

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

23. Jahrgang Heft 1/2011 Februar 2011

1/2011

Impressum

»Die T_EXnische Komödie« ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse dtk erstellt und per E-Mail oder Datenträger (CD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit pdfTeX 3.1415926-1.40.11-2.2 (TeX Live 2010) erstellt. Als Standard-Schriften kamen die Fonts TG Pagella und Bera Mono zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: dante@dante.de
dtkred@dante.de (Redaktion)

Druck: Konrad Triltsch Print und digitale Medien GmbH
Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohestadt

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit: Rudolf Herrmann Bertram Hoffmann Gert Ingold
Rolf Niepraschk Heiko Oberdiek Christine Römer
Volker RW Schaa Gert Seidl

Redaktionsschluss für Heft 2/2011: 15. April 2011

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

wahrscheinlich werden es die meisten unter Ihnen schon gemerkt haben: Diese Ausgabe unserer Zeitschrift erscheint in einer anderen Schriftart. Statt der Latin Modern verwenden wir die TG Pagella des \TeX -Gyre-Projekts, welches wesentlich von DANTE e. V. finanziell unterstützt wurde. Ziel des Projekts war und ist es, eine Überarbeitung der freien Type-1-Varianten der PostScript-Schriften vorzunehmen:

The project started in 2006 and will run for several years. It is funded by DANTE e. V., NTG, TUG, CS TUG, TUG India and GUST. It is an extensive remake and extension of the freely available 35 base PostScript fonts distributed with Ghostscript ver. 4.00. The important aspect of the project is providing not only the support for TeX but also the cross-platform OpenType format of the fonts. (<http://www.gust.org.pl/projects/e-foundry/tex-gyre/>)

Ein wesentlicher Unterschied zu dem freien URW-Klon der PostScript-Schrift Palatino ist insbesondere die Überarbeitung der Kapitälchen: Die Höhe der »kleinen Großbuchstaben« wurde neu festgelegt, so dass sie jetzt der Höhe der normalen Kleinbuchstaben entspricht. Zusätzlich werden die so genannten »oldstylenums« verwendet:

PALATINO-0123456789, wie es jetzt erscheint,

PALATINO-0123456789, mit URW-Klon und

PALATINO-0123456789, mit der originalen kommerziellen PostScript-Variante.

Die \TeX -Gyre-Fonts sind standardmäßig auf allen \TeX -Distributionen vorhanden. Für die Schreibmaschinenschrift hatten wir in der Vergangenheit die Luximono verwendet, die allerdings wegen ihrer Lizenz nicht in \TeX Live vorhanden ist. Die Bera Mono, die wir jetzt in einer skalierten Form verwenden, ist dagegen Bestandteil von \TeX Live.

Inhaltlich finden Sie in dieser Ausgabe die Fortsetzung der von Christine Römer begonnenen Reihe, die sich dieses Mal mit den Schriftoptionen beschäftigt.

Weitere Beiträge finden Sie zu den Themen »Sage \TeX «, »Lebenslauf und Bewerbungsschreiben«, »Kyrillische Schriftzeichen« und »Einlesen und Ausführen von \LaTeX -Code«. Insbesondere der Artikel zum kyrillischen Satz war wieder

mal eine Herausforderung für das Setzen dieser Ausgabe, denn das Laden mehrerer Schriftkodierungen brachte das System zeitweise komplett durcheinander, sodass auch schonmal die gesamte Ausgabe in kyrillischer Schrift erschien, was sicherlich nicht nur für mich das Lesen unserer Zeitschrift erschwert hätte ...

Die Rubrik »Tipps und Tricks« ist diesmal etwas umfangreicher, um auf unsere produktionstechnisch günstige Seitenzahl $n \cdot 16 + 4$ zu kommen, wobei n für eine natürliche Zahl und $+4$ für die vier Umschlagseiten stehen.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe mit T_EXnischen Grüßen,

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

da einer der Unterzeichner berufsmäßig zur Zeit sehr eingespannt ist, während der andere in der Ferne Urlaub macht, hat dieses Grußwort bei der Abstimmung zwischen beiden weite Wege um den Globus zurückgelegt (und wäre dabei fast zu spät gekommen).

Zunächst einmal hoffen wir, dass alle Mitglieder einen guten Start ins Jahr 2011 hatten.

Der Vorstand traf sich Anfang Januar zum ersten Mal seit seiner Wahl vollzählig zu einer Sitzung in Darmstadt. Neben den laufenden Geschäften wurde deshalb insbesondere darüber gesprochen, aus der Vergrößerung des Beirats Nutzen zu ziehen und die Verantwortlichkeit für die verschiedenen Tätigkeitsbereiche auf mehr Schultern zu verteilen und so auch Stellvertreter während beruflicher oder privater Abwesenheiten zu haben. Auf diese Weise soll immer ein erster Ansprechpartner für die Mitglieder zur Verfügung stehen. Eine Liste der Kontaktpersonen wird in absehbarer Zeit auf der Webseite von DANTE e.V. zu finden sein und auch in einer Ausgabe von »Die T_EXnische Komödie« veröffentlicht werden.

In der Sitzung wurde auch der Projektantrag *Extended Euler Font Proposal* von Khaled Hosny einstimmig angenommen. Hier geht es um die Fortführung der begonnenen Überarbeitung des AMS-Euler-Fonts von Hermann Zapf. Insbesondere soll der Font vom Type-1- ins OpenType-Format überführt und der Zeichensatz erweitert werden. So stehen zwar 92 % des Basis-Latin-Blocks (U+0000-007F) zur Verfügung, vom Latin-Supplement und -Extended-Block (U+0080-024F) dagegen weniger als 1 %. Die Abdeckung bei mathematischen Symbolen, Operatoren und Zeichen ist zwar besser (ca. 20 %), muss aber vervollständigt werden. Bei der Entwicklung soll besonderes Augenmerk auf die speziellen OpenType-Extensions gelegt werden, die für das Setzen mathematischer Texte unabdingbar sind. Der im Rahmen dieses Projektes entstehende Font wird nur mit den neuen T_EX-Implementierungen (X_YT_EX, LuaT_EX) nutzbar sein, wird somit also den AMS-

Euler-Font nicht ersetzen. Der komplette Projekt-Antrag wird in der nächsten Ausgabe von »Die T_EXnische Komödie« veröffentlicht werden.

Weiterhin möchten wir noch einmal auf die diesjährige Frühjahrstagung von DANTE e.V. in Bremen vom 30. März bis zum 1. April hinweisen. Auf der Tagungsseite <http://www.dante.de/events/dante2011.html> finden Sie das dazugehörige Programm, die Möglichkeit sich anzumelden und weitere Infos. Wer noch Interesse daran hat, einen kurzen Vortrag zu halten, wird gebeten sich baldmöglichst per E-Mail an dante2011@dante.de zu melden. Wir freuen uns auf viele bekannte und auch neue Gesichter im »Hohen Norden«.

