LaTeX für Naturwissenschaftler

Ansprechender Text- und Formelsatz von Abschlussarbeiten

7. Weitere hilfreiche Pakete

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Dr. Till Biskup

Institut für Physikalische Chemie
Albert-Ludwigs-Universität Freiburg
Sommersemester 2018

Zentrale Aspekte

- Das Rad neu zu erfinden ist selten eine gute Idee.
  Meist gibt es Pakete, die besser sind als eigene Lösungen.
- Viele Pakete dienen der logischen Textauszeichnung.
  Die Idee hinter LaTeX: Fokus auf Inhalt statt Formatierung.
- Wissenschaftliche Abschlussarbeiten stellen eine Reihe immer wiederkehrender Anforderungen.
- Die Liste erwähnter Pakete ist hochgradig subjektiv und der Versuch, häufige Anwendungsfälle zu berücksichtigen.
- Grundsätzlich sollte vor der Verwendung eines Pakets immer erst dessen Dokumentation konsultiert werden. ☺

Übersicht

UNI FREIBURG

Quellcodebeispiele

Abkürzungen und Abkürzungsverzeichnis

Größen und Einheiten

Typografie und Textauszeichnung

UNI FREIBURG

These

Dokumentation von Quellcode durch Aufnahme in Abschlussarbeiten wird viel zu gering geschätzt.

- Programmierung ist oft wesentlicher Teil der Forschung.
  - gilt insbesondere für die physikalische Chemie
- Nachvollziehbarkeit der Datenverarbeitung
  - nur bei Archivierung und Zugänglichkeit des Quellcodes der Auswertungsroutinen gewährleistet

☛ Details bei Interesse in der Vorlesung „Programmierkonzepte in der Physikalischen Chemie“

Experimentelle Arbeiten

- Eigene Auswertungsroutinen sollten dokumentiert werden.
- Ausführliche Quellcode-Auflistungen in den Anhang
- mögliche Alternative: Versionierung und Archivierung

Theoriearbeiten mit Schwerpunkt Implementierung

- im Haupttext
  - kurze Quellcode-Abschnitte
  - Verdeutlichung von Implementationen
  - Pseudocode ist oft gut geeignet.
- im Anhang
  - eigentliches Programm

Quellcodebeispiele

Einbinden des Pakets `listings`

UNI FREIBURG

Listing 1: Einbinden des Pakets `listings`

```
\usepackage{listings}
```

- Gründe für die Verwendung des Pakets
  - wenn Quellcode in der Abschlussarbeit eine Rolle spielt
  - wenn Auswertungsprogramme eingebunden werden sollen, i.d.R. im Anhang einer Arbeit
- Gründe für die spezielle Behandlung von Quellcode
  - Quellcode enthält oft Sonderzeichen, die in LaTeX nicht ohne Weiteres dargestellt werden können.
  - Quellcode sollte nicht extra formatiert werden müssen.
  - Direktes Kopieren (oder Einlesen) sorgt für Konsistenz.

Quellcodebeispiele

Drei Formen von Quellcode im Text

UNI FREIBURG

- einzelne Begriffe oder kurze Teile
  - ähnlich wie mathematische Formeln im fließenden Text
- Blöcke mit Quellcodebeispielen
  - als Gleitumgebung oder statisch
  - ggf. mit Seitenumbruch
  - Gleitumgebungen mit Überschrift und Nummerierung
- ganze Quellcodedateien
  - werden sinnvoller Weise direkt eingelesen

☛ Die LaTeX-Umgebung `verbatim` unterstützt nur die ersten beiden Fälle (und nur in Teilen).

☛ Das Paket `lstlistings` bietet alle Möglichkeiten und darüber hinaus komfortable Formatierungsoptionen.

Listing 2: Beispiel für ein Quellcodefragment im Fließtext

```
\lstinline!\LaTeX{}!
\lstinline[<Optionen>]!\LaTeX{}!
```

- Einsatzgebiete
  - einzelne Befehle im Text
  - Konsistenz bzgl. der restlichen Formatierung von Quellcode
  - umgeht die Probleme mit Sonderzeichen in LaTeX
- Unterschiede zum LaTeX-Befehl `\verb`
  - Stil entspricht den Einstellungen des `listings`-Pakets
  - Stile lassen sich über optionales Argument einstellen
  - sprachabhängige Syntaxhervorhebung funktioniert

Quellcodebeispiele

Beispiel für einen Quellcodeblock

UNI FREIBURG

Listing 3: Beispiel für einen Quellcodeblock

```
\begin{lstlisting}[<Optionen>]
...
\end{lstlisting}
```

