



L

Tobias Berndt



L

Der typographische Einstieg



ADDISON-WESLEY

Mit kompletter
Satzumgebung T_EX Live





3 GNU Emacs – ein Editor

Emacs exists because I felt free to make individually useful small improvements on a path whose end was not in sight.

— RICHARD STALLMAN

QUERLESENDE EINSTEIGER MÖGEN SICH FRAGEN: Was ist ein Editor und warum ist diesem Thema ein ganzes Kapitel gewidmet? Und weil es gar nicht genug betont werden kann: Sie arbeiten zwar *mit* L^AT_EX, aber *im* Editor! Das heißt, von L^AT_EX sehen Sie im Grunde gar nichts, der Editor hingegen ist Ihr ständiger Begleiter. In ihm nämlich verfassen Sie den Text, hinterlegen die Satzanweisungen für L^AT_EX und können sich Ihre Textarbeit durch nützliche Modi zudem noch vereinfachen. Insofern ist Ihr Editor die eigentliche Arbeitsumgebung zum Erstellen von L^AT_EX-Quelltext, die daher auch die nötige Ausstattung mitbringen sollte, wie zum Beispiel:

- eine komfortable Oberfläche zur Texterfassung und zum Editieren
- eine Rechtschreibprüfung sowie eine Syntaxkontrolle
- eine Kennzeichnung der Satzanweisungen (Syntaxhighlighting)
- einige Tastenkürzel für oft verwendete L^AT_EX-Befehle
- einige Modi zur Textstrukturierung
- die Möglichkeit zur Kontrolle der Resultate

Ein Editor aus dem Profilager, der all diese Funktionen beherrscht, ist der Emacs. Natürlich gibt es noch eine Menge anderer guter Editoren. Und eben weil die Vielfalt recht groß ist, sehen Sie mir nach, wenn ich weiterführend nur noch den einen betrachte, von dem ich der Meinung bin, dass er – gerade auch zum Editieren von L^AT_EX-Quelltext – ein Idealwerkzeug ist.

Das Gnu (CONNOCHAETES TAURINUS) oder *wildebeest* ist eine afrikanische Antilope, die in Herden offene Savannen bewohnt. Circa 200 kg schwer und 1,30 m hoch, ist es ein kräftiger Vertreter seiner Art und wohl daher auch ein recht jähzorniges, bockiges Vieh.

Der Gnu Emacs (*CONNOCHAETES EDITORIS*) ist ein Texteditor und ähnlich seiner biologischen Verwandtschaft mitunter etwas ruppig und ungehobelt. Für gewöhnlich in unixartigen Gefilden beheimatet, dringt diese Spezies mittlerweile auch in andere Rechnerarchitekturen vor. Er repräsentiert die Spitze der Evolution in der Editorenwelt, bedrängt höchstens durch eine Spezies, den *VIMO NEANDERTHALENSIS*, kurz Vi genannt.

Was lässt sich sagen über den Emacs? Er ist weder besonders neu noch besonders hübsch. Und doch ist die Wahrscheinlichkeit groß, dass – wenn Sie sich auf ihn einlassen – er die nützlichste Software auf Ihrem Rechner werden könnte! Ich persönlich nutze ihn nun bereits seit beachtlichen Jahren und er hat mit mir stets treu und brav editiert, programmiert, gerechnet, wahlweise Schach oder Nethack gespielt, meine Termine verwaltet, war mein Kalender und Newsreader, kümmerte sich bis vor nicht allzu langer Zeit um meine Mails und ja, mit ihm »texte« ich bereits das eine oder andere Dokument. In all den Jahren hat er nicht ein einziges Mal gestreikt oder sonst irgendwelche Zicken gemacht.

Gut, anfänglich hätte ich ihn, wenn das möglich gewesen wäre, am liebsten angezündet und aus dem Fenster geworfen. Wie ich seinen Erfinder, Richard Stallman, nannte, kann hier nicht wiedergegeben werden, und hätte es die Möglichkeit gegeben, einer Software Schmerz zuzufügen . . . na ja, das ist alles lange her und nunmehr sind wir gute Freunde. Wenn ich also ein Programm in diesem Leben nicht mehr missen möchte, so ist das mein Emacs.

Natürlich ist auch dieses Buch allumfänglich in meinem Emacs entstanden und wenn Sie sich nun wundern, warum ich ständig von meinem Emacs spreche, dann ist es eben deshalb, weil ich ihn mittlerweile so an meine mir eigene Arbeitsweise angepasst habe, dass diese Phrase tatsächlich stimmt! Es gibt ihn übrigens in zwei Varianten: GNU Emacs und XEmacs. Warum es hier zwei Entwicklungszweige gibt, ist mir bis heute nicht so recht einsichtig. Der Name XEmacs ist verwirrend, täuscht er doch vor, dieser lief hier nur unter X¹, während GNU Emacs lediglich etwas für die Konsole sei. Doch das ist nicht der Fall, beide Versionen laufen sowohl unter X als auch unter der Konsole. Es mag das eine oder andere Paket geben, das es nur für die eine oder andere Art gibt, zudem ist XEmacs etwas speicherhungriger als GNU Emacs. Letzterer mag auch in aller Regel etwas konsequenter entwickelt sein. Doch welchen Sie schließlich benutzen möchten, ist im Grunde ganz egal. Da sich die beiden in den von uns zu betrachtenden Gefilden sowieso nicht unterscheiden, verzichte ich auch weiterhin auf jede Art Spezifikation und nenne ihn wie bisher schlicht Emacs.

1 Eine menügesteuerte Oberfläche für unixartige Systeme; ungefähr der gleiche Unterschied wie zwischen dos und Windows.

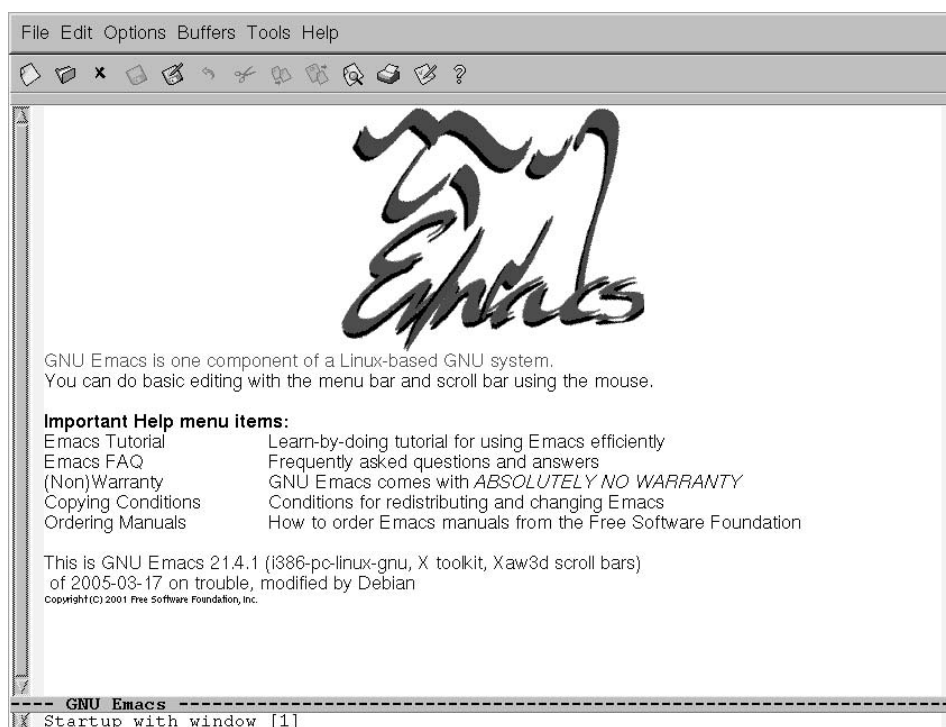


Abbildung 3.1: Das Emacs-Startfenster

Der Emacs, wie eine Reihe anderer Editoren (darunter auch der vi), lässt sich bequem über die T_EXLive-CD samt der Rechtschreibprüfung *Aspell* und der L^AT_EX-Schnittstelle auC_TE_X installieren.

3.1 Bedienung des Emacs

Gut, da Sie in diesem Abschnitt gelandet sind, müssen Sie wohl ernsthaft vorhaben, mit dem Emacs zu arbeiten. Eine sehr gute Wahl, wenn auch eine gewisse Zeit der Gewöhnung nötig sein wird.

Doch zur Dressur des *wildebeest*: Zwar lässt sich im Emacs vieles über das obere Menü mit der Maus steuern, doch empfiehlt es sich für ein effektives Arbeiten sehr, die entsprechenden Tasten-Pendants zu kennen. Grundsätzlich werden Sie es mit zwei Arten von Tastenkombinationen zu tun haben, nämlich **(Strg) + n** für normale Befehle und **(Esc) + n** für Anweisungen über den sogenannten Minipuffer.

Ein typisches Emacs-Kommando ist zum Beispiel **Strg** + **X** + **C** zum Schließen des Emacs. Dazu drücken Sie die **Strg**-Taste, halten diese gedrückt und drücken nacheinander die **C**- und die **X**-Taste. Es ist also nicht nötig, alle drei Tasten gleichzeitig gedrückt zu halten. Die Sache mit der Metataste funktioniert in gleicher Weise. Die Metataste variiert von System zu System, ist jedoch für gewöhnlich neben der Leertaste zu finden: Beim PC ist das die **Alt**-Taste, die **Esc**-Taste funktioniert bei allen Systemen als Metataste, also ganz gleich ob nun am PC, Mac oder einer Unix-Workstation.

Eine Besonderheit ist das Tastenkommando **Esc** + **X**. Dieses nämlich öffnet den bereits oben erwähnten Minipuffer, eine Befehlszeile im unteren Bereich des Editors. Über den Minipuffer können Kommandos, Modiwechsel, Konfigurationen und dergleichen abgesetzt werden. Um also beispielsweise das obere Kommando zum Schließen des Emacs über den Minipuffer anzuweisen, drücken Sie zunächst **Esc** + **X** (es erscheint ein *M-x* im Minipuffer und signalisiert Ihnen seine Bereitschaft, eine Anweisung entgegenzunehmen), schreiben in den Minipuffer die Anweisung `save-buffers-kill-emacs` und bestätigen das Ganze mit **↵**.

