

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>1</b>
<b>1 Einleitung</b>	<b>4</b>
1.1 Aufbau der Anleitung	4
1.2 Notwendigkeit barrierefreier Publikationen	4
1.3 Voraussetzungen für barrierefreie Angebote	5
1.4 LaTeX	6
1.5 Installation	6
<b>2 Kurze Einführung in die Strukturen von LaTeX-Dokumenten</b>	<b>8</b>
2.1 Grundstruktur einer LaTeX-Datei	8
2.1.1 Standard-Dokumentenklassen	9
2.1.2 KOMA-Script-Dokumentenklassen	9
2.1.3 Optionen	9
2.2 Erweiterungspakete	10
2.3 Standardstrukturen	10
2.3.1 Normaler Text	10
2.3.2 Besondere Textumgebungen	10
2.3.3 Strukturen auf Zeilenebene	14
2.3.4 Gleitumgebungen	15
2.3.5 Verzeichnisse	18
2.4 Weiterführende Literatur	19
<b>3 Anleitung zur Gestaltung barrierefreier Dokumente</b>	<b>20</b>
3.1	20
3.1.1 Schrift	20
3.1.2 Alternative Beschreibung	21
3.1.3 Farbe	22
3.1.4 Kontrast	22
3.2 Maßnahmen zur Bedienbarkeit	22
3.2.1 Das hyperref-Paket	22
3.2.2 Seitenzahlen	24
3.2.3 Das makeidx-Paket	24
3.3 Maßnahmen zur Verständlichkeit	25
3.3.1 Das glossary-Paket	26
3.3.2 Das babel-Paket	28
3.4 Technologische Robustheit	28

3.4.1	Metadaten . . . . .	29
3.4.2	Eigene Erweiterungen . . . . .	29
3.4.3	Restriktionen . . . . .	30
<b>4</b>	<b>Die Transformationen</b>	<b>32</b>
4.1	tex4ht . . . . .	32
4.2	pdfTex . . . . .	34
<b>5</b>	<b>Überprüfung der Barrierefreiheit von PDF</b>	<b>36</b>
5.1	Überprüfen, ob das PDF-Dokument mit Tags versehen ist. . . . .	36
5.2	Überprüfen der Tag-Struktur . . . . .	36
5.3	Weitere Testmöglichkeiten . . . . .	37
5.4	Nachbearbeitung PDF . . . . .	38
	<b>Literaturverzeichnis</b>	<b>40</b>
	<b>Abbildungsverzeichnis</b>	<b>40</b>
	<b>Tabellenverzeichnis</b>	<b>41</b>

Diese Autorenanleitung entstand im Verlauf meiner Diplomarbeit zum Thema: „Accessibility-Erhöhung von  $\LaTeX$ -Dokumenten“ [Sch07]. Es soll durch die Verbesserung des Publikationsprozesses, insbesondere die Qualität der erzeugten Dokumente erhöht werden. Ziel sind möglichst barrierefreie XHTML- und PDF-Dokumente, die anschließend in Selbstverwaltung im Internet für interessierte Leser zur Verfügung gestellt werden können.

Viel Erfolg beim Schreiben Ihrer wissenschaftlichen Publikation!

# 1 Einleitung

Das vorliegende Dokument soll Ihnen helfen, ihre Arbeit im  $\LaTeX$  so aufzubereiten, dass sie den Ansprüchen einer wissenschaftlichen Publikation entspricht und den Anforderungen der Barrierefreiheit gerecht wird. Dafür wurde eine Dokumentenvorlage entwickelt, die Sie bei der Erstellung und Strukturierung ihrer Arbeit unterstützen soll. Die Autorin hat sich bemüht alle Schritte, sowie die einzelnen Dateien ausführlich und leicht verständlich auch für unerfahrene  $\LaTeX$ -Nutzer zu erklären.

Diese Anleitung richtet sich speziell an drei Zielgruppen:

- Autoren, die  $\LaTeX$ -Dokumente für die Generierung in barrierefreie PDF-Dokumente erstellen
- Autoren, die  $\LaTeX$ -Dokumente für die Transformation in barrierefreies XHTML anfertigen
- Autoren, die bestehende  $\LaTeX$ -Dokumente bzgl. der Accessibility der Endformate verbessern möchten

## 1.1 Aufbau der Anleitung

Da diese Anleitung sowohl für Einsteiger als auch routinierte  $\LaTeX$ -Nutzer angefertigt wurde, werden zunächst einige grundlegende Dinge, wie der Aufbau und die Strukturelemente von  $\LaTeX$  betrachtet. Neuerungen, die zusätzlich eingeführt oder Veränderungen in der Anwendung der Strukturen bedeuten, sind getrennt hervorgehoben, so dass sie von fortgeschrittenen Anwendern schneller aufgefunden werden können.

## 1.2 Notwendigkeit barrierefreier Publikationen

Eine wissenschaftliche Publikation ist in vielen Fällen die erste komplette Arbeit eines Studenten und ist damit der Beweis seiner Befähigung zum wissenschaftlichen Arbeiten. Eine benutzerfreundliche und barrierefreie Arbeit ist nötig, damit jeder interessierte Leser uneingeschränkten Zugriff auf die bereitgestellten Informationen hat. Dieses Dokument behandelt alle wichtigen Aspekte zur Erstellung einer barrierefreien Publikation.

Es existieren Initiativen, die sich der stetigen Verbesserung der Barrierefreiheit verschrieben haben. Hier können vertiefende Informationen dazu bezogen werden: Warum Barrierefreiheit wichtig ist, aber auch wie alternative Beschreibungen beispielsweise für Bilder zu formulieren sind.

- [Web Access Initiative \(WAI\)](#)

- Aktionsbündnis für barrierefreie Informationstechnik (ABI)
- Arbeitskreis „Barrierefreies Internet“ (AKBI)
- Barrierefrei Informieren und Kommunizieren (BIK),

Mit barrierefreien Dokumenten vergrößern Sie Ihre Leserschaft, indem Sie Menschen mit besonderen Bedürfnissen sowie Menschen, die potentiell vor technischen und anderen Barrieren stehen, einbeziehen. Barrierefreie Dokumentinhalte begünstigen neben Menschen mit Behinderungen, auch Menschen in einer vorübergehend ungünstigen Situation, solche sind eine extrem laute Umgebung, schlechte Beleuchtung oder einer Situation in der die Hände oder Augen anderweitig gebunden sind. Häufig nicht bedacht werden Nutzer mit einer anderen Muttersprache oder/und einem anderen kulturellen Hintergrund.

Durch den Einsatz assistiver Technologien können Behinderungen heute zwar zum Teil kompensiert, sowie neue Wege des Zugangs geschaffen werden, aber sie haben auch ihre Grenzen. Was sie nicht leisten können sind z. B. Grafiken in Text umwandeln, Audio in Text ausgeben oder Inhalte leicht verständlich machen. Daher obliegt es dem Autor das Dokument geeignet aufzubereiten und die nötigen zusätzlichen Informationen zur Verfügung zu stellen. Sie können auf diese Weise den Regierungsverordnungen zu Barrierefreiheit wie der „Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz“ gerecht werden. Öffentliche Einrichtungen haben die Pflicht diese umzusetzen.

### 1.3 Voraussetzungen für barrierefreie Angebote

Bei der Erstellung barrierefreier Dokumente, muss der Autor gewisse Kriterien beachten. Damit ihm dieses möglich ist, sind seine Qualifikation und Bereitschaft wichtige Voraussetzungen. Da Sie dieses Dokument bereits lesen, wird von der Bereitschaft ausgegangen und die nötige Qualifizierung soll durch die nachfolgende Anleitung geschaffen werden.

Die Web Content Accessibility Guideline (WCAG) nennt vier Prinzipien zur Steigerung und Sicherung der Barrierefreiheit wissenschaftliche Dokumente.

**Wahrnehmbarkeit** setzt voraus, dass alle Inhalte und Funktionen so dargestellt werden, dass sie von allen Benutzern bemerkt werden können.

**Bedienbarkeit** bedeutet, dass sich der Leser im Dokument gut orientieren und interessante Bereiche direkt anspringen kann.

**Verständlichkeit** setzt voraus, dass die Inhalte eines Informationsangebots klar und nachvollziehbar ausgedrückt und dargestellt sind.

**Robustheit** der Technik steht für den Einsatz von Technologien, die es ermöglichen, Informationen möglichst unbegrenzt in heutigen und zukünftigen Ein- und Ausgabemedien wiederzugeben.

Wahrnehmbarkeit, Verständlichkeit und Navigierbarkeit von Dokumenten können sehr gut durch den Autor bei der Erstellung beeinflusst werden. Die Robustheit ist hingegen eher auf die Technik bezogen. Was der Autor dabei beachten muss, ist z. B. die Auswahl der richtigen Pakete und eine geeignete Strukturierung der  $\LaTeX$ -Dokumente.

## 1.4 LaTeX

$\LaTeX$  ist ein Satzsystem, das für viele Arten von Schriftstücken verwendet werden kann, von einfachen Briefen bis zu kompletten Büchern. Besonders geeignet ist es aufgrund seiner Fähigkeiten im Formelsatz für wissenschaftliche oder technische Dokumente.  $\LaTeX$  ist für praktisch alle verbreiteten Betriebssysteme verfügbar.

## 1.5 Installation

Die  $\LaTeX$ -Distribution ist der Kern des Systems. Sie enthält die wichtigsten Programme zur Erzeugung von DVI, PostScript und PDF sowie zahlreiche Zusatzprogramme (z. B. für Musiksatz, Standardlayouts für verschiedene Organisationen und ähnliche).

Unter Linux bzw. Unix können die Pakete über den Paketmanager der Distribution installiert werden. Als Textsatzsystem  $\LaTeX$  ist bei den meisten Linux-Distributionen TeTeX dabei. Der wohl beste  $\LaTeX$ -Editor unter Linux ist Kile für KDE. Es sollte ein PDF-Betrachter (z. B. KPDF oder Acroread) sowie GhostScript installiert sein. Zu beachten ist, dass alle PDF-Viewer (außer die von Adobe) die neueren Features der PDF-Spezifikation noch nicht unterstützen.

Die folgende Installationsanleitung ist für Windows, da unter GNU/Linux in der Regel schon alles installiert ist. Für Windows empfiehlt sich  $\text{mikTeX}$ . Es ist einfach zu installieren, funktioniert zuverlässig und wird gut gepflegt.

Neben der eigentlichen  $\LaTeX$ -Distribution benötigt man andere Komponenten. Im Prinzip kann jeder Editor genutzt werden, um die  $\LaTeX$ -Ausgangsdatei zu erzeugen. Eine komfortable Unterstützung bei der Erstellung des Quellcodes bietet TeXnicCenter.

Da die zu installierenden Programme groß sind, ist DSL für eine Installation zu empfehlen. Selbst dann kann die Installation noch wenige Stunden dauern. Die Reihenfolge der einzelnen Schritte ist aufgrund der Programmabhängigkeiten einzuhalten.

- Installiere als  $\LaTeX$ -Distribution  $\text{mikTeX}$ .
  - Lade von [miktex.org](http://miktex.org) die Installationsdatei der aktuellen Version herunter. Hinter dem  $\text{mikTeX}$  Net Installer verbirgt sich das Installationsprogramm. Speichere dieses (derzeit setup-2.5.2545.exe) in einem Verzeichnis auf der Festplatte.
  - Lade die notwendigen Dateien herunter: Starte dazu die Installationsdatei, und bestätige die Lizenzvereinbarung. Wähle als Option dann „Download  $\text{mikTeX}$ “ und anschließend als Download-Option „Complete MiKTeX“. Anschließend sollte ein Server aus Deutschland zum Herunterladen gewählt

werden und schließlich das Verzeichnis, in das die Dateien gespeichert werden soll. Sollte ein Download fehlschlagen so kann es notwendig sein, das Setupprogramm erneut zu starten und einen anderen Server zu wählen.

