Christian Faulhammer ; Mark Trettin: *Das  $\text{E}_{\text{T}}\text{E}_{\text{X}}$ -Tutorium*.  
<http://latex-tutorium.sourceforge.net/latex-tutorium.pdf>, Mai 2004
- [3] Christine Detig: *Der  $\text{E}_{\text{T}}\text{E}_{\text{X}}$ -Wegweiser [wissenschaftliche Texte ohne viel Aufwand, professionelle Ergebnisse mit Standardlösungen, Ausgabe nach PostScript und PDF]*. 2. Auflage. - Bonn : mitp; ISBN 3-8266-1414-3, Juni 2004

# Index

- TeXnicCenter, 6
- TeX-Shells, 6
- TeX-Treiber, 6
  
- Acrobat-Reader, 6
- Anleitungen, 6
- ASCII, 4
- Autor, 1–5
  
- Befehl, 2, 4
- Benutzerschnittstelle, 2
- Betriebssystem, 4, 7
- Bildschirm, 3, 6
- BioTeX, 5
  
- Codierung, 4
- ConTeXt, 5
  
- Deklarationen, 5
- Designer, 1
- Dilettantismus, 1
- Dokumentkategorie, 2
- Dokumentklasse, 2, 3
- Druckerei, 1, 2
- DVI, 6
  
- Editor, 4, 6, 7
  
- Fehler, 1, 4
- Fehlerwarnung, 4
- Formatierer, 6
- Formatierung, 1–4
- Formatierungsbefehl, 3, 4
- Formatpaket, 2–4
- Formatvorlage, 2, 3
- Formel
  - chemische, 5
  - mathematische, 3
- Funktionssammlung, 2
  
- Ghostscript, 6
- Gliederung, 2, 3
  
- Harmonisierung, 3
- Hilfe, 6
- HTML, 3–5
- HyperL<sup>A</sup>T<sub>E</sub>X, 5
  
- Implementation, 5, 6
- Index, 5
- Inhaltsverzeichnis, 5
- Internet, 1, 2, 6
  
- Knuth, 2, 4
- Kommandozeilenebene, 6
  
- Lamport, 2, 4
- Layout, 1–3
- Lektor, 1, 2
- Literaturverzeichnis, 5
  
- Makro, 2, 4
- Makrosprache, 2
- Manuskript, 1
- Markup
  - explizites, 3, 4
  - implizites, 3, 4
  - prozedurales, 3
  - strukturelles, 3
- Menü, 3
- MiKTeX, 6
- MikTeX, 5
- Module, 5
  
- Numerierung, 5
  
- Pakete, 3, 4, 6
- pdf, 5, 6
- pdfL<sup>A</sup>T<sub>E</sub>X, 5
- Personal Computer, 5
- Postscript, 6
- Preview-Programm, 6
  
- Quellcode, 3, 5
- Querverweis, 5
  
- Rechnertypen, 4
- Rechtschreibprüfung, 5
  
- Satzsystem, 4
- Setzer, 1, 2
- Sonderzeichen, 5
- Standardklassen, 3
- Star Writer, 4
- Steuerzeichen, 4

Symbole, 5

Tag, 3

Textsatzsysteme, 2

Textverarbeitungsprogramme, 2-5, 7

typographisch, 4, 5

Unicode, 5

Updates, 6

Verlag, 1, 2

Word, 2-6

WYSIWYG, 4

XML, 4, 5

XML<sub>L</sub>A<sub>T</sub>E<sub>X</sub>, 5

Zeichnungen, 5

zukunftsicher, 4