

**ETH Institut für Verhaltenswissenschaft
Departement Informatik**

Leitprogramm
Einführung in L^AT_EX
Fach: Informatik
Sekundarstufe II
Bearbeitungsdauer: 10 Lektionen à 50
Minuten

Claudia Brauchli
Betreuer: J. Hromkovic

20.Oktober 2006

Inhaltsverzeichnis

1	Vorwort	4
1.1	Lernziel und Inhalt	4
1.2	Verwendete Programme	5
1.3	Text- und Zeichenerklärungen	5
2	L^AT_EX Einführung	6
3	Anwendung	9
3.1	Neue Datei beginnen	9
3.2	Neuen Ordner erstellen	10
3.3	Datei speichern	11
3.4	Datei öffnen	13
3.5	Datei in pdf umwandeln	17
3.6	.pdf, .tex, .toc, .aux, .log	21
3.7	Zusammenfassung	23
3.8	Kapitelprüfung	23
4	Aufbau	25
4.1	Grundgerüst	25
4.2	Package	28
4.3	Titel, Autor, Datum	30
4.4	Inhaltsverzeichnis und Struktur des Dokuments	32
4.5	Zusammenfassung	36
4.6	Kapitelprüfung	36
5	Datei-Format	41
5.1	Textformatierung	41
5.2	Schriftformatierung	46
5.3	Aufzählungen	48
5.4	Zusammenfassung	51
5.5	Kapitelprüfung	51
6	Bilder	53
6.1	Bildertypen	53
6.2	Bilder einfügen	53
6.3	Bildposition	54
6.4	Zusammenfassung	55
6.5	Kapitelprüfung	55
7	Fussnoten und Seitendarstellung	56

<i>INHALTSVERZEICHNIS</i>	3
7.1 Fussnoten	56
7.2 Wichtige Einzelheiten	56
7.3 Zusammenfassung	56
7.4 Kapitelprüfung	57
8 L^AT_EX-Zusammenfassung	58
9 Prüfung	61
10 Literatur	63
11 Lösungen (<i>nur für Lehrperson</i>)	64

1 Vorwort

Ein Leitprogramm ist ein Selbststudienmaterial in Form eines Heftes oder eines elektronischen Angebotes. Es enthält alle notwendigen Unterrichtsinhalte, Übungen, Arbeitsanleitungen, Tests und andere Hilfsmittel, die die Schüler ¹ brauchen, um ohne Lehrperson lernen zu können. Das Heft leitet durch den Unterricht, deshalb heisst es Leitprogramm.

Dieses Leitprogramm führt den Schüler in das Textverarbeitungssystem $\text{\LaTeX}$ ein. Die Kapitel erklären das zu lernende Thema Schritt für Schritt, sie müssen demnach der Reihe nach gelesen werden. Der Schüler arbeitet das Leitprogramm in seinem eigenen Lerntempo zu Hause oder in der Schule durch. Jedes Kapitel wird mit einer Kapitelprüfung abgeschlossen. Nach erfolgreich bestandener Prüfung händigt die Lehrperson dem Schüler das nächste Kapitel aus. Ein Kapitel darf mehrmals durchgearbeitet werden - bis sich der Lernende für sattelfest fühlt.

Im Literaturverzeichnis werden weitere hilfreiche Bücher und Internetseiten angegeben, welche für zusätzliche Informationen herbeigezogen werden können.

1.1 Lernziel und Inhalt

Am Schluss dieses Leitprogrammes kann der Schüler eine eigene $\text{\LaTeX}$ -Datei erstellen und diese in eine PDF-Datei umwandeln.

$\text{\LaTeX}$ eignet sich hervorragend für die Erstellung von Vorträgen, Berichten, Zusammenfassungen, Semesterarbeiten etc. $\text{\LaTeX}$ ist ein Textverarbeitungsprogramm, das anders funktioniert als das bekannte "Microsoft Word". Viele Elemente wie Schriftart und Schriftgrösse sind bereits definiert und lassen verschiedene $\text{\LaTeX}$ -Arbeiten ähnlich aussehen. Dies verleiht den Dokumenten eine gewisse Professionalität.

Dieses Leitprogramm zeigt auf, wie eine $\text{\LaTeX}$ -Datei aufgebaut ist. Dazu gehören das Titelblatt, das Inhaltsverzeichnis, Aufzählungen, Bilder etc. Es werden nur die Grundlagen vermittelt. Für weiterführende Erklärungen wird auf das Literaturverzeichnis verwiesen.

¹In diesem Leitprogramm wird der Begriff „Schüler“ für beide Geschlechter verwendet

Vorkenntnisse:

Vertrautheit mit dem Computer wird vorausgesetzt. Auch sollten die Schüler schon mit anderen Textverarbeitungsprogrammen gearbeitet haben, wie zum Beispiel mit Microsoft Word.

1.2 Verwendete Programme

Dieses Leitprogramm zur Einführung von L^AT_EX basiert auf Microsoft Windows.

- **L^AT_EX zu PDF-Konverter:** MikTeX: www.miktex.de
- **Editor:** Windows-Notepad
- **Konsole:** Windows-Eingabeaufforderung (command prompt)
- **PDF-Reader:** Adobe Reader: www.adobe.com

Die hier verwendeten Programme können auch durch andere, gleichwertige Programme unter Windows oder anderen Plattformen ersetzt werden.

1.3 Text- und Zeichenerklärungen

Dieser Abschnitt erklärt einige Textformen und Zeichen, die im Leitprogramm vorkommen.

- **Wörter in eckigen Klammern, zum Beispiel <Ordnername> oder <Datum>:** Die eckigen Klammern haben nichts mit L^AT_EX zu tun, sondern meinen schlicht, dass die Begriffe mit den Klammern durch ein anderes Wort ersetzt werden sollen. Zum Beispiel durch einen Ordnernamen oder ein gültiges Datum. Statt <Ordnername> schreibst du also z.B. Mein_Latex
- **Fettgedruckte Wörter:** Diese stehen für Fachbegriffe wie L^AT_EX-Befehle, Dateinamen etc.
- **Blaue Querbalken:** In den blauen Querbalken steht der L^AT_EX-Code. Oft kommen in den blauen Balken auch Begriffe in eckigen Klammern vor, die dann durch ein anderes Wort ersetzt werden sollen. z.B.:

```
\date {<Datum>}
```

wobei <Datum> durch ein gewünschtes Datum ersetzt werden muss zB:

```
\date {20. Januar 2005}
```

2 L^AT_EX Einführung

L^AT_EX ist ein Textverarbeitungssystem, das Dokumentationen und Arbeiten gut strukturiert darstellen kann. Ob kurze Memos, Briefe, längere Berichte, Studienarbeiten oder ganze Bücher: In jedem Fall ermöglicht L^AT_EX ein professionell gestaltetes Layout. Auch dieses Leitprogramm wurde mit L^AT_EX erstellt.

Donald E. Knuth, der Gründer von T_EX² begann mit der Entwicklung des T_EX-Systems, weil er mit der immer schlechter werdenden typographischen Qualität seiner Bücher unzufrieden war. T_EX ist griechisch und bedeutet Kunst. Das X in 'T_EX' ist der griechische Buchstabe 'Chi' und somit wird T_EX als 'Tech' ausgesprochen.

1984 wurde T_EX von Leslie Lamport weiterentwickelt. Es war die Geburtsstunde von L^AT_EX, das aus **La** von **Lamports** und **T_EX** zusammengesetzt wird.

Leslie Lamport und viele weitere Professoren, Softwareentwickler und Studenten entwickelten über Jahre hinweg das kostenlose L^AT_EX. Die jahrelangen Erfahrungen von Grafikern und Buchautoren wirkten stark bei der Entwicklung von L^AT_EX mit: Der Schreibende kann sich während der Arbeit auf den Inhalt seines Werkes konzentrieren und braucht sich nicht um gestalterische Aspekte zu kümmern. Nebst dem Inhaltsverzeichnis und der Titelseite ermöglicht L^AT_EX auch das Einsetzen von Grafiken, Tabellen, Aufzählungen, Fussnoten, Querverweisen, Silbentrennungen, etc.

Die wesentlichen Daten werden in einer Textdatei angelegt und können in ein beliebiges Format umgewandelt werden, wie zum Beispiel PS (PostScript), PDF³ oder Webseiten (html). In diesem Leitprogramm wird nur die Umwandlung in eine PDF-Datei betrachtet.

Eine besondere Stärke von L^AT_EX ist der mathematische Formelsatz. Darauf wird in diesem Leitprogramm nicht tiefer eingegangen.

²T_EX ist Vorfahre von L^AT_EX

³Siehe Kapitel 3 'Was ist PDF'

Vorteile von L^AT_EX gegenüber üblichen Textverarbeitungssystemen:

Inhaltsverzeichnis: Der Autor einer L^AT_EX-Arbeit muss sich nicht um das Inhaltsverzeichnis kümmern. Die richtige Nummerierung der Kapitel und Unterkapitel, sowie deren Aussehen, wird mit einem Befehl von L^AT_EX selbständig erledigt.

Fussnoten: Sind oft mühsam und unübersichtlich in üblichen Textverarbeitungssystemen. Fussnoten⁴ lassen sich mit einem Befehl von L^AT_EX selbst an die richtige Stelle bringen.

Unabhängig: L^AT_EX ist plattformunabhängig und kann dadurch auf jedem Betriebssystem wie Windows, Linux oder Mac OS bearbeitet werden.

Aussehen: Äusserlich unterscheiden sich die einzelnen L^AT_EX-Arbeiten kaum. Das vermittelt dem Leser den Eindruck sorgfältig erarbeiteter Dokumente.

Open Source: L^AT_EX kann gratis aus dem Internet heruntergeladen werden. Weit verbreitet ist die Software von: www.miktex.de, welche in diesem Leitprogramm verwendet wird.

Elementare Grundbegriffe:

L^AT_EX: Textverarbeitungssystem

PDF: Druckechtes Dokumentenformat

Datei: Es gibt ausführbare Dateien (Programme, z.B.: Internet-Explorer) und nichtausführbare Dateien (z.B.: Textdateien, Musikdateien). Dieses Leitprogramm behandelt nur Textdateien. Dabei handelt es sich um eine Ansammlung von Buchstaben, oder im Grösseren, von Texten.

Konsole: Wörtlich übersetzt: Unterstützer. Aus einer Konsole können Programme gestartet werden (z.B.: den PDF-Konverter)

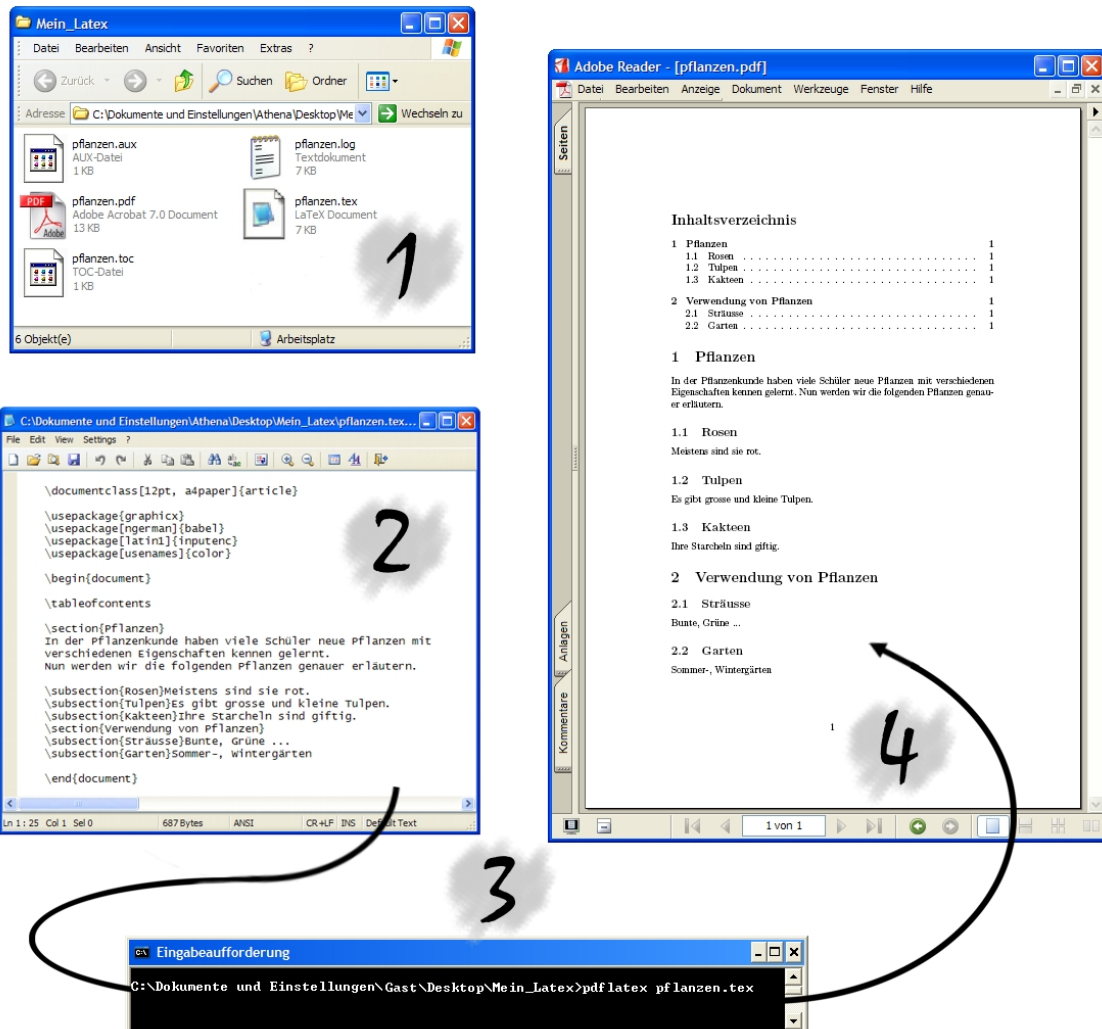
Ordner: In einem Ordner werden Dateien gespeichert und verwaltet.

Editor: Dateien werden in einem Editor erstellt und bearbeitet. Wir verwenden einen Texteditor um den L^AT_EX-Code als Text darin zu programmieren.

⁴Das hier ist eine Fussnote

Übersicht über die zu verwendenden Programme:

Die einzelnen Programme werden im nächsten Kapitel genauer erklärt



1. **Ordner:** Das ist der aktuelle Ordner mit allen Dateien
2. **Editor:** Hier wird der $\text{\LaTeX}$ -Code geschrieben
3. **Konsole:** Damit kann aus einer $\text{\LaTeX}$ -Datei eine lesbare PDF-Datei erstellt werden
4. **PDF:** So sieht die fertige PDF-Datei aus

3 Anwendung

In dieser Lektion lernst du, wie man eine $\text{\LaTeX}$ -Datei erstellt und in eine lesbare Datei umwandelt. Alle $\text{\LaTeX}$ -Zeilen zusammen ergeben eine $\text{\LaTeX}$ -Datei. Man spricht auch von $\text{\LaTeX}$ -Code oder einfach Code. Als lesbare Datei wirst du PDF verwenden. Mehr dazu später.

3.1 Neue Datei beginnen

Als erstes brauchst du einen Editor, in welchen du künftig die $\text{\LaTeX}$ -Dateien schreibst. Unter Windows öffnest du den Editor über das Startmenü. Dort wählst du aus der Liste die Auswahl „Zubehör“. Bei der sich öffnenden Liste wählst du „Editor“. Dieser Vorgang wird in Abbildung 1 dargestellt. Nun öffnet sich ein weisses, leeres Fenster. Dies ist der Editor, wie in Abbildung 2 gezeigt.

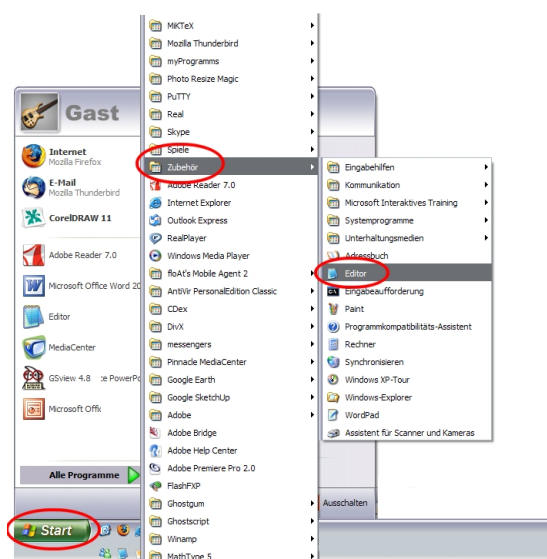


Abbildung 1: Wie man einen Editor öffnet

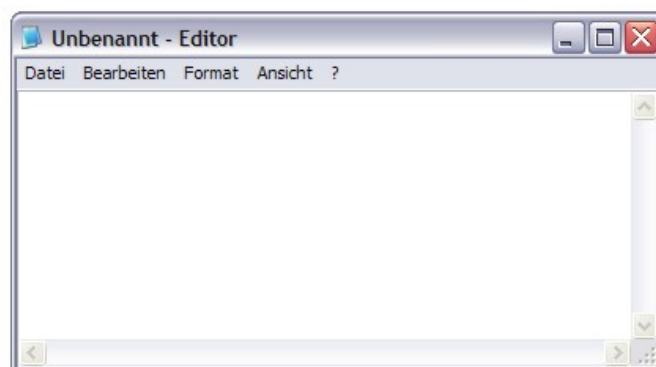


Abbildung 2: Ein Text-Editor

3.2 Neuen Ordner erstellen

Um die künftigen $\text{\LaTeX}$ -Dateien abzuspeichern, benötigt man einen Ordner. Für die kommenden Beispiele und Aufgaben erstellst du diesen neuen Ordner auf dem Desktop. Durch Anklicken der rechten Maustaste auf den Desktop erscheint eine Liste mit möglichen Befehlen. Da du einen neuen Ordner benötigst, wählst du „Neu“ und nachfolgend „Ordner“. In Abbildung 3 wird dieser Vorgang bildlich dargestellt. Gleichzeitig erscheint ein Fenster, in welches du den neuen Ordnernamen eintippen kannst. Benenne diesen Ordner **Mein_Latex** und öffne ihn mit einem Doppelklick. Lass ihn geöffnet, du wirst ihn öfters gebrauchen.

Aufgabe 1 Öffne einen Editor und erstelle einen neuen Ordner **Mein_Latex** auf deinem Desktop, wie oben beschrieben. Editor und Ordner werden im Folgenden weiter verwendet.

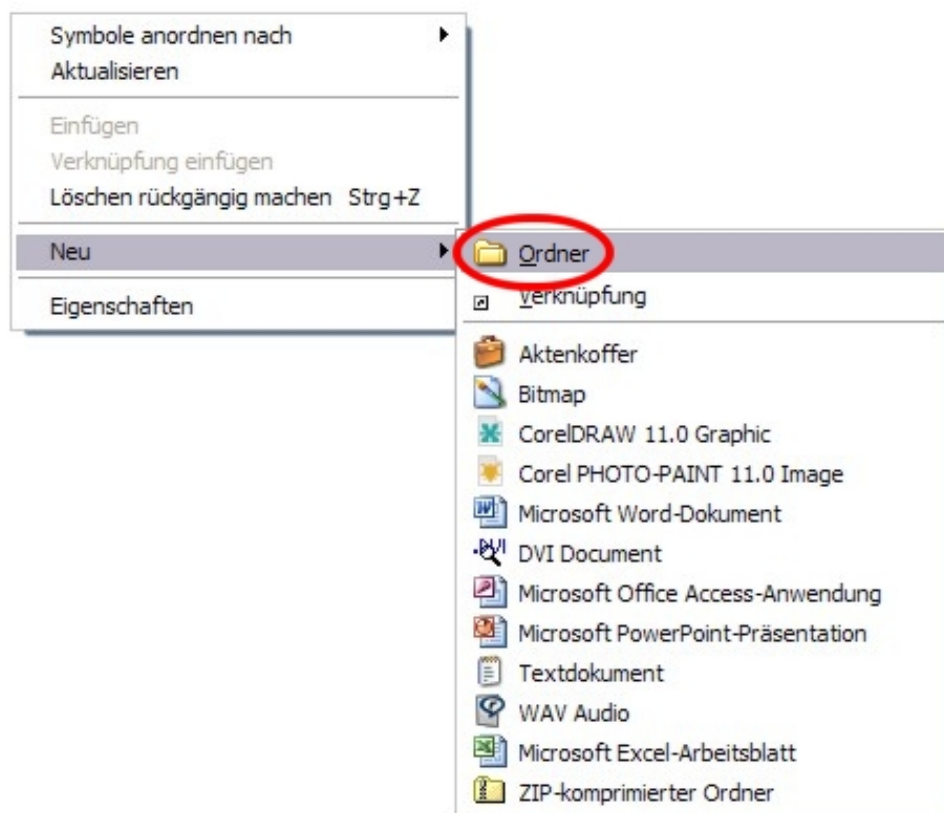


Abbildung 3: So wird ein neuer Ordner auf dem Desktop erstellt

3.3 Datei speichern

Du hast nun einen leeren Editor und einen geöffneten Ordner vor dir. Speichere die $\text{\LaTeX}$ -Datei, die im Editor geschrieben wird, jetzt ab. Und vergiss auch später nicht, die Datei immer wieder abzuspeichern. $\text{\LaTeX}$ -Dateien werden mit der Endung **.tex**⁵ abgespeichert.