- Installiere  $\text{miktE}\text{X}$ : Starte nun das Setup-Programm erneut und wähle die Optionen „Install  $\text{miktE}\text{X}$  “ und anschließend „Complete  $\text{miktE}\text{X}$  “. In der anschließenden Option kann bei der Installation auf einem privaten Rechner die Option „Anyone who uses this computer“ gewählt werden. Danach ist das Verzeichnis zu wählen, von dem  $\text{miktE}\text{X}$  aus installiert werden soll. Dieses ist das Verzeichnis, wohin im vorherigen Schritt die Dateien heruntergeladen wurden. In den darauffolgenden Dialogfenstern können jeweils die voreingestellten Optionen übernommen werden.
- Nicht unbedingt notwendig aber später teilweise nützlich ist die Installation von GhostScript und GhostView. Dies gilt insbesondere, wenn man mit PostScript als Endformat arbeiten möchte. Lade die aktuelle Versionen von AFPL GhostScript und GSView herunter und installiere sie.
- Falls nicht bereits vorhanden, so installiere den Acrobat Reader.
- Installiere den  $\text{L}\text{A}\text{T}\text{E}\text{X}$ -Editor TeXnic-Center
  - Lade die aktuellste Version von [www.texniccenter.org](http://www.texniccenter.org) herunter.
  - Die Installation des TeXnic-Centers läuft weitgehend automatisch ab, die Standardoptionen können übernommen werden.

Für die Generierung von XHTML wird tex4ht benötigt. Es ist sowohl in MikTeX als auch in Tetex enthalten. Zur Konvertierung der Grafiken empfiehlt sich ImageMaverick und zur Betrachtung der Ergebnisse wird ein beliebiger Browser benötigt.

## 2 Kurze Einführung in die Strukturen von LaTeX-Dokumenten

Denjenigen die bisher noch nicht mit  $\LaTeX$  gearbeitet haben, soll diesen Kapitel eine erste Einführung in die verfügbaren Strukturen geben. Da im Rahmes dieses Dokumentes nicht annähernd alle Aspekte der Dokumentenerstellung mit  $\LaTeX$  erörtert werden können, wird am Ende auf ausgewählte Literaturquellen zur Wissensvertiefung verwiesen.

### 2.1 Grundstruktur einer LaTeX-Datei

Ein  $\LaTeX$ -Dokument besteht aus einem *Vorspann* (preamble) und dem eigentlichen *Textteil* (body). Die gesamte Quelldatei besteht aus ASCII-Zeichen. Einige Zeichen haben in  $\LaTeX$  eine besondere Bedeutung. Da ist zum Beispiel der Backslash „\“ zu nennen, der Befehle einleitet.

Der Vorspann muss mindestens aus der Angabe der Dokumentenklasse `\documentclass[<Optionen>]{<Klasse>}` bestehen. Dieser Eröffnungsbefehl steht immer als aller Erstes im Vorspann. Anschließend können weitere Befehle folgen, die die globale Bearbeitungsstruktur des Textes festlegen. Werden keine weiteren Befehle angegeben, wählt der  $\LaTeX$ -Formatierer bestimmte Standardwerte für die Zeilenbreite, die Ränder, die Absatzabstände, die Seitenbreite und vieles mehr. Diese sind auf amerikanische Standards zugeschnitten. Eine umfangreiche Anpassung der Standardklassen an deutsche Verhältnisse bietet das KOMA-Script (dazu mehr in [Unterabschnitt 2.1.2](#)).

Der Vorspann endet mit dem Befehl `\begin{document}`. Alles, was nun bis zum Abschluss der Datei, der durch `\end{document}` markiert ist, folgt, wird als Textteil interpretiert. Der Text enthält den eigentlichen Dokumentinhalt, vermischt mit weiteren Befehlen. Diese Befehle haben im Gegensatz zu denen im Vorspann meist nur einen lokalen Wirkungsbereich. Sie bewirken z. B. eine Behandlung des Textes als Formel, Zitat oder Liste.

Die allgemeine Syntax eines  $\LaTeX$ -Dokumentes lautet damit:

```
1 \documentclass [<Optionen>]{<Klasse>}
2   %Eventuell \usepackage{<Erweiterungspaket>}–Befehle zur
3   %Einbindung von Erweiterungsbefehlen sowie weitere global
4   %wirkende Befehle und Erklärungen.
5 \begin{document}
6   %Text, eventuell vermischt mit weiteren lokal wirkenden Befehlen
7 \end{document}
8   %Darauffolgende Anweisungen sind erlaubt, werden aber nicht
9   %interpretiert.
```



Welche Angaben für *<Optionen>* und *<Klasse>* im `\documentclass`-Befehl möglich bzw. erlaubt sind, wird in den nächsten Abschnitten dargestellt.

### 2.1.1 Standard-Dokumentenklassen

Dokumente gleicher Struktur werden in Klassen zusammengefasst, sie müssen aber nicht notwendigerweise das gleiche typographische Erscheinungsbild ausweisen. Für *<Klasse>* stehen folgende Parameter zur Verfügung, von denen genau einer gewählt werden muss: `article`, `report`, `book`, `letter`, `proc` oder `slides` sowie `ltxdoc`, `ltxguide`, `ltxnews` oder `minimal`.

Wie die Namen schon andeuten, ist `article` für die Bearbeitung von Artikeln geeignet, die sich in Abschnitte und Unterabschnitte gliedern. Für längere Berichte, die darüber hinaus in Kapitel untergliedert werden, bietet sich die Klasse `report`. Beispiel sind Seminararbeiten, Diplomarbeiten oder Belege. Zur Bearbeitung längerer Schriftstücke, die ebenfalls in Kapitel untergliedert sind steht `book` zur Verfügung. Bücher können aus mehreren Bänden bestehen und sind häufig für den doppelseitigen Druck aufbereitet. Die anderen Klassen sollen im Rahmen dieser Anleitung nicht näher beleuchtet werden, interessierte Leser finden Informationen in den zugehörigen Anleitungen sowie [Kop00a, S. 25].

### 2.1.2 KOMA-Script-Dokumentenklassen

Das KOMA-Script passt die Standarddokumentklassen an deutsche Gegebenheiten an, d. h. zu jeder Standardklasse existiert eine deutsche Entsprechung (vgl. Tabelle 2.1).

Originalklasse	Koma-Klasse	Einsatzgebiet
<code>book</code>	<code>scrbook</code>	für komplette Bücher
<code>report</code>	<code>scrreprt</code>	für umfangreiche Dokumente
<code>article</code>	<code>scrartcl</code>	für mittelgroße und kleine Dokumente
<code>letter</code>	<code>srclettr</code>	für Briefe

Tabelle 2.1: Entsprechungen der Standard- $\LaTeX$ -Klassen

### 2.1.3 Optionen

Das Layout einer Dokumentklasse kann durch so genannte Optionen beeinflusst werden; z. B. bereitet die Zeile `\documentclass[paper=a4]{<Klasse>}` den Eingabetext für das Papierformat DinA4 auf. Weitere Optionen und deren Auswirkungen sind in [KM06] beschrieben. Da sie sich nur auf Formatierungen beziehen, ist ihre Betrachtung bzgl. der Barrierefreiheit des Ergebnisdokumentes nicht relevant.

## 2.2 Erweiterungspakete

Im Vorspann folgt nach der Angabe der Dokumentenklasse die Einbindung von Erweiterungspaketen und globalen Formatierungsanweisungen. Die Pakete können mit Hilfe der Anweisung `\usepackage [<Optionen>]{<Paket>}` eingebunden werden. Da sie, wie ihr Name schon sagt, die Funktionalität des  $\LaTeX$ -Systems erhöhen, sollen sie an dieser Stelle noch nicht betrachtet werden, der Leser sei auf [Kapitel 3](#) verwiesen. Zunächst soll auf die Standardstrukturen der  $\LaTeX$ -Sprache eingegangen werden.

## 2.3 Standardstrukturen

Zur Auszeichnung jeglicher Strukturen (Aufzählungen, mehrspaltige Absätze...) sind die dafür vorgesehenen  $\LaTeX$ -Strukturen zu verwenden. D. h. gleichzeitig, dass keine Tabellenumgebungen für die Ausrichtung von Elementen untereinander missbraucht werden sollen. Zudem sollten im eigentlichen Dokument wann immer möglich High-Level-Befehle, anstelle von plain- $\TeX$ -Befehlen verwendet werden. Zudem sollte auf Anweisungen der reinen Formatierung im Text verzichtet werden. Stattdessen sind logische Strukturen zu verwenden. Was dazu nötig ist, wird in [Unterabschnitt 3.4.2](#) erläutert.

### 2.3.1 Normaler Text

Text kann man in  $\LaTeX$  einfach eingeben, Absätze werden durch eine Leerzeile erzeugt. Darüber hinaus wird der eingegebene Text von  $\LaTeX$  automatisch in Zeilen und Seiten umgebrochen. Man kann größere Dokumente auch aufteilen, die Einbindung ausgelagerter Teile kann über `\input{<Dateiname>}` erfolgen.

### 2.3.2 Besondere Textumgebungen

Neben den reinen Textabsätzen existieren auch solche, mit besonderer Bedeutung, die sich zumeist in der Formatierung äußert. Beispiele sind Gliederungsbefehle, Zitate, Formeln oder Codeumgebungen. Die Besonderheit dieser Elemente sollte sich besser in der logischen Struktur des Dokumentes niederschlagen.

## Gliederungsebenen

Unter einer Gliederung versteht man die Einteilung eines Dokuments in Teile, Kapitel, Abschnitte und weitere Gliederungsebenen. Anzahl und Art der zur Verfügung stehenden Gliederungsbefehle richten sich nach der gewählten Dokumentenklasse. Die Standardgliederungsbefehle (vgl. [Tabelle 2.2](#)) gibt es in allen Klassen des KOMA-Scripts außer `srclettr`, darüber hinaus existiert in den Klassen für größere Dokumente noch `\chapter` und `\part` sowie `\minisec` in allen Klassen.

Ein Dokument kann mit Hilfe einfacher  $\LaTeX$ -Befehle in Abschnitte unterteilt werden. Dazu kennzeichnen Sie Überschriften, die von  $\LaTeX$  automatisch durchnummeriert und gleichzeitig in fetter und größerer Schrift vom übrigen Text abgesetzt werden.

Gliederungsbefehl	Bedeutung
<code>\section</code>	Abschnitt
<code>\subsection</code>	Unterabschnitt
<code>\subsubsection</code>	Unterunterabschnitt
<code>\paragraph</code>	Absatz
<code>\subparagraph</code>	Unterabsatz

Tabelle 2.2: Standardgliederungsbefehle

Verwenden Sie zum Formatieren von Text stets die vorgesehenen  $\LaTeX$ -Befehle zur Gliederung. Nutzen Sie diese entsprechend ihrer Rangfolge, folgt z. B. in einem Abschnitt (`\section`) ein Unterunterabschnitt (`\subsubsection`), kann dies zu Problemen (z. B. in der Nummerierung oder der erzeugten Struktur) führen.

### Erstellen von Gliederungsebenen

```

1 ... ein Absatz ...
2 \section{Überschrift}
3 ... noch ein Absatz ...
4 \subsection{Unterüberschrift}
5 ... noch ein Absatz ...
6 \section{Überschrift}
7 ... noch ein Absatz ...
8 \paragraph{Absatzüberschrift}
9 wobei der Absatz fortgesetzt
10 wird...
```

### Darstellung von Gliederungsebenen

... ein Absatz ...

## 1 Überschrift

... noch ein Absatz ...

### 1.1 Unterüberschrift

... noch ein Absatz ...

## 2 Überschrift

... noch ein Absatz ...

**Absatzüberschrift** wobei der Absatz fortgesetzt wird...

Abbildung 2.1: Beispiel für die Verwendung von Gliederungsebenen

### Zitate

Für Zitate stehen die drei Umgebungen `quote`, `quotation` und `verse` zur Verfügung, sie werden jeweils links und recht eingerückt dargestellt. Ein neuer Absatz beginnt in der `quote`-Umgebung linksbündig. Die `quotation`-Umgebung arbeitet genauso, jedoch wird die erste Zeile eines neuen Absatzes jeweils eingerückt (Erstzeileneinzug). In der `verse`-Umgebung werden die einzelnen Zeilen normalerweise durch „`\`“ explizit beendet. Sehr lange Zeilen werden in den nachfolgenden Zeilen zusätzlich eingerückt und so als zusammengehörig gekennzeichnet.

### Erstellen eines Zitates

```
1 ... Text ...  
2 \begin{quote}  
3   "Ich bin eine Zitat."  
4 \end{quote}  
5 ... Text ...
```

### Darstellung des Zitates

... Text ...  
 „Ich bin eine Zitat.“  
... Text ...

