



Helmut Kopka

**L<sup>A</sup>T<sub>E</sub>X**

Band 1: Einführung

3., überarbeitete Auflage

**st**  
scientific tools



L<sup>A</sup>T<sub>E</sub>X

**Band 1: Einführung**  
**3., überarbeitete Auflage**

*Helmut Kopka*

# L<sup>A</sup>T<sub>E</sub>X

**Band 1: Einführung**  
**3., überarbeitete Auflage**

**eBook**

Die nicht autorisierte Weitergabe dieses eBooks  
an Dritte ist eine Verletzung des Urheberrechts!



ein Imprint der Pearson Education Deutschland GmbH

## Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei  
Der Deutschen Bibliothek erhältlich.

Die Informationen in diesem Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Buch erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen oder sollten als solche betrachtet werden.

### Umwelthinweis:

Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt.

Die Einschrumpffolie – zum Schutz vor Verschmutzung – ist aus umweltverträglichem und recyclingfähigem PE-Material.

10 9 8 7 6 5 4 3 2 1

05 04 03 02

ISBN 3-8273-7038-8

© 2000 by Addison-Wesley Verlag

Korrigierter Nachdruck 2002 bei Pearson Studium,

ein Imprint der Pearson Education Deutschland GmbH

Martin-Kollar-Straße 10–12, D-81829 München/Germany

Alle Rechte vorbehalten

Lektorat: Irmgard Wagner, Planegg, [irmgard.wagner@munich.netsurf.de](mailto:irmgard.wagner@munich.netsurf.de)

Korrektur: Andrea Stumpf, München

Herstellung: Kunigunde Huber, [khuber@pearson.de](mailto:khuber@pearson.de)

Satz: Helmut Kopka

Druck und Verarbeitung: Bercker Graphischer Betrieb, Kevelaer

Printed in Germany

# Vorwort

Textbearbeitung durch einen Rechner mit dem Ergebnis des Ausdrucks in Buchdruckqualität ist durch die Entwicklung geeigneter Programme in den letzten Jahren möglich geworden. Einige dieser Programme entstanden als spezielle Auftragsarbeiten von Großdruckereien oder von Herstellern von Setz- und Druckmaschinen und entziehen sich einer verbreiteten Nutzung. Allgemeinere Bedeutung hat das Satzprogramm  $\text{T}_{\text{E}}\text{X}$  gefunden. DONALD E. KNUTH von der Stanford University begann Mitte der 70er Jahre mit der Entwicklung von  $\text{T}_{\text{E}}\text{X}$ . Erste brauchbare Ergebnisse wurden bereits 1978 mit der Version  $\text{T}_{\text{E}}\text{X}78$  erzielt, an deren Verbesserung jedoch kontinuierlich weitergearbeitet wurde, bis schließlich mit  $\text{T}_{\text{E}}\text{X}82$  eine stabile Version bereitgestellt wurde.

Seit dieser Zeit hat  $\text{T}_{\text{E}}\text{X}$  weltweite Verbreitung gefunden und ist inzwischen für nahezu jeden Rechnertyp und jedes Betriebssystem verfügbar, angefangen vom Großrechner bis hin zum PC. Einer der Gründe für die Verbreitung liegt sicher darin, dass DONALD E. KNUTH das Programm zum öffentlichen Eigentum (public domain) erklärt hat. Hinzu kommt, dass das Programm in der Pascal-Metasprache WEB entwickelt wurde und der Original-Quellencode eine ausführliche Dokumentation mit umfassenden Erläuterungen enthält. Dies erst machte die Anpassung an verschiedene Rechner und Betriebssysteme mit vertretbarem Aufwand möglich.

Das Satzprogramm  $\text{T}_{\text{E}}\text{X}$  kann nahezu alle Aufgaben lösen, die bisher dem traditionellen Beruf des Setzers vorbehalten waren. Dies schließt den Satz von komplexen mathematischen Formeln und umfangreichen gerahmten Tabellen ein. Aber ebenso wie der Beruf des Setzers eine mehrjährige Ausbildung verlangt, setzt der erfolgreiche Einsatz von  $\text{T}_{\text{E}}\text{X}$  erhebliche Fachkenntnisse, sowohl von Programmiertechniken wie von Satz und Druck, voraus. Im Vergleich zu den üblichen Programmiersprachen erweist sich der Befehlssatz von  $\text{T}_{\text{E}}\text{X}$  als überaus umfangreich.  $\text{T}_{\text{E}}\text{X}$  kennt rund 900 Befehle, von denen etwa 300 Basisbefehle darstellen, aus denen weitere 600 Makrobefehle, teilweise mit wählbaren Parametern, abgeleitet wurden. Man vergleiche dies mit der Anzahl der Befehle und Sprachstrukturen etwa von C, Pascal oder FORTRAN.

Zwischen dem Autor und dem Setzer ist bei einer Verlagspublikation zusätzlich noch der Layouter eingeschaltet, der dem Manuskript die logische Gliederung entnimmt und hieraus ein Layout erarbeitet, das die logische Gliederung in eine dem Leser entgegenkommende grafische Aufbereitung umsetzt. Diese Aufgabe verlangt Expertenwissen und eine fast künstlerische Kreativität.

Ein wirklich erfolgreicher und zufriedenstellender Einsatz von  $\text{T}_{\text{E}}\text{X}$  setzt somit Programmierkenntnisse, Fertigkeiten der Satztechnik und grafische Kreativität, verbunden mit dem Wissen über deren psychologische Wirkung beim Leser, voraus. Unter diesen Voraussetzungen bliebe die Erstellung eigener Druckvorlagen einem kleinen Kreis begnadeter Alleskünstler vorbehalten.

LESLIE LAMPORNT hat mit  $\LaTeX$  ein Werkzeug zur Verfügung gestellt, mit dem auch wir arbeiten können. Statt mit  $\TeX$ -Befehlen die Arbeit des Setzers vornehmen und gleichzeitig die Kreativität des Layouters in entsprechende  $\TeX$ -Anweisungen umsetzen zu müssen, braucht der Anwender nur die logische Struktur anzugeben. Dazu gehört z. B. die Mitteilung, wann ein neues Kapitel oder ein neuer Abschnitt beginnt, welche Textstellen hervorzuheben sind, ob ein Inhaltsverzeichnis automatisch mitzuerstellen ist und Ähnliches mehr.

$\LaTeX$  übersetzt die angegebene logische Struktur in die gestaltenden  $\TeX$ -Befehle, mit denen der Text dann bearbeitet wird. Zusätzlich stellt  $\LaTeX$  eine Reihe von Layout-Stilen zur Verfügung. Mit der Auswahl eines geeigneten Layout-Stils wird gleichzeitig das grafische Design professionell gestaltet, ohne dass der Anwender eigene Angaben über passende Schriftarten und Schriftgrößen hinzufügen muss.

Das vorliegende Buch soll die Nutzung aller  $\LaTeX$ -Möglichkeiten vermitteln. Diese gestalten selbstverständlich auch individuelle Textformatierungen. Der Anwender sollte jedoch die angebotenen Standardformate bevorzugen, da diese das grafische Design von Fachleuten verwirklichen, mit denen der Normalanwender kaum erfolgreich konkurrieren kann.

Ich möchte an dieser Stelle LESLIE LAMPORNT danken, der mit vielen Anregungen und konstruktiver Kritik am Zustandekommen dieses Buches beteiligt war.

Helmut Kopka, Juli 1988

## Vorwort zur Herausgabe der Buchserie $\LaTeX$

Der Vorläufer dieses Buches mit dem Titel „ $\LaTeX$  – Eine Einführung“ erlebte zwischen 1988 und 1993 vier Auflagen, wobei die vierte Auflage mit den Erscheinungsjahren 1992 und 1993 eigentlich aus zwei eigenständigen Ausgaben bestand, zwischen denen es deutliche Textumstellungen und Ergänzungen gab. Als Ergänzung zur Einführung erschien 1990 ein zweites Buch mit dem Titel „ $\LaTeX$ -Erweiterungsmöglichkeiten“ und dem Untertitel „Mit einer Einführung in METAFONT“, das zwischen 1990 und 1993 in drei jeweils deutlich erweiterten Auflagen herauskam. Dabei kam es zunehmend zu Überschneidungen zwischen beiden Büchern, die überdies immer voluminöser wurden und in ihren Ergänzungen über die Zielrichtung der Buchtitel hinausgingen. Dies hatte zur Folge, dass Leser/innen der Bücher auch mit Darstellungsstoff konfrontiert wurden, der dem aktuellen Nutzungsbedürfnis nicht oder noch nicht zum Zeitpunkt des Einstiegs in  $\LaTeX$  entsprach.

Auf Anregung des damaligen Präsidenten der deutschsprachigen  $\TeX$ -Anwendervereinigung, JOACHIM LAMMARSCH, schlug mir der Verlag vor,  $\LaTeX$  in Form einer dreibändigen Buchserie zu präsentieren:

Band 1:  $\LaTeX$ -Einführung

Band 2:  $\LaTeX$ -Ergänzungen – mit einer Einführung in METAFONT

Band 3:  $\LaTeX$ -Erweiterungen

Der vorliegende Band 1 geht weitgehend auf das ursprüngliche Buch „ $\LaTeX$  – Eine Einführung“ zurück und beschränkt sich nun auf die Vorstellung des internationalen  $\LaTeX$ -Standards, lediglich ergänzt um die Darstellung von `german.sty` zur Einbindung deutscher Besonderheiten in die  $\LaTeX$ -Bearbeitung. Letztere muss man für deutschsprachige Anwender, und an diese richtet sich das Buch, als zu *unserem Standard gehörend* betrachten.

In Bezug auf den internationalen  $\LaTeX$ -Standard ist die Vorstellung in Band 1 aber *vollständig*. Sie schließt somit  $\BIBTeX$  und  $\MAKEINDEX$  ein, da diese  $\LaTeX$ -Ergänzungen

vom Programmautor LESLIE LAMPORT selbst stammen oder unter seiner aktiven Mithilfe entstanden und von ihm zum Bestandteil des  $\LaTeX$ -Gesamtpakets erklärt wurden.

Unmittelbar nach Drucklegung der 1. Auflage von Band 1 erschien auf den öffentlichen  $\TeX$ -Fileservern die neue  $\LaTeX 2_{\epsilon}$ -Version, zunächst als Probeversion und ausdrücklich als solche gekennzeichnet. Nach einer halbjährigen Erprobungsphase mit Behebung einer Reihe von Fehlern und Schwächen wurde im Juni 1994 die  $\LaTeX 2_{\epsilon}$ -Probeversion zur Standard- $\LaTeX$ -Version erklärt. Die neuen Eigenschaften von  $\LaTeX 2_{\epsilon}$  wurden deshalb zunächst in Kapitel 1 von Band 2 nachgetragen. Mit den Neuauflagen von Band 1 erscheinen die Grundeigenschaften von  $\LaTeX 2_{\epsilon}$  nunmehr sachgerecht in der Einführung.

Der Einführungsband 1 schließt ab mit Hinweisen zur  $\TeX$ -Installation und der Erstellung der erforderlichen Formatfiles im Anhang F. Bei der Zufügung dieses 46seitigen Anhangs befand ich mich in einem Konflikt: Er gehört im engeren Sinne sicherlich nicht zum Stoffbereich einer  $\LaTeX$ -Einführung. Alle mir bekannten Bücher über  $\TeX$  und seine Makropakete gehen stillschweigend von der Annahme aus, dass ein lauffähiges  $\TeX$ -Programm mit den erforderlichen Zusatzwerkzeugen im Rechner des Anwenders existiert. Dies war in den Anfangsjahren von  $\TeX$  und  $\LaTeX$  auch sachgerecht, da  $\TeX$  damals zunächst in den Rechenzentren der Hochschulen und Forschungsinstitute bereitgestellt wurde. Alle bei der Installation eines  $\TeX$ -Systems vorausgesetzten Kenntnisse und auftretenden Probleme stellten sich nicht dem Anwender, sondern dem entsprechenden Experten des Rechenzentrums.

Inzwischen hat sich das Anwenderprofil deutlich geändert. Die Mehrzahl der  $\TeX$ - und  $\LaTeX$ -Anwender betreibt das Programm auf einem PC. Für nahezu alle Individualrechner (IBM-PCs und kompatible, Atari, Amiga, Macintosh und UNIX-Workstations) stehen sowohl kommerzielle wie auch kostenlose PD- (Public Domain) oder SW- (Shareware)  $\TeX$ -Pakete zur Verfügung. Dokumentation und Installationshilfen sind je nach Programmquelle unterschiedlich hilfreich.

Der typische  $\TeX$ -Einsteiger will das Programm auf dem eigenen Individualrechner betreiben und die eigenen Kenntnisse auf die Nutzungsbedürfnisse beschränken und nicht mit komplexen Wechselbeziehungen zwischen den diversen Programmteilen einer  $\TeX$ - und  $\LaTeX$ -Installation konfrontiert werden. Das Programmsystem wurde vermutlich als Diskettensatz beschafft oder von einem Kollegen kopiert, was bei den PD- und SW-Produkten auch erlaubt ist. Je nach beigefügter Dokumentation und Installationshilfe kann sich die Installation für den Einsteiger als schwierige Hürde erweisen. Auch wenn die eigentliche Installation ohne Probleme gelingt, meldet das Programm eventuell beim ersten Aufruf, dass es gewisse Teile nicht findet und damit die Bearbeitung abbricht.

Die Ursache für eine solche Meldung kann tatsächlich darin liegen, dass der Diskettensatz für die Installation zwar das ausführbare  $\TeX$ -Programm bereitstellt, die für den praktischen Ablauf aber zwingend erforderlichen Zusatzwerkzeuge, wie bestimmte Zeichensatzfiles und Makropakete, aus anderen Quellen beschafft werden müssen, ohne dass dies in der beigefügten Dokumentation explizit gesagt wird. Häufig liegt die Ursache für die genannte Fehlermeldung aber darin, dass bestimmte Programmteile zwar vorhanden sind, aber für die Nutzung mit dem beigefügten Spezialprogramm INITEX, von dem der Einsteiger bis dahin noch nie etwas gehört hat, aufbereitet werden müssen. Ich hoffe, mit dem Anhang F auch dem Anfänger bei seinem Einstieg in  $\TeX$  auf dem eigenen PC behilflich zu sein.

Band 2 der Buchserie über  $\LaTeX$  beginnt mit der Vorstellung von  $\LaTeX$ -Ergänzungen, die in ihrer Wirkung in Zukunft Bestandteil von  $\LaTeX 3$  werden und die bereits heute in Form allgemein zugänglicher Ergänzungspakete genutzt werden können. Die Standardzeichensätze einer  $\LaTeX$ -Installation waren primär auf Anwendungen aus dem mathematisch-

naturwissenschaftlichen Bereich zugeschnitten. Mit der Ausbreitung auch auf geisteswissenschaftliche Anwendungen treten zunehmend Anforderungen auf, die damit nicht zu erfüllen sind. Inzwischen existieren für nahezu alle Sprachen und Sonderfälle, wie z. B. zur Schachdokumentation oder zum Musiknotensatz, geeignete Zeichensätze. Band 2 stellt eine Vielzahl solcher ergänzender Zeichensätze vor, wobei sich die Ergänzungen in Richtung  $\LaTeX$  3 als ganz besonders hilfreich erweisen. Auch die Einbeziehung von PostScript-Zeichensätzen in die  $\LaTeX$ -Bearbeitung wird angesichts der immer häufiger verwendeten PostScript-fähigen Drucker in Kapitel 5 vorgestellt. Die Nutzung von Zeichensätzen für den Musiknotensatz zusammen mit einem geeigneten Ergänzungspaket wird in Kapitel 4 beschrieben.

Band 2 stellt weiterhin die Möglichkeiten zur Einbindung von Bildern und Grafiken vor, die entweder aus völlig anderen Programmquellen stammen oder mit  $\TeX$ -eigenen Mitteln, wie mit  $\Pi\TeX$ , erzeugt werden. Er schließt ab mit einer Einführung in METAFONT in Kapitel 8. Angesichts der Vielzahl der vorgestellten Zusatz-Zeichensätze ist dies eine folgerichtige Ergänzung, da die Installation der Zusatz-Zeichensätze bei vielen Anwendern aus den Quelldateien zu erfolgen hat, womit der Programmaufruf von METAFONT mit geeigneten Einstellparametern zwingend notwendig wird. Band 2 wendet sich also an Anwender, die über die Möglichkeiten einer Standard- $\LaTeX$ -Installation hinausgehen wollen, ohne hierzu in die Tiefen der Programmierung zur Entwicklung von Eigenerweiterungen steigen zu müssen.

Für solche Entwicklungen ist schließlich der Band 3 gedacht. Er stellt  $\LaTeX$  in seinen internen Strukturen vor, ergänzt um eine Darstellung der wichtigsten  $\TeX$ -Strukturen. Mit diesen Kenntnissen werden dann anschließend Beispiele für anwendereigene  $\LaTeX$ -Erweiterungen vorgestellt. Ebenso werden Interna des  $\text{BIB}\TeX$ -Programms angesprochen, aus denen der Anwender weitere  $\text{BIB}\TeX$ -Stilfiles zur variablen Gestaltung von Literaturverzeichnissen erstellen kann.

Jede  $\TeX$ -Installation kennt weitere  $\TeX$ -Zusatzwerkzeuge, von denen ich hier beispielhaft das Programm `patgen` nenne. Mit diesem Programm kann man für jede Sprache ein  $\TeX$ -spezifisches Trennmusterfile erstellen, indem als Eingabe ein Trennlexikon der entsprechenden Sprache herangezogen wird. Band 3 stellt in seinem Anhang alle  $\TeX$ -Standard-Zusatzwerkzeuge vor und beschreibt deren Anwendung und Eigenschaften.

Ich habe zu den vorausgegangenen  $\LaTeX$ -Büchern eine Vielzahl von Zuschriften mit Anregungen und konstruktiver Kritik erhalten, für die ich mich, soweit die Briefe nicht direkt beantwortet wurden, an dieser Stelle bedanke. Viele der bei der Neuauflage vorgenommenen Korrekturen und Ergänzungen gehen auf solche Zuschriften zurück. Ich würde mich freuen, wenn auch die neue Buchserie den Leserkontakt fortsetzt. Wie in der Vergangenheit werde ich, soweit mir das möglich ist, auch weiterhin alle sachlichen Anfragen beantworten.

Sehr hilfreich war die spontane Bereitschaft einiger Leser und Kollegen, bei der Neuauflage Korrektur zu lesen. Ich möchte mich hierfür, auch im Namen zukünftiger Leser, ganz besonders bei DR. GÜNTER GREEN, Universität Kiel, DR. JOHANN STRUTZ, Universität Klagenfurt, VOLKER SCHAA, Darmstadt, HEINZ KUSZNIER, Linz, DR. PETRA RÜBE-PUGLIESE, Berlin, THOMAS LAUKE, Berlin, STEFAN BARTELS, Hamburg, bedanken.

Ebenso möchte ich mich hier beim ehemaligen Präsidenten der deutschsprachigen  $\TeX$ -Anwendervereinigung, JOACHIM LAMMARSCH, Universität Heidelberg, bedanken. Sein Vorschlag zur Neuauflage der  $\LaTeX$ -Buchserie in drei Bänden war begleitet von einem detaillierten Strukturierungsvorschlag, den ich zur Grundlage der Stoffaufteilung genutzt habe.

Helmut Kopka, September 1993, November 1995 und 1996

## Nachtrag zum Vorwort zur 3. Auflage der $\LaTeX$ -Einführung

Entsprechend der  $\LaTeX$ -Entwicklungsgeschichte bezog sich die erste Auflage des Einführungsbandes dieser Buchserie noch auf  $\LaTeX$  2.09. Die Erstvorstellung von  $\LaTeX$  2 $\epsilon$  erfolgte deshalb zunächst in der Anfangsausgabe von Band 2, die mit der zweiten Auflage der Einführung sachgerecht dort ihren Niederschlag fand. In dieser zweiten Auflage und ihren korrigierten Nachdrucken der  $\LaTeX$ -Einführung wurde sowohl  $\LaTeX$  2.09 als auch  $\LaTeX$  2 $\epsilon$  vorgestellt, so dass es sich eigentlich um eine Einführung in beide  $\LaTeX$ -Versionen handelte. Auf den letztjährigen Treffen der deutschsprachigen  $\TeX$ -Anwendervereinigung wurde ich zunehmend aufgefordert, die Einführung auf  $\LaTeX$  2 $\epsilon$  zu beschränken, da  $\LaTeX$  2.09 inzwischen als überholt anzusehen ist und dessen parallele Einführung für  $\LaTeX$ -Neueinsteiger didaktisch erschwerend wirkt.

Mit der 3. Auflage der  $\LaTeX$ -Einführung bin ich dieser Aufforderung gefolgt, was mir umso leichter fiel, als  $\LaTeX$  2 $\epsilon$  über den sog.  $\LaTeX$  2.09-Kompatibilitätsmodus verfügt. Damit werden alte  $\LaTeX$ -Eingabedateien, die zur Bearbeitung mit  $\LaTeX$  2.09 vorgesehen waren, von  $\LaTeX$  2 $\epsilon$  als solche erkannt und so bearbeitet, wie das früher der Fall war, ohne dass der  $\LaTeX$  2 $\epsilon$ -Anwender die Bearbeitungs- und Eingabeunterschiede für  $\LaTeX$  2.09 kennen muss.

Letzteres wäre in gewissem Umfang nur dann erforderlich, wenn alte  $\LaTeX$  2.09-Eingabetexte modifiziert und anschließend im  $\LaTeX$  2.09-Kompatibilitätsmodus bearbeitet werden sollten. Für solche exotischen Aufgaben kann bei Bedarf der Befehlsindex dieses Buches herangezogen werden, der in lexikalischer Ordnung alle  $\LaTeX$ -Befehle, einschließlich derjenigen aus  $\LaTeX$  2.09, mit einer entsprechenden Kennzeichnung vorstellt und in Kurzfassung erläutert.

Eine weitere deutliche Überarbeitung erfuhr der Anhang F für die 3. Auflage der  $\LaTeX$ -Einführung. Die Hinweise zur Installation von  $\TeX$ -Systemen auf PCs beschränkten sich bei den bisherigen Auflagen auf drei kostenlose Public Domain- bzw. Shareware-Angebote unter MS-DOS sowie einer Ergänzung für OS/2. Mit dem Siegeszug von MS-Windows als vorherrschendes Betriebssystem für PCs bedurften die Installationshinweise für  $\TeX$ -Systeme dringend der Ergänzung für die moderneren 32-bit-Windows- und Linux-Betriebssysteme, was mit der 3. Auflage nun erfolgt. Dabei wurden gleichzeitig die bisherigen Vorstellungen von mehreren  $\TeX$ -Systemen unter MS-DOS auf das gebräuchlichste  $\TeX$ -System reduziert.

Diese Ergänzung der Installationshinweise für weitere PC-Betriebssysteme erhält ihre Berechtigung und Nützlichkeit auch durch die dem Buch beigefügte CD-ROM. Diese enthält für die wichtigsten Betriebssysteme von Workstations und PCs lauffähige  $\TeX$ -Systeme, die beim Anwender auf sein Rechnersystem kopiert und genutzt werden können. Für die Bereitstellung dieser CD-ROM danke ich SEBASTIAN RAHTZ, der zur Einhaltung der Terminvorgaben durch den Addison-Wesley-Verlag für die 3. Auflage dieser Einführung dem Verlag eine Vorläuferversion für die endgültige  $\TeX$ -Live 5c CD-ROM der  $\TeX$ -Anwendervereinigungen bereit und zur Verfügung gestellt hat.