Unter dem Menüknopf **Datei** im Editor kannst du deine Arbeit abspeichern. Beim ersten Mal wirst du nach einem Namen für die neue Datei gefragt. Achte unbedingt darauf, dass die Endung **.tex** ist und nicht wie üblich **.txt**. Dazu muss die Einstellung für den Dateityp „All files (*.*)“ sein. Siehe Abbildung 4.

⁵siehe Kapitel 3.5 '.pdf, .tex, .toc, .aux, .log'

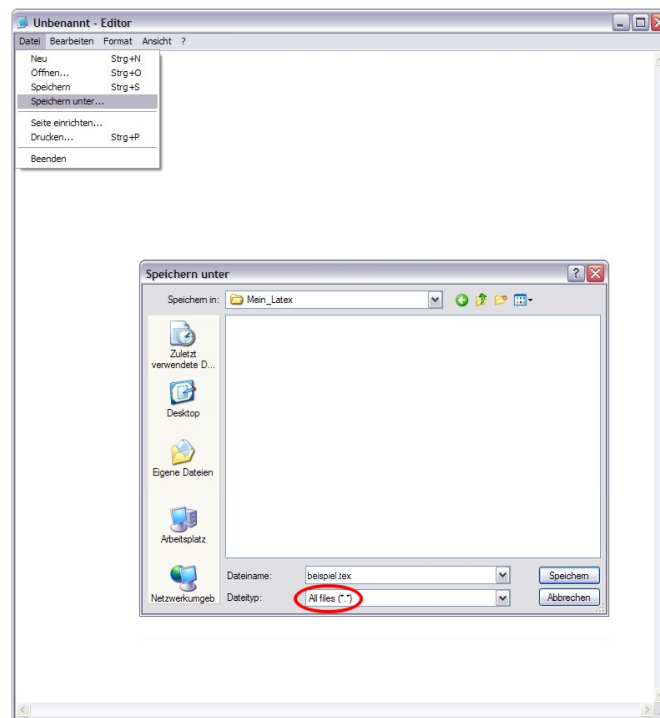
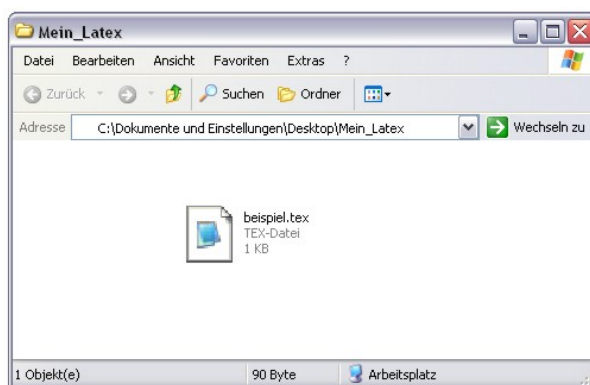


Abbildung 4: Datei abspeichern

Aufgabe 2 Speichere die leere Datei (Editor) mit dem Namen **beispiel.tex** in den neuen Ordner **Mein_Latex**. Abbildung 5 zeigt die abgespeicherte Datei im Ordner. Schau im Ordner nach, ob du die Datei **beispiel.tex** findest.

Abbildung 5: Ordner **Mein_Latex**

3.4 Datei öffnen

Im Normalfall kann eine $\text{\LaTeX}$ -Datei mit einem Doppelklick auf das Dateisymbol geöffnet werden. Falls du eine $\text{\LaTeX}$ -Datei (Endung ist `.tex`) zum ersten Mal öffnest, kann es sein, dass dein Computer diesen Dateityp noch nicht kennt. Er weiss nicht, mit welchem Programm er die Datei öffnen soll. Es erscheint ein Fenster wie in Abbildung 6. Der Computer fragt nach, was er mit der Datei machen soll. Nun klickst du die zweite Auswahl „Programm aus einer Liste auswählen“ an und dann den „OK“-Knopf. Es erscheint nun ein zweites Fenster mit einer Liste aller möglichen Programme. Markiere die Bezeichnung „Editor“ und aktiviere das Kästchen unter der Liste mit einem Klick. Damit ist sichergestellt, dass alle $\text{\LaTeX}$ -Dateien in Zukunft mit dem Editor geöffnet werden. In Abbildung 7 wird dieser Vorgang dargestellt.



Abbildung 6: Dieses Fenster erscheint, wenn der Computer die Datei-Endung nicht kennt.

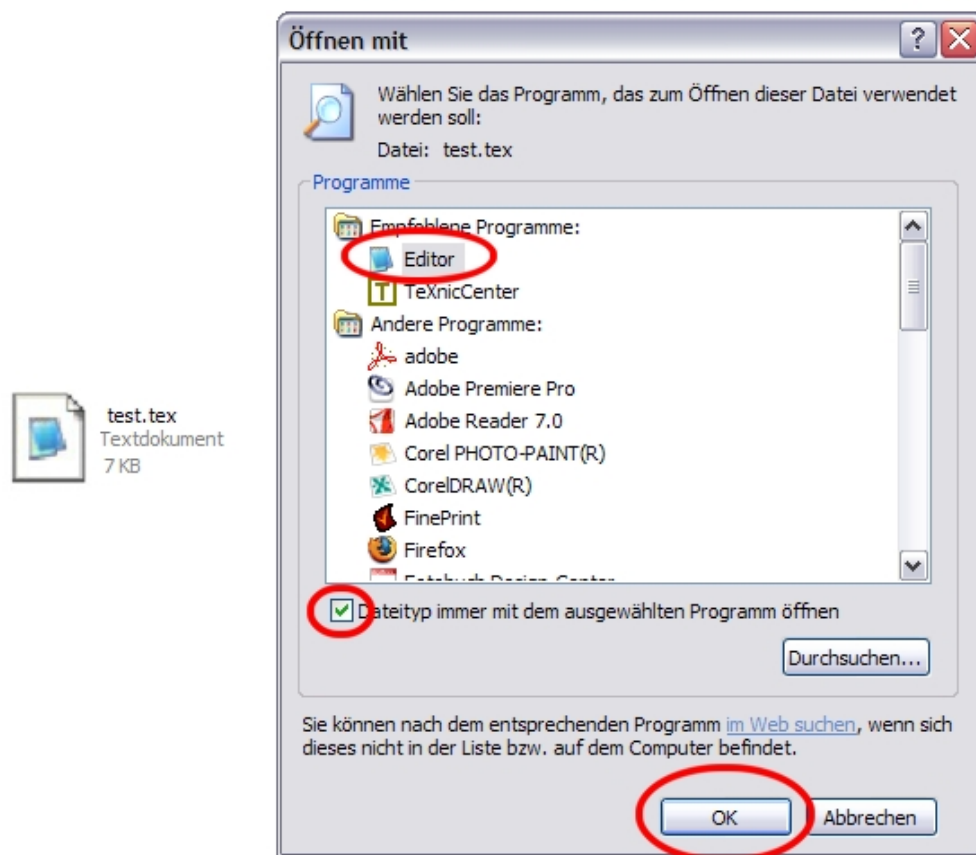


Abbildung 7: Nach einmaliger Programmuweisung werden alle $\text{L}^{\text{T}}\text{E}^{\text{X}}$ -Dateien mit dem Editor geöffnet

Aufgabe 3 Du wirst nun die erste $\text{\LaTeX}$ -Programmierung vornehmen. Schreibe dazu die Zeilen aus der Abbildung 8 in den offenen Editor. Dann die Datei abspeichern (unter gleichem Namen **beispiel.tex**).

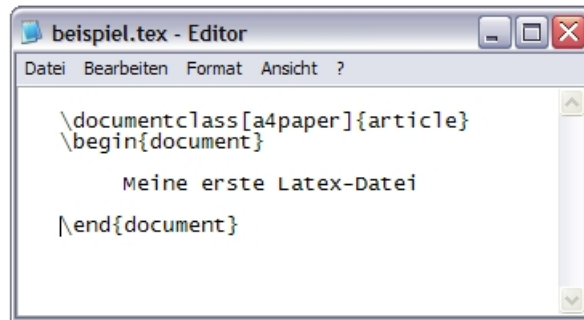


Abbildung 8: $\text{\LaTeX}$ -Datei

3.5 Datei in pdf umwandeln

Was ist PDF: Portable Document Format ist ein Dokumentformat, das druckecht ist. Das heisst, dass die Datei auf dem Bildschirm genau gleich aussieht wie auf jedem Ausdruck auf Papier.

Im Allgemeinen kann an einer PDF-Datei - im Gegensatz zu einem Word Dokument - nichts verändert werden.

Nur wer die zugehörige Datei, zB. eine $\text{\LaTeX}$ -Datei hat, kann Änderungen vornehmen. Der Autor eines PDF-Dokuments kann zudem bestimmen, welche Personen das Dokument lesen oder drucken dürfen. Ein weiterer Pluspunkt von PDF besteht darin, dass die Datei plattformunabhängig ist. Sie kann unter Windows, Linux oder Mac gelesen werden.

Wie erhalte ich eine PDF-Datei? Du wirst in diesem Leitprogramm lernen, eine $\text{\LaTeX}$ -Datei in eine PDF-Datei umzuwandeln. In der Informatik spricht man bei diesem Vorgang von 'kompilieren'. Der Computer erledigt diese schwierige Arbeit für uns. Dazu benötigst du ein wichtiges Instrument: die Konsole. Sie ist ein schwarzes Fenster mit weisser Schrift. Mit ihr kann man durch Ordner wandern, Dateien anschauen, Befehle und Programme ausführen. All diese Operationen werden mittels Befehlen in die Konsole eingegeben. Jeder Befehl muss mit einem **Enter** bestätigt werden. Erst dann wird der Befehl ausgeführt.

Wie öffne ich eine Konsole? Eine Konsole öffnet man über das Startmenü unten links. Drücke den Startknopf und dann den Knopf „Ausführen...“. Wie in Abbildung 9 ersichtlich, öffnet sich dann ein weiteres Fenster mit dem Titel „Ausführen“. Um die Konsole zu öffnen, tippst du die drei Buchstaben „cmd“ ein, was für „Command“ steht. „Commands“ sind Befehle, die man mit Hilfe der Konsole ausführen kann.

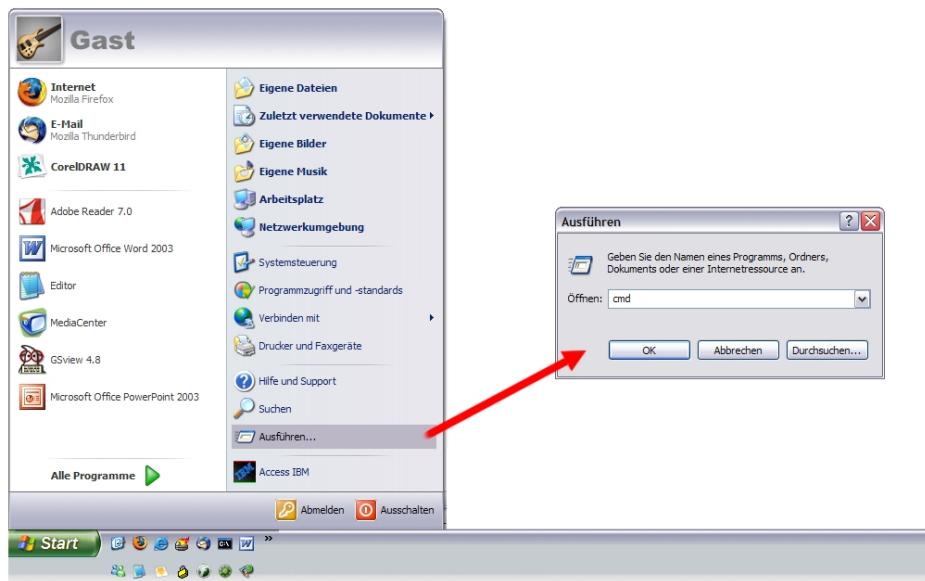


Abbildung 9: Öffnen einer Konsole

Drückt man nach Eingabe von „cmd“ den OK-Knopf, verschwindet das aktuelle Fenster und es erscheint ein schwarzes Fenster, die Konsole. In Abbildung 10 ist diese Konsole abgebildet. Die ersten zwei Zeilen „Microsoft Windows ...“ sind Angaben zur Konsole und für dich nicht weiter interessant.

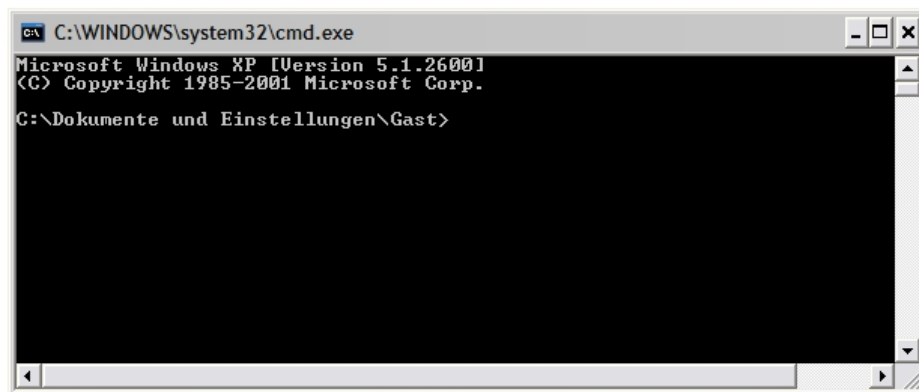


Abbildung 10: Neue Konsole

Beachte die letzte Zeile:

```
C:\Dokumente und Einstellungen\Gast>
```

Diese Zeile kann von deiner Zeile abweichen. Zum Beispiel wird anstelle von „Gast“ dein Name stehen. Lass dich von dieser Abweichung nicht stören. Du wirst nun lernen, wie man mit der Konsole arbeitet. Du befindest dich im Ordner „Gast“, welcher den Ordner „Desktop“ enthält, einen so genannten Unterordner. In Abbildung 11 wird diese Ordnerstruktur bildlich dargestellt. Jede Bezeichnung zwischen zwei Schrägstrichen, z.B. „\ **Dokumente und Einstellungen** \“ stellt einen eigenen Ordner dar.

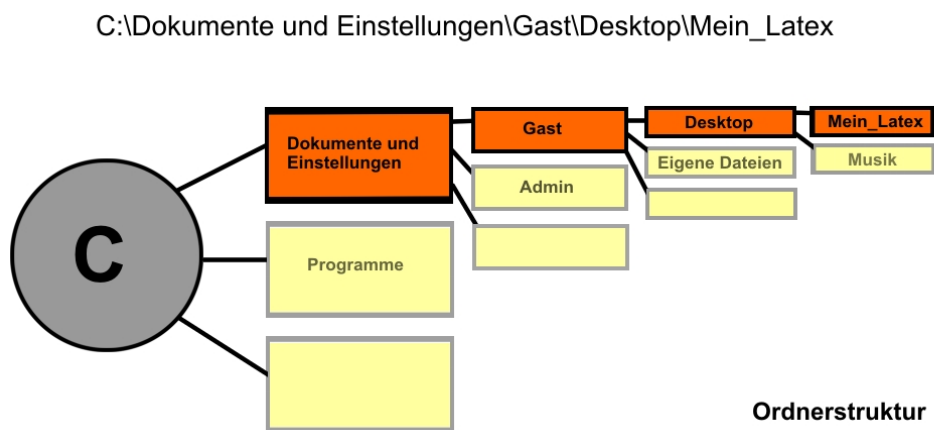


Abbildung 11: Ordnerstruktur

Alle Befehle, die man in die Konsole eingibt, werden auf den Ordner angewendet, in dem man sich gerade befindet. Du befindest dich nun im Ordner „Gast“ und möchtest in den Ordner „Desktop“ gelangen. Dazu benötigt man folgenden Befehl:

```
cd <Ordnername> (Enter)
```

Damit gelangst du in den Ordner mit dem Namen **Ordnername**. Der Befehl „cd“ ist die Abkürzung für „change directory“ was „wechsle den Ordner“ bedeutet. Tippe nun Folgendes in die Konsole ein:

```
cd Desktop (Enter)
```

Nun befindest du dich auf dem Desktop. Dieser Desktop ist der gleiche Desktop, der im Hintergrund erscheint, wenn du alle Fenster schließt und kein Programm geöffnet hast. Natürlich existieren auch Befehle, um einen

Ordner wieder zu verlassen. Tippe dazu zwei Punkte:

```
cd .. (Enter)
```

Damit gelangt man in den Ordner „Gast“ zurück, klettert also eine Hierarchiestufe höher. In Kapitel 3.2 hast du einen neuen Ordner „Mein_Latex“ auf dem Desktop erstellt. Um deine Datei „beispiel.tex“ in eine PDF-Datei umzuwandeln, musst du in diesen Ordner wechseln. Falls du dich noch im Ordner „Gast“ befindest (das siehst du an der Endung der letzten Zeile), gib folgenden Befehl ein:

```
cd Desktop (Enter)
```

Nun befindest du dich wieder auf dem Desktop. Mit dem folgenden Befehl gelangst du in den Mein_Latex-Ordner:

```
cd Mein_Latex (Enter)
```

Den Ordernamen „Mein_Latex“ musst du genau so eingeben, er darf keine Fehler enthalten. Gross- und Kleinschreibung sowie Leerzeichen müssen strikte übernommen werden, ansonsten wird die Konsole eine Fehlermeldung anzeigen und den Befehl nicht ausführen. Um den Inhalt des Ordners anzuzeigen, in dem du dich gerade befindest, wird folgender Befehl benutzt:

```
dir (Enter)
```

Alle enthaltenen Dateien und Ordner werden in einer Listenansicht angezeigt. Dieser Befehl steht für „directory“ oder auf Deutsch „Verzeichnis/Ordner“. Du wirst auch die Datei „beispiel.tex“ entdecken. Diese soll nun in eine PDF-Datei umgewandelt werden. Dazu benötigst du folgenden Befehl:

```
pdflatex <Dateiname>.tex (Enter)
```

Die Datei mit dem angegebenen Dateinamen wird kompiliert und somit in eine PDF-Datei umgewandelt.

Eine Zwischenbemerkung: Solltest du einmal einen Fehler tippen oder wieder zur Ausgangslage zurückgehen wollen, drückst du folgende Tastenkombination

```
Ctrl c
```

Diese Tastenkombination erscheint nicht in der Konsole, sondern bringt dich lediglich zur Ausgangslage zurück.

Für unser Beispiel gibst du nun Folgendes in die Konsole ein:

```
pdflatex beispiel.tex (Enter)
```

Nun erscheinen viele Zeilen in der Konsole, die bei normaler Umwandlung mit dem Befehl: **Transcript written on beispiel.log** enden.

Da $\text{\LaTeX}$ zwei Schritte benötigt, um alle Bilder und das Inhaltsverzeichnis zu erstellen, musst du die gleiche Datei stets zweimal kompilieren.

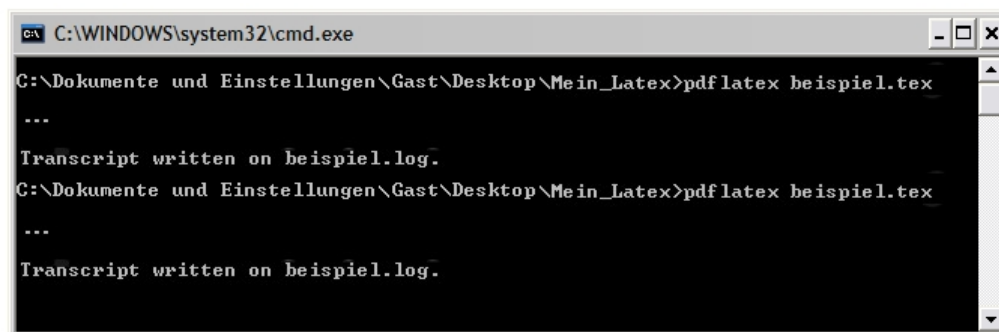


Abbildung 12: Wichtig: Die Datei immer zweimal kompilieren lassen

In Abbildung 12 wird dieser Vorgang ersichtlich. Wurde die erste Kompilation erfolgreich abgeschlossen, kompiliert man die gleiche Datei mit der gleichen Befehlsangabe noch ein zweites Mal.