Abbildung 2.2: Beispiel für die Verwendung eines Zitates

## Codeumgebungen

Die Umgebung `verbatim` gibt den Text genauso aus, wie er eingegeben wurde, ohne dass Befehle oder Sonderzeichen<sup>1</sup> interpretiert werden. Auch wird kein Umbruch durchgeführt, außer er wurde explizit angegeben. Zusätzlich wird auf eine Typewriter-Schrift, also eine Nichtproportionalschrift umgeschaltet. Für kürzere Abschnitte<sup>2</sup> eignet sich der Befehl `\verb|<Text>|`, wobei die Zeichen, die den Text einschließen frei gewählt werden können.

### Erstellen eines Codes

```
1 ... Text ...  
2 \begin{verbatim}  
3   Quellcode  
4 \end{verbatim}  
5 ... Text ...
```

### Darstellung des Codes

... Text ...|  
 Quellcode  
... Text ...

Abbildung 2.3: Beispiel für die Verwendung eines Codes

## Formeln

Formeln sind eine Spezialität von  $\text{\LaTeX}$ , sie werden im Text durch `\(` und `\)` abgegrenzt. Wenn man umfangreichere Formeln setzen möchte, greift man auf eine der Umgebungen `equation`, `eqnarray` oder `\[` und `\]` zurück.

**Hinweis 1** *Abgesetzte Formeln, sollten nicht über `$$...$$` erzeugt werden, dies ist ein `plain-TeX`-Befehl und führt zu inkonsistenten vertikalen Abständen, darüber hinaus funktioniert die Klassenoption `fleqn` nicht mehr (vgl. [Tre04]). Analoges gilt für `$. . . $`.*

<sup>1</sup>Solche Vertreter sind das Kommentarzeichen „%“, das Dollarzeichen „\$“ welches in den Mathemodus umschaltet das „&“ für Tabulatoren und die geschweiften Klammern „{}“. Weitere sind die Tilde „~“ für geschützte Leerzeichen, das Doppelkreuz „#“, der Unterstrich „\_“ sowie das Dach „^“.

<sup>2</sup>Aber: Der Text muss innerhalb einer Zeile auch wieder beendet werden.

Auch `eqnarray`, das die gezielte Positionierung von Gleichungssystemen zu erlauben scheint, sollte nicht mehr verwendet werden (vgl. [Com06, Frage 360])

**Veränderung für Barrierefreiheit 1 (Alternativtext)** Formeln zählen zu den Elementen nicht textlichen Inhaltes. Sie werden häufig als Bilder eingebunden und können deswegen von assistiven Technologien nicht interpretiert werden. Daraus ergibt sich die Notwendigkeit alternativer Texte für Formeln. Sie können zwar aufgrund ihrer Darstellung als Text in  $\LaTeX$  automatisch generiert werden, jedoch kann es hierbei zu Problemen kommen. Mittels Leerzeichen strukturierte oder mit Formatierungsanweisungen durchsetzte  $\LaTeX$ -Formeln sind schwer zu lesen.

#### Erstellen einer Formel

```
1 \alt{a/b=5}
2 \[\frac{a}{b}=5\]
```

#### Darstellung der Formel

$$\frac{a}{b} = 5$$

Abbildung 2.4: Beispiel für die Verwendung einer Formel

### Aufzählungen

Es gibt drei verschiedene Arten von Listen in  $\LaTeX$ , zum Einen, die durch Aufzählungszeichen eingeleitete Liste (`itemize`), zum Anderen nummerierte Listen (`enumerate`). Die letzte Art wird für Begriffserklärungen (`description`) verwendet, wobei das Label den Begriff enthält.

#### Erstellen einer Aufzählung

```
1 \begin{itemize}
2   \item erster Punkt
3   \item zweiter Punkt
4   \item dritter Punkt
5 \end{itemize}
```

#### Darstellung der Aufzählung

- erster Punkt
- zweiter Punkt
- dritter Punkt

Abbildung 2.5: Beispiel für die Verwendung einer Aufzählung

Die einzelnen Punkte der Liste werden durch die Anweisung `\item` eingeleitet. Der gesamte Text, der zu einem Unterpunkt gehört, wird im Blocksatz untereinander gesetzt. Ein Unterpunkt der Liste kann durch Schachtelung der Umgebungen weiter unterteilt werden. Eine tiefere Schachtelung als in vier Ebenen ist nicht möglich.

## Theoreme

In jedem Mathematikbuch können Sie nachsehen: alles wird gezählt. Beispiele, Sätze, Definitionen, Lemmata, Axiome werden zu Zwecken der Übersichtlichkeit durchnummeriert. Auch dabei können Sie sich der Unterstützung von  $\text{\LaTeX}$  gewiss sein. Sie haben die Möglichkeit, sich genau solche Umgebungen (Theoreme) zu definieren, die dann bei ihrer Benutzung automatisch nummeriert werden.

Dazu wird in der Dokumentenpräambel die neue Umgebung mittels `\newtheorem{<Name>}{<Marke>}` propagiert und kann im eigentlichen Text verwendet werden. Beim Eintritt in die so definierten Umgebungen können Sie als Option noch einen Text angeben, der automatisch fett und in Klammern mit ausgegeben wird.

### Erstellen eines Theorems

```
1 \newtheorem{Def}{Definition}
2 \begin{Def} Eine Schätzung
3   heißt erwartungstreu, wenn
4   ihr Erwartungswert ...
5 \end{Def}
```

### Darstellung des Theorems

**Definition 1** *Eine Schätzung heißt erwartungstreu, wenn ihr Erwartungswert ...*

Abbildung 2.6: Beispiel für die Verwendung eines Theorems

### 2.3.3 Strukturen auf Zeilenebene

Elemente auf Zeilenebene kennzeichnen Textteile, die ein bestimmtes Format oder Verhalten aufweisen. Sie können in Elementen auf Blockebene enthalten sein. Sie treten folglich häufig im Fließtext auf. Beispiele sind: Referenzen, Fußnoten, aber auch Formeln oder Codeumgebungen.

#### Referenzen

Die Stelle auf die Sie im Dokument verweisen wollen, markieren Sie mit den normalen  $\text{\LaTeX}$ -Querverweisbefehlen `\label{<Labelname>}`. Der eigentliche Verweis wird mit `\ref{<Labelname>}` für die Nummer oder `\pageref{<Labelname>}` für die Seitenzahl getätigt.

Die Funktionen des `varioref`-Paketes<sup>3</sup> können dabei verwendet werden. Viele Erweiterungen bzgl. Referenzen bietet das `hyperref`-Paket. Es wird im [Unterabschnitt 3.2.1](#) vorgestellt, dort ist auch das zugehörige Beispiel zu finden.

<sup>3</sup>Für weitere Informationen ist die Dokumentation des Paketes `[RO04]` zu Rate zu ziehen.

## Fußnoten

Auch Fußnoten sind eine Art von Referenzen, sie werden mit `\footnote{<Fußnoten-text>}` erzeugt und direkt im Textfluss angegeben, die Nummerierung und Positionierung wird von  $\LaTeX$  übernommen.

### Erstellen einer Fußnote

```
1 ... , ein Absatz mit Fußnote
2 \footnote{Hier ist der Text
3   der Fußnote einzufügen.}
4 und schon kann der Text
5 weitergehen. ...
```

### Darstellung der Fußnote

..., ein Absatz mit Fußnote <sup>a</sup> und schon kann der Text weitergehen. ...

---

<sup>a</sup>Hier ist der Text der Fußnote einzufügen.

Abbildung 2.7: Beispiel für die Verwendung einer Fußnote

## Literaturverweise

Häufig sind im Text zudem Literaturverweise zu finden. Sie werden über `\cite[<weitere Angaben>]{<Schlüssel des Eintrages>}` erzeugt, wobei das erste Argument optional ist und den Literaturverweis näher beschreibt, also z. B. durch Angabe der Seite (S. 35).

### 2.3.4 Gleitumgebungen

Verteilt über das Ihnen vorliegende Dokument finden Sie mehrere Abbildungen, die automatisch durchnummeriert werden und auf deren Nummer überall im Dokument Bezug genommen werden kann. Bei solchen Abbildungen handelt es sich um sogenannte „wandernde Objekte“ oder „Gleitumgebungen“. Sie werden nämlich nicht nur automatisch durchnummeriert, sondern nach gewissen Regeln auch frei positioniert. Es kann Ihnen also durchaus passieren, dass die Abbildung beim Druck Ihres Dokumentes nicht dort erscheint, wo Sie sie in der Eingabe kodiert haben. Mit Hinblick auf den Seitenumbruch wählt  $\LaTeX$  einen Platz für die Abbildung, der ein optimales Layout garantiert.

Tabellen und Bilder treten meist in Gleitumgebungen auf. Die umschließende Umgebung, die auch den `\caption`-Befehl für die Bildunterschrift und die Sprungmarke `\label` enthält, ist die `table`- bzw. `figure`-Umgebung.

## Integration von Bildern

Das Paket `graphicx` dient zur Einbindung von Grafiken in ein  $\LaTeX$ -Dokument. Die Größe und der Drehwinkel der Abbildung kann als `<Parameterliste>` übergeben werden. Die eigentliche Einbindung erfolgt über den Befehl `\includegraphics[<Parameterliste>]{<Pfad/Dateiname>}`. Ein `<Pfad>` ist nur nötig, wenn die Grafiken in einem Unterverzeichnis gespeichert sind, dies wird aus Gründen der Übersichtlichkeit angeraten.

Die Bilder müssen für  $\LaTeX$  im \*.eps-Format vorliegen, pdf $\TeX$  kann darüber hinaus auch mit den Formaten \*.pdf, \*.png und \*.jpg umgehen. Die `\includegraphics`-Angabe sollte ohne Endung gemacht werden, dann suchen sich  $\LaTeX$  beziehungsweise pdf $\TeX$  selbständig das richtige Format aus. Neu hinzu kommt die Angabe von alternativen Beschreibungen (vgl. Unterabschnitt 3.1.2).

## Tabellen

Tabellen sind in  $\LaTeX$  das Einzige, was ein wenig unübersichtlich ist. Es gibt viele Erweiterungen, die sich Tabellen widmen. Für die Erstellung von einfachen, aber auch sehr komplexen Tabellen stehen Ihnen zwei verschiedene Verfahren zur Verfügung: das Benutzen von Tabulatorstellen für ein einfaches Tabellenlayout und das Verwenden von echten Tabellen, die horizontale und vertikale Linien in beliebigen Ausführungen ermöglichen und mehrspaltige Überschriften bieten.

Die erste Variante, das Arbeiten mit Tabulatoren unter Benutzung der `tabbing`-Umgebung gestattet es, in einer Zeile Tabulatorpositionen festzulegen, an denen der Text mit besonderen Sprungkommandos positioniert werden kann. Laut [Jür00] dienen Tabulatoren nicht nur der Gestaltung von Tabellen, sondern können auch für sonstige Einrückungen verwendet werden. Damit findet in diesem Fall keine wirklich logische Auszeichnung statt, zudem eignen sich `tabbing`-Umgebungen auch nur für kleinere Tabellen. Sie sollten also nach Möglichkeit nicht verwendet werden.

Wirklich „schöne“ Tabellen können in der `tabular`-Umgebung erstellt werden. Sie stellt eine große Auswahl weiterer Kommandos zur Gestaltung zur Verfügung. Direkt als Argument wird der `tabular`-Umgebung die Spaltendefinition übergeben. Wobei „l“ für eine linksbündige, „r“ für eine rechtsbündige und „c“ für eine zentrierte Ausrichtung der Spalte steht. Der senkrechte Strich „|“ erzeugt eine senkrechte Linie in Tabellenhöhe. Die Anzahl der Ausrichtungsmerkmale legt zugleich die Spaltenanzahl fest. Innerhalb der Tabelle trennt ein „&“ die einzelnen Zellen, „\\“ die einzelnen Zeilen und „\hline“ erzeugt einen waagerechten Strich. Schönerer Tabellenlinien erzeugt das `booktabs`-Paket.

Welche Probleme bzgl. der Barrierefreiheit in falsch gestalteten Tabellen auftreten können, soll das folgende Beispiel zeigen. Diese Tabelle (vgl. [Abbildung 2.8](#)) wurde in [Jür00] als Beispiel für die Darstellung der Möglichkeiten des Paketes verwendet. Aus Darstellungs- und Platzgründen, wurden die Tabelleneinträge abstrahiert.