Gleichzeitig danke ich der zuständigen Lektorin des Addison-Wesley-Verlags, Frau Irmgard Wagner, die mit großem Entgegenkommen meinen Änderungs- und Terminwünschen nachkam und damit ganz wesentlich am zügigen Zustandekommen dieser deutlich überarbeiteten Neuauflage beteiligt war.

Helmut Kopka, März 2000



# Inhaltsverzeichnis

<b>1</b>	<b>Grundlagen</b>	<b>1</b>
1.1	T <sub>E</sub> X und L <sup>A</sup> T <sub>E</sub> X . . . . .	2
1.2	L <sup>A</sup> T <sub>E</sub> X-Entwicklungsgeschichte . . . . .	3
1.3	Erläuterungen zum vorliegenden Text . . . . .	4
1.4	Text und Befehle . . . . .	5
1.5	Grundstruktur eines L <sup>A</sup> T <sub>E</sub> X-Files . . . . .	6
1.6	L <sup>A</sup> T <sub>E</sub> X-Bearbeitungsmodi . . . . .	7
1.7	Die Erzeugung eines L <sup>A</sup> T <sub>E</sub> X-Dokuments . . . . .	8
1.8	Hinweise für Autoren . . . . .	9
<b>2</b>	<b>Befehle und Umgebungen</b>	<b>11</b>
2.1	Befehlsnamen und Befehlsargumente . . . . .	11
2.2	Umgebungen (environments) . . . . .	13
2.3	Erklärungen (declarations) . . . . .	13
2.4	Maßangaben . . . . .	14
2.4.1	Feste Maße . . . . .	14
2.4.2	Elastische Maße . . . . .	15
2.5	Sonderzeichen . . . . .	15
2.5.1	Die Eingabe der Umlaute und des ß . . . . .	15
2.5.2	Anführungsstriche . . . . .	16
2.5.3	Trenn-, Binde-, Strecken- und Gedankenstriche . . . . .	17
2.5.4	Der Druck von Befehlszeichen . . . . .	17
2.5.5	Die Sonderzeichen §, †, ‡, ¶, © und ℒ . . . . .	17
2.5.6	Sonderbuchstaben in Fremdsprachen . . . . .	17
2.5.7	Akzente . . . . .	18
2.5.8	Ligaturen . . . . .	18
2.5.9	Das Datum . . . . .	18
2.5.10	Vorgriff auf german.sty . . . . .	18
2.6	Zerbrechliche Befehle . . . . .	19
2.7	Übungen . . . . .	20
2.8	Anmerkungen zur L <sup>A</sup> T <sub>E</sub> X-Philosophie . . . . .	24

<b>3</b>	<b>Dokumentklassen und Seitenstil</b>	<b>25</b>
3.1	Die Dokumentklasse . . . . .	25
3.1.1	Dokument-Standardklassen . . . . .	25
3.1.2	Klassenoptionen . . . . .	26
3.1.3	Einstellparameter für einige der vorstehenden Klassenoptionen . . . .	28
3.1.4	Ergänzungspakete . . . . .	29
3.1.5	L <sup>A</sup> T <sub>E</sub> X 2.09-Kompatibilitätsmodus . . . . .	30
3.2	Der Seitenstil . . . . .	31
3.2.1	Kopfdeklarationen . . . . .	32
3.2.2	Seitennummerierung . . . . .	32
3.2.3	Makropaket-Exporte . . . . .	33
3.2.4	Zeilen- und Absatzabstände . . . . .	34
3.2.5	Seitendeklarationen . . . . .	35
3.2.6	Ein- und zweispaltige Seiten . . . . .	36
3.3	Dokumentuntergliederung . . . . .	37
3.3.1	Die Titelseite . . . . .	37
3.3.2	Die Zusammenfassung – der Abstract . . . . .	39
3.3.3	Die fortlaufende Untgliederung . . . . .	40
3.3.4	Der Anhang . . . . .	42
3.3.5	Zusätzliche Buchuntergliederungen . . . . .	42
3.3.6	Die Bearbeitungsklasse proc . . . . .	43
3.4	Das Inhaltsverzeichnis . . . . .	43
3.4.1	Automatische Eintragungen . . . . .	43
3.4.2	Der Ausdruck des Inhaltsverzeichnisses . . . . .	44
3.4.3	Zusätzliche Eintragungen . . . . .	44
3.4.4	Weitere Verzeichnisse . . . . .	45
3.5	Formatierungshilfen . . . . .	45
3.5.1	Zeichen- und Wortabstände . . . . .	45
3.5.1.1	Der . (Punkt) und das Satzende . . . . .	46
3.5.1.2	Frenchspacing . . . . .	46
3.5.1.3	Die Zeichenkombination “ ‘ und ’ ” (kleiner Zusatzzwi- schenraum) . . . . .	46
3.5.1.4	Italic-Korrektur . . . . .	46
3.5.1.5	Die Ausschaltung von Ligaturen . . . . .	47
3.5.1.6	Einfügung beliebiger Zwischenräume . . . . .	47
3.5.1.7	Einfügung von . . . . . und ____ Sequenzen . . . . .	48
3.5.2	Zeilenumbruch . . . . .	48
3.5.2.1	Die Befehle \\ <code> und \\<code>newline . . . . .</code></code>	48
3.5.2.2	Weitere Zeilenumbruchbefehle . . . . .	49
3.5.3	Absatzabstand . . . . .	49
3.5.4	Absatzeinrückungen . . . . .	50

3.5.5	Seitenumbruch . . . . .	50
3.5.5.1	Normale Textseiten . . . . .	50
3.5.5.2	Seiten mit Bildern und Tabellen . . . . .	51
3.5.5.3	Zweispaltige Seiten . . . . .	51
3.5.5.4	Doppelseitiger Druck . . . . .	51
3.5.5.5	Selektive Seitenhöhenänderung . . . . .	52
3.5.5.6	Eingeschränkter Umbruch . . . . .	52
3.5.5.7	Einige Anmerkungen zu manuellen Formatierungshilfen . . . . .	53
3.6	Trennungshilfen . . . . .	54
3.6.1	Direkte Trennungshilfen . . . . .	54
3.6.2	Erzeugung einer Trennungsliste . . . . .	55
3.6.3	Vermeidung von Trennungen . . . . .	55
3.6.4	Zusatzinformation über Trennungen . . . . .	56
<b>4</b>	<b>Texthervorhebungen</b>	<b>57</b>
4.1	Änderung der Schrift . . . . .	57
4.1.1	Umschaltung für Schrifthervorhebungen . . . . .	57
4.1.2	Die Wahl der Schriftgröße . . . . .	58
4.1.3	Zeichensatzattribute . . . . .	59
4.1.4	Zeichensatzbefehle mit Textargumenten . . . . .	62
4.1.5	Zeichensatzauswahl mit $\LaTeX$ 2.09 . . . . .	63
4.1.6	Zusätzliche Schriften . . . . .	64
4.1.7	Zeichensätze und Symbole . . . . .	65
4.2	Textausrichtungen . . . . .	66
4.2.1	Zentrierter Text . . . . .	66
4.2.2	Einseitig bündiger Text . . . . .	66
4.2.3	Beidseitig eingerückter Text . . . . .	67
4.2.4	Verseinrückungen . . . . .	68
4.3	Aufzählungen . . . . .	69
4.3.1	Beispiel <code>itemize</code> . . . . .	69
4.3.2	Beispiel <code>enumerate</code> . . . . .	69
4.3.3	Beispiel <code>description</code> . . . . .	70
4.3.4	Verschachtelte Aufzählungen . . . . .	70
4.3.5	Änderung der Markierungen . . . . .	72
4.3.6	Literaturverzeichnis . . . . .	73
4.4	Allgemeine Listen . . . . .	75
4.4.1	Die Standardmarke . . . . .	75
4.4.2	Die Listenerklärung . . . . .	76
4.4.3	Beispiel für eine benutzergestaltete Liste . . . . .	77
4.4.4	Listendefinitionen als neue Umgebungen . . . . .	78
4.4.5	Triviale Listen . . . . .	79
4.4.6	Verschachtelte Listen . . . . .	79

4.5	Regelsätze . . . . .	80
4.6	Tabulatorsetzungen . . . . .	81
4.6.1	Grundlagen . . . . .	81
4.6.2	Musterzeile . . . . .	81
4.6.3	Tabstops und linker Rand . . . . .	82
4.6.4	Weitere Tabulatorbefehle . . . . .	82
4.6.5	Zusatzbemerkungen . . . . .	83
4.7	Boxen . . . . .	85
4.7.1	LR-Boxen . . . . .	85
4.7.2	LR-Box-Speicherungen . . . . .	86
4.7.3	Vertikale Verschiebungen von LR-Boxen . . . . .	87
4.7.4	Absatzboxen und Teilseiten . . . . .	87
4.7.5	Positionierungsprobleme bei vertikalen Boxen . . . . .	90
4.7.6	Balkenboxen . . . . .	91
4.7.7	Verschachtelte Boxen . . . . .	92
4.7.8	Box-Stilparameter . . . . .	93
4.8	Tabellen . . . . .	94
4.8.1	Die Konstruktion von Tabellen . . . . .	94
4.8.2	Die Änderung des Tabellenstils . . . . .	96
4.8.3	Beispiele von Tabellenkonstruktionen . . . . .	97
4.8.4	Gleitende Tabellen . . . . .	104
4.9	Fußnoten und Randnotizen . . . . .	106
4.9.1	Standardfußnoten . . . . .	106
4.9.2	Abweichungen vom Standard . . . . .	106
4.9.3	Änderung des Fußnotenstils . . . . .	107
4.9.4	Fußnoten in unerlaubten Modi . . . . .	108
4.9.5	Fußnoten in Minipages . . . . .	109
4.9.6	Randnotizen . . . . .	109
4.9.7	Stilparameter für Randboxen . . . . .	111
4.10	Ausdruck von Originaltext . . . . .	111
4.10.1	Standardausgabe . . . . .	111
4.10.2	Das Ergänzungspaket <code>alltt.sty</code> . . . . .	112
4.10.3	Das Ergänzungspaket <code>shortvrb.sty</code> . . . . .	113
4.11	Kommentare im Eingabetext . . . . .	113
<b>5</b>	<b>Mathematische Formeln</b> . . . . .	<b>115</b>
5.1	Mathematische Umgebungen . . . . .	115
5.2	Die Hauptkonstruktionselemente . . . . .	116
5.2.1	Konstanten, Variablen und ihre Verknüpfungen . . . . .	116
5.2.2	Hoch- und Tiefstellungen von Zeichen . . . . .	117
5.2.3	Brüche . . . . .	117
5.2.4	Wurzeln . . . . .	118
5.2.5	Summen und Integrale . . . . .	118
5.2.6	Fortsetzungspunkte – Ellipsen . . . . .	119

5.3	Mathematische Symbole . . . . .	120
5.3.1	Griechische Buchstaben . . . . .	120
5.3.2	Kalligraphische Buchstaben . . . . .	120
5.3.3	Binäre Operationssymbole . . . . .	121
5.3.4	Vergleichssymbole und deren Negation . . . . .	121
5.3.5	Pfeil- oder Zeigersymbole . . . . .	122
5.3.6	Verschiedene sonstige Symbole . . . . .	122
5.3.7	Symbole in zwei Größen . . . . .	123
5.3.8	Funktionsnamen . . . . .	123
5.3.9	Mathematische Akzente . . . . .	124
5.4	Weitere Konstruktionselemente . . . . .	125
5.4.1	Automatische Größenanpassung von Klammersymbolen . . . . .	126
5.4.2	Gewöhnlicher Text innerhalb von Formeln . . . . .	127
5.4.3	Matrizen und Felder . . . . .	127
5.4.4	Über- und Unterstreichen von Teilformeln . . . . .	130
5.4.5	Aufgestockte Symbole . . . . .	130
5.4.6	Zusätzliche mathematische $\TeX$ -Befehle . . . . .	131
5.4.7	Mehrzeilige Formeln . . . . .	132
5.4.8	Gerahmte oder nebeneinander stehende Formeln . . . . .	135
5.4.9	Chemische Formeln – Fettdruck in Formeln . . . . .	136
5.5	Mathematische Formatierungshilfen . . . . .	137
5.5.1	Horizontale Abstände . . . . .	137
5.5.2	Die Wahl der Schriftgrößen in Formeln . . . . .	138
5.5.3	Manuelle Größenwahl der Klammersymbole . . . . .	140
5.5.4	Schriftumschaltung in mathematischen Formeln . . . . .	141
5.5.5	Beseitigung einer $\LaTeX$ -Schwäche mit <code>exscale</code> . . . . .	141
5.5.6	Mathematische Stilparameter . . . . .	143
5.5.7	Einige Zusatzeempfehlungen . . . . .	143
5.5.8	Gerahmte abgesetzte Formeln . . . . .	145
5.5.9	Was ist sonst noch möglich? . . . . .	146
<b>6</b>	<b>Bilder</b>	<b>147</b>
6.1	Maß- und Positionierungsangaben . . . . .	147
6.2	Die Bildumgebung – <code>picture</code> . . . . .	148
6.3	Die Positionierungsbefehle . . . . .	149
6.4	Die Bildobjekt-Befehle . . . . .	150
6.4.1	Text im Bild . . . . .	150
6.4.2	Bildboxen – Rechtecke . . . . .	150
6.4.3	Gerade Linien . . . . .	153
6.4.4	Pfeile . . . . .	154
6.4.5	Kreise . . . . .	155
6.4.6	Ovale und gerundete Ecken . . . . .	155
6.4.7	Bezier-Kurven . . . . .	157
6.4.8	Vertikal aufgestockte Texte . . . . .	158
6.4.9	Textangepasste Rahmen . . . . .	159

6.5	Weitere Bildbefehle und Beispiele . . . . .	160
6.5.1	Strichstärken . . . . .	160
6.5.2	Verschachtelte Bilder . . . . .	160
6.5.3	Speicherung von Bildteilen . . . . .	161
6.5.4	Erweiterte picture-Umgebung . . . . .	164
6.5.5	Weitere Beispiele . . . . .	165
6.5.6	picture-Ergänzungspakete . . . . .	166
6.5.7	Allgemeine Empfehlungen . . . . .	167
6.6	Gleitende Tabellen und Bilder . . . . .	168
6.6.1	Die Platzierung von Gleitobjekten . . . . .	168
6.6.2	Gleitobjekt-Verbote . . . . .	170
6.6.3	Stilparameter für gleitende Objekte . . . . .	170
6.6.4	Über- und Unterschriften für gleitende Objekte . . . . .	171
6.6.5	Beispiele für Gleitobjekte . . . . .	173
6.6.6	Bild- und Tabellenreferenzen im Text . . . . .	175
6.7	Hinweise auf weitere Grafikwerkzeuge . . . . .	175
<b>7</b>	<b>Benutzereigene Strukturen</b>	<b>177</b>
7.1	Zähler . . . . .	177
7.1.1	L <sup>A</sup> T <sub>E</sub> X-eigene Zähler . . . . .	177
7.1.2	Benutzereigene Zähler . . . . .	178
7.1.3	Veränderung der Zählerwerte . . . . .	178
7.1.4	Die Ausgabe von Zählerständen . . . . .	179
7.2	Längen . . . . .	180
7.3	Benutzereigene Befehle . . . . .	181
7.3.1	Befehle ohne Argumente . . . . .	181
7.3.2	Befehle mit Argumenten . . . . .	183
7.3.3	Befehle mit einem zusätzlichen optionalen Argument . . . . .	184
7.3.4	Verbesserung für anwendereigene mathematische Befehle . . . . .	185
7.3.5	Weitere Beispiele . . . . .	185
7.4	Das Ergänzungspaket ifthen . . . . .	189
7.5	Benutzereigene Umgebungen . . . . .	191
7.5.1	Umgebungen ohne Argumente . . . . .	192
7.5.2	Umgebungen mit Argumenten . . . . .	193
7.5.3	Umgebungen mit einem zusätzlichen optionalen Argument . . . . .	194
7.6	Allgemeine Bemerkungen zu Benutzerstrukturen . . . . .	196
7.6.1	Gleiche Befehls- und Zählernamen . . . . .	196
7.6.2	Abspeichern von benutzereigenen Strukturen . . . . .	196
7.6.3	Strukturen zur Abkürzung . . . . .	197
7.6.4	Die Reichweite benutzereigener Definitionen . . . . .	197
7.6.5	Argumentbegrenzungen . . . . .	197
7.6.6	Die Reihenfolge von Strukturdefinitionen . . . . .	198
7.6.7	Weitergereichte Argumente . . . . .	198
7.6.8	Verschachtelte Definitionen . . . . .	199
7.6.9	Unerwünschte Zwischenräume . . . . .	199
7.7	Zwei abschließende Beispiele . . . . .	200

<b>8</b>	<b>L<sup>A</sup>T<sub>E</sub>X-Steigerungen</b>	<b>203</b>
8.1	Behandlung von Teildokumenten . . . . .	203
8.1.1	Der Befehl <code>\input</code> . . . . .	203
8.1.2	Der Befehl <code>\include</code> . . . . .	204
8.1.3	Ein- und Ausgabe am Bildschirm . . . . .	206
8.1.4	T <sub>E</sub> X-Befehle . . . . .	208
8.1.5	Der <code>\special</code> -Befehl . . . . .	208
8.2	Textbezüge . . . . .	209
8.2.1	Querverweise . . . . .	209
8.2.2	Bezüge zum Literaturverzeichnis . . . . .	210
8.2.3	Indexregister . . . . .	212
8.2.4	Glossar . . . . .	214
8.3	MakeIndex – ein Stichwortprozessor . . . . .	214
8.3.1	Nutzungsvoraussetzungen für MakeIndex . . . . .	214
8.3.2	Die Syntax der Indexeinträge für MakeIndex . . . . .	215
8.3.3	Maskierung der MakeIndex-Sonderzeichen . . . . .	217
8.3.4	Die lexikalische Ordnung im Indexregister . . . . .	218
8.3.5	MakeIndex-Programmoptionen . . . . .	218
8.3.6	MakeIndex-Formatänderungsfiles . . . . .	219
8.3.7	Ein Demonstrationsbeispiel . . . . .	223
8.3.8	Einspaltiger Indexvorspann . . . . .	226
8.4	Die verschiedenen L <sup>A</sup> T <sub>E</sub> X-Files . . . . .	227
8.4.1	L <sup>A</sup> T <sub>E</sub> X-Ergebnisfiles . . . . .	227
8.4.2	L <sup>A</sup> T <sub>E</sub> X-Systemfiles . . . . .	230
8.4.3	L <sup>A</sup> T <sub>E</sub> X-Installationsfiles . . . . .	231
8.5	Weitere Zeichensatzbefehle aus L <sup>A</sup> T <sub>E</sub> X . . . . .	231
8.5.1	Die Grundidee des Zeichensatz-Auswahlverfahrens in L <sup>A</sup> T <sub>E</sub> X . . . . .	232
8.5.2	Vereinfachte Zeichensatzauswahl mit L <sup>A</sup> T <sub>E</sub> X . . . . .	235
8.5.3	L <sup>A</sup> T <sub>E</sub> X-Interfacebefehle . . . . .	237
<b>9</b>	<b>Fehlerbehandlung</b>	<b>239</b>
9.1	Grundstruktur der Fehlermeldungen . . . . .	239
9.1.1	T <sub>E</sub> X-Fehlermeldungen . . . . .	239
9.1.2	L <sup>A</sup> T <sub>E</sub> X-Fehlermeldungen . . . . .	242
9.1.3	Fehlermeldungen aus T <sub>E</sub> X-Makros . . . . .	245
9.2	Weitere Fehlerbeispiele . . . . .	247
9.2.1	Fehlerfortpflanzung . . . . .	247
9.2.2	Typische Fehler mit Folgewirkung . . . . .	249
9.2.3	Mathematische Fehlermeldungen . . . . .	251
9.2.4	Fehlermeldungen bei Mehrfiletexten . . . . .	253
9.3	Verzeichnis aller L <sup>A</sup> T <sub>E</sub> X-Fehlermeldungen . . . . .	253
9.3.1	Fehlermeldungen aus dem L <sup>A</sup> T <sub>E</sub> X-Kern . . . . .	254
9.3.2	Fehlermeldungen aus Klassenfiles und Ergänzungspaketen . . . . .	259
9.3.3	Fehlermeldungen bei der Zeichensatzauswahl . . . . .	260
9.3.4	L <sup>A</sup> T <sub>E</sub> X-Hilfswerkzeuge zur Fehlersuche . . . . .	262
9.4	Verzeichnis häufiger T <sub>E</sub> X-Fehlermeldungen . . . . .	263

9.5	Warnungen . . . . .	267
9.5.1	Allgemeine L <sup>A</sup> T <sub>E</sub> X-Warnungen . . . . .	268
9.5.2	L <sup>A</sup> T <sub>E</sub> X-Warnungen aus Klassenfiles und Ergänzungspaketen . . . . .	270
9.5.3	L <sup>A</sup> T <sub>E</sub> X-Zeichensatzwarnungen . . . . .	271
9.5.4	T <sub>E</sub> X-Warnungen . . . . .	271
9.6	Suche nach versteckten Fehlern . . . . .	272
9.7	MakeIndex-Fehlermeldungen . . . . .	273
9.7.1	Formatänderungsfehler . . . . .	274
9.7.2	Fehlermeldungen aus der Lesephase . . . . .	274
9.7.3	Warnungen aus der Schreibphase . . . . .	275
<b>A</b>	<b>Briefe</b> . . . . .	<b>277</b>
A.1	Die L <sup>A</sup> T <sub>E</sub> X-Bearbeitungsklasse letter . . . . .	277
A.2	Eine hauseigene letter-Bearbeitungsklasse . . . . .	281
A.3	Hinweise zur firmenspezifischen Anpassung . . . . .	284
<b>B</b>	<b>Literaturdatenbanken</b> . . . . .	<b>291</b>
B.1	Das BIBT <sub>E</sub> X-Programm . . . . .	291
B.2	Die Erstellung einer Literaturdatenbank . . . . .	293
B.2.1	Die verschiedenen Eingabetypen . . . . .	294
B.2.2	Felder . . . . .	295
B.2.3	Spezielle Feldformate . . . . .	296
B.2.4	Abkürzungen . . . . .	298
<b>C</b>	<b>Zeichensätze</b> . . . . .	<b>299</b>
C.1	Vorbemerkungen . . . . .	299
C.2	Klassifizierung der T <sub>E</sub> X-Grundzeichensätze . . . . .	300
C.3	Proportionalschriften . . . . .	302
C.3.1	Serifen-Schriften . . . . .	302
C.3.1.1	Die Zeichensatzgruppe ‘Roman’ . . . . .	302
C.3.1.2	Die Zeichensatzgruppe ‘Slanted’ . . . . .	304
C.3.1.3	Die Zeichensatzgruppe ‘Italic’ . . . . .	304
C.3.1.4	Die Zeichensatzgruppe ‘Bold Face’ (Fettdruck) . . . . .	305
C.3.2	Sans-Serifen-Schriften . . . . .	306
C.3.2.1	Die ‚aufrechten Sans-Serif‘-Zeichensätze . . . . .	307
C.3.2.2	Die ‚geneigten Sans-Serif‘-Zeichensätze . . . . .	308
C.3.2.3	Die ‚fetten Sans-Serif‘-Zeichensätze . . . . .	309
C.3.2.4	Der Zeichensatz cminch . . . . .	309
C.3.3	Zier- und Sonderschriften . . . . .	311
C.4	Fixschriften – Schreibmaschinenschriften . . . . .	312
C.4.1	Aufrechte Schreibmaschinenschriften . . . . .	312
C.4.2	Großschreibung . . . . .	313
C.4.3	Geneigte Schreibmaschinenschriften . . . . .	313
C.4.4	Mathematische Schreibmaschinenschrift . . . . .	313