Sie starten den Emacs entweder über Ihr Systemmenü oder durch den Aufruf `emacs` in der Konsole.² Wenn Sie nun die **↵**-Taste betätigen, sehen Sie zunächst den *scratch*, also eine Art Schmierzettel. Dies ist ein überaus nützlicher Puffer³ für Notizen, Textproben oder Ähnliches, die Sie nicht speichern, aber eben auch nicht vergessen möchten. Und das sieht so aus:

```
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.
```

Hinterlegen Sie darin also alles, was temporär benötigt wird. Doch Achtung: Emacs fragt beim Schließen nicht nach einer Sicherung!

3.1.1 Grundlegende Kommandos

Folgen Sie diesem freundlichen Hinweis und erzeugen Sie doch einmal eine Datei namens *wolkow.txt*, indem Sie **Strg** + **X** + **F** drücken. Noch einmal: Sie drücken die **Strg**-Taste, halten diese gedrückt und tippen dann **X** und **F** hintereinander. Emacs

² Falls Sie eine Konsolenversion des Emacs unter X wünschen, erreichen Sie das mit dem Aufruf `emacs -nw` für *emacs - no windows*. Näheres dazu finden Anwender unixartiger Systeme in den Manpages: `man emacs`.

³ Eigentlich Puffer, Rahmen und Fenster, doch da Emacs Amerikaner ist, nennt er diese Sachen sowohl in den Anweisungen als auch in Hilfemenüs eben Buffer, Frames und Windows. Um die Verwirrung also in gewissen Grenzen zu halten, verfahren wir von nun an besser genauso. Zu all diesen Dingen gleich mehr.

fragt Sie nun in der kleinen unteren Zeile – dem Minibuffer – nach Pfad und Namen der Datei. Also gut, schreiben Sie: `wolkow.txt`

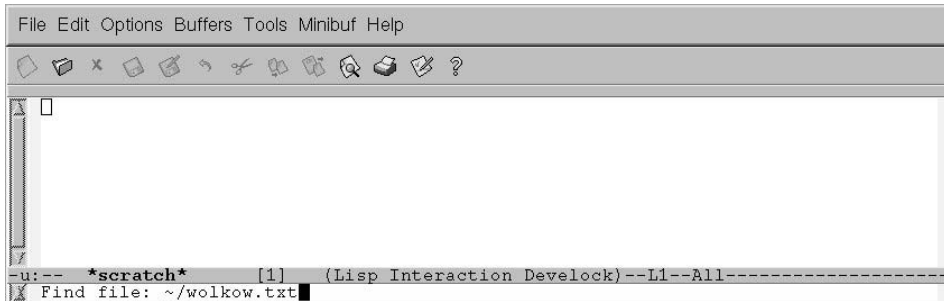


Abbildung 3.2: Das »Finden« (Anlegen, Öffnen) eines Buffers oder einer Datei

und drücken Sie `↵`. Damit öffnet sich ein Buffer, der auf Ihre Eingaben wartet. Geben Sie also etwas Text hinein. Die Zeilen werden vorerst nicht umbrochen, damit befassen wir uns aber gleich – einfach ein paar Zeilen schreiben ...

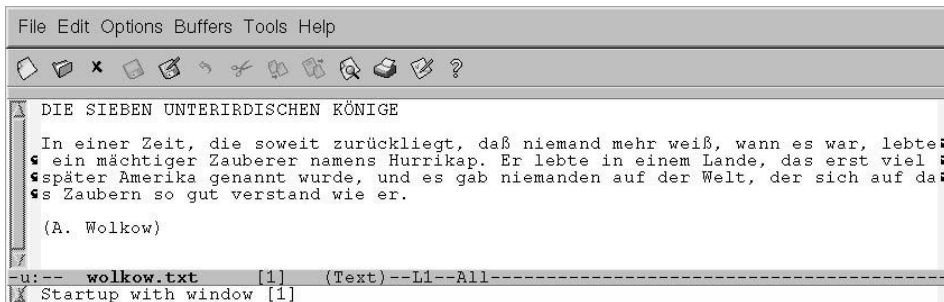


Abbildung 3.3: Der aktuelle Buffer

Im Moment haben Sie noch nichts weiter als einen beschriebenen Buffer – eine Datei existiert also noch nicht! Erst nach dem Speichern ist die Datei `wolkow.txt` vorhanden. Sie speichern Ihre Datei mit der Tastenkombination `(Strg) + (X) + (S)` und schließen selbige mit `(Strg) + (X) + (C)`. Damit verschwindet auch Emacs wieder. Und falls Sie bei all dem Getippe einmal die Übersicht verlieren sollten, auch kein Beinbruch, mit `(Strg) + (G)` brechen Sie das Ausführen von Befehlen ab und mit einem `(Strg) + (C)` machen Sie die letzte Aktion rückgängig.

Tastenkombination	Kommando	Aktion
<code>Strg</code> + <code>X</code> + <code>F</code>	<code>find-file</code>	lädt eine Datei oder (auch leeren) Buffer.
<code>Strg</code> + <code>X</code> + <code>S</code>	<code>save-buffer</code>	speichert die aktuelle Datei.
<code>Strg</code> + <code>X</code> + <code>C</code>	<code>save-buffer-kill-emacs</code>	schließt die aktuelle Datei und Emacs.
<code>Strg</code> + <code>-</code>		macht die letzte Aktion wieder rückgängig.
<code>Strg</code> + <code>G</code>	<code>cancel-command</code>	unterbricht die Befehlsausführung.

Tabelle 3.1: Elementare Kommandos

Sämtliche Tastenkommandos können auch dem Minibuffer – der unteren Zeile – als Anweisung übergeben werden. Doch geschieht auch dies wiederum unter Verwendung eines Shortcuts. Da Sie dadurch Ihren Aufwand lediglich multiplizieren, ist die Kommandofunktion des Minibuffers eher dafür geeignet, sich von Emacs ein wenig auf die Sprünge helfen zu lassen, wenn man einmal eine Befehlsstruktur vergessen hat.

Diese Gedächtnisstütze wird durch die `(S)`-Taste aufgerufen. Möchten Sie zum Beispiel über den Minibuffer Ihr Dokument speichern, so erreichen Sie das durch den Befehl `(Esc) + (X) save-buffer`. Nehmen wir also an, *save* fällt Ihnen gerade noch ein, aber das, was zu sichern ist, haben Sie vergessen. Genau hier hilft Ihnen die `(S)`-Taste! Nach `(Esc) + (X) save` drücken Sie also `(S)`. Daraufhin öffnet Emacs einen weiteren Buffer und zeigt alle möglichen Ergänzungen dieser Anweisung – unter anderem eben *save-buffer*. Ergo: Sie ergänzen Ihre Anweisung, bestätigen mit `(↵)`, der Hilfsbuffer verschwindet wieder und das Kommando wird ausgeführt.

3.1.2 Organisation des Bildschirms

Sofern Sie das letzte Beispiel nachvollzogen haben, ist Ihnen sicher aufgefallen, dass Emacs den Bildschirm kurz teilte, um Ihnen Vorschläge zur Befehlsergänzung anzubieten. Das können Sie als Benutzer freilich auch nutzen. Daher möchte ich Ihnen noch ganz kurz erklären, wie Emacs den Bildschirm organisiert, bevor wir zum Editieren von Dateien kommen.

Emacs unterscheidet zwischen den bereits erwähnten Buffern, Windows und Frames. Buffer sind dabei eine Art Platzhalter für Dateien, Notizen, Programmcodes – was Sie wollen. Erst wenn Sie einen Buffer speichern, ist er als Datei vorhanden. Schließen Sie Emacs ohne zu speichern, gibt er noch einmal Bescheid, dass da noch ungesicherte Buffer vorhanden sind. Verweigern Sie jedoch eine Speicherung, gehen die Inhalte der Buffer nach Beendigung des Editors verloren. Im Grunde nicht anders als bei jeder Textverarbeitung auch, nur dass hier das Ganze einen Namen hat.

Sofern Sie nur ein Fenster benutzen, liegen geladene Buffer »hintereinander«. Sie können sich mit einem Mausklick auf **BUFFERS** kurz davon überzeugen: Dort ist nämlich noch immer Ihr Schmierzettel, der *scratch*, aktiv (ist er übrigens immer), und klicken Sie ihn an, so erscheint nun er im einzig geöffneten Fenster. Dafür ist dann Ihre gerade bearbeitete Datei »dahinter«. Natürlich lässt sich die Navigation durch die Buffer auch mit der Tastatur bewerkstelligen.

Tastenkombination	Kommando	Aktion
<code>Strg + X B</code>	<code>switch-to-buffer</code>	wechselt zum gewählten Buffer.
<code>Strg + X + B</code>	<code>list-buffer</code>	zeigt alle Buffer in Liste an.
<code>Strg + X K</code>	<code>kill-buffers</code>	löscht aktuellen Buffer.

Tabelle 3.2: Bufferbefehle

Die Fenster entstehen durch eine Aufteilung der Arbeitsfläche des Bildschirms. Dabei können Bildschirminhalte, also Buffer, sowohl vertikal als auch horizontal in Fenster gesplittet werden. Nehmen wir also an, Sie hätten zwei Dateien und möchten beide gleichzeitig auf dem Bildschirm anzeigen, um vielleicht bestimmte Passagen zu vergleichen, kopieren, löschen oder dergleichen. Öffnen Sie also einfach die *wolkow.txt* im Emacs, Sie wissen schon: `Strg + X + F`. Die Datei erscheint auf dem Bildschirm. Nun teilen Sie Ihren Bildschirm horizontal: `Strg + X 2`. Die Datei zeigt sich in beiden Fenstern. Wechseln Sie durch Mausklick oder durch ein `Strg + X O` in das untere Fenster und öffnen Sie in diesem eine andere Datei. Voilà!