Die zugrunde liegende Struktur der Tabelle unterscheidet deutlich von der logischen Struktur, die über die Linien angedeutet wurde. So stehen nach Willen des Autors „A3, B3 und C3“ in einer Zelle. Dies ist aus der syntaktischen Struktur jedoch nicht abzuleiten. Den automatischen Transformationswerkzeugen nach XHTML und PDF fehlt diese Information. Sie würden streng nach Transformationsregeln die Zellen und Zeilen erzeugen. Damit wird der Screenreader, die Tabelle in der Reihenfolge ( A1, A2, A3, B1 (leer), B2(leer), B3 ...) vorlesen. Das ist im ersten Fall noch nicht gravierend, allerdings wird nach F2 nicht G2 sondern F3 vorgelesen.



### Erstellen einer Tabelle

```
1 \begin{tabular}{clr}  
2 \hline  
3 A1 & A2 & A3 \\  
4 & & B3 \\  
5 & & C3 \\  
6 D1 & D2 & D3 \\  
7 & E2 & \\  
8 F1 & F2 & F3 \\  
9 & G2 & G3 \\  
10 \end{tabular}
```

### Darstellung der Tabelle

A1	A2	A3
		B3
		C3
D1	D2	D3
	E2	
F1	F2	F3
	G2	G3

Abbildung 2.8: Beispiel für die Verwendung einer Tabelle

### Erstellen einer Tabelle

```
1 \begin{tabular}{p{0,7cm}  
2 p{0,7cm}p{0,7cm}}  
3 \hline  
4 A1 & A2 & A3 B3 C3 \\  
5 D1 & D2 E2 & D3 \\  
6 F1 & F2 G2 & F3 G3 \\  
7 \end{tabular}
```

### Darstellung der Tabelle

A1	A2	A3
		B3
		C3
D1	D2	D3
	E2	
F1	F2	F3
	G2	G3

Abbildung 2.9: Beispiel für die Verwendung einer Tabelle

Durch die Verwendung einer „p“-Spalte wird der automatische Umbruch aktiviert. Unter Angabe der Spaltenbreite, wird der Text in der Tabelle entsprechend umgebrochen. Das eben gebrachte Beispiel (vgl. *Abbildung 2.8*) wurde unter Verwendung dieser Angaben so umformuliert (vgl. *Abbildung 2.9*), dass die Tabelle nun in korrekter Lesereihenfolge vorgelesen wird. Diese Vorgehensweise sollte bei der Konstruktion eigener Tabellen adaptiert werden.

Für längere Tabellen, die die Länge einer Seite überschreiten, sei das `longtable`-Paket empfohlen. Es stellt die Funktionalität für mehrseitige Tabellen bereit. Wobei die Funktionalität der sich wiederholenden Tabellenkopfzeilen die Lesbarkeit erhöht. Das Paket wird aber in der momentanen Version der Transformation noch nicht umgesetzt.

Die Umsetzung der horizontalen Tabellenlinien erfolgt in  $\text{T}_{\text{E}}\text{X}4\text{ht}$  standardmäßig sehr kurios. Es werden Tabellenzeilen erzeugt, die `<hr>`'s enthalten. Das ist weder besonders schön, noch barrierefrei. Da dieses Verhalten bisher nicht geändert werden konnte, empfiehlt sich die Verwendung des `booktabs`-Pakets. Die hier definierten Linienarten werden per CSS umgesetzt und sind auch sonst schöner (vgl. [Rei99]).

### 2.3.5 Verzeichnisse

Aufgrund der Gliederungsbefehle im Dokument ist  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  in der Lage das Inhaltsverzeichnis automatisch zu erstellen. Sie benötigen dazu lediglich den Befehl `\tableofcontents` der dafür sorgt, dass Ihre Überschriften zusammen mit der zugehörigen Seitenzahl in einer zusätzlichen Datei verwaltet werden. Aufgrund dieser Eintragungen wird das Inhaltsverzeichnis an der Stelle im Dokument ausgedruckt<sup>4</sup>, an der der Befehl `\tableofcontents` steht.

Ebenso ist eine automatische Erstellung von Abbildungs- und Tabellenverzeichnissen möglich. Sie sind über `\listoffigures` bzw. `\listoftables` an geeigneter Stelle ins Dokument zu einbinden.

Weitere Verzeichnisse werden durch Erweiterungspakete/-programme bereitgestellt. Nennenswert sind an dieser Stelle das Literaturverzeichnis, der mögliche Index, das Glossar aber auch das Abkürzungsverzeichnis. Sie werden im weiteren Verlauf (vgl. *Unterabschnitt 3.3.1*) besprochen.

Zusammenfassend lässt sich sagen, dass die einzelnen Strukturen gut umgesetzt werden. Die Schachtelung von Strukturen ist jedoch kritisch zu werten. Geschachtelte Listen haben sicher ihren Sinn, solange eine gewisse Tiefe<sup>5</sup> nicht überschritten wird. Ob eine Schachtelung von Listen in Tabellen zielführend ist, hängt vom jeweiligen Nutzungsfall ab. Eine Kombination von mehreren Tabellen ineinander birgt spätestens Probleme in

<sup>4</sup>Sie müssen  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  mindestens zweimal über Ihr Dokument laufen lassen, um ein korrektes Inhaltsverzeichnis zu erstellen.

<sup>5</sup> $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  erlaubt bis zu vier Schachtelungsebenen, je nach verwandter Listenart.

der Transformation. Hier muss im Zweifelsfall immer getestet werden, ob die Strukturinformation korrekt umgesetzt werden, wenn nicht sollte die Schachtelung jedoch vermieden werden.

## 2.4 Weiterführende Literatur

Wer ernsthaft mit  $\LaTeX$  arbeiten möchte, kommt um das intensive Lesen von Dokumentation nicht herum. Im Folgenden habe ich eine Sammlung guter und nützlicher Skripte zusammengestellt.

**The not so Short Introduction to  $\LaTeX$**  Eine allgemeine Einführung in  $\LaTeX$ . Eine solche ist unbedingt notwendig, bevor größere Dokumente mit  $\LaTeX$  erstellt werden.

**Dokumentation des Koma-Scripts** Das Koma-Script ist eine Sammlung von Klassen, die  $\LaTeX$  an deutschsprachige Typographieregeln anpasst. Des Weiteren stellt es sehr mächtige Möglichkeiten zur Anpassung des Seitenlayouts bereit.

**$\LaTeX$ -Sündenregister** Eine Liste häufig gemachter Fehler mit besseren Alternativen.

**TeX-FAQ** Eine umfangreiche Liste häufig gestellter Fragen mit Antworten zu  $\LaTeX$ .

Weitere Bücher und Links sollten sich über die zielgerichtete Suche im Buchladen oder im Internet finden lassen. Ein wichtiger Anlaufpunkt ist Dante, die Deutschsprachige Anwendervereinigung TEX e.V. im Web vertreten unter <http://www.dante.de>.

## 3 Anleitung zur Gestaltung barrierefreier Dokumente

Der Autor sollte die Gedanken der Barrierefreiheit des Dokuments von Anfang an berücksichtigen. Nur so kann ein barrierefreies wissenschaftliches Dokument entstehen. Die zugrunde liegenden Zusammenhänge werden in diesem Kapitel erläutert. Neben einer sinnvollen Strukturierung des Dokumenteninhalts, hängt die Barrierefreiheit der erzeugten Datei auch maßgeblich von dem Dokumenteninhalt, Formatierungen und der Kombination von Erweiterungspaketen ab. Eine Unzulänglichkeit besteht häufig z. B. in der schlechten Strukturierung der Dokumente. Das führt für normal Sehende schon zu Orientierungs- und Verständnisproblemen. Nutzer mit Lern- und Wahrnehmungsbehinderungen haben eine noch geringere Chance die Dokumente zu verstehen.

### 3.1 Maßnahmen zur Wahrnehmbarkeit

Als ein Prinzip der Barrierefreiheit nennt die Web Content Accessibility Guideline (WCAG) die Wahrnehmbarkeit. Sie setzt voraus, dass sämtliche Inhalte und Funktionen so dargestellt werden, dass sie von allen Benutzern erkannt werden können. In der Praxis bedeutet das, dass alle Informationen auch für Nutzer mit Sinnesbeeinträchtigungen<sup>1</sup> zugänglich sein sollen.

#### 3.1.1 Schrift

Wenn Sie Ihr Dokument mit  $\LaTeX$  erstellen, wird standardmäßig der Zeichensatz „Computer Modern“ benutzt. Andere Schriften<sup>2</sup> sind verfügbar, allerdings ist bei ihrer Einbindung einiges zu beachten.

Schriften in PDF-Dateien werden mit schlechter Qualität dargestellt (und evtl. auch gedruckt), wenn sie in der PDF-Datei als Bitmaps enthalten sind, also nicht in Form von Vektorfonts vorliegen. Um das zu vermeiden, dürfen im Dokument nur solche Schriften<sup>3</sup> verwendet werden, die im Format Type-1 (PostScript) vorhanden sind. Schriften, die ausschließlich im Metafont-Format<sup>4</sup> existieren, können hingegen in PDF-Dateien nur in Form von Bitmaps benutzt werden, sind also unbedingt zu vermeiden.

---

<sup>1</sup>Das Sehvermögen spielt im Rahmen visuell präsentierter wissenschaftlicher Publikationen eine größere Rolle als das Hörvermögen.

<sup>2</sup>Alle Schriften müssen für eine Nutzung vorher beschafft und installiert werden, sollten sie nicht in der installierten Distribution enthalten sein.

<sup>3</sup>Welche das sind, sollte aus der Dokumentation des jeweils verwandten  $\LaTeX$ -Systems hervorgehen.

<sup>4</sup>Wenn die Schriften in mehreren Formaten vorliegen, verwendet  $\LaTeX$ /pdf $\TeX$  die Richtige.

Von der Verwendung des AE-Fonts, einem virtuellen Font, kann die Autorin ebenfalls nur abraten, die Zusammensetzung von Zeichen aus mehreren anderen führt zu erheblichen Problemen bei der Textextraktion und bei der Suche nach Wörtern mit Umlauten.

Auch wenn die Schriftart bei der Transformation nach XHTML nicht übernommen wird, spielt sie keine unbedeutende Rolle. Bei der Erstellung der DVI-Datei mit Zusatzinformationen für den T<sub>E</sub>X<sub>4</sub>ht-Generator, müssen alle Sonderzeichen korrekt kodiert sein, also in der Schriftart vorhanden, ansonsten sind auch die betreffenden Zeichen in den XHTML-Dateien falsch.

Um das Problem zu umgehen, sollte man ausschließlich Schriften der T1-Kodierung verwenden, in denen die Umlaute tatsächlich enthalten sind. Zur Zeit sind dies die Schriften, die durch die Pakete lmodern und cm-super bereitgestellt werden. Da die Schriften (mit ihren zahlreichen Schriftschnitten und Sonderzeichen) in das entstehende PDF-Dokument eingebettet werden, erhöht sich die Dateigröße möglicherweise mehr als bei anderen (unvollständigen) Schriftarten. Wobei cm-super wesentlich größere Dateien verursacht, da die zugrunde liegende Schriften über ein Rastern der EC Type3 Bitmap-Schriften entstanden sind, während lmodern eine komplett neu entwickelte Schriftfamilie mit einer minimalen Anzahl an Punkten ist.

**Hinweis 2** *In PDF eingebettete Schriftarten müssen lizenzfrei sein. Die Benutzung von kommerziellen Fonts (z. B. aus dem mathtimes-Paket) ist daher eine mögliche Urheberrechtverletzung. Um zu überprüfen, welche Fonts in Ihr Dokument eingebettet und von welchem Typ diese sind, können Sie im Acrobat Reader unter (Datei → Dokumenteigenschaften → Schriften) nachsehen.*

### 3.1.2 Alternative Beschreibung

Wissenschaftliche Publikationen können grafische Darstellungen (Bilder, Formeln, Grafiken, Diagramme, Schemata und viele andere mehr) enthalten. Für blinde Personen können insbesondere rein grafische Umsetzungen von Inhalten zu nicht überwindbaren Barrieren führen. Ihnen sollte ein beschreibender Text zugeordnet sein, der sowohl den Inhalt als auch die Funktion wiedergibt.

