

**HOCHSCHULE
HANNOVER**
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

–
*Fakultät IV
Wirtschaft und
Informatik*

Wissenschaftliche Arbeiten mit \LaTeX

Prof. Dr. Frauke Sprengel
Abteilung Informatik
Fakultät IV – Wirtschaft und Informatik

Winter 2013



Inhaltsverzeichnis

1. Überblick	1
1.1. Zu Erwartendes	1
1.2. Ziele der Veranstaltung	2
1.2.1. Ihre Ziele	2
1.3. Literatur	2
1.3.1. Bücher	2
1.3.2. Websites und andere Quellen	3
1.4. Nützliche Webseiten	4
1.5. Organisatorisches	5
2. Einführung in das wissenschaftliche Arbeiten	7
2.1. Wissenschaftlich arbeiten / Lesen	7
2.1.1. Notwendigkeit des Lesens	7
2.1.2. Folgerungen für Sie	7
2.1.3. Qualitätskriterien für Materialien	8
2.1.4. Lehrmaterialien	8
2.1.5. Durcharbeiten von Materialien mit PQ4R	9
2.2. Wissenschaftlich arbeiten / Schreiben	10
2.2.1. Notwendigkeit des Schreibens	10
2.2.2. Erstellen von Dokumenten	10
2.2.3. Erstellen von Dokumenten - jetzt aber richtig	12
2.2.4. The Pyramid Principle	13
2.2.5. Gute Dokumente schreiben lernen	15
2.3. Literatur finden und verwalten	15
2.3.1. Aktueller Stand	15
2.3.2. Finden von Materialien	16
2.3.3. Bibliothek	17
2.3.4. Verfügbarer Online Content	17
2.3.5. Werkzeuge zur Recherche und Literaturverwaltung	18
2.4. Literatur und Zusammenfassung	19
2.4.1. Literaturhinweise	19
2.4.2. Zusammenfassung	19
3. Dokumentenverarbeitung	21
3.1. Dokumentmodelle	21
3.2. Text und seine Eigenschaften	23
3.3. Dokumentverarbeitung	25
3.3.1. Zeilen und Absätze	25
3.3.2. Seiten und Satzspiegel	26

3.4.	Beispiel \LaTeX	27
3.4.1.	Besonderheiten von \LaTeX	27
3.4.2.	Erstes Beispiel	28
3.5.	Zusammenfassung	28
4.	\LaTeX-Grundlagen	29
4.1.	Von \TeX zu \LaTeX	29
4.2.	Arbeiten mit \LaTeX	31
4.2.1.	Grundsätzliches Vorgehen	31
4.2.2.	Hallo, \LaTeX !	32
4.2.3.	Befehle	32
4.2.4.	Umgebungen	34
4.2.5.	Kommentare	34
4.2.6.	Dokumentklassen	35
4.3.	Pakete einbinden	36
4.4.	Verwendung von Gliederungsebenen in wissenschaftlichen Arbeiten	36
4.4.1.	Hierarchie	36
4.4.2.	Tiefe der Hierarchie	36
4.4.3.	Breite der Hierarchie	37
4.4.4.	Finden der Hierarchie	37
4.4.5.	Hierarchien in \LaTeX	38
4.4.6.	Überschriften	38
4.5.	Nützliche Kleinigkeiten	38
4.5.1.	Schriftarten	38
4.5.2.	Schriftgrößen	39
4.6.	Texteingabe	40
4.6.1.	Eingabe von Absätzen	40
4.6.2.	Besonderheiten in Absätzen	40
4.6.3.	Zwischen Absätzen	40
4.7.	Zusammenfassung	41
5.	Mathematik, Verzeichnisse, Referenzen und Zitate	43
5.1.	Mathematik	43
5.1.1.	Formelarten	43
5.1.2.	Formelteile	44
5.1.3.	Weitere Befehle und Umgebungen	44
5.1.4.	Symbole	44
5.1.5.	Formelbeispiele	46
5.2.	Gleitobjekte in \LaTeX	47
5.2.1.	Idee	47
5.2.2.	Abbildungen	47
5.2.3.	Grafiken einbinden	47
5.2.4.	Tabellen	48
5.3.	Verzeichnisse	51
5.4.	Referenzen und Verweise	51
5.4.1.	Setzen und Benutzen von Referenzen	51
5.4.2.	Auflösung von Referenzen	52
5.4.3.	Schöne Referenzen	53
5.4.4.	Verweise	53

5.5.	Zitate und Bibliographie	53
5.5.1.	Zitate	54
5.5.2.	Zitierstile	54
5.5.3.	Bib \LaTeX	54
5.5.4.	Wörtliches Zitieren	55
5.5.5.	Nicht-wörtliches Zitieren	55
5.5.6.	Bib \TeX	56
5.6.	MakeIndex	59
5.7.	Zusammenfassung	60
6.	Nützliche Pakete	61
6.1.	Listings	61
6.2.	Ausgefeilte Kopf- und Fußzeilen	62
6.3.	Verschiedene Eingabecodierungen	64
6.4.	Pstricks und Tikz	64
6.4.1.	Pstricks	65
6.4.2.	TikZ/Pgfplots	66
6.5.	ModernCV	68
6.6.	Beamer	70
6.7.	Zusammenfassung	75
7.	Eigene Kommandos, Dokumentenstruktur und externe Programme	77
7.1.	Dokumentenstruktur	77
7.1.1.	Aufteilung von Dokumenten mit input und include	77
7.1.2.	Unterschiede zwischen include und input	78
7.1.3.	Unterschiedliche Behandlung von Einleitung, Hauptteil, Schluss	78
7.1.4.	Änderung der Seitennummerierung	79
7.2.	Zähler	79
7.3.	Eigene Kommandos und Umgebungen	80
7.3.1.	Eigene Kommandos	80
7.3.2.	Eigene Umgebungen	82
7.4.	Eigene Klassen und Packages	83
7.4.1.	Struktur einer Klasse oder eines Paketes	83
7.4.2.	Identifikation	83
7.4.3.	Laden von Klassen und Paketen	84
7.4.4.	Optionen	84
7.5.	(pdf) \LaTeX auf der Kommandozeile	85
7.5.1.	Optionen	85
7.5.2.	Ein Makefile	86
7.6.	Externe Programme	87
7.6.1.	Zeichenprogramme in Verbindung mit \LaTeX	87
7.6.2.	Programme für die PostScript und PDF-Bearbeitung	90
7.6.3.	Weitere Helferlein	92
7.7.	Zusammenfassung	92
A.	Übungen	A–1
A.1.	Wissenschaftliches Arbeiten	A–1
A.1.1.	Übungsaufgaben	A–1

A.2. \LaTeX – erste Schritte	A–3
A.2.1. Übungsaufgaben	A–3
A.3. \LaTeX – Erstes Dokument	A–5
A.3.1. Übungsaufgaben	A–5
A.4. Mathematik, Graphik und Tabellen	A–8
A.4.1. Übungsaufgaben	A–8
A.5. Literatur, Quelltexte und Präsentationen	A–10
A.5.1. Übungsaufgaben	A–10
A.6. Dokumentenstruktur und eigene Befehle	A–12
A.6.1. Übungsaufgaben	A–12
A.7. Freies Üben	A–14
A.7.1. Übungsaufgaben	A–14

1. Überblick

Inhalt

- 1.1. Zu Erwartendes
- 1.2. Ziele der Veranstaltung
- 1.3. Literatur
- 1.4. Nützliche Webseiten
- 1.5. Organisatorisches

Dieses Skript bildet die Grundlage für die Vorlesung „Wissenschaftliche Arbeiten mit L^AT_EX“ (Ergänzendes Fach) des Bachelor-Studienganges „Angewandte Informatik“ im Wintersemester 2013.

Die elektronische Form des Skripts sowie die Termine und Möglichkeit zum Hochladen von Aufgaben finden Sie unter

<https://loncapa.hs-hannover.de>

im zugehörigen Kurs.

Dank

Besonderer Dank gilt Prof. Dr. Stefan Wohlfeil sowie Mark Heisterkamp und Paul Cochra-ne, deren Folien wir fröhlich verwendet haben (und dies auch durften).

Die ersten Fassungen dieses Skripts entstanden in Zusammenarbeit mit Jan Prodöhl.

1.1. Zu Erwartendes

- Einführung ins wissenschaftliche Arbeiten
- Dokumentenverarbeitung allgemein
- L^AT_EX
 - Gliederung
 - Seitengestaltung
 - Tabellen und Abbildungen
 - Referenzen, Zitate, Links, Verzeichnisse

- Mathematiksatz
- Listings
- Graphiken in Latex
- Beamerpräsentationen
- Nützliche Zusatzprogramme

1.2. Ziele der Veranstaltung

1.2.1. Ihre Ziele

Frage (Ihre Wünsche und Ziele)

Bitte formulieren Sie Ihre Wünsche, Ziele oder Befürchtungen für diese Vorlesung! Sie haben 5 min Zeit. Anschließend sammeln wir an der Tafel.

1.3. Literatur

1.3.1. Bücher

Kottwitz, Stefan (2011).

LaTeX Beginner's Guide.

Birmingham, Mumbai: Packt Publishing.

Sturm, Thomas (2011).

LaTeX: Einführung in das Textsatzsystem.

8. Aufl.

Hannover: RRZN.

Schlosser, Joachim (2009).

Wissenschaftliche Arbeiten schreiben mit LaTeX: Leitfaden für Einsteiger.
mitp.

Mittelbach, Frank und Michel Goossens (2010).

Der LaTeX-Begleiter.

Pearson Studium.

Kohm, Markus und Jens U. Morawski (2008).

KOMA-Script – Die Anleitung.

Lehmanns Media.

Voß, Herbert (2009a).

Präsentationen mit LaTeX.

Lehmanns Media.

– (2010a).

PSTricks.

Lehmanns Media.

- (2010b).
L^AT_EX Referenz.
Lehmanns Media.
- (2009b).
Mathematiksatz mit L^AT_EX.
Lehmanns Media.
- Lingnau, Anselm (2007).
L^AT_EX Hacks: Tipps und Techniken für professionellen Textsatz.
O'Reilly.
- Knuth, Donald Ervin (1984).
The T_EXbook.
Addison-Wesley Longman.
- (1998).
The Art of Computer Programming.
Addison-Wesley Longman.
- Lamport, Leslie (1985).
The L^AT_EX Document Preparation System.
Addison Wesley.
- Deining, M., H. Lichter, J. Ludewig und K. Schneider (2002).
Studien-Arbeiten. Ein Leitfaden zur Vorbereitung, Durchführung und Betreuung von Studien-, Diplom- und Doktorarbeiten am Beispiel Informatik (Taschenbuch).
vdf Hochschulverlag AG an der ETH Zürich.
- Minto, Barbara (1987).
The Pyramid Principle – Logic in Writing and Thinking.
London: Pitman Publishing.
- Rechenberg, Peter (2002).
Technisches Schreiben (nicht nur) für Informatiker.
Hanser Verlag.
- Weber, Stefan (2007).
Das Google-Copy-Paste-Syndrom.
Heise Verlag.

1.3.2. Websites und andere Quellen

- Guttenplag (2011).
GuttenPlag Wiki.
URL: http://de.guttenplag.wikia.com/wiki/GuttenPlag_Wiki.
- Sprengel, Frauke und Stefan Wohlfeil (2012).
Wie schreibe ich eine Abschlussarbeit? Ein kurzer Abriss.
URL: </home/daten/skripte/abschlussarbeiten/Thesis-How-To.pdf>.

Nagel, Ulrich (2001).

Hinweise zur formalen Gestaltung einer Diplomarbeit.

URL: http://www-fh-mainz.de/fr_bauing/personal/nagel/material/hinweise-diplomarbeiten.pdf.

1.4. Nützliche Webseiten

- Dokumentation zu den Paketen erhält man mit `texdoc` + Paketname auf der Kommandozeile, falls das nicht funktioniert, ist die online Variante hier: <http://texdoc.net/>
- tug.org: TeX Users Group: alles Wichtige
- dante.de: Deutschsprachige Anwendervereinigung TeX e.V., dort auch zu finden:
- ctan.org: Comprehensive TeX Archive Network, (fast) alles, was es so an nützlichen und unnützen Programmen und Pakten rund um T_EX gibt
- [detexify](http://detexify.com): Symbole zeichnen und den L^AT_EX-Befehl geliefert bekommen
- für noch mehr Links und Bücher: de.wikipedia.org
- Nützliche Foren (nur zu L^AT_EX oder mit passenden Unterforen)
 - golatex.de
 - mrunix.de
 - matheplanet.com
- Beispiele und Blogs
 - <http://www.texample.net/>
 - <http://texblog.net/>
 - <http://www.latextemplates.com/de>

Nützliche Software findet man hier:

- Alles rund um T_EX
 - <http://www.ctan.org/>: Comprehensive TeX Archive Network, (fast) alles, was es so an nützlichen und unnützen Programmen und Pakten rund um T_EX gibt
 - <http://www.tug.org/texlive/> Beliebte T_EX-Distribution, aktuell: 2011, bei aktuellem ubuntu dabei: 2009
- Editoren
 - <http://www.tug.org/texworks/>: vergleichsweise einfacher Editor + Frontend (alle Plattformen, bei ubuntu dabei, bei texlive dabei)
 - <http://texlipse.sourceforge.net/> Texlipse Plugin für eclipse

- <http://borisvl.github.com/Pdf4Eclipse> einfacher PDF-Preview für Eclipse, erlaubt Vor- und Rückwärtssuche
- TexMaker(X) Editor+Frontend
- TexShop nur für Mac, Vorlage für TexWorks und TeXMaker, hat der L^AT_EX-Benutzung auf Macs angeblich sehr gut getan
- <http://www.winedt.com/>, mächtiger Shareware-Editor, nur Windows
- kile: guter Editor für Linuxe (KDE)
- Literaturverwaltung
 - <http://jabref.sourceforge.net/>: BibT_EX-Editor JabRef
 - <http://www.citavi.com/>: Literaturverwaltung Citavi
 - <http://www.mendeley.com/>: Literaturverwaltung und Zusammenarbeit Mendeley
 - <http://www.citeulike.org/>: Literaturverwaltung und Zusammenarbeit CiteULike

1.5. Organisatorisches

- Prof. Dr. Frauke Sprengel
Raum 349, Tel. 0511/9296-1812
frauke.sprengel@hs-hannover.de
- Skript: im Lon-Capa-Kurs
- Vorlesung und Übung: immer im Wechsel, beginnend mit dieser Vorlesung
- Prüfung:
 - 50% Abgabe von einigen Übungen in Lon-Capa zu zweit – bitte vermerken Sie Ihre Partner in der Abgabe (wird bepunktet)
 - 50% Klausur im Januar
 - Bitte Anmeldung (in QisPos) nicht vergessen!
- Zeit und Ort
 - erste Vorlesung 30.9.2013 16:00 in Raum 325
 - Vorlesung in 325
 - Übung in 314

2. Einführung in das wissenschaftliche Arbeiten

Inhalt

- 2.1. Wissenschaftlich arbeiten / Lesen
- 2.2. Wissenschaftlich arbeiten / Schreiben
- 2.3. Literatur finden und verwalten
- 2.4. Literatur und Zusammenfassung

2.1. Wissenschaftlich arbeiten / Lesen

2.1.1. Notwendigkeit des Lesens

- Textverstehendes Lesen von längeren Dokumenten ist die Grundlage allen wissenschaftlichen Arbeitens.
- Hierzu ist erforderlich:
 - Genug **Konzentrationsfähigkeit**, damit auch längere Arbeiten (> 3 Minuten) durchgehalten werden. Keine Ablenkungen.
 - Genug (Fremd-) **Sprachkompetenz**, um zu verstehen, was der Autor *sagt* bzw. *schreibt*.
 - Genug fachliches **Hintergrundwissen**, um zu verstehen, was die Autorin *meint* und wie man die Aussagen *bewerten* kann bzw. muss.

2.1.2. Folgerungen für Sie

- Schaffen Sie sich eine gute Arbeits-Atmosphäre, d.h. eine passende Umgebung, feste Zeiten, feste Lerngruppe, ...
- Bevorzugen Sie Original-Quellen, d.h.
 - Lesen Sie die Standards, Definitionen, ... im Original (keine schlechten/falschen Übersetzungen!)
 - Ziehen Sie gute Bücher hinzu. (Literaturhinweise geben alle Dozentinnen und Dozenten!)

- Lernen Sie genug Englisch, um sicher verstehen zu können und sich hinreichend ausdrücken (mündlich und schriftlich) zu können.

2.1.3. Qualitätskriterien für Materialien

Kriterien

- **Korrektheit:** Stimmt das, was behauptet wird?
- **Aktualität:** Ist das, was behauptet wird, noch aktuell? Gibt es neuere Erkenntnisse?
- **Relevanz:** Ist das, was behauptet wird, überhaupt interessant?

Frage (Anhaltspunkte)

Woran kann man erkennen, ob diese Kriterien erfüllt sind?

Anhaltspunkte für die Erfüllung der Kriterien

- Art des Materials (Buch, Aufsatz, Internet-Seiten, usw.)
- Datum der Veröffentlichung: Bei Standardwerken älter, bei Forschungen besser jünger.
- Renommé/Ruf der Autorin und des Verlegers/Publizierers
- Werden Qualitätssicherungsmethoden im Entstehungsprozess angewendet?
- Stichhaltiger Aufbau des Dokumentes: Sind die Behauptung überprüfbar? Werden Annahmen explizit genannt?
- Dauerhaftigkeit: Steht das Material auch morgen noch zur Verfügung? (Weblinks sind oft sehr vergänglich)

2.1.4. Lehrmaterialien

Klassische Lehrmaterialien

- Vorlesungs- und Übungsunterlagen (Skripte-Server)
 - **Vorsicht: Auf den Folien steht nicht immer alles, sondern sie sind nur verständlich, wenn man zusätzlich den Vortrag hört und aktiv mitdenkt!**
- Gute Bücher (werden von DozentInnen genannt)
 - kompetente/r Autor/in, Buch enthält Grundlagenwissen und allgemeine Prinzipien, evtl. auf englisch
 - “xyz for Dummies“ und andere “Klickanleitung“ besser vergessen oder nur für den Einstieg benutzen.
 - Lexika (wie Wikipedia) eignen sich für den **Einstieg** in ein Thema.

- Wissenschaftliche Aufsätze wie
 - Konferenzbeiträge in einschlägigen Konferenzen, die meistens von Fachorganisationen organisiert werden (GI, ACM, IEEE, usw.) Erscheinen i.d.R. jährlich.
 - Aufsätze in Fachzeitschriften (ACM, IEEE, usw.) Erscheinen i.d.R. monatlich.
- Jemand der sich damit auskennt, z.B.
 - DozentInnen,
 - KommilitonInnen höherer Semester, usw.

“Moderne“ Lehrmaterialien

- Spezifikationen, Tutorials, White Paper, FAQs, ...
 - i.d.R. auf englisch, aber dafür meist sehr gut.
- Suchmaschinen wie: Google
- Wikipedia
 - zweifelhafte Genauigkeit, unklare Kompetenz/Motive der Autoren, Vergänglichkeit der Artikel,
 - eingeschränkte Themenauswahl, eingeschränkte Quellenangaben.
 - *Nie Wikipedia als alleinige Quelle benutzen!*
- Diskussionsforen
 - Vorsicht, stark schwankende Qualität. Oft findet man “Laien helfen Ahnungslosen“-Foren.

2.1.5. Durcharbeiten von Materialien mit PQ4R

- **Preview:** Überblick verschaffen (Autor, Titel, Vorwort, Inhaltsverzeichnis, Abbildungen, Fazit) und “diagonal lesen“.
- **Questions:** Aufschreiben, welche Fragen man durch den Text beantwortet bekommen will.
- **Read:** Lesen und Fragen beantworten. Relevante Stellen anstreichen oder abschreiben.
- **Reflect:** Bezug zu Bekanntem herstellen, Argumente, Gegenargumente suchen, Beispiele, ...
- **Recite:** Versuchen, den Text mit eigenen Worten wieder zu geben.
- **Review:** Konnten alle Fragen geklärt werden, Kritik am Text formulieren
- Falls sich herausstellt, dass der Text die Erwartungen nicht erfüllt, dann abbrechen und das als Kritik zum Text festhalten.

- Sonst persönliche Zusammenfassung erstellen.

2.2. Wissenschaftlich arbeiten / Schreiben

2.2.1. Notwendigkeit des Schreibens

- Letztlich muss das Ergebnis der eigenen Arbeit (die eigenen Gedanken) jemandem mitgeteilt werden. Dazu ist meistens die schriftliche Form gefragt, sei es als
 - Bericht, Bachelorarbeit, Entscheidungsvorlage, usw.
 - Präsentationsfolien
- Wichtige Voraussetzungen:
 - Genug Sprachkompetenz, um sich verständlich auszudrücken.
 - Kenntnis der Fachbegriffe, so dass man diese richtig und einheitlich benutzt.

2.2.2. Erstellen von Dokumenten

Erstellen von Dokumenten - Vorgehen 1

- Schlüsselwörter der Aufgabenstellung bei google eintippen.
- Die erste Treffer-Seite oberflächlich durchsehen und die “schönsten“ Dokumente heraussuchen.
- Falls ein Dokument genau das ist was man sucht: *Hurra, kopieren, fertig!*
- Falls nicht, dann leider mehr Arbeit, nämlich:
 - Aus den gefundenen Dokumenten die Textbausteine um die Schlüsselwörter herum markieren und in die Zwischenablage kopieren.
 - Leeres Word-Dokument öffnen und Textbausteine einfügen.
- Redaktionelle Überarbeitung durchführen, d.h.:
 - Den eigenen Namen als Verfasser einfügen.
 - Evtl. Übersetzen von Fremdsprache nach Deutsch.
 - Füllsätze einfügen, damit die Übergänge glatter aussehen.
 - Formatierung anpassen und vereinheitlichen, evtl. den Vorgaben anpassen.

So nicht!!!

Bewertung dieses Erstellungsprozesses

Google ist nicht allwissend. Probleme sind:

- **Keine Treffer:** Es gibt Quellen, die stehen bei google gar nicht im Index.
- **Zu viele Treffer:** Bei bestimmten Suchbegriffen übersieht man interessante Hinweise, weil sie zu weit unten in der Liste stehen.
- Wikipedia taucht i.d.R. weit vorne in der Trefferliste auf.
- Fördert “Textkultur ohne Hirn“ (Weber 2007)
- Man erstellt ein **Plagiat**:
 - Die geistige Leistung eines Anderen wird als die eigene Leistung ausgegeben und/oder verwertet.
 - Salopp gesagt: abschreiben, kopieren. Geistiger Diebstahl!
- Plagiatstypen:
 - Total-Plagiat: Komplette Kopie
 - Teil-Plagiat: Kopieren größerer Abschnitte
 - Ideen-Plagiat: Übernehmen der Idee, mit Umformulierung (Paraphrasierung)
- **Der sichere Weg zur sechs!**

Folgen eines Plagiats

- Seminararbeit / Abschlussarbeit wird mit “nicht ausreichend” bewertet
- das niedersächsische Hochschulgesetz erlaubt sogar sofortige Exmatrikulation
- bei späterem Auffliegen: erworbener Hochschulgrad wird aberkannt (geht auch beim Bachelor)
- lassen Sie die Suche nach Plagiaten in Ihren Arbeiten also nicht so aussehen:

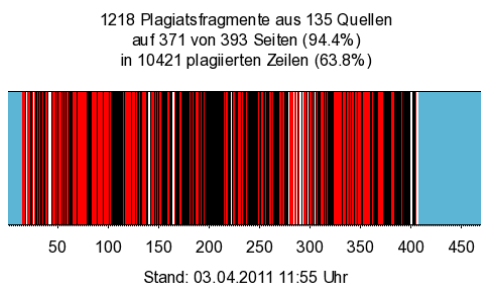


Abbildung 2.1.: Barcode aus dem GuttenPlagWiki, weiß: kein, schwarz: ein, rot: mehrere Funde nichtzitatierter Quellen (Guttenplag 2011).