Zu guter Letzt möchten wir noch die Gelegenheit nutzen, auf die diesjährige EuroT_EX-Tagung in Bachotek bei Brodnica, Polen hinzuweisen. Diese wird vom 29. April bis 3. Mai unter dem Thema »Aesthetics and Effectiveness of the Message, Cultural Contexts« stattfinden. Das besondere Flair am Tagungsort, an dem seit vielen Jahren die jährliche Tagung der polnische T_EX-Users-Group GUST als »Bachot_EX« stattfindet, ist dabei besonders zu empfehlen. Wer Interesse hat, kann sich gern auf der Tagungsseite <http://www.gust.org.pl/bachotex/2011-en> oder in Bremen bei *alten* Bachot_EX-Besuchern über die Tagung näher informieren.

Bis dahin verbleiben wir mit den besten Wünschen,

Volker RW Schaa	Adelheid Grob
Vorsitzender	Stellvertretende Vorsitzende

Bretter, die die Welt bedeuten

Gewichten – Wichtiges und Unwichtiges mit \LaTeX
markieren.

Teil 2: Auszeichnungen

Christine Römer

Auszeichnungen dienen primär der Lesesteuerung. Die Intensität, mit der die entsprechenden typografischen Hervorhebungsmittel angewendet werden, hängt von der Textsorte, dem Inhalt, dem Leserkreis und dem angestrebten Zweck ab. Außerdem sind sie auch dem Zeitgeist unterworfen. Deshalb sind allgemeine Anwendungsregeln, die dies nicht berücksichtigen, in der Regel wenig sinnvoll. \LaTeX ermöglicht die Umsetzung aller Auszeichnungsarten, wenngleich es bei einigen nicht voll ausgebauten Schriften Einschränkungen gibt.

Einleitung

In der mündlichen Kommunikation erfolgen Hervorhebungen über prosodische und intonatorische Gestaltungsmittel. Dies sind unter anderem Modifikationen in der Sprechgeschwindigkeit, Sprechstärke, Stimmfarbe oder Rhythmik. Das Setzen von Akzenten und Pausen gehört ebenso dazu. Im Beispiel (1) verändert sich die Äußerungsbedeutung, je nachdem ob Subjekt oder Objekt durch Betonung hervorgehoben wird. (Die // werden in grammatischen Texten teilweise als Begrenzungszeichen für Tongruppengrenzen eingesetzt.)

- (1) // **wir** beraten sie //
// wir beraten **sie** //

Intensitätsgrade lassen sich auch lexikalisch auf unterschiedliche Weise codieren; beispielsweise über Quantoren, Partikel oder Dimensions- und Wertungsadjektive (wie in (2)).

- (2) Eine sehr große Auswahl an ganz vielen tollen Sachen.

Syntaktisch kann auch durch abweichende Satzgliedstellung oder Ausrahmung hervorgehoben, fokussiert werden (3).

(3) Jetzt kommt auf die Bühne Renée Fleming.

In der Typografie wird das besondere Hervorheben einzelner oder mehrerer Wörter oder ganzer Textpassagen als Auszeichnung bezeichnet. Dabei hängt es von der Textsorte ab, wie bedeutsam Auszeichnungen sind. Friedrich Schiller hatte 1793 in den »Kallias-Briefen« darauf hingewiesen, dass sich beim literarischen Lesen die Form gegenüber dem Inhalt nicht in den Vordergrund schieben darf.

Die Natur der Sprache [...] muß in der ihr gegebenen Form völlig untergehen, der Körper muß sich in der Idee, das Zeichen in dem Bezeichneten, die Wirklichkeit in der Erscheinung verlieren. [6, S. 433]

Die typografische Gestaltung epischer Literatur muss die Prinzipien des »linearen Lesens« berücksichtigen.

Der Leser soll sich in einen Roman ungestört versenken können. Damit das erstaunliche Phänomen – nämlich dass der Leser durch die Buchstaben hindurch die Handlung sehen, ja regelrecht halluzinieren kann – gelingt, bedarf es einer visuell zurückhaltenden Typografie: der klassischen Buchtypografie [...]. [3, S. 80]

Anders ist das beim »informierenden Lesen«, beim Lesen von Zeitungen oder Sachbüchern. Hier wird nicht ein ungestörtes Versinken in den Text angestrebt.

[Durch eine ökonomische Informationsgewichtung] soll dem Leser ermöglicht werden, in möglichst kurzer Zeit dem Text gezielt Informationen entnehmen zu können. Der Text muss so aufbereitet sein, dass ein diagonales Querlesen möglich ist. [Weshalb die Typografie hier] – anders als beim linearen Lesen – nicht mehr »unsichtbar« bleiben kann, sondern eindeutige, visuelle Strukturierungen setzen muss. [3, S. 81]

In [3] wird noch eine dritte typografische Lesart (in Schulbüchern beispielsweise) unterschieden: die selektierende Lesart. Ein »Patchwork-Lesen«, das mit ganz verschiedenen textlichen Einheiten (Sachtexten, Arbeitsanregungen, Tabellen und Merksätzen) konfrontiert wird. Die Typografie muss hier das Kunststück vollbringen, dass diese Textbausteine sowohl in Verbindung miteinander als auch unabhängig voneinander gelesen werden können.

Das Repertoire der selektierenden Gestaltungsart ist sehr umfassend, so verfügt hier der Typograf u. a. über verschiedene Schriftarten und -schnitte, Schriftgrößen, Überschriftenhierarchien, Raster- und Farbunterlegungen, Piktogramme und Grafiken. [3, S. 81]

Besonders das Typo-Design von Werbetexten ist dadurch gekennzeichnet, dass die Schrift nicht primär zur Informationsvermittlung dient, sondern ein eigener semiotischer Code ist, der zusätzliche denotative und konnotative Informationen liefert [1], wie in der folgenden Werbeanzeige für Holzbuchstaben¹ zu sehen ist.

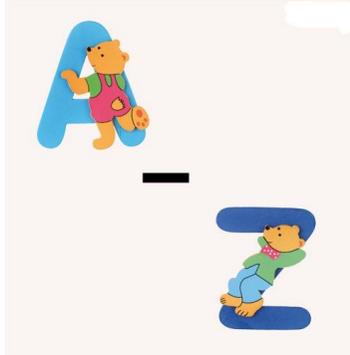


Abbildung 1: Buchstaben als semiotischer Code

Auszeichnungsarten

Nach [2] können wir die nachfolgend angesprochenen Auszeichnungsarten unterscheiden.

Kursivschnitte – `\textit{Kursivschnitte}` – kommen als unaufdringliche Auszeichnungsvarianten gegenüber einer geradestehenden Schrift zur Anwendung. Allerdings entfällt diese Funktion weitgehend, wenn die kursive Auszeichnung in Fachsprachen eine spezielle Funktion hat, wie dies in der Linguistik der Fall ist, wo sprachliche Beispiele damit im Text als solche markiert werden.