**Anforderung 1** *Für jeden visuellen Inhalt sind geeignete äquivalente Inhalte, die den gleichen Zweck oder die gleiche Funktion wie der originäre Inhalt erfüllen, bereitzustellen.*

**Veränderung für Barrierefreiheit 2** *Der Befehl für die textliche Beschreibung wird mit dem accessibility-Paket bereitgestellt. Er wird über `\alt{<Alternativ Text>}` eingebunden. Er sollte vor der eigentlichen Grafikeinbindung stehen, damit er verarbeitet werden kann.*

### 3.1.3 Farbe

**Anforderung 2** *Alle mit Farbe dargestellten Informationen müssen auch ohne Farbe verfügbar sein, z. B. durch den Kontext oder die hierfür vorgesehenen Elemente.*

Befehle zur Hervorhebung durch Hintergrundschattierungen oder Gestaltung mittels Farben, werden in  $\LaTeX$  über das Paket `color` bereitgestellt. Bei der Verwendung von Farbe zur Hervorhebung ist darauf zu achten, dass Farbe nie alleiniges Unterscheidungsmerkmal ist. Dazu sollten gegebenenfalls eigene logische Strukturen (Makros) definiert werden (vgl. [Unterabschnitt 3.4.2](#)).

### 3.1.4 Kontrast

Wählen Sie einen ausreichenden Farbkontrast für alle Elemente des Dokumentes. Für Menschen mit Sehbehinderung sind diese so einfacher zu lesen.

Bisher sind die Einstellungsmöglichkeiten im Adobe Acrobat noch nicht ausgereift. Aber kann sowohl die Hinter- als auch die Vordergrundfarbe ändern, was Menschen hilft, die bestimmt Kontrastverhältnisse benötigen.

Ein schwarzer Text vor weißem Hintergrund<sup>5</sup> führt bei inverser Farbeinstellung zu „weiß auf weiß“, da Hintergrundboxen nicht mit eingefärbt werden. Eine mögliche Option ist auch, die ganze Schrift durch weiße zu ersetzen, daraus ergeben sich jedoch Schwierigkeiten mit hellen Hintergründen. Bisher kann von einer Verwendung von Hintergründen nur abgeraten werden, es bleibt jedoch zu hoffen, dass dieses Problem in zukünftigen Versionen der Adobe-Software beseitigt wird.

## 3.2 Maßnahmen zur Bedienbarkeit

Bedienbarkeit stellt sicher, dass die Benutzeroberfläche von allen Benutzern bedient werden kann. Damit trägt auch sie maßgeblich zur Barrierefreiheit bei. Der folgende Abschnitt enthält daher Richtlinien zur Sicherstellung der Bedienbarkeit.

### 3.2.1 Das `hyperref`-Paket

Damit Hypertext-Informationen in ein Dokument eingebunden werden können, wird das  $\LaTeX$ -Paket `hyperref`, welches von Heiko Oberdiek und Sebastian Rahtz entworfen worden ist, benötigt. Es erweitert die Funktionalität vorhandener Verweis-Befehle von  $\LaTeX$  und fügt neue Befehle hinzu, um z. B. auf externe Dokumente zu verweisen. Das Paket wird über `\usepackage{hyperref}` eingebunden und muss als letztes Paket im jeweiligen  $\LaTeX$ -Dokument deklariert werden, da es viele  $\LaTeX$ -Befehle neu definiert und somit Befehle anderer Pakete überschreiben könnte (vgl. [\[RO04\]](#)).

---

<sup>5</sup>Zum Testen wurde dieser Teil des Textes weiß hinterlegt. Die Einstellung ist unter „Bearbeiten“ → „Grundeinstellung...“ → „Zugriff“ zu finden.

Ist das Paket `hyperref` geladen, werden vorhandene Verweise automatisch zu aktiven Hyperlinks. Das erhöht die Interaktivität und Bedienbarkeit des Ergebnisdokumentes. Im Normalfall werden diese farbig umrahmt, durch Optionen wie `colorlinks=true` können diese aber auch selbst farbig hervorgehoben werden. Zu beachten sind hierbei die Regeln zur Verwendung von Farbe [Unterabschnitt 3.1.3](#).

Da `hyperref` sehr viele Möglichkeiten und Befehle zur Verfügung stellt, werden hier nur ein paar Beispiele, die besonders häufig benötigt werden, erläutert. Genaueres können Sie z. B. unter [\[RO04\]](#) nachlesen.

**Anforderung 3** *Der Nutzerin, dem Nutzer sind Informationen zum Kontext und zur Orientierung bereitzustellen.*

Mit der Option `bookmarks` des `hyperref`-Paketes werden Lesezeichen erstellt, die durch PDF-Viewer angezeigt werden können. Sie dienen der Navigation in PDF-Dokumenten und sind mit Navigationsleisten in XHTML vergleichbar. Mit der Option `bookmarks-open=true` wird erreicht, dass der Baum der Lesezeichen beim Öffnen des Dokuments bis zur kompletten Tiefe geöffnet wird, dass erleichtert den Überblick.

### Verweis auf eine andere Stelle im selben Dokument

**Veränderung für Barrierefreiheit 3 (Bedienbarkeit)** *Besser als die Verwendung von `\ref{<Labelname>}` ist die Nutzung des `\autoref{<Labelname>}` Befehls aus dem `hyperref`-Paket. Dieser erspart bei der Eingabe die Angabe der Verweisart, denn die Art des Labels (Abbildung, Tabelle etc.) wird automatisch<sup>6</sup> gesetzt. Damit können Änderungen im gesamten Dokument schneller erfolgen und man spart sich Schreibarbeit. Zudem sind die gesetzten Linkbereiche durch die Mitverlinkung der Labelart größer und können vom Leser des Dokumentes besser erreicht werden.*

#### Erstellen einer Referenz

```
1 .., ein Absatz mit einem  
2 Verweis auf dieses Kapitel  
3 \ref{sec:Einfuehrung} und als  
4 Vergleich ein Verweis auf das  
5 \autoref{sec:Einfuehrung}
```

#### Darstellung der Referenz

.., ein Absatz mit einem Verweis auf dieses Kapitel 2 und als Vergleich ein Verweis auf das [Kapitel 2](#)

Abbildung 3.1: Beispiel für die Verwendung einer Referenz

<sup>6</sup>Für die Benennung der Label greift `hyperref` auf die Klassenoption `ngerman` zu, anstatt auf die Sprachvariable des `babel`-Paketes.

## Verweis auf eine URL im Internet

Möchten Sie auf eine beliebige Internetadresse verweisen, bzw. einen Link darauf in Ihre Arbeit einfügen, so ist dies auch über das hyperref-Paket realisierbar. Für eigene externe Verknüpfungen wird der Befehl `\href{<URL>}{<Text>}` bereitgestellt. Als `<URL>` muss eine vollständige Internetadresse angegeben werden, der `<Text>` erscheint später im Dokument.

**Veränderung für Barrierefreiheit 4** *Es wird empfohlen die vollständige URL in Klammern hinter dem verlinkten Text oder als Fußnote anzugeben. Geschieht dies nicht, besteht die Gefahr, dass die zugrunde liegende URL nicht mehr zugänglich ist, z. B. durch einen Schwarz-Weiß-Ausdruck.*

### 3.2.2 Seitenzahlen

Einige Dokumentenklassen, wie die book-Klasse, ändern automatisch die Darstellung der Seitennummerierung. Das ist für den menschlichen Leser nützlich, allerdings irritiert, dass das hyperref-Paket. Durch den Neustart der Nummerierung kommt es zu doppelten Seitenzahlen. Glücklicherweise gibt es Optionen, die hyperref das Richtige tun lassen.

Zum Einen kommt die Option `plainpages=false` in Frage. Hiermit werden die Seitenanker als formatierte Seitennummer (z. B. „ii“ oder „2“) gesetzt, wodurch eine Unterscheidung möglich ist. Standardmäßig ist diese Option allerdings nicht gesetzt.

Die andere Variante ist `pdfpagelabels` damit werden die Seitenzahlen im Acrobat Reader angepasst. D. h. es wird „ii (4 von 40)“ statt nur „4 von 40“ angezeigt.

Die Optionen sollten benutzt werden, wann immer die Seitennummerierung nicht nur „1..n“ ist. Diese Lösung ist noch nicht perfekt, wenn nach einer unnummerierten Titelseite der Zähler zurückgesetzt wird, kommt es trotzdem zu einer Warnung „duplicate destinations“ von pdfTeX. Empfohlen wird die Seitenzahlen von „1..n“ durchzunummerieren, das ist für unerfahrende Leser intuitiver.

### 3.2.3 Das makeidx-Paket

L<sup>A</sup>T<sub>E</sub>X bringt ein vielfältiges Programm mit dem Namen „MakeIndex“ mit, das es leicht macht Dinge zu indizieren. Es wurde von Pehong Chen und Michael A. Harrison entworfen. Es stellt neue Index-Befehle zur Verfügung. Es kann ein sortiertes Sachregister erstellt werden. Beispielsweise können die Einträge beliebig formatiert oder und die Seitenbereiche angegeben werden.

Ein Paket, dass diese Funktionalität nutzt ist makeidx. Es dient der Erstellung von Registern, die zumeist am Ende des Dokuments eingebunden werden und schnellen Zugriff auf wichtige Stellen im Dokument bieten. Durch die Verwendung eines Registers kann also die Bedienbarkeit eines Dokumentes verbessert werden.



```

1 \documentclass []{scrrept}
2 \usepackage [ngerman]{babel}
3 \usepackage [latin1]{inputenc}
4 \usepackage{makeidx}
5   \makeindex
6 \begin{document}
7   %
8   \index{Latex}
9   %
10  \printindex
11 \end{document}

```

Wenn Sie einen Index benötigen, so wird  $\LaTeX$  in der Präambel mit `\makeindex` ankündigen, dass es die nötigen Dateien erstellen soll. Einzelne Einträge werden mittels `\index{<Eintrag>}` erstellt. Der `<Eintrag>` kann aus einem einzelnen Wort oder einer Kombination von Wörtern bestehen, die durch ein „!“ voneinander getrennt sind. Das Ausrufezeichen erzeugt Untereinträge. Jedem Eintrag sind die Seitenzahlen des Auftretens zugeordnet.

### 3.3 Maßnahmen zur Verständlichkeit

Verständlichkeit bedeutet, dass die Inhalte eines Informationsangebots klar und nachvollziehbar ausgedrückt und dargestellt sind. Die Verständlichkeit des Inhaltes kann nur durch den Autor selbst erreicht werden. Damit der Rezipient die präsentierten Informationen verstehen kann, muss berücksichtigt werden, dass Menschen auf unterschiedliche Art lernen und dass sie verschiedene Hintergründe und Erfahrungen haben. Die Einhaltung aufgeführter Maßnahmen ist Prämisse zur Gewährleistung der Barrierefreiheit.

Das allgemeine Verständnis des Dokumenteninhaltes kann durch die Verwendung einer einfachen Wortwahl gesteigert werden. Diese Forderung nützt Menschen anderer Muttersprache oder solchen mit kognitiven Einschränkungen. Es darf jedoch nicht soweit vereinfacht werden, dass wichtige Inhalte verloren gehen. Eine Anleitung z. B. bietet die „europäische Richtlinie für die Erstellung von leicht lesbaren Informationen für Menschen mit geistiger Behinderung“.

In wissenschaftlichen Publikationen betrifft diese Forderung, neben der möglichst klaren Strukturierung insbesondere die Verwendung von Fachwörtern oder Abkürzungen bzw. Akronymen. Sie sollten bei ersten Vorkommen im Text gekennzeichnet und erläutert werden. Für umfangreichere Arbeiten bieten sich zudem ein Abkürzungsverzeichnis oder Glossar<sup>7</sup> an.

<sup>7</sup>Ein Glossar (engl.: glossary) ist ein Spezialverzeichnis, z. B. eine alphabetische Anordnung der verwendeten Grundbegriffe nebst Erläuterung. [Kop00a] Es unterstützt damit die Verständlichkeit von wissenschaftlichen Texten, in denen häufig Fachbegriffe verwendet werden.