2.2.3. Erstellen von Dokumenten - jetzt aber richtig

Immer von Null starten?

- Man muss nicht immer wieder bei Adam und Eva starten, wenn man etwas erstellt.
- Aufsetzen auf bekannten Ergebnissen ist gute Praxis in jeder Forschung. Es nicht zu tun, wäre Ressourcen-Verschwendung.
- Wenn man auf der Arbeit von Anderen aufsetzt, dann muss man Ross und Reiter nennen, d.h. korrekt zitieren!
 - Typographisch klarmachen, welche Stellen ein Zitat sind; entweder durch Anführungszeichen oder Einrücken von Absätzen.
 - Quellen direkt am Zitat nennen.
 - Literaturverzeichnis (Liste aller Quellen) erstellen.

Prozess der Dokumenterstellung

- Erstelle eine Gliederung des zu schreibenden Dokumentes
 - Was will man sagen?
 - In welcher Reihenfolge?
 - Wie detailliert?

Pyramiden-Prinzip anwenden!

- Fülle die einzelnen Abschnitte mit Inhalt
 - Fasse das, was zu sagen ist, in Worte.
 - Beachte dabei den richtigen Einsatz von Fachbegriffen, Grammatik, Rechtschreibung, Stil, usw.
- Immer wieder Korrekturlesen und verbessern.
 - Nicht zu kurze Intervalle wählen.

Orientierung an EmpfängerInnen

- Unbedingt an EmpfängerInnen der Informationen orientieren. Eingehen auf deren
 - Bedürfnisse (Wozu brauchen sie ihr Ergebnis?)
 - Wünsche (Was genau wollen sie wissen, wie detailliert, usw.?)
 - Hintergrundwissen (Was wissen die EmpfängerInnen schon?)
- Unterschiedliche Empfängertypen
 - Wissenschaftliche Ergebnisse für WissenschaftlerInnen

- Technische Ergebnisse für Vorgesetzte
- Wirtschaftliche Ergebnisse für ManagerInnen

2.2.4. The Pyramid Principle

Ideen

- Quelle: Barbara Minto: *The Pyramid Principle – Logic in Writing and Thinking*; also (Minto 1987)
- Ziel des Buches: Wie schreibt man klar und leicht verständliche (Geschäfts-) Dokumente.
- Ideen:
 - Trenne den Denkprozess vom Aufschreibprozess.
 - Finde die richtige (logische) Reihenfolge für die Gedanken und schreibe sie in dieser Reihenfolge auf.
 - Erkenne was Hauptaussagen und was Unteraussagen (sie unterstützen die Hauptaussagen) sind und in welcher Beziehung sie stehen (typischerweise hierarchisch = Pyramide)

Beispiele

Typisches schlechtes Beispiel:

“John Collins hat angerufen und sagt, dass er morgen um 15:00 nicht an der Besprechung teilnehmen kann. Hal Johnson sagt, dass er auch später kann, auch übermorgen, aber dann nicht vor 10:30. Don Cliffords Sekretärin sagt, dass Clifford nicht vor morgen aus Frankfurt zurück ist. Der Besprechungsraum ist morgen reserviert, aber donnerstags frei. Donnerstag um 11:00 scheint zu passen. Kannst du da auch?” (Minto 1987)

Frage

Welche Probleme gibt es mit diesem Dokument?

Bessere Alternative:

“Können wir die Besprechung von morgen auf Donnerstag um 11:00 verschieben? Das passt Collins und Johnson besser und außerdem könnte dann auch Clifford teilnehmen.” (Minto 1987)

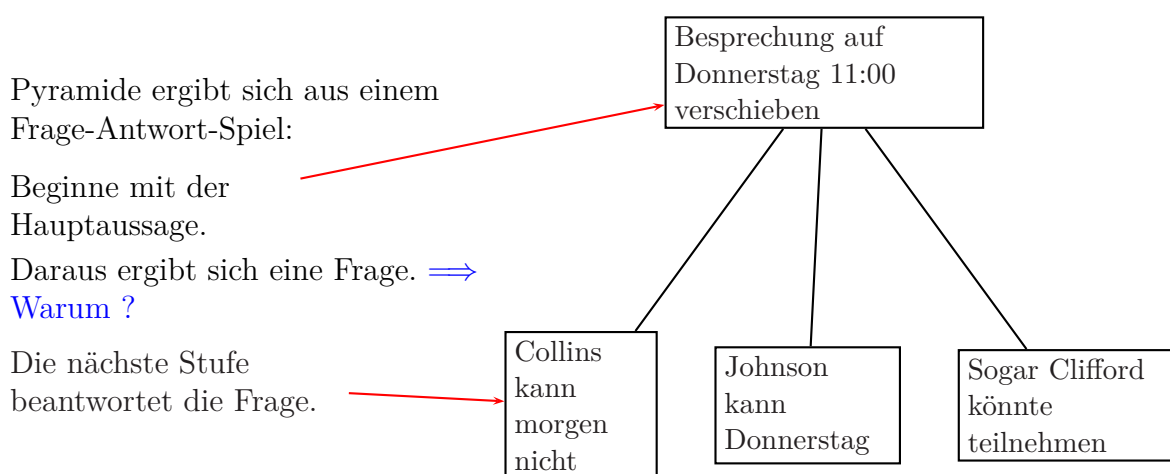
Frage

Was genau ist hier besser?

Grundprinzipien des Pyramidenprinzips

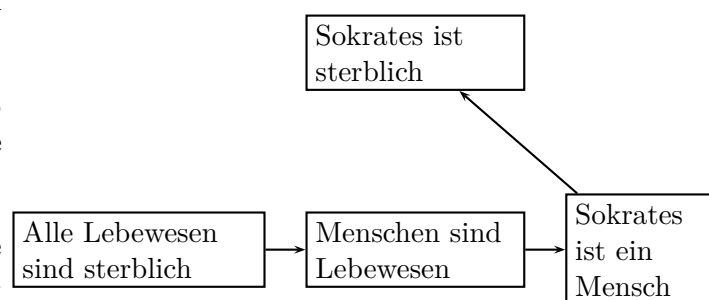
- Beginne mit dem zusammenfassenden Gedanken, bevor die einzelnen Gedanken aufgeführt werden.
- Die Gedanken auf jeder Stufe der Pyramide sind immer eine Zusammenfassung der Gedanken der darunterliegenden Stufe.
- Nebeneinander liegende Gedanken sind immer von demselben “Typ”.
- Die Gedanken einer Stufe sind immer logisch geordnet.
 - Deduktiv oder chronologisch oder strukturiert oder vergleichend

Pyramidenerstellung



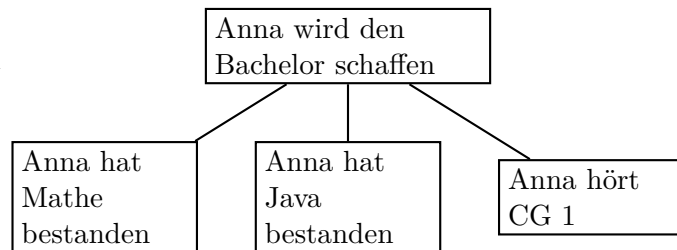
Deduktive Argumentationskette

- Eine Kette von Argumenten stützt die Hauptaussage.
- Das n -te Argument bezieht sich immer auf das $(n - 1)$ -te Argument.
- Reihenfolge der Argumente im Dokument ergibt sich natürlich.



Induktive Argumentationsketten

- Einige unterschiedliche Ideen, Fakten, Argumente gehören irgendwie zusammen.
- Übergeordnetes Argument sagt, inwiefern der Zusammenhang existiert.



2.2.5. Gute Dokumente schreiben lernen

- Übung macht den Meister. Nicht entmutigen lassen, sondern immer weiter machen. Aus den eigenen Fehlern lernen.
- Tipp: Die Erstellung einer Vorlesungsmitschrift/ -zusammenfassung parallel zu den Vortragsfolien ist eine gute Übung.
- Tipp: Lies **alle (!)** Dokumente kritisch. Ist der Aufbau klar, wird die Fachsprache korrekt benutzt, liest es sich leicht, ...? Was würde man selbst anders formulieren?
- Versuche in der persönlichen Zusammenfassung eigene, bessere Formulierungen zu finden.
- Fehlende Inhalte nie durch Sprachkomplexität vertuschen!

2.3. Literatur finden und verwalten

2.3.1. Aktueller Stand

Frage (Literatur finden)

Wie finden Sie momentan brauchbare Texte?

Frage (Literatur verwalten)

Wie merken Sie sich,

- was Sie gefunden haben?
- wo Sie es gefunden haben?
- wozu es gut war?

2.3.2. Finden von Materialien

Die Art des Herangehens hängt von verschiedenen Dingen ab:

- Ist schon Literatur vorhanden/bekannt?
- Ist das Thema extrem neu auf dem Markt oder altbekannt?
- Handelt es sich um Grundlagen oder aktuelle Forschung?
- ...

Starten von Null

Google oder Wikipedia sind nicht der schlechteste Einstieg (aber auch nur das)

- dadurch
 - Finden besserer Stichwörter
 - Finden von AutorInnen
 - Finden von Originalliteratur
 - * Bücher
 - * Original-Artikel aus Zeitschriften / Konferenzen / Sammelbänden
 - Finden von Original-Spezifikationen (W3C, Khronos, ...), White Papers, FAQs
- dann: Suchen in speziellen Datenbanken

Anderes Suchen

Suchen in speziellen Datenbanken wie

- Bibliothekskataloge
- spezielle Suchmaschinen wie Google Scholar
- Seiten von Revieworganen, z.B. Zentralblatt für Mathematik <http://www.zentralblatt-math.org>
- Seiten von ACM, IEEE, ... (siehe später)

nach

- bekannten AutorInnen
- prägnanten Schlüsselwörtern

Referenzen verfolgen

Wenn man schon Material hat, ist die wichtigste Quelle für weitere Materialien das Literaturverzeichnis, also

- ein- und ausgehende Referenzen, dabei
- Kontext beachten
- nicht alle Referenzen verfolgen

Nur jene Referenzen weiter verfolgen, die aus den Erläuterungen in der gerade vorliegenden Quelle sinnvoll erscheinen.

2.3.3. Bibliothek

Am einfachsten verfügbar ist:

- unsere Bibliothek:
<http://www.hs-hannover.de/bibl/>
- Kataloge
<http://www.hs-hannover.de/bibl/bub/kataloge/>
 - unsere Bibliothek
 - HOBSY - alle in Hannover
 - GVK-plus - alle norddt. Bundesländer plus Online-Contents
 - KVK - Karlsruher virtueller Katalog, weltweiter Verbund, alle dt. Bibliotheksverbünde

2.3.4. Verfügbarer Online Content

Interessant und (mindestens aus dem FH-Netz, oder mit VPN) verfügbar:

<http://www.hs-hannover.de/bibl/fachinformationen/datenbanken>

- ACM Digital Library
- IEEE Xplore (Electronic Library Online)
- Springer Ebooks Informatik

2.3.5. Werkzeuge zur Recherche und Literaturverwaltung

Nur ein paar Beispiele:

- ***Citavi***

- <http://www.citavi.com/>
- Literaturverwaltung und Wissensorganisation
- freie (mit Einschränkungen) und kommerzielle Version (FH: Campuslizenz)
- stand-alone, nur für Windows
- Bibliotheksrecherche mindestens deutschlandweit
- Webimport möglich
- Bib_TE_X-Export möglich oder auch formatiertes Word-Literaturverzeichnis
- Teamarbeit ist möglich

- ***CiteULike***

- <http://www.citeulike.org/>
- Bib_TE_X-Im- und Export möglich
- Gruppenunterstützung
- Einige Journals suchbar
- rein webbasiert

- ***Mendeley***

- <http://www.mendeley.com/>
- Bib_TE_X-Im- und Export möglich, Word/OpenOffice-Plugin
- Gruppenunterstützung
- Kombination von on- und offline Komponenten (rein online möglich, einfache Synchronisation)
- verfügbar für Windows/Mac/Linux/Iphone
- einfacher Webimport
- Webimport einfach auch von amazon oder GoogleScholar möglich

2.4. Literatur und Zusammenfassung

2.4.1. Literaturhinweise

- M. Deininger, H. Lichter, J. Ludewig, K. Schneider: *Studien-Arbeiten. Ein Leitfaden zur Vorbereitung, Durchführung und Betreuung von Studien-, Diplom- und Doktorarbeiten am Beispiel Informatik (Taschenbuch)*. vdf Hochschulverlag AG an der ETH Zürich; 4. Auflage; 2002
- Barbara Minto: *The Pyramid Principle – Logic in Writing and Thinking*. Pitman Publishing, London, England; 1987
- Peter Rechenberg: *Technisches Schreiben (nicht nur) für Informatiker*. Hanser Verlag; 2002
- Stefan Weber. *Das Google-Copy-Paste-Syndrom*. Heise Verlag; 2007

2.4.2. Zusammenfassung

- Siehe Übungsblatt [A.1.1](#)

3. Dokumentenverarbeitung

Inhalt

- 3.1. Dokumentmodelle
- 3.2. Text und seine Eigenschaften
- 3.3. Dokumentverarbeitung
- 3.4. Beispiel \LaTeX
- 3.5. Zusammenfassung

3.1. Dokumentmodelle

In Abhängigkeit von der Sicht auf Dokumente, kann man verschiedene Modelle identifizieren:

Beispiel: Buch

- Sicht des Autors (logische Struktur)
- Sicht des Lesers (physische Struktur)

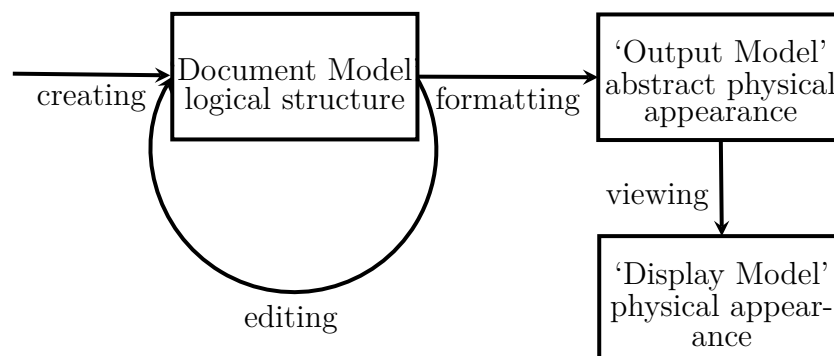


Abbildung 3.1.: Modell von Furuta, Scofield und Shaw

Logische Struktur

Alle größeren Dokumente brauchen eine vernünftige logische Struktur, sonst sind sie nicht lesbar/verständlich.

Die meistverwendete Struktur ist eine hierarchische Struktur, bzw. eine Baum-Struktur.

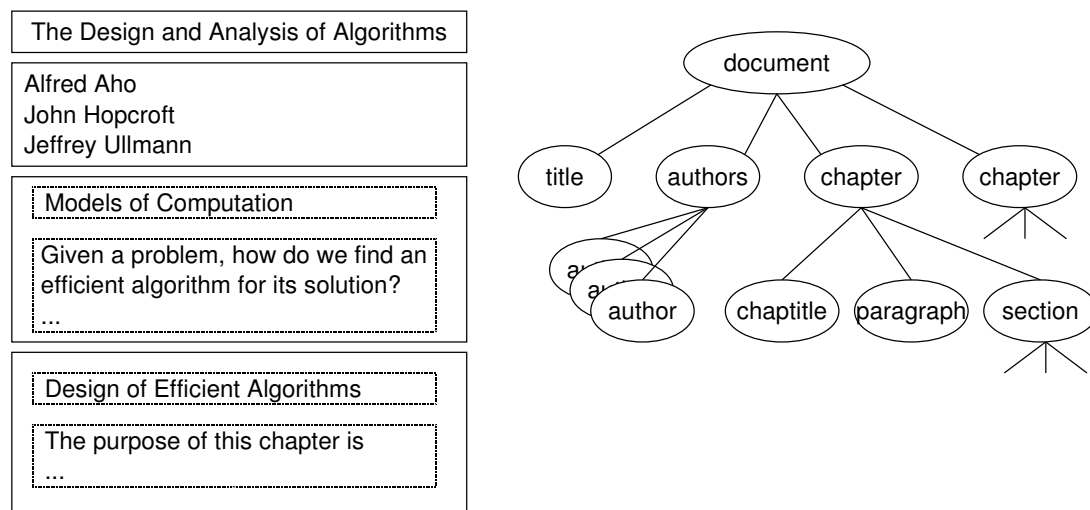


Abbildung 3.2.: Beispiel einer logischen Struktur

Beispiel: Ein Buch besteht aus Kapiteln, Kapitel bestehen aus Abschnitten, Abschnitte aus Unterabschnitten, Unterabschnitte aus Absätzen, etc.

Logische Objekte

Alle logischen Objekte haben einen **Typ**, der beschreibt:

- Welchen Namen das logische Objekt hat (authors)
- Welchen Inhalt das logische Objekt haben darf (author, Text etc.)
- Weitere Informationen zu dem Objekt

Es gibt **atomare** logische Objekte:

- Enthalten keine anderen Objekte, sondern nur noch Daten (Text).
- Sind die Blätter im Baum.

Und es gibt **zusammengesetzte** logische Objekte

- Enthalten auch andere logische Objekte.
- Sind die inneren Knoten im Baum.

Eigenschaften logischer Strukturen

Problem: Verweise auf andere logische Objekte

Beispiele: »Siehe Abbildung 3.«, »Wird in Abschnitt 1.3 erklärt« usw.

Ausgabemodell

Beschreibt das **Aussehen** eines formatierten Dokumentes noch unabhängig vom Ausgabemedium. Es werden zwei Beschreibungen benötigt:

- Wie sehen Seiten aus und welche Bereiche werden bedruckt?
- Wo auf der Seite wird was gedruckt?

Seitentemplates beschreiben den Aufbau von Seiten

- Linke Seiten vs. Rechte Seiten in Büchern
- Kopfzeilen und Fußzeilen
- Bereiche für Abbildungen und Fußnoten

Spezielle Sprachen beschreiben, was gedruckt werden soll

- Bitmap-Beschreibungen für Buchstaben oder Grafiken
- Skalierbare Beschreibungen (Vektoren, Kurven)

Heute wird hierzu überwiegend **PDF** verwendet.

Anzeigemodell

Beschreibt das physische Aussehen eines Dokumentes abhängig vom Ausgabegerät.

Bildschirme haben immer einen Bitmap-Aufbau \Rightarrow Approximation von Kurven auf Pixel erforderlich.

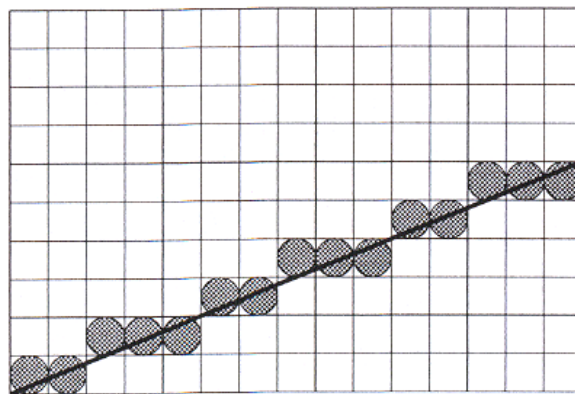


Abbildung 3.3.: Beispiel: Bresenham-Algorithmus.

3.2. Text und seine Eigenschaften

Buchstaben

- Besitzen ein Aussehen und verschiedene Größendimensionen
- **Höhe** Oberkante des Buchstabens bis zur **Grundlinie**
- **Tiefe** Unterkante des Buchstabens bis zur Grundlinie
- **Breite** Linker Rand des Buchstabens bis zum rechten Rand

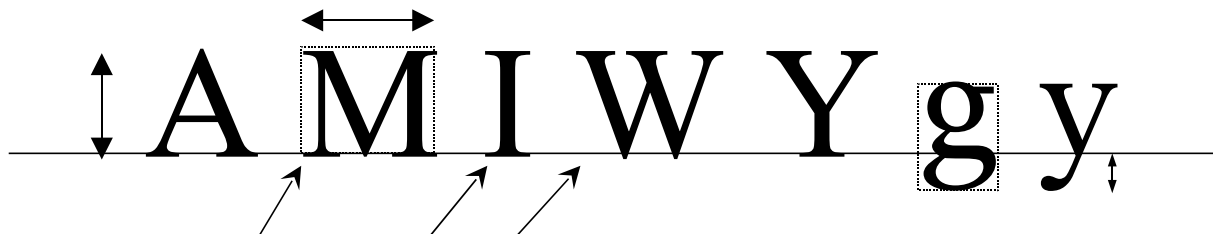


Abbildung 3.4.: Buchstaben mit ihren Dimensionen.

Attribute von Buchstaben

- Größe _A A A A A A A A
- Schriftfamilie: normal, *geneigt*, *kursiv*, **halbfett**, KAPITÄLCHEN.
- Schriftsippe: Computer Modern Roman, Computer Modern Sans Serif, Computer Modern Typewriter.

Typografische Größenangaben

Neben Millimeter und Zoll gibt es spezielle typografische Größenangaben:

Absolut:

- Didot Point dd
ca. 0,376 mm (traditionell in Europa üblich.)
- Printer's Point pp¹
ca. 0,351 mm (traditionell in den USA üblich.)
- DTP Point pt²
ca 0,353 mm.

Relativ:

- em – entspricht der Breite des Gevierts ■, also ungefähr der Breite des M.
- ex – entspricht der Höhe eines Kleinbuchstabens ohne Oberlänge, also der Höhe des x.

Welche Schrift?

Für lange Texte:

- Festlegen auf eine oder maximal zwei Schriftfamilien
 - Am besten eine Antiqua und eine **Grotesk**.
 - Vorzugsweise bei der selben Sippe bleiben.
 - **Achtung:** Gute Schriftauswahl ist sehr schwierig!

¹In L^AT_EX pt abgekürzt.

²In L^AT_EX bp (big point) abgekürzt.

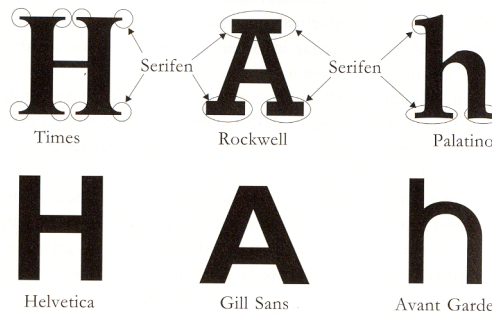


Abbildung 3.5.: Vergleich einiger Antiqua- (mit Serifen) und Grotesk-Schriften (ohne Serifen).