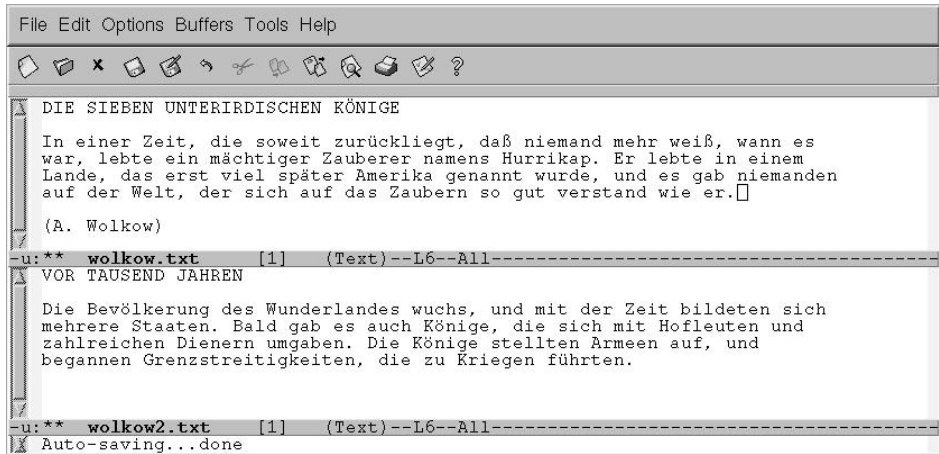


Abbildung 3.4: Horizontale Fensteranordnung

Das Ganze funktioniert freilich auch vertikal: `Strg + X 3`, und sooft Sie mögen:

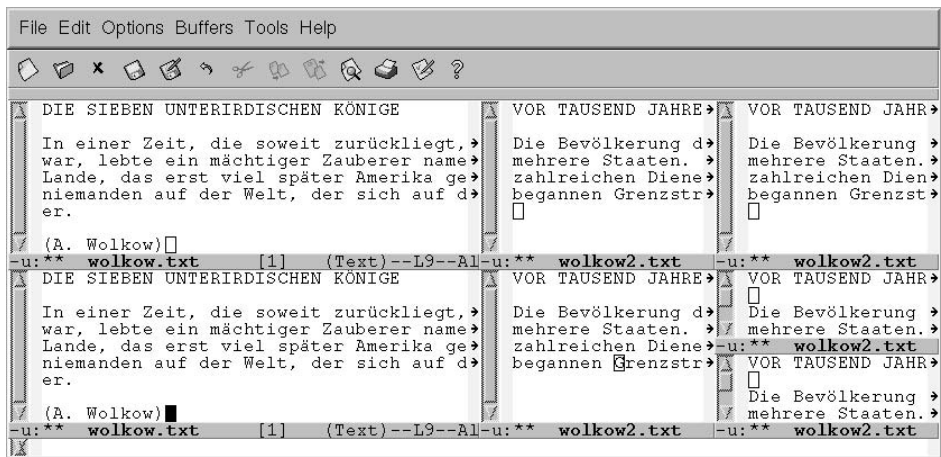


Abbildung 3.5: Horizontale und vertikale Fensteranordnung

Sie werden den Fensterreigen wieder los mit **Strg** + **X** **1**, für »ein Fenster«. Bleiben noch die Rahmen, also *frames*. Eigentlich nur eine Frage des Geschmacks: Arbeiten Sie an mehreren Dateien lieber mit oder ohne den gesamten Oberflächenkomfort? Falls Sie also auf die Arbeitsleisten oben nicht verzichten wollen, teilen Sie den Bildschirm nicht in verschiedene Fenster, sondern laden die nächsten Dateien einfach in Extrarahmen, und zwar mittels **Strg** + **X** **5** **2**:



Abbildung 3.6: Zwei Rahmen

Keine große Geschichte: Mit **Strg** + **X** **5** **0** wechseln Sie zwischen den einzelnen Frames, **Strg** + **X** **5** **0** löscht den aktiven Rahmen und mit **Strg** + **X** **5** **2** laden Sie einen aktuellen Buffer in einen neuen Frame – das ist alles. Hier noch einmal die gebräuchlichsten Befehle:

Tastenkombination	Kommando	Aktion
Strg + 2	split-window-vertically	teilt den Bildschirm in Oben und Unten.
Strg + X 3	split-window-horizontally	teilt den Bildschirm in Links und Rechts.
Strg + X 1	delete-other-windows	löscht andere Fenster.
Strg + X 5 2	make-frame	öffnet einen neuen Rahmen.
Strg + X 5 0	other-frame	wechselt in einen anderen Rahmen.
Strg + X 5 0	delete-frame	löscht einen Rahmen.

Tabelle 3.3: Befehle für Windows und Frames

3.1.3 Eigene Anpassungen

Nehmen wir an, jemand beginnt 1988 damit, einen Roman zu schreiben, und überlegt sich, dieses Unterfangen mit dem Emacs respektive mit $\text{\LaTeX}$ zu wagen. Ganz genau wie Sie öffnet er also den Editor und beginnt mit der Arbeit:

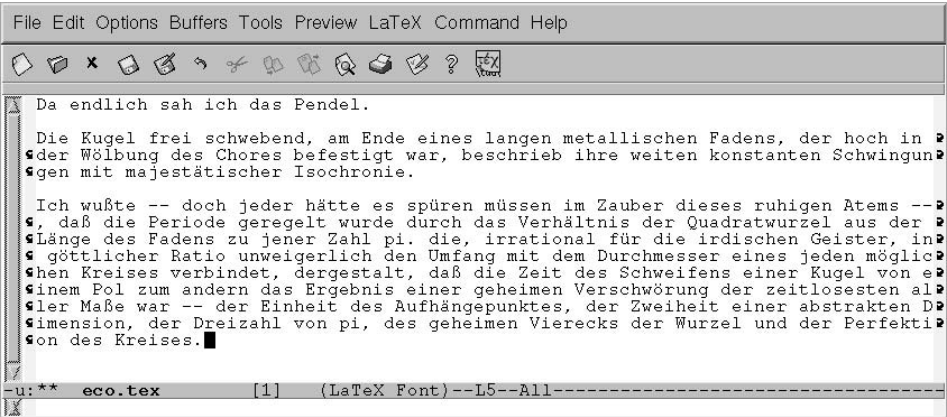


Abbildung 3.7: Textbuffer ohne automatischen Zeilenumbruch

An dieser Stelle unterbricht er und schaut auf den Bildschirm. Die Zeilen werden nicht umbrochen! So ist dem Text freilich kaum zu folgen. Die Anleitung hilft: **Esc** + **Q** ist die Lösung. Nur wäre es lästig, jeden Absatz von Hand umbrechen zu müssen. Alles kein Problem, denn schließlich kann ein solches Kommando auch für die gesamte

Sitzung bereitgestellt werden, in diesem Fall mit `[Esc] + [X]` `auto-fill-mode`. Doch ist es durchaus möglich, dass der Autor beim Editieren des Textes mehrere Extrawünsche hat. Es wäre ein mühseliges Unterfangen, wenn er solche Befehle an zahllosen Stellen im Text eingeben müsste.

Nun ist es so weit, die ersten Anpassungen auf die eigene Arbeitsweise sind fällig. Verantwortlich für Aussehen, Verhalten und Steuerung des Emacs ist die `.emacs`, eine Datei im *home*-Verzeichnis des Benutzers. Sie ist die nächste Futterkrippe für das Gnu. Der Emacs sucht also zunächst hier nach Anweisungen. Wird er nicht fündig, dann grast er die Systemdateien durch, bis er die globale Steuerungsdatei entdeckt, und richtet sich nach dieser. Wenn Sie die Erziehung Ihres *wildebeest* also nicht dem System überlassen wollen, legen Sie sich gegebenenfalls eine entsprechende Datei in Ihrem *home*-Verzeichnis an.

Sie haben mit dieser Datei eine Schaltzentrale zur Steuerung Ihres Editors geschaffen. Sehr gut, denn von nun an können Sie so ziemlich alles ändern, löschen und hinzufügen, was Sie am Emacs für änderungs-, löschungs- und hinzufügenswürdig halten. Was Sie nicht bereits im Netz finden, das können Sie sich kurzerhand selbst programmieren. Emacs ist in *LISP* geschrieben, und wenn Sie entsprechende Kenntnisse haben ... doch zunächst wollen wir sehr einfach beginnen: Unser Autor ist also noch immer mit dem Zeilenumbruchsproblem konfrontiert, das er für immer vom Tisch haben möchte. Und wenn Sie nun denken, das wäre für Ihre eigene Bearbeitung auch keine schlechte Idee, dann machen Sie doch gleich mit.

1. Öffnen Sie eine Datei `.emacs` in Ihrem Home-Verzeichnis:

```
emacs .emacs
```

2. Schreiben Sie dort hinein:

```
; Zeilen automatisch umbrechen
(add-hook text-mode-hook turn-on-auto-fill)
```

3. Speichern Sie die Datei:

`[Strg] + [X] + [S]`

Die Zeile nach dem Semikolon ist ein sogenannter Kommentar, das heißt, es handelt sich um eine Anmerkung, die Sie lesen können, nicht aber das Programm. Kommentare dienen also zur Erläuterung von Quelltexten – schließlich wird das nicht Ihr letzter Eintrag gewesen sein und wenn Sie sich später noch auskennen wollen, dann kommentieren Sie Ihre Änderungen besser. Wenn Sie nun den Emacs starten und eben ein paar Zeilen schreiben, werden Sie feststellen, dass er nun die Zeilen automatisch umbricht.