C.5	Mathematik- und Symbolzeichensätze . . . . .	314
C.5.1	Mathematische Textzeichensätze . . . . .	314
C.5.2	Mathematische Symbole . . . . .	315
C.5.3	Variable Symbole . . . . .	316
C.5.4	Zusätzliche Zeichensätze . . . . .	316
C.5.4.1	Die $\LaTeX$ -lasy-Zeichensätze . . . . .	317
C.5.4.2	Zeichensätze zur Erzeugung von Bildern . . . . .	317
C.5.4.3	Logo-Zeichensätze . . . . .	317
C.5.4.4	PostScript-Zeichensätze . . . . .	317
C.6	Die Anordnung innerhalb der cm-Zeichensätze . . . . .	318
C.7	Erweiterte $\TeX$ -Zeichensätze . . . . .	323
C.7.1	Der Erweiterungsvorschlag von Cork . . . . .	323
C.7.2	Installation der ec-Schriften . . . . .	324
C.7.3	Aktivierung der ec-Schriften . . . . .	325
C.7.4	Das Ordnungsprinzip der ec-Schriften . . . . .	326
C.7.5	Die Namenskonventionen der ec-Schriften . . . . .	328
C.7.6	Die tc-Schriftergänzungen . . . . .	329
C.8	Die cm-Zeichensatzfiles . . . . .	332
C.8.1	Die Grundnamen der cm-Zeichensatzfiles . . . . .	332
C.8.2	Vergößerte Zeichensätze . . . . .	332
C.8.3	Pixel-Kodierung . . . . .	334
C.8.4	Gepackte Kodierung . . . . .	335
C.9	Anmerkungen zu METAFONT . . . . .	335
C.9.1	Die Nutzung von METAFONT . . . . .	336
C.9.2	METAFONT-Geräteanpassung . . . . .	337
<b>D</b>	<b><math>\LaTeX</math>-Ergänzungen</b>	<b>339</b>
D.1	Der deutsche $\TeX$ -Befehlszusatz . . . . .	339
D.1.1	Der Aufruf des german.sty-Files . . . . .	340
D.1.2	Die Umlaute und das $\beta$ . . . . .	340
D.1.3	Trennungshilfen . . . . .	341
D.1.4	Aufhebung von Ligaturen . . . . .	341
D.1.5	Deutsche Anführungszeichen . . . . .	342
D.1.6	Französische Anführungszeichen . . . . .	342
D.1.7	Sprachumschaltung . . . . .	343
D.1.8	Umschaltung auf das $\TeX$ -Original . . . . .	344
D.1.9	Besonderheiten aus (n)german.sty . . . . .	344
D.2	Einrichtung und Dokumentation von (n)german.sty . . . . .	346
D.2.1	Die Quellenfiles für (n)german.sty . . . . .	346
D.2.2	Installation von german.sty und ngerman.sty . . . . .	346
D.2.3	Strukturhinweise zu (n)german.sty . . . . .	348
D.2.4	Wirkungsunterschiede zwischen german.sty-Versionen . . . . .	351
D.2.5	Direkte Umlaut- und $\beta$ -Eingabe . . . . .	352
D.3	Weiterführende Literaturhinweise . . . . .	353

<b>E</b>	<b>Projektionsvorlagen</b>	<b>355</b>
E.1	Die Bearbeitungsklasse slides.cls . . . . .	355
E.1.1	Folienvortexte . . . . .	356
E.1.2	Folienvorlagen . . . . .	356
E.1.3	Wechselfolien . . . . .	357
E.1.4	Anmerkungen zu Projektionsvorlagen . . . . .	357
E.1.5	Ein vollständiges Folienbeispiel . . . . .	358
E.2	Schriftarten und Größen in slides . . . . .	360
E.2.1	Die slides-Schriftarten . . . . .	360
E.2.2	Die slides-Schriftgrößen . . . . .	361
E.3	Weitere slides-Bearbeitungsmöglichkeiten . . . . .	361
E.3.1	Seitenstilarten mit slides.cls . . . . .	361
E.3.2	Selektive Folienbearbeitung . . . . .	362
E.3.3	Anmerkungen zu SLIT <sub>E</sub> X . . . . .	363
E.3.4	Farbdruck mit L <sup>A</sup> T <sub>E</sub> X . . . . .	363
E.3.5	Positionierungsprobleme bei Wechselfolien . . . . .	365
<b>F</b>	<b>T<sub>E</sub>X-Installation und Beschaffung</b>	<b>369</b>
F.1	Das T <sub>E</sub> X-System im Überblick . . . . .	369
F.1.1	Ein T <sub>E</sub> X-Minimalsystem . . . . .	369
F.1.2	T <sub>E</sub> X-Programmaufrufe durch Befehlsdateien . . . . .	374
F.1.3	Systemgrenzen für mehrsprachige Formatfiles . . . . .	375
F.1.4	Das T <sub>E</sub> X-Filesystem . . . . .	376
F.1.5	Dateienstrukturierung mittels Umgebungsvariablen . . . . .	381
F.1.6	BIGT <sub>E</sub> X . . . . .	382
F.1.7	Ein METAFONT-Minimalsystem . . . . .	383
F.1.8	Drucker-Zeichensätze und Druckertreiber . . . . .	385
F.1.9	Die T <sub>E</sub> X-Quellenfiles . . . . .	386
F.2	Das L <sup>A</sup> T <sub>E</sub> X-System . . . . .	386
F.2.1	Das L <sup>A</sup> T <sub>E</sub> X-Grundsystem . . . . .	386
F.2.2	Das vollständige L <sup>A</sup> T <sub>E</sub> X-System . . . . .	389
F.2.3	L <sup>A</sup> T <sub>E</sub> X-Ergänzungen . . . . .	390
F.3	T <sub>E</sub> X-Installation auf PCs . . . . .	391
F.3.1	Ein T <sub>E</sub> X-System für DOS und OS/2 (emT <sub>E</sub> X) . . . . .	392
F.3.1.1	Strukturbeschreibung des emT <sub>E</sub> X-Pakets . . . . .	392
F.3.1.2	Installationsvorbereitung – unzip.exe und emxrsx.zip . . . . .	393
F.3.1.3	Installationshinweise . . . . .	394
F.3.1.4	Dokumentation und Benutzeroberflächen . . . . .	395
F.3.1.5	Zeichensatzfiles des emT <sub>E</sub> X-Pakets . . . . .	396
F.3.2	T <sub>E</sub> X unter 32 bit-WINDOWS-Systemen . . . . .	397
F.3.2.1	Evtl. Entpackungsvorbereitungen zur T <sub>E</sub> X-Installation . . . . .	397
F.3.2.2	MiK <sub>T</sub> E <sub>X</sub> . . . . .	398
F.3.2.3	Die WinEdt-Benutzeroberfläche . . . . .	398
F.3.2.4	Weitere empfohlene Programmergänzungen zu MiK <sub>T</sub> E <sub>X</sub> . . . . .	399
F.3.3	Alternative T <sub>E</sub> X-Systeme für DANTE-Mitglieder – pdfT <sub>E</sub> X u. a. . . . .	399
F.3.4	T <sub>E</sub> X unter LINUX . . . . .	401

F.4	T <sub>E</sub> X auf weiteren Individualrechnern . . . . .	402
F.4.1	T <sub>E</sub> X für ATARI-Rechner . . . . .	402
F.4.2	T <sub>E</sub> X für AMIGA-Rechner . . . . .	403
F.4.3	T <sub>E</sub> X für den Macintosh . . . . .	403
F.4.4	T <sub>E</sub> X auf Workstations . . . . .	405
F.5	Öffentliche Fileserver als Beschaffungsquellen . . . . .	408
F.5.1	Filetransfer mit dem ftp-Programm . . . . .	409
F.5.2	Fileanforderungen mittels E-Mail . . . . .	411
F.5.3	T <sub>E</sub> X-Anwendervereinigungen . . . . .	412
F.5.4	Die CD-ROM-Buchbeilage . . . . .	414
<b>Literaturverzeichnis</b>		<b>415</b>
<b>Befehlsindex</b>		<b>419</b>
	Kurzbeschreibung der L <sup>A</sup> T <sub>E</sub> X-Befehle . . . . .	419
	Zusammenfassende Tabellen und Diagramme . . . . .	483
	Verbotene T <sub>E</sub> X-Befehle . . . . .	491
<b>Stichwortverzeichnis</b>		<b>493</b>

## Tabellenverzeichnis

### Zeichenanordnung in Zeichensätzen

<b>Tabelle 1:</b> <code>cmr10</code> (Standard) . . . . .	318
<b>Tabelle 2:</b> <code>cmcsc10</code> (Kapitalchen) . . . . .	319
<b>Tabelle 3:</b> <code>cmti10</code> (Text-Italic) . . . . .	319
<b>Tabelle 4:</b> <code>cmtt10</code> (Schreibmaschine) . . . . .	320
<b>Tabelle 5:</b> <code>cmtex10</code> (Math.-Schreibmaschine) . . . . .	320
<b>Tabelle 6:</b> <code>cmmi10</code> (Math.-Italic) . . . . .	321
<b>Tabelle 7:</b> <code>cmsy10</code> (Mathematische Symbole) . . . . .	321
<b>Tabelle 8:</b> <code>cmex10</code> (Variable math. Symbole) . . . . .	328

### Zusammenfassungen

<b>Tabelle 1:</b> Schriftumschaltungen mit $\LaTeX 2_{\epsilon}$ . . . . .	483
<b>Tabelle 2:</b> Mathematische Schriftumschaltungen mit $\LaTeX 2_{\epsilon}$ . . . . .	463
<b>Tabelle 3:</b> Schriftenerklärungen mit $\LaTeX 2.09$ . . . . .	483
<b>Tabelle 4:</b> Schriftgrößen . . . . .	483
<b>Tabelle 5:</b> Maßeinheiten . . . . .	483
<b>Tabelle 6:</b> Akzente . . . . .	483
<b>Tabelle 7:</b> Sonderbuchstaben in Fremdsprachen . . . . .	484
<b>Tabelle 8:</b> Sonderzeichen . . . . .	484
<b>Tabelle 9:</b> Befehlszeichen . . . . .	484
<b>Tabelle 10:</b> Umgebungsnamen . . . . .	484
<b>Tabelle 11:</b> Griechische Buchstaben . . . . .	484
<b>Tabelle 12:</b> Binäre Operationssymbole . . . . .	484
<b>Tabelle 13:</b> Mathematische Vergleichssymbole . . . . .	485
<b>Tabelle 14:</b> Negierte Vergleichssymbole . . . . .	485
<b>Tabelle 15:</b> Pfeil- und Zeigersymbole . . . . .	485
<b>Tabelle 16:</b> Sonstige mathematische Symbole . . . . .	485
<b>Tabelle 17:</b> Mathematische Symbole in zwei Größen . . . . .	486
<b>Tabelle 18:</b> Funktionsnamen . . . . .	486
<b>Tabelle 19:</b> Mathematische Akzente . . . . .	486
<b>Tabelle 20:</b> Klammersymbole . . . . .	486
<b>Tabelle 21:</b> Die Grundnamen der $\TeX$ -Standardzeichensätze . . . . .	486
<b>Tabelle 22:</b> Die Grundnamen der $\LaTeX$ -Zusatzschriften . . . . .	487
<b>Tabelle 23:</b> Vergrößerungsstufen und Skalierungsfaktoren . . . . .	487
<b>Tabelle 24:</b> Beziehungen zwischen Schriften und Größenoptionen . . . . .	487

## Bildverzeichnis

<b>Diagramm 1:</b> Einspaltiges Seitenformat . . . . .	488
<b>Diagramm 2:</b> Zweispaltiges Seitenformat . . . . .	489
<b>Diagramm 3:</b> Listenformat der <code>list</code> -Umgebung . . . . .	490

# Kapitel 1

## Grundlagen

Textverarbeitung mit einem Rechner kann in vielfältiger Weise erfolgen. Eigenschaften und Leistungsfähigkeit sind hierbei weniger vom jeweiligen Rechnertyp als vielmehr vom verwendeten *Textverarbeitungsprogramm* bestimmt. Textverarbeitungsprogramme existieren in großer Zahl auf dem Rechnermarkt. Die meisten von ihnen sind auf bestimmte Hauptanwendungsfälle, z. B. die Büro- oder Geschäftskorrespondenz, zugeschnitten.

Alle Textverarbeitungsprogramme basieren auf einem von zwei ganz unterschiedlichen Grundkonzepten. Bei den sog. *Wortprozessoren* erscheint auf dem Bildschirm zunächst ein *Menü* mit den möglichen Bearbeitungseigenschaften. Diese müssen zunächst gewählt werden, meist einfach dadurch, dass man mit dem Cursor auf die angebotenen Eigenschaften fährt und diese dadurch auswählt. Nach der Festlegung der Bearbeitungseigenschaften wird der Text über die Tastatur eingegeben und erscheint auf dem Bildschirm genau in der Weise, wie er auch beim Druck ausgegeben wird. Der Anwender kann damit sofort bei der Eingabe feststellen, ob der bearbeitete Text seinen Vorstellungen entsprechend erzeugt wird. Ist dies nicht der Fall, so kann mit speziellen Funktionstasten eine Korrektur durch den Anwender vorgenommen und das Ergebnis unmittelbar überprüft werden.

Das andere Konzept beruht auf einem zweistufigen Vorgang: Der Texteingabe und ggf. Korrektur mit einem *Editor* des Rechners folgt anschließend die Bearbeitung durch ein sog. *Formatierungsprogramm*.

Erst danach wird der bearbeitete Text auf einem Ausgabegerät, das ein Drucker oder ein hochauflösender Bildschirm sein kann, ausgegeben. Ist der Anwender mit dem Ergebnis nicht zufrieden, so muss der im Rechner gespeicherte Text geändert oder korrigiert und dann erneut mit dem Formatierungsprogramm bearbeitet werden.

Auf den ersten Blick erscheint das erste Konzept als das ideale. Die meisten Textverarbeitungsprogramme sind auch hierauf aufgebaut. Sie ersetzen mehr und mehr die herkömmliche Schreibmaschine, mit der sie in Konkurrenz stehen und der sie weit überlegen sind. Formatierungsprogramme stehen weniger in Konkurrenz zur Schreibmaschine als vielmehr zum Druckereiwesen. Hier haben sie innerhalb weniger Jahre den traditionellen Beruf des Setzers praktisch zum Verschwinden gebracht.

Beide Konzepte haben ihre spezifische Bedeutung und Leistungsfähigkeit. Soll das Ergebnis der Textverarbeitung Buchdruckqualität erreichen, insbesondere bei wissenschaftlichem Text mit komplexen mathematischen Formeln, so wird ein geeignetes Formatierungsprogramm erforderlich sein. Formatierungsprogramme entfalten ihre besondere Leistung auch

dann, wenn derselbe Text in unterschiedlicher Weise bearbeitet werden soll. Ist z. B. ein Text einmal einspaltig formatiert und ausgedruckt worden, so kann mit der Änderung eines einzigen Befehls erreicht werden, dass derselbe Text bei einer erneuten Bearbeitung nunmehr pro Seite zweispaltig formatiert wird und damit ein vollständig anderes Aussehen in Bezug auf Zeilen- und Seitenumbruch erhält.

## 1.1 $\text{T}_{\text{E}}\text{X}$ und $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Das wohl leistungsfähigste Formatierungsprogramm zur Erzeugung wissenschaftlich-technischer Texte in Buchdruckqualität stammt von DONALD E. KNUTH [10]. Das Programm hat den Namen  $\text{T}_{\text{E}}\text{X}$  (gesprochen Tech), was die griechische Schreibweise in Großbuchstaben von  $\tau\epsilon\chi$  widerspiegeln soll. Neben  $\text{T}_{\text{E}}\text{X}$  wurde vom selben Autor ein weiteres Programm mit dem Namen METAFONT entwickelt, das zur Erzeugung von Zeichensätzen dient. Standardmäßig enthält das  $\text{T}_{\text{E}}\text{X}$ -Programmpaket 75 Zeichensätze für verschiedene Entwurfsgrößen, wobei jeder dieser Zeichensätze zusätzlich in bis zu acht verschiedenen Vergrößerungsstufen bereitsteht. Alle diese Zeichensätze wurden mit dem Programm METAFONT erzeugt. Bei verschiedenen Anwendern wurden weitere Zeichensätze erzeugt, z. B. kyrillische und sogar japanische Zeichensätze, mit denen Texte auch in diesen Schriften in Buchdruckqualität ausgegeben werden.

Die enorme Leistungsfähigkeit von  $\text{T}_{\text{E}}\text{X}$  hat ihren Preis: Die Anwendung und besonders die Ausschöpfung der Möglichkeiten setzt erhebliche Erfahrung mit Programmiertechniken voraus. Die Anwendung bleibt daher meist auf Profis aus dem Programmierbereich beschränkt. Aus diesem Grund wurde von dem amerikanischen Computerwissenschaftler LESLIE LAMPORT [1] das Programmpaket  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  entwickelt, das seinerseits auf  $\text{T}_{\text{E}}\text{X}$  zurückgreift, aber zwischen  $\text{T}_{\text{E}}\text{X}$  und dem Anwender eine sehr viel benutzerfreundlichere Ebene schafft. Mit  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  wird auch der Anwender ohne Programmierkenntnisse in die Lage versetzt, die Möglichkeiten von  $\text{T}_{\text{E}}\text{X}$  weitgehend auszuschöpfen und bereits nach kurzer Zeit eine Vielzahl von Textausgaben in Buchdruckqualität erzeugen zu können. Dies gilt ganz besonders auch für die Erzeugung komplexer Tabellen und mathematischer Formeln.

Dies setzt voraus, dass der Anfänger die Standardformatierungen von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  akzeptiert und nicht eigenwillige Sonderwünsche an den Anfang setzt. LESLIE LAMPORTS Philosophie bei der Entwicklung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  war, den Anwender von eigenen Formatierungsüberlegungen freizustellen, und dieses Angebot sollte er nutzen. Natürlich gestattet  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  auch, individuelle Anwenderwünsche zufriedenzustellen, und es ist der Zweck dieses Buches, dem Anwender alle Möglichkeiten von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zu erschließen. Diese entwickeln sich mit zunehmender Praxis und verlangen, dass der Hauptteil dieses Buches (Kap. 1–9) sowie die Anhänge A und D einmal durchgearbeitet werden. Danach kann es als Nachschlagewerk dienen, wofür sich insbesondere der Befehlsindex am Ende des Buches eignet. Dieser enthält eine Kurzbeschreibung aller Befehle mit Verweisen auf deren Vorstellung im Hauptteil.

Individuelle Formatierungen sollten aber die Ausnahme bleiben und nicht die Regel sein. Hinter den angebotenen Standardformatierungen verbirgt sich Fachwissen von professionellen Druckern und Grafikern, mit denen der Normalanwender nicht in Konkurrenz treten sollte. Anwender ohne fundierte Fachkenntnisse aus dem Druck- und Satzwesen werden mit den Standardformaten stets ein auch aus professioneller Sicht akzeptables Ergebnis erzeugen, was leider auf viele individuell gestaltete Formate nicht gleichermaßen zutrifft.

## 1.2 L<sup>A</sup>T<sub>E</sub>X-Entwicklungsgeschichte

Die Entwicklungsarbeit für L<sup>A</sup>T<sub>E</sub>X als komfortables und gleichzeitig einfach zu handhabendes T<sub>E</sub>X-Zugangspaket begann unmittelbar nach der Verfügbarkeit von T<sub>E</sub>X 82 und wurde mit der Bereitstellung einer stabilen L<sup>A</sup>T<sub>E</sub>X-Version mit der etwas eigenartig anmutenden Versionsnummer 2.09 Mitte 1985 abgeschlossen. Vom Programmautor LESLIE LAMPORT erschien im gleichen Jahr auch die Nutzungsbeschreibung [1]. Versionsnummer und Nutzungsbeschreibung blieben nahezu 9 Jahre bis Mitte 1994 unverändert, was die Stabilität von L<sup>A</sup>T<sub>E</sub>X 2.09 als Standard-Zugangspaket zu T<sub>E</sub>X demonstriert.

Die Beibehaltung der Versionsnummer 2.09 bedeutet aber nicht, dass L<sup>A</sup>T<sub>E</sub>X-Pakete mit dieser Kennzeichnung identisch sind. Unter der gleichen Versionsnummer erschienen, durch ihre Erstellungsdaten gekennzeichnet, unterschiedliche Realisierungen. So wurde z. B. L<sup>A</sup>T<sub>E</sub>X, das von seinem Autor primär für die Bearbeitung englischer Texte eingerichtet wurde, internationalisiert. Trägt ein L<sup>A</sup>T<sub>E</sub>X-Paket 2.09 ein Erstellungsdatum ab dem 1. Dezember 1991, so handelt es sich um die internationale Version, die zusammen mit sprachspezifischen Stiloptionen, wie z. B. *german*, dann automatisch erscheinende Begriffe wie „Chapter“, „Contents“ u. a. in Deutsch als „Kapitel“, „Inhaltsverzeichnis“ usw. ausgibt.

Die unterschiedlichen Varianten mit verschiedenen Erstellungsdaten haben den sog. L<sup>A</sup>T<sub>E</sub>X-Kern, von Fehlerkorrekturen abgesehen, jedoch nie verändert. Die wesentlichen Änderungen erfolgten in den zusätzlichen Stilfiles. Andere Ergänzungen erforderten dagegen die Erstellung neuer oder weiterer Formatfiles (s. Anh. F.1.1). So war es bis zur Bereitstellung von T<sub>E</sub>X 3.0 stets erforderlich, bei mehrsprachigen Anwendungen für jede Sprache je ein L<sup>A</sup>T<sub>E</sub>X-Formatfile mit dem jeweils sprachspezifischen Trennmusterfile zu erstellen. Ab T<sub>E</sub>X 3.0 können im Prinzip bis zu 256 verschiedene Trennmusterfiles in einem Formatfile zusammengefasst werden, wobei das jeweils zu aktivierende Trennmuster mit einem geeigneten *Sprachschalter* eingestellt wird.

Die Schriftauswahl oder Schriftänderung erfolgt in L<sup>A</sup>T<sub>E</sub>X 2.09 mit sog. *Zeichensatzauswahlbefehlen*, denen explizite Zeichensätze zugeordnet sind. Ein viel flexibleres und dem Anwender sehr viel natürlicher erscheinendes Zeichensatz-Auswahlverfahren stammt von FRANK MITTELBACH und RAINER M. SCHÖPF. Es erschien unter der Bezeichnung NFSS („New Font Selection Scheme“). Sein Einsatz verlangt unter L<sup>A</sup>T<sub>E</sub>X 2.09 ebenfalls die Erzeugung eines weiteren oder ersetzenden Formatfiles.

Die Verwendung verschiedener Formatfiles ist unproblematisch, solange die Bearbeitung nur lokal, also beim Anwender, der diese Formatfiles erzeugte, erfolgt. Sollen dagegen L<sup>A</sup>T<sub>E</sub>X-Textfiles im Originalzustand per E-Mail oder als Diskettenkopien versandt und an anderer Stelle mit L<sup>A</sup>T<sub>E</sub>X bearbeitet werden, so kann dies wegen verschiedener Formatfiles Kompatibilitätsprobleme hervorrufen.