- Hervorheben von Wörtern durch *kursiv setzen*.
- Besonderes Hervorheben von Wörtern durch **fett setzen** – im Fließtext meiden.
- Hervorhebungen durch Unterstreichungen *meiden!*
- Überschriften durch verschiedene Schriftgrößen oder -schnitte kenntlich machen.

Ziele der Typografie

- Gute Lesbarkeit!!!
- Form unterstützt Inhalt und Zweck eines Werkes
 - Meistens: Optimale Lesbarkeit
 - Manchmal: Maximale Unleserlichkeit
- Hilfreich:
 - Einheitlicher Grauwert der Seite.
 - Gut erkennbare (ausreichend große) Buchstaben.
 - Kein Durcheinander von vielen verschiedenen Formatierungsmöglichkeiten oder sonstigen Optionen.

3.3. Dokumentverarbeitung

3.3.1. Zeilen und Absätze

Wortzwischenräume (Leerzeichen) werden als Boxen mit gegebener Dehnbarkeit und Stauchbarkeit modelliert.

First-Fit: Versucht eine Zeile voll zu packen und staucht die Leerzeichen, um evtl. ein weiteres Wort unter zu bringen (z. B. Word oder Writer).

Best-Fit: Versucht eine Zeile optimal zu packen, d.h. die Zeile wird möglicherweise nicht so voll wie es ginge (z. B. Troff).

Optimal-Fit: Betrachtet einen Absatz als Ganzes und versucht einen Umbruch zu finden, der alle Zeilen gut umbricht (z. B. \TeX).

(a) In olden times when wishing still helped one, there lived a king
 whose daughters were all beautiful; and the youngest was so
 beautiful that the sun itself, which has seen so much, was aston-
 ished whenever it shone in her face. Close by the king's castle lay
 a great dark forest, and under an old lime-tree in the forest was
 a well, and when the day was very warm, the king's child went
 out into the forest and sat down by the side of the cool fountain;
 and when she was bored she took a golden ball, and threw it up
 on high and caught it; and this ball was her favorite plaything.

(b) In olden times when wishing still helped one, there lived a king
 whose daughters were all beautiful; and the youngest was so
 beautiful that the sun itself, which has seen so much, was aston-
 ished whenever it shone in her face. Close by the king's castle
 lay a great dark forest, and under an old lime-tree in the forest
 was a well, and when the day was very warm, the king's child
 went out into the forest and sat down by the side of the cool
 fountain; and when she was bored she took a golden ball, and
 threw it up on high and caught it; and this ball was her favorite
 plaything.

Abbildung 3.6.: First-Fit im Vergleich zu Best-Fit.

(b) In olden times when wishing still helped one, there lived a king
 whose daughters were all beautiful; and the youngest was so
 beautiful that the sun itself, which has seen so much, was aston-
 ished whenever it shone in her face. Close by the king's castle
 lay a great dark forest, and under an old lime-tree in the forest
 was a well, and when the day was very warm, the king's child
 went out into the forest and sat down by the side of the cool
 fountain; and when she was bored she took a golden ball, and
 threw it up on high and caught it; and this ball was her favorite
 plaything.

(c) In olden times when wishing still helped one, there lived a
 king whose daughters were all beautiful; and the youngest was
 so beautiful that the sun itself, which has seen so much, was
 astonished whenever it shone in her face. Close by the king's
 castle lay a great dark forest, and under an old lime-tree in the
 forest was a well, and when the day was very warm, the king's
 child went out into the forest and sat down by the side of the
 cool fountain; and when she was bored she took a golden ball,
 and threw it up on high and caught it; and this ball was her
 favorite plaything.

Abbildung 3.7.: Best-Fit im Vergleich zu Optimal-Fit.

- Anfang des Absatzes wird durch Einzug kenntlich gemacht
- Zeilenabstände innerhalb eines Absatzes sind konstant, außer es kommen besondere Zeichen (sehr groß oder höhergestellt oder tiefergestellt) vor

3.3.2. Seiten und Satzspiegel

Schwieriger als Zeilenumbruch

- Keine Dehnbarkeit und Stauchbarkeit zwischen Zeilen in einem Absatz
- Fließende Objekte, z.B. Abbildungen und/oder Fußnoten
- Weitere Randbedingungen (keine Schusterjungen oder Hurenkinder, Absatzkontrolle)

Es gibt noch keine Programme, die optimalen Seitenumbruch implementiert haben.

In den meisten Programmen wird ein First-Fit Algorithmus benutzt und der Anwender muss manuell Verbesserungen vornehmen, wenn er mit dem Ergebnis nicht zufrieden ist.

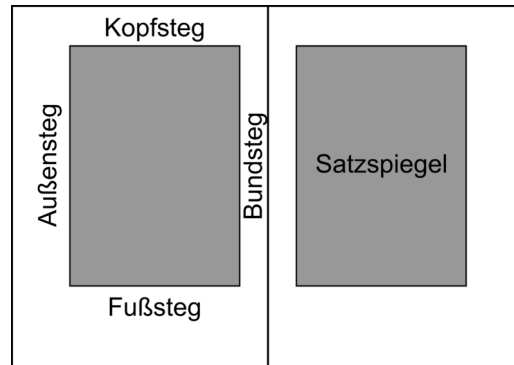


Abbildung 3.8.: Seitenaufbau eines Buches. (Quelle: User Metoc, Wikimedia Commons)

- Ränder an den Seiten dürfen nicht zu klein gewählt werden.
- Bei doppelseitigem Druck (Buchdruck) ist der innere Rand halb so klein wie der äußere, da bei aufgeschlagenem Buch die beiden inneren Ränder zusammen liegen.
- Gut lesbare Zeilenbreite. Richtwert: 45–65 Zeichen pro Zeile
- Mehrspaltiger Satz bei kleinem Schriftgrad (Zeitungen)
- Gleitende Objekte (Abbildungen/Tabellen) kommen gewöhnlich an den Seitenanfang
- Fußnoten kommen an das Seitenende

3.4. Beispiel \LaTeX

3.4.1. Besonderheiten von \LaTeX

Besonderheiten von \LaTeX :

- Unterstützt logisches Markup, also Auszeichnungen unabhängig von der Formatierung
- Viele Pakete für Dokumentklassen existieren und können einfach benutzt werden
- Spezielle Pakete für spezielle Formatierungsprobleme existieren und können einfach benutzt werden
- Spezielle Programme zur Erstellung von Literaturverzeichnissen sind vorhanden (bibtex)
- Spezielle Programme zur Erstellung von Indices sind vorhanden (makeindex)

3.4.2. Erstes Beispiel

```
\documentclass{scrlettr2}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}
\begin{document}
\begin{letter}{%
Chuck Norris\\
1 Chuck Norris Street\\
Norris Town\\
USA%
}
\opening{Lieber Chuck,}
du bist so toll!
\closing{In Erfurcht}
\end{letter}
\end{document}
```

Chuck Norris
1 Chuck Norris Street
Norris Town
USA

5. Mai 2011

Lieber Chuck,

du bist so toll!

In Erfurcht

Abbildung 3.9.: Ein Brief in L^AT_EX.

3.5. Zusammenfassung

- Dokumentmodelle (logische Sicht, Ausgabesicht, Betrachtungssicht)
- Text und Buchstaben, Attribute und Schriftfamilien
- Dokumentverarbeitung (Wörter aus Buchstaben, Zeilen aus Wörtern, Seiten aus Zeilen bilden)
- T_EX und L^AT_EX

4. L^AT_EX-Grundlagen

Inhalt

- 4.1. Von T_EX zu L^AT_EX
- 4.2. Arbeiten mit L^AT_EX
- 4.3. Pakete einbinden
- 4.4. Verwendung von Gliederungsebenen in wissenschaftlichen Arbeiten
- 4.5. Nützliche Kleinigkeiten
- 4.6. Texteingabe
- 4.7. Zusammenfassung

4.1. Von T_EX zu L^AT_EX

T_EX ist ein Satzsystem – es simuliert den klassischen Bleisatz des Buchdrucks.

Mögliche Ausgabeformate sind (z. B.): DVI, PDF, PS, HTML ...

Der Mathematiker Donald E. Knuth war unzufrieden mit den Probedrucken der zweiten Auflage von Band 2 seiner Buchreihe *The Art of Computer Programming*. Die erste Auflage war klassisch gesetzt worden.

Die zweite mit dem Computer – besonders die mathematischen Formeln sahen furchtbar aus.

Ende der 1970er entstand T_EX – oder TEX (Tau Epsilon Chi) als Abkürzung für Τέχνη, (alt-) griechisch für **Fähigkeit, Kunstfertigkeit, Handwerk**.

Aussprache eigentlich Tech mit ch wie in Buch. Im Deutschen aber meist wie in *Technik* – im Englischen wie in *technology*.

T_EX ist im mathematischen naturwissenschaftlichen Bereich quasi ein Standard.

T_EX wird nicht mehr aktiv weiterentwickelt, weil es praktisch fehlerfrei ist.

Gearbeitet wird an Nachfolgeprodukten wie pdfTeX, luaTeX oder XeTeX.

Wir arbeiten hier mit pdfTeX.

L^AT_EX ist ein **Makropaket**, welches auf das Textsatzsystem T_EX aufsetzt, entwickelt von Leslie Lamport Anfang der achtziger Jahre. Entsprechend steht L^AT_EX für *Lamport's T_EX*.

Die **Struktur** eines Dokument wird durch Auszeichnungen, Tags, markiert und später durch Übersetzung des Quelltextes in formatierten Text verwandelt.

T_EX ist ein **Textsatzprogramm**, welches sich besonders für den wissenschaftlichen Einsatz eignet, weil es:

- ohne Probleme mit großen Dokumenten umgehen kann,
- Formeln ansprechend darstellt und
- feine Kontrolle über die Typografie erlaubt.

Warum L^AT_EX?

- plain T_EX ist kompliziert.
- Über Layout und Formatierung braucht man sich keine Gedanken mehr zu machen.
- Man kann aber, wenn man möchte.
- Hervorragende Satzqualität.
- Viele Ausgabeformate.
- Geräteunabhängig.
- Äußerst stabil.
- Sehr gut dokumentiert.

Wofür eignet sich L^AT_EX?

- Jede Art wissenschaftlicher Veröffentlichung.
- Bücher (Sachbücher, Romane, Lexika etc.).
- Bereiche, in denen Texte automatisch erzeugt werden.
- Serienbriefe.
- Vorträge (zum Beispiel dieser hier).
- (Technische) Zeichnungen.
- Gedichte.
- Lebensläufe.
- Schach, Bridge, Kreuzworträtsel, Sudoku etc.
- Und mehr!

Wofür eignet sich L^AT_EX weniger gut?

- Zeitungssatz.
- Plakate, Flyer.
- Sehr kurze Texte.

- Alle Bereiche, in denen die Elemente einer Seite möglichst frei angeordnet werden sollen.¹

Normale Desktop-Publishing-Programme sind WYSIWYG ‘What You See Is What You Get.’

L^AT_EX ist WYGIWYM

‘What You Get Is What You Mean’

L^AT_EX ist freie Software und Teil jeder gängigen Linux- oder BSD-Distribution.

Auch Mac-Benutzer gehen nicht leer aus.

Selbst exotische oder veraltete Systeme werden unterstützt.

4.2. Arbeiten mit L^AT_EX

4.2.1. Grundsätzliches Vorgehen

Grundsätzliches Vorgehen:

Anders als z. B. Writer:

1. Verfassen des Textes.
2. Übersetzen.
3. Ergebnis betrachten, kontrollieren. Gegebenenfalls korrigieren und zurück zu 2!
4. Drucken.

Man vergleiche dieses Vorgehen mit dem Modell von Furuta, Scofield und Shaw, Abbildung 3.1 auf Seite 21.

L^AT_EX-Quellcode liegt im Nur-Text-Format vor. Beim Editor hat man also freie Wahl!

Zum Beispiel:

- vim (für alle Plattformen)²
- emacs (für alle Plattformen)
- texmaker (für alle Plattformen)
- kile (unixuide, inkl. Mac OS X)
- TeXShop (Mac OS X)
- TeXnicCenter (Windows)
- TeXworks (für alle Plattformen)
- TeXlipse (Plugin für Eclipse)

¹Wer dies möchte und trotzdem nicht auf T_EX verzichten möchte, möge sich einmal ConT_EXt anschauen.

²Mit *alle Plattformen* sind unixoide (Linux, BSD, Solaris etc.), Mac OS X (eigentlich auch unixoid, aber häufig von Programmen nicht, oder ausschließlich unterstützt) und Windows gemeint.

Der Editor, mit dem man am besten umgehen kann, ist der beste!

Wer noch keinen Lieblings-Editor hat, möge **TeXworks** ausprobieren.

TeXworks ist nicht mit Funktionen überladen und daher einfach zu benutzen. Der eingebaute PDF-Betrachter erlaubt es, von jeder Stelle im Dokument zur entsprechenden Stelle im Quelltext zu springen – besonders beim Korrigieren ausgesprochen hilfreich.

4.2.2. Hallo, L^AT_EX!

Grundgerüst eines L^AT_EX-Dokuments

- | | |
|----------------------------------|---|
| <code>\documentclass{...}</code> | Die Dokumentenklasse (auch Style- oder Class-File legt das Grundlayout fest; z. B. Buch , Bericht , Artikel . |
| <code>\begin{document}</code> | Anfang des eigentlichen Dokumenteninhalts. |
| <code>\end{document}</code> | Ende des Dokuments. |

```
\documentclass{article}
\begin{document}
Hello , world!
\end{document}
```

Das Ergebnis dieses Codes befindet sich auf der nächsten Seite.

Eingaberegeln:

- Einzelne Befehle und Wörter werden durch Leerzeichen voneinander getrennt.
- Leere Zeilen trennen Absätze. (Alternativ kann der Befehl `\par` verwendet werden.)³
- Mehrere Leerzeichen oder -zeilen werden wie ein Leerzeichen oder eine Leerzeile angesehen.
- Ohne Zusatzpakete ist nur ASCII-Text erlaubt.

4.2.3. Befehle

Befehle...

- ... beginnen mit einem Backslash: `\`.
- ... enden mit
 - einem Leerzeichen oder jedem anderen Zeichen, das kein Buchstabe ist, z. B. einer Zahl
 - einer (leeren) Gruppe `{}`,
 - dem Beginn des nächsten Befehls.

³Es gibt auch den Befehl `\`. Dieser dient zum Erzwingen eines *Zeilenumbruchs*, **nicht zum Beenden eines Absatzes**. Den doppelten Rückschrägstrich als Absatztrenner zu missbrauchen ist ein typischer Anfängerfehler, ähnlich wie dem, Kaffee in der Mensa zu kaufen.

Hello, world!

1

Abbildung 4.1.: Hallo, \LaTeX !

- ... unterscheiden zwischen Groß- und Kleinschreibung: `\large` ungleich `\LARGE`!
- ... haben einen Gültigkeitsbereich: `{\large large}` nicht `large`
- ... fügen keinen Leerraum hinter sich an: `\LaTeX Fun` — `\LaTeX{} Fun` führt zu L^AT_EXFun – L^AT_EX Fun.
- ... können Parameter haben: `\large{large text}`

4.2.4. Umgebungen

Umgebungen werden so angegeben:

```
\begin{umgebung}
Inhalt
\end{umgebung}
```

Umgebungen dienen der Auszeichnung von größeren Textmengen.

- Gruppieren größere Bereiche des Dokumentes, vergleichbar zu Elementen in XML.
- Umgebungen stehen entweder
 - nebeneinander
 - komplett in einander verschachtelt

Beispiele:

- `\begin{abstract} ... \end{abstract}`
(Abstrakt)
- `\begin{itemize} ... \end{itemize}`
(Aufzählung mit Punkten)
- `\begin{enumerate} ... \end{enumerate}`
(nummerierte Aufzählung)
- `\begin{center} ... \end{center}`
- `\begin{quotation} ... \end{quotation}`

4.2.5. Kommentare

Kommentare werden in L^AT_EX mit einem %-Zeichen gemacht:

```
Text .
% Comment
Text . % Comment
```

(Das Prozentzeichen muss daher als `\%` geschrieben werden.)

4.2.6. Dokumentklassen

Jedes Dokument beginnt mit dem Befehl `\documentclass`.

Die Dokumentenklasse bestimmt das Layout für das Dokument, z. B. einen Artikel:

```
\documentclass{article}
```

Die meisten Dokumentenklassen akzeptieren optionale Parameter, die das Layout anpassen, z. B.:

```
\documentclass[a4paper,twoside,twocolumn]{article}
```

Der Artikel hat nun das DIN-A4-Format (statt US Letter), wird doppelseitig gedruckt und zweispaltig gesetzt.

Standard-Dokumentklassen

article Für kurze Texte wie Zeitschriftenartikel entworfen. Artikel haben keine Kapitel, die höchste Gliederungsebene sind Abschnitte. Einseitig.⁴

report Für längere Texte wie Berichte entworfen. Berichte enthalten Kapitel. Einseitig.

book Für lange Texte wie Bücher entworfen. Doppelseitig.

letter Für Briefe entworfen.

KOMA-Script

Die Standardklassen sind von Leslie Lamport in den 80er Jahren in den USA entworfen worden – und das sieht man ihnen an.

KOMA-Script hingegen:

- bietet erweiterte Möglichkeiten und mehr Befehle,
- entspricht europäischen Normen und europäischer Typografie,
- wird ständig weiterentwickelt und
- verfügt über eine außerordentlich umfangreiche deutschsprachige Dokumentation.

Tabelle 4.1.: KOMA-Script-Entsprechungen der Standard-Klassen.

Standard	KOMA
article	scrartcl
report	scrreprt
book	scrbook
letter	scrlettr2

⁴Das heißt, *nicht doppelseitig* und nicht *langweilig*.

4.3. Pakete einbinden

In der Präambel eines Dokuments (der Teil zwischen `\documentclass` und `\begin{document}`) können zusätzliche Pakete mit dem Befehl `\usepackage` eingebunden werden.

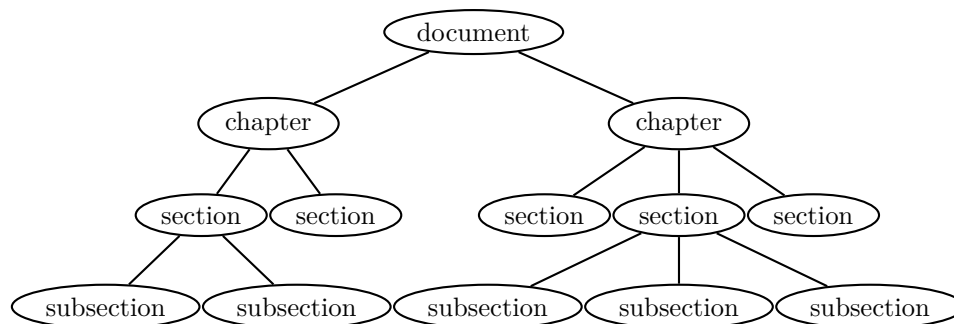
Mindestens sollten geladen werden:

```
\documentclass{scrartcl}
\usepackage[utf8]{inputenc}      % UTF8-Kodierung
\usepackage[T1]{fontenc}        % Erweiterter Zeichenvorrat
\usepackage[ngerman]{babel}     % deutsche Silbentrennung
\begin{document}
Hallo , Welt!
\end{document}
```

4.4. Verwendung von Gliederungsebenen in wissenschaftlichen Arbeiten

4.4.1. Hierarchie

Eine Gliederung ist ein Baum.



- Wie tief darf der Baum werden?
- Wieviele Kinder darf ein innerer Knoten im Baum haben?
- Wie finde ich die richtige Hierarchie?

4.4.2. Tiefe der Hierarchie

Die Tiefe einer Hierarchie hängt von der Größe des Dokumentes ab. Ein langes Buch, in dem viele verschiedene Fachgebiete diskutiert werden, braucht eine tiefere Hierarchie als ein Kurzbericht, der auf drei Seiten passt.

In einer Abschlussarbeit kann man fast immer mit **drei** Hierarchieebenen auskommen.

4.4.3. Breite der Hierarchie

Für die Anzahl der Kinder eines inneren Knotens gibt es keine festen Vorgaben. Allerdings ist nur ein Kind etwas sehr wenig. Warum hat man es überhaupt eingeführt?

Nach oben sollte man versuchen, nicht mehr als 7 Kinder eines inneren Knotens zu haben. Soviele Punkte kann sich ein Leser so gerade noch merken.

In Ausnahmefällen können auch mehr als 7 Kinder auftreten. Ein Beispiel hierfür wäre eine Marktübersicht, in der man Produkte verschiedener Hersteller vergleicht und jedes Produkt in einem eigenen Unterkapitel erklärt. Nun ist die Zahl sieben nicht mehr so wichtig, denn ein Leser würde sich nicht jeden Produktnamen merken wollen.

Also: i. A. *mehr als ein, nicht mehr als sieben* Unterabschnitte.

4.4.4. Finden der Hierarchie

Wie finde ich eine Hierarchie? Die Antwort auf diese Frage ist nicht trivial. Sie muss in jedem Einzelfall neu gefunden werden. Man kann jedoch einige allgemeine Richtlinien geben:

- Der Hierarchiebaum sollte einigermaßen ausgeglichen sein. Jeder Knoten auf einer Ebene hat die gleiche Struktur oder eine sehr ähnliche Struktur.
- In einer Abschlussarbeit gibt es auf der obersten Hierarchieebene fast immer die Abschnitte „Einleitung“ und „Zusammenfassung und Ausblick“. Sie stehen am Anfang und Ende des Dokumentes.
- Die inneren Knoten der Hierarchie ergeben sich aus dem Themengebiet und der Art und Weise wie die Autorin das Themengebiet für sich selbst strukturiert hat.

Beispiele:

- Schreibt man über einen Prozess, d.h. die zeitliche Abfolge von Operationen, so kann man das direkt auf die Struktur übertragen. Die Kinder eines inneren Knotens bestehen dann aus den einzelnen Operationen.
- Oder wenn man eine Tatsache aus verschiedenen Blickwinkeln analysiert, so kann man das direkt in eine Struktur übertragen. Jeder Blickwinkel wird dann ein eigener Kindknoten.

4.4.5. Hierarchien in L^AT_EX

Tabelle 4.2.: Verfügbare Hierarchien in L^AT_EX

Name	Anmerkungen
<code>\book</code>	Spezifisch für die Klasse <i>Memoir</i> .
<code>\part</code>	Nicht in <i>Article</i> oder <i>scrartcl</i> verfügbar.
<code>\chapter</code>	Nicht in <i>Article</i> oder <i>scrartcl</i>
<code>\section</code>	
<code>\subsection</code>	
<code>\subsubsection</code>	
<code>\paragraph</code>	
<code>\subparagraph</code>	
<code>\minisec</code>	Nur KOMA-Script

4.4.6. Überschriften

- L^AT_EX nummeriert die Überschriften automatisch.
- Die Formatierung der Überschriften erfolgt automatisch.
- Kopf- und Fußzeilen werden automatisch generiert.
- Bei `\xxx*` unterbleibt die automatische Numerierung!