Außerdem weiß unser Autor, dass es auf Dauer wesentlich augenschonender und damit der Konzentration förderlicher ist, weiße Schrift auf schwarzem Hintergrund zu bearbeiten statt umgekehrt. Die `.emacs` ist ja noch offen, also:

4. Fügen Sie Ihrer *.emacs* noch hinzu:

```
; Weiss-Schwarz anstatt Schwarz-Weiss
(set-background-color "black")
(set-foreground-color "white")
```

Außerdem nerven den Autor die Werkzeugleiste und der Scrollbalken:

5. Fügen Sie Ihrer *.emacs* noch hinzu:

```
; Werkzeugleiste und Scrollbalken verschwinden lassen
(tool-bar-mode -1)
(toggle-scroll-bar -1)
```

6. Speichern und beenden Sie die Datei:

```
(Strg) + (X) + (S) (Strg) + (X) + (C)
```

Unser Autor ist zudem Italiener, er verpasst also nur ungern seine Siesta. Daher ist es ihm wichtig, dass Emacs ihm auch gleich noch die Uhrzeit ausgibt. Falls Sie auch Italiener sind oder aber gern Siesta feiern, kommentieren und ergänzen Sie Ihre *.emacs* entsprechend. Der Eintrag lautet:

```
(display-time)
(setq display-time-24hr-format t)
```

Das soll es für den Anfang gewesen sein. Zumindest erwähnen möchte ich, dass die meisten Einstellungen auch über das Menü OPTIONS/CUSTOMIZE EMACS erreichbar sind. Doch das ist eine arg verschachtelte Navigation und oftmals ist es wesentlich einfacher die *.emacs* zu editieren.⁴ Der Bildschirm unseres Autor sieht mittlerweile so aus:⁵



Abbildung 3.8: Textbuffer nach den o. g. Einstellungen

- 4 Im Internet wimmelt es übrigens nur so von nützlichen Einträgen für die *.emacs* und außerdem gibt es jede Menge Informationen über den Emacs, zum Beispiel <http://de.wikipedia.org/wiki/Emacs> oder www.gnu.org/software/emacs.
- 5 Man beachte: Dies ist bereits eine TeX-Datei mit der Endung *.tex*.

So weit, so gut. Müßig zu erwähnen, dass dies hier nicht einmal Oberflächengekratze in Sachen Emacs-Einstellungen war. Ich tue es trotzdem ...

3.1.4 Editieren von Text

Klar, der Emacs ist ein Editor, ergo: Es soll schlussendlich auch noch ein wenig editiert werden – schließlich werden Sie genau das tun, wenn Sie $\text{\TeX}$ en. Nun höre ich schon die meisten maulen: Häää, ich weiß doch, wie ich den Cursor vor- und zurückbewege, können wir uns doch schenken ... Doch wissen Sie auch, wie Sie den Cursor wortweise bewegen? Satzweise, absatzweise, seitenweise? Wie Sie Zeichen, Wörter oder Zeilen vertauschen, ohne erst herumzucopypasten? Wie Sie ohne großes -Gehämmere von Groß- auf Kleinschreibung wechseln oder versehentlich klein Geschriebenes schnell ausbessern? Wie Sie effektiv löschen und Texte markieren, was ein *kill-ring* ist? Ja? Bravo, dann blättern Sie beruhigt vor ...

Ahhh, doch noch jemand, der weiterliest. Da wir ja nun praktisch unter uns sind, werde ich Sie eben in die Welt der Texteditierung einführen. Eine Welt, die sich zu entdecken lohnt, denn was auch immer Sie mit Texten anstellen: Die meiste Zeit editieren Sie! Wie schön also, wenn das flott und möglichst effektiv von der Hand geht. Ich möchte Ihnen ein Beispiel geben. Nehmen wir an, Sie vertippen sich beim Wort »Strategie« dergestalt, dass so etwas wie »startegie« dabei herauskommt, und bemerken dies schier in dem Augenblick, in dem Sie es geschrieben haben – der Cursor liegt also genau hinter dem letzten Buchstaben. Es ist also der kleine a-r-Dreher zu beseitigen und zudem soll das Wort freilich großgeschrieben werden. Was nun folgt, ist oft so etwas wie das: Es wird sich über die Pfeiltasten zu der entsprechenden Stelle hingehangelt, sie wird ausgebessert und dann geht es – wie Tarzan – wieder zurück:

           + S     A     

Na bravo, dann verlassen Sie sich entweder gleich auf Ihre Rechtschreibprüfung oder vertippen sich niemals, weil Sie andernfalls kaum noch Zeit finden werden, etwas anderes zu tun. Das Ganze ginge auch so: den Cursor auf das *r* und dann  + , um die Buchstaben zu vertauschen. Ein anschließendes  +  schubst den Cursor zum Wortbeginn, an dem mit  +  der erste Buchstabe großgeschrieben und der Cursor gleich wieder an das Wortende befördert wird. Zugegeben, das mag sich auch umständlich anhören, doch probieren Sie es aus – Sie sind wesentlich schneller! Und auch wenn es im Einzelfall nicht darauf ankommt, die Summe macht's. Werden Sie also entweder tippfehlerkonsistent oder editiersicher, doch glauben Sie mir: Die zweite Alternative ist die einfachere.

Navigieren im Text

Zunächst tatsächlich erst einmal die Cursorbewegungen für links, rechts, oben und unten. Zwar gibt es auch hier Shortcuts, doch kenne ich niemanden, der dazu nicht die Pfeiltasten benutzt.

Tastenkombination	Kommando	Aktion
Strg + F	forward-char	bewegt den Cursor ein Zeichen nach rechts.
Strg + B	backward-char	bewegt den Cursor ein Zeichen nach links.
Strg + P	previous-char	bewegt den Cursor eine Zeile nach oben.
Strg + N	next-line	bewegt den Cursor eine Zeile nach unten.

Tabelle 3.4: Cursorbewegungen

Viel öfter als zeichenweise navigiert man horizontal wort- oder zeilenweise. Doch auch hier sind die »üblichen Verdächtigen« naheliegender als die Emacs-Kommandos. So bringen Sie die Tasten **Pos1** und **Ende** zum Zeilenanfang und den Zeilenschluss⁶; die Pfeiltasten (rechts/links), diesmal in Verbindung mit der **Strg**-Taste wortweise vorwärts oder rückwärts. Gleichwohl, hier die Emacs-Kommandos:

Tastenkombination	Kommando	Aktion
Esc + F	forward-word	bewegt den Cursor ein Wort nach rechts.
Esc + B	backward-word	bewegt den Cursor ein Wort nach links.
Strg + A	beginning-of-line	bewegt den Cursor zum Zeilenanfang.
Strg + E	end-of-line	bewegt den Cursor zum Zeilenende.

Tabelle 3.5: Cursorbewegungen II

Es wird ersichtlich, worauf die Sache hinausläuft: *f* steht für *forward* und *b* steht für *back*, *a* für »Anfang« und *e* für *end* ... auch bei den größeren Sprüngen. Gut, so weit zu den grundsätzlichen Cursorbewegungen. Was ist noch häufig und damit wichtig? Natürlich, das ...

6 Um doch noch ein Argument für die Emacs-Shortcuts anzubringen: Wenn Sie über **Pos1** und **Ende** in den Zeilen navigieren, haben Sie im Grunde – wenn auch in abgeschwächter Form – das gleiche Problem wie mit der Maus: Ihre Hand muss die eigentliche Tastatur verlassen und das lenkt auf Dauer ab, kostet Zeit und Konzentration. Bleiben Sie hingegen mit **Strg**+**A** oder **Strg**+**E** auf der Tastatur, arbeiten Sie wesentlich leichter. Doch müssen diese Tastenkombinationen natürlich erst einmal »trainiert« werden.

Tastenkombination	Kommando	Aktion
<code>Esc</code> + <code>E</code>	<code>forward-sentence</code>	bewegt den Cursor einen Satz nach vorn.
<code>Esc</code> + <code>A</code>	<code>backward-sentence</code>	bewegt den Cursor einen Satz nach hinten.
<code>Esc</code> + <code>I</code>	<code>forward-paragraph</code>	bewegt den Cursor einen Absatz tiefer.
<code>Esc</code> + <code>J</code>	<code>backward-paragraph</code>	bewegt den Cursor einen Absatz höher.
<code>Esc</code> + <code>V</code>	<code>scroll-down</code>	bewegt den Cursor eine Seite tiefer.
<code>Strg</code> + <code>V</code>	<code>scroll-up</code>	bewegt den Cursor eine Seite höher.
<code>Esc</code> + <code>></code>	<code>end-of-buffer</code>	bewegt den Cursor an das Ende des Buffers.
<code>Esc</code> + <code><</code>	<code>beginning-of-buffer</code>	bewegt den Cursor an den Anfang des Buffers.

Tabelle 3.6: Cursorbewegungen III

Löschen von Text

Sehr oft wird Text noch während der Bearbeitung geändert. Das Löschen der zu korrigierenden Zeichen, Wörter oder Textpassagen ist ebenfalls Kerngeschäft des Editierens, und auch dazu hat der Emacs einiges zu bieten.

Der einfachste Weg, ein Zeichen zu löschen, ist sicher auch im Emacs zunächst einmal die `⇐`-Taste, daran will ich auch gar nicht rütteln. Wenn der Cursor sowieso in der Nähe der entsprechenden Zeichen ist, belassen Sie es bei dieser Gewohnheit. Doch in dem Moment, in dem Sie beispielsweise ein ganzes Wort löschen, sind Sie mit der Ergänzung `Esc` + `⇐` respektive dem Emacs-Befehl `Esc` + `Enlf` bereits wieder wesentlich schneller. Sagen wir, Sie möchten die drei Worte vor dem Cursor löschen, dann platzieren Sie wahrscheinlich den Cursor hinter das letzte zu löschende Wort und gehen daraufhin wie wild mit der `⇐`-Taste zu Werke. Viel besser ist da das `Esc` + `⇐`, und clever verhalten Sie sich, wenn Sie den Cursor erst gar nicht an das Ende der zu löschenden Wörter bewegen, sondern diesen einfach vor dem entsprechenden Wort stehen lassen und mit `Esc` + `D` zur Tat schreiten – probieren Sie es aus!