- Einsatzgebiete
  - Quellcode-Blöcke im Dokument
  - erlaubt zusätzliche Formatierungen und Anmerkungen (Für Details vgl. die Dokumentation.)
- Unterschiede zur LaTeX-Umgebung `verbatim`
  - Syntaxhervorhebung je nach Programmiersprache
  - Standardschriftart anders als bei `verbatim`
  - weitaus flexibler hinsichtlich der Formatierung

Quellcodebeispiele

Einbinden einer Datei als Quellcodeblock

UNI FREIBURG

Listing 4: Einbinden einer Datei als Quellcodeblock

```
\lstinputlisting[<Optionen>]{<Dateiname>}
```

- Einsatzgebiete
  - Einlesen kompletter Dateien
  - flexibel auch Teile von Dateien einlesbar
  - gut geeignet für den Anhang einer Arbeit
  - Lauffähiger Quellcode kann direkt eingebunden werden.
- kein Pendant im Standardumfang von LaTeX
  - `\input` funktioniert nur mit LaTeX-Dateien
  - `\input` innerhalb von `verbatim` ohne Wirkung

Quellcodebeispiele

Überschriften und Verweise

UNI FREIBURG

Listing 5: Beispiel für Überschriften und Verweise

```
\begin{lstlisting}[caption={[...]...},label={lst:bsp}]
...
\end{lstlisting}
```

- optionale Parameter in der `lstlisting`-Umgebung
  - `caption`
  - `label`
- `caption`
  - in geschweiften Klammern
  - intern zusätzliche eckige Klammern für Verzeichnis möglich
- Verweis in LaTeX wie üblich über `\ref`

Quellcodebeispiele

Globale Anpassung der Darstellung

UNI FREIBURG

- zwei Befehle
  - `\lstset`
  - `\lstdefinestyle`
- `\lstset`
  - globale Einstellungen für das Aussehen
  - beliebig viele Befehle nacheinander angebbar
  - Gruppierung möglich und sinnvoll
- `\lstdefinestyle`
  - definiert Stile
  - können über `style=<Stilname>` aufgerufen werden
  - praktisch für unterschiedliche Arten von Quellcode in einem Dokument (z.B. mehrere Programmiersprachen)

☛ alle Einstellungen auch für einzelne Listings möglich

Listing 6: Beispiel für die Anpassung der Darstellung von Listings

```
\lstset{
    basicstyle=\footnotesize\ttfamily, % Standardschrift
    showstringspaces=false, % Leerzeichen in Strings zeigen?
    tabsize=2,                  % Groesse von Tabs
    breaklines=true,            % Zeilen werden umgebrochen
    prebreak=\dots,             % Zeichen vor dem Umbruch
    keywordstyle=\color{blue}, % Schluesselwoerter
}
```

- einfaches Beispiel
  - Es gibt noch viel mehr Optionen (vgl. die Dokumentation).
- Schlüssel-Wert-Paare
  - Werte ggf. in geschweiften Klammern
  - Trennung mehrerer Paare durch Kommata

Quellcodebeispiele

Tricks zum Umgang mit Umlauten/Sonderzeichen im Quellcode

UNI FREIBURG

Listing 7: Direkte Nutzung von Umlauten in Quellcode

```
\lstset{
        literate=%
        {Ö}{{\"O}}1
        {Ä}{{\"A}}1
        {Ü}{{\"U}}1
        {ß}{{\ss}}1
        {ü}{{\"u}}1
        {ä}{{\"a}}1
        {ö}{{\"o}}1
        {~}{{\textasciitilde}}1
}
```

- erfordert Pakete wie `inputenc`
- kann als eigener `\lstset`-Block geschrieben werden

Quellcodebeispiele

Verzeichnis von (nummerierten) Quellcodebeispielen

UNI FREIBURG

Listing 8: Verzeichnis von (nummerierten) Quellcodebeispielen

```
\lstlistoflistings
```

- funktioniert wie alle anderen Verzeichnisse
  - Überschrift lässt sich anpassen
- schreibt Informationen in eine externe Datei
  - Grundname identisch mit LaTeX-Dokument
  - Endung: `.lol`
- braucht zwei LaTeX-Durchläufe zur Aktualisierung
  - wie alle anderen Verzeichnisse

☛ nur sinnvoll bei entsprechend vielen Quellcodebeispielen

Abkürzungen und Abkürzungsverzeichnis

Zum Umgang mit Abkürzungen in wissenschaftlichen Texten

UNI FREIBURG

- Umgang mit Abkürzungen in wissenschaftlichen Texten
  - alle Abkürzungen definieren
  - bei der ersten Erwähnung ausschreiben
  - fortan immer die Abkürzung verwenden
- Welche Abkürzungen sollen definiert werden?
  - einfache Antwort: alle (ja, auch EPR oder NMR)
  - Denkt an fachfremde Leser …
- Was man sich wünschen würde …
  - Liste mit Definition von Abkürzungen
  - automatische Handhabung von Langform und Abkürzung
  - (automatisch erstelltes) Abkürzungsverzeichnis

☛ Das Paket `acronym` nimmt viel Arbeit ab.