Nun entstehen im Ordner **Mein_Latex** automatisch vier neue Dateien. Siehst du die Datei **beispiel.pdf**? Öffne sie mit einem Doppelklick und schau nach, was du in $\text{\LaTeX}$ geschrieben hast.

Gratulation, du hast deine erste $\text{\LaTeX}$ -Datei erstellt!

Diese PDF-Datei muss stets wieder geschlossen werden, bevor die Datei neu kompiliert wird. Wird dies vergessen, erinnert die Zeile „I can’t write on File“ daran. Schliesse in einem solchen Fall die offene PDF-Datei und kompiliere erneut. Danach wird es funktionieren.

Aufgabe 4 Kompiliere die Datei **beispiel.tex** anhand des unter dem Kapitel 3.4 'Datei in PDF umwandeln' beschriebenen Vorganges in eine PDF-Datei. Öffne diese PDF-Datei. In Abbildung 13 siehst du das Resultat.

3.6 .pdf, .tex, .toc, .aux, .log

Im Ordner **Mein_Latex**, wie in Abbildung 14, erscheinen folgende Dateien:



Abbildung 13:

beispiel.tex: Deine eigenhändig geschriebene $\text{\LaTeX}$ -Datei.

beispiel.pdf: Die druckbare Datei der $\text{\LaTeX}$ -Datei.

beispiel.aux: Eine Hilfsdatei, in der wichtige Informationen, wie zum Beispiel Kapitel, Bilder etc. gespeichert sind.

beispiel.log: Für Rückmeldungen und Fehlermeldungen.

beispiel.toc: Enthält das Inhaltsverzeichnis. Diese Datei wird nur erstellt, wenn im Dokument ein Inhaltsverzeichnis vorhanden ist.

Die Dateien **.toc**, **.aux** und **.log** werden nur vom Computer benutzt. Du brauchst dich nicht um sie zu kümmern.

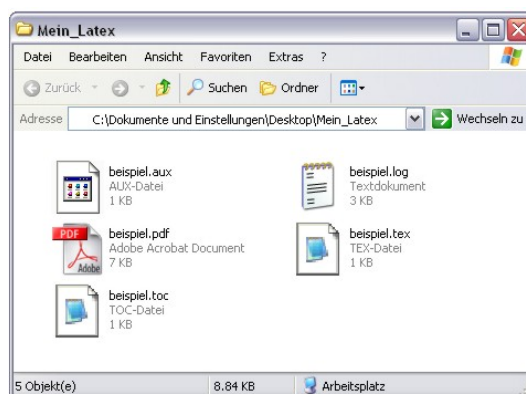


Abbildung 14: Unser Ordner nach dem Kompilieren

3.7 Zusammenfassung

Du hast nun ein Textverarbeitungssystem kennen gelernt, mit dessen Hilfe du eine einheitlich formatierte Dokumentation erstellen kannst. Angefangen mit einigen einfachen $\text{\LaTeX}$ -Zeilen in einem Texteditor hast du die Datei mit der Endung **.tex** in einem Ordner abgespeichert. Dann hast du diese **.tex**-Datei mit Hilfe einer Konsole in eine lesbare PDF-Datei umgewandelt.

Schritt für Schritt Anleitung zur Erstellung einer PDF-Datei ausgehend von einer $\text{\LaTeX}$ -Datei.

- **Editor** öffnen
- $\text{\LaTeX}$ -Code in den **Editor** schreiben
- Diese Datei mit Endung **.tex** speichern
- Konsole öffnen
- In den richtigen Ordner springen
- Mit **pdflatex <Dateiname>.tex** $\text{\LaTeX}$ -Datei in der Konsole in PDF umwandeln

3.8 Kapitelprüfung

Erstelle eine $\text{\LaTeX}$ -Datei mit unten stehendem Inhalt aus Abbildung 15. Speichere sie unter dem neuen Name **Kapitel3.tex** in den Ordner **Mein_Latex** ab und wandle sie in eine PDF-Datei um.

Der Befehl

```
\newline
```

oder

```
\\
```

erzeugt eine neue Linie, wie die Enter-Taste im Word. Der nachfolgende Text fängt auf einer neuen Zeile an.

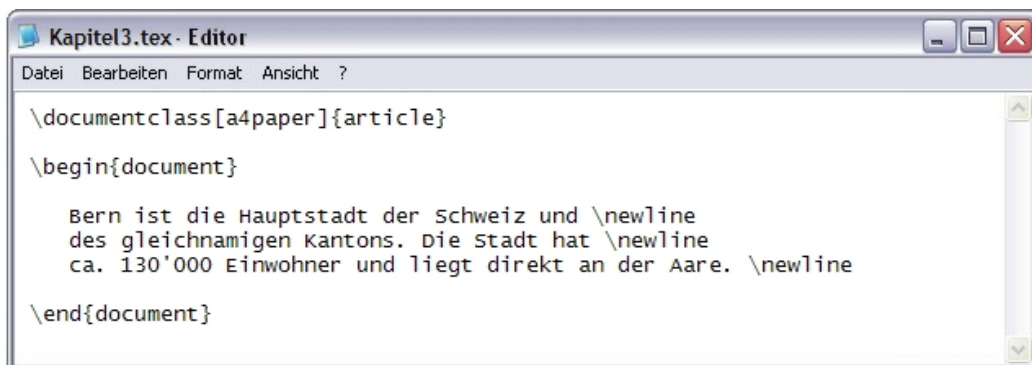


Abbildung 15: Inhalt für Kapitelprüfung 3

4 Aufbau

Dieses Kapitel zeigt die Vorzüge von $\text{\LaTeX}$ auf. $\text{\LaTeX}$ eignet sich für das Verfassen eines Buches, einer Dokumentation oder auch eines Blattes. Das Erstellen eines Inhaltsverzeichnisses wird dir nach dieser Lektion einfach fallen. Ebenso deinen Namen mit Datum richtig auf der Vorderseite der Arbeit zu platzieren.

4.1 Grundgerüst

Jede $\text{\LaTeX}$ -Datei hat folgendes Gerüst:

```
\documentclass[<option>]{<class>}
\begin{document}
Text
\end{document}
```

In diesem Code stehen zwei Werte (**option** und **class**), die zuerst gesetzt werden müssen. So wie hier aufgeführt, funktioniert der Code nicht.

class: Bezeichnet den Typ der Arbeit. Zum Beispiel ein Buch oder eine Folienpräsentation. Zu Beginn der Arbeit muss demnach stets festgelegt werden, um was für eine Arbeit es sich handelt.

<class> muss durch eine der folgenden Möglichkeiten ersetzt werden:

- **article:** Für Artikel, Vorträge, Semesterarbeiten, Einladungen, etc. Wird am Häufigsten verwendet.
- **report:** Für längere Berichte, Diplomarbeiten, Skripte
- **book:** Für Bücher
- **letter:** Für Briefe
- **slides:** Für Präsentationsfolien

Dieses Leitprogramm ist als **article** geschrieben. Da dies die meist genutzte **class** ist, werden alle weiteren Beispiele und Aufgaben darauf aufbauen.

option: Hier wird festgelegt, wie die Datei auszusehen hat. Jede **class** kann mit einer, mehreren oder gar keiner Option erweitert werden. Man legt die Schriftgröße, Seitengröße etc. für alle folgenden Seiten des Dokuments fest. Beispiele für **option** sind Folgende:

- **10pt:** Setzt die Schriftgrösse auf 10 Punkte (1 Punkt = 1/3 mm)
- **12pt:** Setzt die Schriftgrösse auf 12 Punkte
- **a4paper:** Es wird nur A4 (normales Arbeitspapier) benutzt.
- **letterpaper:** Diese Papiergrösse wird meist in Amerika verwendet.

Werden keine Optionen gewählt, so setzt $\text{\LaTeX}$ 'Standardwerte' ein, so genannte Default-Werte. Jede $\text{\LaTeX}$ -Datei ist auf folgende Default-Werte gesetzt: 10pt, letterpaper

Es können mehrere Optionen mit Komma getrennt ausgewählt werden. Ein Beispiel hierfür mit neuen Werten:

```
\documentclass[12pt,a4paper]{<class>}
```

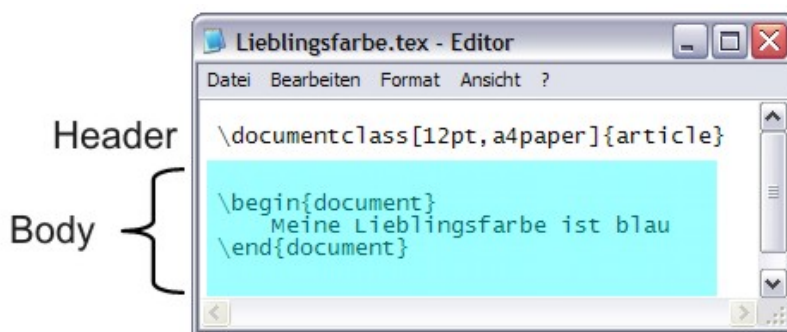


Abbildung 16: Gerüstaufbau

Header: Alles rund um das Aussehen und der Typ der Arbeit wird im Header definiert. Mit **option** und **class** können diese Werte gesetzt werden.

Body: Der restliche Textinhalt wird zwischen **begin** und **end** des Dokuments geschrieben. Nach dem **end** darf nichts mehr stehen.

Mit diesem Grundgerüst wie in Abbildung 16 erledigt $\text{\LaTeX}$ schon den grössten Teil der Arbeit. Das Ergebnis als PDF-Datei ist in Abbildung 17 ersichtlich.



Abbildung 17: Erzeugte PDF-Datei aus Code in Abbildung 16

Aufgabe 5 Erstelle das Grundgerüst für ein Buch mit Schriftgröße 12 und A4-Papier.

Aufgabe 6 Wie sieht der $\text{\LaTeX}$ -Code aus für eine Folienpräsentation?

Aufgabe 7 Können Header und Body vertauscht werden? Begründe deine Antwort.

4.2 Package

Würdest du nun dieses Leitprogramm zur Seite legen und auf eigene Faust zu programmieren beginnen, sähest du dich bald mit einigen Problemen konfrontiert: Die Basis von $\text{\LaTeX}$ beinhaltet nur schwarzen Text, keinerlei Graphiken und erkennt auch keine sprachbezogenen Eigenschaften wie zum Beispiel Silbentrennungen, Umlaute (ä, ö, ü) etc. Dazu benötigt man Erweiterungen bzw. Anpassungen, die Zusatzpakete oder **Packages** genannt werden.

Es gibt unzählige **Packages**, wobei dieses Leitprogramm nur auf die Wichtigsten eingeht.

Die Auflistung der **Packages** erfolgt im Header (siehe Abbildung 18) und jedes beginnt auf einer neuen Zeile.

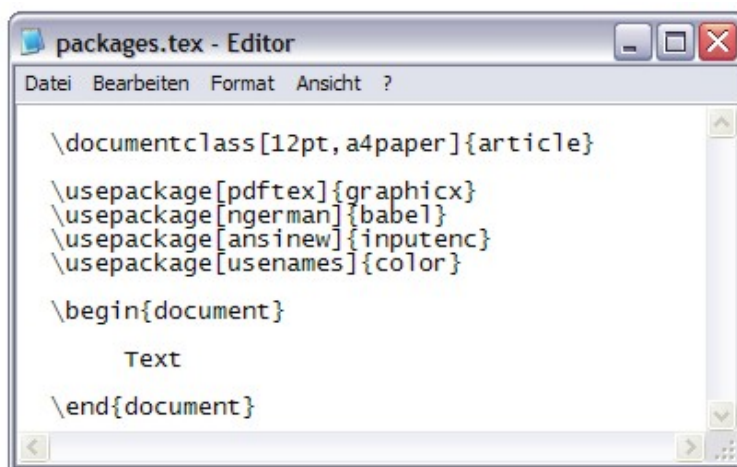


Abbildung 18: $\text{\LaTeX}$ -Code mit eingebundenen **Packages**

```
\usepackage[<option>]{<name>}
```

Nun wird ein **Package** mit dem Namen '**name**' und der Option '**option**' in den Header des $\text{\LaTeX}$ -Codes eingebunden. Natürlich müssen diese zwei Werte richtig gesetzt werden.

Für sprachbezogene Eigenschaften werden folgende **Packages** verwendet:

```
\usepackage[<language>]{babel}
```

wobei **language** durch eine der folgenden Sprachen ersetzt werden muss:

- **ngerman**: Deutsch
- **frenchb**: Französisch
- **portuguese**: Portugiesisch
- **etc.**: ...

```
\usepackage[<os>]{inputenc}
```

ist für die Umlaute zuständig. Hier muss explizit angegeben werden, welches Betriebssystem **<os>** verwendet wird, da die Tastaturen verschieden eingestellt sind.

- **ansinew**: Windows
- **latin1**: Unix
- **applemac**: Mac

Um die ganze Arbeit mit Bildern, farbiger Schrift und mathematischen Ausdrücken zu erweitern, werden folgende **Packages** benötigt:

- **amsmath**: Für mathematische Symbole
- **color**: Mit der Option **usenames** sind farbige Schriften zugelassen
- **graphicx**: Um Bilder und Grafiken einzufügen. Mit der Option **pdftex** sind die wichtigen Formate .jpg, .png und .pdf erlaubt.

Abbildung 18 zeigt eine Arbeit mit den üblichen eingebundenen **Packages**.

Aufgabe 8 Erstelle den $\text{\LaTeX}$ -Code für ein Buch der Schriftgröße 10pt. Bilder und Grafiken müssen ermöglicht sein.

Aufgabe 9 Programmiere den $\text{\LaTeX}$ -Code für einen französischen Artikel, der auf einem Unix-Rechner geschrieben wird.

4.3 Titel, Autor, Datum

Das Deckblatt einer $\text{\LaTeX}$ -Arbeit (mit `class: article`) enthält den Titel der Arbeit, den Namen des Autors und das Erstellungsdatum. Die Befehle für das Deckblatt lauten:

```
\title{Arbeit über Pflanzen}  
\author{Monika Muster}
```

Mit diesen zwei Zeilen definiert man den Titel der Arbeit (hier: 'Arbeit über Pflanzen'), den Autor (hier: 'Monika Muster') und automatisch das Tagesdatum. Der $\text{\LaTeX}$ -Code und die zugehörige PDF-Datei sind in Abbildung 19 ersichtlich.

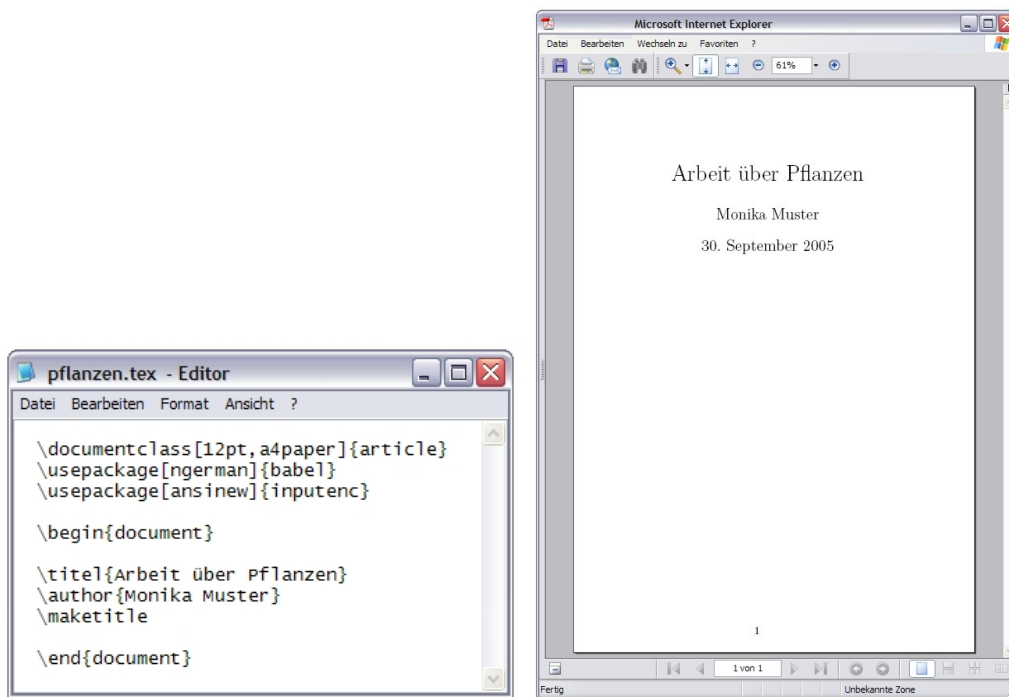


Abbildung 19: Code für Pflanzenarbeit

Nun sind Titel und Autor gesetzt, werden aber erst mit der folgenden Zeile auf dem Deckblatt sichtbar.

```
\maketitle
```

Das Datum wird mit der Zeile `\maketitle` automatisch auf das Tagesdatum gesetzt.

Möchte man ein anderes Datum setzen, wird der Befehl

```
\date{<Datum>}
```

benötigt. <Datum> wird durch das gewünschte Datum ersetzt. Es wird genau so auf dem Deckblatt erscheinen, wie man es eingibt. Damit das Titelblatt wirklich nur den Titel, den Autoren und das Datum enthält, muss man L^AT_EX eine neue Seite aufzwingen. Mit dem Befehl:

```
\newpage
```

wird an dieser Stelle eine neue Seite erzeugt und auf dieser fortgefahren. Folgendes Beispiel erstellt ein Titelblatt mit gewünschtem Datum:

```
\title{Arbeit über Pflanzen}  
\author{Monika Muster}  
\date{18.5.2000}  
\maketitle  
\newpage
```

Aufgabe 10 Wo im Code muss/kann `\date{4. Juni 2005}` stehen? Im Body, im Header oder spielt es keine Rolle? Begründe deine Antwort.

Aufgabe 11 Wie sieht der L^AT_EX- Code aus für einen Artikel der Schriftgröße 12 mit dem Titel: 'Paradiso' und dem Autor 'Lesly Porton'? Es sollen Graphiken ermöglicht und der Text in deutscher Sprache schreibbar sein.

Aufgabe 12 Programmiere den vollständigen L^AT_EX-Code für folgende PDF-Datei in Abbildung 20.



Abbildung 20: Titelblatt einer Arbeit

4.4 Inhaltsverzeichnis und Struktur des Dokuments

Für ein sauberes Inhaltsverzeichnis benötigt man nur diese Codezeilen:

```
\tableofcontents  
\newpage
```

Der erste Befehl teilt $\text{\LaTeX}$ mit, dass man ein Inhaltsverzeichnis auf der ersten, respektive zweiten Seite wünscht, je nachdem ob ein Deckblatt verwendet wird. Der Befehl für eine neue Seite stellt sicher, dass das Inhaltsverzeichnis vom nachfolgenden Text getrennt wird.

$\text{\LaTeX}$ kennt **Kapitel** (fett geschrieben) und Unterkapitel (eingerrückt), so genannte **sections** und subsections. Diese müssen alle aufgelistet werden, damit $\text{\LaTeX}$ ein Inhaltsverzeichnis erstellen kann.

Der Befehl für ein Kapitel lautet:

```
\section{<titel>}<inhalt>
```

und für ein Unterkapitel:

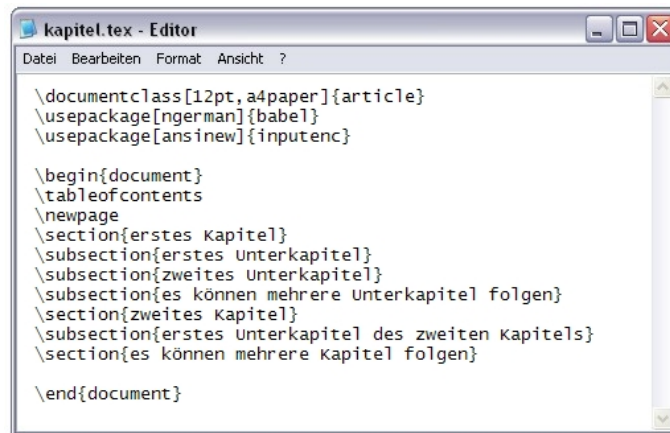
```
\subsection{<titel>}<inhalt>
```

<titel> wird durch einen geeigneten Titel ersetzt, welcher im Inhaltsverzeichnis und in der Arbeit selbst erscheint. Der **<titel>** eines Kapitels (section) erscheint auf jeder Seite als erste Zeile⁶. Und zwar bis ein neues Kapitel folgt. Der Leser weiss auf diese Weise stets, in welchem Kapitel er sich befindet.