Tastenkombination	Kommando	Aktion
<code>Enlf</code>	<code>delete-backward-char</code>	löscht das Zeichen direkt rechts vom Cursor.
<code>Esc</code> + <code>D</code>	<code>kill-word</code>	löscht das Wort rechts vom Cursor.
<code>Esc</code> + <code>Enlf</code>	<code>backward-kill-word</code>	löscht das Wort links hinter dem Cursor.
<code>Strg</code> + <code>K</code>	<code>kill-line</code>	löscht die Zeile rechts vom Cursor.
<code>Esc</code> + <code>K</code>	<code>kill-sentence</code>	löscht den Satz hinter dem Cursor ⁷ .
<code>Strg</code> + <code>W</code>	<code>kill-region</code>	schneidet einen markierten Textausschnitt aus.

Tabelle 3.7: Löschbefehle

Bezüglich der letzten Löschanweisung sollten Sie freilich erst einmal wissen, wie Text markiert wird. Unter X geschieht dies wohl trotz aller Tastatureffizienz eben schnell mit der Maus.

⁷ Ist jedoch mit Vorsicht zu genießen: löscht mitunter den gesamten Absatz.

Weitere wichtige Editierbefehle

Weitere wichtige Editierbefehle sind natürlich Markieren, Kopieren, Ausschneiden. Wie gesagt, das Markieren unter X funktioniert mit der Maus wie in jedem anderen Programm. Nur arbeiten nicht alle ständig unter X und zudem gibt es sicher Nutzer, die bei der Textarbeit nichts mehr mit ihrem Nager am Hut haben (wollen).

- 1. Platzieren Sie den Cursor direkt vor den zu markierenden Bereich.
 - 2. Drücken Sie nun **Strg** + **[]** (im Minibuffer erscheint *Mark set*).
 - 3. Bewegen Sie den Cursor zum Ende des zu markierenden Bereichs.
- Die Stelle, an der Sie den Cursor »abstellen« ist das Ende des Markierungsbereichs. Alles, was Sie also anschließend tun, bezieht sich auf den Bereich zwischen **Strg** + **[]** und der Stelle, an der sich gerade der Cursor befindet. Sie können den markierten Bereich ausschneiden oder kopieren und damit verfahren, wie Sie möchten.

Tastenkombination	Kommando	Aktion
Strg + []	set-mark-command	markiert den Beginn eines Bereichs.
Esc + H	mark-paragraph	markiert einen Absatz.
Strg + X + P	mark-page	markiert eine Seite.
Strg + X H	mark-whole-buffer	markiert alles, also den gesamten Buffer.
Strg + W	kill-region	schneidet einen markierten Bereich aus.
Esc + W	kill-ring-save	kopiert einen markierten Bereich.
Strg + Y	yank	fügt zuvor markierten/gelöschten Text wieder ein.
Esc + Y H	yank-pop	fügt nach <i>yank</i> markierten/gelöschten Text wieder ein.

Tabelle 3.8: Copy&Paste

Falls Sie nun einen etwas zu exzessiven Gebrauch von den oben aufgelisteten Löschoptionen gemacht haben, keine Sorge, anders als bei Textverarbeitungen ist beim Emacs alles noch im Hinterkopf. Das heißt, Sie können nicht nur eine Anweisung zurücknehmen, unmittelbar Gelöschtes wieder einfügen, sondern eben auch davor gelöschte Textbereiche wieder herbeizaubern. Möglich wird das durch ein recht nützliches Konzept namens *kill-ring*. Das hört sich brutal an, ist aber ganz harmlos: Alles, was Sie löschen, landet nämlich zunächst einmal darin und kann infolgedessen auch wieder aufgerufen werden. Einzige Bedingung: Sie müssen zunächst das normale Einfügekommmando **Strg** + **Y** geben. Danach können Sie mit **Esc** + **Y** fröhlich schauen, was der *kill-ring* sonst noch so zu bieten hat. Schrittweise wird Emacs nun alles, was Sie vorher so gelöscht haben, wieder einfügen – sehr nützlich!

Ebenfalls unter der Rubrik »sehr nützlich« einzuordnen sind Korrekturbefehle für die kleinen Verhaspelungen. Werden nämlich Buchstaben oder Wörter in der falschen

Reihenfolge geschrieben, ist oftmals heftig-hektisches, aber eben auch zeitschindendes Gehacke die Folge.

Tastenkombination	Kommando	Aktion
Strg + T	<code>transpose-chars</code>	vertauscht zwei Zeichen.
Esc + T	<code>transpose-words</code>	vertauscht zwei Wörter.
Strg + X + T	<code>transpose-lines</code>	vertauscht zwei Zeilen.
	<code>transpose-sentence</code>	vertauscht zwei Sätze.
	<code>transpose-paragraph</code>	vertauscht zwei Absätze.

Tabelle 3.9: Tauschbefehle

Tastenkombination	Kommando	Aktion
Esc + C	<code>capitalize-word</code>	schreibt den Anfangsbuchstaben eines Wortes groß.
Esc + U	<code>upcase-word</code>	schreibt das ganze Wort in Großbuchstaben.
Esc + L	<code>downcase-word</code>	schreibt das ganze Wort in Kleinbuchstaben.

Tabelle 3.10: Groß- und Kleinschreibung

Erwähnenswert ist noch das: Für den Fall, dass – sagen wir ein Stromausfall – das Programm beendet, keine Sorge! Emacs rettet mit seinem letzten Atemzug noch heldenhaft Ihre Datei und legt sie unter dem Namen *datei.tex~* ab.

Tun Sie sich selbst einen Gefallen und vollziehen Sie die Editieranweisungen alle nochmals an unserer Testdatei *eco.tex* nach (speichern Sie Ihr Werk im Anschluss an Ihre Übungen, wir benötigen die Datei später noch). Bei der Arbeit mit L^AT_EX sind Sie – wie bei jeder anderen Textarbeit auch – nahezu ausschließlich im Editor tätig, also mit Editierarbeiten beschäftigt. Sie sind das A und O der Textarbeit und je mehr Sie dazu den Editierkomfort des Emacs ausnutzen, desto effektiver werden Sie schließlich arbeiten. Hier lauert ein echter Profi und darum: Werden Sie Editierprofi, denn der Emacs ist es schon – nur das nützt Ihnen per se erst einmal wenig!

3.2 Nützliche Modi zur Textbearbeitung

Der Emacs kennt Modi en masse: Für Programmiersprachen, statistische Analysen, zur Kommunikation – Arbeitsoberflächen für alles Mögliche. Uns interessieren, neben aucteX, hier ein paar nützliche Helferlein.

3.2.1 flyspell-mode

Die von Ihnen installierte Rechtschreibprüfung *Aspell* oder auch *Ispell* tritt freilich erst in Aktion, wenn sie dazu veranlasst wird. Unter TOOLS—SPELL CHECKING muss dazu zunächst einmal ein Wörterbuch gewählt werden. Anschließend kann dann der gesamte Buffer, ein bestimmter Bereich oder auch nur ein einzelnes Wort (`[Esc] + [S]`) geprüft werden. Findet die Rechtschreibprüfung ein verdächtiges Wort, so stoppt der Durchlauf. Sie haben dann die Möglichkeit, mit `[ ]` die Suche ohne Intervention fortzusetzen, mit `[R]` (für *replace*) das Wort im Minibuffer zu korrigieren, mit `[I]` (für *insert*) ins Wörterbuch aufzunehmen oder aber mittels der Zahlenauswahl die oben aufgeführten Vorschläge zu akzeptieren. Das ist für die meisten umfangreicheren Texte das übliche Vorgehen.

Doch was man dabei bemerkt, sind oftmals vermeidbare krasse Tippfehler, die, wären sie vorher aufgefallen, besser gleich beim Verfassen des Textes korrigiert worden wären. Und genau hier kommt *flyspell* ins Spiel. Zuschaltbar über `[Esc] + [X]` *flyspell-mode*, werden Ihre Eingaben bereits beim Tippen überprüft. Schleicht sich nun ein **Tippfehler** ein, wird dieser leuchtend rot markiert, unterstrichen und fett ausgezeichnet⁸ – den können Sie gar nicht übersehen! Bessern Sie das betreffende Wort aus, verschwinden die Auszeichnungen wieder. Wollen Sie das Wort in das Wörterbuch übernehmen, ein Klick mit der mittleren Maustaste (oder ein gleichzeitiger auf die linke und rechte Taste) ermöglicht auch das. Insgesamt eine feine Sache, deren Benutzung sehr zu empfehlen ist.

3.2.2 outline-minor-mode

Erinnern Sie sich an das zu Beginn Gesagte: Editorenarbeit folgt dem Prinzip *what you see is what you mean*. Das heißt aber auch, Ihr Text präsentiert sich bei dessen Bearbeitung nicht hübsch nach Seiten sortiert, sondern zunächst als unstrukturierter

⁸ Zeichnen Sie so nur ja nie im eigentlichen Text aus, denn das ist ein typographischer Offenbarungseid allererster Güte! Doch dazu im zweiten Teil des Buches mehr.

Fließtext. Für längere Texte kann das recht unübersichtlich werden, da bei eventuellen Korrekturen in bereits bearbeiteten Passagen eine mühevollte Suche nötig ist.