### 3.3.1 Das glossary-Paket

Das glossary-Paket von Nicola L.C. Talbot stellt Befehle bereit die Autoren bei der Generierung eines Glossars unterstützen. Darüber hinaus bietet es auch die Möglichkeit Akronyme und Abkürzungen zu handhaben.

#### Glossar

Zusätzlich muss der Befehl `\makeglossary` in der Dokumentenpräambel stehen. Die eigentliche Definition von Glossareinträgen kann an beliebiger Stelle im Dokumente mittels `\glossary{<Glossareintrag>}` vorgenommen werden. Der Begriff hat keinen Bezug zum verwendeten Begriff im Fließtext. Sinnvoller ist daher die Verwendung des Befehls `\storegloentry[<Optionen>]{<Label>}{<Glossareintrag>}` zum Speichern der Glossareinträge. Auf den Eintrag kann später noch über `\useGloentry{<Label>}{<Text>}` zugegriffen werden. Wobei der `<Text>` im Fließtext erscheint und auf den mittels `<Label>` angegebenen Glossareintrag verlinkt<sup>8</sup> ist. Noch kürzer ist die Verwendung von `\gls{<Label>}`, die Funktionsweise ist mit der von `\useGloentry` vergleichbar, allerdings wird der verlinkte Text automatisch<sup>9</sup> aus dem Glossar extrahiert.

Für die einmalige Verwendung eignet sich auch der Befehl `\xglossary{<Glossareintrag>}{<Text>}` wobei sowohl der `<Text>` im Fließtext mit einem Link auf den `<Glossareintrag>`, als auch dieser erzeugt wird. Untenstehender Dokumentauszug zeigt exemplarisch die Verwendung der Makros.

```
1 \documentclass []{scrrept}
2 \usepackage[ngerman]{babel}
3 \usepackage[latin1]{inputenc}
4 \usepackage{glossary}
5   \makeglossary
6 \begin{document}
7   \printglossary
8   %
9   \storegloentry{Latex}
10    {name={\LaTeX\ (kurz: Lamport \TeX)}, sort={Latex},
11     description={Makropaket zu \TeX\ von Leslie Lamport, dass dem
12     Benutzer das Erstellen von Dokumenten mit \TeX\ erleichtert.}}
13   %
14   ... \useGloentry{Latex}{\LaTeX}...
15 \end{document}
```

Eine Sammlung der Begriffe wird durch den Befehl `\printglossary` als Liste von Beschreibungseinträgen ins Dokument eingebunden. Eine detaillierte Dokumentation ist in dem Paket selbst enthalten (vgl. [Tal06]).

<sup>8</sup>Das glossary-Paket muss nach hyperref eingebunden werden, sonst wird im PDF nicht verlinkt.

<sup>9</sup>Vorsicht ist in diesem Fall bzgl. der korrekten Beugungsform geboten.

## Abkürzungen und Akronyme

**Anforderung 4** *Abkürzungen und Akronyme sind an der Stelle ihres ersten Auftretens im Inhalt zu erläutern und durch die hierfür vorgesehenen Elemente der verwendeten Sprache kenntlich zu machen.*

Ein neues Akronym wird mit `\newacronym[<Befehl>]{<Akronym>}{<Langform>}{<Glossareintrag>}` zur Verfügung gestellt (also z. B. `\newacronym[BITV]{Barrierefreie Informationstechnik Verordnung}{description=Verordnung ...}`). Im weiteren Verlauf kann dann mittels `\useacronym{BITV}` oder `\BITV` auf das Akronym zugegriffen werden, dabei entscheidet das System selbst, ob es das erste Vorkommen im Text ist. Wird dabei auf die Kurzform zugegriffen, so ist diese auf den `<Glossareintrag>` verlinkt. Über `\BITVlong` bzw. `\BITVshort` kann explizit auf die Lang- bzw. Kurzform (bezogen auf das obige Beispiel: Barrierefreie Informationstechnik Verordnung bzw. BITV) zugegriffen werden, der Befehl für die Langform erzeugt allerdings keinen Hyperlink.

**Hinweis 3** *Mit `\newacronym` angelegte Akronyme werden nur nach Verwendung in Glossar aufgenommen. Der Befehl `\useglossentry{*}` ermöglicht die Eintragung sämtlicher gespeicherter Glossareinträge ins Glossar.*

Ebenso wie bei Glossareinträgen kann ein Eintrag auch über `\acronym{<Glossareintrag>}{<Text>}` erzeugt werden, dieser `<Text>` ist verlinkt, der Eintrag kann jedoch kein zweites Mal verwendet werden, da kein `<Label>` vergeben wurde.

Je nach Wunsch des Autors kann das Abkürzungsverzeichnis getrennt oder mit im Glossar verwaltet werden. Dies wird über die Paketoption `acronym=true` geregelt. Es kann prinzipiell die Taktik verfolgt werden, nur das erste Vorkommen im Text zu kennzeichnen oder alle<sup>10</sup>. Es empfiehlt sich alle weiteren Vorkommen zu kennzeichnen, so kann der Leser auch später zur Erklärung gelangen. Für weitere Erläuterungen ist die Dokumentation des Paketes [Tal06] zu Rate zu ziehen.

Zur Erzeugung des Verzeichnisses werden zwei Übersetzungen sowie der Aufruf von `MakeIndex` benötigt. Im ersten Lauf werden alle Glossareinträge in der `*.glo`-Datei gesammelt. Dann erfolgt der `MakeIndex`-Aufruf (es wird von einem Dokument mit dem Namen `test.tex` ausgegangen):

```
1 makeindex -s test.ist -t test.glg -o test.gls test.glo
```

Das Programm erzeugt die `*.gls`-Datei, welche die sortierten und formatierten Einträge des Glossars (unseres Abkürzungsverzeichnisses) enthält. Für dessen Einbindung benötigt es eine weitere  $\LaTeX$ -Übersetzung

<sup>10</sup>Die Funktion „Suchen & Ersetzen“ kann hier für die Schlussüberarbeitung hilfreiche Dienste erweisen.

**Hinweis 4** Wird das `xspace`-Paket vor `Glossary` eingebunden, so wird es automatisch genutzt und man braucht sich über Leerzeichen nach Abkürzungen keine Gedanken machen.

### 3.3.2 Das `babel`-Paket

**Anforderung 5** Sprachliche Besonderheiten wie Wechseln der Sprache sind erkennbar zu machen.

Wenn Sie Texte in einer bestimmten Sprache verfassen wollen, so sollten Sie  $\LaTeX$  dieses mitteilen. Entsprechende Befehle werden durch das Babel-System bereitgestellt. Es ist ein Paket für  $\LaTeX$ , das die Anpassung an viele Sprachen bietet. Es wird seit 1992 von Johannes L. Braams und dem  $\LaTeX$ -Team entwickelt und ist Bestandteil der offiziellen  $\LaTeX$ -Verteilung. Es unterstützt über 40 Sprachen (z.B. `english` und `ngerman` (neue deutsche Rechtschreibung)), teilweise mit mehreren Dialekten.

Die Sprachdateien beinhalten eine ganze Reihe von Regeln und Definitionen, die speziell auf den jeweiligen Sprachraum abgestimmt sind. Es werden Eigenschaften, wie korrekte Silbentrennung, automatische Kategorieüberschriften, Datumsformat, Umlaute, Anführungszeichen, Ligaturen u.v.a.m. festgelegt.

#### Hauptsprache des Dokumentes

Man lädt das Paket mit `\usepackage[<Sprache, Sprache>]{babel}` ein. In der Präambel müssen alle im Dokument enthaltenden Sprachen gelistet werden. Die Letzte wird als Hauptsprache für das Dokument ausgewählt. Sie kann aber mittels `\selectlanguage{english}` auch explizit gesetzt werden. Dieser allgemeine Schalter, ändert die Sprache bis zum Dokumentende bzw. bis zur nächsten Änderung.

#### Sprachwechsel

Der Befehl `\foreignlanguage{<Sprache>}{<Text>}`, ändert die Sprache nur für den geklammerten Bereich und sollte nur für einzelne Wörter und kurze Bereiche<sup>11</sup> verwendet werden. Längere Abschnitte können durch `\begin{otherlanguage}{<Sprache>}` und `\end{otherlanguage}` markiert werden. Weitere Informationen finden Sie unter [Kop00b, S. 53-65]. Erläuterungen zur Sprachumschaltungen und ihren Auswirkungen sind darüber hinaus in [Kop00a, S. 343] nachzulesen.

### 3.4 Technologische Robustheit

Ein weiteres wichtiges Prinzip ist, das der robusten Technologien. Ihre Verwendung soll es ermöglichen, Informationen möglichst unbegrenzt in heutigen und zukünftigen Ein- und Ausgabemedien wiederzugeben.

<sup>11</sup>Es ist auf eine korrekte Schachtelung zu achten.

### 3.4.1 Metadaten

Metadaten können über das neu entwickelte Paket `metadata` eingebunden werden. Die 15 Elemente des Dublin Core können über Wertepaare beim Laden des Pakets mit angegeben oder nachträglich über `\setkeys{<Schlüssel1=Wert1, Schlüssel2=Wert2,...>}` spezifiziert werden. Darüber hinaus stellt das Paket einen eigener Befehl zur Verfügung. Mit `\setmetadata{<Schlüssel>}{<Wert>}` kann die Zuordnung erfolgen. Die zur Verfügung stehenden Schlüssel sind in der [Tabelle 3.1](#) angegeben.

Name	Inhalt
<i>Title</i>	Der Name der Ressource
<i>Creator</i>	Verantwortlicher für den Inhalt
<i>Subject</i>	Thema der Ressource
<i>Description</i>	Beschreibung des Inhalts
<i>Publisher</i>	Verantwortlicher für die Veröffentlichung
<i>Contributor</i>	Zur Ressource Beitragende
<i>Date</i>	Datum eines Ereignisses im Lebenszyklus der Ressource
<i>Type</i>	Beschaffenheit oder Genre des Inhalts
<i>Format</i>	Festlegung über die physische oder digitale Erscheinungsform der Ressource
<i>Identifier</i>	Eindeutige Referenz zur Identifikation der Ressource
<i>Source</i>	Referenz zu der Ressource, von der die gegenwärtige abgeleitet ist
<i>Language</i>	Verwendete Sprache der Ressource
<i>Relation</i>	Referenz zu Ressourcen, die mit der gegenwärtigen in Verbindung stehen
<i>Coverage</i>	Ausmaß oder Bereich der Ressource
<i>Rights</i>	Informationen zu den Rechten innerhalb oder über die Ressource

Tabelle 3.1: Metadatenelemente des Dublin Core (vgl. [\[Dub04\]](#))

### 3.4.2 Eigene Erweiterungen

Die Erstellung von Makros, also eigenen  $\text{\LaTeX}$ -Befehlen, ist in vieler Hinsicht nützlich. Zum Einen können Sie kleine Makros als Abkürzung für lange Befehle schreiben, zum Anderen können Sie eine umfangreiche Eingabe mehrerer Befehle auf einen einzigen Befehl reduzieren. Sie haben dabei sogar die Möglichkeit der Parameterübergabe und der Verwendung von optionalen Argumenten. Die Angabe des Makroinhaltes kann so-

wohl aus Text, als auch aus  $\LaTeX$ -Befehlen bestehen. Die zu wählenden Namen sollten möglichst selbsterklärend sein.

Insbesondere für häufig wiederkehrende Strukturen, bieten sich Makros folglich an. Im Verlauf dieser Arbeit werden an verschiedenen Stellen auf die Namen von Paketen verwandt. Dazu wurde am Anfang ein neues Makro mit dem Namen `\packname{<Paketname>}` definiert, mit dem jeder `<Paketname>` der im Verlauf des Dokumentes auftritt ausgezeichnet wurde. Durch die Verwendung eines Makros kann im Dokument ganz leicht – je nach Vorgabe oder Belieben – die Auszeichnung von *unterstrichen* auf beispielsweise *kursiv* gewechselt werden, wodurch der Grad der Wiederverwendung von Dokumenten und Makros erhöht wird (im Gegensatz zur direkten Codierung der Schriftauszeichnung). Zudem wäre es künftig möglich die enthaltene logische Strukturinformation bei der Transformation zu erhalten.

Makros sind in jedem Fall zu testen, insbesondere in Bezug auf die erzeugte Struktur im Ergebnisdokument.