<inhalt> wird durch den gesamten Textinhalt ersetzt, welcher in diesem Kapitel/Unterkapitel folgt. Siehe dazu Abbildung 21.

Nochmals zur Klärung: $\text{\LaTeX}$ benötigt nur eine Codezeile, um das Inhaltsverzeichnis zu erstellen. Die darauf folgenden „sections“ und „subsections“ und deren Inhalte stellen bereits die Struktur und den Inhalt des Dokumentes dar. Ihnen entnimmt $\text{\LaTeX}$ die Informationen für das Inhaltsverzeichnis. Somit werden die Kapitel- und Unterkapitel-Titel nur einmal geschrieben, obwohl sie dann in der fertigen PDF-Datei zweimal erscheinen: einmal im Inhaltsverzeichnis und einmal auf der jeweiligen Seite als Überschrift.

⁶Siehe Kapitelüberschrift 'Aufbau' am Anfang dieser Seite



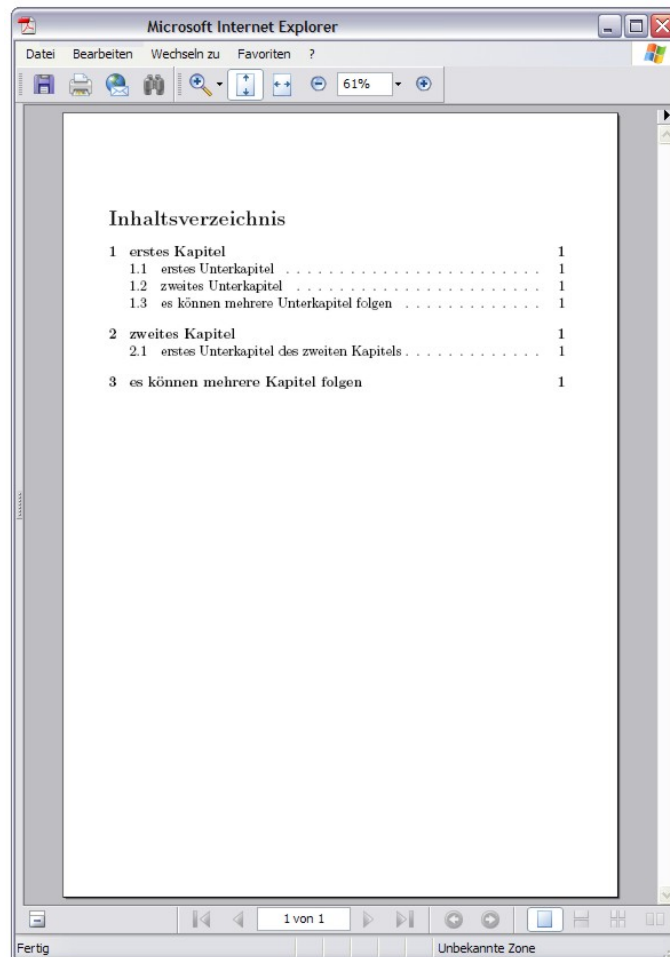
```

\documentclass[12pt,a4paper]{article}
\usepackage[ngerman]{babel}
\usepackage[ansinew]{inputenc}

\begin{document}
\tableofcontents
\newpage
\section{erstes Kapitel}
\subsection{erstes Unterkapitel}
\subsection{zweites Unterkapitel}
\subsection{es können mehrere Unterkapitel folgen}
\section{zweites Kapitel}
\subsection{erstes Unterkapitel des zweiten Kapitels}
\section{es können mehrere Kapitel folgen}

\end{document}

```



Inhaltsverzeichnis	
1	erstes Kapitel 1
1.1	erstes Unterkapitel 1
1.2	zweites Unterkapitel 1
1.3	es können mehrere Unterkapitel folgen 1
2	zweites Kapitel 1
2.1	erstes Unterkapitel des zweiten Kapitels 1
3	es können mehrere Kapitel folgen 1

Abbildung 21: $\text{\LaTeX}$ -Code mit Inhaltsverzeichnis

Aufgabe 13 Programmiere mit L^AT_EX ein Inhaltsverzeichnis mit folgender Struktur (7 Kapitel mit den jeweiligen Unterkapiteln):

- Montag
 - Fussballtraining
 - Handarbeit
- Dienstag
 - Klavier
- Mittwoch
 - Mittagessen bei Müller
 - Abendessen bei Kuster
- Donnerstag
 - Handballtraining
 - Minigolf
- Freitag
 - Frei
- Samstag
 - Zelten in Genf
- Sonntag
 - Frühstück
 - Fernsehen

Aufgabe 14 Was geschieht, wenn der Befehl

```
\tableofcontents
```

nicht geschrieben wird?

4.5 Zusammenfassung

Du hast gelernt, eine einfache L^AT_EX-Datei mit Titel, Autor und Inhaltsverzeichnis zu erstellen. Ausserdem hast du geübt, **Packages** einzubinden, welche man zum Beispiel für die Silbentrennung einer Sprache benötigt.

4.6 Kapitelprüfung

1. Schreibe einen L^AT_EX-Code für folgende Bedingungen:

- article
- Schriftgrösse: 12 pt
- letterpaper
- in deutscher Sprache (alles was dazu benötigt wird einbinden)
- Autor: Jan Seeberg
- Titel: Arbeit über fahrbare Vehikels
- Datum: 28. Juni 2010
- neue Seite
- Mit 5 Kapiteln:
 1. Auto
 2. Velo
 3. Mofa
 4. Zug
 5. Schiff

wobei das Kapitel Auto zwei Unterkapitel (Personenwagen und Bus) enthält

- ohne Bilder

2. Schreibe ein $\text{\LaTeX}$ -Dokument, das folgende vierseitige PDF-Datei erzeugt. Benutze den Befehl

```
\newpage
```

um an beliebiger Stelle im Text eine neue Seite zu erzeugen.

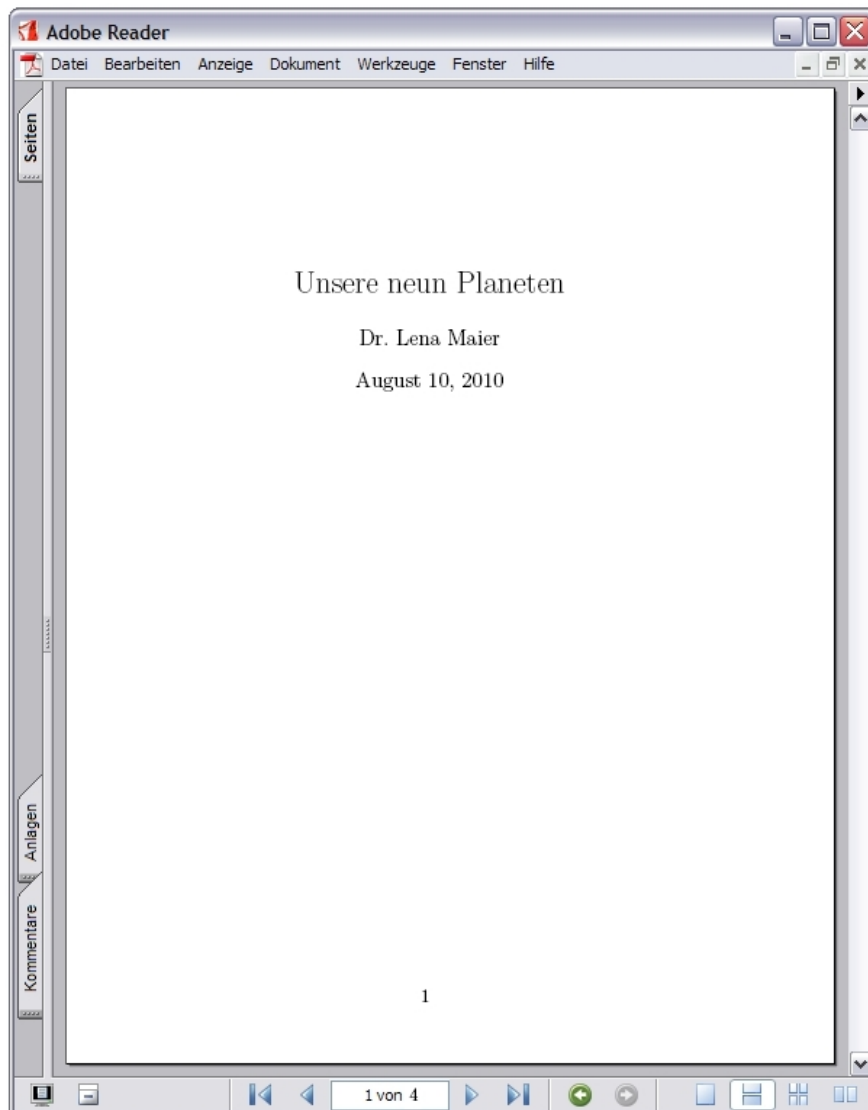
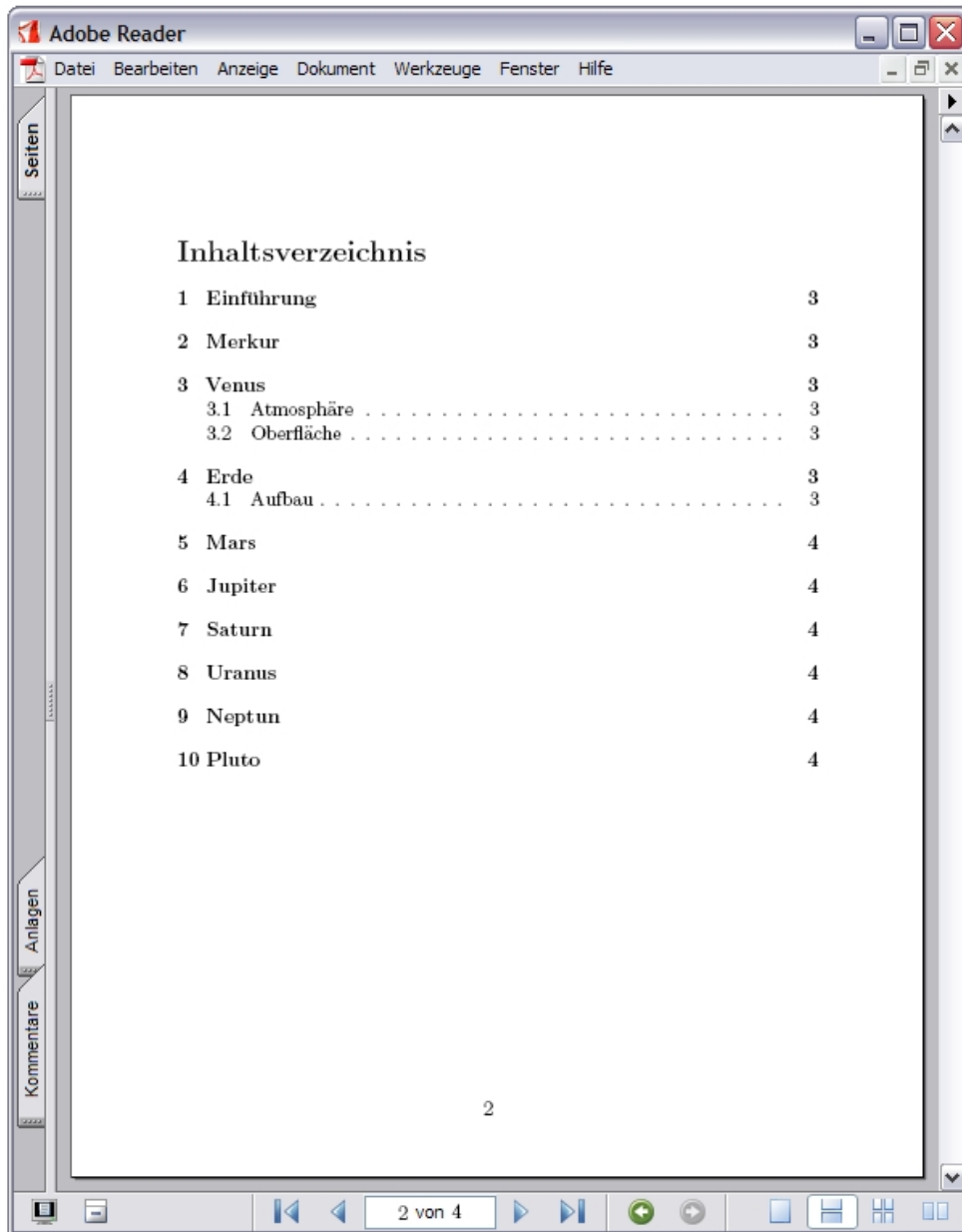


Abbildung 22: Titelblatt



Inhaltsverzeichnis	
1 Einführung	3
2 Merkur	3
3 Venus	3
3.1 Atmosphäre	3
3.2 Oberfläche	3
4 Erde	3
4.1 Aufbau	3
5 Mars	4
6 Jupiter	4
7 Saturn	4
8 Uranus	4
9 Neptun	4
10 Pluto	4

2

Abbildung 23: Inhaltsverzeichnis

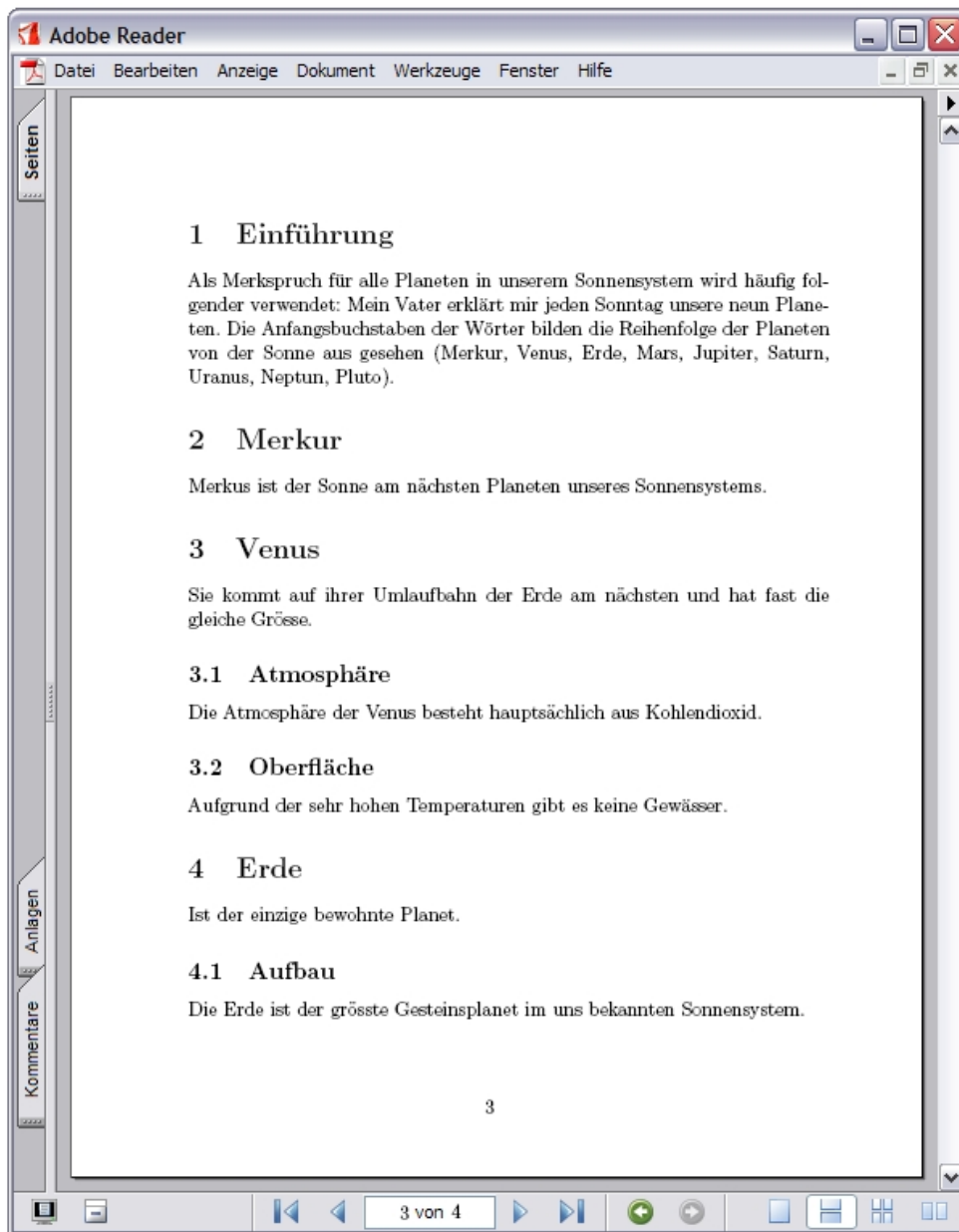


Abbildung 24: Textseite

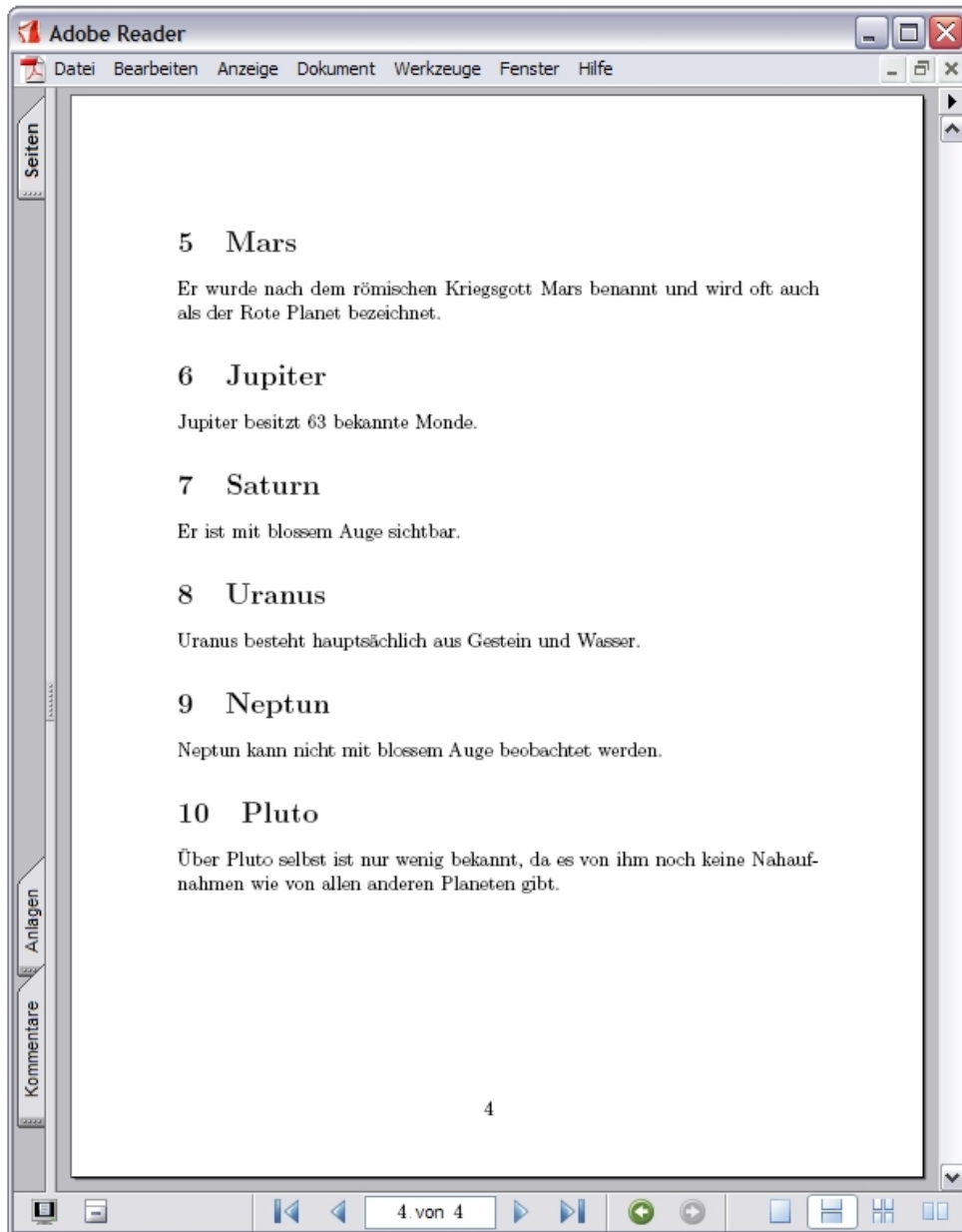


Abbildung 25: Textseite

5 Datei-Format

5.1 Textformatierung

L^AT_EX erstellt Text standardmässig im Blocksatz, also links- und rechtsbündig (so wie dieser Text). Soll der Text links-, rechtsbündig oder zentriert erscheinen, wird dies durch folgende Befehle ermöglicht:

```
\begin{<Texttype>}  
...  
\end{<Texttype>}
```

Als **Texttype** können folgende Optionen gewählt werden:

- **center**: zentriert
- **flushleft**: linksbündig
- **flushright**: rechtsbündig

In den Abbildungen 26,27,28 und 29 werden die einzelnen Begriffe erläutert.