Abkürzungen und Abkürzungsverzeichnis

Das Paket `acronym`

UNI FREIBURG

Listing 9: Einbinden des Pakets `acronym`

```
\usepackage{acronym}
\usepackage[printonlyused]{acronym}
```

- Gründe für die Verwendung des Pakets
  - konsistente Handhabung von Abkürzungen
  - Liste von (im Text verwendeten) Abkürzungen
- bereitgestellte Funktionalität
  - Liste der Abkürzungen
  - automatisch Handhabung von Erst- und Folgeverwendung
  - Pluralformen etc.

☛ Hier werden nur grundlegende Aspekte vorgestellt.

Abkürzungen und Abkürzungsverzeichnis

Umgebung für Abkürzungen

UNI FREIBURG

Listing 10: Umgebung für Abkürzungen

```
\begin{acronym}[<breiteste Abkürzung>]
\acro{DFT}{Dichtefunktionaltheorie}
\acro{ESR}{Elektronenspinresonanz}
\end{acronym}
```

- Liste der Abkürzungen
  - Je nach Paketoption werden nur Verwendete ausgegeben.
  - erstes Argument des Befehls `\acro` dient dem Zugriff auf die Abkürzung im Text (z.B. mit `\ac`)
- Hinweise zur Verwendung
  - keine automatische alphabetische Sortierung der Einträge
  - Umgebung kann in eigene Datei ausgelagert werden
  - optionaler Parameter sorgt für passende Einrückung

Abkürzungen und Abkürzungsverzeichnis

Abkürzungen im Text verwenden

UNI FREIBURG

Listing 11: Abkürzungen im Text verwenden

```
Die \ac{DFT} ... Mit der \ac{DFT}...
```

- Unterschiede zwischen Erst- und Folgeverwendung
  - beim ersten Mal Langtext und Abkürzung in Klammern
  - bei nachfolgender Verwendung nur Abkürzung
- weitere Möglichkeiten
  - explizite Ausgabe von Kurz- oder Langform
  - Pluralformen (explizit definierbar)
  - bestimmte und unbestimmte Artikel

☛ Für Details vgl. die Paketdokumentation.

⚠ Grundregel

Physikalische Größen bestehen aus Zahl und Einheit.

- Umgang mit Größen und Einheiten
  - wurde bereits ausführlicher behandelt
  - klare Regeln, u.a. von der IUPAC
  - Typografie transportiert hier Bedeutung.
- Aspekte korrekter Typografie
  - aufrechte Schrift, keine eckigen Klammern
  - Abstände zwischen Zahl und Einheit
  - korrekte Exponenten
  - konsistente Formatierung

Größen und Einheiten

Einbinden des Pakets `siunitx`

UNI FREIBURG

Listing 12: Einbinden des Pakets `siunitx`

```
\usepackage{siunitx}
```

- Gründe für die Nutzung
  - semantische Textauszeichnung
  - einfache Handhabung von Einheiten
  - korrekte Typografie (Abstände etc.)
  - einfache Anpassungen ohne Änderungen im Text möglich
- bereitgestellte Funktionalität
  - Zahlen inklusive Dezimaltrennzeichen, Exponenten
  - Gradangaben
  - Einheiten inklusive Exponenten

Listing 13: Befehle für Zahlen und Einheiten

```
\num{<Zahl>}
\ang{<Grad>}
\si{<Einheit>}
\SI{<Zahl>}{<Einheit>}
```

- Exponenten
  - für Zahlen und Einheiten einfach schreibbar
- Einheiten
  - direkt oder über Befehle semantisch schreibbar
- Dezimaltrennzeichen
  - Komma und Punkt beide möglich
  - Ausgabe unabhängig von der Eingabe

Listing 14: Definition des Dezimaltrennzeichens

```
\sisetup{output-decimal-marker={,}}
```

- Dezimaltrennzeichen international
  - Im angelsächsischen Sprachraum ist es der Punkt.
  - In der *Mehrheit* der Länder/Sprachen ist es das Komma.
- Eingabe über die Befehle des `siunitx`-Pakets
  - unterstützen Punkt und Komma
  - unabhängig von der Ausgabe
  - Ausgabe durch Konfiguration festgelegt (s.o.)

☛ LaTeX interpretiert Kommata *per se* als Satzzeichen.
Der eingefügte Leerraum wird durch `siunitx` verhindert.