Damit geht eine Grundeigenschaft von L<sup>A</sup>T<sub>E</sub>X verloren: L<sup>A</sup>T<sub>E</sub>X sollte auf allen Rechnern und unter allen Betriebssystemen identische Ergebnisse liefern. Auf Initiative von FRANK MITTELBACH und RAINER M. SCHÖPF entstand die internationale Entwicklungsgruppe für das „L<sup>A</sup>T<sub>E</sub>X 3-Projekt“, an dem sich auch LESLIE LAMPORT beratend beteiligt. Zu den Zielen des L<sup>A</sup>T<sub>E</sub>X 3-Projekts gehört es, einen L<sup>A</sup>T<sub>E</sub>X-Kern zu entwickeln, der für alle zukünftigen Ergänzungen, wie auch immer sie geartet sein mögen, nur ein Formatfile vorhält. Auch das im letzten Absatz angesprochene Zeichensatz-Auswahlverfahren NFSS soll in seiner Wirkung Bestandteil des L<sup>A</sup>T<sub>E</sub>X-Kerns werden. Auf weitere Zielsetzungen für das L<sup>A</sup>T<sub>E</sub>X 3-Projekt wird, mit Ausnahme der zwingenden Forderung der Kompatibilität zur bisherigen L<sup>A</sup>T<sub>E</sub>X 2.09-Version, hier nicht eingegangen.

Im Dezember 1993 stellte der Arbeitskreis für das  $\LaTeX$  3-Projekt eine neue  $\LaTeX$ -Version mit der Bezeichnung  $\LaTeX 2_\epsilon$  als sog. Testversion auf den internationalen  $\TeX$ -Servern bereit. Nach einer halbjährigen Erprobung und der Behebung einiger Fehler und Schwächen sowie der Übernahme weiterer Nutzer-Anregungen wurde  $\LaTeX 2_\epsilon$  im Juni 1994 zur offiziellen  $\LaTeX$ -Version erklärt. Die frühere  $\LaTeX$ -Version 2.09 wird bis auf weiteres auf den internationalen  $\TeX$ -Fileservern vorgehalten, dort aber explizit als `latex209` gekennzeichnet, während unter dem Namen `latex` nunmehr die neue Version ohne explizite Versionskennung vorgehalten wird.

Als Konsequenz erschien 1994 das Buch von LESLIE LAMPORT [1] in zweiter Auflage, mit der  $\LaTeX$  in der Version  $2_\epsilon$  dargestellt wird, was gleichermaßen nun auch ab der 2. Auflage von Band 1 dieser Buchserie gilt.  $\LaTeX 2_\epsilon$  kennt einen Kompatibilitätsmodus zu  $\LaTeX 2.09$ , so dass Eingabetexte, die ursprünglich zur Bearbeitung mit  $\LaTeX 2.09$  vorgesehen waren, auch mit der neuen  $\LaTeX 2_\epsilon$ -Version wie mit der früheren  $\LaTeX 2.09$ -Version bearbeitet werden.

Bei der Vorstellung und Wirkungsbeschreibung der Befehle wurden in der 2. Auflage dieses Buches diejenigen Befehle, die nur mit  $\LaTeX 2_\epsilon$  verfügbar sind oder die mit  $\LaTeX 2_\epsilon$  eine Erweiterung oder geänderte Syntax gegenüber gleichnamigen Befehlen aus  $\LaTeX 2.09$  erfahren haben, mit einem vorangestellten  $\boxed{2_\epsilon}$  markiert. Umgekehrt wurden Befehle aus  $\LaTeX 2.09$ , die nur im Kompatibilitätsmodus oder mit einer alten  $\LaTeX 2.09$ -Version zu verwenden sind, mit einem vorangestellten  $\boxed{2.09}$  markiert. Die allermeisten Befehlsvorstellungen blieben ohne solche Markierungen, da ihre Wirkung unabhängig von der benutzten  $\LaTeX$ -Version ist.

Inzwischen gilt  $\LaTeX 2.09$  als überholt, wenn nicht gar als obsolet. Es steht auf den  $\TeX$ -Fileservern zwar immer noch zur Verfügung, doch es unterliegt dort keinerlei Wartung mehr. Aus diesem Grund werden bei der Vorstellung und Wirkungsbeschreibung der Befehle ab der 3. Auflage dieses Buches nur noch die Eigenschaften von  $\LaTeX 2_\epsilon$  vorgestellt, womit der Bedarf einer expliziten Kennzeichnung ihrer Versionsherkunft entfällt. Lediglich bei der alphabetischen Auflistung und Kurzbeschreibung aller  $\LaTeX$ -Befehle im Befehlsindex dieses Buches erfolgt nach wie vor eine solche Herkunftskennzeichnung, damit der Anwender bei Übernahme alter  $\LaTeX$ -Eingabetexte, die im 2.09-Kompatibilitätsmodus von  $\LaTeX$  bearbeitet werden sollen, die Bearbeitungseigenheiten dieses Modus nachschlagen kann.

### 1.3 Erläuterungen zum vorliegenden Text

Dieses Buch richtet sich an  $\LaTeX$ -Anwender, die keine oder nur geringe Kenntnisse im Umgang mit Rechnern haben. Es basiert auf mehreren Kursen, die ich an meiner Arbeitsstätte für die dort beschäftigten Schreibkräfte und Sekretärinnen gehalten habe, wo inzwischen die meiste Textverarbeitung mittels  $\LaTeX$  erfolgt.

Das Buch ist eine Mischung aus Lehrbuch und Nachschlagewerk. Ich hoffe, dass es mit Unterstützung eines fachkundigen Betreuers für den Schreibdienst beide Zwecke erfüllt. Es enthält keine Informationen über rechner- oder systemspezifische Maßnahmen wie das *Einloggen*, den *Aufruf des Editors*, die *Handhabung des Editors*.

Wiederholungen im Text, besonders in der ersten Hälfte, sind von mir gewollt, da so dem Leser nicht zugemutet wird, nach einer kurzen Definition eines Begriffs viele Seiten später, insbesondere als Anfänger, diesen Begriff voll zu beherrschen. Jedoch sollte sich der Benutzer von Anbeginn ein Verständnis für die in 2.1–2.4 vorgestellten Grundbegriffe verschaffen.

Ich habe mich bemüht, *Computerslang* zu vermeiden, auch wenn mir das nicht vollständig gelungen ist. Ich hoffe, dass Begriffe wie „File“, „Datei“ oder „Editor“ auch bei den Anwendern im Schreibdienst inzwischen geläufig sind und keiner weiteren Erläuterung bedürfen. Abkürzungen wie i. Allg. (im Allgemeinen), d. h. (das heißt), u. a. (und andere), u. ä. (und ähnliche), bzw. (beziehungsweise), ggf. (gegebenenfalls), z. B. (zum Beispiel), usw. (und so weiter) sollten im Textzusammenhang verständlich sein.

Hinweise auf externe Literatur erfolgen in eckigen Klammern, wie [4b] oder [9]. Die Angaben innerhalb der eckigen Klammern beziehen sich auf die entsprechenden Markierungen im Literaturverzeichnis dieses Buches.

Bei der Beschreibung der Syntax der Befehle wird Schreibmaschinenschrift für die Teile des Befehls verwendet, die genauso, wie sie angegeben sind, einzugeben sind. *Kursivschrift* wird für die Teile des Befehls verwendet, die verschiedene Werte oder den zu verarbeitenden Text enthalten.

```
\begin{tabular}{sp_form} Zeilen \end{tabular}
```

stellt den Befehl zur Erzeugung einer Tabelle dar. Die Teile in Schreibmaschinenschrift sind zwingend. *sp\_form* steht für eine wählbare Form der Spaltenformatierung. Welche Werte bzw. Kombinationen hierfür möglich und erlaubt sind, wird bei der Beschreibung dieses Befehls im Einzelnen angegeben. *Zeilen* steht für die einzelnen Zeileneintragen der Tabelle und ist damit Teil des Textes.

Im laufenden Text tritt häufig der Begriff der „Return-Taste“ auf, weil diese bei den internationalen Terminaltastaturen meistens den Aufdruck „Return“ trägt. Bei PC-Tastaturen wird sie dagegen mit dem Wort „Enter“ und/oder dem Symbol  gekennzeichnet. Der Leser möge bitte den verwendeten Begriff der Return-Taste stets in die bei ihm vorhandene Tastenkennung übersetzen. Formulierungen wie „eingegebener Text, *gefolgt* von der Return-Taste“ oder „. . . , *abgeschlossen* mit der Return-Taste“ sind zwar ungenau, aber nicht missverständlich. Gemeint ist an solchen Stellen natürlich die *Betätigung* dieser Taste.

Textteile in kleinerer Schrift, wie der nachfolgende Abschnitt 1.6 über die  $\LaTeX$ -Bearbeitungsmodi, enthalten Informationen, die für eine vertiefte Kenntnis von Nutzen sind. Sie können zu Beginn übersprungen werden. Diese Kenntnis wird dann erforderlich, wenn individuelle Gestaltungswünsche die  $\LaTeX$ -Standardformatierungen ergänzen sollen. Die mit einem hochgestellten TLL gekennzeichneten Textstellen sind, mit freundlicher Genehmigung durch den Autor, dem Buch LESLIE LAMPORTS [1] entnommen.

## 1.4 Text und Befehle

Jeder Text besteht aus *Zeichen*, die zu *Wörtern* zusammengefügt sind. Die *Wörter* bilden *Sätze* und diese wiederum *Absätze*. Absätze können zu größeren Einheiten wie *Abschnitten* und *Kapiteln* zusammengefügt werden.

Wörter bestehen aus einem oder mehreren *Zeichen*, die durch *Leerzeichen* (*Blanks* = *Leertaste*) oder die *Return-Taste* (= *Zeilenwechsel*) getrennt sind.  $\TeX$  interpretiert Leerzeichen und Zeilenwechsel als Wortende. Dabei ist es gleichgültig, ob ein oder mehrere Leerzeichen zwischen den Wörtern auftreten. Der Wortabstand wird hierdurch nicht beeinflusst.

Absätze werden durch eine oder mehrere *Leerzeilen* voneinander getrennt. Der Abstand zwischen den Absätzen wird hierdurch nicht beeinflusst.  $\TeX$  behandelt die Wörter eines Absatzes als eine lange Kette von Wörtern, zwischen denen ein Wortabstand so gewählt wird, dass innerhalb eines Absatzes die Wortabstände möglichst gleich sind und die einzelnen

Zeilen links- und rechtsbündig abschließen. Der Zeilenumbruch erfolgt also unabhängig von der Texteingabe *automatisch*.

Der Zeilenabstand hängt von der gewählten Schriftgröße ab. Absätze werden durch Einrücken der ersten Zeile und/oder einen vergrößerten Zeilenabstand zwischen den Absätzen gekennzeichnet. Die Absatzabstände sind wie die Wortabstände leicht variabel.  $\TeX$  bzw.  $\LaTeX$  wählt sie so, dass der Text einer vollen Seite jeweils den gleichen oberen und unteren Rand hat, wobei der obere und untere Rand innerhalb des Dokuments unterschiedlich gewählt sein kann. Auch der Seitenumbruch erfolgt wie der Zeilenumbruch automatisch.

Im einfachsten Fall besteht ein Textfile nur aus dem so eingegebenen Text. Bei einer Behandlung mit  $\TeX$  wird dieser Text mit einer Standardbreite und Standardseitenhöhe formatiert, d. h., Zeilen-, Absatz- und Seitenumbruch erfolgen wie bei einem gesetzten Text nach allen Seiten bündig.

Jedes  $\LaTeX$ -Dokument besteht im Allgemeinen aber aus *Text*, der zu verarbeiten ist, und *Befehlen*, mit denen gesagt wird, wie der Text zu bearbeiten ist. Dies macht es notwendig, zwischen Text und Befehlen zu unterscheiden. Befehle bestehen entweder aus einigen einzelnen Sonderzeichen, die nicht als Textzeichen Verwendung finden, oder aus Wörtern, denen ein bestimmtes Sonderzeichen, nämlich der Rückstrich  $\backslash$  (engl. backslash), unmittelbar vorangesetzt ist.

## 1.5 Grundstruktur eines $\LaTeX$ -Files

Jedes  $\LaTeX$ -File besteht aus dem *Vorspann* (*preamble*) und dem *Textteil* (*body*).

Der Vorspann besteht ausschließlich aus Befehlen, mit denen die globale Bearbeitungsstruktur des nachfolgenden Textes festgelegt wird, also z. B. die Angabe des Papierformats, die Wahl der Textbreite und -höhe, die Gestaltung der Ausgabeseiten in Bezug auf ihre Nummerierung und die Erzeugung von automatischen Seitenköpfen oder -füßen. Der Vorspann muss mindestens aus dem Befehl `\documentclass` bestehen, mit dem die globale Bearbeitungsklasse des Dokuments festgelegt wird. Dies ist im Allgemeinen auch der erste Befehl des Vorspanns.

Wenn keine weiteren Befehle im Vorspann aufgeführt werden, wählt  $\LaTeX$  bestimmte Standardwerte für die Zeilenbreite, die Ränder, die Absatzabstände, die Seitenbreite und -länge und vieles mehr. Da diese in der Originalversion auf amerikanische Verhältnisse zugeschnitten sind, sollte eine  $\LaTeX$ -Ergänzung für deutsche Texte zur Verfügung stehen. Diese wird durch den Nutzungsbefehl `\usepackage{german,a4}` nach dem `\documentclass`-Befehl aktiviert, womit die Abmessungen für das Papierformat DIN A4 eingestellt werden und die Eingabe der Umlaute und des  $\beta$  einfacher als beim Original erfolgt sowie sonstige deutsche Besonderheiten berücksichtigt werden. Diese Ergänzungen werden durch zwei sog. Ergänzungspakete `german.sty`<sup>1</sup> und `a4.sty`<sup>2</sup> realisiert.

<sup>1</sup>Das Ergänzungspaket `german.sty` wird ausführlich in Anhang D.1 beschrieben.

<sup>2</sup> $\LaTeX 2_{\epsilon}$  kennt für den `\documentclass`-Befehl die Optionsangabe `a4paper` (s. 3.1.2) zur Einstellung des Papierformats DIN A4. Das damit erzielte Bearbeitungsergebnis entspricht jedoch bezüglich der Textbreite und -höhe kaum den Vorstellungen, die deutschsprachige Anwender mit dem DIN-A4-Papierformat verknüpfen. Mit der Bereitstellung eines eigenen Ergänzungspakets `a4.sty` kann eine gezieltere Anpassung erreicht werden.

Übung 2.3 aus Abschnitt 2.7 auf S. 23 enthält einen Vorschlag zur Erstellung eines eigenen Ergänzungspakets `a4.sty`, falls ein solches beim Anwender noch nicht existiert. Ebenso enthält die Übung 2.1 aus dem gleichen Abschnitt eine sehr vereinfachte und sicher nur vorläufige Alternative zum Ergänzungspaket `german.sty`, falls auch dieses beim Anwender bisher fehlt.

Der Vorspann endet mit dem Befehl `\begin{document}`. Alles, was nach diesem Befehl folgt, wird als *Textteil* interpretiert. Der Textteil besteht aus dem eigentlichen Text, vermischt mit weiteren Befehlen. Diese Befehle haben im Gegensatz zu denen im Vorspann nur lokale Wirkung, d. h., sie wirken im Allgemeinen nur auf Teile des laufenden Textes, wie zum Beispiel *Einrückungen, Behandlung von Text als Formeln, vorübergehende Umschaltung der Schriftart* u. a.

Der Textteil endet mit dem Befehl `\end{document}`. Dies ist im Allgemeinen auch das Ende des Files. Die allgemeine Syntax eines L<sup>A</sup>T<sub>E</sub>X-Files lautet damit:

```
\documentclass [Optionen] {Klasse}
  Evtl. \usepackage{erg_paket}-Nutzungsbefehle zur Aktivierung von Ergän-
  zungspaketen sowie
  evtl. weitere global wirkende Befehle und Erklärungen
\begin{document}
  Text, evtl. vermischt mit weiteren lokal wirkenden Befehlen
\end{document}
  Darauffolgende Fileinträge sind zwar erlaubt, sie haben für die LATEX-
  Bearbeitung aber keine Bedeutung und bleiben ohne jede Wirkung.
```

Welche Angaben für *Optionen* und *Klasse* im `\documentclass`-Befehl möglich und erlaubt sind, wird in 3.1 dargestellt. Mit `a4paper` für *Optionen* im `\documentclass`-Befehl ist eine alternative Papierformatvorgabe möglich. Bezüglich ihres Ergebnisses verweise ich auf Fußnote 2 auf der vorangegangenen Seite.

In einem Rechenzentrum wird die Information über die verschiedenen angepassten L<sup>A</sup>T<sub>E</sub>X-Versionen und/oder vorhandenen Ergänzungspakete im Allgemeinen durch einen sog. *Local Guide* beschrieben. Dieser sollte auch alle Angaben über die Art der Befehlsaufrufe für die L<sup>A</sup>T<sub>E</sub>X-Bearbeitung sowie der verfügbaren Ausgabegeräte wie Drucker, Mikrofilm, Grafikstationen u. ä. und deren Aktivierung enthalten. Die Gerätetreiber kennen ggf. Optionen, mit denen der Ausdruck in verschiedenen diskreten Vergrößerungen erfolgen kann.<sup>3</sup>

## 1.6 L<sup>A</sup>T<sub>E</sub>X-Bearbeitungsmodi

Bei der Bearbeitung von Texten befindet sich L<sup>A</sup>T<sub>E</sub>X stets in einem von drei Modi:

1. Absatzmodus – auch Paragraph Mode genannt
2. Mathematischer Modus – oder mathematischer Mode
3. LR-Modus – bzw. LR-Mode

Der *Absatzmodus* (engl. paragraph mode) ist der normale Bearbeitungsmodus. In ihm betrachtet L<sup>A</sup>T<sub>E</sub>X die Texteingabe als eine Sequenz von Wörtern und Sätzen, die in Zeilen, Absätze und Seiten (automatisch) umbrochen werden.

L<sup>A</sup>T<sub>E</sub>X schaltet in den *mathematischen Modus*, wenn durch bestimmte Befehle gesagt wird, dass der folgende Text eine *Formel* darstellt. Innerhalb des mathematischen Modus bleiben Leerzeichen

---

<sup>3</sup>PC-Anwender stehen als Einsteiger vor der Schwierigkeit, das T<sub>E</sub>X- und L<sup>A</sup>T<sub>E</sub>X-System zu installieren und die Teilkomponenten auf die *richtigen* Unterverzeichnisse aufzuteilen, wobei die erwähnten Anpassungen für die deutsche Textbearbeitung evtl. selbst vorzunehmen sind. Die beigefügten Installationshinweise sind, je nach Ausgangsquelle, unterschiedlich informativ oder verständlich. Ich hoffe, mit dem Anhang F dieses Buches PC-Anwendern behilflich zu sein, wenn die Installationshilfen und die Dokumentation des Lieferanten sich als beschwerlich oder nicht ausreichend erweisen.

unberücksichtigt. Der Text  $i s$  oder  $i \quad s$  wird als das Produkt von  $i$  und  $s$  interpretiert und erscheint als  $i s$ .  $\LaTeX$  schaltet zurück in den Absatzmodus, wenn durch entsprechende Befehle mitgeteilt wird, dass der vorangegangene Text als Formel beendet ist.

Der *LR-Modus* ist dem Absatzmodus ähnlich:  $\LaTeX$  behandelt den Eingabetext von links nach rechts als eine Kette von Wörtern, *zwischen denen kein Zeilenumbruch stattfinden kann*. In diesem Modus befindet sich  $\LaTeX$  z. B., wenn innerhalb von Formeln Text eingebettet ist oder mit einem speziellen Befehl, wie `\mbox{Teiltext}`, erreicht werden soll, dass innerhalb von *Teiltext* kein Zeilenumbruch erlaubt ist.

Die Unterscheidung und Kenntnis der Bearbeitungsmodi ist darum wichtig, weil einige Befehle entweder nur in bestimmten Modi erlaubt sind oder ihre Wirkung für die verschiedenen Bearbeitungsmodi unterschiedlich ist.

Soweit bestimmte Bearbeitungseigenschaften für Absatz- und LR-Modus gleich sind, wird im Folgenden auch vom Textmodus oder von Textmodi gesprochen. Diese stehen gemeinsam häufig im Gegensatz zum mathematischen Bearbeitungsmodus.

## 1.7 Die Erzeugung eines $\LaTeX$ -Dokuments

Die Erzeugung eines  $\LaTeX$ -Dokuments von der Texteingabe bis zur Druckausgabe ist ein dreistufiger Vorgang. Zunächst wird mit dem Editor des Rechners ein Textfile erzeugt (oder korrigiert). Das Textfile besteht aus dem eigentlichen Text, vermischt mit  $\LaTeX$ -Befehlen.

Der Name des Textfiles sollte den Anhang `.tex` enthalten. Die Syntax für den Filenamen des Textfiles ist dann *name.tex*. Das Betriebssystem des Rechners schreibt meistens weitere Bedingungen für die Wahl von Filenamen vor, wie z. B. die maximale Namenslänge oder das Verbot von Sonderzeichen in Filenamen. Der gewählte Name einschließlich der Endung `.tex` muss innerhalb der maximalen Namenslänge liegen. Ist diese z. B. 10, so wäre *muster.tex* ein erlaubter Name, nicht dagegen *entwurf.tex*, da Letzterer aus 11 Zeichen besteht.

Das Textfile muss sodann durch das  $\LaTeX$ -Programm bearbeitet werden. Der Aufruf zum Ablauf des  $\LaTeX$ -Programms ist systemabhängig. Bei mir lautet dieser Befehl einfach `latex`, gefolgt von dem Filenamen des Textfiles, jedoch *ohne* den Anhang `.tex`.  $\LaTeX$  kann auch Filenamen mit einem anderen Anhang bearbeiten. Dieser muss dann aber beim Bearbeitungsaufufr explizit angegeben werden.

Wurde als Name für das Textfile *muster.tex* bzw. *muster.ltx* gewählt, so lautet bei mir der Aufruf für die  $\LaTeX$ -Bearbeitung

```
latex muster      bzw.    latex muster.ltx
```

Während der Bearbeitung erscheinen auf dem Bildschirm die einzelnen Seitennummern der Bearbeitung, ggf. vermischt mit Warnungen und Fehlermeldungen. Der Interpretation von Fehlermeldungen und ihrer Behandlung ist das Kapitel 9 gewidmet. Nach der Bearbeitung des Textfiles durch  $\LaTeX$  ist ein weiteres File entstanden, und zwar mit dem gewählten Grundnamen und dem Anhang `.dvi`. Für das obige Beispiel wäre das *muster.dvi*.

Dieses DVI-File (*device independent*) enthält den *formatierten* Text sowie die Information über die benötigten Zeichensätze, jedoch in einer von dem verwendeten Drucker unabhängigen Form. Ein solches geräteunabhängiges Ausgabefile wird *Metafile* genannt.