Mit dem Terminus *Italics* meint man den echten Kursivschnitt, der Schreibschriftähnlichkeit hat. Die *Oblique* – `\textsl{Oblique}` – ist lediglich eine schräg gestellte Version des geraden Schriftschnitts. Nur wenige Schriften haben sowohl einen Italics- als auch einen Oblique-Schriftschnitt. Am unterschiedlichen Aussehen des Buchstaben a kann man sehen, ob es sich um einen echten kursiven Schnitt handelt: *A a A a*. Das echte kursive (links) sieht handschriftlich und das unechte (rechts) gedruckt aus.

In einem mit gerader Schrift gesetzten Text wird mit dem Befehl `\emph{}` (*emphasis*) auf Hervorhebung mittels kursiver Schrift umgeschaltet. In einer kursiven Schrift geschieht dann die Hervorhebung mittels gerader Schrift.

¹ <http://www.kleiner-schatz.de/Happy-Nature-Buchstaben-Tiere>.

KAPITÄLCHEN (*small caps*, kleinere Großbuchstaben)

– `\textsc{Kapitälchen}` – und VERSALIEN (Großbuchstaben) werden zur Auszeichnung von Namen und Überschriften eingesetzt. Nach Meinung von [7, S. 186] sind beide schlecht lesbar, aber Kapitälchen »fügen sich besser ins Satzbild ein. Sie sollten aber etwas spationiert werden.« Sie kämen jedoch nur für eine Serifenschrift in Frage. Eine serifenlose Schrift in Kapitälchen sehe »einfach unprofessionell« aus.

Sperren zur Hervorhebung wird ebenso heftig abgelehnt, wie die Verwendung von Großbuchstaben, »weil es den „Grauwert“ des Textes ändert und damit den Lesefluss stört.« [4, S. 94] Es wird auch, wie schon angesprochen, genutzt, um großgeschriebenen Text (SPERREN) zu verbessern. Dies kann u. a. mit dem Paket `soul` geschehen, das jedoch nicht die Umlautezeichencodierung mit Hilfe von `utf8` vornehmen kann, weshalb diese umschrieben werden müssen (`Übung` `\so{"Übung}`). Das neue `soulutf8`-Paket von Heiko Oberdiek erweitert das `soul`-Paket und ermöglicht `utf8`- und `utf8x`-Kodierungen und erlaubt das Arbeiten mit plain \TeX , \LaTeX und $\epsilon\text{-}\TeX$.

```
\usepackage{soul}

\so{Sperren}
\caps{SPERREN}
```

(Halb)fette Schrift (*bold face*) sticht im Text mehr hervor als kursiv formatierte. Die in der Typografie üblichen Schriftstärken bzw. Fettgrade *extraleicht*, *leicht*, *mager*, *normal*, *halbfett*, *fett*, *extrafett* und *ultrafett* sind in \LaTeX in einem Text für die Veränderung einzelner Textteile nicht realisierbar. Es steht hier nur der Wechsel zwischen *normal* und *halbfett* zur Verfügung. Dies kann auch auf Schriften als Ganzes angewendet werden (`\normal font` bzw. `\bfseries`). Die Schriften unterscheiden sich zum Teil deutlich in ihrer Grunddicke. Für einen Aushang beispielsweise, der weithin gelesen werden soll, ist dann eine dickere Schrift angebrachter. Manchmal wird eine dickere Schrift auch für die Kapitelüberschriften genommen, wie man in dem kürzlich erschienen »Inside WikiLeaks« von D. Domscheit-Berg sehen kann. Als Brotschrift wurde dort die Sabon² und für die Überschriften die Compacta³ benutzt.

```
\usepackage[T1]{fontenc}
\usepackage{cyklop}

\usepackage[T1]{fontenc}
\usepackage[light,condensed,math]{iwona}
```

² [http://de.wikipedia.org/wiki/Sabon_\(Schriftart\)](http://de.wikipedia.org/wiki/Sabon_(Schriftart))

³ <http://www.ascenderfonts.com/font/compacta.aspx>

Die dicke Cyclop. Die dünne Iwona.

Abbildung 2: Dicke und dünne Schriften

Mit `\fontseries{}` kann die Schriftbreite mit der Schriftdicke eingestellt werden (siehe [4, Kap. 7.9.1]).

Unterstreichen beeinflusst das Schriftbild deutlich, weil die Position der Unterstreichungslinien von der Buchstabenlinienhöhe abweicht und bei Glyphen mit Unterlängen diese zwar mit `\underline{}` nicht durchgestrichen werden, die Linie aber noch weiter herunterrutscht und die nächste Zeile gedrungen aussieht. Deshalb wird in typografischen Ratschlägesammlungen oftmals formuliert »Unterstreichen vermeiden«, beispielsweise in »Goldene Regeln der Typographie« von Gerhard Kahofer (http://www-sst.informatik.tu-cottbus.de/~db/doc/Schreiben_Vortragstechnik/Typographie/GoldReg.htm).

Das Paket `ulem` ermöglicht unter anderem doppelte und unterschlängelte Unterstreichungen; ~~Durchstreichen~~ und ~~Ausstreichen~~ sind auch möglich. Die Silbentrennung muss dann in den ausgezeichneten Passagen von Hand mit weichem Trennstrich `\-` vorgenommen werden. Beachtet werden muss, dass in diesem Paket `\emph{}` anders definiert ist. Es wird hier zum einfachen Unterstreichen benutzt. Wenn man dann an anderen Textstellen `\emph{}` zum Hervorheben benutzt, erscheinen diese auf einmal mit Unterstreichung und ohne Schrägstellung. Dem kann beispielsweise damit abgeholfen werden, dass man mit `\textit{}` hervorhebt. Die einfache Unterstreichung mit `\emph{}` hat den Vorteil, dass bei der Unterstreichung Zeilenumbrüche vorgenommen werden können. Das Paket `ulem` bietet unabhängig von der Option `normalem` `\uline` zur einfachen Unterstreichung.

```
\usepackage{ulem}

\uuline{doppelte} und \uwave{unterschlängelte} Unterstreichungen,
\sout{Durchstreichen} und \xout{Ausstreichen}
```

Wenn mehrere Wörter mit `\underline{}` bzw. `\emph{}` oder `\uline{}` gekennzeichnet werden, sind diese durchgängig unterstrichen (4):

(4) Das ist eine Aussage.

Wenn man die Wörter wortweise unterstreichen will, muss man sie einzeln mit dem Unterstreichungsbehl versehen. Falls dann, wie im Beispiel 5, ein Wort mit Unterlänge dabei ist, sieht das nicht gefällig aus.

(5) Das ist eine Aussage.

Man kann auch Zusammenfassungen oder Ähnliches **anstreichen**. Dies kann beispielsweise mit dem Paket `framed` geschehen.

```
\begin{leftbar}
Man kann auch Zusammenfassungen oder Ähnliches {\large anstreichen}. Dies kann
beispielsweise mit dem Paket \texttt{framed} geschehen.
\end{leftbar}
```

Hervorhebung durch **Größe** – {\Huge Größe} – ist eine einfache, aber effektive Möglichkeit zur Hervorhebung und Aufmerksamkeitssteuerung, besonders wenn sie mit einer auffälligen Positionierung kombiniert wird. Mit dem Verwenden von unterschiedlichen Schriftgrößen können bestimmte Textteile betont oder bei Verkleinerung als weniger wichtig markiert werden.