`\section*{Einleitung ohne Nummer}`

4.5. Nützliche Kleinigkeiten

4.5.1. Schriftarten

```
\documentclass{article}
\begin{document}

{\rmfamily   Hello World ! (roman)}\\
{\sffamily   Hello World ! (sans serif)}\\
{\ttfamily   Hello World ! (Schreibmaschine)}\\
{\bfseries   Hello World ! (fett)}\\
{\itshape    Hello World ! (italic)}\\
{\slshape    Hello World ! (slanted)}\\
{\scshape    Hello World ! (small caps)}\\
{\itshape\bfseries Hello World ! (italic bold)}\\
{\sffamily\bfseries Hello World !
  (sans serif bold)}\\

\end{document}
```

Ergebnis:

Hello World ! (roman)

Hello World ! (sans serif)

Hello World ! (Schreibmaschine)

Hello World ! (fett)

Hello World ! (italic)

Hello World ! (slanted)

HELLO WORLD ! (SMALL CAPS)

Hello World ! (italic bold)

Hello World ! (sans serif bold)

4.5.2. Schriftgrößen

```
\documentclass{article}
\begin{document}
{\tiny winzig}\\
{\scriptsize sehr klein}\\
{\footnotesize Fussnote}\\
{\small klein}\\
{\normalsize normal}\\
{\large gross}\\
{\Large groesser}\\
{\LARGE noch groesser}\\
{\huge riesig}\\
{\Huge gigantisch}
\end{document}
```

Ergebnis:

winzig

sehr klein

Fußnote

klein

normal

groß

größer

noch größer

riesig

gigantisch

4.6. Texteingabe

4.6.1. Eingabe von Absätzen

- Keine Formvorschriften (Vergleiche C oder JAVA)
- L^AT_EX berechnet den optimalen Zeilenumbruch selbstständig
- Neue Absätze beginnen, wenn zwei RETURNS direkt hinter einander stehen (also eine Leerzeile da steht)

Da das gern ein langer Text ist, werden subsections benutzt. Einige Hinweise sollte man aber beachten.

Man sollte nach jedem Satzende eine neue Zeile beginnen. Ausserdem ist es geschickt, wenn die Zeilen im Editor komplett zu sehen sind.

4.6.2. Besonderheiten in Absätzen

- Deutsche Anführungszeichen können auf verschiedenen Arten eingegeben werden.
 - Als Kommando: `\glqq` und `\grqq`
 - Kurzform: `"` und `“`
- Wenn Silbentrennung nicht erkannt wird, dann können Trennstellen mit dem Kommando
`\-` oder `"-` explizit eingegeben werden
- Beispiel: `Tren\-\nungs\-\hil\-\fe`
- Sonderzeichen, die von L^AT_EX benutzt werden, müssen mit `\` maskiert werden, z.B.
`\{` = {, `\}` = }, `\%` = %, `\&` = &, `\S` = §, `\backslash` = \

4.6.3. Zwischen Absätzen

- Abstand wird von L^AT_EX selbst bestimmt.
- Braucht man einmal größere Abstände, so gibt es das Kommando
 - `\vspace{2cm}` z.B. Abstand von 2 cm zwischen Absätzen
- Um das Seitenende explizit an eine bestimmte Stelle zu setzen gibt es die Kommandos:
 - `\pagebreak` fängt neue Seite an und dehnt den Inhalt der letzten Seite so weit, dass sie unten bündig abschließt.

- `\newpage` fängt neue Seite an und lässt auf der vorhergehenden Seite unten freien Platz.
- `\clearpage` fängt neue Seite an und lässt auf der vorhergehenden Seite unten freien Platz. Außerdem werden alle noch nicht platzierten Gleitobjekte platziert.

4.7. Zusammenfassung

- \TeX ist ein mächtiges Textsatzsystem.
- \LaTeX trennt Formatierung und Struktur.
- Freie Wahl des Editors.
- Es gibt Befehle, Umgebungen und verschiedene Dokumentenklassen.
- Mit Paketen erweiterbar.
- Dokumente haben Hierarchien.
- \LaTeX macht Spaß.

5. Mathematik, Verzeichnisse, Referenzen und Zitate

Inhalt

- 5.1. Mathematik
- 5.2. Gleitobjekte in \LaTeX
- 5.3. Verzeichnisse
- 5.4. Referenzen und Verweise
- 5.5. Zitate und Bibliographie
- 5.6. MakeIndex
- 5.7. Zusammenfassung

5.1. Mathematik

Der Mathematiksatz ist *die* große Stärke von \LaTeX . Alle mathematischen Journale verlangen heute (spätestens seit Beginn der 1990er) die Einsendung von Artikeln in \LaTeX (vorher in \TeX). Die meisten Bücher mit hohem Mathematikanteil sind in \LaTeX gesetzt.

5.1.1. Formelarten

In \LaTeX gibt es zwei Arten von Formeln (Matheumgebungen):

- ***Textformeln***, wie $a = \sum_{i=0}^n b_i$, eingeschlossen in $\$$ -Zeichen, also

```
$a = \sum_{i=0}^n b_i$
```

- ***abgesetzte Formeln***, die in $\left[\dots \right]$ eingeschlossen werden, oder extra-Umgebungen haben (später), wie

$$a = \sum_{i=0}^n b_i$$

also

```
\[ a = \sum_{i=0}^n b_i \]
```

5.1.2. Formelteile

In L^AT_EX stehen zur Verfügung, z.B.:

- Hoch- und Tiefstellung (siehe eben)
- griechische Buchstaben wie α, β , als `\alpha, \beta`
- Summen-, Produkt-, Unendlich- und noch viel mehr Zeichen \sum, \prod, ∞ als `\sum, \prod, \infty`
- Brüche $\frac{a+b}{c}$ als `\frac{a+b}{c}`
- Befehle mit etwas Englisch und gesundem Menschenverstand erwartbar

5.1.3. Weitere Befehle und Umgebungen

- nachzuschlagen in der Symbolliste und dem Mathemoduspapier

```
texdoc symbols
texdoc mathmode
```

- bei guten Editoren: Klick auf eine Auswahlliste
- nummerierte Formeln mit *equation*-Umgebung
- zusätzliche Befehle und Umgebungen für den Mathematiksatz haben die Pakete, die für die AMS (American Mathematical Society) geschrieben wurden: *amsmath* und *amssymb*, darin so nützliche Umgebungen wie *cases* oder *align*

5.1.4. Symbole

Tabelle 5.1.: Theoretische Informatik

\rightarrow	<code>\to</code>	
\Rightarrow	<code>\Longrightarrow</code>	
\vdash	<code>\vdash</code>	Konfigurationsübergang
\perp	<code>\perp</code>	Kellerbottomsymbol, senkrecht
ε	<code>\varepsilon</code>	leeres Wort
δ	<code>\delta</code>	
Δ	<code>\Delta</code>	
Σ	<code>\Sigma</code>	Alphabet
Γ	<code>\Gamma</code>	Kelleralphabet

Tabelle 5.2.: Logik

\Rightarrow	<code>\Longrightarrow</code>	daraus folgt
\forall	<code>\forall</code>	für alle ...
\exists	<code>\exists</code>	es existiert ein ...
\neg	<code>\neg</code>	negiert
\wedge	<code>\wedge</code>	und
\vee	<code>\vee</code>	oder

Tabelle 5.3.: Mengen

\cap	<code>\cap</code>	Durchschnitt
\cup	<code>\cup</code>	Vereinigung
\cap	<code>\bigcap</code>	Durchschnitt (mit Indizes)
\cup	<code>\bigcup</code>	Vereinigung (mit Indizes)
\emptyset	<code>\emptyset</code>	leere Menge
\in	<code>\in</code>	ist Element von
\notin	<code>\notin</code>	ist kein Element von
\subset	<code>\subset</code>	ist echte Teilmenge von
\subseteq	<code>\subseteq</code>	ist Teilmenge von
$\not\subseteq$	<code>\not\subseteq</code>	ist keine Teilmenge von
\setminus	<code>\setminus</code>	minus (für Mengen)
Δ	<code>\Delta</code>	symmetrische Differenz

Tabelle 5.4.: Relationen

\leq	<code>\leq</code>	kleiner gleich
\geq	<code>\geq</code>	größer gleich
\neq	<code>\neq</code>	ungleich
\approx	<code>\approx</code>	annähernd
\equiv	<code>\equiv</code>	äquivalent

Tabelle 5.5.: Akzente an Variablen

\tilde{x}	<code>\tilde{x}</code>
\hat{x}	<code>\hat{x}</code>
\overline{x}	<code>\overline{x}</code>
\underline{x}	<code>\underline{x}</code>
\vec{x}	<code>\vec{x}</code>

Tabelle 5.6.: Andere Schriftarten

\mathbb{N}	<code>\mathbb{N}</code>	Blackboard Bold
\mathcal{L}	<code>\mathcal{L}</code>	kalligraphisch, nur für Großbuchstaben
\mathbf{A}	<code>\mathbf{A}</code>	fette Symbole
\mathbf{A}	<code>\mathbf{A}</code>	fett und aufrecht
\mathbf{A}	<code>\pmb{A}</code>	poor man's bold

5.1.5. Formelbeispiele

$$\mathbb{Q} = \left\{ \frac{z}{n} \mid z \in \mathbb{Z}, n \in \mathbb{N} \right\}$$

```
\[ \mathbb{Q} = \left\{ \frac{z}{n} \right\}; \Big| \;
```

```
z \in \mathbb{Z}, n \in \mathbb{N} \right\}
```

`\left .. \right` sorgen dafür, dass das benutzte Klammersymbol sich automatisch der Größe des Inhalts anpasst.

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n = e$$

```
\[ \lim_{n \rightarrow \infty} \left( 1 + \frac{1}{n} \right)^n = e \]
```

$$\int_a^b \sin^2 x \, dx = \sin x \cos x \Big|_a^b + \int_a^b \cos^2 x \, dx$$

```
\[ \int_a^b \sin^2 x \, dx =
```

```
\sin x \cos x \Big|_a^b
```

```
+ \int_a^b \cos^2 x \, dx \]
```

`\,` und `\;` sorgen für kleine Abstände

Matrizen:

$$\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{I}_n$$

$$= \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}}_{n\text{-mal}}$$

```
\begin{align*}
```

```
\boldsymbol{A} \cdot \boldsymbol{A}^{-1}
```

```
&= \boldsymbol{I}_n \\\
```

```
&= \underbrace{\begin{pmatrix}
```

```
1 & 0 & \cdots & 0 \\\
```

```
0 & 1 & \ddots & \vdots \\\
```

```
\vdots & \vdots & \ddots & \vdots \\\
```

```
0 & \cdots & 0 & 1
```

```
\end{pmatrix}}_{n\text{-mal}}
```

```
\end{align*}
```

5.2. Gleitobjekte in L^AT_EX

5.2.1. Idee

- Wegen der Größe von Grafiken und Tabellen ist der Seitenumbruch schwierig.
- Wenn eine Grafik auf die nächste Seite verschoben werden muss, dann wandert (gleitet, fließt) der nachfolgende Text an der Grafik vorbei auf die vorherige Seite.
- L^AT_EX besitzt zwei Umgebungen für Gleitobjekte
 - `\begin{figure} ... \end{figure}`
 - `\begin{table} ... \end{table}`

5.2.2. Abbildungen

Bildunterschriften stehen unter dem Bild.

```
\begin{figure}[ort]
% Inhalt des Bildes
\caption[kurzunterschrift]{Bildunterschrift}
\label{bildlabel}
\end{figure}
```

Ort bezeichnet, wohin das Bild gleiten darf

- p eigene Seite (Page)
- h Hier (Here)
- t Oben auf der Seite (Top)
- b Unten auf einer Seite (Bottom)

Die Kurzunterschrift wird für das Abbildungsverzeichnis verwendet.

5.2.3. Grafiken einbinden

Grafiken werden i.A. außerhalb erstellt und dann als (encapsulated) PostScript (L^AT_EX) oder PDF, JPEG, PNG (pdfL^AT_EX) eingebunden.

- Einbindung hängt von Ausgabegerät ab!
- Das Paket **graphicx** erlaubt es, Grafiken einzubinden.
- Beispiel:

```
\usepackage{graphicx} ...
\begin{figure}[ort]
\includegraphics{dateiname}
\caption[kurzunterschrift]{Bildunterschrift}
\label{bildlabel} \end{figure}
```

- man kann auch Größenangaben, wie die Breite angeben:

```
\includegraphics [width=10cm]{dateiname}
```

5.2.4. Tabellen

Tabellenüberschriften stehen über der Tabelle.

```
\begin{table}[ort]
\caption[kurzunterschrift]{Tabellenuberschrift}
\label{tabellenlabel}
\begin{tabular}{spaltenbeschreibung}
Z1S1& Z1S2 & ... \\ Z2S1& Z2S2 & ... \\
\end{tabular}
\end{table}
```

Die Umgebung **tabular** setzt die eigentliche Tabelle.

Für jede Spalte ist ein Eintrag in der Spaltenbeschreibung

- c – zentrierte Spalte
- l – linksbündig
- r – rechtsbündig
- p{6cm} – Box von 6cm Breite
- z.B. `\begin{tabular}{|p{3cm}|lr|}`

nötig.

Mit dem Befehl `\multicolumn{n}{d}{c}` kann man verschiedene Spalten zu einer zusammenfassen:

```
\begin{tabular}{ll}
Foo & Bar \\
\multicolumn{2}{c}{Foobar} \\
Foo & Bar \\
\end{tabular}
```

```

      Foo  Bar
      Foobar
      Foo  Bar

```

Tabellen lassen sich auch verschachteln.

```
\begin{tabular}{ccc}
A & B & C\\
D & E & \begin{tabular}{ll}
F1 & F2\\
F & f\\
\end{tabular}\\
G & H & I
\end{tabular}
```

A	B	C
D	E	F1 F2
		F f
G	H	I

Ein | in der Spaltenbeschreibung ergibt eine vertikale Linie, um eine horizontale Linie zu erhalten dient der Befehl `\hline`:

```
\begin{table}[H]\centering
\caption{Eine Tabelle mit Excel-artigem Layout.}
\begin{tabular}{|l|l|l|l|}
\hline
Kopf1 & Kopf2 & Langer, langer, langer Kopf3 & \\\hline
Data & Data & Data & \\\hline
Data & Data & Data & \\\hline
Data & Data & Data & \\\hline
\end{tabular}
\end{table}
```

Das Ergebnis ist eher fragwürdig.

Tabelle 5.7.: Eine Tabelle mit Excel-artigem Layout.

Kopf1	Kopf2	Langer, langer, langer Kopf3
Data	Data	Data
Data	Data	Data
Data	Data	Data

Weniger Linien lenken die Aufmerksamkeit auf den Inhalt und lassen die Tabelle „leichter“ erscheinen.

Tabelle 5.8.: Eine Tabelle mit ansprechendem Layout.

Kopf1	Kopf2	Langer, langer, langer Kopf3
Data	Data	Data
Data	Data	Data
Data	Data	Data

Das Paket *booktabs* hilft bei der Umsetzung:

```
\begin{table}[H]\centering
\caption{Eine Tabelle mit ansprechendem Layout.}
\begin{tabular}{lll}
\toprule
Kopf1 & Kopf2 & \parbox{7em}{Langer , langer , langer Kopf3} \\
\midrule
Data & Data & Data \\
Data & Data & Data \\
Data & Data & Data \\
\bottomrule
\end{tabular}
\end{table}
```

Normalerweise werden Zahlen in Tabellen nicht am Dezimaltrenner ausgerichtet:

Tabelle 5.9.: Zahlen ohne Ausrichtung am Dezimaltrenner

Zahl
3,1
3,14
3,141
42,0

Abhilfe schafft das unschöne Konstrukt `r@{\,}\l`:

```
\begin{tabular}{r@{\,}l}
\toprule
\multicolumn{2}{l}{Zahl} \\
\midrule
3&1 \\
3&14 \\
3&142 \\
42&0 \\
\bottomrule
\end{tabular}
```

Zahl
3,1
3,14
3,142
42,0

Es gibt eine Vielzahl von Paketen, mit denen man Tabellen beeinflussen kann. Eine *kleine* Auswahl:

datatool zum Einlesen und Bearbeiten von CSV-Dateien.

dcolumn zum Ausrichten von Zahlen am Dezimaltrenner.

longtable zum Setzen von Tabellen über mehrere Seiten.

spreadtab zum Rechnen in Tabellen.

widetable zum Setzen von Tabellen über mehrere Seiten.

xcolor zum farbigen Gestalten von Tabellen.

5.3. Verzeichnisse

Das Inhaltsverzeichnis wird mit dem Kommando

```
\tableofcontents
```

erstellt und an dieser Stelle eingefügt.

Überschriften kommen automatisch in das Inhaltsverzeichnis

Weitere Einträge kann man mit dem Kommando

```
\addcontentsline{toc}{Eintragtext}
```

einfügen.

Die Listen der Abbildungen und Tabellen werden durch eigene Kommandos erstellt:

```
\listoffigures oder \listoftables
```

5.4. Referenzen und Verweise

5.4.1. Setzen und Benutzen von Referenzen

- Mit dem Befehl `\label{labelname}` kann man eine Markierung in einer bestimmten Stelle des Dokumentes setzen.
- Mit dem Befehl `\ref{labelname}` kann man auf diese Markierung verweisen, d.h. `\ref{labelname}` wird durch die Nummer zu dieser Markierung ersetzt.
- Mit dem Befehl `\pageref{labelname}` kann man sich auf die Seitenzahl einer bestimmten Stelle beziehen.

Beispiel

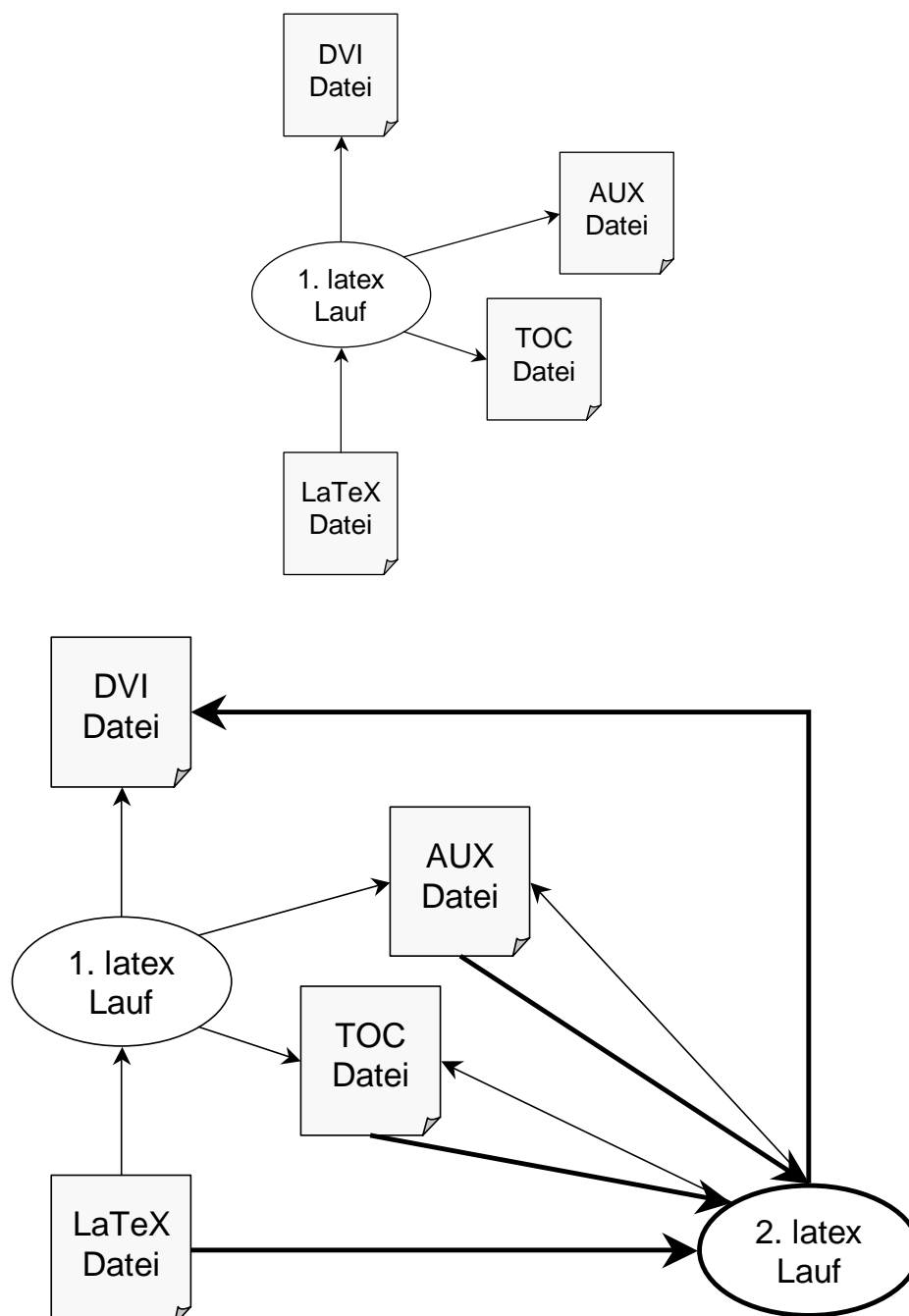
```
\chapter{Einleitung}\label{einleitung}
```

Das hier ist die Einleitung. Die Details werden in Kapitel~\ref{grundlagen} besprochen.

```
\chapter{Grundlagen}\label{grundlagen}
```

Die Details sind wie folgt: ...

5.4.2. Auflösung von Referenzen



5.4.3. Schöne Referenzen

Bindet man das Paket *varioref* ein, dann werden schöne Referenzen unterstützt:

- stellt z.B. `\vref{}` und `\vpageref{}` bereit
- Verweise gleichzeitig auf Label und Seite möglich, auch verbale Ausdrücke wie “auf der folgenden Seite” usw.
- weitere Eigenschaften: siehe Übung

5.4.4. Verweise

Bindet man das Paket *hyperref* ein, dann werden auch Verweise unterstützt (falls man PDF-Dateien erstellt):

- Verweise innerhalb eines Dokumentes (Inhaltsverzeichnis, Überschriften, Bildunterschriften, LiteraturVerweise, ...)
- Verweise auf Internet-Seiten, d.h. URLs

Einsatz:

- `\usepackage{hyperref}` in den Kopf des Dokumentes aufnehmen, möglichst als letztes Paket laden
- Verweise auf Internet-Seiten werden dann mit dem Kommando `\href[options]{URL}{text}` oder `\url{URL}` gesetzt.

5.5. Zitate und Bibliographie

In jeder wissenschaftlichen Arbeit muss die benutzte Literatur genannt werden.

- Das geschieht einmal im Text als Zitat.
- Am Ende des Dokumentes kommt dann noch eine Liste der benutzten Werke.

Das Programm *bibtex* (mittlerweile neu: *biber*) unterstützt die Erstellung von Literaturverzeichnissen.

5.5.1. Zitate

Im Text wird das Kommando `\cite{key}` mit einem Schlüssel benutzt. Dieser Schlüssel muss später auch in der Literaturdatenbank vorkommen!