Das DVI-Metafile muss schließlich durch ein *druckerspezifisches* Programm, einen sog. *Druckertreiber*, behandelt werden, um auf dem vorhandenen Drucker ausgegeben werden zu können. Der Befehlsaufruf für den Druckertreiber muss dem *Local Guide* oder dem

Treiberhandbuch entnommen werden. Bei dem verbreiteten PostScript-Treiber von TOMAS ROKICKI lautet er

```
dvips file_grundname z.B. dvips muster
```

womit gleichzeitig die automatische Ausgabe auf dem PostScript-Standarddrucker verknüpft ist. Der Aufruf mit der zusätzlichen Optionsangabe `-o`, also `dvips file_grundname -o`, führt dagegen zur Ausgabe eines Files mit dem angegebenen Grundnamen und dem Anhang `.ps`, das seinerseits dann mit dem Druckbefehl des jeweiligen Rechners ausgedruckt werden kann. Alle mir bekannten Druckertreiber verhalten sich bezüglich ihrer Aufrufe ähnlich. Sie unterscheiden sich lediglich durch ihren Aufrufnamen und die Angabe für den Namensanhang beim Druckbefehl, z. B. als `.bit` bei dem von mir geschriebenen Treiber `dvihp` für den HP-Laserjet.

Die meisten Druckertreiber kennen eine Vielzahl von Aufrufoptionen, z. B. für *selektiven Seitenausdruck*, *Seitenreihenfolge*, *Anzahl der Ausgabekopien*, *horizontale und vertikale Seitenverschiebungen*, *Vergrößerung oder Verkleinerung des Ausgabetextes* und einiges mehr. Auch die direkte Druckausgabe bzw. die Umlenkung in ein File kann, wie am Beispiel von `dvips` dargestellt, meistens ebenfalls durch Aufrufoptionen ausgewählt werden. Für Einzelheiten muss auf das jeweilige Treiberhandbuch verwiesen werden. In einem Rechenzentrum sollten diese und weitere Informationen in einem  $\TeX$ - oder  $\LaTeX$ -*Local-Guide* zusammengefasst sein und allen Benutzern zur Verfügung stehen.

## 1.8 Hinweise für Autoren

Die Autoren von Veröffentlichungen, wissenschaftlichen Publikationen oder Büchern übergaben in der Vergangenheit und teilweise bis heute dem Verlag üblicherweise ein Schreibmaschinenmanuskript – wenn nicht gar nur ein handgeschriebenes – oder eine Kombination aus beidem. Komplexere mathematische Formeln werden z. B. häufig per Hand in ein maschinengeschriebenes Manuskript eingefügt.

Der Verlagslayouter, der bei wissenschaftlichen Verlagen über ein beträchtliches Fachwissen aus dem Gebiet des Autors verfügt, legt für das angenommene Manuskript die äußere Gestaltung in Bezug auf Zeilenlänge, Schriftarten und -größen für Text, Hervorhebungen, Überschriften, Verlagsbesonderheiten u. a. fest.

Die Layoutanweisungen für das Manuskript stellen die Arbeitsanweisungen an den Setzer dar, der hieraus die Druckvorlage erstellt. An den Korrekturabzügen der ersten Vorlage wird der Autor kaum je das Layout bemängeln – die meisten Verlage würden eine autoren-gewünschte Designänderung oft auch gar nicht akzeptieren. Die einzige Einflussnahme des Autors besteht in diesem Stadium nur noch darin, Schreib- und Sachfehler zu erkennen und zu korrigieren.

Mit  $\LaTeX$  wird der Autor und/oder seine Sekretärin in die Lage versetzt, dem Verlag eine druckfertige Vorlage zu liefern. Damit werden die Kosten für wissenschaftliche Veröffentlichungen der bekanntesten Journale teilweise drastisch gesenkt, da dem Verlag die gesamten Satz- und Korrekturaufwendungen erspart bleiben. Diese Kostenminderung führt bei etlichen Verlagen zu einem erhöhten Autorenhonorar, was für Berufsautoren ein nicht unerheblicher Anreiz zur Verwendung von  $\LaTeX$  sein mag.

Damit wird der Autor gleichzeitig sein eigener Layouter, Setzer und Probedrucker. Das Layout wird hierbei weitgehend durch  $\LaTeX$  bestimmt, das seinerseits  $\TeX$  zur eigentlichen

Textbearbeitung aufruft. So wie der Setzer das Manuskript mit den Gestaltungsanweisungen des Verlagslayouters zur Druckvorlage gestaltet, erzeugt  $\text{T}_{\text{E}}\text{X}$  mit den aus  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  stammenden  $\text{T}_{\text{E}}\text{X}$ -Anweisungen die endgültige Ausgabe.

Im Unterschied zu einem Menschen ist kein Computerprogramm wirklich vernunftbegabt. Entsprechend kann  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  nicht den Sinn des Textes oder einer mathematischen Formel erfassen, um daraus eine logische Gliederung und Ordnung allein aus dem Text zu bewerkstelligen. Dem Programm  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ist also die *logische Gliederung* des eigentlichen Textes mitzuteilen, d. h. anzugeben, wann z. B. ein neues Kapitel beginnt und welcher Teil des niedergeschriebenen Textes als Kapitelüberschrift anzusehen ist. Andere Hinweise können darin liegen, dass ein Stück Text in *hervorgehobener* Schriftart gesetzt oder eingerückt werden soll. Hierzu dienen die eigentlichen  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Befehle, aus denen das Programm  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  dann das grafische Layout selbst bestimmt.

Bei der Verwendung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  verfallen viele Autoren in den Fehler, nach Angabe der logischen Struktur das von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  bestimmte Layout *verbessern* zu wollen. Dabei spielen für sie ästhetische Gesichtspunkte die entscheidende Rolle: Das fertige Schriftstück soll *schön* aussehen, wobei das jeweilige Schönheitsideal natürlich persönlichkeitsgefärbt ist.

Professionelle Verlagsdesigner erstellen dagegen ein Layout, das vorrangig die leichtere Lesbarkeit und bessere Verständlichkeit beim Leser zum Ziel hat. Hinter dem Handwerk des typographischen Designs verbirgt sich die mehr als 500jährige Erfahrung seit Gutenbergs Erfindung der Druckerkunst. Hierzu gehören die Schriftarten und *richtigen* Größenverhältnisse zwischen Kapitel-, Abschnitts-, Unterabschnittsüberschriften zum übrigen Text und deren Nummerierung, die maximale Zeilenlänge in Abhängigkeit von der gewählten Schriftgröße, die Einrücktiefe bei Hervorhebungen durch Einrückungen, Aufzählungen und vieles mehr.

Bei der Festlegung des Layouts für die angegebene logische Struktur nutzt  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , wie bereits erwähnt, das Profiwissen der Verlagslayouters. Das erstellte Layout möge der Autor üblicherweise ebenso akzeptieren, wie er bisher das Layout der Korrekturfahnen des Verlages für sein hand- oder schreibmaschinengeschriebenes Manuskript übernahm. Dies liefert stets ein übersichtlich gegliedertes und gut lesbares Ergebnis.

Die einzigen Angaben zum Layout sollten sich auf die Verlagsvorgaben für die Maßangaben von Textbreite und -höhe und evtl. die verlangten Schriftarten beschränken. Ansonsten sollten Autoren ihre Kreativität auf den Inhalt und nicht auf die Form ausrichten. Selbst wenn die  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Bearbeitung von einer Sekretärin oder Schreibkraft erledigt wird, sollte deren Gestaltungseffizienz nicht durch Sonderwünsche des Autors zum Layout beeinträchtigt werden. Mir sind Beispiele bekannt, bei denen Schreibkräfte sich zu Recht darüber beklagen, dass sie nach der Einführung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zwar mehr als 90 % ihrer Schreibarbeiten nunmehr in rund 50 % ihrer Arbeitszeit erledigen können, aber für weniger als 10 % der Textarbeiten die andere Hälfte der Arbeitszeit benötigen, da einige wenige ihrer Auftraggeber sie stets mit immer neuen Layoutforderungen nerven.

# Kapitel 2

## Befehle und Umgebungen

Der zur  $\LaTeX$ -Bearbeitung vorgesehene Text ist zunächst mit einem beliebigen *Texteditor* zu erzeugen, zu korrigieren, zu ergänzen oder zu ändern. Hierzu ist jeder Editor geeignet, z. B. `edit` unter MS-DOS, `emacs` oder `vi` unter UNIX, `lscd` unter VMS. Der Anwender sollte den ihm vertrautesten Editor verwenden. Wichtig ist, dass der verwendete Editor ausschließlich *sichtbare* Zeichen erzeugt. Das ist bei den Eingabeprogrammen einiger Textverarbeitungsprogramme, wie z. B. MS-Word, nicht immer der Fall. Jenes erzeugt auch spezielle *nicht sichtbare* Zeichen, die als interne Befehle von dem Textverarbeitungsprogramm verstanden werden.

Der Eingabetext für die  $\LaTeX$ -Bearbeitung erscheint im Editor mit allen seinen Zeichen auf dem Bildschirm. Dies gilt auch für die sog.  $\LaTeX$ -Befehle, die die logische Struktur des eigentlichen Textes kennzeichnen oder die bewirken, dass bestimmte Textstellen in besonderer Weise zu bearbeiten sind. Auch die  $\LaTeX$ -Befehle bestehen ausschließlich aus sichtbaren Zeichen, so dass für die Editoreingabe kein formaler Unterschied zwischen dem zu bearbeitenden Text und den  $\LaTeX$ -Befehlen besteht.

Dem Anwender muss dagegen bekannt sein, welche Eingabezeichen und/oder Zeichengruppen als Befehle interpretiert und bei der  $\LaTeX$ -Bearbeitung aus dem vollständigen Eingabetext herausgefiltert werden, so dass der eigentliche Eingabetext zielgerichtet bearbeitet wird.

### 2.1 Befehlsnamen und Befehlsargumente

Die Zeichen `#` `$` `&` `~` `_` `^` `%` `{` `}` stellen spezielle Befehle dar, deren Bedeutung an anderer Stelle erklärt wird. Sollen diese Zeichen als Textzeichen benutzt werden, so ist ihnen ein Rückstrich `\` (engl. backslash) voranzusetzen.

Rund 20 Befehle bestehen aus zwei Zeichen, nämlich dem `\` unmittelbar gefolgt von einem Zeichen, das kein Buchstabe ist. Auch diese *Zweizeichenbefehle* werden an anderer Stelle erläutert. So stellt z. B. `\$` den Befehl dar, die Befehlsbedeutung des Zeichens `$` aufzuheben und stattdessen dieses Zeichen als Text zu drucken.

Die große Mehrzahl von Befehlen besteht aus einem Rückstrich `\`, unmittelbar gefolgt von einem oder mehreren Buchstaben und endend vor dem ersten Zeichen, das kein Buchstabe ist. Viele Befehle haben *Argumente*, das sind variable Ergänzungen für den entsprechenden

Befehl. Argumente können *optional* sein, d. h. wahlweise benutzt oder weggelassen werden, oder sie sind *zwingend*, d. h., mindestens ein Argument aus einer Liste der zulässigen Argumente muss aufgeführt sein. Die Syntax solcher Befehle ist

$$\backslash\textit{befehlsname}[\textit{optionale Argumente}]\{\textit{zwingendes Argument}\}$$

d. h., optionale Argumente stehen in eckigen Klammern [ ] und zwingende Argumente stehen in geschweiften Klammern { }. Werden mehrere optionale Argumente benutzt, so sind sie durch Kommata (Beistriche) voneinander zu trennen, wobei die Reihenfolge gleichgültig ist, falls dies bei den Befehlsvorstellungen nicht ausdrücklich untersagt wird. Wird kein optionales Argument benutzt, so können die eckigen Klammern ebenfalls weggelassen<sup>1</sup> werden.

Einige Befehle kennen mehrere zwingende Argumente. Diese stehen jeweils für sich in { }-Paaren bei Einhaltung der Reihenfolge, wie sie bei der Beschreibung dieser Befehle angegeben ist. Beispiel:

$$\backslash\textit{rule}[\textit{lift}]\{\textit{breite}\}\{\textit{höhe}\}$$

erzeugt ein gefülltes Rechteck der Breite *breite* und der Höhe *höhe*, das um *lift* gegenüber der augenblicklichen Grundlinie verschoben ist. Ein solches Rechteck der Breite 10 mm und der Höhe 3 mm, dessen Unterkante mit der Grundlinie der laufenden Zeile übereinstimmt, wird demnach mit  $\backslash\textit{rule}\{10\textit{mm}\}\{3\textit{mm}\}$  erzeugt. Der optionale Parameter *lift* ist hier entfallen. Die Reihenfolge von Breite und Höhe ist durch die Syntax des Befehls festgelegt und darf nicht vertauscht werden.

Manche Befehle erscheinen in zwei Formen, der Standardform und der sog. \*-Form. Letztere ist dadurch gekennzeichnet, dass der zugehörige Befehlsname mit einem \* endet, und zwar vor den eventuellen Klammerpaaren [ ] oder { } für optionale oder zwingende Argumente. Die Unterschiede zwischen Standard- und \*-Form werden bei der Beschreibung der einzelnen Befehle angegeben.

Befehlsnamen enden mit dem ersten Zeichen, das kein Buchstabe ist. Folgen auf einen Befehlsnamen optionale oder zwingende Argumente, so endet der Befehlsname vor der [- oder {-Klammer, da diese kein Buchstabe ist. Viele Befehle aber kennen keine Argumente, sie bestehen nur aus einem Namen, wie z. B. die Befehle  $\backslash\textit{LaTeX}$  und  $\backslash\textit{LaTeXe}$  zur Erzeugung des  $\textit{L}^{\textit{A}}\textit{T}_{\textit{E}}\textit{X}$ - bzw. des  $\textit{L}^{\textit{A}}\textit{T}_{\textit{E}}\textit{X}_{\textit{2}_{\textit{e}}}$ -Symbols. Folgt auf solche Befehle ein Satzzeichen wie Komma oder Punkt, so ist klar, wo der Befehlsname endet. Folgt jedoch hierauf ein normales Wort, so werden die Leerzeichen zwischen dem Befehlsnamen und dem folgenden Wort als Befehlsende interpretiert: Das  $\backslash\textit{LaTeX}$  Symbol ergibt „Das  $\textit{L}^{\textit{A}}\textit{T}_{\textit{E}}\textit{X}$ Symbol“, d. h., das Leerzeichen wurde *nur* als Befehlsende angesehen und nicht als Zwischenraum zwischen zwei Wörtern.

Damit ein Zwischenraum nach einem Befehl, der nur aus einem Namen besteht, eingefügt wird, muss vor dem Leerzeichen eine Leerstruktur { } oder ein \ angebracht werden. Die Eingaben Das  $\backslash\textit{LaTeX}\{\}$  Symbol oder Das  $\backslash\textit{LaTeX}\backslash$  Symbol ergeben die gewünschte Form: „Das  $\textit{L}^{\textit{A}}\textit{T}_{\textit{E}}\textit{X}$  Symbol“. Alternativ kann der Befehl selbst in geschweifte Klammern eingeschlossen werden. Die Angabe Das  $\{\backslash\textit{TeX}\}$  Symbol erzeugt ebenfalls den gewünschten Ausdruck mit dem eingefügten Leerzeichen: „Das  $\textit{T}_{\textit{E}}\textit{X}$  Symbol“. (Beide Beispiele sind etwas gekünstelt, da sie korrekterweise mit einem Bindestrich  $\backslash\textit{TeX}\textit{-Symbol}$  zu schreiben sind, womit das Leerzeichenproblem nicht auftritt.)

<sup>1</sup> Bei Befehlen, die nur optionale Argumente kennen, kann folgendes Problem auftreten: Werden keine optionalen Argumente benutzt und beginnt der Text nach dem Befehl mit [, so interpretiert  $\textit{L}^{\textit{A}}\textit{T}_{\textit{E}}\textit{X}$  den darauffolgenden Text als *optionales Argument* und stellt i. Allg. fest, dass dieser Text kein zulässiges optionales Argument ist. Zur Vermeidung dieser Fehlinterpretation kann die eckige Klammer [ mit geschweiften Klammern { } eingeschlossen werden.

## 2.2 Umgebungen (environments)

Eine *Umgebung* wird mit dem Befehl `\begin{umgebung}` geschaffen und endet mit dem Befehl `\end{umgebung}`. Welche Werte für *umgebung* erlaubt sind, wird bei den einzelnen Befehlsgruppen aufgeführt.

Eine Umgebung bewirkt zunächst einmal, dass der innerhalb der Umgebung stehende Text entsprechend dem Umgebungsparameter anders behandelt wird als der außerhalb der Umgebung stehende Text. Zum anderen können innerhalb einer Umgebung zusätzlich bestimmte Bearbeitungsmerkmale, wie z. B. Einrücktiefe, Textbreite, Zeichensatz und viele mehr, geändert werden. Diese Änderung wirkt aber nur innerhalb der betreffenden Umgebung. Beispiel:

*vorangehender Text*

```
\begin{quote} textteil1 \small textteil2 \bfseries textteil3 \end{quote}
```

*nachfolgender Text*

Mit der `quote`-Umgebung wird der zwischen `\begin{quote}` und `\end{quote}` stehende Text gegenüber dem vorangehenden und nachfolgenden Text links und rechts eingerückt. Im Beispiel sind das die drei Textteile *textteil1*, *textteil2* und *textteil3*. Nach dem *textteil1* steht der Befehl `\small`, der bewirkt, dass der folgende Text in einer kleineren Schrift erscheint. Nach *textteil2* steht zusätzlich der Befehl `\bfseries`, mit dem erreicht wird, dass der folgende Text in Fettdruck erscheint. Die Wirkung dieser beiden Befehle endet mit `\end{quote}`.

Die drei Textteile innerhalb der `quote`-Umgebung werden also gegenüber dem vorangehenden und nachfolgenden Text beidseitig eingerückt. Der „*textteil1*“ erscheint hierbei in Normalschrift, d. h. derselben Schrift, wie sie außerhalb der Umgebung auftritt. „*textteil2*“ und „*textteil3*“ erscheinen in einer kleineren Schrift, wobei „*textteil3*“ **zusätzlich in Fettdruck erscheint**.

Nach Beendigung der `quote`-Umgebung wird der nachfolgende Text wieder in derselben Schrift wie vor der Umgebung ausgegeben.

Die meisten Befehlsnamen können auch als Umgebungsnamen benutzt werden. In diesem Fall ist der Befehlsname *ohne* den vorangehenden `\` als Umgebungsname zu benutzen. So schaltet z. B. der Befehl `\small` auf eine verkleinerte Schriftgröße um. Entsprechend schafft `\begin{small}` eine Umgebung, in der die Schriftgröße `\small` bis zum Ende der Umgebung durch `\end{small}` wirkt.

Eine namenlose Umgebung wird durch Klammerung mit einem `{ . . }`-Paar für den in diesen Klammern stehenden Textteil geschaffen. Die Reichweite von Änderungsbefehlen innerhalb einer namenlosen Umgebung endet mit der schließenden Klammer.

## 2.3 Erklärungen (declarations)

Eine *Erklärung* ist ein Befehl, mit dem der Wert oder die Bedeutung von einigen Parametern oder Befehlen verändert wird. Die Reichweite der Erklärung beginnt mit der Erklärung selbst und endet entweder mit einer weiteren Erklärung desselben Typs, spätestens aber mit dem Auftreten des Befehls `\end{ }` oder der schließenden Klammer `}`, mit der die augenblickliche Umgebung beendet wird.

Eine *Zahlenerklärung* ist die Zuweisung eines Zahlenwertes an einen  $\LaTeX$ -Zähler oder die Änderung dieses Zahlenwertes. Eine *Längenerklärung* ist die Zuweisung einer *Maßangabe* (s. u.) an einen  $\LaTeX$ -Längenbefehl oder die Änderung dieser Maßangabe. Die Zuweisungs- und Änderungs-befehle werden vollständig in 7.1.3 und 7.2 vorgestellt. Eine Kurzvorstellung für Längenerklärungen mit `\setlength` erfolgt bereits im nächsten Unterabschnitt.

Beispiele:

`{\bfseries Dieser Text erscheint in Fettdruck}` Die Erklärung `\bfseries` bewirkt eine Schriftänderung: **Dieser Text erscheint in Fettdruck**. Die Wirkung dieser Erklärung endet mit der schließenden Klammer `}`.

`\setlength{\parindent}{0.5cm}` Die Einrücktiefe der ersten Zeile eines Absatzes (`\parindent`) wird auf 0.5 cm gesetzt. Die Wirkung dieser Erklärung endet mit einer weiteren Erklärung für `\parindent`, spätestens aber mit dem `\end`-Befehl, der die laufende Umgebung beendet.

`\pagenumbering{roman}` Die Seitennummerierung erfolgt in römischen Ziffern.

Einige Erklärungen, wie etwa das letzte Beispiel, sind global, d. h., ihre Wirkung ist nicht auf die augenblickliche Umgebung beschränkt. Es sind dies die Befehle, deren Bedeutung später erklärt wird:

<code>\newcounter</code>	<code>\pagenumbering</code>	<code>\newlength</code>
<code>\setcounter</code>	<code>\thispagestyle</code>	<code>\newsavebox</code>
<code>\addtocounter</code>		

Erklärungen mit diesen Befehlen wirken von der Stelle ihres Auftretens, bis sie durch eine aufhebende Erklärung beendet werden. Durch das letzte Beispiel würde die Seitennummerierung so lange in römischen Ziffern erfolgen, bis sie durch eine erneute Erklärung, z. B. `\pagenumbering{arabic}`, abgelöst und damit aufgehoben wird.

## 2.4 Maßangaben

### 2.4.1 Feste Maße

*Maßangaben* bestehen aus einer Dezimalzahl mit einem möglichen Vorzeichen (+ oder -), zwingend gefolgt von einer Maßeinheit. Die folgenden Maßeinheiten sind erlaubt:

cm	Zentimeter	bp	big point (1 in = 72 bp)
mm	Millimeter	dd	Didot (1157 dd = 1238 pt)
in	Inches (Zoll = 2.54 cm)	cc	Cicero (1 cc = 12 dd)
pt	Punkte (1 in = 72.27 pt)	sp	scaled point (1 pt = 65536 sp)
pc	Picas (1 pc = 12 pt)		
em	Die Breite des Geviertstrichs [—] im jeweils aktiven Zeichensatz.		
ex	Die Höhe des Buchstabens x im jeweils aktiven Zeichensatz.		

Dezimalzahlen können in  $\TeX$  und  $\LaTeX$  sowohl in englischer wie in deutscher Schreibweise mit einem *Dezimalpunkt* oder *Komma* geschrieben werden: 12.5cm und 12,5cm, aber auch 3cm ohne Punkt oder Komma ist eine erlaubte Eingabe.

Die Erklärung zur Zuweisung einer Maßangabe an einen Längenbefehl erfolgt durch `\setlength`, wie bereits oben erwähnt. Die Syntax dieser Zuweisungserklärung lautet:

```
\setlength{\längen_befehl}{maßangabe}
```

Die Einstellung der Textbreite, die den Zeilenumbruch bestimmt, wird in  $\LaTeX$  durch die Erklärung von `\textwidth` gesteuert. Dieser ist standardmäßig ein von Bearbeitungsstil und Schriftgröße abhängiges Maß zugewiesen. Mit

```
\setlength{\textwidth}{12.5cm}
```

wird statt der Standardeinstellung die Textbreite (= Zeilenbreite) mit 12.5 cm gewählt.

$\LaTeX$ -Anwender mit  $\TeX$ -Erfahrung verwenden häufig eine abkürzende Schreibweise für die Zuweisung einer Maßangabe an einen Längenbefehl. Eine Längenerklärung kann auch durch unmittelbares Anhängen der Maßangabe an den Erklärungsnamen erfolgen: `\textwidth12.5cm` hat dieselbe Wirkung wie die vorangegangene Einstellung mit `\setlength`. Die Einstellung mit `\setlength{\textwidth}{12.5cm}` führt intern zu der Ablauffolge `\textwidth12.5cm\relax`. Obwohl der  $\TeX$ -Leerbefehl `\relax` die Wirkung des *Nichtstuns* hat, kann die Zwischenschaltung in Bezug auf nachfolgende Befehle bedeutsam sein. Neue  $\LaTeX$ -Anwender sollten sich die Unart der abgekürzten Zuweisung deshalb nicht angewöhnen. Die Kenntnis der verkürzten Zuweisung sollte nur für das vertiefte Verständnis von Makropaketen, wie z. B. den  $\LaTeX$ -Stilfiles, genutzt werden.