Andere Schriftarten in einem Printerzeugnis zu verwenden, kann das Layout unübersichtlich und unruhig für den Betrachter werden lassen. Andererseits können Schriftmischungen »grundsätzlich die Lesemotivation [fördern] und ein schnelles Auffinden bestimmter Inhalte bzw. ein rascheres Querlesen [ermöglichen]. Deshalb stellt die Schriftmischung einen wichtigen Parameter dar, um schwierige und komplexe Inhalte schnell und erfolgreich zu vermitteln.«⁴ Die Typografie propagiert deshalb verschiedene Grundregeln zur Schriftmischung. Beispielsweise⁵

- Schriften mit gleichem Duktus und ähnlichen Proportionen lassen sich gut mischen.
- Es sollten deutliche Kontraste gesetzt werden, das erhöht die Aufmerksamkeit.

Ein Schriftwechsel im Dokument kann unter anderem mit dem Befehl `\newfont` deklariert und dann mit dem Befehl `\<Name>` abgeändert werden. So kann die Sütterlin-Schreibschrift mit `\newfont{\suet}{suet14}` aufgerufen mit `\suet <Text> \normalfont Text` in dieser Schrift eingefügt werden. `\newfont` geht allerdings am NFSS2 von L^AT_EX vorbei und sollte daher nur verwendet werden, wenn es keine L^AT_EX-Anpassungen für die Schrift gibt.

Erste Wahl für Schriftwechsel ist der Befehl `fontfamily{}`. Mit ihm kann auch für einen kleinen Textabschnitt eine andere Schrift eingerichtet werden.

```
{\fontfamily{<Font>}\selectfont Andere Schriftart}
```

⁴ <http://www.webmasterpro.de/design/article/typografie-12-wichtige-grundlagen-fuer-den-richtigen-einsatz-von-schriften-teil2.html>

⁵ <http://www.ediscio.de/lernkarten/94859>

Für einen Zeichensatzwechsel⁶ benötigt man auch den richtigen Code für . Mit `\fontfamily{pzc}` wird beispielsweise auf die Schrift Zapf Chancery (pzc) umgeschaltet und mit `\fontfamily{pag}` auf Avant Garde.

Farbige Schriftauszeichnung – `\textcolor{red}{Farbige}` – kann die Aufmerksamkeit auf Wichtiges lenken und auch der Lesesteuerung beim Auffinden von Informationstypen dienen, wenn eine konsistente Verwendung der gleichen Farbe für spezifische Sachverhalte realisiert wird. Mit dem Paket `soul` kann auch farbig unterstrichen werden, und mit `colortbl` können Zeilen in Tabellen eingefärbt werden. Zum Einsatz von farbigen Kennzeichnungen in mathematischen Texten siehe [8]⁷.

Negativschrift bzw. Inversschrift erzeugt einen hohen Kontrast und ist damit ziemlich auffällig, dies trifft besonders auf einen dunklen Hintergrund mit heller Schrift zu. Man kann mit `\colorbox{<Hintergrundfarbe>}{<Text>}` und `\fcolorbox{<Rahmenfarbe>}{<Hintergrundfarbe>}{<Text>}` Textteile farbig unterlegen.

```
\colorbox{black}{\textcolor{white}{\large\textbf{Negativschrift}}}
```

Mit den Paketen `framed` und `xcolor` kann man beispielsweise Merksätze, die über mehrere Zeilen gehen, farbig hinterlegen. Die gewünschte Hintergrundfarbe muss mit `shadecolor` definiert werden. Der Merksatz wird dann in die Umgebung `shaded` gesetzt.

```
\usepackage{framed,xcolor}
\colorlet{shadecolor}{red!25}
\begin{shaded}
<Text>
\end{shaded}
```

Das Paket `mdframed` kann farbige Rahmen setzen, wie in »Die T_EXnische Komödie« Heft 3/2010 von dem Mitautor des `mdframed`-Pakets Marco Daniel beschrieben wurde.

INITIALE können auf den Beginn eines Kapitels oder Artikels hinweisen, im Text wichtige Punkte anzeigen oder nur als Schmuckelement dienen. Sie haben eine lange Tradition und sind bis auf die ausgeschmückten Handschriften aus der Zeit vor der Einführung des Buchdrucks zurückführbar.

⁶ Einen instruktiven Überblick über die Schriften gibt <http://astro.uni-tuebingen.de/~nagel/pub/Kolleg/Latex/VorlesungSchriften.pdf>.

⁷ <http://www.ctan.org/tex-archive/info/math/voss/mathCol/mathCol.pdf>

```
\usepackage{lettrine}

\lettrine[lines=3,findent=4.5pt,nindent=-0.5pt]{I}{\textsc{nitiale}}
```

In [5] werden außerdem noch die folgenden Auszeichnungsarten beschrieben.

Dreidimensionale Buchstaben »geben einem typografischen Design mehr Gewicht und Wirkung. [...] Tiefe und Masse sorgen dafür, dass die Schrift bei einer dreidimensionalen Nutzung wie z. B. bei Leit- und Orientierungssystemen und Ausstellungsgrafik wahrgenommen wird, auch Druck- und Digitalprojekte können davon profitieren.« [5, S. 80]

```
\usepackage{pst-text,pst-blur,pst-slope,graphicx}
\DeclareFixedFont{\RM}{T1}{ptm}{b}{n}{3.5cm}
\DeclareFixedFont{\Rm}{T1}{ptm}{m}{n}{2mm}

\psset{shadow=true,blur=true,shadowsize=10pt,blurradius=5pt}
\begin{pspicture}(\linewidth,3.5cm)
\psset{fillstyle=ccslopes}
\resizebox{\linewidth}{!}{\pscharpath{\RM Dreidimensionale Buchstaben}}
\end{pspicture}
```

Diesem Zweck können auch andere grafische Zeichenmanipulationen, die mit dem Paket `pst-text` möglich sind, dienen. Damit kann man beispielsweise kleine »Textbilder« schaffen (wie Abbildung 3; zum Quellcode siehe <http://www.tug.org/PSTricks/main.cgi?file=Examples/textpath>), die in der Lage sind, »auf der Gefühlsebene schneller und nachhaltiger Informationen zu transportieren, als dies der Text jemals könnte« [7, S. 226].