Listing 15: Tabellenspalten mit Ausrichtung am Dezimaltrennzeichen

```
\begin{tabular}{S}
3.5
\\
12.847
\\
6.022e23
\\
\end{tabular}
```

- Spaltendefinition mit „`S`“
  - Ausrichtung am Dezimaltrennzeichen
  - erkennt, ob Text oder Zahlen in einer Zelle stehen
  - gleiche Formatierung wie in `\num` erlaubt

Listing 16: Einbinden des Pakets `microtype`

```
\usepackage{microtype}
```

- Gründe für die Verwendung des Pakets
  - sorgt für verbesserte Mikrotypografie
  - sorgt insgesamt für ein besseres Schriftbild
- Nutzungsszenarien
  - kann gefahrlos immer geladen werden
  - Änderungen nur für Profis direkt sichtbar

☛ Details zum genauen Vorgehen in der Dokumentation

Listing 17: Einbinden des Pakets `csquotes`

```
\usepackage{csquotes}
```

- Gründe für die Verwendung des Pakets
  - unterstützt sprachabhängig korrekte Anführungszeichen
  - geschachtelte Anführungszeichen automatisch korrekt
  - logische Textauszeichnung
- Hinweise
  - sollte vor `biblatex` geladen werden
  - übernimmt Sprache via `babel` oder globaler Option
  - Optionen für andere deutsche Anführungszeichen

Listing 18: Standard-Befehle des Pakets csquotes

```
\enquote{Text in Anführungszeichen}
```

- \enquote
  - Text wird von Anführungszeichen umgeben
  - innerhalb des normalen Textflusses
- weitere Befehle
  - für längere Texte, längenabhängig auch als Blockzitate
  - Umgebungen für Zitate inklusive Quellenangabe
  - fremdsprachige Zitate (mit Sprachangabe)

☛ Details finden sich in der Dokumentation.

Listing 19: Einbinden des Pakets `booktabs`

```
\usepackage{booktabs}
```

- Gründe für die Verwendung des Pakets
  - typografisch korrekte Tabellen
  - Details wurden bereits behandelt
- Hinweise
  - keine vertikalen Linien in Tabellen
  - horizontale Linien sparsam einsetzen
  - unterschiedliche Linienstärken je nach Ort in der Tabelle

☞ sollte immer geladen werden, sobald Tabellen in einem textlastigen Dokument erscheinen

UNI FREIBURG

Listing 20: Einbinden des Pakets `ragged2e`

```
\usepackage[newcommands]{ragged2e}
```

- Gründe für die Verwendung des Pakets
  - LaTeX unterstützt im normalen Flattersatz („linksbündig“) keine Silbentrennung, was zu unruhigem Textsatz führt.
- Anwendungsszenarien
  - Flattersatz insbesondere für Literaturverzeichnisse
- Hinweise
  - Paket mit der Option `newcommands` laden
  - ansonsten müssen die paketeigenen Befehle (z.B. `FlushLeft`) verwendet werden

Listing 21: Einbinden des Pakets `hyperref`

```
% Am Ende der Präambel
\usepackage{hyperref}
```

- Funktionen des `hyperref`-Pakets
  - automatische Erzeugung von Links im PDF-Dokument
  - Inhaltsverzeichnis, Verweise etc. werden verlinkt
  - Unterstützung vieler Optionen für PDF-Dateien
- zusätzliche Funktionalität
  - Links aus dem PDF auf Webseiten etc.
  - Ausgabe von URLs (mit Sonderzeichen)
  - Setzen der Metainformationen in PDF-Dokumenten (Titel, Autor, Schlagworte, . . .)

Listing 22: Verweise auf externe Quellen mit `\href`

```
\href{<URL>}{<Text>}
```

- URL
  - relative Pfade erlaubt
  - `hyperref` versucht, den Typ zu erkennen (Webseite, lokales PDF-Dokument, Email-Adresse)

Grundregel

Das Paket `hyperref` sollte immer am Ende der Präambel geladen werden, da es viele LaTeX-Interna umdefiniert.

Zentrale Aspekte

UNI FREIBURG

- Das Rad neu zu erfinden ist selten eine gute Idee.
  Meist gibt es Pakete, die besser sind als eigene Lösungen.
- Viele Pakete dienen der logischen Textauszeichnung.
  Die Idee hinter LaTeX: Fokus auf Inhalt statt Formatierung.
- Wissenschaftliche Abschlussarbeiten stellen eine Reihe immer wiederkehrender Anforderungen.
- Die Liste erwähnter Pakete ist hochgradig subjektiv und der Versuch, häufige Anwendungsfälle zu berücksichtigen.
- Grundsätzlich sollte vor der Verwendung eines Pakets immer erst dessen Dokumentation konsultiert werden. ☺