In 3.5.1.6, 3.5.2.1 und 3.5.3 werden  $\LaTeX$ -Abstandsbeefehle vorgestellt, die eine Maßangabe als Befehlsargument erwarten. Beispiel: `\hspace{1.5em}`.

## 2.4.2 Elastische Maße

Einige Erklärungen erwarten *elastische* Maßangaben. Das sind Längen, die um einen bestimmten Betrag gedehnt oder gestaucht werden können. Die Syntax für elastische Maße lautet:

```
sollwert plusdehnwert minusstauchwert
```

wobei *sollwert*, *dehnwert* und *stauchwert* jeweils eine Maßangabe ist. Die elastischen Anteile sind mit ihren Schlüsselwörtern `plus` bzw. `minus` optional. Die Zuweisung erfolgt ebenfalls mit dem Befehl `\setlength` oder als Argument in einem Abstandsbeefehl.

```
\setlength{\parskip}{1ex plus0.5ex minus0.2ex}
```

bewirkt: Der Abstand, der zusätzlich zum Zeilenabstand zwischen Absätzen eingefügt wird (`\parskip`), ist gleich der Höhe eines `x` aus dem momentanen Zeichensatz. Dieser Abstand kann jedoch bis auf das 1.5fache gedehnt oder das 0.8fache zusammengedrückt werden.

Ein besonderes *elastisches* Maß ist `\fill`. Dieses hat die natürliche Länge *Null*, die auf jede beliebige Länge gedehnt werden kann. Die Zuweisung erfolgt mit `\setlength` oder als Argument in einem Abstandsbeefehl: `\hspace*{\fill}`.

## 2.5 Sonderzeichen

### 2.5.1 Eingabe der Umlaute und des ß (s. D.1.1)

Die Eingabe der Umlaute erfolgt durch unmittelbares Voranstellen des " : "a wird also ä, "o ö, "u ü, "A Ä, "O Ö und "U wird Ü. Die Eingabe des ß erfolgt durch "s. Dies setzt das deutsche Ergänzungspaket `german.sty` oder hilfsweise das Anpassungsfile `danp.sty` gemäß Übung 2.1 (S. 22) voraus.

In der Originalversion von  $\LaTeX$  werden Umlaute durch das Voranstellen des Befehls `\` erzeugt, z. B. `sch\`on für „schön“. Das  $\beta$  wird im Original durch `\ss` erzeugt, wobei seine Verwendung im Wortinnern und am Wortende unterschiedlich zu handhaben ist: Der Befehl `\ss` endet mit dem ersten Zeichen, das kein Buchstabe ist. Ein solches Zeichen könnte ein Leerzeichen sein, das im Wortinnern nur das Befehlsende kennzeichnet. Am Wortende muss dagegen ein Leerzeichen zugefügt werden, was z. B. durch `\ss\` erreicht werden kann (`\` steht zur Verdeutlichung eines Leerzeichens). Alternativ könnte der Befehl einheitlich in geschweiften Klammern als `{\ss}` geschrieben werden. Beispiel: `h\"a\ss lich` oder `h\"a{\ss}lich` bzw. `\mu\ss\` oder `\mu{\ss}`.

Die Erzeugung von Umlauten und insbesondere des  $\beta$  in der Originalversion ist nicht nur umständlicher, sondern erschwert auch das Lesen des Eingabetextes erheblich. Das Beispiel

```
Die h\"a\ss liche Stra{\ss}e mu\ss\ sch\"oner werden.
Die h\"a\"sliche Stra\"se mu\"s sch\"oner werden.
```

zeigt in der ersten Zeile die erforderliche Eingabe für die Originalversion. Die Vereinfachung und bessere Lesbarkeit durch die eingangs beschriebene deutsche  $\LaTeX$ -Option demonstriert die zweite Zeile überzeugend. In beiden Fällen erscheint als Ausgabebetext:

```
Die häßliche Straße muß schöner werden.
```

Viele  $\LaTeX$ -Anwender verwenden auf einem PC im deutschsprachigen Raum eine deutsche statt der internationalen ASCII-Tastatur. Die deutsche Tastatur enthält für die Umlaute und das  $\beta$  entsprechende Eingabetasten. Diese können durch Aktivierung des Ergänzungspakets `inputenc.sty` (s. Anh. D.2.5) mit dem Vorspannbefehl

```
\usepackage[cp850]{inputenc}
```

genutzt werden. Das auf PCs sehr verbreitete Programmpaket `emTeX` von EBERHARD MATES kennt weitere Möglichkeiten zur Nutzung der deutschen Sondertasten. Anhang F.3.1.3 beschreibt die erforderlichen Maßnahmen.

**Achtung:** Falls beim Leser eine deutsche Tastatur zur Anwendung kommt, so möge er bei den abgedruckten Beispielen dieses Buches die dortigen Angaben für die Umlaute und das  $\beta$  als "a, "o, "u, "A, "O, "U und "s durch die deutschen Sondertasten als ä, ö, ü, Ä, Ö, Ü und ß ersetzen.

## 2.5.2 Anführungsstriche

Die auf der Schreibmaschine vorkommenden Anführungsstriche " werden beim Druck nicht verwendet. Hier werden die am Anfang und Ende unterschiedlichen Zeichen ‘Wort’ oder ‘Satz’ benutzt. Diese werden erzeugt durch ‘ für ‘, ’ für ’ sowie durch das zweimalige Eintippen von ‘ ‘ für “ und ’ ’ für ”. (In der  $\LaTeX$ -Originalversion erzeugt die Eingabe von " ebenfalls ”; bei deutschen  $\LaTeX$ -Versionen oder -Anpassungen hat " dagegen Befehlsbedeutung, u. a. zur Erzeugung der Umlaute!)

Bei deutschen Texten sind statt der englischen “Quotes” die Anführungsstriche in Form der „Gänsefüßchen“ gebräuchlicher. Die unteren (linken) Striche könnten durch zwei Kommata (Beistriche) angenähert werden: , , deutsche ‘ ‘ Form ergibt „deutsche“ Form. Erfolgt die deutsche Anpassung durch das `german.sty`-File, so können sie korrekter durch die Befehle " ‘ und " ’ erzeugt werden: vgl. das Ergebnis von " ‘ deutsche " ’ Form als „deutsche“ Form mit dem ersten Beispiel. Die unteren Anführungsstriche sind *enger* als beim Doppelkomma. Weitere Möglichkeiten enthält Anhang D.1.5.

### 2.5.3 Trenn-, Binde-, Strecken- und Gedankenstriche

Bei gedruckten Texten finden Striche unterschiedlicher Länge Verwendung: -, –, —. Der kurze *Trennstrich* wird als Trennungszeichen bei Worttrennungen und als Bindezeichen in zusammengesetzten Begriffen (*Bindestrich*) benutzt. Der längere *Streckenstrich* wird bei *Strecken*-Angaben wie Hamburg–München sowie für *von–bis*-Angaben wie z. B. 1–4 verwendet. Der *Gedankenstrich* — dies könnte z. B. hier erfolgen — dient zum Einfügen von Satzteilen. Erzeugt werden diese Striche durch ein, zwei oder drei Trennzeichen, also - ergibt -, -- – und --- —. Das Minuszeichen – entsteht im mathematischen Modus durch ein -.

Die Verwendung unterschiedlich langer Strecken- und Gedankenstriche entsprach einer früheren Duden-Empfehlung, bei der ihr Längenverhältnis mit 2 : 3 angegeben wurde (Duden, Band 1, 17. Aufl.). Später (20. Aufl.) wurde der kürzere Streckenstrich, allerdings mit voran- und nachgestelltem Leerraum, auch als Gedankenstrich vorgeschrieben. Dies entspricht der typographischen Praxis in Deutschland. Für weitere Einzelheiten verweise ich auf den Abschnitt „Vorschriften für den Schriftsatz“ im Duden, Band 1. Im englischen Sprachraum kommt dagegen der längere, mit --- erzeugte Gedankenstrich zur Anwendung, wobei die verknüpften Satzteile *ohne* voran- und nachgestellten Leerraum miteinander verbunden werden.

### 2.5.4 Der Druck von Befehlszeichen

Wie unter 2.1 erwähnt, werden die Zeichen # \$ & ~ \_ ^ % { } als Befehle interpretiert. Sollen sie als Textzeichen Verwendung finden, so sind die meisten durch das Voranstellen des \ zu erhalten. Zur Ausgabe der Textzeichen ^ und ~ wird auf 4.1.7, S. 65, verwiesen.

# = \#    \$ = \\$    & = \&    \_ = \\_    % = \%    { = \{    } = \}

### 2.5.5 Die Sonderzeichen §, †, ‡, ¶, © und £

Diese Sonderzeichen stehen auf der Tastatur nicht zur Verfügung. Solche in deutschen oder englischen Texten gelegentlich vorkommenden Zeichen können durch spezielle Befehle erzeugt werden:

§ = \S    † = \dag    ‡ = \ddag    ¶ = \P    © = \copyright    £ = \pounds

Die Erzeugung griechischer Buchstaben und mathematischer Zeichen wird im Kapitel 5 (Mathematische Formeln) dargestellt.

### 2.5.6 Sonderbuchstaben in Fremdsprachen

Sonderbuchstaben, die in europäischen Sprachen vorkommen, stellt T<sub>E</sub>X ebenfalls zur Verfügung. Dies sind:

Œ={\OE}    Æ={\AE}    Å={\AA}    Ø={\O}    ŠS={\SS}    Ł={\L}    ¡=! ‘  
œ={\oe}    æ={\ae}    å={\aa}    ø={\o}    ß={\ss}    †={\l}    ¿=? ‘

Ångström wird also {\AA}ngstr{\O}m geschrieben, und „Tromsø in Norwegen“ wird durch "‘Troms{\O} in Norwegen"’ erzeugt. Die ß-Großform SS wird mit \SS bereitgestellt.

### 2.5.7 Akzente

Die europäischen Sprachen kennen eine Vielzahl von Akzenten. Die meisten von ihnen stehen in  $\TeX$  zur Verfügung:

```

ò=\`{o}   ó=\`{o}   ô=\^{o}   ö=\\"{o}   õ=\~{o}   õ=\={o}   ô=\. {o}
ô=\r{o}   õ=\u{o}   õ=\v{o}   õ=\H{o}}   õó=\t{o}o}
q=\c{o}   q=\d{o}   q=\b{o}

```

Statt des ‘o’ kann jeder Buchstabe stehen. Beim ‘i’ und ‘j’ ist zu beachten, dass bei einem Akzent über ihnen zunächst der Punkt zu entfernen ist. Dies geschieht einfach durch das Voranstellen des \. Man erhält also i und j aus \i und \j, und ĭ bzw. ĵ wird durch \u{i} bzw. \H{j} erzeugt.

Die Akzente der ersten Zeile der vorangegangenen Tabelle können vereinfacht auch ohne Einschluss in geschweiften Klammern erzeugt werden:

```

ò=\`o   ó=\`o   ô=\^o   ö=\\"o   õ=\~o   õ=\=o   ô=\.o

```

Die Akzente der zweiten und dritten Zeile sollten dagegen stets durch Einschluss in { }-Paare erzeugt werden.

### 2.5.8 Ligaturen

Bei gedruckten Texten werden bestimmte Buchstabenkombinationen nicht durch Aneinanderreihung der betreffenden Buchstaben, sondern durch *Ligaturen* als eigenes Kombinationszeichen gesetzt.  $\TeX$  erzeugt die Buchstabenfolgen ff, fi, fl, ffi und ffl nicht als

```
ff, fi, fl, ffi, ffl   sondern als   ff, fi, fl, ffi, ffl
```

Abschnitt 3.5.1.5 beschreibt die Maßnahmen, wenn von diesem Standard abgewichen, d. h. eine der vorstehenden Buchstabengruppen durch Aneinanderreihung von Einzelbuchstaben gebildet werden soll. Dies kann bei zusammengesetzten Wörtern sinnvoll sein, wie bei „Auflage“ statt „Auflage“. Weitere Möglichkeiten zur Aufhebung von Ligaturen, bei denen gleichzeitig eine evtl. Trennhilfe erfolgt, werden in Anhang D.1.4 vorgestellt.

### 2.5.9 Das Datum

Das aktuelle Datum kann mit dem Befehl \today an jeder Stelle im Text eingefügt werden und erscheint in der US-Form „February 24, 2002“. Steht das im Anhang D.1 beschriebene german.sty-File zur Verfügung, so erzeugt der Befehl \today die deutsche Form „24. Februar 2002“, wenn der Vorspannbefehl \usepackage{german} benutzt wurde. Durch Aktivieren eines *Sprachschalters* kann \today das Datum auch in österreichischer, englischer, französischer oder in der ursprünglichen US-Form erzeugen. Einzelheiten hierzu sind dem Anhang D.1.7 zu entnehmen.

### 2.5.10 Vorgriff auf german.sty

In den vorangegangenen Unterabschnitten war mehrfach auf das deutsche Ergänzungspaket german.sty verwiesen worden. Es sollte zum Standardbestandteil einer jeden deutschsprachigen  $\TeX$ -Installation gehören, was mit den nachfolgenden Beschaffungshinweisen inzwischen stets vorausgesetzt werden kann.

Die Bearbeitungsoptionen für deutschsprachige Texte werden durch die alternativen Ergänzungspakete `german.sty` (alte Rechtschreibung) und `ngerman.sty` (neue Rechtschreibung) realisiert. Beide ermöglichen eine einfachere Form der Eingabe für die Umlaute und das ß. Auf die Möglichkeiten zur Erzeugung der deutschen Anführungsstriche (Gänsefüßchen) wurde bereits in 2.5.2 hingewiesen. Schließlich sorgen sie dafür, dass bestimmte Begriffe, wie „Kapitel“, „Inhaltsverzeichnis“, „Literaturverzeichnis“ u. a., die von  $\LaTeX$  automatisch zugefügt werden, mit ihren deutschen Bezeichnungen statt des englischen Originals „Chapter“, „Contents“, „Bibliography“ usw. ausgegeben werden. Mit `german.sty` erfolgt außerdem, eine Unterstützung der deutschen Trennregeln für die alte Rechtschreibung beim `‘ck’` und bei zusammengesetzten Wörtern mit Doppelkonsonanten.

Die Eigenschaften von `german.sty` zur Bearbeitung deutschsprachiger Texte werden geschlossen in Anhang D.1 vorgestellt. Beim Durcharbeiten dieses Buches empfiehlt es sich, an dieser Stelle auf die Abschnitte D.1.1–D.1.9 (S. 340–345) vorzugreifen.

Anwender mit Internet-Zugang können die jeweils neueste Version von `(n)german.sty` von den öffentlichen  $\TeX$ -Fileservern abrufen. Nähere Hinweise enthält Anhang F.5. Als weitere Beschaffungsquelle sei die deutschsprachige  $\TeX$ -Anwendervereinigung DANTE e. V. (s. F.5.3) genannt, die ihre Mitglieder mit jährlich aktualisierten CDs für alle kostenlosen  $\TeX$ -Produkte versorgt. Die beiden Ergänzungspakete `german.sty` und `ngerman.sty` findet man auch auf der beiliegenden CD mit dem Titel  $\TeX$  Live 5c (s. hierzu auch F.5.4).

## 2.6 Zerbrechliche Befehle

Einige Befehle entfalten ihre Wirkung nicht nur an der Stelle ihres Auftretens, sondern zusätzlich an weiteren Stellen des Dokuments. So erzeugen die Gliederungsbefehle wie z. B. `\chapter{Überschrift}` eine Überschrift an der Stelle dieses Befehls. Die Überschrift wird ggf. auf den folgenden Seiten in einer anderen Schriftart in der Kopfzeile erscheinen und evtl. auch in nochmals einer anderen Schriftart im Inhaltsverzeichnis. Ein Argument, das an mehreren Stellen des Dokuments in Erscheinung tritt, wird als *wanderndes Argument* bezeichnet.

Beim Wandern wird ein solches Argument, bildlich gesprochen, heftig *geschüttelt*. Enthält ein wanderndes Argument weitere Befehle, so können einige davon während der *Wanderung* durch das heftige *Schütteln* zerbrechen. Andere Befehle erweisen sich gegen jede noch so harte Belastung gefeit. Diese heißen *robust*, die ersten *zerbrechlich*.

Grundsätzlich sind in  $\LaTeX$  2.09 alle Befehle, die optionale Parameter kennen, sowie die `\begin-` und `\end-`Befehle zerbrechlich. Einige dieser Befehle erweisen sich in  $\LaTeX$  2 $\epsilon$  als robust. Zerbrechliche Befehle in einem wandernden Argument können durch das Voranstellen des Befehls `\protect` vor dem Zerbrechen geschützt werden.

Zerbrochene Befehle können von  $\LaTeX$  nicht korrekt bearbeitet werden und erzeugen deshalb eine Reihe von Fehlermeldungen auf dem Bildschirm. Mit der Return-Taste kann versucht werden, trotz fehlerhafter Bearbeitung dieses Befehls mit der weiteren Textbearbeitung fortzufahren. Als Folge davon werden zunächst weitere Fehlermeldungen auf dem Bildschirm erscheinen. Nach mehrfach wiederholter Betätigung der Return-Taste setzt  $\LaTeX$  i. Allg. schließlich die Bearbeitung des Textes fort, es sei denn, dass die Wirkung des zerbrochenen Befehls eine Weiterverarbeitung nicht mehr zulässt und die Bearbeitung damit beendet wird.

Wandernde Argumente kennen nur die folgenden Befehle:

- Alle Befehle, die Textinformation in Inhaltsverzeichnisse übertragen. Dies sind die Gliederungsbefehle (3.3.3), `\addtocontents` und `\addcontentsline` (3.4.3) sowie `\caption` (6.6.4). Siehe hierzu die Anmerkungen zur Übung 3.12 auf S. 45
- Die Befehle `\typein` und `\typeout` (8.1.3) sowie `\bibitem[...]` (4.3.6, 8.2.2)

- Die Befehle `\markboth` und `\markright` (3.2.1)
- Der `\thanks`-Befehl für die Titelseite (3.3.1)
- `@`-Ausdrücke (s. 4.8.1, Seite 95)
- Der Befehl `\begin{letter}`, falls der `\makelabels`-Befehl aktiviert ist (s. A.1 auf Seite 281)

`\protect`-Befehle zur Sicherung zerbrechlicher Befehle sind nur erforderlich, wenn diese Befehle als Argumente in einem der vorstehenden Befehle auftreten. Mit Ausnahme des Befehls `\value` (s. 7.1.3, S. 179) sowie der Längenbefehle (2.4, 7.2) darf allen  $\LaTeX$ -Befehlen ein `\protect` vorangestellt werden, das ggf. ohne Wirkung bleibt. Diese Möglichkeit kann bei Unsicherheit über eine evtl. Zerbrechlichkeit vorsorglich genutzt werden.

## 2.7 Übungen

Für ein erfolgreiches Selbststudium sind praktische Übungen unerlässlich. Neben den zahlreichen Beispielen, die zur Übung nachvollzogen werden sollten, werden im weiteren Verlauf Übungsaufgaben vorgeschlagen, die der Leser unbedingt ausführen sollte. Bei den Übungen wird vorausgesetzt, dass die vorhandene  $\LaTeX$ -Implementation über das deutsche Ergänzungspaket `german.sty` verfügt und  $\LaTeX$  in der Version  $2_{\epsilon}$  zur Anwendung kommt. Falls der Anwender dies nicht weiß, wird ihm dies die erste Übung sofort verraten.

**Übung 2.1:** Mit dem Editor wird folgender Text erzeugt

```
\documentclass{article}
\usepackage{german}
\setlength{\parindent}{0pt}
\begin{document}
Aller Anfang ist schwer -- doch nicht bei \LaTeX\
"A, "0, "U -- "a, "o, "u -- "s -- heute = \today
\end{document}
```

und unter dem Namen `uebung.tex` abgespeichert. Der Aufruf zur  $\LaTeX$ -Bearbeitung ist systemabhängig und muss vom Rechenzentrum erfragt oder dem Installationshandbuch entnommen werden. Angenommen, er lautet `latex`, was fast immer der Fall ist, dann erfolgt die Bearbeitung durch

```
latex uebung
```

Anmerkung: Obwohl der Filename `uebung.tex` lautet, ist beim Aufruf nur der Grundname `uebung` anzugeben.

Sind beim Anwender die Bearbeitungsvoraussetzungen vorhanden, d. h.,  $\LaTeX$  steht in der Version  $\LaTeX 2_{\epsilon}$  zur Verfügung und das deutsche Anpassungsfile `german.sty` kann bei der Bearbeitung als Ergänzungspaket eingebunden werden, dann erscheinen nach dem obigen Bearbeitungsaufruf auf dem Bildschirm die folgenden Mitteilungen (am Beispiel von  $\emTeX$  auf meinem PC) ohne die hier vorangestellte Zeilennummer:

```
1 This is emTeX (tex386), Version 3.14159 [4a] (no format preloaded)
2 **&latex uebung
3 (uebung.tex
4 LaTeX2e <1998/06/01>
5 (c:/emtex/texinput/latex2e/article.cls
6 Documentclass: article 1998/05/05 v1.3y Standard LaTeX document class
7 (c:/emtex/texinput/latex2e/size10.clo))(c:/emtex/texinput/german/german.sty
8 Package 'german', Version 2.5e of 1998-07-08
9 No file uebung.aux
```

```

10 [1] (uebung.aux)
11 Output written on uebung.dvi (1 page, 472 bytes)
12 Transcript written on uebung.log

```

Ähnliche Bildschirmausgaben entstehen bei allen  $\LaTeX$ -Aufrufen. Sie sollen deshalb hier kurz erläutert werden. Die erste Zeile gibt die Programmquelle an, hier  $\text{em}\TeX$ , zusammen mit der Versionsnummer. Auf meinem UNIX-Rechner wird hierfür „This is TeX, C Version 3.14159“ ausgegeben. Die nächste Zeile besagt, dass das  $\TeX$ -Programm zur Bearbeitung des Files `uebung.tex` ein *Formatfile* `latex.fmt` vorab einliest, was durch das vorangestellte  $\&$  symbolisiert wird. Dieses Formatfile stellt das eigentliche  $\LaTeX$ -Interface bereit.

Die dritte Zeile mit der öffnenden Klammer verweist darauf, dass mit dem Lesen von `uebung.tex` begonnen wurde und sein Inhalt dann mit der in Zeile 4 genannten  $\LaTeX$ -Version bearbeitet wird. Dazu wird das in Zeile 5 genannte *Klassenfile* `article.cls` geladen, das sich im Verzeichnis `c:\emtex\texinput\latex2e` befindet. Zeile 6 dokumentiert dessen Erstellungsdatum. Aus dem gleichen Verzeichnis wird dann das *Optionsfile* `size10.clo` und aus `c:\emtex\texinput\german` das Ergänzungspaket `german.sty` hinzugeladen, wobei Zeile 8 Version und Erstellungsdatum von `german.sty` dokumentiert.