Warum also nicht alles, was gerade nicht benötigt wird, »einklappen«, wie in Abbildung 3.9 zu sehen. Die Vorteile: Der Text stört nicht mehr, man hat eine Art Inhaltsverzeichnis vor Augen, die Navigation durch den Text wird wesentlich vereinfacht und bei Textumstellungen geht alles schneller von der Hand. Beachten Sie zudem die neuen Menüeinträge **SHOW** und **HIDE** zum Ein- und Ausklappen von Textteilen sowie **HEADINGS** zum Navigieren zwischen den Überschriften. Also, mit **[Esc] + [X]** `outline-minor-mode`⁹ sind Sie im Spiel.

```
File Edit Options Buffers Tools Preview LaTeX Command Headings Show Hide Help

\subsection*{Grundlegende Kommandos}\index{Editoren!Emacs!Grundlegende...}
\subsection*{Organisation des Bildschirms}\index{Editoren!Emacs!Bildschirm}...
\subsection*{Eigene Anpassungen}\index{Editoren!Emacs!Eigene Anpassungen}...
\subsection*{Editieren von Text}\index{Editoren!Emacs!Editierbefehle}...
\subsubsection*{Bewegungen im Text}...
\subsubsection*{Löschen von Text}...
\subsubsection*{Sonstige Editierbefehle}...
\subsection*{Nützliche Modi zur Textbearbeitung}...
\subsubsection*{flyspell-mode}...
\subsubsection*{outline-minor-mode}

Erinnern Sie sich an das zu Beginn Gesagte: Editorenarbeit ist
\emph{what you see is what you mean.} Das heißt aber auch, Ihr Text
legt bei dessen Bearbeitung nicht hübsch nach Seiten sortiert vor,
sondern vielmehr als Lauftext. Bei längeren Texten kann das recht
unübersichtlich sein, da, bei eventuellen Korrekturen in bereits
bearbeiteten Passagen eine mühevollte Suche und Scrollerei beginnt.

\section{AUCiTeX in Action}

--:** kapitel_03.tex [1]11:31 0.34 (LaTeX Outl Font Fly Fill)--L1256--46%----
```

Abbildung 3.9: »Eingeklappter« Text durch `outline-minor-mode`

Unten in der Statusleiste werden Sie über die gerade verwendeten Modi informiert. Da hier bereits einige Modi laufen, wurde der Minormodus, also eine Art Untermodus, gewählt. Es gibt jedoch auch den `outline-modus` als Hauptmodus. Im `LaTeX-Modus` orientiert sich der Outline-Modus an den Satzanweisungen. Es ist jedoch auch möglich, unformatierten Text im Outline-Modus zu bearbeiten. Die einzelnen Ebenen werden dann durch Sternchen (*) definiert – je mehr Sternchen, desto niedriger die Ebene.

⁹ Tippen Sie einfach nur `out` und drücken Sie dann die **[Esc]**-Taste.

3.2.3 font-lock-mode

Sie haben nun bereits einige Modi des Emacs kennengelernt: nämlich den L^AT_EX-Modus – den wir gleich etwas genauer betrachten wollen; den Auto-Fill-Modus zum automatischen Zeilenumbruch; den Fly-Spell-Modus zur laufenden Rechtschreibprüfung; den Outline-Modus für die bessere Übersicht. Doch einer fehlt noch: der sogenannte Font-Lock-Modus.

Wie Sie bereits wissen, besteht ein Quelltext nicht nur aus Ihrem Fließtext, sondern eben auch aus Satzanweisungen. Damit diese bei der Arbeit mit dem eigentlichen Text möglichst wenig stören, können und sollten Sie Emacs anweisen, solche Auszeichnungen, Befehle oder Umgebungen farblich zu kennzeichnen. Glauben Sie mir, das erleichtert die Orientierung in umfangreichen Texten ganz erheblich.

Sie können diesen Modus durch `[Esc] + [X]` `font-lock-mode` für die aktuelle Sitzung zu- oder durch einen Eintrag in die `.emacs` für ein permanentes Font-Lock sorgen. Die Einträge lauten:

```
;; Syntaxhighlighting
(global-font-lock-mode t)
```

Und wenn Ihnen die verwendeten Farben und Auszeichnungen nicht passen sollten, Emacs ist nicht nachtragend: Ändern Sie diese gemäß Ihren Vorstellungen. `[Esc] + [X]` `list-faces-display` befördert Sie geradewegs in die entsprechende Werkstatt.

3.3 AUC_TE_X

Also gut, was ist nun dieses bereits öfters erwähnte ominöse AUC_TE_X? Nichts anderes als eine Erweiterung des Emacs, die Ihnen die Arbeit im Umgang mit L^AT_EX-Dokumenten deutlich erleichtert – im Grunde eine Art Tipphilfe oder besser ein Tippersatz. Die Fülle der möglichen Satzanweisungen ist derart vielfältig, dass es völlig unmöglich ist, sie alle in Form von Icons klickfreundlich zur Verfügung zu stellen. Darum sind Sie bei der Verarbeitung Ihrer Quelltexte gezwungen, die Syntax sämtlicher Satzanweisungen direkt einzugeben. Möchten Sie beispielsweise ein Wort kursiv auszeichnen, so heißt der entsprechende Befehl `\textit{Wort}` – ein recht aufwendiges Unterfangen. Und genau hier kommt AUC_TE_X ins Spiel. Dieses Programm stellt ein entsprechendes Menü, nämlich L^AT_EX—INSERT FONT—ITALIC, und eine praktische Tastenkombination, in diesem Fall `[Strg] + [F] + [I]`, zur Verfügung, um diese Anweisung ratzfatz ins Dokument einzufügen. Wenn Sie nun jede einzelne Klammer – und die meisten Anweisungen haben mindestens zwei davon – ausschreiben müssten, denn wäre dies eine mühselige Angelegenheit. So aber lassen sich die gebräuchlichsten An-

weisungen gleich per Tastenkombination in den Quelltext einfügen. Daher ist es besser, Sie überlegen gar nicht erst, ob Sie AUCTeX einsetzen oder nicht, sondern akzeptieren es von Anfang an als unumstößliche Notwendigkeit. Andernfalls landen Sie eines schönen Tages noch bei irgendeiner windigen $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Oberfläche oder verabschieden sich aus falschem Grund von der schönen Idee gesetzter Texte.

AUCTeX vereinfacht das Eingeben von Befehlen, Umgebungen, Überschriften ... bis hin zum Aufruf des eigentlichen Rechendurchlaufs, mit dem die Satzanweisungen verarbeitet werden – installieren Sie es also besser gleich mit. Falls Sie AUCTeX nicht gleich von $\text{T}_{\text{E}}\text{XLive}$ übernehmen wollen, weil eventuell bereits eine neuere Version zur Verfügung steht, hier werden Sie fündig: www.gnu.org/software/auctex/download.html, und auch schlau, denn die Seite hält auch eine umfassende Dokumentation bereit, die auch mit Installationshinweisen aufwartet. Und weil ich Ihnen viel erzählen kann, Sie jedoch durch dieses Theorie-Geplauder nicht eben schlauer werden, sei nun die Zeit gekommen, den ersten Text zu $\text{T}_{\text{E}}\text{X}$ en. Schließlich haben Sie bereits ein solches Dokument: Ihre *eco.tex*, die Datei, die für die Editierübungen herhalten musste. Öffnen Sie doch einmal das gute Stück im Emacs. Das Ganze sollte noch immer so aussehen wie in Abbildung 3.9, also noch ohne jedwede Satzanweisung. Das wird sich nun ändern.

3.4 Der $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Emacs- AUCTeX -Praxistest

Schwierige Situation: Mit dem Emacs so weit fertig, in AUCTeX eingeführt, jedoch mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ noch gar nicht so recht begonnen – und dennoch juckt es nach über 50 Seiten langsam in den Fingern, das Ganze endlich einmal ausprobieren zu wollen. Recht so! Obwohl wir uns in einer Zwischenwelt befinden, wollen wir sofort loslegen, ohne noch weiterhin lange um den sprichwörtlichen heißen Brei herumzureden.

Hier, sowie auch im nächsten Kapitel, wird dem engagierten Einsteiger ein schneller Weg in das Thema gegeben. Das Buch vor Ihrer Nase ist nicht das klassische Anfängerbuch und insofern soll nun auch nicht im Urschleim begonnen werden. Vielmehr bin ich der Meinung, dass jedes weitere Minischrittverharre eine mögliche Motivation nur ausbremst.

Gleichwohl war es der von Anfang an gehegte Wunsch des Autors, auch (und gerade) den $\text{T}_{\text{E}}\text{X}$ -Einsteiger mit auf diese Reise zu nehmen. Daher folgt die Vermittlung der Grundlagen nunmehr im Schnellverfahren, angefangen bei der gleich im Anschluss folgenden Weiterentwicklung der *Eco*-Datei. In ihr wollen wir bereits die populärsten Elemente eines $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Dokuments integrieren, auf dass Sie mit einem scharfen Blick in den Quelltext bereits loslegen können. Danach folgt das Schnelleinstieg-Kapitel.

Das stellt freilich mehr vor, als es erläutert. Doch keine Sorge, vieles von dem darin Behandelten wird in den dann folgenden Kapiteln noch näher betrachtet.

Sinn des Ganzen ist also nicht, mit dem motivierten Einsteiger in ewiger Unterforderung und mit Tippielschrittchen durch die technischen Grundlagen zur Typographie zu schlürfen, sondern das T_EX-Neuland forsch zu durchschreiten. Natürlich wird das etwas anstrengend. Doch T_EX ist kein böhmisches Dorf und Sie werden sehen: Diese Gangart wird vollkommen ausreichen, um sich in eine entsprechend komfortable Ausgangsposition zu bringen. Na dann – *less talk, more texing* – fangen wir endlich an! Jedes L^AT_EX 2_ε-Dokument beginnt mit dem Aufruf der Dokumentenklasse. Dann folgt der Aufruf aller verwendeten Pakete. Der eigentliche Text erfordert bereits eine Umgebung, also die Definition von Anfang und Ende eines bestimmten Bereichs. Diese Umgebung ist zunächst das Dokument selbst, doch der Reihe nach. Öffnen Sie Ihre bereits vorhin bearbeitete Datei *eco.tex* im Emacs selbst oder von der Konsole aus:

```
emacs eco.tex
```

Emacs öffnet daraufhin den Text, doch das arme Tier: Da noch keinerlei Kommandos für L^AT_EX im Text enthalten sind, hält sich der Emacs an die Dateiendung und nimmt zunächst an, es handle sich um ein T_EX-Dokument. Helfen Sie ihm ein wenig auf die Sprünge, indem Sie AUC_TE_X laden:

```
[Esc] + [X] latex-mode [↵]
```