Beispiel

In diesem Dokument diskutieren wir die verschiedenen Verfahren zum Sortieren von Datensätzen. Mit diesem Thema hat sich bereits Knuth `\cite{knuth1998}` sehr ausführlich beschäftigt.

Am Ende des Dokumentes sagt man dann noch, dass man ein Literaturverzeichnis braucht.

```
\bibliographystyle{plain}  
\bibliography{datenbankname1,datenbankname2, ...}
```

5.5.2. Zitierstile

- Bib_TE_X unterstützt einige Zitierstile von sich aus:
 - plain: alphabetisch sortiert und durchnummeriert, wie [1]
 - abbrv: aber Vornamen (und ein paar andere Sachen) abgekürzt
 - alpha: alphabetisch sortiert, Autornamen (abgekürzt) + Jahreszahl, wie [KNU98]
 - unsrt: wie plain, aber sortiert nach Auftreten im Text
- weitere Zitierstile kann man sich dazuladen (bst-Dateien)

5.5.3. Bib_LA_TE_X

- Mehr Flexibilität bietet das zusätzliche Package `biblatex` (hier in diesem Skript benutzt)
- bei höheren Ansprüchen an Zitierweisen bitte gleich das benutzen
- automatische Bib_TE_X/Biber-Läufe werden noch nicht von allen Editoren unterstützt

```
\usepackage[style=authoryear,babel=hyphen,autocite=inline,maxnames=7,  
            url=true,minnames=1]{biblatex}  
\addbibresource{bibfile1.bib}  
\addbibresource{bibfile2.bib}  
\addbibresource[location=remote]{http://www.citeulike.org/bibtex/group/9517}
```

5.5.4. Wörtliches Zitieren

Zum Zitieren haben wir in (Sprengel und Wohlfeil 2012) einige Regeln zusammengefasst, z.B.

Es wird zwar eher selten vorkommen, dass Sie wörtliche Zitate benutzen, wenn Sie dies jedoch tun, sollte es so aussehen, wie in Nagel (Nagel 2001) beschrieben: „Zitierte Quellen im Text müssen gekennzeichnet werden und im Literaturverzeichnis mit Verfasser, Titel, Ausgabe, Verlag, Ort und Erscheinungsjahr angegeben werden. Auch Abbildungen und Tabellen müssen mit Quellenangaben versehen werden. Die Kennzeichnung erfolgt grundsätzlich durch die Angabe der Quellennummer ...(in eckigen Klammern, Anm. der Autoren) Nach Möglichkeit sollte auf längere wörtliche Zitate verzichtet werden. Wörtliche Zitate werden durch Anführungszeichen eingeschlossen. Auslassungen innerhalb von wörtlichen Zitaten werden durch drei Punkte angezeigt. Die zugehörigen Quellen werden im Literaturverzeichnis aufgeführt.“

Im Klartext:

Wörtliche Zitate gehören in Anführungszeichen (oder anders: wie hier abgesetzt, Anm. der Autorin) und müssen mit einem Verweis auf die Literaturstelle (das Buch, die Internetseite etc.) versehen werden.

Absetzen von Zitaten beherrscht L^AT_EX auch und stellt zwei Umgebungen bereit:

- für kürzere Zitate, die sich über maximal einen Absatz erstrecken, die *quote*-Umgebung:

```
\begin{quote}
Dies ist ein Zitat.
\end{quote}
```

liefert

Dies ist ein Zitat.

- für längere Zitate die *quotation*-Umgebung

```
\begin{quotation}
Dies ist ein (nicht wirklich) langes Zitat.
\end{quotation}
```

liefert

Dies ist ein (nicht wirklich) langes Zitat.

5.5.5. Nicht-wörtliches Zitieren

Auch in (Sprengel und Wohlfeil 2012) finden Sie:

Eher häufig wird es vorkommen, dass Sie Bücher, Artikel oder Internetquellen benutzen, aber nicht wörtlich zitieren. Dann müssen Sie auch auf diese Quellenangaben verweisen.

Das sollte dann irgendwie flüssig im Text und nicht einfach mit der Nummer hinter/vor dem Abschnitt erfolgen.

Empfehlungen:

- Eine gute Einführung zum Thema findet man bei AutorXY [XY(Jahr)], worauf der folgende Abschnitt basiert.
- Wir stützen uns im Folgenden auf AutorXY [XY(Jahr)], ...
- Weiterführende Erläuterungen finden sich in AutorXY [XY(Jahr)], worauf auch die obenstehende Zusammenfassung maßgeblich basiert.
- ...

5.5.6. BibTeX

Aufbau der BibTeX-Datei

Literaturdatenbanken sind BibTeX-Dateien: der Dateiname muss auf .bib enden.

Für die meisten Literaturtypen gibt es eigene Eintragstypen in der Datei:

- Bücher `book`
- Artikel `article`
- Bachelor/Master- oder Doktorarbeit `thesis`

Beispiel für einen Bucheintrag:

```
@Book{knuth1998,  
  author = {Donald Ervin Knuth},  
  title = {The Art of Computer Programming},  
  publisher = {Addison-Wesley Longman},  
  year = {1998}  
}
```

Besonderheiten

Der BibTeX-Schlüssel darf keine Sonderzeichen (ä, ö, ü, ß, Ä, Ö, Ü, _) enthalten!

Machen Sie sich ein Schema, z.B. ***nameJahr*** oder z.B.

- Schema: Erster Buchstabe des Autors “-” Erste Buchstaben des Titels “-” Jahreszahl
- Beispiel: Wohlfeil, IT-Sicherheit; 2005 ergibt “w-is-2005”

Im Titel werden Großbuchstaben automatisch durch Kleinbuchstaben ersetzt, außer:

- Der erste Buchstabe des Titels bleibt, wie er ist.
- Ein Buchstabe steht in geschweiften {K}lammern
- im Titel von Büchern

Mehrere Autoren eines Werkes werden nicht durch Komma, sondern durch das Schlüsselwort **and** getrennt!

Erstellen des Literaturverzeichnisses

Es sind mehrere Schritte nötig:

1. **latex dokument.tex** erstellt die Hilfsdateien anhand der `\cite{key}` Kommandos im Dokument.
2. **bibtex dokument** erstellt aus den Hilfsdateien und der Literaturdatenbank das Literaturverzeichnis.
3. **latex dokument.tex** bindet das Literaturverzeichnis in das Dokument ein.
4. **latex dokument.tex** ist evtl. erforderlich, damit alle Referenzen stimmen.

An Stelle von latex kann natürlich auch pdflatex benutzt werden.

Achtung: nur Literatur, die im Text mit `\cite{key}` oder auch `\nocite{key}` als „benutzt“ gekennzeichnet ist, taucht auch im Literaturverzeichnis auf!.

Erstellen der Literaturdatenbank

Erstellen/Erzeugen kann man bibtex-Dateien auf viele Arten

- Manuell mit beliebigem Editor.
 - texlipse (eclipse plugin): verschiedene Templates mit Autovervollständigung
 - (x)emacs: gute Unterstützung, verschiedene Templates klickbar, Sortierfunktion
 - Spezielle Programme benutzen, die diese Dateien erstellen und verwalten können, z.B. **JabRef**
 - In Java implementiert, so dass es unter Linux/Windows/MacOS eingesetzt werden kann (in aktuellen ubuntu-Distributionen enthalten)
 - Hat graphische Oberfläche.
 - Kennt die wichtigen Felder für jeden Eintragstyp.
 - Anbindung an ACM, IEEE, u.a. Web-Suchen
- oder **Mendeley** (siehe erste Vorlesung)

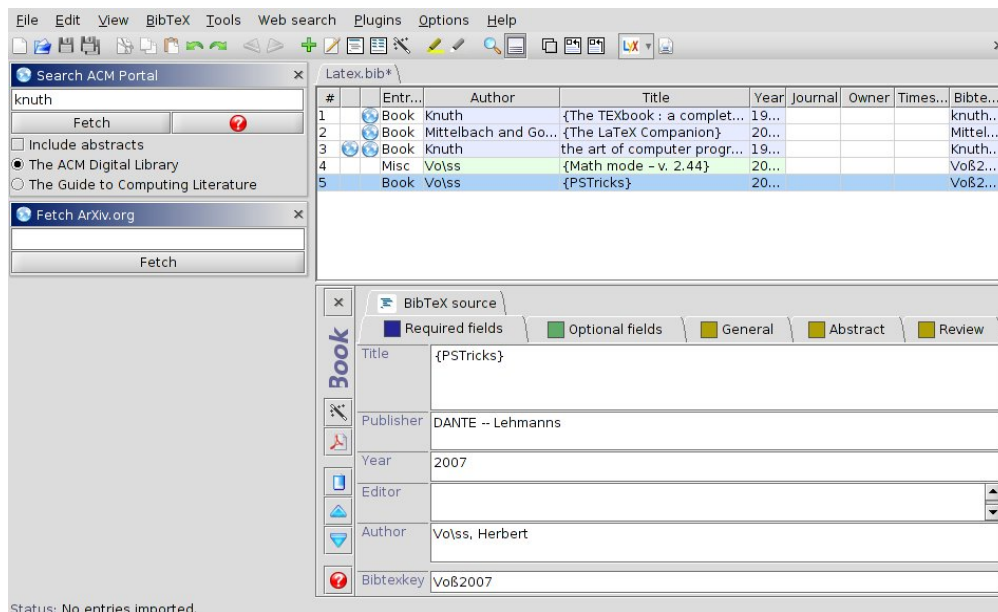


Abbildung 5.1.: Verwaltung der Einträge einer BibTeX-Datei z.B. in JabRef

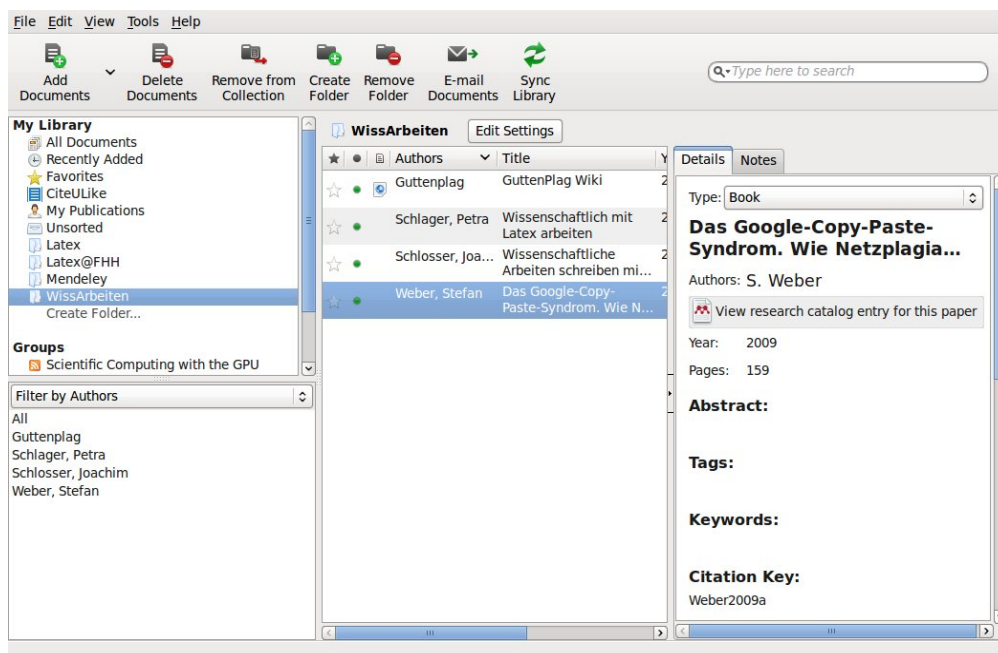


Abbildung 5.2.: Verwaltung der Einträge einer BibTeX-Datei z.B. im Mendeley Desktop

Literaturtypen

- Bücher (engl. *book*):
 - Obligatorische Attribute: Autor, Titel, Jahr, Verlag
 - Optionale Attribute: Verlagsanschrift, Serie, Auflage, Monat

- Beispiele:

Ian Sommerville; *Software-Engineering*; Pearson Studium; 2001

Ian Sommerville; *Software-Engineering*; 2nd Edition; Pearson Studium; München, Deutschland; 2001
- (Zeitschriften-)Artikel (engl. *article*):
 - Obligatorische Attribute: Autor, Titel, Zeitschriftentitel, Jahr
 - Optionale Attribute: Ausgaben-Nr, Monat, Seiten
 - Beispiele:

Stephen Newcomb et. al.: The HyTime. *Communications of the ACM*, 1991

Stephen Newcomb et. al.: The HyTime. *Communications of the ACM*, 34(11):67-83, Nov. 1991
- Internet-Quellen
 - Obligatorische Attribute: Autor, Titel, Jahr, URL
 - Beispiel:

Robin Fairbairns; TeX frequently asked questions; 2002; <http://www.tex.ac.uk/faq/>

Als BibTeX entwickelt wurde, gab es das WWW noch nicht. Deshalb gibt es keinen expliziten Literaturtyp für WWW-Dokumente. Man nimmt statt dessen den Typ "Verschiedenes". Damit die URLs anklickbar werden können, das Paket hyperref nicht vergessen.

Beispiel:

```
@Misc{UK:FAQ,
editor = {Robin Fairbairns},
title = {TeX Frequently Asked Questions on the Web},
year = 2002,
url = {http://www.tex.ac.uk/faq/} }
```

5.6. MakeIndex

Zum Erstellen eines Index sind auch mehrere Schritte nötig:

- In der Präambel des Dokumentes müssen mit dem Kommando `\makeindex` die später folgenden Index-Kommandos aktiviert werden.
- Im L^AT_EX-Dokument wird mit dem Kommando


```
\index{Indexbegriff}
```

 ein Begriff zur Aufnahme in den Index markiert.
- Am Ende des Dokumentes baut man den Index in das Dokument ein:


```
\input{dokument.ind}
```

- Das Programm ***makeindex*** erstellt dann aus den Hilfsdateien den index (dokument.ind), der anschließend durch einen erneuten L^AT_EX-Lauf in das Dokument eingefügt wird.

5.7. Zusammenfassung

- Mathematiksatz von L^AT_EX ist mittlerweile Quasi-Standard
- Gleitobjekte werden platzsparend positioniert
- Erstellung von Verzeichnissen vollautomatisch
- gute Unterstützung bei Referenzen und Links
- gute Unterstützung von Zitaten
- BibT_EX für Literaturdatenbanken
- MakeIndex zur Indexerstellung

6. Nützliche Pakete

Inhalt

- 6.1. Listings
- 6.2. Ausgefeilte Kopf- und Fußzeilen
- 6.3. Verschiedene Eingabecodierungen
- 6.4. Pstricks und Tikz
- 6.5. ModernCV
- 6.6. Beamer
- 6.7. Zusammenfassung

6.1. Listings

- Um die Quelltexte von eigenen Programmen in ein \LaTeX Dokument einzufügen existiert das Paket *listings*.
- Quelltexte können:
 - Direkt in der \LaTeX -Datei in der Umgebung `lstlisting` stehen.
 - Aus einer externen Datei in das LaTeX-Dokument eingebunden werden (vergleiche `#include`).
- Erkennen von Schlüsselwörtern, Kommentaren, Strings, Bezeichnern der Programmiersprache und entsprechende typographische Behandlung.
- Änderung der Typographie für Schlüsselwörter, Bezeichner, Kommentare, Strings ist durch einfache Deklaration möglich.

```
\lstset { basicstyle=\small, commentstyle=\color{white}}
```

- Numerierung der Zeilen

```
\lstset {numbers=left, numberstyle=\tiny, stepnumber=2, numbersep=5pt}
```

- Ausschnitt aus Dateien einbinden

```
\lstinputlisting [ firstline =3,lastline=13]{dateiname.java}
```

- Man kann eigene Stile für neue Programmiersprachen schreiben. Das braucht man eher selten, da für die meisten Sprachen bereits Stile existieren.

- Man kann Rahmen um ein Listing setzen, Referenzen auf Zeilen eines Listings aus dem normalen Text heraus benutzen.
- Für kurze Code-Ausschnitt kann das Kommando `\lstinline` benutzt werden. Es fügt Quellcode direkt in einen Absatz ein.

Beispiel

Die Deklaration `\lstinline !char **i [];! ist nicht trivial .`

Ausgabe:

Die Deklaration `char **i [];` ist nicht trivial.

- Ausführliche Dokumentation des Paketes existiert.

Beispiel : Ausschnitt aus einem C-Quelltext

Eingabe:

```
\lstinputlisting[ basicstyle=\small, numbers=left ,
stepnumber=2, firstline=147, lastline=156,
language=C]{ nullstelle .c}
```

Ausgabe:

```
1 double newton(double (*funk)(double),
                double (*abl)(double),
3             double x, double eps)
// Newtonverfahren mit Start x fuer Funktion funk
5 // mit Ableitung abl bis zur Genauigkeit eps
{
7     double xalt=1e+10;
    int it=0;
9
    while (fabs(x-xalt)>eps && it < maxit)
```

6.2. Ausgefeilte Kopf- und Fußzeilen

- Möchte man spezielle Kopf- und Fußzeilen definieren, dann gibt es hierfür zwei Pakete:
 - ***fancyhdr*** (nicht ***fancyheadings***, das ist veraltet!!!). Diese Paket kann mit jeder Dokumentklasse zusammen verwendet werden.
 - ***scrpage2*** (nicht ***scrpage***, das ist veraltet!!!). Dieses Paket gehört zu KOMA-Script und passt deshalb besonders zu den dort definierten Klassen (scrreprt, scrartcl, scrbook).
- Allgemeine Hinweise zu den L^AT_EX-Seitenstilen
 - ***Empty***: Keine Kopfzeile, keine Fußzeile. Beispiel: Erste Seite eines Kapitels

- **Plain:** Keine Kopfzeile, Fußzeile mit Seitenzahl in der Mitte. Beispiel: Einfache Artikel
- **Headings:** Kopfzeile und Fußzeile mit Seitennummer im Kopf, Kapitel und Abschnittsüberschriften im Kopf Beispiel: Bücher
- **Myheadings:** Selbst definierbare Kopf- und Fußzeilen

Beispiel : Kopf-/Fußzeilen in KOMA-Script mit scrpage2

```
\ohead{\pagemark} \rehead{Jahresbericht 2001} \lohead{\headmark}
\cefoot{Firma WasWeissIch} \cofoot{Abteilung Entwicklung}}
```

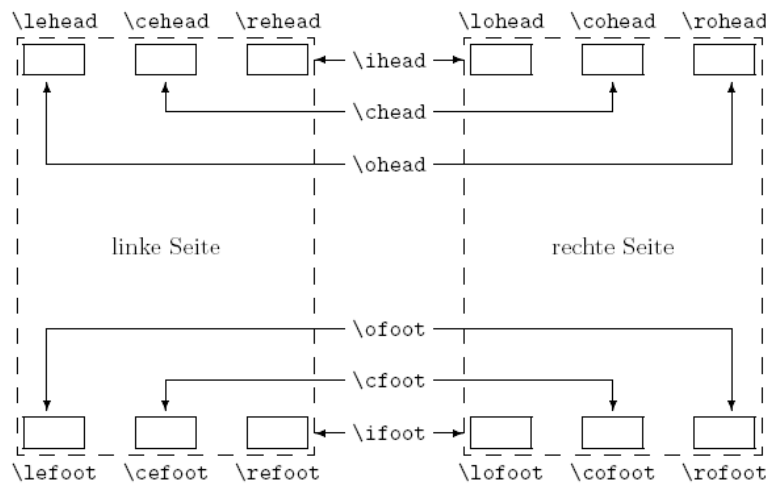


Abbildung 6.1.: Ergebnis mit scrpage2

Beispiel : Kopf-/Fußzeilen in KOMA-Script mit scrpage2

Die Kopf- und Fußzeilen dieses Vorlesungsskripts sind gemacht mit:

```
\defpagestyle{meinstil}{%
{\headmark \hfill}
{\hfill \headmark}
{\hfill \headmark \hfill }
(\textwidth,.4 pt)
}%
(\textwidth,.4 pt)
{\pagemark \hfill \today}
{Frauke Sprengel, Wissenschaftliche
Arbeiten mit \LaTeX \hfill \pagemark}
{\today \hfill \pagemark}
}
\pagestyle{meinstil}
```

6.3. Verschiedene Eingabecodierungen

- Als $\text{T}_{\text{E}}\text{X}$ und $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ entwickelt wurden, gab es noch kein Unicode und die Entwickler sahen ASCII als Standard-Codierung vor.
 - Einfache und portierbare Dokumente
 - Umlaute müssen umständlich eingegeben werden: `\"a`, `\"o`, `\"U`, usw.
 - Problematisch bei automatischer Silbentrennung.
- Das Paket ***inputenc*** erlaubt es, erweiterte Zeichensatzcodierungen in einem Dokument zu benutzen.
- Eingabe:

```
\usepackage[utf8]{inputenc}...  
\begin{document}  
Hier dürfen utf8 codierte Umlaute stehen.
```

Andere mögliche Eingabecodierungen wären z. B.:

latin1 für Windows.

applemac für Mac OS bis Version 9.

Das Problem: $\text{T}_{\text{E}}\text{X}$ bildet die Eingabekodierung auf die eigene Kodierung ab. Symbole die $\text{T}_{\text{E}}\text{X}$ nicht kennt, können also nicht angezeigt werden, auch wenn sie in der verwendeten Schrift vorhanden sind.

Die Lösung: Neuere $\text{T}_{\text{E}}\text{X}$ -Maschinen als $\text{pdfT}_{\text{E}}\text{X}$ unterstützen nativ UTF-8:

- $\text{luaT}_{\text{E}}\text{X}$ (designierter Nachfolger von pdfTeX)
- $\text{XeT}_{\text{E}}\text{X}$

6.4. Pstricks und Tikz

Die Pakete ***pstricks*** und ***tikz*** und ihre diversen Zusatzpakete bieten beide viele Möglichkeiten, um direkt in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ zu zeichnen, zu positionieren usw.

Die meisten AutorInnen verwenden nur eine der beiden Möglichkeiten (das ist fast eine Glaubensfrage). Manches geht in der einen, manches in der anderen Variante besser.

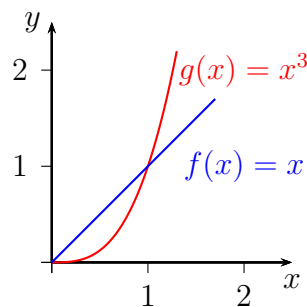
6.4.1. Pstricks

PStricks ist etwas älter und extrem gut dokumentiert, es wird immer erst PostScript produziert, mit dem Paket ***auto-pst-pdf*** ist allerdings auch die Verwendung direkt mit pdf_latex möglich (es wird zwischendurch noch ein normaler latex-Lauf eingeschoben.) Dann muss pdf_latex mit der Option `--shell-escape` benutzt werden.

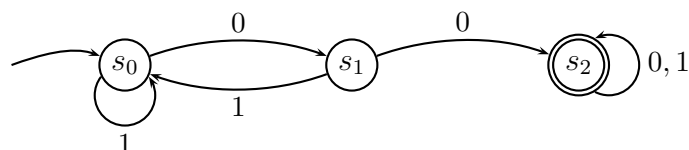
Die Folien der ersten Vorlesung zum Pyramiden-Prinzip haben Pstricks verwendet. Es gibt Zusatzpakete für alles Mögliche, z.B.