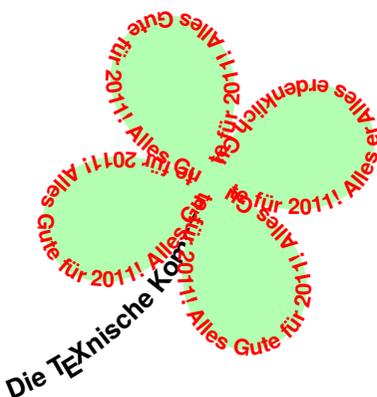


Abbildung 3: Ein Textbild

Dekonstruktive Typen und Schnörkel

Einzelne Teile von typografischen Formen werden als Schmuck, Orientierungshilfen oder als Muster verwendet. Verschnörkelte Schmuckelemente »wirken am besten, wenn sie mit sehr schlichten Typographieelementen oder Layouts kombiniert werden, da so ihr überladener Charakter ausgeglichen wird.« [5, S. 20]

```
\usepackage{pstricks}

\begin{pspicture}(6,0.5)
\pscurve[linecolor=gray](3.1,0)(3,-0.3)(6.5,0)
\pscurve[linecolor=gray](3.2,0)(3,-0.3)(7,0)
{\large Dekonstruktive Typen und Schnörkel}
```

Hinzugefügt werden könnte noch das **Rotieren** von Textteilen, das beispielsweise für lange Überschriften in Tabellenspalten genommen wird.

```
\usepackage{rotating}

\begin{rotate}{20}\large Rotieren\end{rotate}
```

Harmonische vs. kontrastreiche Auszeichnungen

Bei [7, S. 186] wird zwischen harmonischer und kontrastreicher Auszeichnung unterschieden. Die erstere verändert die Grauwirkung des Grundtextes nicht, und man zeichnet deshalb im Prinzip nur mit Kursivschnitt aus. Sie ist somit für das vollständige und literarische Durchlesen besonders geeignet. Die kontrastreichere Auszeichnung dagegen unterstützt das informierende und selektierende Lesen. Sie kann für den Lesenden mehrere Einstiege in den Text vorsehen. [5, S. 110] schreibt auch von »Typografie, die ins Auge fällt«, auffallende Typografie »als die Fähigkeit, in Kombination mit einem überzeugenden Bild ein Gleichgewicht zu schaffen – zusammen gelingt es ihnen eine starke Aussage zu formulieren.«

Schlussfolgerung

Ein gedruckter Buchstabe und ein schön gestaltetes Buch sind etwas Beständiges, Bleibendes im Vergleich zu dem schnellen Zugriff zu einer Information im Internet und dessen Flüchtigkeit der Wiedergabe am Bildschirm. [9, S. 96]

Wenn etwas über den flüchtigen Augenblick Hinausgehendes geschaffen werden soll, gilt es zu bedenken, dass sich auch beim Thema typografische Auszeichnungen die Volkweisheit bewahrheitet, dass bei sinnvollen Anliegen der Zweck

in einem gewissen Rahmen die Mittel heiligt. Wie viele und welche Auszeichnungsmittel man verwendet, hängt nämlich maßgeblich vom Zweck, dem Inhalt, der Textsorte und dem soziolektalen Kommunikationsbereich (Wissenschaft, Freizeit, Werbung, ...) ab. Bei der Verwendung von \LaTeX hängt die Umsetzung der zahlreichen Möglichkeiten auch von der/den gewählten Schriften ab. Deren Auswahl sollte folglich auch unter Einbeziehung ihrer Funktionalität erfolgen. Von allgemeinen Ratschlägen, wie »man darf keinen Fettdruck in ansprechender Typografie verwenden«, sollte deshalb genauso Abstand genommen werden, wie von der Behauptung, man bräuchte für \LaTeX eigentlich keine Fußnoten(befehle). Goldene Regeln von vor 100 Jahren sollten mindestens hinterfragt werden, da sich sowohl die Sehgewohnheiten als auch der »typografische Zeitgeist« aufgrund der veränderten Wirklichkeit geändert haben (wie neue Medien oder »Reizüberflutung«).

Literatur

- [1] Nina Janich: *Werbesprache*. narr studienbücher. Gunter Narr Verlag: Tübingen 1999.
- [2] Robert Klanten, Mika Mischler und Silja Bilz (Hrsg.): *Der kleine Besserwisser Grundwissen für Gestalter*. Die Gestalten Verlag: Berlin 2007.
- [3] Thorsten Lorenz und Cornelia Glaser: *Die Typografie des Schulbuchs und die Steuerung des Sinns*. In: *Der Deutschunterricht* 4/2010, S. 78–89.
- [4] Frank Mittelbach & Michel Goossens: *Der \LaTeX -Begleiter*. Pearson Studium: München, 2. überarbeitete und erweiterte Auflage, 2005.
- [5] Ina Saltz: *Typografie. 100 Prinzipien für die Arbeit mit Schrift*. Stiebner Verlag: München 2009.
- [6] Friedrich Schiller: *Kallias oder Über die Schönheit*. In: Ders., *Sämtliche Werke* Bd. V. Wissenschaftliche Buchgesellschaft: Darmstadt, 1993, 394–433.
- [7] Ralf Turtschi: *Praktische Typografie*. Verlag Niggli: Sulgen 1994.
- [8] Herbert Voß: *Farbige Mathematik*. In: *Die \TeX nische Komödie*, 2/2004, S. 81–87.
- [9] Hermann Zapf: *Alphabetgeschichten. Eine Chronik technischer Entwicklungen*. Linotype GmbH: Bad Homburg 2007.

SageTeX

Günter Rau

Einleitung

Bei dem Programm, das hier vorgestellt werden soll, handelt es sich um ein Mathematik-Softwaresystem, bestehend aus fast 100 Open-Source-Komponenten, welche über ein gemeinsames Python-Interface zugänglich sind. Für dieses sehr umfangreiche Mathematik-Softwaresystem gibt es ein L^AT_EX-Paket, um Ergebnisse direkt in ein L^AT_EX-Dokument einfügen zu können. Dieses Softwaresystem heißt Sage und lässt sich selbstverständlich plattformunabhängig installieren (siehe <http://www.sagemath.org>). Unter der Webadresse <http://www.sagenb.org> kann man auch online das Programm kennenlernen. Die Beispiele in diesem Artikel beziehen sich auf die Version 4.6 und wurden unter Debian-Linux (Lenny) erstellt.

Erste Schritte

Die Installation von Sage bereitet keine Schwierigkeiten. Bei der Nutzung von Sage ist es sehr hilfreich, wenn man die Programmiersprache Python in ihren Grundzügen beherrscht. Dann kann man nämlich leicht eigene Funktionen bzw. Prozeduren programmieren. Wir beginnen mit einfachen Eingaben in der Konsole:

```
sage: 5+3
```

ergibt

```
sage: 5+3
8
```

was nicht weiter verwundert.

Es lassen sich aber auch Funktionen leicht definieren und deren Ableitung bilden.

```
sage: f(x) = x^2
sage: f.diff(x)
x |--> 2*x
```

Auch Gleichungen können leicht gelöst werden.

```
sage: solve(x^2 == 16,x)
[x == -4, x == 4]
```

Auf die einzelnen Lösungen kann man folgendermaßen zugreifen.

```
sage: lsg = solve(x^2 == 16,x)
sage: lsg[0]
x == -4
sage: lsg[0].rhs()
-4
```

In der ersten Zeile wird die Lösung der Gleichung $x^2 = 16$ berechnet und als Gleichungsliste der Variablen `lsg` zugewiesen. In der zweiten Zeile wird auf das erste Element dieser Liste zugegriffen. Den rechten Teil der Gleichung $x = -4$ erhält man, wenn man die `Righthandside`-Methode auf $x = -4$ anwendet.