Gut, nun sollte statt Tex ein L^AT_EX im Menü stehen. Sorgen Sie ganz oben im Dokument für ein wenig Raum. Platzieren Sie nun den Cursor in die erste Zeile und geben Sie die Dokumentenklasse an, also:

```
\documentclass{article}
```

Bereits für das geschweifte Klammernpaar können Sie die Hilfe von AUC_TE_Xs in Anspruch nehmen. Mit [Strg] + [C] [↵] werden die entsprechenden Klammern eingefügt. Doch da sich hier keinerlei Tipparbeit einsparen lässt, ist das eher Geschmackssache. Etwas unterhalb der Dokumentenklasse nun die erste und unverzichtbare Umgebung. Entweder Sie schreiben nun `\begin{document}` respektive `\end{document}` unmittelbar vor sowie nach dem eigentlichen Text oder Sie fügen diese Umgebung wiederum mit AUC_TE_X-Kommandos ein:

```
[Strg] + [C] + [E] document [↵]
```

Nun müssen Sie freilich noch den Text zwischen die Anfangs- und Endmarkierung des Dokuments platzieren. Also mit der Maus markieren, ein [Strg] + [W] (schnipp, weg ist er), Cursor in die Umgebung navigieren und den Text mit [Strg] + [Y] wieder einfügen.

```
File Edit Options Buffers Tools Preview LaTeX Command Help

\documentclass{article}

\begin{document}

DAS FOUCAULTSCHE PENDEL

Da endlich sah ich das Pendel.

Die Kugel frei schwebend, am Ende eines langen metallischen Fadens,
der hoch in der Wölbung des Chores befestigt war, beschrieb ihre
weiten konstanten Schwingungen mit majestätischer Isochronie.

Ich wußte -- doch jeder hätte es spüren müssen im Zauber dieses
ruhigen Atems --, daß die Periode geregelt wurde durch das Verhältnis
der Quadratwurzel aus der Länge des Fadens zu jener Zahl  $\pi$ . die,
irrational für die irdischen Geister, in göttlicher Ratio unweigerlich
den Umfang mit dem Durchmesser eines jeden möglichen Kreises
verbindet, dergestalt, daß die Zeit des Schweifens einer Kugel von
einem Pol zum andern das Ergebnis einer geheimen Verschwörung der
zeitlosesten aller Maße war -- der Einheit des Aufhängepunktes, der
Zweiheit einer abstrakten Dimension, der Dreizahl von  $\pi$ , des geheimen
Vierecks der Wurzel und der Perfektion des Kreises.

(Umberto Eco)

\end{document}

--:-- eco.tex [1]17:32 0.91 (LaTeX Font Fill)--L25--All-----
```

Abbildung 3.10: Eco-Quelltext (Rohversion)

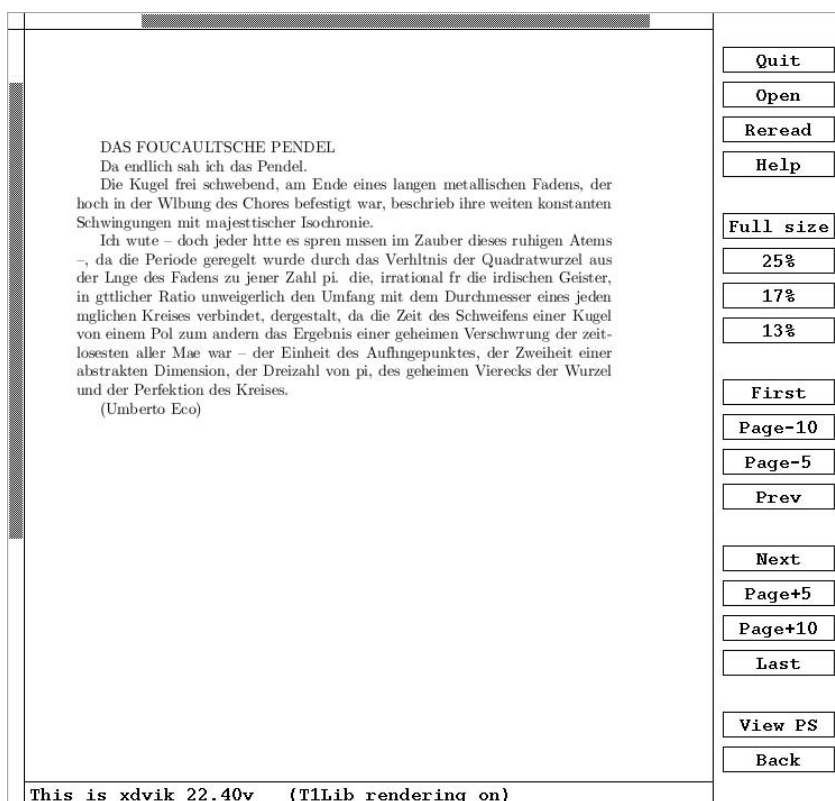
Was soll ich sagen: Ihr erstes $\text{\TeX}$ -Dokument ist reif für den Satz. Dazu speichern Sie das Dokument, öffnen eine Konsole, laden das Verzeichnis Ihrer Datei und übergeben den Befehl `latex eco.tex` an die Konsole oder mit $\text{\LaTeX}$:

`Strg` + `C` + `C` `Y` `↵`

Und schon wird gesetzt. Um sich Ihr Werk anzuschauen, können Sie freilich wiederum die Konsole mit `xdvi eco.dvi` bemühen: Mit $\text{\LaTeX}$ geht es so:

`Strg` + `C` + `C` `↵` `↵`

Wenn Ihnen Emacs nun im Minibuffer mitteilt: *LaTeX: successfully formatted {1} pages*, haben Sie es – Ihre erste gesetzte Seite!

Abbildung 3.11: Das Zwischenformat: *eco.dvi*

Doch Schreck, was ist geschehen? Es fehlen die Umlaute und das β geht ab. Außerdem sieht die Sache noch gar nicht sooo schön aus. Nun ja, zunächst ist T_EX Amerikaner und deshalb sind ihm Umlaute so fremd wie den meisten von uns Erdnussbutter zum Frühstück. Sie müssen dem Programm also erst einmal mitteilen, dass es sich um einen deutschen Text handelt. Auch der Satz wirkt noch eher grob – ist er auch.

Die meisten werden es bemerkt haben: Bei unserem Autor handelt es sich um keinen Geringeren als Umberto Eco, der gerade *Das Foucaultsche Pendel* beginnt. Lassen Sie uns die Sache etwas vertiefen und schauen, was sich hinter einem solchen Pendel eigentlich verbirgt. Dazu soll ein kleines Dokument verfasst werden. Hier der Quelltext, nachfolgend sein Ergebnis. Wenn Sie beides einmal ausgiebig vergleichen, werden Sie sehr schnell verstehen, wie die Sache läuft! Es ist nämlich bereits einiges enthalten: Zweispaltensatz, eine kleine Zeichnung, Fußnoten, ein Textkasten sowie eine Nummerierung – viel mehr werden Sie am Anfang kaum benötigen.

```
\documentclass[twocolumn]{article}      % sorgt für zweispaltigen Satz
\usepackage[ngerman]{babel}            % für die (neuen) deutschen Sprachregeln
\usepackage[T1]{fontenc}               % Zeichenausgabe
\usepackage[latin1]{inputenc}          % automatische Umlaute etc.
\usepackage{epic,eepic}               % Pakete für die Zeichnung
```

```
\begin{document}                       % Beginn des Texts
```

```
\thispagestyle{empty}                 % unterdrückt die Seitennummerierung
```

```
\section{Was ist nun gemeint, mit dem Foucaultischen Pendel?}
```

Und sie dreht sich-- die Erde. Den Nachweis erbrachte ein Franzose namens Jean B.\,L.\,Foucault, indem er 1851 einen 67 Meter langen Draht an die Decke des Pariser Pantheons band und eine 28\,kg schwere Kugel daran befestigte. Ein Stift an deren Unterseite beschrieb dann einen Kreis im feuchten Sand. Und damit war klar: Ja, sie dreht sich!

% die Skizze:

```
\begin{picture}(0,0)(155,25)
\linethickness{2pt}                  % Linienstärke
  \drawline(115,-200)(115,10)
\linethickness{.5pt}
  \dashline{4}(30,-190)(115,10)      % Strichellinie
  \dashline{4}(115,10)(200,-190)
  \put(115,10){\circle*{5} \scriptsize\textsf{Aufhängepunkt}}
  \put(115,-200){\circle*{5} \scriptsize\textsf{Ruhelage}}
  \put(200,-190){\circle*{5} ~\scriptsize\textsf{-punkte}}
  \put(30,-190){\circle*{15} ~\scriptsize\textsf{Umkehr-}}
  \put(115,-200){\circle{15}}        % Kreis
  \put(200,-190){\circle{15}}
\linethickness{.25pt}
  \qbezier(30,-190)(115,-210)(200,-190)
\end{picture}
\vspace{8cm}                         % schafft vertikalen Abstand
```

Ein Pendel ist jeder Körper, der in einem festen Punkt oder einer festen Achse drehbar aufgehängt ist und unter dem Einfluss der Schwerkraft Schwingungen\footnote{Diese werden infolge des Luftwiderstandes und der Reibung des Fadens am Aufhängepunkt gedämpft.} um eine stabile Gleichgewichtslage auszuführen vermag. Die Anzahl der Vollschrwingungen in einer Sekunde heißt \emph{Schwingungszahl} (Frequenz). Die Zeit, die während einer Vollschrwingung verstreicht heißt \emph{Schwingungsdauer}. Schwingungen dieser Art werden auch Sinusschrwingungen genannt, weil sie der mathematischen Funktion des Sinus entsprechen.