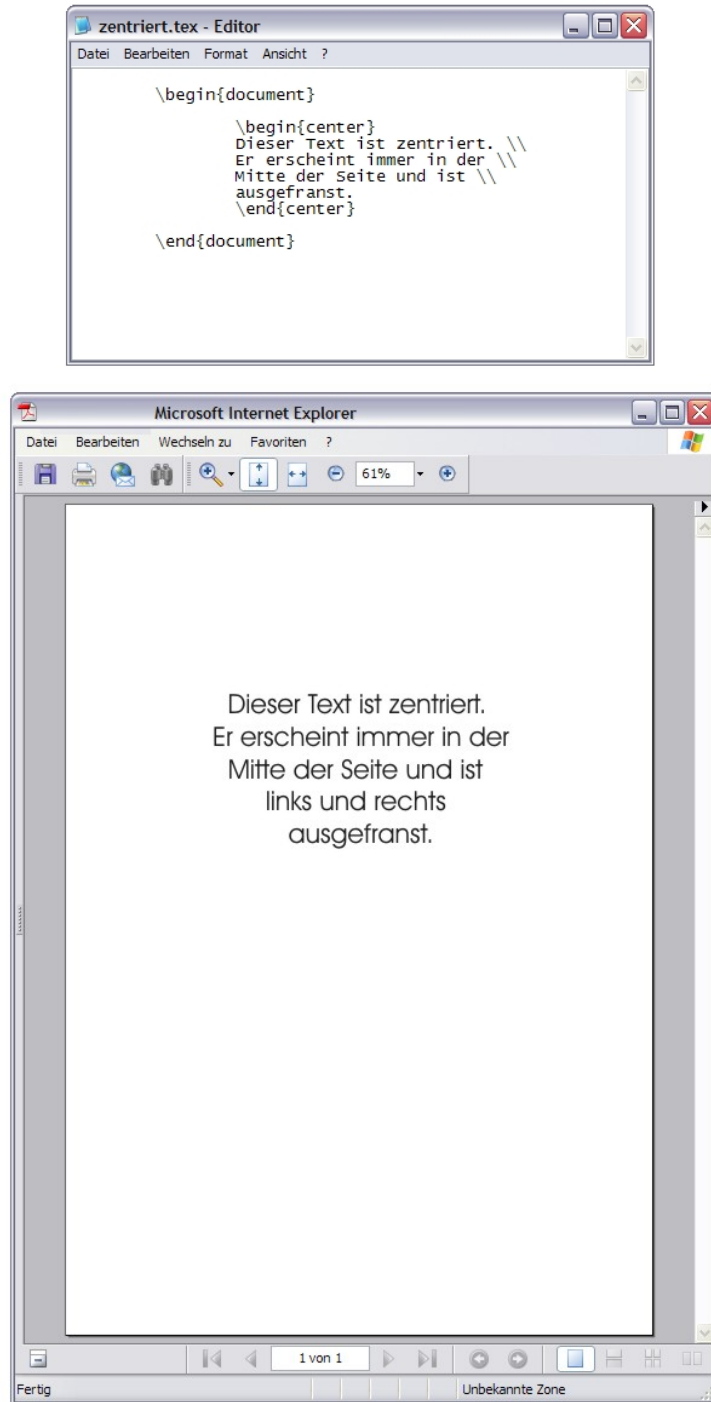


Abbildung 26: Textfluss ist zentriert

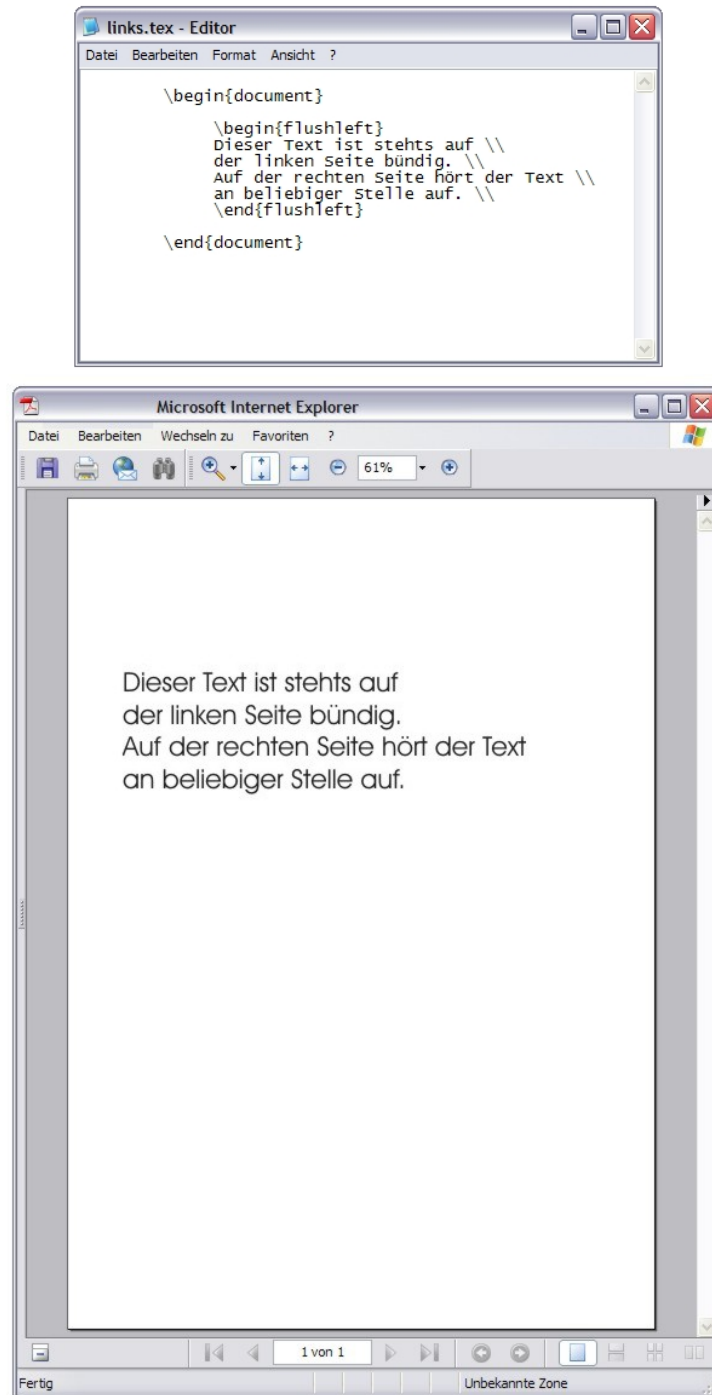


Abbildung 27: Textfluss ist linksbündig

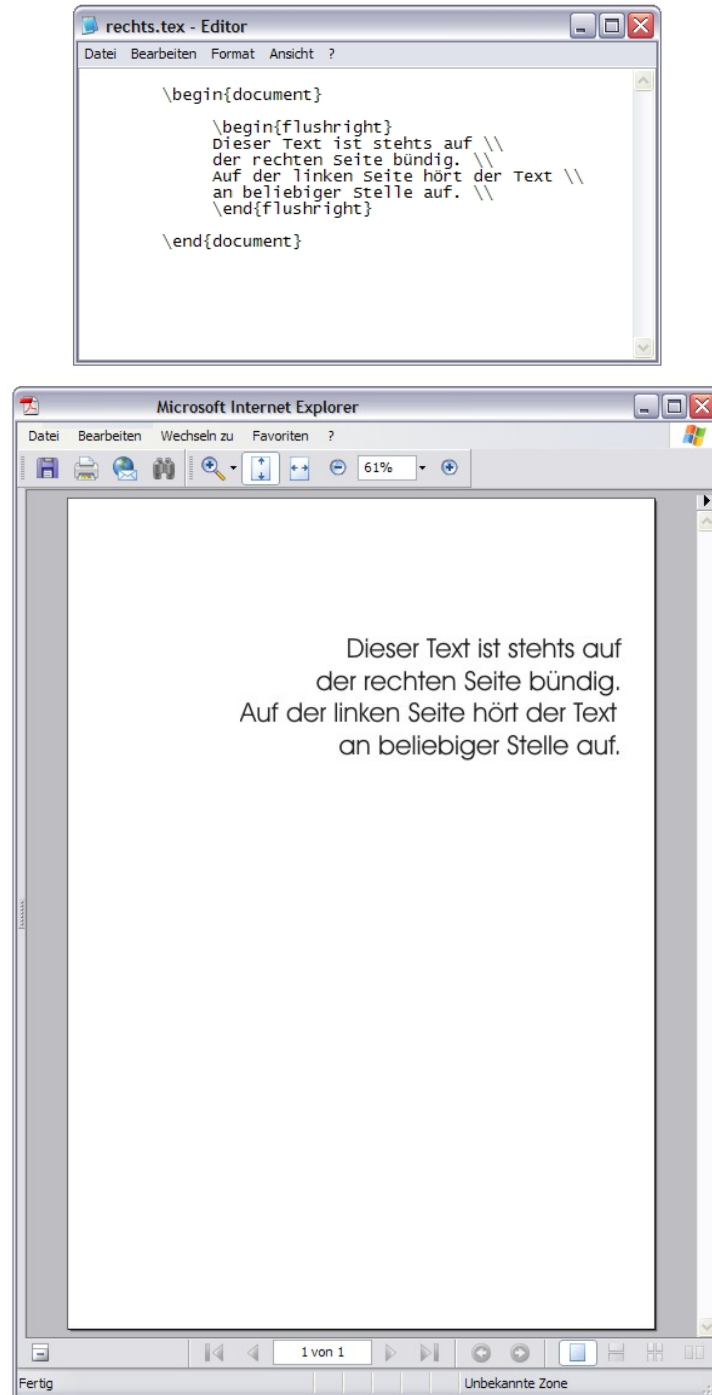


Abbildung 28: Textfluss ist rechtsbündig

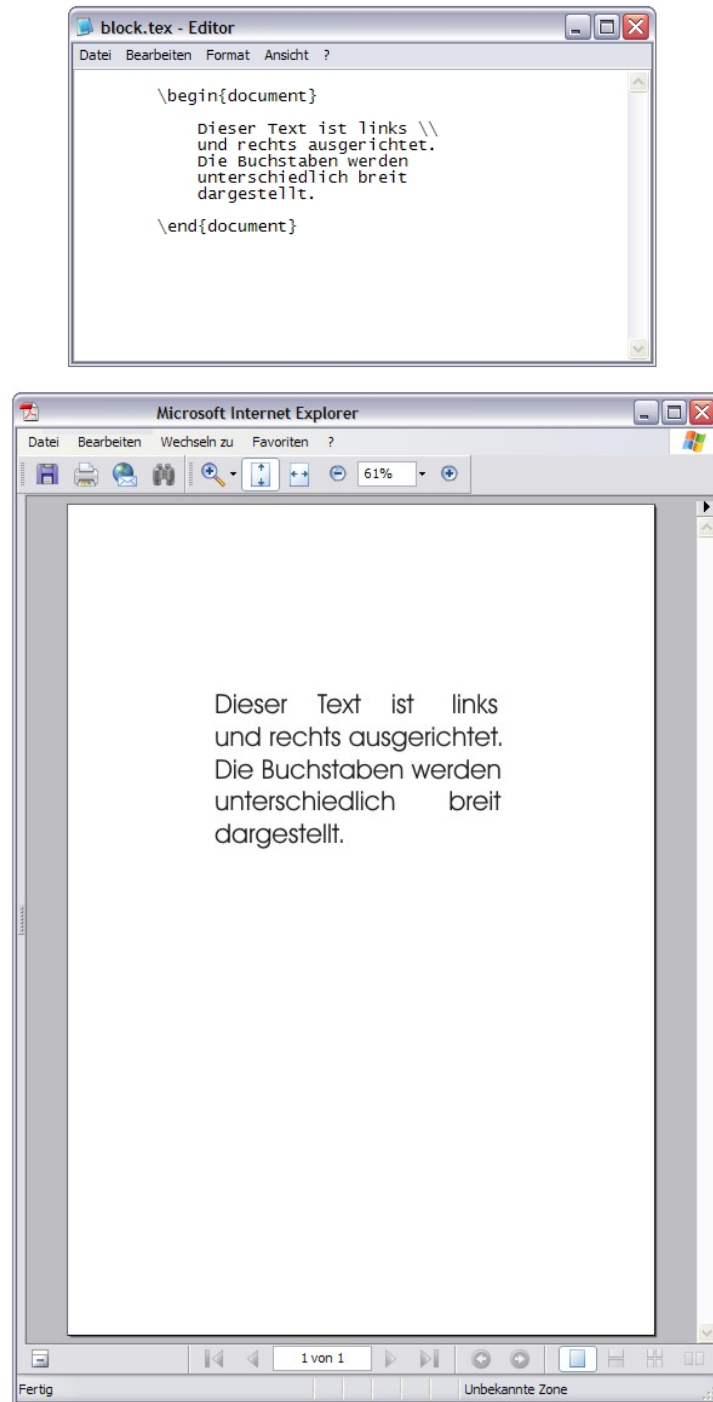


Abbildung 29: Ohne Angabe wird der Text in der Blockschrift geschrieben

Aufgabe 15 Programmiere das Grundgerüst für folgende Vorgaben:

- **rechtsbündig:** Dieser Text steht nun ganz rechts.
- **linksbündig:** Ganz links sollte dieser Text erscheinen.
- **zentriert:** Und in der Mitte wird dieser Text erscheinen.

Es genügt, nur die Grundgerüste für rechts- linksbündig und zentriert zu notieren.

5.2 Schriftformatierung

Auch die Schrift kann auf die individuellen Bedürfnisse abgestimmt werden. Es existieren folgende Schrift-Optionen, wobei der **Text** frei gewählt werden kann und für **Option** folgende Möglichkeiten zu Verfügung stehen:

{Option Text}

\bf: Schreibt den Text in fetter Schrift

\it: Der Text erscheint in kursiver Schrift

\tiny: Der Text erscheint in winzig kleiner Schrift

\Large: Der Text wird mit grosser Schrift geschrieben

\Huge: Der Text wird in gigantischer Schrift wiedergegeben

\sc: ALLE BUCHSTABEN IM TEXT WERDEN GROSS GESCHRIEBEN. DER ANFANGSBUCHSTABE IST SOGAR EINE SPUR GRÖßER ALS DIE FOLGENDEN

Beispiel für einen kursiv geschriebenen Text:

{\it Dieser Text innerhalb der Klammer ist kursiv geschrieben} und alles ausserhalb wird wieder in normaler Schrift wiedergegeben.

Erscheint in einer PDF-Datei wie folgt:

Dieser Text innerhalb der Klammer ist kursiv geschrieben und alles ausserhalb wird wieder in normaler Schrift wiedergegeben.

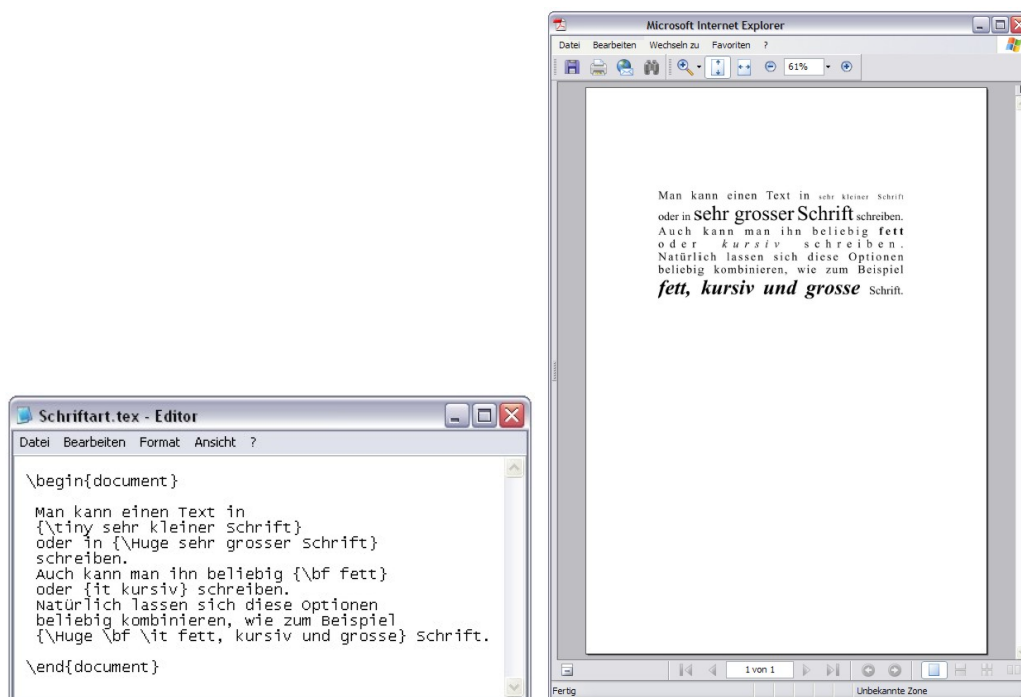


Abbildung 30: Mögliche Schriftarten

Aufgabe 16 Erstelle den Code für den untenstehenden Text, der als zentrierter, kursiver und fetter Text erscheinen soll.

Der Erdmond, ist der einzige natürliche Begleiter der Erde. Er ist der einzige Himmelskörper außerhalb der Erde, der jemals von Menschen betreten wurde.

Aufgabe 17 Erstelle eine $\text{\LaTeX}$ -Datei unter Berücksichtigung folgender Punkte:

- **fett, kursiv:** *Schweiz*
- **klein(tiny):** Deutschland
- **fett, Grossbuchstaben:** FRANKREICH

Es genügt, nur die Grundgerüste für die Schriftarten zu schreiben.

5.3 Aufzählungen

Grundsätzlich werden zwei Arten von Auszählungen unterschieden: nummerierte und nicht nummerierte. Ob man nun eine Aufzählung mit Nummern (**enumerate**) oder ohne Nummern (**itemize**) benötigt: In $\text{\LaTeX}$ bleibt das Grundgerüst dasselbe. Mit **begin** wird die Aufzählung eingeschaltet, mit **end** wird sie wieder ausgeschaltet.

Grundgerüst für die Aufzählung

```
\begin{<type>}
\item <Text>
\item <Text>
\item <Text>
\end{<type>}
```

Als **<type>** können folgende Arten verwendet werden:

- **enumerate** für eine nummerierte Aufzählung (Abbildung 31)
- **itemize** für eine nicht nummerierte Aufzählung (Abbildung 32)

Mit dem Befehl **item** fügt man eine weitere Aufzählungslinie ein. Gleich nach dem Wort **item** und einem Leerzeichen folgt der Inhalt **<Text>**. Die Aufzählungspunkte werden automatisch erstellt und etwas eingerückt, damit sie besser sichtbar werden. Bei der nummerierten Aufzählung verteilt $\text{\LaTeX}$ die Nummern automatisch der Reihe nach. Fügt man zwischen zwei Nummern ein neues Item ein, so passen sich die Nummern dementsprechend an. Als Aufzählungspunkt bei der nicht nummerierten Aufzählung verwendet $\text{\LaTeX}$ runde, schwarze Punkte.

Aufgabe 18 Programmiere den $\text{\LaTeX}$ -Code, der jeden Monat des Jahres durchnummeriert wiedergibt.