- ***pst-tree*** - für Bäume
- ***pst-node*** - für Graphen, allg. Verbindungen von Knoten, endliche Automaten
- ***pst-plot*** - für Funktionsgraphen
- ***pst-3D*** - für Zeichnungen in 3D
- ...

Beispiele



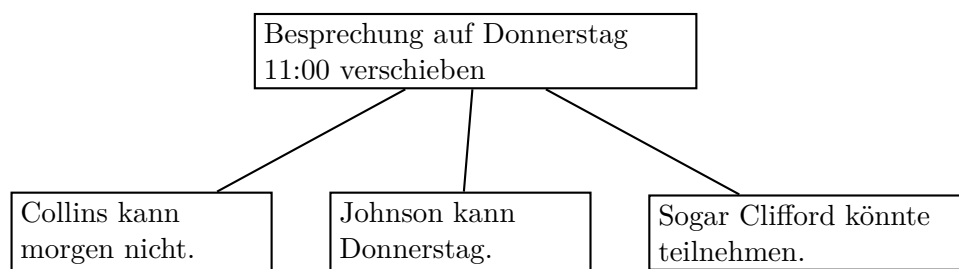
```
\psset{unit=7ex}
\begin{pspicture}(-0.5,-0.5)(2.5,2.5)
\psaxes[showorigin=false,tickstyle=bottom,
        labelsep=5pt]{->}(2.5,2.5)
\rput(2.5,-.2){$x$}
\rput(-.2,2.5){$y$}
\rput(2.0,2.0){\color{red}$g(x)=x^3$}
\psplot[plotpoints=200,linecolor=red]{0.0}{1.3}{x^3 exp}
\rput(2.0,1.0){\color{blue}$f(x)=x$}
\psplot[plotpoints=200,linecolor=blue]{0.0}{1.7}{x}
\end{pspicture}
```



```

\psset{arrows=->, unit=3cm, arcangle=20,
  labelsep=2.5pt, subgriddiv=10, shortput=nab}
\begin{pspicture}(-0.5,-0.4)(2.5,0.25)
  \pnode(-0.5,0){0} %Leerer Node fuer Pfeil zum Ausgangszustand
  % die Nodes
  \cnodeput(0,0){s0}{s_0$}
  \cnodeput(1,0){s1}{s_1$}
  \cnodeput[doubleline=true](2,0){s2}{s_2$}
  % die Edges
  {\small
    \ncarc{0}{s0} % -> zum Startzustand
    \ncarc{s0}{s1}^{$0$}
    \ncarc{s1}{s2}^{$0$}
    \ncarc{s1}{s0}^{$1$}
  }
  % Loops
  \nccircle[angle=180]{s0}{0.4cm}_{$1$} % nach unten
  \nccircle[angle=-90]{s2}{0.4cm}_{$0,1$} % nach rechts
}
\end{pspicture}

```



```

\pstree[levelsep=0.15\textwidth]{%
\Tr{\psframebox{\parbox{0.35\textwidth}{\raggedright\small
  Besprechung auf Donnerstag 11:00 verschieben}}}{%
\Tr{\psframebox{\parbox{0.2\textwidth}{\raggedright\small%
  Collins kann morgen nicht.}}}%
\Tr{\psframebox{\parbox{0.2\textwidth}{\raggedright\small%
  Johnson kann Donnerstag.}}}%
\Tr{\psframebox{\parbox{0.25\textwidth}{\raggedright\small%
  Sogar Clifford k"onnte teilnehmen.}}}%
}

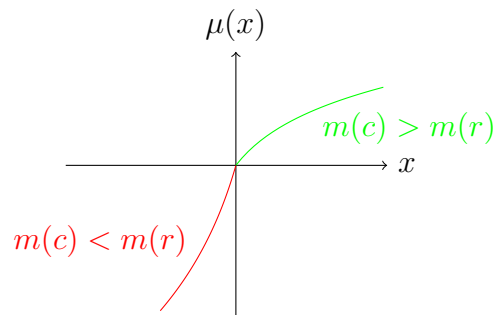
```

6.4.2. TikZ/Pgfplots

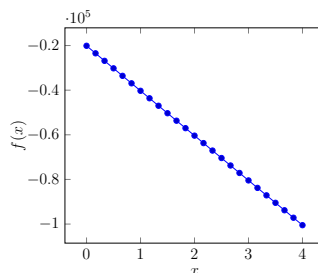
TikZ ist neuer als Pstricks, hat aber auch ein umfangreiches Manual und benötigt den Umweg über PostScript nicht. Für Plots kann gnuplot verwendet werden.

Auch hier muss pdflatex mit der Option `--shell-escape` benutzt werden.

Beispiele



```
\begin{tikzpicture}[domain=-5:3,xscale=0.5,yscale=0.5]
\draw[very thin,color=gray] (0.0,0.0) grid (0,0);
\draw[->] (-4.5,0) — (4.0,0) node[right] {$x$};
\draw[->] (0,-4.0) — (0,3.0) node[above] {$\mu(x)$};
\draw[green,domain=0.0:3.9] plot (\x,{1.3*ln(\x+1)});
\draw (2,1) node[right,green] {$m(c)>m(r)$};
\draw[red,smooth,samples=100,domain=-2.0:0.0]
    plot(\x,{ -3.5*ln(-\x+1)});
\draw (-1,-2) node[red,left] {$m(c)<m(r)$};
\end{tikzpicture}
```



```
\begin{tikzpicture}[xscale=0.5,yscale=0.5]
\begin{axis}[domain=0:4, xlabel={$x$}, ylabel={$f(x)$}]
\addplot gnuplot {1.0064925+1.005*x-(sqrt(1.011035+1.005*x**2+2.04106*x))/0.0000};
\end{axis}
\end{tikzpicture}
```



Abbildung 6.2.: Map of scientific interactions (Andrei Sobolevski <http://texample.net>)

6.5. ModernCV

Auch Lebensläufe lassen sich elegant mit \LaTeX erstellen.

Es gibt viele Pakete, zum Thema Lebenslauf mit (La) \TeX : `currvita`, `curve`, `cv`, `cvsty`, `ecv`, `esieecv`, `europcv`, `koma-moderncvclassic`, `<2>moderncv`, `moderntimeline`, `mycv`, `res`, `resumemac`, `simplecv`, `translation-ecv-de`, `translation-europcv-de`, `tucv`, `vita` und `vita209`.

`moderncv` ist eine flexible und hübsche Klasse und soll daher hier vorgestellt werden.

Der Nachteil: Es gibt **keine Dokumentation**. Stattdessen gibt es Beispiele auf CTAN.¹

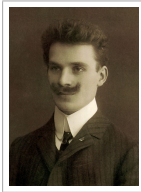
Der Quelltext für den Lebenslauf aus Abb. 6.3 sieht folgendermaßen aus:

```
\documentclass{moderncv}
\usepackage[utf8]{inputenc}

\firstname{Karl}
\familyname{Ranseier}
\address{Gesmannstr. 77}{30459 Hannover}
\phone{0511 5558485}
\email{k.ranseier@fictitious.example}
\photo[64pt]{photo}

\begin{document}
\maketitle
```

¹<http://ctan.org/tex-archive/macros/latex/contrib/moderncv/examples>



Karl Ranseier

Ausbildung

Schulbildung

1996–2000 **Grundschulabschluss**, *Frauke-Sprengel-Schule*, Hannover, Notenschnitt 2,3.
Klassensprecher, Klasse 4

2009–2009 **Abitur**, *Mierscheid Gesamtschule*, Hannover, Notenschnitt 1,7.
Klassensprecher, Klassen 5–13; Administrator der Schulrechner

Studium

2009–2014 **Master of Science, Angewandte Informatik**, *Hochschule Hannover*, Hannover,
Abschlussnote 1,3.
Mitglied im FSR und StuPa

Master-Arbeit

Information
Visualisation in
the 21st Century

Arbeit zum Thema 3D-Modellierung mit luaTeX zur Wiedergabe auf 3D-fähigen eInk-Displays.

Sprachen

Deutsch **Muttersprachler**

Englisch **CPE grade A**

Zahlreiche mehrwöchige Auslandsaufenthalte in USA, Australien, Irland, UK; Auslandssemester 2012 Oxford, UK

Qualifikationen

- LPI-304
- Führerschein Klasse B

Gesmannstr. 77 – 30459 Hannover
0511 5558485 • k.ranseier@fictitious.example

Abbildung 6.3.: Lebenslauf mit moderncv

```
\section{Ausbildung}
\subsection{Schulbildung}
\cventry{1990--1994}{Grundschulabschluss}%
{Frauke-Sprengel-Schule}{Hannover}{Notenschnitt
2,3}{Klassensprecher, Klasse 4}
\cventry{1994--2009}{Abitur}{Mierscheid
Gesamtschule}{Hannover}{Notenschnitt 1,7}%
{Klassensprecher, Klassen 5--13; Administrator der
Schulrechner}

\subsection{Studium}
\cventry{2009--2014}{Master of Science, Angewandte
Informatik}{Hochschule Hannover}{Hannover}%
{Abschlussnote 1,3}{Mitglied im FSR und StuPa}

\section{Master-Arbeit}
\cvline{Information Visualisation in the 21st
Century}{Arbeit zum Thema 3D-Modellierung mit
lua\TeX\ zur Wiedergabe auf 3D-f\ahigen
eInk-Displays.}

\section{Sprachen}
\cvlanguage{Deutsch}{Muttersprachler}{}
\cvlanguage{Englisch}{CPE grade A}{Zahlreiche
mehr\ochige Auslandsaufenthalte in USA,
Australien, Irland, UK; Auslandssemester 2012
Oxford, UK}

\section{Qualifikationen}
\cvlistitem{LPI-304}
\cvlistitem{F\uhrerschein Klasse B}

\end{document}
```

6.6. Beamer

Mit der *beamer*-Klasse gibt es eine sehr mächtige Möglichkeit für alle Arten von Präsentationen. Mit *powerdot* gibt es noch (mindestens) eine Alternative, die auf PSTricks basiert und angeblich noch mehr Layouts ermöglicht.

```
\documentclass{beamer}
\begin{document}
\begin{frame}{Hello World}
  Dies ist ein Hello Word — Beispiel !
\end{frame}
\end{document}
```

liefert

Hello World

Dies ist ein Hello Word – Beispiel !

Navigation icons

```
\documentclass{beamer}
\usepackage[ngerman]{babel}\usepackage{lmodern}
\usepackage[utf8]{inputenc}\usepackage[T1]{fontenc}

\setbeamertemplate{background canvas}[vertical
  shading][bottom=white,top=structure.fg!25]
\usetheme{Warsaw}
\setbeamercovered{transparent}

\title{Mein Vortrag}
\author{Frauke Sprengel}
\institute{FH Hannover}
\AtBeginSection{
  \begin{frame}<beamer>{Inhalt}
    \tableofcontents[currentsection,hideothersubsections]
  \end{frame}
}

\begin{document}

\begin{frame}
  \titlepage
\end{frame}

\begin{frame}{Inhalt}
  \tableofcontents[hideallsubsections]
\end{frame}

\section{Einleitung}
\begin{frame}{Hello World}
  Dies ist ein Hello Word — Beispiel !
\end{frame}

\section{Noch was}
\subsection{Ein Unterpunkt}
```

```

\begin{frame}[fragile]{\ "Uberschriften sollen informativ sein.\
  Korrekte Gro{\ss}–/Kleinschreibung beachten.}{ Untertitel sind optional.}

\begin{itemize}
\item
  Viel \texttt{itemize} benutzen.
\item
  Sehr kurze Sätze oder Satzglieder verwenden.
\end{itemize}
\end{frame}

\subsection{Zweiter Unterpunkt}

\begin{frame}{\ "Uberschriften sollen informativ sein.}

Man kann Overlays erzeugen \dots
\begin{itemize}
\item mit dem \texttt{pause}–Befehl:
  \begin{itemize}
\item
  Erster Punkt.
  \pause
\item
  Zweiter Punkt.
\end{itemize}
\item
  mittels Overlay–Spezifikationen:
  \begin{itemize}
\item<3->
  Erster Punkt.
\item<4->
  Zweiter Punkt.
\end{itemize}
\item
  mit dem allgemeinen \texttt{uncover}–Befehl:
  \begin{itemize}
\item
    \uncover<5->{\item
      Erster Punkt.}
\item
    \uncover<6->{\item
      Zweiter Punkt.}
\end{itemize}
\end{itemize}
\end{frame}

\end{document}

```

liefert

Einleitung
Noch was

Mein Vortrag

Frauke Sprengel
FH Hannover
3. Mai 2011

Frauke Sprengel Mein Vortrag

Einleitung
Noch was

Inhalt

- 1 Einleitung
- 2 Noch was

Frauke Sprengel Mein Vortrag

Einleitung
Noch was

Inhalt

- 1 Einleitung
- 2 Noch was

Frauke Sprengel Mein Vortrag

Einleitung
Noch was

Hello World

Dies ist ein Hello Word – Beispiel !

Frauke Sprengel Mein Vortrag

Einleitung
Noch was
Ein Unterpunkt
Zweiter Unterpunkt

Inhalt

- 1 Einleitung
- 2 Noch was
 - Ein Unterpunkt
 - Zweiter Unterpunkt

Frauke Sprengel Mein Vortrag

Einleitung
Noch was
Ein Unterpunkt
Zweiter Unterpunkt

Überschriften sollen informativ sein. Korrekte Groß-/Kleinschreibung beachten. Untertitel sind optional.

- Viel itemize benutzen.
- Sehr kurze Sätze oder Satzglieder verwenden.

Frauke Sprengel Mein Vortrag

Einleitung Noch was Ein Unterpunkt Zweiter Unterpunkt

Überschriften sollen informativ sein.

Man kann Overlays erzeugen...

- mit dem `pause`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
 - Erster Punkt.
 - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.

Frauke Sprengel Mein Vortrag

Einleitung Noch was Ein Unterpunkt Zweiter Unterpunkt

Überschriften sollen informativ sein.

Man kann Overlays erzeugen...

- mit dem `pause`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
 - Erster Punkt.
 - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.

Frauke Sprengel Mein Vortrag

Einleitung Noch was Ein Unterpunkt Zweiter Unterpunkt

Überschriften sollen informativ sein.

Man kann Overlays erzeugen...

- mit dem `pause`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
 - Erster Punkt.
 - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.

Frauke Sprengel Mein Vortrag

Einleitung Noch was Ein Unterpunkt Zweiter Unterpunkt

Überschriften sollen informativ sein.

Man kann Overlays erzeugen...

- mit dem `pause`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
 - Erster Punkt.
 - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.

Frauke Sprengel Mein Vortrag

Einleitung Noch was Ein Unterpunkt Zweiter Unterpunkt

Überschriften sollen informativ sein.

Man kann Overlays erzeugen...

- mit dem `pause`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
 - Erster Punkt.
 - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.

Frauke Sprengel Mein Vortrag

Einleitung Noch was Ein Unterpunkt Zweiter Unterpunkt

Überschriften sollen informativ sein.

Man kann Overlays erzeugen...

- mit dem `pause`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.
- mittels Overlay-Spezifikationen:
 - Erster Punkt.
 - Zweiter Punkt.
- mit dem allgemeinen `uncover`-Befehl:
 - Erster Punkt.
 - Zweiter Punkt.

Frauke Sprengel Mein Vortrag

Es gibt eine sehr ausführliche Dokumentation zur ***Beamer***-Klasse, man kann Hintergründe ändern, Themen, Fonts, Overlays und noch viel mehr.

Benutzt man statt beamer

```
\documentclass{article}  
\usepackage{beamerarticle}
```

kann man aus seinen Folien auch gleich einen Artikel machen (das kennen Sie von meinen Skripten).

6.7. Zusammenfassung

Es gibt viele nützliche Pakete. Einige davon sind:

- Listings
- scrpage2
- TikZ/pgf und PSTricks
- moderncv
- Beamer

7. Eigene Kommandos, Dokumentenstruktur und externe Programme

Inhalt

- 7.1. Dokumentenstruktur
- 7.2. Zähler
- 7.3. Eigene Kommandos und Umgebungen
- 7.4. Eigene Klassen und Packages
- 7.5. (pdf)L^AT_EX auf der Kommandozeile
- 7.6. Externe Programme
- 7.7. Zusammenfassung

7.1. Dokumentenstruktur

7.1.1. Aufteilung von Dokumenten mit input und include

Große Dokumente können auf verschiedene Dateien aufgeteilt werden. In einem Hauptdokument werden die Teildokumente dann zusammengebaut:

```
\documentclass[12pt]{book}
% Packages einbinden, eigene Makros
% definieren, etc.
\begin{document}
\input{einleitung.tex}
\include{hauptteil.tex}
\include{schluss teil.tex}
\appendix
\input{anhang.tex}
% Bibliography-Kommandos, Index Kommandos, etc.
\end{document}
```

Die entsprechende Datei wird an der Stelle des `\input`- bzw. `\include`-Befehls eingefügt.

7.1.2. Unterschiede zwischen `include` und `input`

Manchmal ist es egal, ob man `\input` oder `\include` verwendet, aber es gibt Unterschiede:

- `\input` ist beliebig tief schachtelbar, `\include` nicht
- `\include` erzeugt an der Stelle des Befehls einen Seitenumbruch
- mit `\includeonly{dateiname}` kann eingestellt werden, dass nur diese Datei gerade eingebunden wird (Referenzen nach außerhalb funktionieren trotzdem), das kann bei längeren Dokumenten hilfreich sein
- **Merkregel:**

Im Main-Dokument mit `\include` für die Kapitel arbeiten (bei `report` oder `book`), ansonsten `\input` verwenden.

7.1.3. Unterschiedliche Behandlung von Einleitung, Hauptteil, Schluss

Für die Klasse *scrbook* kann man folgende Einteilung vornehmen

<code>\frontmatter</code> Vorspann <code>\mainmatter</code> Hauptteil <code>\backmatter</code> Nachspann

Vorspann

Der Vorspann wird mit `\frontmatter` eingeleitet. Die Auswirkungen sind:

- römische Seitenzahlen
- Kapitelüberschriften nicht nummeriert
- besser keine Unterabschnitte verwenden (sonst komische Nummerierung)

Hier hinein gehören (wenn vorhanden)

- Titelei
- Inhaltsverzeichnis
- Abbildungsverzeichnis
- Tabellenverzeichnis
- Verzeichnis der Listings
- Vorwort

Hauptteil

Der Hauptteil wird mit `\mainmatter` eingeleitet (kann entfallen, wenn es keinen Vorspann gibt). Die Auswirkungen sind:

- arabische Seitenzahlen, beginnend bei 1
- normale Abschnittsnummerierung

Nachspann

Der Nachspann wird mit `\backmatter` eingeleitet. Die Auswirkungen sind:

- keine extra Seitennummerierung (läuft weiter)
- Kapitelüberschriften nicht nummeriert
- besser keine Unterabschnitte verwenden (sonst komische Nummerierung)

Hier hinein gehören (wenn vorhanden)

- Literaturverzeichnis
- Stichwortverzeichnis
- evtl. der Anhang

7.1.4. Änderung der Seitennummerierung

Eine Änderung der Seitennummerierung ist jederzeit möglich mit `\pagenumbering{Stil}`. Ein solcher Aufruf setzt immer die Seitenzahl auf 1 zurück.

Tabelle 7.1.: Nummerierungsstile

Stil	Beispiel	Bedeutung
<code>arabic</code>	8	arabische Zahlen
<code>roman</code>	viii	kleine römische Zahlen
<code>Roman</code>	VIII	große römische Zahlen
<code>alph</code>	h	Kleinbuchstaben
<code>Alph</code>	H	Großbuchstaben

7.2. Zähler

L^AT_EX hat viele Nummerierungen (wie z.B. `section`, `page`) und nummerierte Umgebungen (wie z.B. `figure`, `table`).

Der Name der Zähler ist der gleiche wie der der Umgebung oder des Befehls:

part	subparagraph	enumi
chapter	page	enumii
section	equation	enumiii
subsection	figure	enumiv
subsubsection	table	
paragraph	footnote	

Die Zähler kann man

- ausgeben mit z.B. `\roman{counter}` als kleine römische Ziffern (Befehle entsprechen den Stilen der Nummerierung bei Seiten)
- auf einen Wert setzen mit `\setcounter{counter}{value}`
- um einen Wert erhöhen mit `\addtocounter{counter}{value}`
- um 1 erhöhen mit `\stepcounter{counter}`
- neu definieren mit `\newcounter{counter}` und diesen dann benutzen

7.3. Eigene Kommandos und Umgebungen

7.3.1. Eigene Kommandos

Definition

- Man kann zu den bereits definierten Kommandos (Makros) in \LaTeX auch eigene Kommandos schreiben (vergleichbar zum Schreiben eigener Methoden in Java).
- **Syntax:** `\newcommand{\befehl}{Inhalt}`
- **Voraussetzung:** Das neu zu definierende Kommando darf noch nicht existieren!
- **Bestandteile:** Der Inhalt kann aus
 - normalem Text oder
 - anderen \LaTeX Kommandos bestehen.
- **Beispiel:**

```
\newcommand{\z}{\textbf{kleine Zwerge}}
\newcommand{\E}{\textit{mittlere Elben}}
Auf ihrer Reise trafen sie auch \z{} und \E.
```

Vorteile

- Spart Tipp-Arbeit, wenn lange Texte durch kurze Kommandos ersetzt werden.
 - Beispiel:


```
\newcommand{\myF}{Meine ganz eigene, spezielle Firma}
```
- Vereinfacht spätere Änderungen, wenn ein bestimmter Text später durch einen anderen Text ersetzt werden soll.
 - Beispiel: `\newcommand{\myF}{Meine ganz eigene, besondere Firma}`
- Stellt einheitliches Layout für bestimmte Bestandteile sicher.
 - Beispiel:


```
\newcommand{\mLin}{\textit{LINUX}}
\newcommand{\mWin}{\textit{Windows}}
```
- Ermöglichen logisches Markup im Dokument, indem spezielle Bestandteile durch sinnvolle (Kommando-)Namen ausgedrückt werden können.

Kommandos mit Parametern

- Mehr Flexibilität erreicht man mit Parametern in Kommandos:
- **Syntax:** `\newcommand{\befehl}[paramzahl]{Inhalt}`
- **Voraussetzungen:**
 - Das neu zu definierende Kommando darf noch nicht existieren!
 - Zahl der Parameter darf maximal neun sein.
- **Bestandteile:** Der Inhalt kann aus
 - normalem Text oder
 - Wert der aktuellen Parameter oder
 - anderen \LaTeX Kommandos bestehen.
- **Beispiel:**

```
\newcommand{\fachbegriff}[1]{\emph{#1}}
Unter \fachbegriff{thrashing} versteht man ...
```

7.3.2. Eigene Umgebungen

- **Syntax:** `\newenvironment{umgeb}[paramzahl]{begin-Inhalt}{ende-Inhalt}`
- **Voraussetzungen:**
 - Die neu zu definierende Umgebung darf noch nicht existieren!
 - Zahl der Parameter darf maximal neun (Verwendung nur im begin-Inhalt) sein.
- **Bestandteile:** Der Inhalt kann aus
 - normalem Text oder
 - Wert der aktuellen Parameter oder
 - anderen L^AT_EX Kommandos bestehen.
- **Semantik:**
 - Eine neue Umgebung `umgeb` wird erzeugt.
 - Jedes `\begin{umgeb}` wird später durch den begin-Inhalt ersetzt.
 - Jedes `\end{umgeb}` wird später durch den ende-Inhalt ersetzt.