### Änderung des Schriftstils

Globale Schrifteinstellungen und Layoutänderungen sind möglichst im Header zu deklarieren. Befehle zur lokalen Veränderung des Schriftstils (wie `\textbf{<Text>}`, `\textit{}`, `\textsl{}` ...) sind im Dokument generell zu vermeiden, erst recht die plain- $\TeX$ -Varianten (wie `\bf` etc.). Statt dessen sollte auf logisches Markup wie `\emph{}` zurückgegriffen werden. Zum Zwecke der Definition eigener Makros sollte hingegen eher auf die plain- $\TeX$ -Varianten zugegriffen werden. Sie werden bisher nicht in Strukturinformationen überführt. Damit geht zwar ihre Hervorhebung verloren, sie werden aber auch nicht x-fach ausgezeichnet, nur weil der Text farbig hinterlegt, fett und kursiv dargestellt wird. Angestrebtes Ziel sollte es hingegen sein, die logischen Informationen zu erhalten. Die nötige Schnittstelle besteht bisher leider nicht (vgl. [Sch07]).

### 3.4.3 Restriktionen

Neben explizit erwähnten Erweiterungspaketen existieren zusätzliche, die aufgrund ihrer Reichhaltigkeit und Vielfalt nicht betrachtet werden konnten. Da sich  $\LaTeX$  in den letzten Jahren immer weiterentwickelt hat sind einige Pakete inzwischen nicht mehr aktuell oder basieren auf der alten  $\LaTeX$ -Version 2.09. Deshalb wird von deren Verwendung abgeraten (vgl. [Tre04]). Die im folgenden aufgelisteten Pakete sind veraltet, der Gebrauch ist nicht zukunftssicher und kann problematisch sein und zu Fehlern führen.

- a4
- a4wide
- ae
- bbm
- bezier
- bframe
- boxedminipage
- caption2
- doublespace
- epsf
- euler
- fancyheading
- fleqn
- fraktur
- graphics
- ilineno
- isolatin
- isolatin1
- itemrule
- leqno
- mathpple

- mathptm
- mlineno
- numquote
- olabels
- openbib
- palatino
- paper
- pgonek
- pslatex
- pstricks
- rlineno
- t1enc
- thema
- times
- ulem
- umlaut
- umlaute

Vor der Verwendung jeglicher Pakete sollte jeweils die zugehörige Dokumentation sowie „Das L<sup>A</sup>T<sub>E</sub>X-Sündenregister“ [Tre04] zwecks Neuerungen konsultiert werden

## 4 Die Transformationen

Da  $\LaTeX$  nur selten das Format der Publikation ist, werden in diesem Kapitel die Möglichkeiten der anschließenden Transformation betrachtet. Im Internet haben sich für Veröffentlichungen mehrheitlich die Formate XHTML und PDF herauskristallisiert. Aus  $\LaTeX$  lassen sich zusätzlich auch PostScript und DVI (Device Independent) Dateien generieren, die zugrunde liegenden Spezifikationen sehen jedoch keine speziellen Vorrichtungen zur Barrierefreiheit vor und sind damit zur Erzeugung barrierefreier Dokumente unbrauchbar.

### 4.1 tex4ht

Das Paket TEX4ht ist eine Möglichkeit die Vorteile des digitalen Mediums WWW, für die Publizierung von  $\LaTeX$ -Dokumenten, zu nutzen. Es erzeugt aus einem  $\TeX$ -Dokument (X)HTML Seiten, die mit Hilfe von Links und Navigationsleisten durchsucht werden können.

Damit ein XHTML-Dokument aus einer  $\TeX$ -Datei erzeugt werden kann, muss das Paket über `\usepackage{tex4ht}` eingebunden sein. Um nun die Konvertierung zu starten wird folgender Befehl `ht latex <Dateiname>.tex` in der Kommandozeile eingegeben.

Es ist auch möglich, das Paket nicht einzubinden und stattdessen eine Konfigurationsdatei anzulegen. Diese wird dann beim Aufruf von tex4ht mit angegeben. Sie ermöglicht eine umfassendere Konfiguration zusätzlich zur Beeinflussung über die Parameter des Paketes. Der komplette Aufruf lautet dann `ht latex <Dateiname>.tex <Konfigurationsdateiname>`.

Der Aufruf von Tex4HT ist im TexnicCenter standardmäßig leider nicht als Ausgabeprofil enthalten. Um den Aufruf über die Kommandozeile zu umgehen, kann man ein zusätzliches Ausgabeprofil definieren. Über „Ausgabe“ – „Ausgabeprofil definieren...“ wird ein Dialogfenster geöffnet, indem die weiteren Einstellungen getätigt werden können. Es bietet sich an, das Profil „LaTeX => DVI“ zu kopieren und umzubenennen. Anschließend muß auf der 2. Karteikarte ein neuer Postprozessor eingerichtet werden. Ihn nennt man beispielsweise „tex4ht“, als Anwendung ist „xhtmlatex.exe“ im Installationspfad im Unterverzeichnis „miktex\bin“ anzugeben (z. B. „C:\Programme\MiKTeX 2.5\miktex\bin\xhtmlatex.exe“). Die notwendigen Argumente sind „%tm“ "*<Name der Konfigurationsdatei>*".



Zur Beeinflussung der Transformation kann zwischen verschiedene Optionen ausgewählt werden. Zunächst sollte XHTML (`xhtml`) als der zu erzeugende XML-Dialekt ausgewählt werden. Wird XHTML anstelle von HTML erzeugt, so wird der neuere Standard erzeugt und die generierte Datei kann besser weiterverarbeitet werden. Als Kodierungsstandard sei die UTF-8-Kodierung (`utf-8`) empfohlen.

Wurden mit  $\LaTeX$  viele Formeln gesetzt, so sollten diese in MATHML umgewandelt werden. Die nötige Option heißt `mathml`. Wird MATHML vom verwendeten Browser bereits unterstützt, bietet diese Variante bessere Möglichkeiten der Skalierung und ist auch für assistive Technologien besser zugänglich, als vergleichbare Umsetzungen über Bilder.

Zur Kompatibilität mit dem Betriebssystem Microsoft Dos, ist es möglich, dass beim Erzeugen der `xhtml`-Dateien der Dateiname auf 8 Zeichen und die Endung auf 3 Zeichen (`.htm`), zu beschränken. Hierzu ist die Option `html` durch die Option `htm` zu ersetzen.

Über die hierarchische Tiefe wird die Aufteilung des Inhaltes auf einzelne Unterseiten bestimmt. Eine Tiefe von 1 würde z. B. für jedes Kapitel eine neue Seite erzeugen. `TeX4ht` kann eine hierarchische Tiefe bis zur vierten Ebene erzeugen. Standardmäßig wird keine Aufteilung vorgenommen und der gesamte Inhalt erscheint auf einer Seite. Die Unterteilung sollte in Abhängigkeit der Länge und Gliederung des Dokumentes gewählt werden.

Wurde eine Unterteilung gewählt, bietet es sich an, die Darstellung mittels Frames (`frames`) zu wählen. Sie erlaubt die Ansicht des Inhaltsverzeichnisses parallel zum Text des Dokumentes. Dadurch erhöht sich die Bedienbarkeit des Dokumentes und wenn der Leser etwas sucht muss er nicht immer zum Inhaltsverzeichnis zurückspringen.

Wird eine einseitig Darstellung gewünscht, bietet die Option `section+` die Möglichkeit, eine zusätzlich Rückwärtsverlinkung der Überschriften der Abschnitte zum Inhaltsverzeichnis zu erzeugen.

`Tex4HT` versucht die im  $\LaTeX$ -Dokument gewählten Schriftarten auch im XHTML-Dokument umzusetzen, wenn die Option `Fonts` gewählt ist. Das heißt, Schriftformatierungen werden durch `<Span>`-Tags umgesetzt. Die nötige Schriftart muss allerdings beim Leser vorhanden sein. Das ist für die  $\LaTeX$ -Schriften jedoch recht unwahrscheinlich. Es sollte daher immer die Option `NoFonts` gewählt werden,

Die Optionen werden entweder beim Laden des Paketes mit angegeben oder in der Konfigurationsdatei über den `\Preamble`-Befehl gesetzt.

Bei der Erzeugung mehrerer Unterseiten wird automatisch eine Navigationsleiste eingefügt. Die die meist sechs Schaltflächen (`next`, `prev`, `prevtail`, `front`, `tail`, `up`) beinhaltet. Diese Benennung ist für den deutschen Leser nicht besonders intuitiv und sollte daher verändert werden.

```
\Configure{crosslinks}{[]}{[]}{next}{prev}{prevtail}{front}{tail}{up}
```

Der Befehl kann entweder zu Beginn des Dokumentes oder ebenfalls in der Konfigurationsdatei angegeben werden. Als mögliche Belegungen werden „Nächste Seite“, „Letzte Seite“, „Letztes Seitenende“, „Seitenanfang“, „Seitenende“ und „Hoch“ empfohlen. Die eckigen Klammern, dienen als Begrenzer der Navigationsfelder, sie können ausgetauscht werden. Zur Trennung der Hyperlinks sollte aber mindestens ein druckbares Zeichen dienen.

Bei der Einbindung jeder Grafik muss die alternative Beschreibung mit Hilfe des Befehls `\Configure{GraphicsAlt}{<Der Text der Beschreibung>}` spezifiziert werden, dies sollte jeweils vor der Grafik geschehen. Durch die Zeile `\newcommand{\alt}[1]{\Configure{GraphicsAlt}{#1}}` im Deklarationsteil der Datei, kann die Definition der Alt-Tags auf die im Hauptteil dieser Arbeit beschriebene Weise erfolgen (vgl. [Unterabschnitt 3.1.2](#)).

Werden diese Hinweise eingehalten, kann die Accessibility der generierten Dokumente gesteigert werden. Leider lassen sich nicht alle Barrieren so einfach beseitigen. Fortgeschrittene  $\LaTeX$ - bzw.  $\text{TeX4HT}$ -Nutzer finden in der parallel entstandenen Diplomarbeit [Sch07] weitere Verbesserungshinweise. Sollten die dort empfohlenen Dinge in einer weiter benutzbaren Konfigurationsdatei implementiert sein, so kann diese zukünftig übers Netz verteilt und auch von den Autoren benutzt werden. Bisher fehlt es leider an einer solchen vollständigen Konfigurationsdatei.

## 4.2 pdfTeX

Um das Dokument mit pdf $\text{\TeX}$  zu erzeugen, muss man `pdftex` und eventuell noch `bibtex` bzw. `makeindex` aufrufen, je nachdem welche Zusatzprogramme noch benötigt werden.

```
1 pdflatex <Dateiname >
2 bibtex <Dateiname >
3 pdflatex <Dateiname >
```

In den verfügbaren Autorenwerkzeugen existieren meist Schaltflächen in den Symbolleisten, die diesen Kommandozeilenaufruf vereinfachen und beschleunigen. Im Texnic-Center ist das Ausgabeprofil „ $\text{LaTeX} \Rightarrow \text{PDF}$ “ zu wählen und anschließend der Button „Aktives Dokument erstellen und betrachten“ zu wählen.

Ist das Dokument nach den in [Kapitel 3](#) vorgestellten Grundsätzen gestaltet, muss das `accessibility`-Paket nur noch in die Dokumentenpräambel eingebunden werden. Grundsätzlich gibt es zwei Parameter, zum Einen kann zwischen den Tiefen der Schachtelung (`highstructure` oder `flatstructure`) und zum Anderen zwischen den Arten der Tag-Erzeugung (`tagged` oder `untagged`) gewählt werden.

Weitere Einflussmöglichkeiten offeriert das `hyperref`-Paket. Sie sollten so beeinflusst werden, dass Lesezeichen (`bookmarks`) generiert, sowie Fußnoten und Index verlinkt werden. Die alleinige Auszeichnung der Links durch Farbe ist jedoch kritisch (vgl. [Unterabschnitt 3.1.3](#)). Die Menüleisten sollten nicht verändert werden.

Die Implementierung ist bisher nur prototypisch, das erzeugte Dokument muss also unbedingt bezüglich der eingebundenen Struktur und deren Fehlerfreiheit überprüft werden. Es hat sich als praktisch erwiesen, die Generierung regelmäßig zu testen und möglichst wenig Zusatzpakete zu verwenden.