Bei der allerersten  $\LaTeX$ -Bearbeitung eines Files `name.tex` existiert noch kein internes Hilfsfile `name.aux`, was Zeile 9 wiedergibt. Das eingelesene Textfile `uebung.tex` wird dann formatiert und das Formatierungsergebnis mit den entstehenden Seitennummern protokolliert [1]. Bei dem vorliegenden kurzen Übungstext bleibt dieser auf eine Seite beschränkt. Anschließend wird das interne Hilfsfile erzeugt und unter dem Namen `uebung.aux` abgelegt.

Das Bearbeitungsergebnis wird in `uebung.dvi` abgelegt, dessen Größe mit der Seiten- und Byteanzahl angegeben wird (Zeile 11). Schließlich wird mit der Zeile 12 mitgeteilt, dass das Bearbeitungsprotokoll unter `uebung.log` abgelegt worden ist. Dieses enthält neben der Wiederholung der Bildschirmausgaben eine Reihe weiterer interner Informationen.

Das erzeugte File `uebung.dvi` kann nun über den Druckertreiber auf dem Drucker oder evtl. als Preview auf dem Bildschirm ausgegeben werden. Die hierzu erforderlichen Aufrufe müssen der Systembeschreibung (Local Guide) oder dem Handbuch des Druckertreibers entnommen werden. Bei mir lauten sie `dvihp1j` oder `dvips` für die Druckerausgabe und `v` für die Bildschirmausgabe, jeweils gefolgt von dem Filegrundnamen, also

```
dvihp uebung oder dvips uebung bzw. v uebung
```

Als Bearbeitungsergebnis erscheint auf dem Drucker oder Bildschirm, abgesehen vom aktuellen Datum:

```

Aller Anfang ist schwer – doch nicht bei  $\LaTeX$ 
Ä, Ö, Ü – ä, ö, ü – ß – heute = 24. Februar 2002

```

Erscheint nach dem Bearbeitungsaufwurf `latex uebung` eine Fehlermeldung der Form

```

! Undefined command sequence
1.1 \documentclass

```

dann steht  $\LaTeX$  auf dem Rechner des Anwenders noch nicht in der Version  $2_{\epsilon}$  zur Verfügung. In diesem Fall sollte die Bearbeitung mit der Eingabe von `X` abgebrochen werden. Vor dem Weiterstudium von  $\LaTeX$  mit dieser „Einführung“ muss sich der Anwender dann zunächst die aktuelle  $\LaTeX$ -Version, also  $\LaTeX 2_{\epsilon}$  beschaffen und installieren, z. B. von der beiliegenden CD-ROM.

Erscheint bei der  $\LaTeX$ -Bearbeitung von `uebung.tex` die Fehlermeldung

```

! LaTeX Error: File 'german.sty' not found
Type X to quit or <RETURN> to proceed,
. . . . .

```

so sollte auch hiernach die Bearbeitung mit der Eingabe von `X` zunächst abgebrochen werden.

Die Fehlermitteilung, dass das File mit dem Namen `german.sty` nicht gefunden wurde, muss nicht bedeuten, dass es auf dem Rechner des Anwenders nicht existiert. Letzteres ist eher unwahrscheinlich, da alle  $\TeX$ -Verteilungsmedien aus dem deutschsprachigen Raum das deutsche Anpassungsfile enthalten. Damit ist eher zu vermuten, dass bei der  $\TeX$ -Installation `german.sty` in ein Verzeichnis kopiert wurde, das bei der  $\TeX$ - und  $\LaTeX$ -Bearbeitung bei der Suche nach angeforderten Files nicht durchmustert wird. Der Anwender sollte deshalb sein Festplattensystem nach dem File `german.sty` durchmustern, wozu jedes Betriebssystem geeignete Suchbefehle bereitstellt.

Wird `german.sty` auf diese Weise gefunden, so sollte es zunächst auch in das aktuelle Verzeichnis kopiert werden, also in das Verzeichnis, in dem die vorstehende Übung mit dem Editor eingerichtet wurde und aus dem heraus der  $\LaTeX$ -Bearbeitungsaufwurf erfolgte, da das aktuelle Verzeichnis bei der Filesuche stets ebenfalls durchmustert wird. Das Anpassungsfile `german.sty` sollte dann jedoch alsbald anhand der Installationsbeschreibung in das systemspezifische Zielverzeichnis verschoben werden. Entsprechendes gilt für weitere Files aus dem  $\TeX/\LaTeX$ -System, falls solche ebenfalls abweichend von der Installationsbeschreibung kopiert wurden.

Fehlt dagegen das deutsche Anpassungsfile `german.sty` auf dem Rechner des Anwenders, so möge er sich vorübergehend mit dem Editor den folgenden Text erzeugen:

```
\catcode'\= \active \def"#1{\if#1s{\ss}\else\accent'177 #1}
\def\today{\number\day.\space\ifcase\month\or Januar\or Februar\or
M"arz\or April\or Mai\or Juni\or Juli\or August\or September\or
Oktober\or November\or Dezember\fi\space\number\year}
\def\contentsname{Inhaltsverzeichnis} \def\indexname{Index}
\def\bibname{Literaturverzeichnis} \def\tablename{Tabelle}
\def\listfigurename{Bildverzeichnis} \def\figurename{Bild}
\def\listtablename{Tabellenverzeichnis} \def\partname{Teil}
\def\chaptername{Kapitel} \def\abstractname{Zusammenfassung}
```

und diesen unter dem Namen `danp.sty` (Anpassung für deutsche Texte) abspeichern. Im Übungsfile für die Übung 2.1 ist lediglich die Angabe `german` im `\usepackage`-Befehl durch `danp` zu ersetzen. Nach der  $\LaTeX$ -Bearbeitung dieses Ersatzfiles für `uebung.tex` sollte bei der Druckerausgabe für die Umlaute, das  $\beta$  und das Datum ein ähnliches Ergebnis herauskommen, wie es beim Vorhandensein von `german.sty` gezeigt wurde. Der Anwender sollte sich jedoch bemühen, `german.sty` so bald wie möglich zu beschaffen, da dieses weitere deutsche Besonderheiten, z. B. die bei der Trennung von 'ck' und bei Doppelkonsonanten, berücksichtigt.

Gegen eine Umbenennung von `danp.sty` in `german.sty` spricht die Trivialität, da mit Ausnahme der vereinfachten Umlauteingabe und des  $\beta$  sowie der Bereitstellung einiger deutscher Begriffe keine weiteren deutschen Besonderheiten berücksichtigt werden.

**Übung 2.1 (Fortsetzung):** Falls der Leser eine deutsche Tastatur verwendet, dann sollte er in der vorangegangenen Übung den `\today`-Befehl in der zweiten Textzeile mit dem Zeilenumbruchbefehl `\\` abschließen und die anschließende Eingabe für die Umlaute und das  $\beta$  mit den Tasten  $\ddot{A}$ ,  $\ddot{O}$ ,  $\ddot{U}$  --  $\ddot{a}$ ,  $\ddot{o}$ ,  $\ddot{u}$  --  $\beta$  fortsetzen. Außerdem ist ein weiterer `\usepackage`-Befehl der Form

```
cp850 unter DOS u. OS/2
\usepackage[dec_name]{inputenc} mit ansinew unter WINDOWS xx bzw.
latin1 unter LINUX
```

für `dec_name` im Vorspann einzufügen (s. Anh. D.2.5!). Das  $\LaTeX$ -Bearbeitungsergebnis enthält nun eine weitere Zeile mit den gleichen Umlauten und dem  $\beta$  als Folge der direkten Sondertasten-Eingabe.

Die meisten Anwender mit einer deutschen Tastatur werden vermutlich für die direkte Tasteneingabe der Umlaute und des  $\beta$  plädieren. Das ist für  $\LaTeX 2_{\epsilon}$  auch sachgerecht, da das Ergänzungspaket `inputenc.sty` Bestandteil aller  $\LaTeX 2_{\epsilon}$ -Grundinstallationen ist, womit die Kompatibilität auf allen Rechnern garantiert wird. Zwar war es auch mit  $\LaTeX 2.09$  im Prinzip möglich, die Sondertasten für die Umlaute und das  $\beta$  für die  $\LaTeX$ -Bearbeitung zu aktivieren. Dies verlangte jedoch die Erstellung eines speziellen Formatfiles, womit die rechnerübergreifende Kompatibilität in Frage gestellt wurde.

**Übung 2.2:** Erzeugen Sie die Befehlsfolgen

```
\documentclass{article}
\usepackage{german}
\usepackage[decode_name]{inputenc} (s. nachstehenden Erläuterung)
\begin{document}
. . . . .
\end{document}
```

und fügen Sie zwischen `\begin{document}` und `\end{document}` ein Stück Text ein, das etwa eine dreiviertel Seite füllt, indem Sie einen entsprechenden Text aus einem Buch oder einer Zeitschrift abschreiben oder selbst formulieren. Beachten Sie hierbei, dass Absätze einfach durch Leerzeilen getrennt werden. Legen Sie diesen Text in einem File mit einem geeigneten Namen wie z. B. `uebung22.tex` ab und wiederholen Sie die Bearbeitungsprozedur einschließlich des Ausdrucks wie bei der vorangegangenen Übung. Für `decode_name` ist je nach Betriebssystem `cp850` (DOS und OS/2), `ansinew` (WINDOWS *xx*) bzw. `latin1` (LINUX) zu wählen (s. Anh. D.2.5).

Anmerkung: Der benutzte Text sollte keine besonderen Strukturen wie Blockeinrückungen, verschiedene Schriften, zentrierte Textzeilen, Aufzählungen, mathematische Formeln, Tabellen u. ä. enthalten. Die Erzeugung solcher Strukturen werden Sie in Kürze erlernen.

Erscheinen bei der  $\text{\LaTeX}$ -Bearbeitung Bildschirmwarnungen der Form `Overfull \hbox . . .`, so kann  $\text{\TeX}$  die entsprechenden Zeilen nicht sauber umbrechen. Im Ausdruck ragen diese Zeilen über den rechten Rand hinaus, weil das über den Rand ragende Wort oder der Wortteil nicht weiter getrennt werden konnte. Korrekturmöglichkeiten folgen in Kürze (Abschnitt 3.6).

**Übung 2.3:** Ergänzen Sie den Übungstext der letzten Übung, so dass er deutlich mehr als eine Seite umfasst, und bearbeiten Sie diesen Text erneut. Ändern Sie anschließend den Befehl `\usepackage{german}` ab in `\usepackage{a4,german}`. Bearbeiten Sie Ihren Text mit dieser Änderung nochmals. Vergleichen Sie den neuen Ausdruck bezüglich der Seitenabmessungen und der Zeilenumbrüche mit dem alten.

Anmerkung: Das in dieser Übung angeforderte Ergänzungspaket `a4.sty` gehört nicht zum Standardumfang von  $\text{\LaTeX}$ . Bei vielen Installationen existiert es jedoch und stellt für das Papierformat DIN A4 passende Textbreiten und -höhen ein. Bleibt die Bearbeitung mit der Fehlermeldung stehen, dass das File `a4.sty` nicht gefunden wird, so ist die Bearbeitung zunächst abzubrechen. Dies kann durch Eingabe von `X` oder bei vielen Betriebssystemen mit der Tastenkombination `Ctrl C`, also dem gleichzeitigen Betätigen der Umschalttaste Ctrl oder Strg und des `C`, erreicht werden. Anschließend sollte der Programmtext

```
\if@twocolumn \textwidth175mm \marginparsep2.5mm
\oddsidemargin-7.9mm \evensidemargin-7.9mm
\else \ifcase \@ptsize\relax
\textwidth145mm \oddsidemargin7.1mm \evensidemargin7.1mm \or
\textwidth154mm \oddsidemargin2.6mm \evensidemargin2.6mm \or
\textwidth164mm \oddsidemargin-2.4mm \evensidemargin-2.4mm \fi
\marginparsep4mm \fi
\ifcase \@ptsize\relax \textheight 59\baselineskip \or
\textheight 52\baselineskip \or
\textheight 49\baselineskip \fi
\addtolength{\textheight}{\topskip} \topmargin-10mm
```

mit dem Editor eingegeben und unter dem Namen `a4.sty` abgespeichert werden. Die Bedeutung der hier angegebenen Befehlsfolgen kann zum gegenwärtigen Zeitpunkt übergangen werden. Danach kann die Übung wiederholt werden, bei deren Bearbeitung dann dieses File als Ergänzungspaket eingelesen wird. Es stellt für die wichtigsten Bearbeitungsmodi die Einstellwerte für das Papierformat DIN A4 bereit. Ein umfassenderes Formatierungsfile für DIN A4 ist in [5c, Abschn. 6.2.1] abgedruckt.

$\LaTeX 2_\epsilon$  erlaubt die Optionsangabe `a4paper` im `\documentclass`-Befehl. Die damit eingestellte Textbreite und Texthöhe entspricht jedoch nicht den Werten, die die Mehrzahl der Anwender mit dem Papierformat DIN A4 verknüpft (s. auch Fußnote 2 auf S. 6).

## 2.8 Anmerkungen zur $\LaTeX$ -Philosophie

Mit den soeben durchgeführten Übungen hat sich der Leser mit dem Bearbeitungsaufwurf und der Druckausgabe einer  $\LaTeX$ -Bearbeitung vertraut gemacht. Beim Vergleich der Texteingabe mit dem ausgedruckten Text wird der Unterschied eines Formatierungsprogramms gegenüber einer *ausdrucksorientierten* Bearbeitung eines WYSIWYG-Programms (*What you see is what you get*) deutlich. Während bei Letzterem der auf dem Bildschirm abgebildete Text identisch – bis auf Unterschiede in der Druckqualität – mit dem ausgedruckten Text ist, sind Bildschirmeingabe- und Druckerausgabertext für die  $\LaTeX$ -Bearbeitung zwei völlig unterschiedliche Strukturen.

Dies gilt selbst für einen so einfachen Eingabetext, wie er bei der letzten Übung verwendet wurde. Der Zeilen- und Seitenumbruch erfolgt unabhängig von dem auf dem Bildschirm erscheinenden Eingabetext und die beim Ausdruck auftretenden Seitennummern waren bei der Eingabe nicht einmal erwähnt worden. Ebenso wurde bei der Eingabe zur Kennzeichnung neuer Absätze nichts über die Einrücktiefe der ersten Absatzzeile gesagt. Zeilenbreite und Seitenhöhe wurden bei der ersten Bearbeitung mit voreingestellten Werten gewählt, die bei der zweiten Bearbeitung mit der Option `a4` an das Papierformat DIN A4 angepasst wurden.

Das gesamte visuelle Layout auf den ausgegebenen Seiten wurde allein durch die Einstellparameter beim Befehl `\documentclass` und seinen evtl. `\usepackage`-Ergänzungen bestimmt. Komplexere Texte enthalten üblicherweise weitere Formatierungseinstellungen. Diese werden entweder über weitere Ergänzungspakete mit `\usepackage` oder über explizite Erklärungen (s. Abschn. 2.3) im Vorspann ausgewählt und wirken dann auf den gesamten nachfolgenden Text. Durch Änderung der globalen Einstellparameter kann der gleiche nachfolgende Eingabetext in völlig unterschiedlicher Form für die Ausgabe bearbeitet werden.

Der eigentliche Eingabetext enthält üblicherweise weitere *logische* Strukturangaben wie Gliederungsanweisungen für Kapitel, Abschnitte usw. und deren Überschriften. Anweisungen für Auswahl und Größe der Schriften dieser Überschriften und deren Anordnung zum umgebenden Text entfallen dagegen. Es ist gerade die Aufgabe des Formatierungsprogramms, die Umsetzung der logischen Eingabe- in die visuellen Ausgabestrukturen auf Grund der Layout- oder expliziten Formatierungsvorgaben vorzunehmen und den Autor oder die Schreibkraft hiervon freizustellen und damit zu entlasten.

Auch solche visuellen Strukturen wie Blockeinrückungen, Listen, Tabellen, Fußnoten und sogar mathematische Formeln werden bei der Eingabe durch logische Strukturangaben gekennzeichnet, ohne dass deren typographische Umsetzung für die konkrete Druckausgabe anzugeben ist. Diese erfolgt durch interne Programmvorgaben mit den sog. Stilfiles, deren Einstellwerte durch eigene Erklärungen zum Teil verändert werden können.

Neben den logischen Strukturangaben kennt  $\LaTeX$  auch explizite Anweisungen für lokale Layoutänderungen, wie z. B. die Änderung der Schriftart und Schriftgröße oder des Zeilenumbruchs, z. B. für zentrierten Zeilentext. Die nachfolgenden Kapitel 3–8 beschreiben das gesamte Gestaltungsspektrum, das  $\LaTeX$  für die Textformatierung bereitstellt.

# Kapitel 3

## Dokumentklassen und Seitenstil

### 3.1 Die Dokumentklasse

Der erste Befehl im Vorspann eines  $\LaTeX$ -Files legt die globale Bearbeitungsklasse für das ganze Dokument fest. Die Syntax für diesen Befehl lautet

```
\documentclass [optionen] {klasse} [vers_datum]
```

Die zulässigen Angaben für *klasse*, *optionen* und *vers\_datum* werden in den nachfolgenden Unterabschnitten vorgestellt.

#### 3.1.1 Dokument-Standardklassen

Für *klasse* stehen beim `\documentclass`-Befehl standardmäßig folgende Parameter zur Verfügung, von denen genau einer gewählt werden muss: `article`, `report`, `book`, `letter`, `proc` oder `slides` sowie `ltxdoc`, `ltxguide`, `ltxnews` oder `minimal`.

Die Klassenauswahl `article` ist für die Bearbeitung von Artikeln geeignet, die ihrerseits ggf. in fortlaufende Abschnitte und Unterabschnitte untergliedert werden sollen. Für längere Berichte, die in Kapitel mit eigenen Abschnitten und Unterabschnitten untergliedert werden sollen, ist `report` vorgesehen. Zur Bearbeitung eines Buches, das ebenfalls in Kapitel, Abschnitte, Unterabschnitte usw. untergliedert werden kann, steht `book` zur Verfügung. Kapitel beginnen in beiden Fällen stets mit einer neuen Seite.

Die Bearbeitungsklasse `letter` ist zur Bearbeitung von Brieftexten gedacht. Sie ist in der Originalversion auf eine US-typische Briefform zugeschnitten, die bei uns ungebräuchlich ist. Eine Anpassung an deutsche Briefformen verlangt Änderungen und Ergänzungen beim zugehörigen Klassenfile `letter.cls`. Die Beschreibung der Eigenschaften der `letter`-Klasse erfolgt daher erst im Anhang A. Dort werden auch die Änderungen beschrieben, um gleichzeitig einen persönlichen oder firmenspezifischen Briefkopf zu erstellen.

Die Bearbeitung von Sitzungsprotokollen erfolgt mit der Bearbeitungsklasse `proc`. Ihre Eigenschaften werden in 3.3.6 vorgestellt. Die Bearbeitungsklasse `slides` dient zur Erstellung ein- oder mehrfarbiger Folienvorlagen. Ihre Eigenschaften werden im Anhang E vorgestellt.

Die unterschiedlichen Klassenangaben beim `\documentclass`-Befehl werden durch gleichnamige Klassenfiles realisiert. Ihr Grundname entspricht dem Klassennamen, der mit dem Anhang `.cls` gekennzeichnet ist.

Die Bearbeitungsklassen `ltxdoc`, `ltxguide` und `ltxnews` dienen zur Formatierung und Aufbereitung von  $\LaTeX$  2<sub>ε</sub>-Dokumentations- oder -Quellenfiles. Sie kommen für anwendereigene `\documentclass`-Befehle kaum in Betracht. Die erwähnten  $\LaTeX$  2<sub>ε</sub>-Dokumentations- oder -Quellenfiles enthalten bereits den erforderlichen `\documentclass`-Befehl mit der jeweils geeigneten Klassenangabe. Ihre  $\LaTeX$ -Bearbeitung erstellt wohlformatierte Dokumentationen oder Handbücher ohne zusätzliche Anwendervorgaben. Die meisten dieser Dokumentationen oder Handbücher sind nur für Anwender von Nutzen, die in die Details der internen  $\LaTeX$ -Strukturen Einblick gewinnen wollen. Von einem kurzen Hinweis im Anhang F abgesehen, stelle ich sie deshalb erst in [5b] und [5c] vor.

Die letzte der aufgezählten Bearbeitungsklassen `minimal` greift auf ein Klassenfile `minimal.cls` zurück, das nur die Minimalvorgaben für ein Klassenfile enthält. Dieses ist als Prototyp zur Entwicklung eigener Klassenfiles gedacht, worauf erst in [5c] eingegangen wird. Eine alleinige Nutzung mit `\documentclass{minimal}` ist mit dem derzeitigen Kenntnisstand des Lesers wenig sinnvoll.

### 3.1.2 Klassenoptionen

Die Klassenauswahlbefehle `\documentclass` gestatten die Angabe von *Optionen*, mit denen weitere oder geänderte Bearbeitungseigenschaften eingestellt werden. Die zulässigen Optionen sind in der nachfolgenden Liste in Gruppen zusammengefasst, wobei die Optionen einer Gruppe durch senkrechte Striche `|` voneinander getrennt werden. Von den zulässigen Optionen einer Gruppe darf jeweils *höchstens* eine angegeben werden. Optionsangaben aus unterschiedlichen Gruppen dürfen dagegen beliebig miteinander kombiniert werden.

`10pt | 11pt | 12pt` Die Auswahl der Standardschriftgröße als 10 pt-, 11 pt- oder 12 pt-Schrift. Ohne Angabe einer Größenoption wird standardmäßig 10pt gewählt. Die explizite Optionsangabe `10pt` ist erlaubt, entfällt aber in den meisten Fällen, da dies die Standardvorgabe ist, die auch ohne Angabe einer Größenoption gewählt wird. Mit der Schriftgrößenoption wird die Schriftgröße für den laufenden Fließtext eingestellt. Dies ist gleichzeitig die Bezugsgröße für andere Schriftgrößen, z. B. bei den Gliederungsüberschriften oder in Fußnoten, die  $\LaTeX$  bei der Bearbeitung eines Dokuments automatisch auswählt.

`onecolumn | twocolumn` Die Formatierung der einzelnen Seiten erfolgt *ein-* bzw. *zweispaltig*. Die Standardvorgabe ist `onecolumn`, d. h., die Formatierung erfolgt ohne Vorgabe der Spaltenoption ebenfalls einspaltig.

`oneside | twoside` Die Seitenformatierung erfolgt für *ein-* bzw. *doppelseitige* Ausgabe. Bei Letzterer werden die linksseitigen Ränder auf geraden und ungeraden Seiten unterschiedlich eingestellt, damit die Textränder auf Doppelseiten übereinstimmen. Bei den Klassen `article`, `letter` und `report` ist `oneside` die Standardvorgabe, die auch ohne Optionsangabe gewählt wird. Bei der Klasse `book` wird dagegen ohne explizite Optionsangabe standardmäßig `twoside` gewählt.

`notitlepage | titlepage` Bei den Bearbeitungsklassen `book` und `report` erscheint der Buch- oder Berichtstitel auf einer eigenen Seite, bei der Bearbeitungsklasse

`article` horizontal zentriert oberhalb des nachfolgenden Textes (s. 3.3.1). Dieses klassenabhängige Standardverhalten kann mit den vorstehenden Optionen abgeändert werden. `titlepage` würde auch bei der Bearbeitungsklasse `article` eine eigene Titelseite erzwingen und umgekehrt `notitlepage` bei `book` und `report` die Titelangaben zentriert über dem nachfolgenden Text anordnen.