Aufgabe 19 Zähle folgende Punkte der Reihe nach auf (itemize):
Hund, Katze, Maus, Hase, Kuh, Floh, Elefant, Pferd

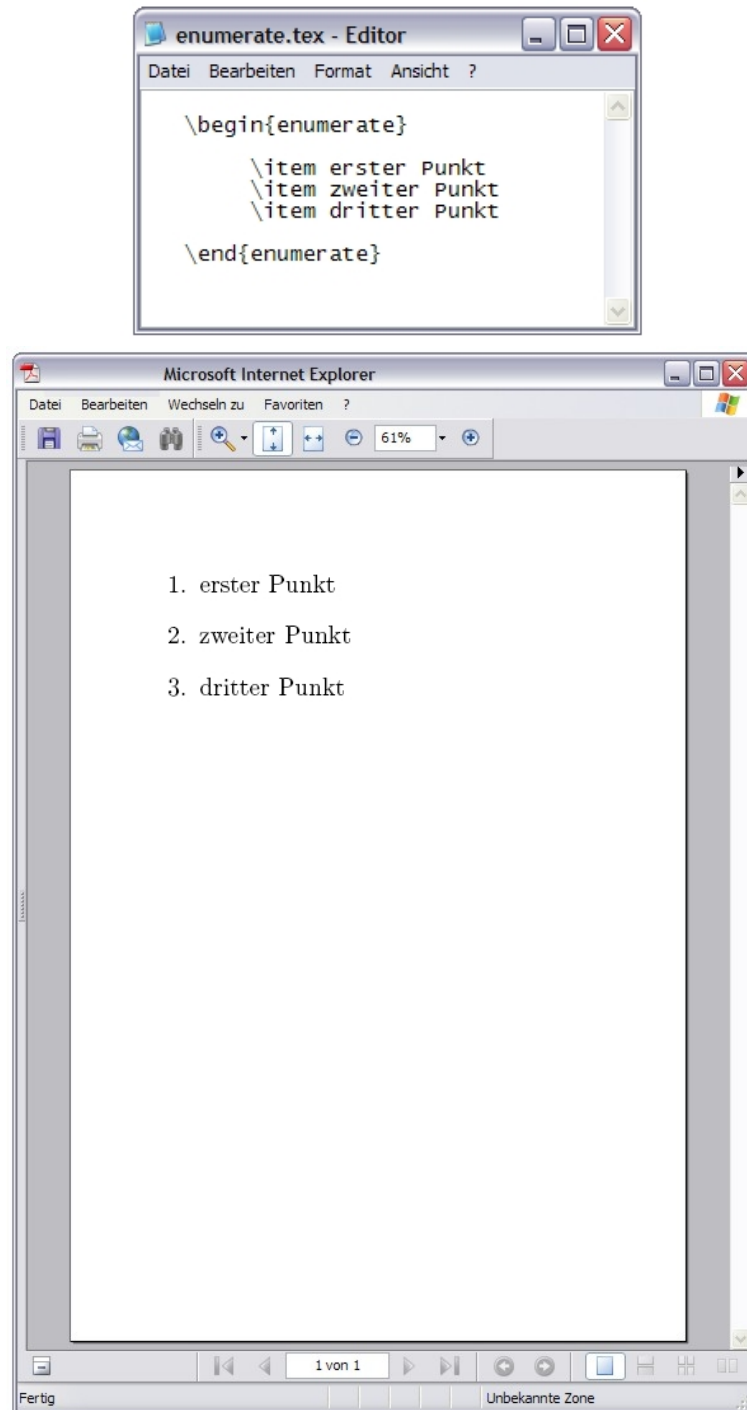


Abbildung 31: Beispiel einer durchnummerierten Aufzählung

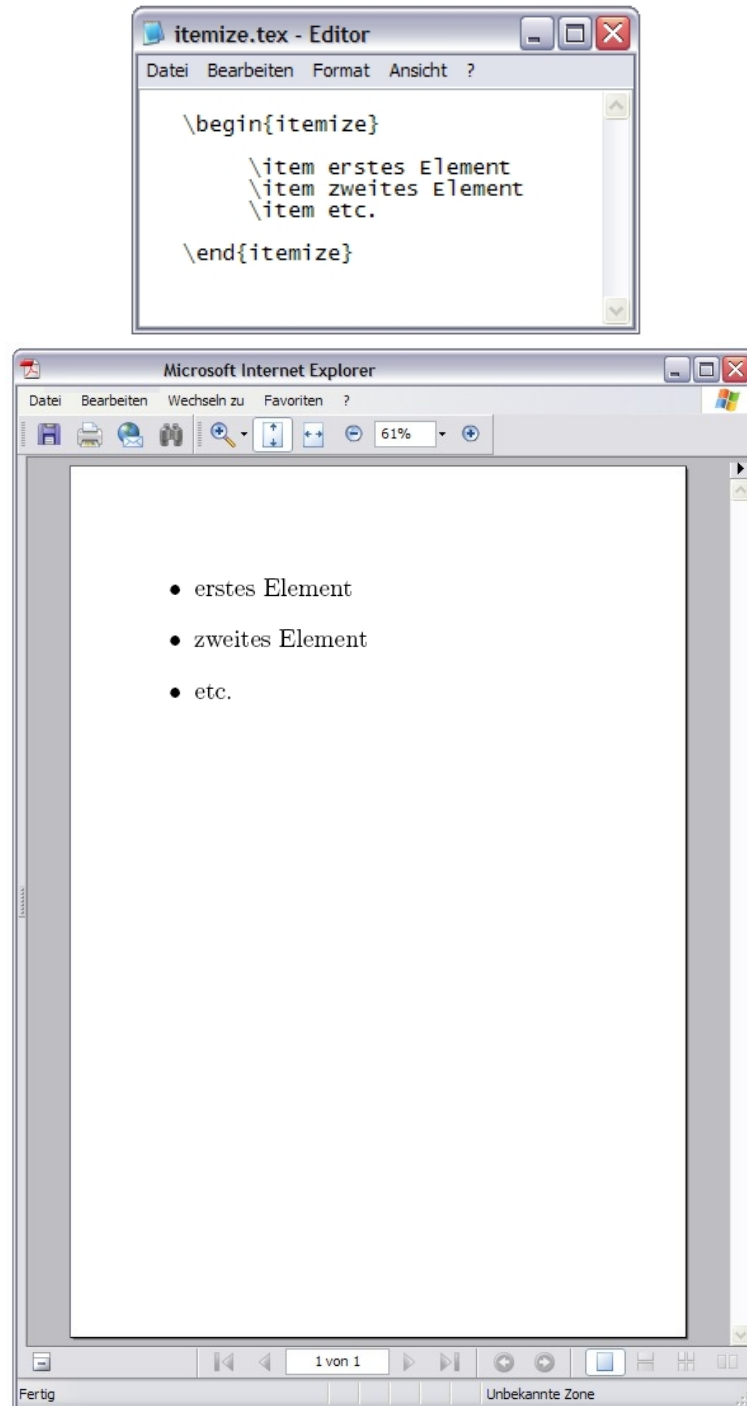


Abbildung 32: Beispiel einer Aufzählung ohne Nummern

5.4 Zusammenfassung

In dieser Lektion hast du mehrere Formatierungsmöglichkeiten kennengelernt. Du kannst nun einen Text links-, rechtsbündig oder zentriert schreiben. Auch weißt du, wie die Befehle für eine fette, kursive, kleine oder grosse Schrift lauten. Du kennst ausserdem die nummerierte Aufzählung **enumerate** und die punktweise Aufzählung **itemize**.

5.5 Kapitelprüfung

Versuche, den Code für folgende PDF-Datei in Abbildung 34 zu schreiben. Benutze das vorgegebene Grundgerüst aus Abbildung 33.

Neue Zeilen können mit dem Befehl:

```
\newline
```

erzeugt werden (siehe Kapitel:7.3 'Wichtige Einzelheiten')

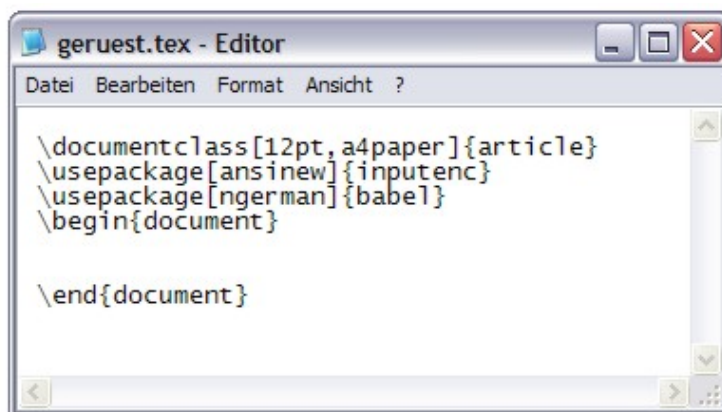


Abbildung 33: Gerüst für die Kapitelprüfung



Abbildung 34: Zu erstellende PDF-Datei

6 Bilder

Bilder lassen eine Arbeit lebendiger erscheinen und enthalten oft wichtige Informationen. Bilder können auch der Orientierung dienen und zu kurzen Lesepausen anregen. Es wird dir nach dieser Lektion leicht fallen, ein neues Bild in deinen Text einzubauen.

6.1 Bildertypen

Das Bildformat ist je nach verwendetem $\text{\LaTeX}$ -Kompilierer anders definiert. Das hier verwendete MikTeX erlaubt unter anderem die Bildertypen JPEG, GIF, PNG, etc.

6.2 Bilder einfügen

Das Bild muss im gleichen Ordner liegen wie die entsprechende **.tex-Datei**.

```
\begin{figure}[where]
\includegraphics[<width>]{<name>.<format>}
\caption{<description>}
\end{figure}
```

Mit diesem Grundgerüst lässt sich ein Bild mit dem Namen **name** und dem Format **format** linksbündig einfügen. (Bsp.: lemmon.jpg).

where definiert die Position des Bildes:

- **t** für top (oben)
- **b** für bottom (unten)
- **p** für eine neue page (Seite)
- **h** für genau hier, wo man den Code hinschreibt

Mit dem Befehl **width** kann die Bildbreite festgelegt werden (Bsp.: **width= 3 cm**). Zu jedem Bild lässt sich eine Bildlegende hinzufügen, die so genannte **description**. Sie erscheint unmittelbar unter dem Bild (Siehe caption **Eingefügtes Bild** in Abbildung 35).

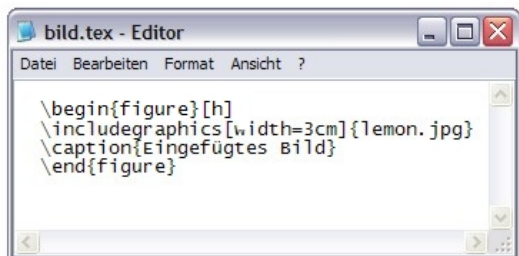


Abbildung 35: Eingefügte Bild

Aufgabe 20 Wie sieht das Grundgerüst für ein Bild mit dem Namen **picture.jpg** aus? Das Bild soll auf einer neuen Seite erstellt werden.

Aufgabe 21 Muss die Breite (width) immer angegeben werden? Was geschieht, wenn der Wert fehlt?

6.3 Bildposition

Standardmässig werden die Bilder linksbündig auf einer Seite platziert. Um das Bild zentriert oder rechtsbündig darzustellen, wird derselbe $\text{\LaTeX}$ -Code benötigt, wie bei der Textformatierung in Kapitel 5.1.

```
\begin{figure}[ht]
\begin{center}
\includegraphics[width=3cm]{lemon.jpg}
\caption{Eingefügte Bild}
\end{center}
\end{figure}
```

Dieses Bild wird nun in der Mitte der Seite erscheinen. Anstelle von **center** kann natürlich auch **flushright** oder **flushleft** stehen.

6.4 Zusammenfassung

Du hast in diesem Kapitel gelernt, Bilder in eine $\text{\LaTeX}$ -Datei einzubauen, die Bildbreite zu bestimmen und eine Legende beizufügen. Leider sucht sich $\text{\LaTeX}$ selbst den geeignetsten Platz für die Bilder aus. Bei Unstimmigkeiten mit dem Programm kannst du mit dem Befehl:

```
\newpage
```

Platz (eine neue Seite fordern) für die Bilder schaffen. Häufig lassen sich so grobe Unschönheiten verbessern.

6.5 Kapitelprüfung

Für diese Aufgabe benötigst du ein Bild mit der Endung „.jpg“. Schreibe den Code, um dieses Bild mit einer Legende, wie zum Beispiel „Das ist mein Bild“, einzufügen. Das Bild soll im Zentrum auf einer eigenen Seite erscheinen und 10 cm breit sein. Achte darauf, dass du alle nötigen Packages zur Verfügung hast.

7 Fussnoten und Seitendarstellung

7.1 Fussnoten

An einigen Stellen in dieser Arbeit wurden Fussnoten verwendet.
Mit dem Befehl

```
\footnote{<Text>}
```

erkennt L^AT_EX allfällige Fussnoten und ersetzt diese Stelle durch eine durchnummerierte, leicht erhöhte Zahl. Dieselbe Zahl erscheint am unteren Ende der Seite mit dem dazugehörigen **Text**.

Dieser Text mit der dazugehörigen Fussnote

```
Albert Einstein\footnote{Er ist der Erfinder der Relativitätstheorie} wurde  
70 Jahre alt.
```

wird wie folgt dargestellt:

Albert Einstein⁷ wurde 70 Jahre alt.

7.2 Wichtige Einzelheiten

Hier noch drei weitere, wichtige Befehle:

```
\newline
```

oder

```
\\
```

Beide Befehle erzwingen einen Linienumbruch (wie die Taste **Enter** im Word).

```
\newpage
```

Es wird auf einer neuen Seite fortgefahren.

7.3 Zusammenfassung

Mit den soeben eingeführten Befehlen kannst du deiner Arbeit 'den letzten Schliff' geben.

⁷Er ist der Erfinder der Relativitätstheorie

7.4 Kapitelprüfung

Programmiere für folgenden Text die $\text{\LaTeX}$ -Datei:

Nun bin ich im letzten Kapitel⁸ angelangt und beherrsche Latex⁹ so gut, dass ich eigenständig eine Arbeit erstellen kann.

⁸ Kapitel 7

⁹ Ist ein Textverarbeitungspogramm

8 L^AT_EX-Zusammenfassung

Es folgt nun eine Zusammenfassung der wichtigsten Befehle und Gerüste.

Grundgerüst für eine L^AT_EX-Datei:

```
\documentclass[<option>]{<class>}
\begin{document}
Text
\end{document}
```

class: article, report, book, letter, slides

option: 10pt, 12pt, a4paper, letterpaper

Die üblichen **Packages**:

```
\usepackage[ngerman]{babel}
```

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

```
\usepackage[usenames]{color}
```

```
\usepackage[pdftex]{graphicx}
```

Titel und Autor:

```
\title{<titel>}
```

```
\author{<name>}
```

Beide Angaben erscheinen auf dem Deckblatt erst nach diesem Befehl:

```
\maketitle
```

Inhaltsverzeichnis wird mit dieser Zeile erstellt:

```
\tableofcontents
```

Dazu gehören Kapitel und Unterkapitel:

```
\section{<titel>}<inhalt>
```

```
\subsection{<titel>}<inhalt>
```

Der Textfluss wird wie folgt angegeben:

```
\begin{<Texttype>}
Text
\end{<Texttype>}
```

Texttype: center, flushleft, flushright

Schriftart ändern:

```
\{<Option> <Text>\}
```

Option: \bf, \it, \tiny, \Large, \Huge, \sc

Aufzählung:

```
\begin{<type>}
\item <Text>
\item <Text>
\item <Text>
\end{<type>}
```

type: enumerate (nummerierte Aufzählung), itemize (punktweise Aufzählung)

Bilder einfügen:

```
\begin{figure}[ht]
\begin{center}
\includegraphics[width=3cm]{lemon.jpg}
\caption{Eingefügte Bild}
\end{center}
\end{figure}
```

Fussnote erstellen:

```
\footnote{<Text>}
```

Neue Linie:

```
\newline
```

oder

```
\\
```

Neue Seite:

```
\newpage
```

9 Prüfung

Programmiere folgendes L^AT_EX-Dokument.

Gehe der Reihe nach durch (Deckblatt, Inhaltsverzeichnis, Kapitel, ...) und teste dein Programm so oft wie möglich. Auf diese Weise vermeidest du späteres Fehlersuchen. Die Lösung deines Programms solltest du als **.tex**-Datei und als **.pdf**-Datei deiner Lehrperson abgeben.

- Arbeit als **article**
- A4-Format
- Titel: Herr Der Ringe
- Autor: <Dein Name>
- Tagesdatum
- Inhaltsverzeichnis auf einer separaten Seite
- Fünf Kapitel mit den folgenden Titeln:
 - Vom Auenland nach Bruchtal
 - Die Gemeinschaft der Gefährten
 - Entwicklung in Rohan
 - Frodo, Sam und Gollum
 - Die Entscheidungsschlacht
- Inhalt des 1. Kapitels: **Vom Auenland nach Bruchtal** (rechtsbündig, fett):

Der Roman beginnt im beschaulichen Auenland mit der Feier zum 111. Geburtstag des Hobbits Bilbo Beutlin. Dieser hat sich entschlossen, an diesem Tag das Auenland für immer zu verlassen.
- Inhalt des 2. Kapitels: **Die Gemeinschaft der Gefährten** (linksbündig):

Das Schicksal des Rings soll nun vom Rat Elronds bestimmt werden, an dem Vertreter aller freien Rassen teilnehmen. Nur die Zerstörung des Rings im Schicksalsberg im Land Mordor, dem Reich Saurons, hält Mittelerde sicher.
- Inhalt des 3. Kapitels: **Entwicklung in Rohan** (kursiv):

Die Orks verschleppen Merry und Pippin durch Rohan, um Isengard, die Residenz Sarumans, zu erreichen.

- Inhalt des 4. Kapitels: **Frodo, Sam und Gollum**: Nummerierte Aufzählung dieser Namen mit folgenden Inhalten für:
 1. Frodo: Adoptivsohn von Bilbo Beutlin. Er ist der Besitzer des Ringes.
 2. Sam: Frodos bester Freund. Zusammen erleben sie die gefährlichsten Abenteuer.
 3. Gollum: Er versucht Frodo und Sam zu töten, um ihnen den Ring wieder abzunehmen.
- Schreibe eine Fussnote zu Frodo mit dem Inhalt: Ein Hobbit
- Inhalt des 5. Kapitels: **Die Entscheidungsschlacht**(in **LARGE**-Schrift): Mit der Zerstörung des Ringes fallen auch die Armeen Saurons auseinander.
- Füge am Schluss im Zentrum dieses Bild ein:
<http://ia.imdb.com/media/imdb/01/I/25/16/28m.jpg> mit dem folgenden Text: 'Herr Der Ringe'. Das Bild soll width=3cm haben. (Lade zuerst das Bild in den Ordner, wo du deine **.tex**-Datei abspeicherst).
- Beginn Kapitel drei (Entwicklung in Rohan) auf einer neuen Seite

10 Literatur

- [1] L.Lamport: DAS $\text{\LaTeX}$ -HANDBUCH. Addison-Wesley Deutschland (1995)
- [2] Dalheimer, Matthias Kalle: $\text{\LaTeX}$, O'Reilly Verlag
- [3] <http://people.ee.ethz.ch/~oetiker/lshort/lshort.pdf>
- [4] <http://lefti.amigager.de/latex/main.html>

11 Lösungen (*nur für Lehrperson*)

Aufgabe 1

Diese Aufgabe sollte der Schüler alleine lösen können. Falls er Hilfe benötigt, sollte die Lehrperson dem Schüler das Programm **Editor** zeigen. Durch Doppelklick lässt sich ein neuer Editor öffnen. Um einen neuen Ordner auf dem Desktop zu erstellen, sollte der Schüler das 'Rezept' nochmals genau durchlesen. Allenfalls einmal vorzeigen, Ordner wieder löschen und zuschauen, wie es der Schüler macht.

Aufgabe 2

Falls ein Schüler Hilfe benötigt, wird ihm ein anderer Schüler zugewiesen. Zu zweit sollte die Aufgabe lösbar sein.

Aufgabe 3

Gleiche Antwort wie Aufgabe 2.

Aufgabe 4

Bei dieser Aufgabe werden manche Schüler mit Erstaunen zusehen, wie in der Konsole seitenweise Anweisungen des Kompilierens durchlaufen. Hier eventuell die einzelnen Schritte zusammen anschauen und den Vorgang des Kompilierers erläutern. Nicht zu tief in die Materie eindringen. Der Schüler soll danach in erster Linie wissen, dass der Computer ein PDF erstellt.

Aufgabe 5

```
\documentclass[a4paper,12pt]{book}
\begin{document}
Text
\end{document}
```

Aufgabe 6

```
\documentclass{slides}
\begin{document}
Text
\end{document}
```

Die Aufgabe wird auch als richtig angesehen, wenn andere Optionen angegeben wurden.

Aufgabe 7

Nein, der Header und Body müssen genau in dieser Reihenfolge eingegeben werden. $\text{\LaTeX}$ sucht beim Kompilieren zuerst den Befehl für den Header, führt diese Befehle aus und sucht anschliessend den Befehl für den Body.

Aufgabe 8

```
\documentclass[10pt]{book}
\usepackage[pdftex]{graphicx}
\begin{document}
Text
\end{document}
```

Aufgabe 9

```
\documentclass{article}
\usepackage[frenchb]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
Text
\end{document}
```

Aufgabe 10

Nur im Body! Der Header beinhaltet nur Zeilen, welche das Aussehen der Dokumentation beschreiben. Zusätzlich noch die **Packages**. Das Datum wird im Body vor den Befehl `\maketitle` gesetzt.