Sage und L^AT_EX

Mit Sage können mathematische Ausdrücke wie z. B. $1/2$ in L^AT_EX-Code umgewandelt werden.

```
sage: latex(1/2)
\frac{1}{2}
sage: latex(3*x^2-x+exp(-x))
3 \, x^{2} - x + e^{\left(-x\right)}
```

Als Bonbon gibt es nun ein L^AT_EX-Paket (`sagetex.sty`), mit dessen Hilfe man in einem L^AT_EX-Dokument Sage benutzen kann. Eine einfache Beispieldatei sieht folgendermaßen aus:

```
\documentclass[a4paper,12pt]{article} % Datei Beispiel1.tex
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{sagetex} % Sage wird eingebunden

\begin{document}
$3+5=\sage{3+5}\$ \qquad % Sage berechnet die Summe 3+5
$\sage{(x+2)*(x-3)} = \sage{expand((x+2)*(x-3))}$\
Die Ableitung von  $f(x)=x^2-x$  ist
 $f'(x)=\sage{diff(x^2-x,x)}$ 
\end{document}
```

Diese Datei wird zuerst mit `(pdf)latex` bearbeitet. Dabei wird die Datei `Beispiel1.sage` erzeugt. Dies ist eine Datei, die alle Sage-Befehle aus der L^AT_EX-Datei enthält. Sage erzeugt aus dieser Datei die Datei `Beispiel1.sout`. Durch einen erneuten Aufruf von `(pdf)latex` werden die Daten aus `Beispiel.sout` in das L^AT_EX-Dokument eingebaut. Das Ergebnis ist in unserem Fall:

$$3 + 5 = 8$$

$$(x + 2)(x - 3) = x^2 - x - 6$$

Die Ableitung von $f(x) = x^2 - x$ ist $f'(x) = 2x - 1$

Darüber hinaus ist es auch möglich, mehrzeiligen Sage-Code in das L^AT_EX-Dokument einzufügen. Auch eigene Funktionen lassen sich definieren, wie folgendes Beispiel zeigt.

```
\begin{document}
\begin{sagesilent}
def Wendestelle(term):
    f(x) = term
    f2(x) = diff(diff(f(x),x),x)
    lsg = solve(f2(x)==0,x)
    return lsg
\end{sagesilent}

Die Funktion  $f(x) = \text{sage}\{1/3*x^3-x^2\}$  hat
bei  $\text{sage}\{\text{Wendestelle}(1/3*x^3-x^2)[0]\}$  eine Wendestelle.
\end{document}
```

Nach der Bearbeitung mit L^AT_EX und Sage erhält man:

Die Funktion $f(x) = \frac{1}{3}x^3 - x^2$ hat bei $x = 1$ eine Wendestelle.

Aufgabenblätter mit Lösungen

Der Autor verwendet Sage, um Aufgabenblätter mit den zugehörigen Lösungen automatisch zu erstellen. Dazu wird das L^AT_EX-Kommando `\Aufgabe` mit zwei Parametern definiert. Der erste Parameter enthält die Aufgabenstellung und der zweite die zugehörige Lösung.

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{sagetex}

\newif\ifANGABE

\newcommand\Aufgabe[2]{ \ifANGABE #1 \else #2 \fi }

\newcommand\sageGleichung[1]{%
  \ifANGABE
    \ensuremath{\sage{#1}}
  \else
    \ensuremath{\sage{solve(#1,x)}}
  \fi}
```

```

\newcommand\sageAbleitung[1]{%
  \ifANGABE
    \ensuremath{\sage{#1}}
  \else
    \ensuremath{\sage{diff(#1,x)}}
  \fi}

\newcommand\Aufgabenliste{%
\begin{enumerate}
  \item \Aufgabe{Löse die Gleichung:  $2x+3 = 11$ }%
    {$x=4$}
  \item \Aufgabe{Löse die Gleichung \sageGleichung{3*x+1 == 22}}%
    {\sageGleichung{3*x+1 == 22}}
  \item \Aufgabe{Löse die Gleichung \sageGleichung{1/2*x^2 == 8}}%
    {\sageGleichung{1/2*x^2 == 8}}
  \item \Aufgabe{Bilde die erste Ableitung von
    $f(x)=\sageAbleitung{x^3-\cos(x)}$}%
    {\sageAbleitung{x^3-\cos(x)}}
\end{enumerate}%
}

\begin{document}
\ANGABEtrue
\Aufgabenliste
\newpage
\ANGABEfalse
\Aufgabenliste
\end{document}

```

Der Schalter `\ifANGABE` entscheidet darüber, ob die Aufgabenstellung oder die Lösung gesetzt werden soll.

Bei obigem Beispiel erhält man das Resultat:

1. Löse die Gleichung: $2x + 3 = 11$
2. Löse die Gleichung $3x + 1 = 22$
3. Löse die Gleichung $\frac{1}{2}x^2 = 8$
4. Bilde die erste Ableitung von $f(x) = x^3 - \cos(x)$

und auf der zweiten Seite:

1. $[x = 4]$
2. $[x = 7]$
3. $[x = (-4), x = 4]$
4. $3x^2 + \sin(x)$

Matrizen und Vektoren

Auch das Rechnen mit Vektoren und Matrizen gestaltet sich mit Sage ganz einfach. Dabei ist zu beachten, dass Vektoren grundsätzlich als Zeilenvektoren interpretiert werden.

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{sagetex}

\begin{document}
\begin{sagesilent}
A=matrix(QQ,2,2,[[1,2],[3,4]])
a=vector(QQ,[1,2,3])
b=vector(QQ,[4,5,6])
\end{sagesilent}

Das Inverse der Matrix  $A=\text{sage}\{A\}$  ist
 $A^{-1}=\text{sage}\{A^{-1}\}$ , auch transponieren
ist ganz einfach:  $A^t=\text{sage}\{A.transpose()\}$ 

Das Skalarprodukt ist
 $\text{sage}\{a.transpose()\}\bullet\text{sage}\{b.transpose()\} = \text{sage}\{a*b\}$ 
\end{document}
```

Dieses Listing zeigt den Code für das Ergebnis:

Das Inverse der Matrix $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ ist $A^{-1} = \begin{pmatrix} -2 & 1 \\ 3 & -\frac{1}{2} \end{pmatrix}$, auch transponieren ist ganz einfach: $A^t = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$

Das Skalarprodukt ist $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \bullet \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} = 32$

Schluss

Dies waren nur ein paar wenige Beispiele, die einen kleinen Eindruck vermitteln sollen, was mit diesen mächtigen Werkzeugen L^AT_EX und Sage gemacht werden kann. Sage wird ständig weiterentwickelt. Zwischen der Version 4.5 und 4.6 hat die Anzahl der Dateien um Hunderte zugenommen. Dieser kleine Beitrag kann nur dazu dienen, etwas Appetit zu machen.