`final` | `draft` Mit der Optionsangabe `draft` werden Zeilen, deren Umbruch nicht sauber gelingt und die deshalb etwas über den rechten Rand hinausragen, mit einem dicken schwarzen Randbalken gekennzeichnet. Der zusätzliche Randbalken entfällt bei der Option `final`, die gleichzeitig auch die Standardvorgabe darstellt.

`leqno` Die Formelnummern in abgesetzten Formeln erscheinen linksbündig statt sonst rechtsbündig (s. 5.1).

`fleqn` Abgesetzte Formeln werden nicht zentriert, sondern linksbündig mit einer wählbaren Einrücktiefe ausgegeben (s. 5.1 und 3.1.3).

`openbib` Ein evtl. Literaturverzeichnis erscheint in einer gegenüber dem Standard abgeänderten Form. Hierauf wird in 4.3.6 mit Übung 4.9 näher eingegangen.

Die vorstehenden Optionsangaben waren auch in  $\LaTeX$  2.09 bekannt und erlaubt, doch durften dort für die Standardvorgaben deren zugeordnete Optionen *nicht* explizit angegeben werden. Letzteres ist nunmehr erlaubt, wenn auch überflüssig, weil ohne diese explizite Angabe deren Auswahl ja standardmäßig erfolgt.

$\LaTeX$  kennt ab der Version 2 $\epsilon$  drei weitere Optionsgruppen für den `\documentclass`-Befehl. Dies sind zum einen die alternativen Papierformatoptionen, von denen höchstens eine angegeben werden darf:

<code>a4paper</code>	297 × 210 mm	<code>letterpaper</code>	11 × 8.5 Zoll
<code>a5paper</code>	210 × 148 mm	<code>legalpaper</code>	14 × 8.5 Zoll
<code>b5paper</code>	250 × 176 mm	<code>executivepaper</code>	10.5 × 7.25 Zoll

Standardmäßig, also ohne explizite Vorgabe für das Papierformat, wird `letterpaper` eingestellt. Die Wirkung der Formatangabe `a4paper` wird mit Übung 3.3 nachgereicht. Die Einstellung der Papiergröße kann ergänzt werden durch die Forderung nach Formatierung im Querformat. Die entsprechende Optionsangabe lautet `landscape`. Sie bewirkt eine Vertauschung von Seitenhöhe und Seitenbreite bei den vorangegangenen Papierformatoptionen.<sup>1</sup>

Bei der  $\LaTeX$ -Bearbeitungsklasse `book` beginnt ein neues Kapitel standardmäßig mit einer rechten (ungeraden) Seite. Endet das vorangehende Kapitel ebenfalls mit einer ungeraden Seite, so wird eine *leere* gerade Seite eingefügt. Mit dem alternativen Optionspaar

`openright` | `openany` kann dieses Verhalten abgeändert werden. Mit der Optionsangabe `openany` im `\documentclass`-Befehl wird dieses Verhalten abgeschaltet. Ein neues Kapitel beginnt dann stets auf der nächsten Seite, unabhängig davon, ob diese eine linke oder rechte Seite ist. Dies ist gleichermaßen auch das Standardverhalten bei der Bearbeitungsklasse `report`. Hier kann umgekehrt mit der Optionsangabe `openright` erzwungen werden, dass ein neues Kapitel stets mit einer rechten, ungeradzahligen Seite beginnt.

---

<sup>1</sup>Die Optionsangabe `landscape` führt bei der Druckerausgabe nicht automatisch zu der dann erforderlichen Drehung der Druckzeichen um 90°. Diese muss durch eine entsprechende Optionsangabe beim Aufruf des Druckertreibers aktiviert werden. Wie sie dort lautet, muss dem Treibermanual entnommen werden.

Mehrere zulässige Optionen innerhalb des `\documentclass`-Befehls werden durch Kommata, ohne zusätzliche Leerzeichen, voneinander getrennt. Damit sind

```
\documentclass[12pt,leqno,a4paper,draft]{article} oder
\documentclass[11pt,twocolumn,openright]{report}
```

zulässige Aufrufe. Die Reihenfolge der Optionsangaben ist dabei gleichgültig. Entfallen Optionsangaben vollständig, so können auch die umschließenden eckigen Klammern fortgelassen werden. Dies gilt, wie soeben demonstriert, auch für die zweite Optionsangabe [*vers\_datum*] beim `\documentclass`-Befehl. Hier darf eine Datumsangabe der Form [*jahr/monat/tag*] mit der Syntax [*yyyy/mm/dd*] stehen. Liegt das Erstellungsdatum für das angeforderte Klassenfile vor dem angegebenen Datum, so führt dies beim  $\text{\LaTeX}$ -Programmaufruf zu einer entsprechenden Warnung auf dem Bildschirm.

Die gewählten Optionen werden, soweit die erforderlichen internen Befehlsstrukturen nicht bereits mit den Klassenfiles bereitgestellt werden, in  $\text{\LaTeX}$  durch zusätzliche Klassenoptionsfiles realisiert, die durch den Anhang `.clo` gekennzeichnet sind. Standardmäßig sind dies `bk10.clo`, `bk11.clo` und `bk12.clo` zur Realisierung der Größenoption bei der Bearbeitungsklasse `book` bzw. `size10.clo`, `size11.clo` und `size12.clo` bei den Bearbeitungsklassen `article`, `report` und `proc` sowie `leqno.clo` und `fleqn.clo` zur Realisierung der gleichnamigen Optionen `leqno` bzw. `fleqn`. Für alle anderen Klassenoptionen sind die erforderlichen Befehlsstrukturen bereits in den Klassenfiles eingebaut, die mit der entsprechenden Optionsangabe aktiviert werden.

**Übung 3.1:** Ändern Sie im File der Übung 2.3 den dortigen `\documentclass`-Befehl zunächst in `\documentclass[11pt]{article}` und drucken Sie das Ergebnis nach der  $\text{\LaTeX}$ -Bearbeitung aus. Wiederholen Sie die Bearbeitung anschließend mit der Option `12pt` und vergleichen Sie beide Ausdrücke in Bezug auf den vorgenommenen Zeilenumbruch mit dem der Übung 2.3.

**Übung 3.2:** Verwenden Sie nun `\documentclass[twocolumn]{article}` in Ihrem Übungsfile. Falls während der  $\text{\LaTeX}$ -Bearbeitung nun Underfull `\hbox ...`-Warnungen auf dem Bildschirm erscheinen, so werden die entsprechenden Zeilen zwar rechtsbündig umbrochen, die Wörter der entsprechenden Spaltenzeilen sind jedoch eventuell zu weit auseinander gezogen. Prüfen Sie beim Ausdruck, ob die Wortabstände in diesen Zeilen noch tolerabel sind. Ist dies nicht der Fall, so hilft häufig eine Trennungsvorgabe (s. 3.6) in den darauffolgenden Wörtern der nächsten Zeile.

Anmerkung zu den Bearbeitungsklassen `book` und `report`: Wenn Sie bei den bisherigen Übungen die Klassenangabe `book` oder `report` statt `article` wählen, so werden Sie im Ausdruck keine Unterschiede feststellen. Diese werden erst bei späteren Strukturelementen wirksam.

### 3.1.3 Einstellparameter für einige der vorstehenden Klassenoptionen

Zu einigen Optionen gehören Längenerklärungen, mit denen bestimmte Parameterwerte gesetzt werden:

- `\mathindent` gibt an, wie weit die mit der Option `fleqn` linksbündig angeordneten Formeln nach rechts eingerückt sind (s. 5.1).
- `\columnsep` bestimmt den Abstand zwischen den beiden Spalten für die Klassenoption `twocolumn` (s. S. 489).
- `\columnseprule` legt die Breite der vertikalen Linie zwischen den beiden Spalten für die Klassenoption `twocolumn` fest. Der Standard ist Null und entspricht einer unsichtbaren vertikalen Linie (s. S. 489).

Die Zuweisung von Werten erfolgt als Längenerklärung gemäß 2.4 mit dem  $\LaTeX$ -Befehl `\setlength`:

```
\setlength{\mathindent}{2.5cm} oder \setlength{\columnsep}{5pt}
```

Diese Erklärungen können sowohl im Vorspann als auch an beliebigen Stellen im Dokument gesetzt werden. Erklärungen im Vorspann gelten für das ganze Dokument. Entsprechende Erklärungen im Text gelten nur bis zur nächsten Änderung, längstens aber bis zum Ende der augenblicklichen Umgebung, in der diese Erklärungen gesetzt werden (s. 2.3). Danach gelten wieder die vorangehenden Erklärungen.

### 3.1.4 Ergänzungspakete

Ergänzungspakete sind  $\LaTeX$ -Makrosätze, mit denen über die Optionsmöglichkeiten der Bearbeitungsklassen hinaus weitere Bearbeitungseigenschaften geändert oder bereitgestellt werden. Ergänzungspakete werden durch Files mit dem Grundnamen des Pakets und dem Anhang `.sty` bereitgestellt und mit dem Vorspannbefehl

```
\usepackage[optionen]{erg_paket}[vers_datum]
```

aktiviert. Der Befehl wurde als *Vorspannbefehl* bezeichnet, weil er nur im Vorspann, also nur *nach* `\documentclass` und *vor* `\begin{document}` erlaubt ist. Er wurde bereits in den Übungen 2.1 bis 2.3 zur Aktivierung von `german.sty` und `a4.sty` verwendet, ohne dort näher vorgestellt zu werden. Die optionale Angabe `vers_datum` hat dieselbe Wirkung und Syntax wie die gleichnamige Angabe beim `\documentclass`-Befehl: Hat das angesprochene Ergänzungspaket ein älteres Erstellungsdatum als das angeforderte Versionsdatum, so führt das beim Programmaufruf zu einer Bildschirmwarnung.

Für `erg_paket` ist nur der Grundname des angeforderten Ergänzungspakets anzugeben. Mit einem `\usepackage`-Befehl können mehrere Ergänzungspakete angefordert werden, wie dies bereits bei Übung 2.3 mit der Anforderung `\usepackage{a4,german}` zur Aktivierung von `a4.sty` und `german.sty` geschah. Bei der Grundinstallation von  $\LaTeX$  ( $\LaTeX 2_{\epsilon}$ ) entstehen als Ergänzungspakete insgesamt

<code>alltt.sty</code>	<code>doc.sty</code>	<code>exscale.sty</code>	<code>flafter.sty</code>
<code>fontenc.sty</code>	<code>graphpap.sty</code>	<code>ifthen.sty</code>	<code>inputenc.sty</code>
<code>latexsym.sty</code>	<code>makeidx.sty</code>	<code>newlfont.sty</code>	<code>oldlfont.sty</code>
<code>pict2e.sty</code>	<code>shortvrb.sty</code>	<code>showidx.sty</code>	<code>syntonly.sty</code>
<code>tlenc.sty</code>	<code>textcomp.sty</code>	<code>tracefnt.sty</code>	

die im weiteren Verlauf dieses Buches vorgestellt werden und die alle mit entsprechenden `\usepackage`-Befehlen im Vorspann zu aktivieren sind. Bei der  $\LaTeX$ -Gesamtinstallation entstehen evtl. weitere Ergänzungspakete. Diese gehen über die  $\LaTeX$ -Standardeigenschaften hinaus. Sie werden deshalb erst in Band 2 [5b] vorgestellt.

Im `\usepackage`-Befehl steht *optionen* für eventuelle Optionsangaben. Sie entfalten Wirkung nur dann, wenn das angeforderte Ergänzungspaket auf solche Optionsangaben vorbereitet ist. Unbekannte Optionsangaben bleiben wirkungslos. Bei der Beschreibung der aufgezählten Ergänzungspakete wird angegeben, ob das entsprechende Paket Optionen kennt und was damit erreicht wird.

Optionsangaben innerhalb von `\usepackage`-Befehlen entfalten eine Wirkung nur auf die im gleichen Befehl angeforderten Ergänzungspakete. Sie bleiben wirkungslos für Ergänzungspakete, die mit weiteren `\usepackage`-Befehlen angefordert werden. Optionsangaben in `\usepackage`-Befehlen wirken somit nur *lokal* auf die angeforderten Ergänzungspakete. Optionsangaben aus dem `\documentclass`-Befehl können dagegen Wirkung auf alle mit anschließenden `\usepackage`-Befehlen angeforderten Ergänzungspakete entfalten. Optionsangaben aus dem `\documentclass`-Befehl wirken somit evtl. auch *global* auf alle anschließend eingelesenen Ergänzungspakete.

### 3.1.5 L<sup>A</sup>T<sub>E</sub>X 2.09-Kompatibilitätsmodus

In 1.2 wurde bereits darauf hingewiesen, dass L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> über einen Kompatibilitätsmodus zu L<sup>A</sup>T<sub>E</sub>X 2.09 verfügt, der sicherstellt, dass ältere L<sup>A</sup>T<sub>E</sub>X-Eingabetexte, die zur Bearbeitung mit L<sup>A</sup>T<sub>E</sub>X 2.09 vorgesehen waren, auch mit der aktuellen L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Version bearbeitet werden können und dabei so formatiert werden, wie dies unter L<sup>A</sup>T<sub>E</sub>X 2.09 der Fall gewesen wäre. Zur Umschaltung in den 2.09-Kompatibilitätsmodus muss man wissen, dass in L<sup>A</sup>T<sub>E</sub>X 2.09 an Stelle der Auswahl einer Bearbeitungsklasse dort die Auswahl eines Bearbeitungsstils stand. Dieser erfolgt mit dem Eröffnungsbefehl

```
\documentstyle [optionen] {stil}
```

Als Stilnamen und damit als Bearbeitungsstile konnten standardmäßig `article`, `report`, `book` und `letter` gewählt werden. Diese Bearbeitungsstile entsprechen in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> den dort gleichnamigen Bearbeitungsklassen. Die weiteren Bearbeitungsklassen aus L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> gemäß 3.1.1 sind in L<sup>A</sup>T<sub>E</sub>X 2.09 als äquivalente Bearbeitungsstile *nicht* bekannt.

Als Stiloptionen können im `\documentstyle`-Befehl gewählt werden:

```
11pt|12pt, twoside, twocolumn, titlepage, leqno und fleqn
```

Die Wirkung dieser Stiloptionen entsprechen den gleichnamigen Optionen beim `\documentclass`-Befehl aus L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Die Alternativoptionen `10pt`, `oneside`, `onecolumn` und `notitlepage` aus L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> stehen in L<sup>A</sup>T<sub>E</sub>X 2.09 nur implizit zur Verfügung, da diese standardmäßig eingestellt werden, wenn die entsprechenden Alternativoptionen (`11pt|12pt`, `twoside`, `twocolumn` bzw. `titlepage`) im `\documentstyle`-Befehl entfallen. Ihre explizite Angabe für *optionen* ist nicht erlaubt.

Trifft L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> bei der Bearbeitung eines Eingabefiles auf den `\documentstyle`-Befehl statt auf `\documentclass`, so entnimmt es hieraus, dass ein Textfile vorliegt, das so bearbeitet werden soll, wie dies mit L<sup>A</sup>T<sub>E</sub>X 2.09 geschehen würde. Damit wird die Bearbeitungskompatibilität bisheriger Texte sichergestellt. Ein solcher Eingabetext darf keine Befehlsstrukturen enthalten, die nur mit L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> bereitgestellt werden, was für ältere Eingabetexte, die zur Bearbeitung mit L<sup>A</sup>T<sub>E</sub>X 2.09 vorgesehen waren, vermutlich auch nicht der Fall ist.

Neue Eingabetexte zur Bearbeitung im L<sup>A</sup>T<sub>E</sub>X 2.09-Kompatibilitätsmodus sollten dagegen nicht mehr erstellt werden, da dies die Kenntnis der Detailunterschiede zwischen L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> und L<sup>A</sup>T<sub>E</sub>X 2.09 verlangen würde, worauf ich in dieser Einführung nicht mehr eingehe, da L<sup>A</sup>T<sub>E</sub>X 2.09 inzwischen als obsolet gilt. Für detailversessene Anwender verweise ich bei Bedarf auf den Befehlsindex dieses Buches mit einer Kurzbeschreibung aller L<sup>A</sup>T<sub>E</sub>X-Befehle. In diesem Befehlsindex sind die Befehle, die zwischen L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> und L<sup>A</sup>T<sub>E</sub>X 2.09 unterschiedlich verfügbar sind oder dort unterschiedliche Wirkung entfalten, mit den vorangestellten Markierungen als 2<sub>ε</sub> bzw. 2.09 gekennzeichnet.

Auf einen der Hauptunterschiede zwischen  $\LaTeX 2_\epsilon$  und  $\LaTeX 2.09$  möchte ich hier aber doch hinweisen. Dieser betrifft die Methoden der Schriftauswahl. Die Charakterisierung und damit Auswahl von Schriften durch voneinander unabhängige Schriftattribute in  $\LaTeX 2_\epsilon$ , die in 4.1.3 vorgestellt werden und die dem Anwender als ganz natürliche Auswahl schemata erscheinen, sind in  $\LaTeX 2.09$  vollkommen unbekannt. Dort stehen nur die sog. Zeichensatz-Auswahlbefehle in Form der Erklärungen

<code>\rm</code>	Roman	<code>\it</code>	<i>Italic</i>	<code>\sc</code>	SMALL CAPS
<code>\bf</code>	<b>Bold Face</b>	<code>\sl</code>	<i>Slanted</i>	<code>\sf</code>	Sans Serif
<code>\tt</code>	Typewriter	<code>\em</code>	<i>Hervorhebung</i>	Hervorhebung	

bereit, denen entsprechende Zeichenätze jeweils fest zugeordnet sind.  $\LaTeX$ -Neueinsteiger mögen diese Kurzvorstellung der  $\LaTeX 2.09$ -Zeichensatzauswahlbefehle, die dadurch gekennzeichnet sind, dass ihre Befehlsnamen nur aus zwei Buchstaben bestehen, die in Kurzform die Haupteigenschaft der betreffenden Schrift charakterisieren, nur als historische Reminiszenz betrachten und diese Zweibuchstaben-Auswahlbefehle selbst nie verwenden.

$\LaTeX 2.09$  kennt keinen zu `\usepackage` äquivalenten Befehl. Seine Ergänzungspakete werden durch Angabe des zugehörigen Grundnamens als Option im `\documentstyle`-Befehl aktiviert. Alle Ergänzungspakete aus einer früheren  $\LaTeX 2.09$ -Version können auf diese Weise auch von  $\LaTeX 2_\epsilon$  im Kompatibilitätsmodus genutzt werden.

Die Klassen- und Klassenoptionsfiles, die in  $\LaTeX 2_\epsilon$  durch die Anhänge `.cls` bzw. `.clo` gekennzeichnet sind, werden in  $\LaTeX 2.09$  durch sog. Stilfiles mit den gleichen oder sinnverwandten Grundnamen, aber dem Anhang `.sty` realisiert. Gleichnamige Files mit dem Anhang `.sty` entstehen auch bei der  $\LaTeX 2_\epsilon$ -Installation. Sie sind hier jedoch nahezu leer und lesen ihrerseits nur das entsprechende `.cls`-File ein. Sie dienen in  $\LaTeX 2_\epsilon$  nur zur Sicherung des Kompatibilitätsmodus.

## 3.2 Der Seitenstil

Mit dem Seitenstil wird der grundsätzliche Aufbau einer Seite bestimmt. Er wird, von einer Ausnahme abgesehen, im Vorspann festgelegt. Der Seitenstilbefehl lautet:

```
\pagestyle{stil}
```

An zwingenden Parametern *stil* stehen zur Verfügung:

- |                         |  |
|-------------------------|--|
| <code>plain</code>      | Der Seitenkopf ist leer, die Fußzeile besteht aus der zentrierten Seitennummer. Dies ist auch das Standardseitenformat, wenn der Befehl <code>\pagestyle</code> im Vorspann nicht auftaucht.   |
| <code>empty</code>      | Kopf- und Fußzeile bleiben leer, d. h., es wird auch keine Seitennummer ausgedruckt.   |
| <code>headings</code>   | Der Seitenkopf enthält die Seitenzahl sowie Überschriftinformationen, die durch die gewählte Bearbeitungsklasse bestimmt wird (im Allgemeinen die augenblickliche Kapitel- oder/und Abschnittsüberschrift, s. die Kopfzeilen dieses Buches). Die Fußzeile bleibt leer. Dies gilt nicht bei den Kapitelanfangsseiten! |
| <code>myheadings</code> | Wie <code>headings</code> , jedoch wird der Seitenkopf nicht automatisch, sondern durch die Erklärungen <code>\markright</code> bzw. <code>\markboth</code> (s. u.) bestimmt.  |

Der Befehl

```
\thispagestyle{stil}
```

entspricht genau dem Befehl `\pagestyle`, mit der Ausnahme, dass er sich nur auf die laufende Seite bezieht und somit im laufenden Text statt im Vorspann auftritt. Soll z. B. auf der laufenden Seite keine Seitennummer ausgedruckt werden, so kann dies mit `\thispagestyle{empty}` erreicht werden. Die unterdrückte Seitennummer wird jedoch für die Seitennummerierung der nachfolgenden Seiten mitgezählt.

### 3.2.1 Kopfdeklarationen

Für den Seitenstil `headings` und `myheadings` kann die Kopfinformation mit den Erklärungen

```
\markright{rechter Kopf} bzw. \markboth{linker Kopf}{rechter Kopf}
```

gesetzt werden. (Ihre Wirkung beginnt erst ab der zweiten Seite des Gesamttextes!)

Die Deklaration `\markboth` korrespondiert mit der Klassenoption `twoside`, wobei geradzahlige Seiten als *linke* Seiten mit dem *linken Kopf* und ungeradzahlige Seiten als *rechte* Seiten mit dem *rechten Kopf* versehen werden. Zusätzlich wird die Seitennummer auf linken Seiten linksbündig und auf rechten Seiten rechtsbündig in die Kopfzeile gesetzt.

Bei einseitig bedruckten Seiten gilt jede Seite als rechte Seite. Hierfür ist die Deklaration `\markright` geeignet. `\markright` kann aber auch bei doppelseitigem Ausdruck verwendet werden. Hierdurch wird der *rechte Kopf* in `\markboth` überschrieben.

Bei dem Seitenstil `headings` werden diese Deklarationen standardmäßig durch die folgenden Gliederungsbefehle (s. 3.3.3) mit ihren Überschriften gesetzt:

Druckstil	Befehl	Bearbeitungsklasse	
		book, report	article
doppelseitig	<code>\markboth{l}</code>	<code>\chapter</code>	<code>\section</code>
	<code>\markboth{r}</code>	<code>\section</code>	<code>\subsection</code>
einseitig	<code>\markright</code>	<code>\chapter</code>	<code>\section</code>

TLL
-----

Mit expliziten `\markboth`- und `\markright`-Erklärungen können die Standardvorgaben aus den Gliederungsbefehlen überschrieben werden. Soweit auf einer Seite mehrere Gliederungsbefehle `\section` oder `\subsection` (s. 3.3.3) stehen, wird für die Kopfzeile auf linken Seiten die jeweils letzte und auf rechten Seiten die erste Überschrift benutzt. Betrachten Sie die Kopfzeilen dieses Buches als Beispiel.

### 3.2.2 Seitennummerierung

Die Erklärung des Stils der Seitennummerierung lautet:

```
\pagenumbering{num_stil}
```

Als Nummerierungsstilarten `num_stil` stehen zur Verfügung:

- `arabic` für normale Nummerierung (mit arabischen Ziffern)
- `roman` für römische Kleinnummerierung (mit kleinen römischen Ziffern)
- `Roman` für römische Großnummerierung (mit großen römischen Ziffern)