Aufgabe 11

```
\documentclass[12pt]{article}
\usepackage[ngerman]{babel}
\usepackages[pdftex]{graphicx}
\begin{document}
\title{Paradiso}
\author{Lesly Porton}
\maketitle
\enddocument
```

Aufgabe 12

```
\documentclass[12pt,a4paper]{article}
\begin{document}
\title{Der blaue Ball}
\author{Ferdinand Ballonso}
\date{Dezember 5, 2010}
\maketitle
\end{document}
```

Aufgabe 13

```
\documentclass[12pt,a4paper]{article}
\begin{document}
\tableofcontents
\section{Montag}
\subsection{Fussballtraining}
\subsection{Handarbeit}
\section{Dienstag}
\subsection{Klavier}
\section{Mittwoch}
\subsection{Mittagessen bei Müller}
\subsection{Abendessen bei Kuster}
\section{Donnerstag}
\subsection{Handballtraining}
\subsection{Minigolf}
\section{Freitag}
\subsection{Frei}
\section{Samstag}
\subsection{Zelten in Genf}
\section{Sonntag}
\subsection{Frühstück}
\subsection{Fernsehen}
\end{document}
```

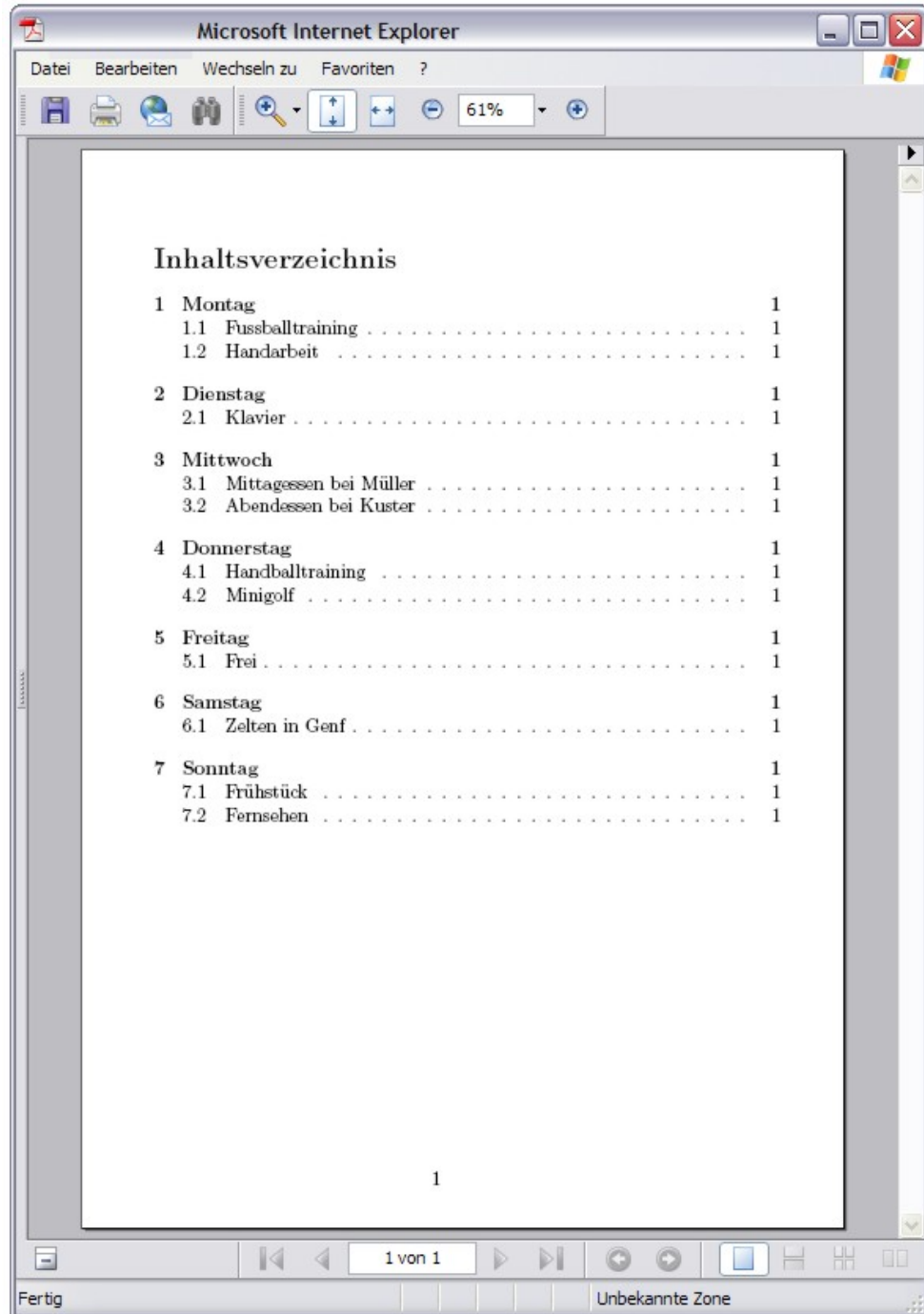


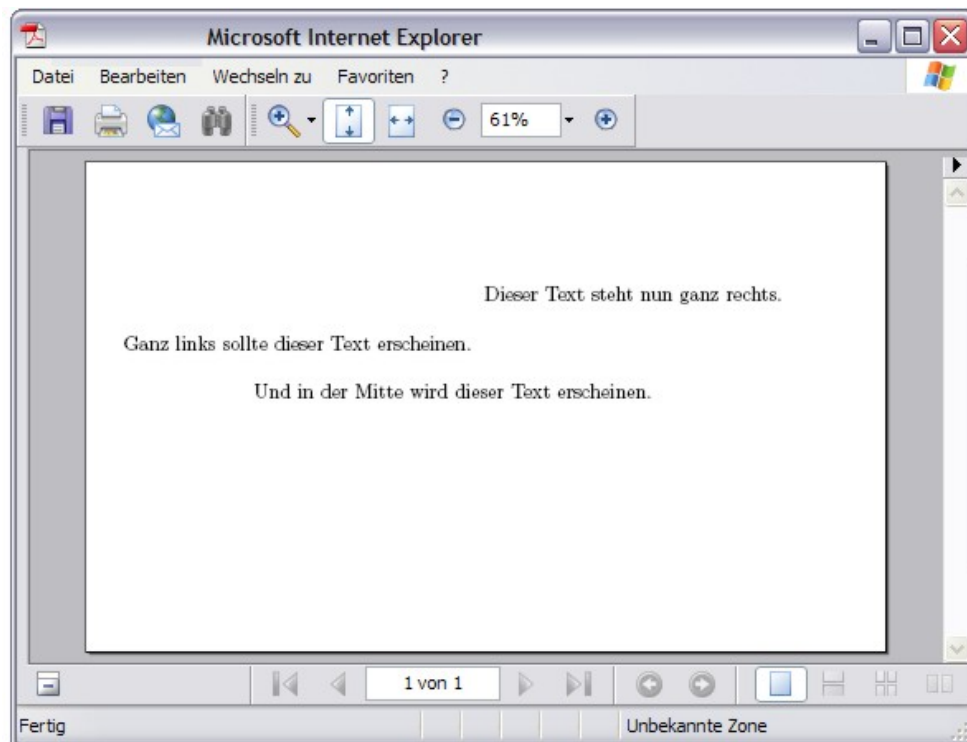
Abbildung 36: Lösung zu Aufgabe 13

Aufgabe 14

Durch ausprobieren (Befehl weglassen) bemerkt man, dass $\text{\LaTeX}$ das Inhaltsverzeichnis nicht ausgibt. Die Kapitel mit den Unterkapiteln werden aber wie üblich ausgegeben.

Aufgabe 15

```
\begin{flushright}
Dieser Text steht nun ganz rechts.
\end{flushright}
\begin{flushleft}
Ganz links sollte dieser Text erscheinen.
\end{flushleft}
\begin{center}
Und in der Mitte wird dieser Text erscheinen.
\end{center}
```



Aufgabe 16

```
\begin{center}  
{\it \bf Der Erdmond, ist der einzige natürliche Begleiter der Erde.\\  
Er ist der einzige Himmelskörper außerhalb der Erde, der jemals von Men-  
schen betreten wurde.}  
\end{center}
```

Aufgabe 17

```
{\bf \it Schweiz }  
{\tiny Deutschland }  
{\bf \sc Frankreich }
```

Aufgabe 18

```
\begin{enumerate}  
\item Januar  
\item Februar  
\item März  
\item April  
\item Mai  
\item Juni  
\item Juli  
\item August  
\item September  
\item Oktober  
\item November  
\item Dezember  
\end{enumerate}
```

Aufgabe 19

```
\begin{itemize}  
\item Hund  
\item Katze  
\item Maus  
\item Hase  
\item Kuh  
\item Floh  
\item Elefant  
\item Pferd \end{itemize}
```

Aufgabe 20

```
\begin{figure}[p]  
\includegraphics{picture.jpg}  
\end{figure}
```

Aufgabe 21

Nein, die Breite muss nicht jedesmal angegeben werden. Ohne die Breitenangabe wird das Bild genau so breit wie es das Seitenlayout zulässt (Seitenbreite).

Kapitelprüfung 3

Das PDF des Schülers muss identisch sein mit der Abbildung 37.
Die Lehrperson kontrolliert auch den Namen der Datei (**beispiel.tex**)
und ob sich die Datei und auch die PDF-Datei im Ordner **Mein_Latex**
befindet.

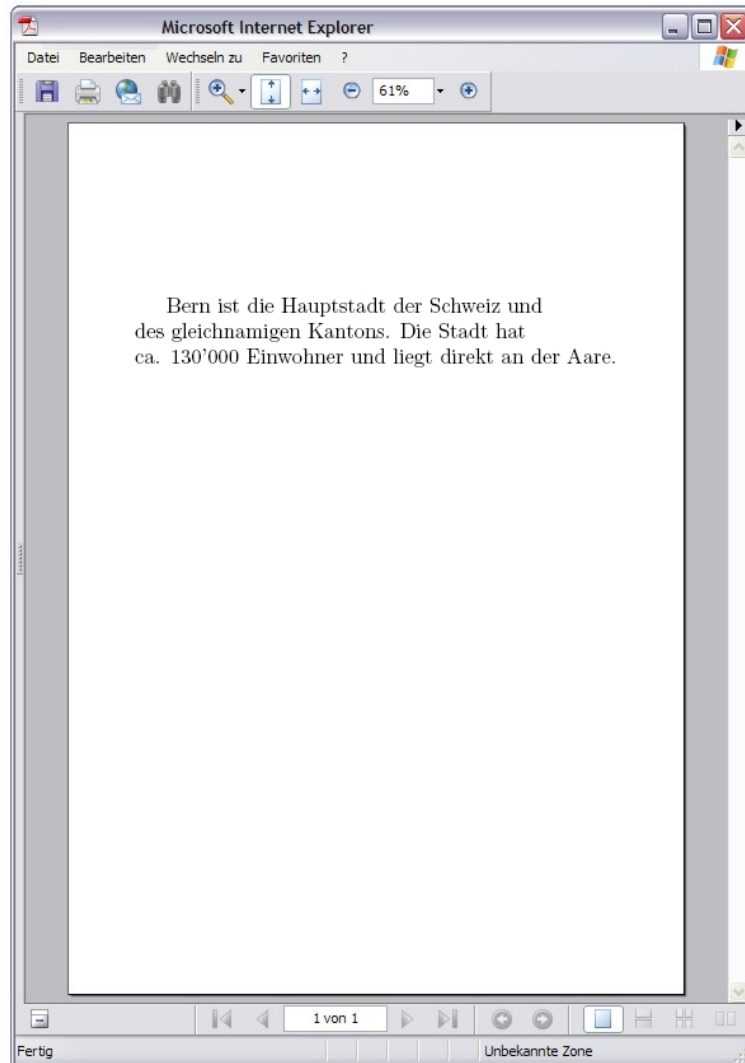


Abbildung 37: Erhaltene PDF-Datei

Kapitelprüfung 4

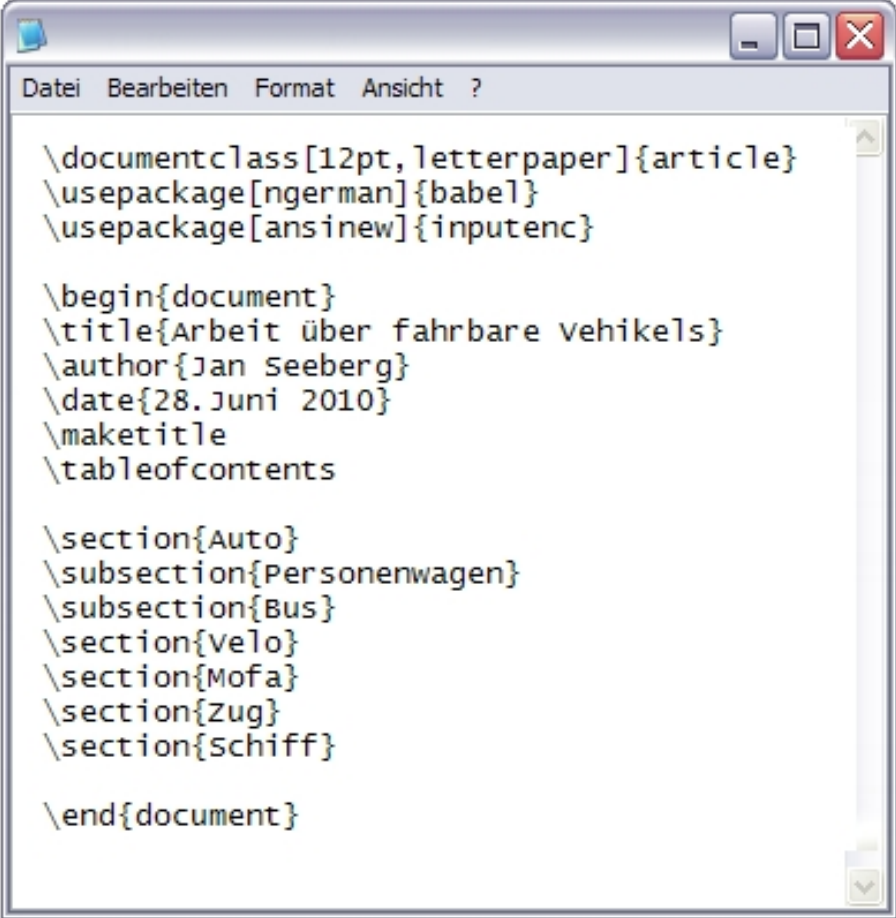
1. Aufgabe:

Der selbständig geschriebene L^AT_EX-Code ist in Abbildung 38 gezeigt. Wichtig ist, dass das Package für Bilder (`\package[graphicx]{pdftex}`) nicht eingebunden wurde. Der Schüler soll lernen, nur nötige Package einzubinden und nicht eine Serie Package als Grundausrüstung zu verwenden (in der `.tex` Datei nachkontrollieren). Den Schüler wird es allenfalls verwirren, dass der ganze Inhalt auf einer Seite erscheint. Dies geschieht, wenn nicht explizit eine neue Seite für das Inhaltsverzeichnis gefordert wurde (mit dem Befehl `\newpage`).

Abbildung 39 zeigt die erhaltene PDF-Datei.

2. Aufgabe:

Der L^AT_EX-Code für das vorgegebene Planeten-PDF wird in Abbildung 40 wiedergegeben. Der Code muss nicht zwingend in der gleichen Darstellung vorhanden sein. Einrückungen (z.B. bei Kapiteln) oder leere Zwischenzeilen im Editor haben für die PDF-Ausgabe keinen Einfluss.



```
\documentclass[12pt,letterpaper]{article}
\usepackage[ngerman]{babel}
\usepackage[ansinew]{inputenc}

\begin{document}
\title{Arbeit über fahrbare vehikels}
\author{Jan Seeberg}
\date{28. Juni 2010}
\maketitle
\tableofcontents

\section{Auto}
\subsection{Personenwagen}
\subsection{Bus}
\section{Velo}
\section{Mofa}
\section{Zug}
\section{Schiff}

\end{document}
```

Abbildung 38: $\text{\LaTeX}$ -Code für 3. Kapitelprüfung

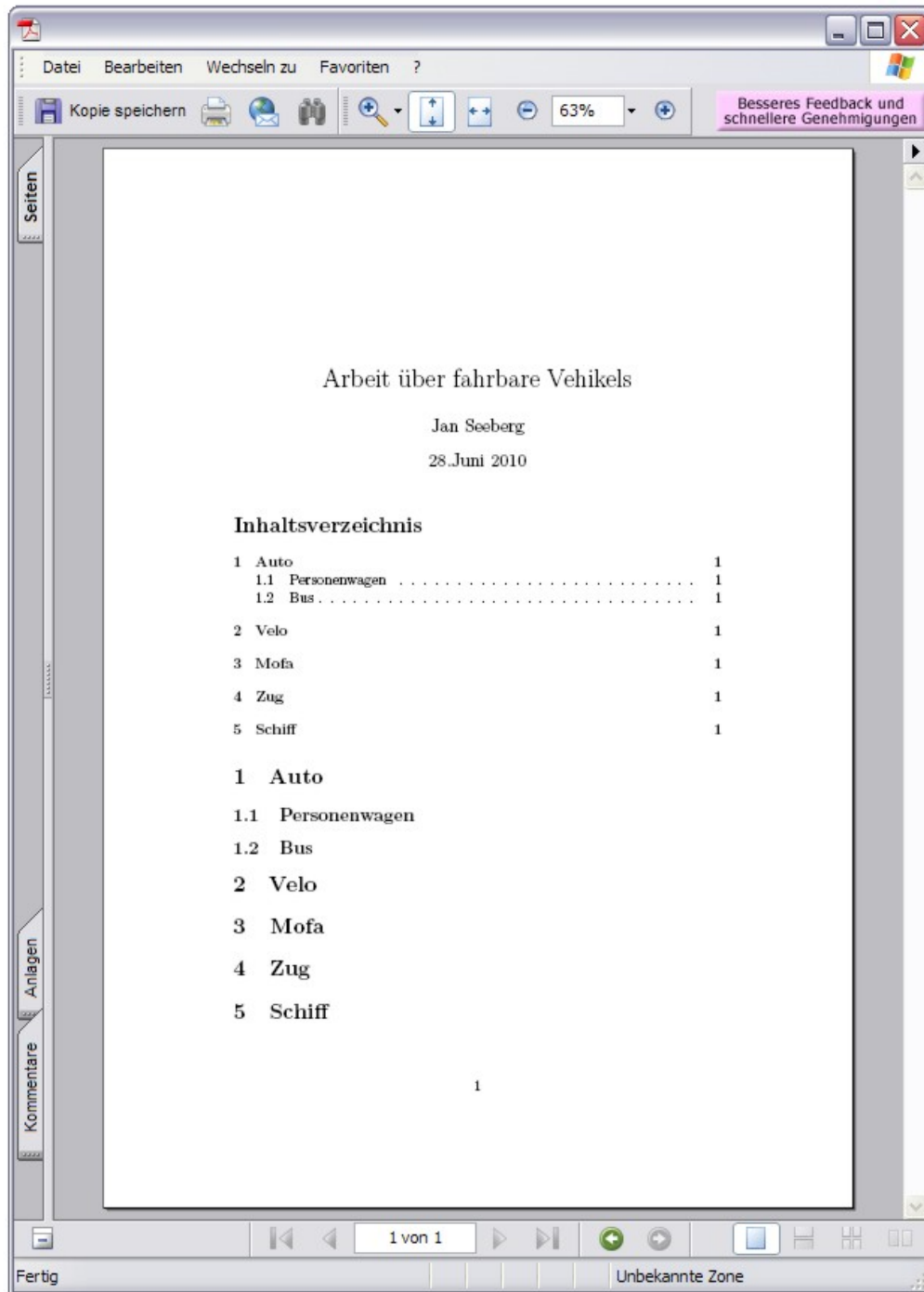


Abbildung 39: Erhaltene PDF-Datei



Abbildung 40: Lösung für Planeten-PDF

Kapitelprüfung 5

Bei dieser Aufgabe gibt es verschiedene Lösungen, wobei diese Lösung als Referenz genommen werden kann. Die Schriftarten (fett, kursiv, etc.) werden sehr unterschiedlich gehandhabt. Auch die Leerzeilen werden an unterschiedlichen Stellen eingesetzt. Aber die erstellte PDF-Datei sollte dem Abbild 34 (in Kapitel 5) ähnlich sein.

```
\documentclass[12pt,letterpaper]{article}
\documentclass[12pt,a4paper]{article}
\usepackage[ansinew]{inputenc}
\usepackage[ngerman]{babel}
\begin{document}
\begin{flushright}
Heute muss ich \newline
Einkaufen gehen, \newline
da mir viele Dinge \newline
fehlen. Der ganze \newline
Kühlschrank ist leer.
\endflushright
{\Huge Einkaufsliste Migros}
\begin{itemize}
\item {\it Äpfel}
\item {\sc Brot}
\item {\tiny Butter}
\end{itemize}
{\bf Einkaufsliste Coop}
\begin{enumerate}
\item Sirup
\item {\Large Wein}
\item {\bf Kuchen}
\end{enumerate}
\end{document}
```

Kapitelprüfung 6

Für diese Aufgabe kann der Schüler ein eigenes Bild benutzen und einen eigene Bildbeschreibung erstellen. Hier wird das Bild **sonne.jpg** benutzt.