Beispiel

```
\newenvironment{aufgabe}[2]{%
\noindent\textbf{Aufgabe #1:}\hfill (#2 Punkte)\par%
\smallskip\noindent}{
\mbox{}\hfill X\par\bigskip}
...
\begin{aufgabe}[1]{20}%
Das Verschlüsselungsverfahren AES basiert auf
Runden.
Erklären Sie kurz, was in einer Runde passiert.
\end{aufgabe}
...
```

Beispiel mit Zähler

```
\newcounter{aufg}
\setcounter{aufg}{1}
\newenvironment{aufgabe}[1]{%
\noindent\textbf{Aufgabe \arabic{aufg}:}\hfill %
(#1 Punkte)\par%
\smallskip\noindent\stepcounter{aufg}{
\mbox{}\hfill X\par\bigskip}
}
...
\begin{aufgabe}[20]%
```


Das Verschlüsselungsverfahren AES basiert auf Runden.
Erklären Sie kurz, was in einer Runde passiert.
`\end{aufgabe}`
...

7.4. Eigene Klassen und Packages

Wie man eigene Klassen und Pakete prinzipiell baut, findet man am besten mit `texdoc clsguide` heraus. Dort findet man auch die erste zu berücksichtigende Frage:

If the commands could be used with any document class, then make them a package; and if not, then make them a class.

7.4.1. Struktur einer Klasse oder eines Paketes

Jede Klasse / Paket hat in etwa folgende Struktur

Identifikation als Klasse / Paket mit Namen und kurzer Beschreibung

Deklarationen neue Befehle werden deklariert und andere Dateien geladen

Optionen werden deklariert und ausgeführt

mehr Deklarationen im Hauptteil für neue Befehle, Variablen, Fonts, Laden anderer Dateien.

7.4.2. Identifikation

Ein Paket (Dateiname *package.sty*) identifiziert sich selber durch

`\NeedsTeXFormat{LaTeX2e} \ProvidesPackage{<package>}[<date> <information>]`
z.B.

```
\typeout{Extended Package 'fs_def' <2011-06-29>
        Frauke Sprengel}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{fs_def}[2011/06/09]
```

Eine Klasse (Dateiname *class-name.cls*) identifiziert sich selber durch

`\NeedsTeXFormat{LaTeX2e} \ProvidesClass{<class-name>}[<date> <information>]`
z.B.

```
\typeout{Combined beamer and article class
  for lectures <2012-06-06> Frauke Sprengel}
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{fs_beamerskript}[2012/06/06 Combined
  beamer and article class]
```

7.4.3. Laden von Klassen und Paketen

Ein Paket wird innerhalb einer Klassen- oder Paketdatei so geladen:

```
\RequirePackage[<options>]{<package>}[<date>]
```

also z.B.

```
\RequirePackage{fs_def}[2012/06/06]
```

Eine Klasse wird so geladen

```
\LoadClass[<options>]{<class>}[<date>]
```

z.B.

```
\LoadClass[twocolumn]{article}
```

7.4.4. Optionen

Klassen und Pakete können Optionen haben. Diese müssen zunächst deklariert werden mit

```
\DeclareOption{<option>}{<code>}
```

z.B.

```
\DeclareOption{a4paper}{%
  \setlength{\paperheight}{297mm}%
  \setlength{\paperwidth}{210mm}%
}
```

Unbekannte Optionen führen normalerweise zu Warnungen (Klassen) oder Fehlern (Pakete), das kann man ändern durch

```
\DeclareOption*{<code>}
```

z.B.

```
\DeclareOption*{%
\PassOptionsToClass{\CurrentOption}{article}%
}
```

hier werden alle unbekannten Optionen an die Klasse `article` weitergereicht. Am Schluss muss noch dafür gesorgt werden, dass alle Optionen, die deklariert sind und die beim Aufruf auch verwendet werden auch ausgeführt werden. Dies geschieht mit:

```
\ProcessOptions\relax
```

7.5. (pdf) \LaTeX auf der Kommandozeile

7.5.1. Optionen

(pdf) \LaTeX kann auf der Kommandozeile mit diversen Optionen gestartet werden. Einige wichtige sind

<code>-halt-on-error</code>	stop processing at the first error
<code>-interaction=STRING</code>	set interaction mode (<code>STRING=batchmode/nonstopmode/ scrollmode/errorstopmode</code>)
<code>-output-directory=DIR</code>	use existing DIR as the directory to write files in
<code>[-no]-shell-escape</code>	disable/enable <code>\write18{SHELL COMMAND}</code>
<code>-shell-restricted</code>	enable restricted <code>\write18</code>
<code>-synctex=NUMBER</code>	generate SyncTeX data for previewers if nonzero
<code>-help</code>	display this help and exit
<code>-version</code>	output version information and exit

Synctex (wird benötigt für Vorwärts/Rückwärtssuche tex/pdf):

- 0: kein synctex
- 1: komprimierte synctex-Datei
- -1: unkomprimierte synctex-Datei

Interaction modes

- `errorstopmode`: stoppt bei jedem Fehler, bei Fehlern im source code oder auch nicht existierenden Dateien
- `scrollmode`: stoppt nicht bei Fehlern im source code, aber bei nicht existierenden Dateien (und verlangt einen anderen Datei-Namen)
- `nonstopmode`: stoppt bei ernsten Fehlern, verlangt aber keine Eingabe
- `batchmode`: unterdrückt zusätzlich noch die Ausgabe

In allen Fällen werden alle Fehler in die log-Datei geschrieben.

7.5.2. Ein Makefile

Damit kann man sich auch ein Makefile bauen (dieses hier ist von Stefan Wohlfeil)

```
#
# Makefile fuer LaTeX-Dateien
#
# Wenn eine Datei noch nicht existiert, sagt man z.B.
# 'make bild.tex' oder # 'make art.dvi', wenn die
# Datei aber schon da ist, reicht einfach 'make'.
#
.SUFFIXES: .2ps .ps .dvi .tex .fig .pdf

# .fig-Dateien werden in TeX umgewandelt.
# Verkleinerung auf 1/4
.fig.tex:
# entweder LaTeX
#     fig2dev -Llatex -m 0.25 -l 2 $< $@
# oder PICTeX
#     fig2dev -Lpictex -m 1.00 $< $@
# oder in PostScript und TeX
#     fig2dev -L pstex -m 0.5 $< $*.ps
#     figmath $< | fig2dev -L pstex_t -m 0.5 > $*.tex

# .tex wird zu .dvi (mit LaTeX).
.tex.dvi:
    latex $<
    @if grep 'Rerun to get cross-references' $*.log >/dev/null; \
    then latex $<; fi
# Trick: weil xdvi waehrend der Ausfuehrung des latex-
# Befehls nicht mehr die # .dvi-Datei anzeigen kann,
# wird eine Kopie .ndvi angelegt, diese wird von xdvi
# angezeigt. Erst nach dem latex-Befehl wird diese
# dann aktualisiert.
    cp $@ $*.ndvi

# .tex wird .pdf (mit pdflatex).
.tex.pdf:
    pdflatex $<
    @if grep 'Rerun to get cross-references' $*.log >/dev/null; \
    then pdflatex $<; fi

# .dvi wird zu .ps (zum Drucken).
# Der Drucker sek hat 600 dpi!
.dvi.ps:
    dvips $< -o

# .ps wird zu .2ps (Zwei Seiten nebeneinander auf einem Blatt)
.ps.2ps:
    psnup -nup 2 $*.ps $*.2ps

# Default ist, alle .pdfs auf den neuesten Stand zu bringen.
all: *.pdf

bib:
    bibtex thesis

thesis.pdf:      thesis.tex normen.tex form.tex allgtips.tex \
```

```

thesis.bbl

clean:
    rm -f *.dvi *.log *.ps *.ndvi *% core

realclean:
    rm -f *.dvi *.log *.ps *.ndvi *% core *.aux

deutschspell:
    ispell -Tutf8 -t \
        -d/usr/lib/ispell/deutsch.hash $*.tex

spell:
    ispell -Tutf8 -t $*.tex

```

7.6. Externe Programme

Die Aufzählung ist natürlich nicht vollständig, wenn Sie “schöne” Programme kennen, die Ihrer Ansicht hier fehlen, können Sie mir die gern nennen.

7.6.1. Zeichenprogramme in Verbindung mit \LaTeX

Natürlich können Sie Graphiken mit jedem Werkzeug machen, das Ihnen gefällt.

Zum Darstellen von mathematischen Inhalten etwas komplizierterer Art kann man gut Computeralgebra-Systeme und Verwandte benutzen, also:

- kommerziell: Mathematica, Maple, Matlab
- frei: Maxima (bei ubuntu dabei)

Alle können *.eps oder *.pdf exportieren. Die nachfolgend vorgestellten Programme haben alle etwas gemeinsam: Sie können noch besser mit \LaTeX zusammenarbeiten, sodass der gleiche Font verwendet wird und auch Formeln in den Bildern möglich sind.

- Xfig
- Ipe
- Jpgfdraw
- gnuplot (wird nicht vorgestellt)
- Dia

Xfig

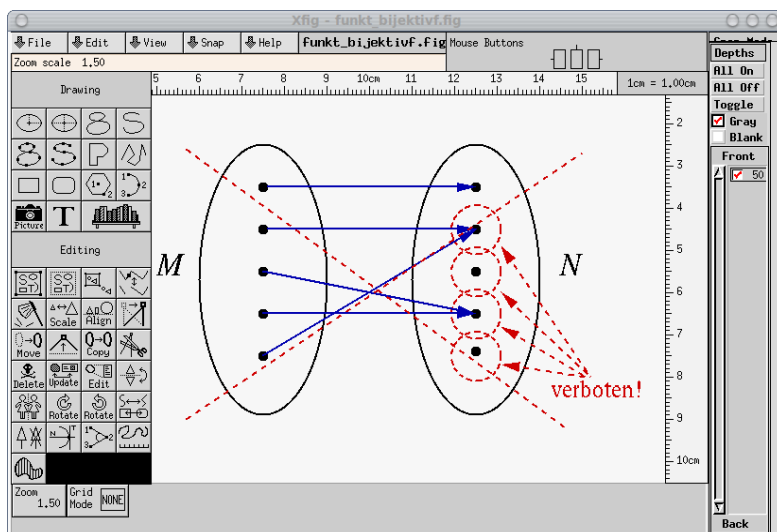


Abbildung 7.1.: Screenshot Xfig mit Beispiel aus Mathematik 1.

- <http://www.xfig.org/>, bei ubuntu dabei
- mittlerweile in die Jahre gekommen, aber sehr vielseitig
- kann *.eps, *.pdf, *.tex und Kombinationen davon exportieren

Ipe

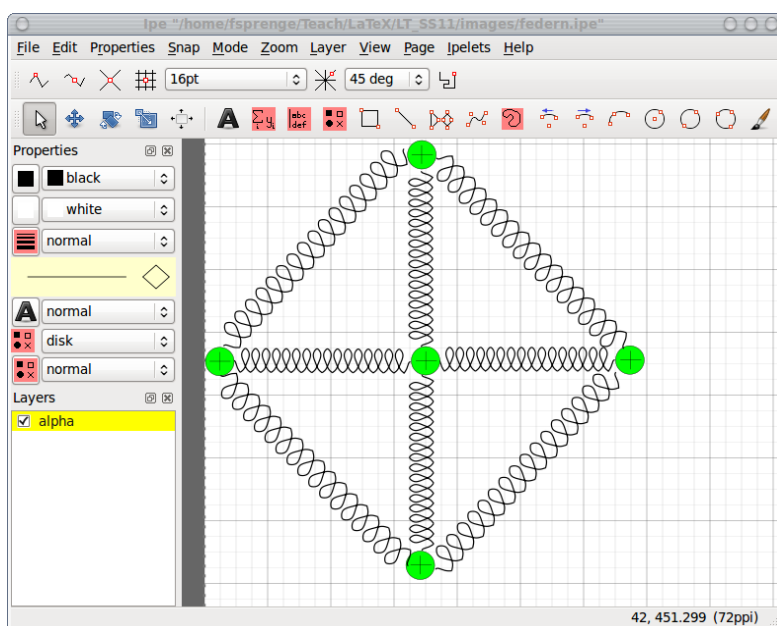


Abbildung 7.2.: Screenshot Ipe mit Beispiel zum Graphen zeichnen (federbasierte Methoden).

- ipe7.sourceforge.net, bei ubuntu dabei
- XML-basiert, kann *pdf erzeugen, selbst erzeugte *pdf auch wieder lesen und bearbeiten
- kann \LaTeX -Formeln darstellen, braucht \LaTeX -Installation

Jpgfdraw

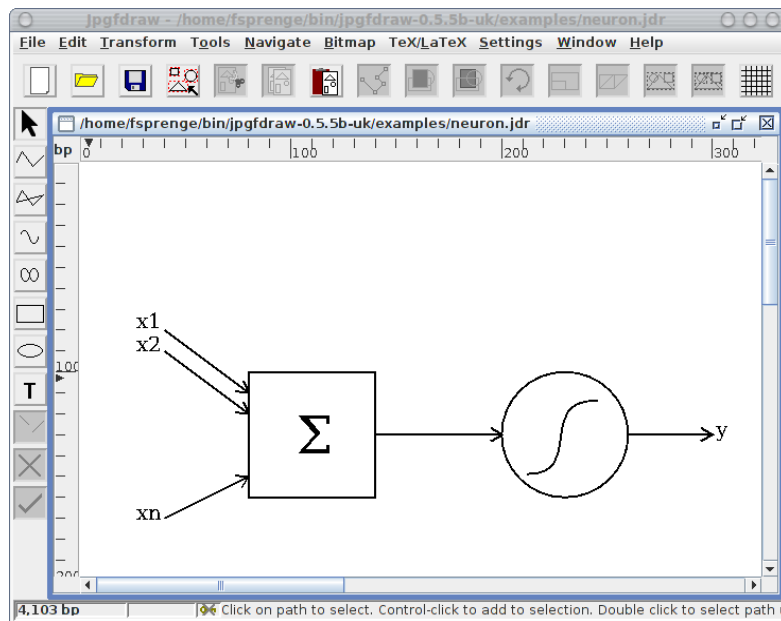


Abbildung 7.3.: Screenshot Jpgfdraw mit Beispiel zu neuronalen Netzen (mitgeliefert).

- <http://theoval.cmp.uea.ac.uk/~nlct/jpgfdraw/>
- Java-basiert
- liefert \LaTeX -Code (für pgfplots), den man in seine Dateien einbinden kann

Dia

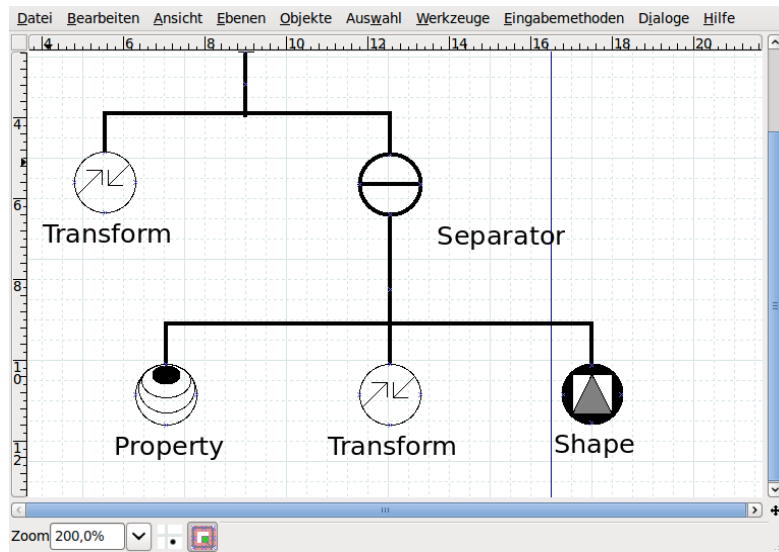


Abbildung 7.4.: Screenshot Dia mit Beispiel zu einem Szenegraphen.

- dia-installer.de
- bei ubuntu dabei
- gut für UML
- liefert \LaTeX -Code (für pgfplots oder PStricks), den man in seine Dateien einbinden kann
- kann auch SVG, EPS, PDF exportieren

7.6.2. Programme für die PostScript und PDF-Bearbeitung

PostScript

Für die PostScript-Bearbeitung haben sich bewährt:

- pstops (insbesondere Druckaufbereitung, 4 auf 1, Buchdruck usw.)
- alles was im ubuntu-Paket pstools noch dabei ist

Für die Umwandlung in PDF

- pstopdf
- epstopdf

Pdftk

Pdftk ist ein Alleskönner.

- <http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/>, bei ubuntu dabei
- von der Seite:

```
pdftk <input PDF files | - | PROMPT>
[input_pw < input PDF owner passwords | PROMPT>]
[< operation > < operation arguments >]
[output < output filename | - | PROMPT >]
[encrypt_40bit | encrypt_128bit]
[allow < permissions >]
[owner_pw < owner password | PROMPT >]
[user_pw < user password | PROMPT >]
[flatten] [compress | uncompress]
[keep_first_id | keep_final_id]
[drop_xfa] [verbose] [dont_ask | do_ask]
```

Where: <operation> can be empty, or:

```
[ cat | shuffle | burst |
generate_fdf | fill_form |
background | multibackground |
stamp | multistamp |
dump_data | dump_data_utf8 |
dump_data_fields | dump_data_fields_utf8 |
update_info | update_info_utf8 |
attach_files | unpack_files ]
```

- gern benutzt:

```
pdftk inputfile cat anf-endseite output outputfile
```

Pdftcrop

- <http://www.ctan.org/tex-archive/support/pdftcrop>, meist bei der L^AT_EX-Installation dabei (Heiko Oberdiek)
- schneidet weiße Ränder von PDFs ab
- Perl-Script, braucht ghostscript und pdftex
- wird bei Verwendung von pstricks mit auto-pst-pdf automatisch verwendet
- **Achtung:** Bei *sourceforge* gibt es ein anderes Perl-Script gleichen Namens mit ähnlicher Funktion

Syntax: `pdftcrop [options] <input[.pdf]> [output file]`

Über die Options ist es z. B. möglich mit `--margins 20` einen Rand von 20 Punkt (Didot) hinzuzufügen.

7.6.3. Weitere Helferlein

- impressive (Vorführung von beamer-pdfs)
- rtf2beamer (Umwandlung von als rtf exportierten Powerpoints nach latex-beamer)
- pdfsizeopt (Größen-Optimierung von PDFs)
- pdfbook (Umwandlung von PDF-A4 nach 2x A5, so dass es als Buch gebunden werden kann)

7.7. Zusammenfassung

- große Dokumente sollte man auf mehrere Dateien aufteilen
- eigene Kommandos und Umgebungen können das Leben erleichtern – immer in der Präambel definieren
- Zähler
- eigene Pakete und Klassen
- externe Zeichenprogramme: Xfig, Ipe, Jpgfdraw
- externe Programme zur Pdf-Bearbeitung
- pdflatex auf der Komandozeile

A. Übungen

A.1. Wissenschaftliches Arbeiten

A.1.1. Übungsaufgaben

In dieser Übung sollen Sie sich mit dem Thema “*wissenschaftlich arbeiten*” auseinander setzen und die Erkenntnisse in den späteren Übungen benutzen. Insbesondere wenn wir demnächst Dokumentenverarbeitung machen, werden Sie Dokumente erstellen müssen. Als Inhalte sollen Sie dann das benutzen, was Sie sich in dieser Übung erarbeitet haben.

Aufgabe A.1.1 Recherche zum Thema “wissenschaftlich arbeiten”

- a). Es gibt viele Bücher zu diesem Thema. Finden Sie mindestens ein Buch hierzu in unserer Bibliothek und leihen es aus.
- b). Suchen Sie gute Internet-Quellen, die sich dem Thema widmen und speichern Sie die URLs.

Aufgabe A.1.2 Dokument erstellen

- a). Erstellen Sie eine Gliederung für eine schriftliche Ausarbeitung zum Thema “wissenschaftlich Arbeiten” (Zielgruppe: Studierende, Umfang der gesamten Ausarbeitung: ca. 5-10 DIN A4 Seiten).
- b). Schreiben Sie das Dokument in einem einfachen Editor wie gedit als einfachen Nurtext (utf8) ohne spezielle Formatierungen. Eine A4-Seite reiner Text umfasst ungefähr 2 kB.

Hinweis: Sie sollen nicht das komplette Thema “wissenschaftlich Arbeiten” in Ihrem Dokument abhandeln, sondern Sie dürfen einen beliebigen Ausschnitt/Teilaspekt wählen und diese Inhalte dann auf 5-10 Seiten eigenem Text zusammenschreiben. Den Text sollen Sie bei den L^AT_EX-Übungen später als Beispiel-Dokument benutzen.

Aufgabe A.1.3 Dokument abgeben

Zippen Sie die .txt-Datei Ihres Artikels und laden Sie das zip-File in Lon-Capa hoch (Termin siehe dort).

Aufgabe A.1.4 Zusammenfassung erstellen

Erstellen Sie eine kurze schriftliche Zusammenfassung der wichtigsten Punkte aus der ersten Vorlesung.

Aufgabe A.1.5 Recherche zum Thema “Textverarbeitung”

Lesen Sie das Dokument *Textverarbeitungen: Dumm und Ineffizient* von Allen Cottrell. Sie finden es im Unterverzeichnis **Materialien** im Lon-Capa-Kurs.

Aufgabe A.1.6 Recherche zum Thema Plagiat

- a). Finden Sie Dokumente, in denen der Begriff *Plagiat* erklärt wird.
- b). Welche rechtlichen Konsequenzen können Plagiate in Deutschland nach sich ziehen?

A.2. L^AT_EX – erste Schritte

A.2.1. Übungsaufgaben

In dieser Übung sollen die ersten Grundlagen des Satzsystems L^AT_EX eingeübt und gleichzeitig selbständiges Recherchieren geübt werden. Dabei fallen die folgenden Teilaufgaben an:

Aufgabe A.2.1 Dokumentation recherchieren

Zu L^AT_EX gibt es eine ganze Menge an nützlichen Informationen im Internet. Suchen sie die folgenden Dokumente und arbeiten das erste anschließend durch:

- Die L^AT_EX2e Kurzbeschreibung (**l2kurz.pdf**, bzw. **l2short.pdf**)
- Die *frequently asked questions* (FAQ) der Newgruppe de.comp.text.tex
- Den *Scrguide* zum koma Paket.
- Das Dokument *l2tabu*. Darin steht welche Kommandos veraltet sind und nicht mehr benutzt werden sollten.

Hinweis: Benutzen Sie die Dokumente b) bis d) als Nachschlagewerk und schauen dort erst bei Bedarf hinein. Drucken Sie die Dokumente aus, wenn Sie auf Papier besser als vom Bildschirm lesen können.

Aufgabe A.2.2 Daueraufgabe: Typische Fehler sammeln

Wenn Sie in den nächsten Übungen Fehler machen (die L^AT_EX Ihnen meldet), dann dokumentieren Sie diese mit einem Screenshot und schicken Sie sie uns per E-Mail (jeden Fehler bitte nur einmal). Am Ende der Vorlesung machen wir ein Quiz darüber.