% die Box für das Zitat:

```
\hbox{}\hfill\fbbox{}               % die Textbox
\begin{minipage}{7.5cm}
\noindent{\itshape                  % kein Zeileneinzug, kursiv
```

Ich wußte~-- doch jeder hätte es spüren müssen im Zauber dieses ruhigen Atems~--, daß die Periode geregelt wurde durch das Verhältnis der Quadratwurzel aus der Länge des Fadens zu jener Zahl π , die, irrational für die irdischen Geister, in göttlicher Ratio unweigerlich den Umfang mit dem Durchmesser eines jeden möglichen Kreises verbindet, dergestalt, daß die Zeit des Schweifens einer Kugel von einem Pol zum andern das Ergebnis einer geheimen Verschwörung der zeitlosesten aller Maße war~-- der Einheit des Aufhängepunktes, der Zweiheit einer abstrakten Dimension, der Dreizahl von π , des geheimen Vierecks der Wurzel und der Perfektion des Kreises.\

\hfill~--- UMBERTO ECO}

\end{minipage}
 \hfill\hbox{}\\[\baselineskip]

Die Schwingungsdauer t berechnet sich über die Pendellänge l und g (beträgt ca. 981 cm s^{-2}) wie folgt:
 % die Formel:

$$t = 2\pi \sqrt{\frac{l}{g}}$$

PENDELGESETZE:

% eine Aufzählung:

\begin{enumerate}

\item Schwingungszahl und -dauer sind bei kleinen Ausschlägen von Amplitude und Masse unabhängig.

\item Je länger das Pendel, desto größer die Schwingungsdauer t und desto kleiner die Schwingungszahl.

\item Die Länge des Sekundenpendels ist je nach geografischer Breite des Aufstellortes verschieden.

\item Die Geschwindigkeit des Pendels ist an den Umkehrpunkten gleich null und beim Durchgang durch die Ruhelage am größten.

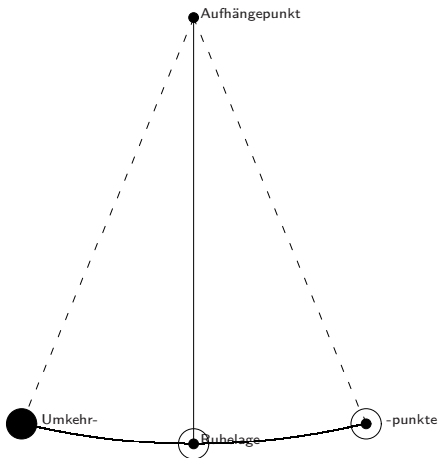
\end{enumerate}

\end{document}

% Ende des Dokuments

Was ist nun gemeint, mit dem Foucaultischen Pendel?

Und sie dreht sich – die Erde. Den Nachweis erbrachte ein Franzose namens Jean B.L. Foucault, indem er 1851 einen 67 Meter langen Draht an die Decke des Pariser Pantheons band und eine 28 kg schwere Kugel daran befestigte. Ein Stift an deren Unterseite beschrieb dann einen Kreis im feuchten Sand. Und damit war klar: Ja, sie dreht sich!



Ein Pendel ist jeder Körper, der in einem festen Punkte oder einer festen Achse drehbar aufgehängt ist und unter dem Einfluss der Schwerkraft Schwingungen¹ um eine stabile Gleichgewichtslage auszuführen vermag. Die Anzahl der Vollschrwingungen in einer Sekunde heißt *Schwingungszahl* (Frequenz). Die Zeit, die während einer Vollschrwingung verstreicht, heißt *Schwingungsdauer*. Schwingungen dieser Art werden auch Sinusschwingungen genannt, weil sie der mathematischen Funktion des Sinus entsprechen.

¹Diese werden infolge des Luftwiderstandes und der Reibung des Fadens am Aufhängepunkt gedämpft.

Ich wußte – doch jeder hätte es spüren müssen im Zauber dieses ruhigen Atems –, daß die Periode geregelt wurde durch das Verhältnis der Quadratwurzel aus der Länge des Fadens zu jener Zahl π , die, irrational für die irdischen Geister, in göttlicher Ratio unweigerlich den Umfang mit dem Durchmesser eines jeden möglichen Kreises verbindet, dergestalt, daß die Zeit des Schweifens einer Kugel von einem Pol zum andern das Ergebnis einer geheimen Verschwörung der zeitlosesten aller Maße war – der Einheit des Aufhängepunktes, der Zweiheit einer abstrakten Dimension, der Dreizahl von π , des geheimen Vierecks der Wurzel und der Perfektion des Kreises.

— UMBERTO ECO

Die Schwingungsdauer t berechnet sich über die Pendellänge l und g (beträgt ca. $981 \text{ cm} \cdot \text{sec}^{-2}$) wie folgt:

$$t = 2\pi \sqrt{\frac{l}{g}}$$

PENDELGESETZE:

1. Schwingungszahl und -dauer sind bei kleinen Ausschlägen von Amplitude und Masse unabhängig.
2. Je länger das Pendel, desto größer die Schwingungsdauer (t) und desto kleiner die Schwingungszahl.
3. Die Länge des Sekundenpendels ist je nach geographischer Breite des Aufstellortes verschieden.
4. Die Geschwindigkeit des Pendels ist an den Umkehrpunkten gleich null und beim Durchgang durch die Ruhelage am größten.

Gut, wir haben also einen Quelltext und (oben) sein Ergebnis. Natürlich interessieren uns nicht wirklich die Gesetzmäßigkeiten des Pendels als vielmehr gebräuchliche Textbestandteile wie Spaltensatz, Überschriften, Fußnoten, Textblöcke oder Formeln – eine kleine Zeichnung soll es auch sein – was man halt so braucht ... und das alles hübsch auf einer Seite.

Mit dem ersten Paket `ngerman` sorgen Sie für neue deutsche Rechtschreibung sowie die entsprechenden Trennmuster. Das Paket `inputenc` gibt Ihnen unter anderem die deutschen Umlaute und unsere meistbenutzte Ligatur: das β .

Bei der Lektüre des Quelltextes fallen zunächst die eingeschobenen Kommentare auf. Diesen ist das Prozentzeichen `%` vorangestellt, welches bewirkt, dass nur Sie, nicht aber L^AT_EX damit etwas anzufangen weiß. Die Folge: Textpassagen, die dem Prozentzeichen folgen, sind »auskommentiert« und demzufolge im gesetzten Dokument nicht vorhanden. Kommentare dienen also ausschließlich der Orientierung bei der Lektüre des Quelltextes – was uns sofort zu der Frage bringt: Wenn ein Prozentzeichen Text verschwinden lässt, was, wenn dieses Sonderzeichen im Text selbst verwendet werden soll? Es gibt einige für T_EX reservierte Sonderzeichen, doch keine Sorge, auch diese können freilich im Text benutzt werden. Im Falle des Prozentzeichens geschieht das mit Hilfe des Backslashes, also `\%`.

Dazu später mehr, wie zu den meisten anderen Dingen auch. Dies sei nur einmal ein kleiner Crash-Text, damit Sie schon einmal ein Gefühl dafür bekommen, wohin die Reise geht. Wir wollen den Quelltext kurz abschnittsweise erläutern.

Der erste Abschnitt, also der Aufruf der Dokumentenklasse, ihrer Optionen sowie der Pakete, ist die sogenannte Präambel. Bereits die Dokumentenklasse lässt sich per AUC_TE_X aufrufen, nämlich mit `[Strg] + [E] document [↵]`.

Nach der Bestätigung fragt das System nach der Art des Dokuments. Die Voreinstellung ist *article*, Sie können also einfach mit `[↵]` bestätigen. Mit der nächsten Frage können Sie noch Optionen angeben (in unserem Fall beispielsweise *twocolumn* für den zweispaltigen Text). Sämtliche Umgebungen, also Textabschnitte, die mit einem *begin* starten und mit einem *end* aufhören, lassen sich also bequem über die Tastenkombination `[Strg] + [C] + [E]` erreichen. Drücken Sie anschließend die `[↵]`-Taste, dann zeigt Ihnen Emacs die zur Verfügung stehenden Alternativen.

Eine Überschrift der Art *section* (erste Ebene) bekommen Sie mit der Tastenkombination `[Strg] + [C] + [S]`. Bereits im ersten Textabschnitt sind einige mikrotypographische Feinheiten berücksichtigt. Der doppelte Bindestrich sorgt für einen typographisch korrekt langen Gedankenstrich und die Zeichenkette `\.` für ein halbes Leerzeichen, zumeist hinter abkürzenden Punkten zu verwenden.

Was folgt, ist die kleine Skizze. Diese besteht aus Geraden, Kurven und Kreisen in einer *picture*-Umgebung. Diese Umgebung wird benötigt, um mit dem `put`-Kommando die Elemente der Skizze zu platzieren. Die Beschriftung erfolgte in einer serifenlosen Schrift. Ein solcher Wechsel lässt sich in AUC_TE_X durch die Befehle `[Strg] + [C] + [F]` für `\textsf{}` oder `[Strg] + [C] + [I]` für `\textit{}` erreichen. Sollen hingegen ganze Textpassagen kursiv erscheinen, wie das Eco-Zitat in der Textbox, so ist dies über den Aufruf `\itshape` möglich.

Damit $\LaTeX$ eine Formel erkennt, sind die darin befindlichen Argumente in eine solche Umgebung: `\[ . . . \]`, zu betten. Eine Unsitte zur Formelkennzeichnung ist `$. . . $` – ein reiner $\TeX$ -Befehl. Doch ganz ehrlich, für ein schnelles π kann ich den Dollarzeichen auch nicht widerstehen (siehe Quelltext).

Fußnoten sind leicht über das Kommando `\footnote{Text}` realisiert. Warum es dafür in $\text{AUC}\TeX$ keinerlei Entsprechung gibt, ist unverständlich. Die Aufzählung zum Schluss ist abermals eine Umgebung, Sie wissen ja: `[Strg]+[C]+[E]` . . . Übrigens können Sie mit `[Esc]+[↵]` schnell einen neuen Aufzählungspunkt (*item*) einfügen.

Natürlich gibt es noch jede Menge andere Kommandos. In diesem Zusammenhang sei auf die *AUC $\TeX$ Reference Card* verwiesen, überall im Internet, wie zum Beispiel unter www.cip.mathematik.uni-stuttgart.de/Hilfe/refcards.html, erhältlich.