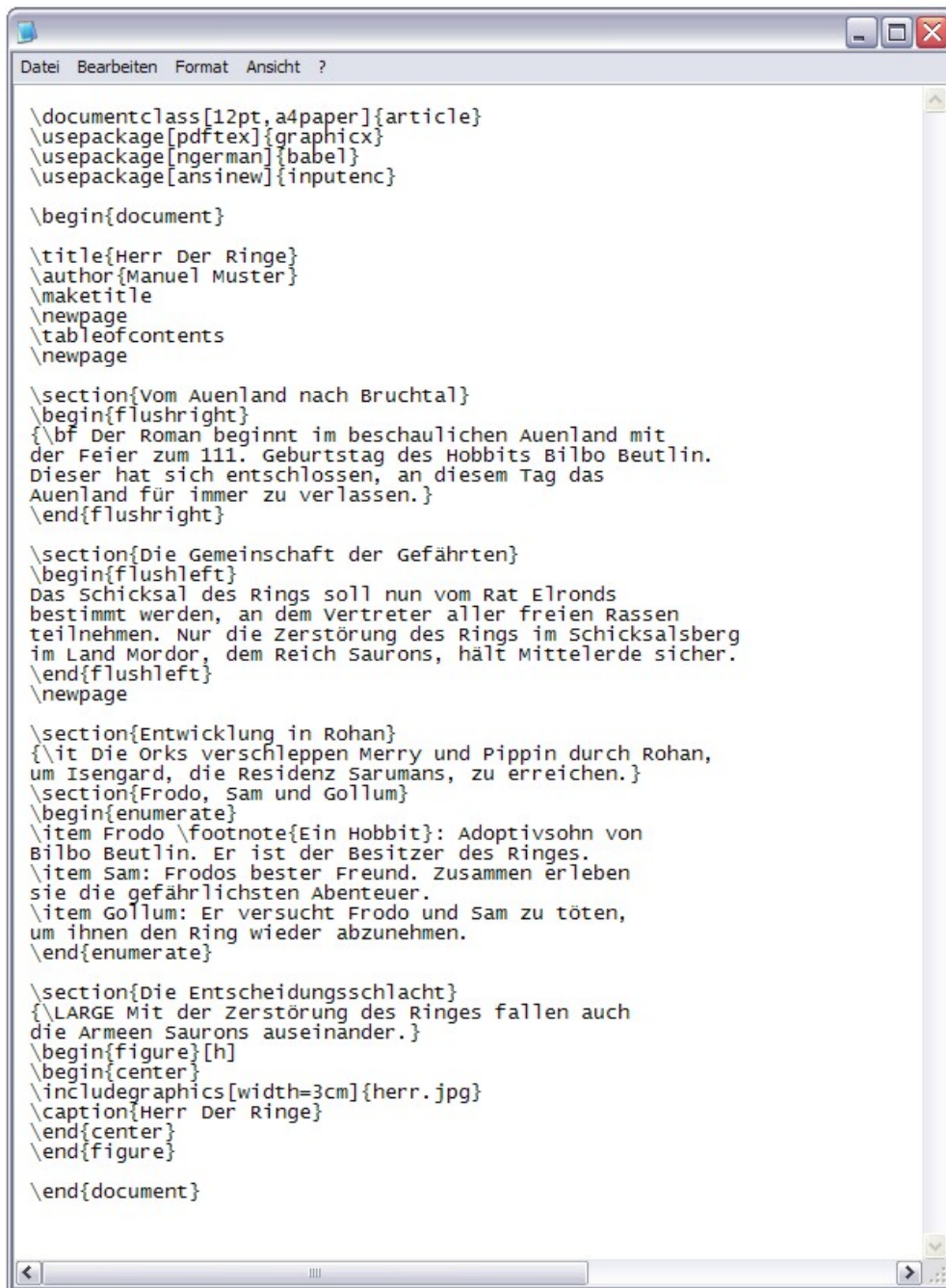
```
\begin{figure}[p]
\begin{center}
\includegraphics[width=10cm]{sonne.jpg}
\caption{Unsere Sonne ist 4.6 Mrd. Jahre alt}
\end{center}
\end{figure}
```

Kapitelprüfung 7

Nun bin ich im letzten Kapitel \footnote{Kapitel 7} angelangt und \newline beherrsche Latex \footnote{Ist ein Textverarbeitungsprogramm} so gut, \newline dass ich eigenständig eine Arbeit erstellen kann.

Schlussprüfung

Auf den folgenden Seiten wird der L^AT_EX-Programmtext und die dazugehörigen PDF-Seiten abgebildet. Auch hier gibt es wieder unterschiedliche Lösungen, da Leerschläge und Zeilenumbrüche individuell gesetzt werden können. Die Lehrperson kann bestimmen, ob diese Prüfung benotet werden soll oder nicht.

Abbildung 41: L^AT_EX-Datei

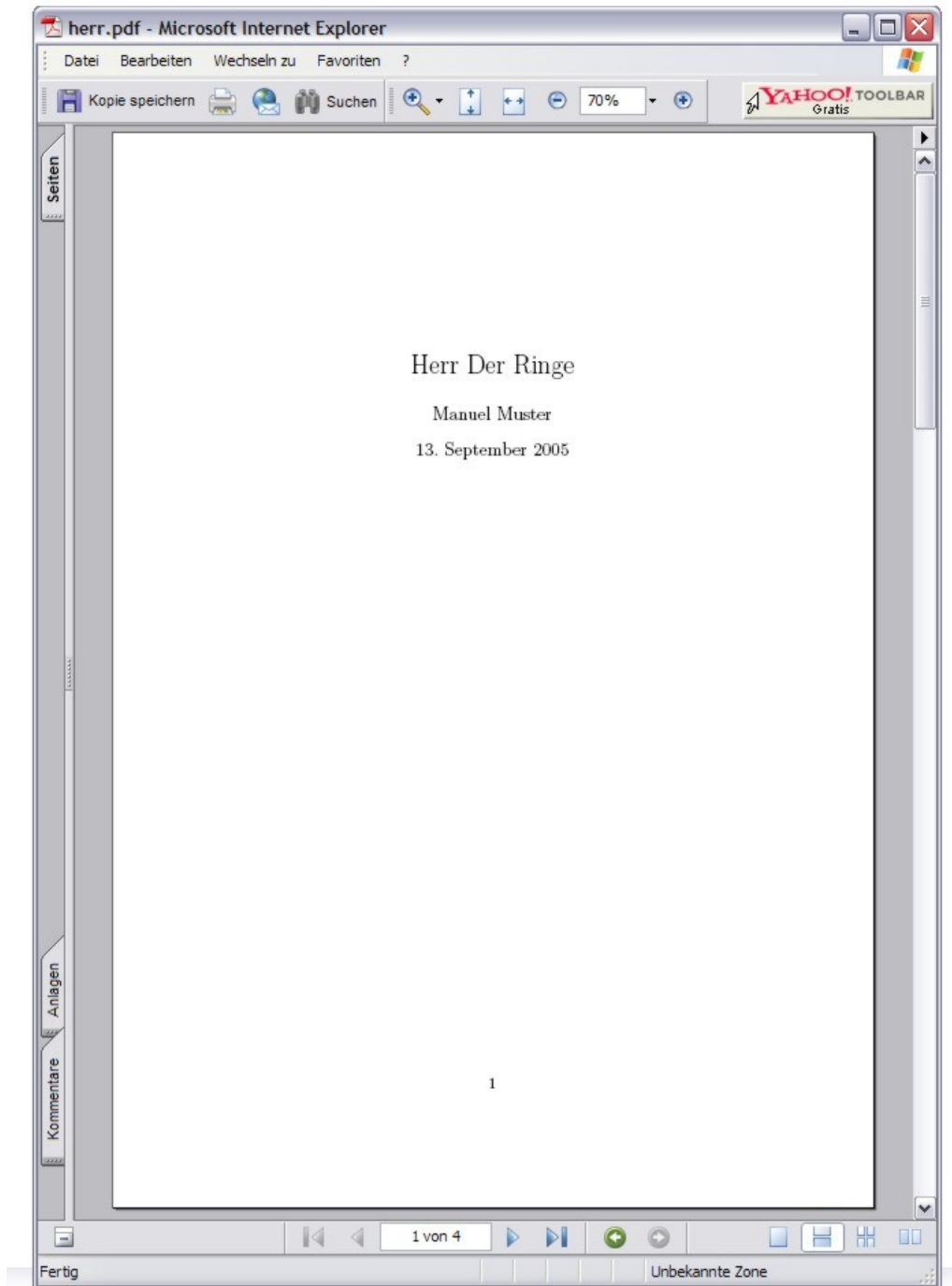


Abbildung 42: Erste Seite

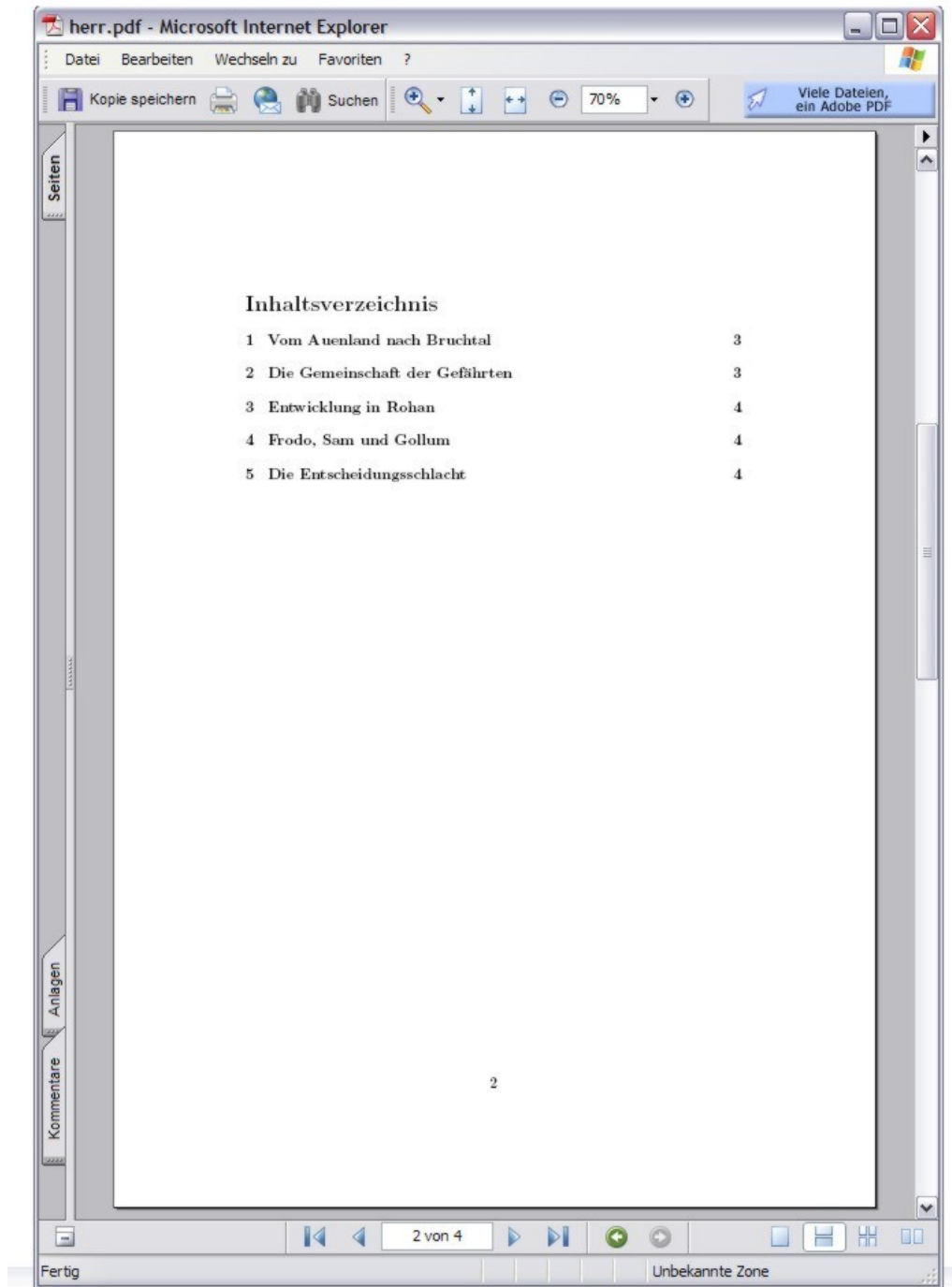


Abbildung 43: Zweite Seite

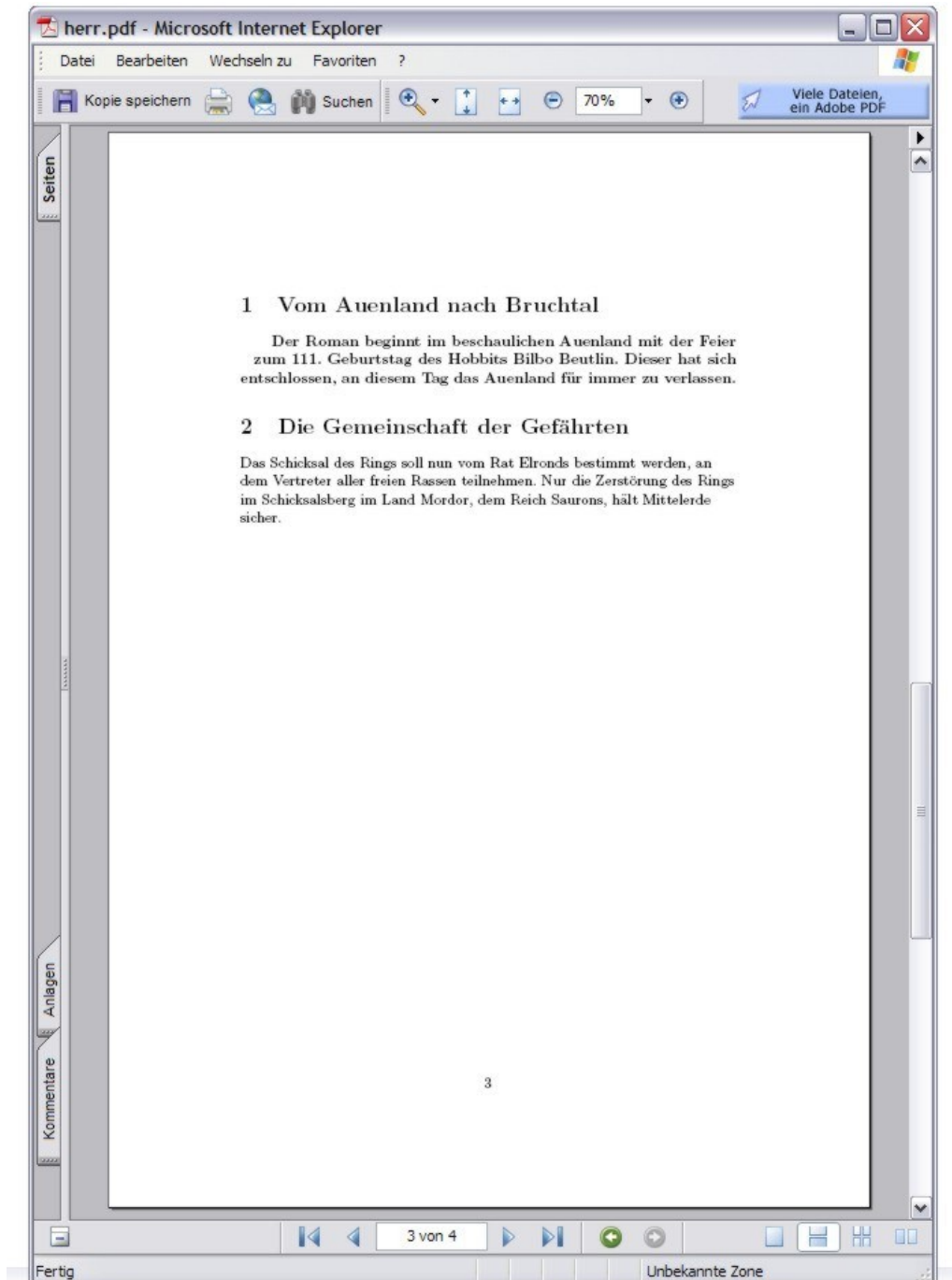


Abbildung 44: Dritte Seite

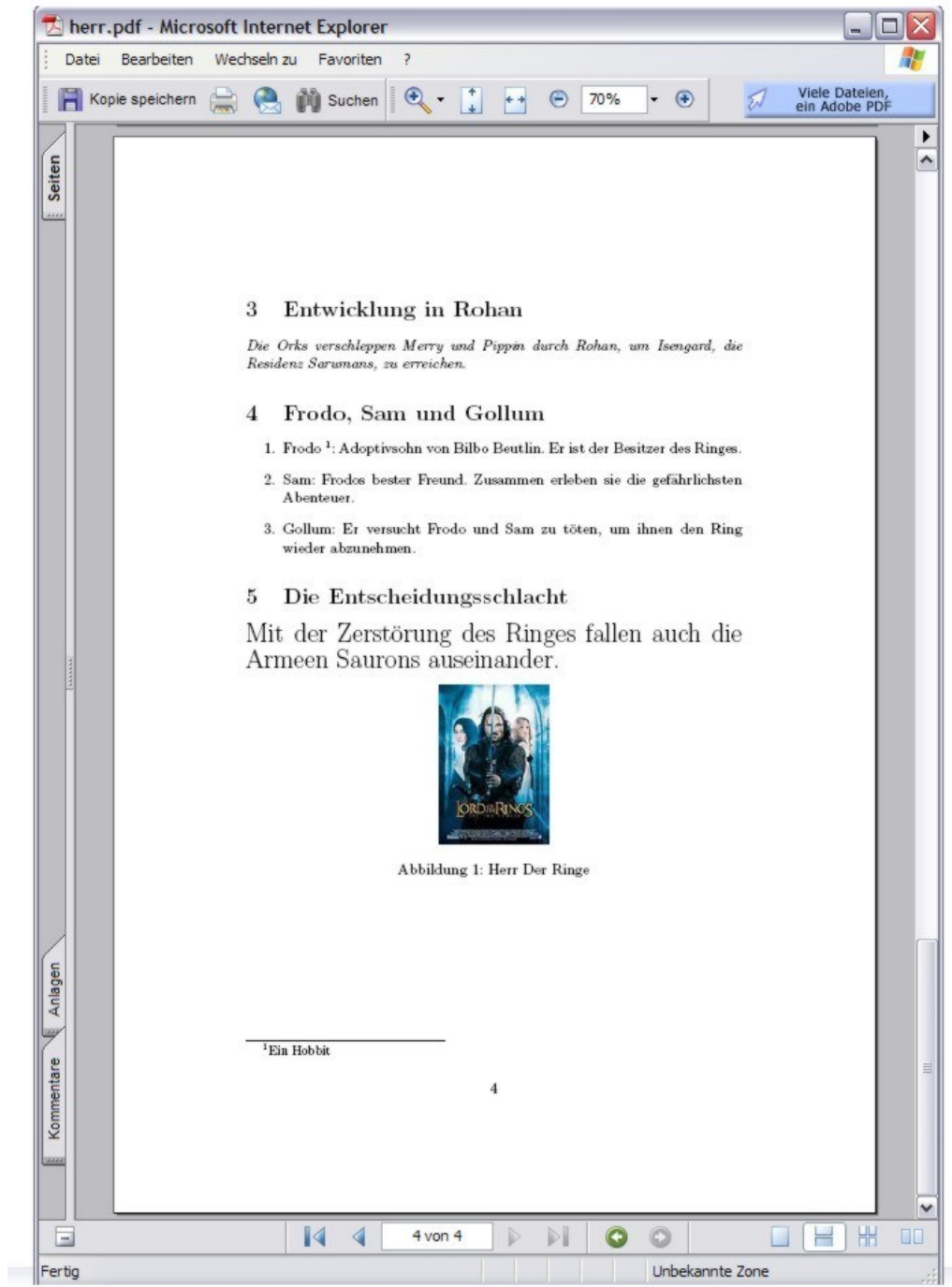


Abbildung 45: Vierte Seite