Aufgabe A.2.3 Ein erstes Dokument erstellen

Benutzen Sie einen einfachen Editor Ihrer Wahl (gedit, emacs, vim, ...).

Nehmen sie ihr Dokument vom ersten Übungsblatt (ein kurzes (ca. 5–10 Seiten) Dokument über das Thema “Wissenschaftliches Arbeiten”) und fügen die L^AT_EX Kommandos der Artikel-Klasse von KOMA-Skript ein. In diesem Dokument sollen alle typischen Elemente eines Artikels vorkommen. (Titel, Autoren, Zusammenfassung, Abschnitte und Unterabschnitte)

Aufgabe A.2.4 Erstellen einer PDF-Version des Dokuments

Erstellen Sie mit pdf_latex eine .pdf-Datei des Dokumentes (auf der Konsole)

Aufgabe A.2.5 Änderungen am Dokument vornehmen

- a). Erkundigen Sie sich über *Ligaturen*¹. Lesen Sie sich nun den Abschnitt über Ligaturen in `12kurz.pdf` durch. Müssen Sie Änderungen an Ihrem Dokument vornehmen? Finden Sie auf CTAN ein Programm, das Ihnen die Arbeit erleichtert?
- b). Ändern Sie die Größe des Zeichensatz auf 12 pt und formatieren das Dokument neu.
- c). Ändern Sie die Papiergröße von DIN A4 auf DIN A5 und formatieren das Dokument neu.
- d). Setzen Sie den Artikel zweispaltig auf DIN A4 Papier.

Falls Fragen bei der Lösung der Aufgabe auftauchen, versuchen Sie zuerst die Fragen anhand der Dokumentation aus Aufgabe 1 selbst zu beantworten. Falls das nicht klappt, fragen Sie die Betreuenden.

Aufgabe A.2.6 Dokument abgeben

Zippen Sie die `.tex`- und die `.pdf`-Datei Ihres Artikels (einspaltig, A4, 11pt) und laden Sie das zip-File in Lon-Capa hoch (Termin siehe dort).

¹Zum Beispiel auf <http://www.typolution.de/ligaturen>.

A.3. L^AT_EX – Erstes Dokument

A.3.1. Übungsaufgaben

Arbeiten Sie zunächst weiter mit Ihrem einfachen Editor plus pdflatex auf der Konsole.

Aufgabe A.3.1 Quelltext lesbar gestalten

Nehmen Sie ihren Artikel aus der letzten Übung. Formatieren Sie die Eingabedatei (also die .tex-Datei) manuell so, dass nach jedem Satz im Editor eine neue Zeile beginnt und dass keine Zeile mehr als ca. 70 Zeichen enthält. Wenn es die Übersichtlichkeit verbessert, dann fügen Sie zusätzliche Leerzeilen z. B. vor neuen Absätzen ein. Außerdem können Sie Texte auch einrücken, so wie Sie es aus dem Programmiersprachen kennen.

Aufgabe A.3.2 Report

Falls Sie bisher mit der Dokumentenklasse article gearbeitet haben, ändern Sie dies auf scrartcl und vergleichen Sie die Ergebnisse!

Benutzen Sie jetzt entweder die Koma-Klassen oder die klassischen Klassen (je nachdem, was Ihnen besser gefällt).

Ändern Sie die Klasse auf report/scrreprt und vergleichen Sie das Ergebnis mit dem vorherigen Artikel. Passen Sie die Gliederungsebenen an!

Aufgabe A.3.3 Verzeichnisse und Referenzen

Erweitern Sie nun den Report wie folgt:

- a). Fügen Sie hinter der Zusammenfassung (engl. abstract) ein Inhaltsverzeichnis ein. Sehen Sie sich das Ergebnis an.
- b). Fügen Sie nun einen neuen Abschnitt an das Ende des Artikels. Lassen Sie latex **ein** Mal laufen und sehen sich das Ergebnis an.
 - i) Was steht nach diesem Lauf in den Hilfsdateien?
 - ii) Wie oft muss latex (oder pdflatex) laufen, damit das Inhaltsverzeichnis wieder stimmt?
- c). Fügen Sie im dritten Abschnitt einen Verweis auf den ersten Unterabschnitt im zweiten Abschnitt ein. Kontrollieren Sie, ob die Abschnittsnummer im Verweis korrekt ist. Fügen Sie dann zwischen den ersten und zweiten Abschnitt einen weiteren Abschnitt ein. Prüfen Sie wieder, ob der Verweis immer noch die richtige Abschnittsnummer enthält.

Aufgabe A.3.4 Listen

In Informatik-Texten wird gern aufgezählt und nummeriert. Recherchieren Sie, wie solche Listen formatiert werden.

Fügen Sie im letzten Abschnitt je eine nummerierte und eine unnummerierte Aufzählung mit einer Schachtelungstiefe von mindestens zwei Ebenen ein.

Aufgabe A.3.5 Dokument abgeben

Zippen Sie die .tex- und die .pdf-Datei Ihres Reports und laden Sie das zip-File in Lon-Capa hoch (Termin siehe dort).

Aufgabe A.3.6 Editoren ausprobieren

In den Pools sind auch weitere Hilfsmittel (Editoren, IDEs) installiert bzw. installierbar:

- a). kile
- b). texworks (ist bei texlive mit dabei)
- c). eclipse mit texlipse plugin und pdf4Eclipse plugin
- d). um das texlipse Plugin für sich zu Hause zu installieren, gehen Sie bitte wie folgt vor:
 - i) Eclipse starten
(Es gibt bekannte Probleme bei Eclipse 4 (also ab Juno), verwenden Sie am besten etwas Bewährtes wie Indigo, für Juno-Verwender gibt es ein Workaround mit etwas Basteln.)
 - ii) Neue Software installieren
In der Eclipse Umgebung: Help → Install new Software
 - iii) Quelle angeben
Work with: <http://texlipse.sourceforge.net>
 - iv) TeXlipse und Pdf4Eclipse auswählen
Haken bei TeXlipse und Pdf4Eclipse setzen
 - v) Software installieren
Einfach auf “Next” klicken, Lizenz akzeptieren und anschließend auf “Finish” klicken
 - vi) Pdf4Eclipse kann man alternativ auch direkt finden unter <http://borisvl.github.com/Pdf4Eclipse>
 - vii) Ein paar Settings setzen, das Wichtigste:
Window → Preferences → TeXlipse → Builder Settings, Bin directory of TeX Distribution setzen

- viii) Für weitere Einstellungen: siehe <http://texlipse.sourceforge.net/manual/installation.html>
- ix) Latex-Perspektive wählen, in den Projekt-Properties als output-Format .pdf wählen, als Sprache de (oder en) wählen
- x) Spellchecking: falls die Dictionaries nicht schon dabei sind, kann man sie hier: <http://texlipse.sourceforge.net/manual/installation.html> herunterladen und in das eclipse-Verzeichnis legen und in den Settings einstellen

Probieren Sie die Editoren aus, entscheiden Sie sich für einen und arbeiten Sie künftig damit.

A.4. Mathematik, Graphik und Tabellen

A.4.1. Übungsaufgaben

Aufgabe A.4.1 Mathematik

Erarbeiten Sie sich selbstständig weitere $\text{T}_{\text{E}}\text{X}$ und $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ Kommandos, mit denen mathematische Formeln erstens im fließenden Text und zweitens in abgesetzten Zeilen gesetzt werden.

Stellen Sie sicher, dass Sie keine veralteten Kommandos benutzen. Probieren Sie z.B. `texdoc mathmode`, `texdoc amsmath` und `texdoc l2tabu`.

Setzen Sie dann diesen Text in genau dieser Form:

Die Fakultät einer Zahl n ist definiert als $n! = 1 \times 2 \times \cdots \times n$. Mathematiker definieren das präziser als

$$n! = \prod_{i=1}^n i$$

Aber das ist sowieso unvollständig, da ja auch $n = 0$ gelten kann. Richtig vollständig muss man also eine Fallunterscheidung treffen und kann bei der Gelegenheit das Ganze auch rekursiv definieren als

$$\text{Fak}(n) = \begin{cases} 1 & \text{falls } n = 0, \\ n \cdot \text{Fak}(n - 1) & \text{sonst.} \end{cases}$$

Man beachte das in dieser Formel veränderte Multiplikationssymbol. Außerdem benutzen Mathematiker auch gerne griechische Buchstaben, wie α , β , γ . Wenn dann die Buchstaben knapp werden, dann wird hoch- oder tiefgestellt, so wie in x_1, x_2, \dots, x_n .

Aufgabe A.4.2 Graphik und Abbildungen

Erstellen Sie mit einem beliebigen Grafikprogramm (z.B. Dia, Xfig, Ipe7, ...) eine **Vektorgraphik** (z.B. ein UML-Diagramm, eine Kurve im Koordinatensystem, ...). Bauen Sie dann die Grafik in ihren Artikel als Gleitobjekt ein. Beachten Sie, dass die Programme `latex` und `pdflatex` unterschiedliche Graphikformate einbinden können. Es gibt viele Programme, die Grafiken von einem Format automatisch in ein anderes Format umwandeln können.

- Skalieren Sie die Grafik im Artikel so, dass sie $2/3$ der Textbreite einnimmt.
- Variieren Sie die Ort-Parameter der Gleitumgebung und sehen sich an, welche Auswirkungen das auf die Platzierung der Grafik hat.
- Was ist der Unterschied zwischen den Kommandos `\clearpage` und `\newpage` in Hinblick auf Gleitobjekte?

Aufgabe A.4.3 Tabelle

Erstellen Sie eine Tabelle mit fiktiven Klausurergebnissen. Das Ergebnis soll dieser Tabelle so ähnlich wie möglich sehen:

Nr.	Immatrikulationsnummer	Nachname	Vorname	Versuch	Note
1	1111111	Duck	Dagobert	1	1,3
2	2222222	Gans	Gustav	2	3
3	3333333	Duck	Donald	2	—
4	4444444		

Um die Zeilen farblich hervorzuheben, benutzen Sie bitte den Befehl `\rowcolors` aus dem Paket `xcolor`. Versuchen Sie, den Befehl offline zu recherchieren. Tipp: \TeX Live wird mit dem Dokumentations-Tool `texdoc` ausgeliefert.

Schaffen Sie es, dass die laufende Nummer automatisch generiert wird?

Aufgabe A.4.4 Referenzieren

Fügen Sie `\label` an verschiedenen Stellen Ihres Artikels ein und greifen Sie mit `\ref`, `\vref`, `\pageref` und `\vpageref` darauf zu. Wie unterscheiden sich die Ausgaben?

Schlagen Sie in der `varioref`-Dokumentation nach, welche besondere Gefahr besteht an Seitengrenzen? Können Sie den Fehler in Ihrem Dokument nachvollziehen? Wie lässt sich das Problem umgehen?

Aufgabe A.4.5 URLs einfügen

Sorgen Sie dafür, dass man in der PDF-Datei ihres Artikels den Verweisen (z.B. in andere Abschnitte oder auf Abbildungen) durch einen Klick folgen kann. Bauen Sie außerdem einen Verweis auf die FAQ des DANTE e.V. zu \LaTeX in ihren Artikel ein. Wenn man auf diesen Verweis klickt, dann soll die FAQ vom Webserver des DANTE e.V. geladen und angezeigt werden.

Aufgabe A.4.6 Dokument abgeben

Zippen Sie die `.tex`-, die Grafik- und die `.pdf`-Datei Ihres Artikels und laden Sie das zip-File in Lon-Capa hoch (Termin siehe dort).

A.5. Literatur, Quelltexte und Präsentationen

A.5.1. Übungsaufgaben

Aufgabe A.5.1 Literaturdatenbank erstellen

Überlegen Sie sich, wie sie persönlich am liebsten eine BibTeX -Datenbank bearbeiten wollen. Im Prinzip ist jeder Editor geeignet. Wenn Sie graphische Benutzeroberflächen vorziehen, installieren Sie sich JabRef oder das Mendeley-Desktop (beides in den Pools vorhanden).

- a). Suchen Sie sich jeweils zwei Bücher, zwei Zeitschriftenartikel, und zwei Internet-Quellen heraus und erstellen eine möglichst vollständige BibTeX-Datenbank für diese Quellen.
- b). In der Informatik werden öfter auch Konferenzbeiträge und Buchbeiträge zitiert. Suchen Sie in der Dokumentation von BibTeX, wie man diese Literaturtypen in eine BibTeX-Datenbank eingeben kann.
- c). Suchen Sie sich mindestens zwei Konferenzbeiträge und zwei Buchbeiträge heraus und tragen Sie sie in ihre BibTeX-Datenbank ein.

Aufgabe A.5.2 Literaturverzeichnis erstellen

- a). Nehmen Sie den Artikel aus der letzten Übung und bauen einige Zitate aus den Quellen (aus Aufgabe 2) in den Artikel ein.
- b). Erstellen Sie mit bibtex ein Literaturverzeichnis zu ihrem Artikel. Die Verweise sollen in der Form [KNU65] erscheinen (also Autornamen und Jahreszahl abgekürzt).
- c). Was muss man ändern, wenn die Verweise durchlaufend nummeriert sein sollen, beispielsweise wie in [2].
- d). Erstellen Sie eine PDF-Datei ihres Artikels mit Inhaltsverzeichnis, Abbildungen, auf die im Text natürlich verwiesen wird, und dem Literaturverzeichnis.

Aufgabe A.5.3 Quelltexte einfügen

Nehmen Sie die Quelltexte des Java- oder C-Programms, das sie zuletzt geschrieben haben. Fügen Sie die zweite Methode der ersten Klasse dieses Programms in ihren Artikel ein. Für die weiteren Aufgaben werden Sie wohl in die Dokumentation sehen müssen.

- a). Schalten Sie das passende Syntax-Highlighting an!
- b). Verpassen Sie dem Listing eine Box mit Hintergrundfarbe und abgerundeten Ecken!
- c). Geben Sie ihm eine Über-/Unterschrift!
- d). Lassen Sie automatisch ein Verzeichnis aller Listings erstellen!

Aufgabe A.5.4 Beamer-Präsentation

Erstellen Sie eine Beamer-Präsentation mit

- a). einem beamer-theme Ihrer Wahl (vorgestellt im beamerusguide)
- b). Titelfolie
- c). Inhaltsverzeichnis vor jeder section
- d). einigen Folien mit Titeln und manchmal Untertiteln
- e). einem Button auf Folie 3, der dafür sorgt, dass zu Folie 7 gesprungen wird
- f). eine Folie, deren Inhalt “schrittweise aufgedeckt” wird (z.B. eine Aufzählung mit drei Punkten)

Aufgabe A.5.5 Bézierkurve zeichnen

Nehmen Sie ein Zeichenprogramm Ihrer Wahl aus der Menge { Ipe, Xfig, Jpgfdraw }, die ersteren sind in den Pools installiert, das letzter können Sie leicht selbst installieren, und realisieren Sie die folgende Aufgabe:

- a). Zeichnen Sie eine (kubische) Bézierkurve und ein Achsenkreuz!
- b). Beschriften Sie die Kurve und die Achsen!
- c). Exportieren Sie Ihre Zeichnung!
- d). Binden Sie Ihre Grafik in Ihre Beamerpräsentation ein!

Aufgabe A.5.6 Dokumente abgeben

Zippen Sie alle Dateien Ihres Artikels (.bib,.tex,.pdf,...) und Ihrer Präsentation (.tex,.pdf,Grafik,...) und laden Sie das zip-File in Lon-Capa hoch (Termin siehe dort).

A.6. Dokumentenstruktur und eigene Befehle

A.6.1. Übungsaufgaben

Aufgabe A.6.1 Dokumentenstruktur und eigene Befehle

- a). Teilen Sie Ihren Artikel auf mindestens drei Unterdokumente auf, die Sie in das Hauptdokument einbinden!
- b). Definieren Sie sich einen neuen Befehl `markiere{Inhalt}` der dafür sorgt, dass der `Inhalt` im Text hervorgehoben wird durch Fettdruck, geneigte Schrift und rote Schriftfarbe. Verwenden Sie den Befehl im Text, um einige Worte hervorzuheben.²
- c). Definieren Sie sich eine Umgebung `imprint` und benutzen Sie sie, um ein Impressum zu Ihrem Dokument hinzuzufügen. Welcher Teil des Impressums könnte vordefiniert sein, welcher gehört in die Umgebung?
- d). Definieren Sie sich eine Umgebung `umgebung`, die im Verhalten so aussieht

Umgebung I

Eine erste Umgebung.

*

Umgebung II

Noch eine Umgebung.

*

erzeugt durch

```
\begin{umgebung}
Eine erste Umgebung.
\end{umgebung}

\begin{umgebung}
Noch eine Umgebung.
\end{umgebung}
```

Die Umgebung soll also die Überschrift, die fortlaufende (erzeugte) Nummerierung in römischen Ziffern und den Abschluss generieren.

Aufgabe A.6.2 Eigene Klasse und eigenes Paket

- a). Schreiben Sie eine eigene Klasse und ein eigenes Paket !
- b). Die Klasse soll die Klasse `scrartcl` laden und die meisten Optionen an diese Klasse durchreichen. Nur wenn die Option `ngerman` oder `english` kommt, soll diese an das Paket `babel` durchgereicht werden.
- c). Fassen Sie alle anderen zu ladenden Pakete und Ihre eigenen Befehle/Umgebungen (s.o.) in einem eigenen Paket zusammen, das in Ihrer Klasse geladen wird.

²Diese Aufgabe ist akademisch. Normalerweise sollten Sie mehrfache Hervorhebungen meiden!

- d). Die Präambel Ihres Artikels sollte am Ende nur noch aus `\documentclass` mit Optionen und Ihrer Klasse lauten.

Aufgabe A.6.3 Dokument abgeben

Zippen Sie sämtliche .cls, .sty, .tex-, .bib, Grafik- und die .pdf-Dateien Ihres Artikels und laden Sie das zip-File in Lon-Capa hoch (Termin siehe dort).

A.7. Freies Üben

A.7.1. Übungsaufgaben

Diese Übungen sollen Ihnen die Möglichkeit geben das, Gelernte zu vertiefen und das nicht Gelernte doch noch zu lernen. Diese Übungen sind als Vorschläge zu verstehen. Wenn Sie wollen, können Sie die Zeit auch nutzen, um Dilbert-Cartoons zu lesen, wir raten jedoch davon ab.³ Möglicherweise reicht die Zeit nicht für alle Aufgaben. Lösen Sie die Aufgaben, die Ihnen am interessantesten oder sinnvollsten erscheinen zuerst.

Aufgabe A.7.1 Probleme lösen

Hatten Sie bei der Erstellung Ihrer bisherigen Dokumente Probleme? Versuchen Sie sie jetzt zu lösen.

Aufgabe A.7.2 Pakete finden

Schauen Sie sich auf CTAN um. Finden Sie Pakete, die Sie für Ihre Seminar-Ausarbeitung gebrauchen können?

Fällt Ihnen ein Problem ein, für das es ein Paket geben könnte und wenn ja, finden Sie es?

Erscheinen Ihnen die folgenden Pakete nützlich?

- csquotes
- hyphsubst
- microtype

Aufgabe A.7.3 Einarbeitung in die Regeln der Typografie

Befassen Sie sich mit den Regeln der Mikrotypografie ein. Lesen Sie zum Beispiel *typokurz* im Materialien-Ordner. Wie lassen sich die Regeln in L^AT_EX umsetzen?

Finden Sie einen Weg, sog. Hurenkinder und Schusterjungen automatisch zu verhindern?

Aufgabe A.7.4 Lebenslauf

Schreiben Sie Ihren Lebenslauf. Recherchieren Sie die Gestaltungsregeln für Lebensläufe. Welche Informationen sollten enthalten sein, gefällt Ihnen eine chronologische oder achronologische Anordnung besser? (Beides ist zulässig.) Welcher Stil von moderncv gefällt Ihnen am besten? Verwenden Sie bei der Umsetzung ruhig Ihre echten Daten, dann sparen Sie am Ende Ihres Studiums ein wenig Zeit.

³Obwohl viele davon wirklich gut sind.

Liste der Übungen

Aufgabe A.1.1 <i>Recherche zum Thema “wissenschaftlich arbeiten”</i>	A-1
Aufgabe A.1.2 <i>Dokument erstellen</i>	A-1
Aufgabe A.1.3 <i>Dokument abgeben</i>	A-1
Aufgabe A.1.4 <i>Zusammenfassung erstellen</i>	A-2
Aufgabe A.1.5 <i>Recherche zum Thema “Textverarbeitung”</i>	A-2
Aufgabe A.1.6 <i>Recherche zum Thema Plagiat</i>	A-2
Aufgabe A.2.1 <i>Dokumentation recherchieren</i>	A-3
Aufgabe A.2.2 <i>Daueraufgabe: Typische Fehler sammeln</i>	A-3
Aufgabe A.2.3 <i>Ein erstes Dokument erstellen</i>	A-3
Aufgabe A.2.4 <i>Erstellen einer PDF-Version des Dokuments</i>	A-3
Aufgabe A.2.5 <i>Änderungen am Dokument vornehmen</i>	A-3
Aufgabe A.2.6 <i>Dokument abgeben</i>	A-4
Aufgabe A.3.1 <i>Quelltext lesbar gestalten</i>	A-5
Aufgabe A.3.2 <i>Report</i>	A-5
Aufgabe A.3.3 <i>Verzeichnisse und Referenzen</i>	A-5
Aufgabe A.3.4 <i>Listen</i>	A-6
Aufgabe A.3.5 <i>Dokument abgeben</i>	A-6
Aufgabe A.3.6 <i>Editoren ausprobieren</i>	A-6
Aufgabe A.4.1 <i>Mathematik</i>	A-8
Aufgabe A.4.2 <i>Graphik und Abbildungen</i>	A-8
Aufgabe A.4.3 <i>Tabelle</i>	A-9
Aufgabe A.4.4 <i>Referenzieren</i>	A-9
Aufgabe A.4.5 <i>URLs einfügen</i>	A-9
Aufgabe A.4.6 <i>Dokument abgeben</i>	A-9
Aufgabe A.5.1 <i>Literaturdatenbank erstellen</i>	A-10
Aufgabe A.5.2 <i>Literaturverzeichnis erstellen</i>	A-10
Aufgabe A.5.3 <i>Quelltexte einfügen</i>	A-10
Aufgabe A.5.4 <i>Beamer-Präsentation</i>	A-11
Aufgabe A.5.5 <i>Bézierkurve zeichnen</i>	A-11
Aufgabe A.5.6 <i>Dokumente abgeben</i>	A-11
Aufgabe A.6.1 <i>Dokumentenstruktur und eigene Befehle</i>	A-12

Aufgabe A.6.2 <i>Eigene Klasse und eigenes Paket</i>	A-12
Aufgabe A.6.3 <i>Dokument abgeben</i>	A-13
Aufgabe A.7.1 <i>Probleme lösen</i>	A-14
Aufgabe A.7.2 <i>Pakete finden</i>	A-14
Aufgabe A.7.3 <i>Einarbeitung in die Regeln der Typografie</i>	A-14
Aufgabe A.7.4 <i>Lebenslauf</i>	A-14