## 5 Überprüfung der Barrierefreiheit von PDF

Eine Voraussetzung für die Barrierefreiheit von PDF ist das Vorhandensein so genannter Tag. Die PDF-Tags (vgl. [Abbildung 5.1](#)) sind mit denen von XHTML vergleichbar. Sie strukturieren den Inhalt eines PDF-Dokumentes und erhöhen damit die Barrierefreiheit. Tags haben keinen Einfluss auf die Darstellung des Dokumentes. Sie spezifizieren die Lesereihenfolge oder knüpfen die Verbindung einer Grafik zu seiner Bildunterschrift.

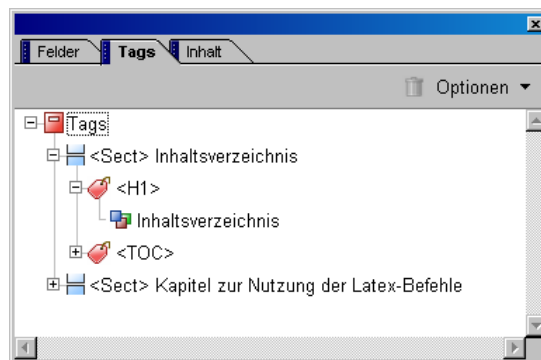


Abbildung 5.1: Tags im Adobe Acrobat Professional

### 5.1 Überprüfen, ob das PDF-Dokument mit Tags versehen ist.

Die einfachste Methode herauszufinden, ob eine PDF-Dokument bereits mit Tags versehen wurde, besteht darin, den Status in den Dokumenteigenschaften zu überprüfen.

1. Öffnen Sie die Datei im Acrobat Reader.
2. Wählen Sie „Datei“ → „Dokumenteigenschaften“.
3. Wenn auf der Registerkarte „Beschreibung“ neben dem Eintrag „PDF mit Tags“ ein „Nein“ angezeigt wird, bedeutet dies, dass die Datei nicht mit Tags versehen ist. Wird „Ja“ angezeigt, so ist die Datei mit Tags versehen.

### 5.2 Überprüfen der Tag-Struktur

Im Adobe Professional<sup>1</sup> können Sie sich die Tags eines Dokumentes anzeigen lassen, indem Sie den Tag-Baum öffnen.

1. Öffnen Sie das mit Tags versehene PDF-Dokument.

<sup>1</sup>Diese Funktion steht im Acrobat Reader nicht zur Verfügung.

2. Wählen Sie „Anzeige“ → „Navigationsregisterkarten“ → „Tags“. Dadurch wird die Registerkarte „Tags“ als verschiebbares Fenster geöffnet.
3. Optional: Unter „Optionen“ können Sie sich den zugehörigen „Inhalt markieren“ lassen.
4. Klicken Sie auf die Plus-Zeichen, um die hierarchische Tag-Liste zu erweitern.

Die Informationen sind in einer hierarchischen Struktur angeordnet, die der Oberfläche bekannter Dateimanager ähnelt. In den spitzen Klammern (<>) ist der Typ des Tag-Elementes angegeben.

Zu beachten ist bei der Überprüfung der Tag-Struktur insbesondere die korrekte Schachtelung. Speziell Seitenumbrüche können bisher nicht immer korrekt aus den  $\text{\LaTeX}$ -Dokumenten übernommen werden, was eine manuelle Kontrolle nötig macht. Gerade diese Fehler wirken sich häufig auf nachfolgende Strukturen aus.

### 5.3 Weitere Testmöglichkeiten

*Vollständige Ausgabehilfeprüfung (nur Acrobat Professional).* Verwenden Sie dieses Hilfsmittel, um Abbildungen mit fehlendem Alternativtext und andere Probleme zu finden. Bei der vollständigen Prüfung wird rein detaillierter Bericht über alle festgestellten Probleme generiert, der auch Vorschläge zur Fehlerbehebung enthält.

*Anzeigemodus „Umfließen“.* Wäre eine gute Testmöglichkeit für die Lesereihenfolge und Kennzeichnung von Artefakten, funktioniert aber bisher aufgrund von Problemen im Algorithmus (vgl. [Sch07]) nicht.

Das Vorhandensein von alternativen Beschreibungen, Langformen für Akronyme und Abkürzungen sowie Alternativtext für Verknüpfungen kann manuell, durch die Eigenschaften eines Tag-Elementes im Tag-Baum überprüft werden. Zum Eigenschaftendialog (vgl. [Abbildung 5.2](#)) gelangt man, indem man ein bestimmtes Tag-Element im Strukturbaum auswählt und über das Menü der linken Maustaste den Eintrag „Eigenschaften...“ wählt.

Durch das *Verwenden der Sprachausgabe* kann sowohl die Lesereihenfolge als auch die Angaben von Alternativtexten überprüft werden. Wählen Sie zum Aktivieren dieser Funktion „Anzeige“ → „Sprachausgabe“. Weitere Informationen sind in der Acrobat-Hilfe unter „Sprachausgabefunktion verwenden“ zu finden.

Durch das *Speichern des Dokumentes als zugänglicher Text* kann die Lesereihenfolge visuell mit einem einfachen Texteditor überprüft werden.

Ob die *Metadaten in das PDF-Dokument* übernommen wurden, kann unter „Datei“ → „Dokumenteneigenschaften“ überprüft werden. Hier kann auch die eingestellte Dokumentensprache überprüft werden.

Wichtig ist, dass auch im Nachhinein keine Festlegungen zu den Sicherheitseinstellungen getroffen werden, die die Barrierefreiheit des Dokumentes beeinträchtigen. Wenn der

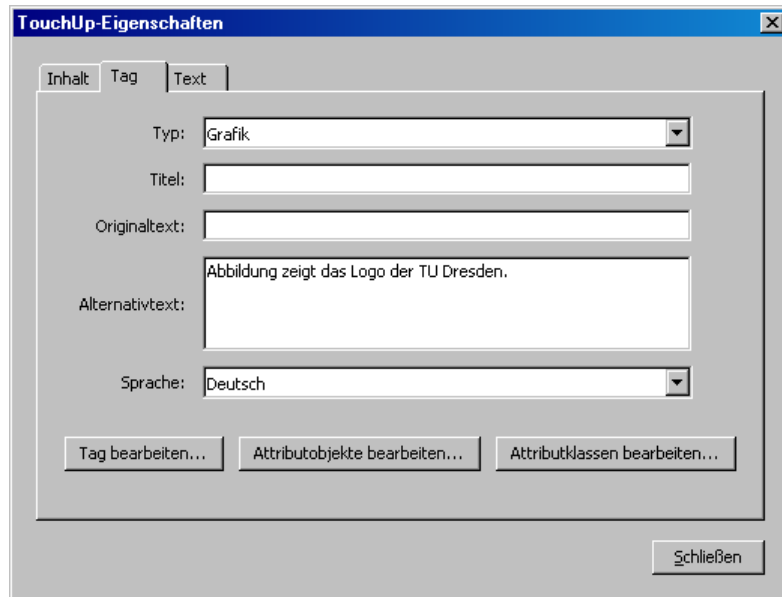


Abbildung 5.2: Eigenschaftendialog eines Tag-Elementes

Autor eines Dokumentes die PDF-Sicherheitseinstellungen ungeschickt einsetzt, kann er unbeabsichtigt verhindern, dass sich der Acrobat Reader und der Screenreader verständigen können, das heißt das Dokument kann nicht gelesen werden. Eine Einstellung, die sowohl die Schutzbedürfnisse des Urhebers, als auch die Ansprüche eines Screenreaders berücksichtigt, erlaubt das „Drucken“, „Kopieren bzw. Entnehmen von Inhalten“ und 'Inhalt für Zugriff entnehmen'. Alle anderen Funktionen können eingeschränkt werden.

## 5.4 Nachbearbeitung PDF

Die Nachbearbeitung von PDF gestaltet sich aufgrund fehlender Werkzeuge als schwierig. Sie ist bisher nur mit Hilfe des kostenpflichtigen Werkzeugs „Adobe Professional“ möglich. Selbst unter Verwendung dieser Software ist das Erstellen von Tags per Hand alles andere als benutzerfreundlich. Die einzelnen Schritte sind bisher unhandlich und uneffektiv. Durch das Fehlen eine Funktion zum „Rückgängig“ machen, muss die Arbeit häufig von vorn begonnen werden. Es empfiehlt sich ein häufiges Zwischenspeichern der aktuellen Version.

Wie aus reinen PDF solche mit Tags werden, beschreiben zahlreiche Tutorials. Sie gehen fast immer von der Erstellung mit MS Word oder OpenOffice aus, die Schritte der Nachbearbeitung gestalten sich jedoch analog. Informationen, wie z. B. aus Office-Dokumenten barrierefreie PDF-Dokumente werden können, sind im Internet-Angebot

von Adobe<sup>2</sup> oder über die Linksammlung im Informationsportal<sup>3</sup> des Aktionsbündnisses für barrierefreie Informationstechnik zu finden. [Pro05]

---

<sup>2</sup><http://www.adobe.de/epaper/tips/acr5access/pdfs/AccessGuide.pdf>

<sup>3</sup><http://www.wob11.de/links/anleitungen.html>

## Literaturverzeichnis

- [Com06] COMMITTEE OF THE UK TeX USERS GROUP: *The UK TeX FAQ – Your 408 Questions Answered*, Oktober 2006.
- [Dub04] DUBLIN CORE METADATA INITIATIVE: *Dublin Core Metadata Element Set, Version 1.1: Reference Description*, Dezember 2004.
- [Jür00] JÜRGENS, MANUELA: *LaTeX – eine Einführung und ein bisschen mehr...*, März 2000.
- [KM06] KOHM, MARKUS und JENS-UWE MORAWSKI: *KOMA-Script – Die Anleitung*, Juli 2006.
- [Kop00a] KOPKA, HELMUT: *Band 1: Einführung*. LaTeX. Addison-Wesley Verlag, 3. Auflage, 2000. ISBN 3-8273-7038-8.
- [Kop00b] KOPKA, HELMUT: *Band 2: Ergänzungen*. LaTeX. Addison-Wesley Verlag, 3. Auflage, 2000. ISBN 3-8273-7038-8.
- [Pro05] PROJEKTGRUPPE E-GOVERNMENT IM BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK: *Barrierefreies E-Government – Leitfaden für Entscheidungsträger, Grafiker und Programmierer*, 2005.
- [Rei99] REICHERT, AXEL: *Satz von Tabellen*, Februar 1999.
- [RO04] RAHTZ, SEBASTIAN und HEIKO OBERDIEK: *Hypertext marks in LaTeX: a manual for hyperref*, Februar 2004.
- [Sch07] SCHALITZ, BABETT: *Accessibility-Erhöhung in LaTeX-Dokumenten*. Diplomarbeit, Technische Universität Dresden, Informatik Fakultät, April 2007.
- [Tal06] TALBOT, NICOLA L.C.: *LaTeX<sub>2 $\epsilon$</sub>  Package to Assist Generating Glossaries*, Juli 2006.
- [Tre04] TRETTIN, MARK: *Das LaTeX<sub>2 $\epsilon$</sub> - Sündenregister oder Veraltete Befehle, Pakete und andere Fehler*, Dez 2004.



## Abbildungsverzeichnis

2.1	Beispiel für die Verwendung von Gliederungsebenen . . . . .	11
2.2	Beispiel für die Verwendung eines Zitates . . . . .	12
2.3	Beispiel für die Verwendung eines Codes . . . . .	12
2.4	Beispiel für die Verwendung einer Formel . . . . .	13
2.5	Beispiel für die Verwendung einer Aufzählung . . . . .	13
2.6	Beispiel für die Verwendung eines Theorems . . . . .	14
2.7	Beispiel für die Verwendung einer Fußnote . . . . .	15
2.8	Beispiel für die Verwendung einer Tabelle . . . . .	17
2.9	Beispiel für die Verwendung einer Tabelle . . . . .	17
3.1	Beispiel für die Verwendung einer Referenz . . . . .	23
5.1	Tags im Adobe Acrobat Professional . . . . .	36
5.2	Eigenschaftendialog eines Tag-Elementes . . . . .	38

## Tabellenverzeichnis

2.1	Entsprechungen der Standard- $\LaTeX$ -Klassen . . . . .	9
2.2	Standardgliederungsbefehle . . . . .	11
3.1	Metadatenelemente des Dublin Core . . . . .	29