Kyrillische Schriftzeichen im \LaTeX -Dokument

Patrick Oßmann

Im westeuropäischen Sprachraum etablierte sich die Verwendung lateinischer Buchstaben. Meist ist die Verwendung dieser Schriftzeichen auch ausreichend, um im Alltag »überleben« zu können; bei der Beschäftigung mit komplexeren mathematischen Zusammenhängen stößt man teilweise an die Grenzen der Verwendung lateinischer – und auch griechischer – Schriftzeichen. Gelegentlich ist es notwendig, kyrillische Schriftzeichen in bestimmten Situationen¹ in ein \LaTeX -Dokument einzubinden. Dies erscheint im ersten Moment trivial, was es – wie sich herausstellte – nicht ist. Der vorliegende Beitrag beschäftigt sich mit diesem Thema und soll im Idealfall auch als eine Anleitung zum Einbinden kyrillischer Schriftzeichen in ein mit lateinischer Schrift (in T1-Schriftkodierung) gesetztes \LaTeX -Dokument dienen.

Input- und Font-Kodierung

Die Eingabekodierung definiert die Beziehung zwischen den 8-Bit-Zeichen der .tex-Datei und ihrer Interpretation, das heißt, die Beziehung zwischen den Zeichen, die im Texteditor eingegeben (angezeigt) werden und der im Betriebssystem hinterlegten binären Kodierung [6, Seite 7]. Viele Betriebssysteme unterscheiden sich dabei in der Eingabekodierung. Generell weist man den \LaTeX -Interpreter durch den Aufruf des Pakets `\usepackage[<Optionen>]{inputenc}`² an, auf eine bestimmte Eingabe mit einer korrespondierenden Verarbeitung zu reagieren beziehungsweise auf eine definierte Eingabe die im Betriebssystem hinterlegte Kodierung zu nutzen. Die im Betriebssystem hinterlegte Eingabekodierung wird der *verwendete Zeichensatz* genannt. Der korrekte Gebrauch des Pakets `inputenc` wird im Verlauf dieses Dokuments vorausgesetzt.

Nachdem durch die Eingabekodierung definiert wird, wie ein im Editor eingegebenes Zeichen ver- und entschlüsselt werden soll, muss die Ausgabe eines eingegebenen Zeichens ebenso definiert werden. \TeX muss mitgeteilt werden, wie ein im Editor dargestelltes Zeichen letztlich in der Ausgabe repräsentiert werden soll. Hierzu dient das Paket `fontenc`. Dieses lädt (mit den entsprechenden

¹ Beispielsweise die durch das kyrillische Symbol »III« repräsentierte Dirac-Stoß-Folge. IPA-Aussprache nach [1, Seite 7]: <ʃ >.

² Statt *option* wird z. B. `utf8` für GNU/Linux Systeme, `ansinew` für MS-Windows- oder `applemac` für Macintosh Betriebssysteme angegeben.

Informationen), wie die zu setzenden Schriftzeichen geometrisch beschrieben werden. Letztendlich wird also über das Paket `fontenc` dem Interpreter mitgeteilt, welches Zeichen auf eine definierte Eingabe ausgegeben werden soll. Somit repräsentiert die Schriftkodierung eine Abbildung des verwendeten Eingabezeichens auf ein hinterlegtes Bildzeichen in der `.dvi`-Datei. Entsprechend der Eingabekodierung sind dies bei T_EX eine Folge von 8-Bit-Zeichen, die wiederum $2^8 = 256$ verschiedene Schriftzeichen bereitstellen [6, Seite 11].

Meyer beschreibt in [5] die Verarbeitung verschiedener Schriftzeichensysteme. Dies kann bspw. die Vereinigung lateinischer, kyrillischer oder klingonischer³ Zeichen in einem gemeinsamen Dokument sein. Da es sich bei T_EX sowohl bei der Eingabe- als auch Ausgabekodierung um ein 8-Bit-System handelt, sind Schriftzeichen in einer normalen T_EX-Schriftdatei mit einem »Namen« versehen, der einer bestimmten Kennzahl zwischen 0 und 255 zugeordnet ist. Ein gewöhnlicher T_EX-Font enthält dementsprechend maximal 256 verschiedene Zeichen [5, 6]. Durch dieses Wissen ist es naheliegend, dass in bestimmten Dokumenten die Verwendung eines Font-Satzes wie z. B. T1 alleine nicht ausreicht. Der Aufruf von `\usepackage[<Optionen>]{fontenc}` unterstützt grundsätzlich die Verwendung verschiedener Schriftsätze in einem gemeinsamen Dokument; die verschiedenen Schriftkodierungen weisen jedoch signifikante Unterschiede untereinander auf und es obliegt dem Autoren, dafür Sorge zu tragen, dass der »richtige« Schriftzeichensatz verwendet wird.⁴ Des Weiteren ist grundsätzlich das Paket `\usepackage[russian,ngerman]{babel}` einzubinden, sofern nicht anders angegeben.

Verwendung der Schriftcodierung OT2

Durch Einbinden des Pakets `babel` ist es möglich, in einem gemeinsamen Dokument unterschiedliche Sprachen (i. e. Schriftzeichen unterschiedlicher Schriftsätze) zu verwenden. Im ersten Moment ist es naheliegend, die von Donald Knuth für den Satz kyrillischer Zeichen entwickelte OT2-Schriftkodierung zu verwenden. Dies führt unter Eingabe des folgenden Quellcodes zu dem in Abbildung ?? gezeigten Ergebnis.⁵

³ Das Paket `piq` verwendet bspw. Okuda-Fonts, die in der METAFONT Datei `piq.mf` hinterlegt sind, zum Satz klingonischer Sprache. Die klingonische Grammatik obliegt dem Autor!

⁴ Diese Unterschiede werden im Folgenden durch die Darstellung des kyrillischen Zeichens Ъ aufgezeigt. IPA-Aussprache nach [1, Seite 9]: <öj >.

⁵ Es handelt sich bei dem Beispielbuchstaben um ein kyrillisches Zeichen, das zum serbischen und serbokroatischen Alphabet gehört; in anderen slawischen Sprachen, wie dem Russischen, wird es jedoch aktuell nicht verwendet.

```

\documentclass{scrartcl}           % KOMA-Artikel als Testumgebung
\usepackage[ansinew]{inputenc}    % Input-Encoding fuer MS-Windows
\usepackage[OT2,T1]{fontenc}      % Beachte: 'OT2' wird vor 'T1' geladen
\usepackage[russian,ngerman]{babel} % Beachte: 'russian' vor 'ngerman'
\begin{document}
Hallo Welt. \foreignlanguage{russian}{Dj}. % Wechselt die Sprache
Hallo Welt. {\cyr Dj}.              % Sprachwechsel im Fliesstext
\end{document}

```



Abbildung 1: Das kyrillische Zeichen д als Bitmap.

Komfortabel bei dieser Lösung ist, dass kyrillische Zeichen entsprechend des kyrillischen Alphabets, wie es beispielsweise in [9, Seite 4] gegeben ist, lateinisch eingegeben werden können, ohne dass der Autor die binäre (hexadezimale) Kodierung einzelner Zeichen kennen muss. Es ist möglich, einzelne kyrillische Buchstaben oder ganze kyrillische Wörter im Fließtext zu nutzen, deren Nomenklatur an das kyrillische Alphabet angepasst ist. Nachteilig erkennt man deutlich das Bitmap-Outline anstatt der – gewohnten Type-1 – Vektorschrift des Schriftzeichens, wie es beim Satz lateinischer Dokumente üblich ist [2]. Insofern ist die oben vorgeschlagene Möglichkeit, kyrillische Zeichen in ein lateinisches Dokument einzubinden, zwar verhältnismäßig bequem, aber optisch nicht akzeptabel.

Verwendung der Schriftcodierung T1

Es ist möglich, kyrillische Schriftzeichen zu verwenden, ohne die Option `russian` an das Paket `babel` zu übergeben. Hierfür muss jedoch etwas Vorarbeit geleistet werden: Die Datei `cyracc.def` muss eingebunden werden, die bspw. von [4] bezogen werden kann, falls nicht ohnehin bereits die $\mathcal{A}\mathcal{M}\mathcal{S}$ -Fonts auf der entsprechenden Distribution installiert sind. Zudem muss über das Makro `\font\tencyr=wn Cyr10` die zu verwendende Schrift spezifiziert werden, in diesem Fall `wn Cyr10`, also Washington Cyrillic. Dies führt unter Eingabe des folgenden Quellcodes zu dem in Abbildung 2 gezeigten Ergebnis.

```

\documentclass{scrartcl}           % KOMA-Artikel als Testumgebung
\usepackage[ansinew]{inputenc}   % Input-Encoding fuer MS-Windows
\usepackage[T1]{fontenc}         % Beachte: nur T1 laden
\usepackage[ngerman]{babel}     % Beachte: kein russian laden
\input{cy racc.def}              % Lade Schriftdefinitionen. Schlecht!
\font\tencyr=wncyr10             % Setze Font auf wncyr10
\def\cyr{\tencyr\cyracc}        % Definiere neue Umgebung
\begin{document}
Hallo Welt. {\cyr Dj}.           % Wechselt die Sprache
Hallo Welt. {\cyr\char"42}.      % Vergleiche z. B. mit 'X2'
\end{document}

```



Abbildung 2: Das kyrillische Zeichen đ als Vektorzeichen, jedoch mit falschen Bogen und Armen [8].

Die Ausgabe erscheint im ersten Moment gut. Jedoch bringt die Verwendung dieses Quellcodes auch einige stilistische Unschönheiten mit sich, sodass der Einsatz dieser Lösung nicht empfohlen wird [3]. In der Datei `cy racc.def` sind bei weitem nicht alle kyrillischen Zeichen definiert, sondern nur einige wenige. Dies hat zur Folge, dass unter Umständen genau ein solches Zeichen benötigt wird, das nicht definiert ist. Ein weiterer Nachteil der oben genannten Lösung ist, dass die Washington Cyrillic Schrift nur zum Setzen russischer Referenzen entwickelt wurde. Bestimmte Eigenschaften sind folglich nicht implementiert, beziehungsweise funktionieren nicht richtig. Hierzu zählen z. B. die Darstellung korrekter Ligaturen, und bei bestimmten Zeichen sind auch die Kapitälchen mangelhaft. Dem Leser wird insofern von der Verwendung dieser Lösung abgeraten.

Verwendung der Schriftcodierung X2

Es bietet sich an, nach alternativen Schriftkodierungen zu suchen, in denen der vollständige kyrillische Zeichenvorrat bereitgestellt und der korrekte Satz der Zeichen berücksichtigt wird. Ein populäres Beispiel ist die X2-Schriftkodierung, die neben dem vollständigen russischen Alphabet auch noch Zeichen weiterer sla-

wischer Sprachen bereitstellt. Dies führt unter Eingabe des folgenden Quellcodes zu dem in Abbildung 3 gezeigten Ergebnis.

```
\documentclass{scrartcl} % KOMA-Artikel als Testumgebung
\usepackage[ansinew]{inputenc} % Input-Encoding fuer MS-Windows
\usepackage[X2,T1]{fontenc} % Beachte: X2 vor T1 laden
\usepackage[russian,ngerman]{babel} % Beachte: russian vor ngerman
\begin{document}
Hallo Welt. \foreignlanguage{russian}{\char"42}. % Wechselt die Sprache
Hallo Welt. {\cyr\char"42}. % Sprachwechsel im Fliesstext
\end{document}
```



Abbildung 3: Das kyrillische Zeichen Ѣ als Vektorzeichen und typografisch korrekt gesetzt.

Bei dieser Lösung erkennt man, dass das entsprechende kyrillische Zeichen auf Grund der Verwendung des Schriftsatzes X2 in einer Vektorschrift gesetzt wird. Ebenso fällt auf, dass die Bögen, Endstriche und Arme der Zeichen anders (richtig) gesetzt werden, im Gegensatz zur Washington Cyrillic Schrift.

Bei der Verwendung dieses Schriftsatzes ist es nicht mehr möglich, durch die Eingabe lateinischer Zeichen im Editor den vollständigen Zeichensatz wiederzugeben. Es muss ein kryptischer Aufruf erfolgen (vgl. Code-Zeilen 6 und 7 des Minimalbeispiels). Der Autor muss Kenntnis über die Tabelle der Schriftkodierung besitzen, um das gewünschte kyrillische Zeichen setzen zu können. Eine vollständige Kodierungstabelle findet man beispielsweise in [7, X2 speziell auf Seite 33].

Unter Anwendung des obigen Minimalbeispiels erhält man das optisch beste und typografisch korrekte Ergebnis zum Satz kyrillischer Schriftzeichen. Den unumgänglichen Nachteil, Kenntnis über die Tabelle der Schriftkodierung besitzen zu müssen, muss der Autor an dieser Stelle in Kauf nehmen. Um allerdings die meist mühsame Suche nach den X2 Zeichenkodierungen zu ersparen, ist diese auf Seite 28 tabellarisch angegeben. Als Quelle dient [7, Seite 33].

Zusammenfassung und abschließende Betrachtungen

Abbildung 1 zeigt die gewonnenen Ergebnisse in einer direkten Gegenüberstellung. Man erkennt den signifikanten Nachteil der OT2-Schriftkodierung, da die Zeichen als Bitmap repräsentiert werden (Abbildung 4a). Die Verwendung der T1-Schriftkodierung stellt die Zeichen zwar vektoriell dar, jedoch sind diese – neben weiteren Nachteilen – typografisch nicht korrekt gesetzt (diesen Sachverhalt verdeutlicht die Abbildung 4b). Lediglich unter Einbinden der Schriftkodierung X2 werden die kyrillischen Schriftzeichen sowohl vektoriell als auch typografisch korrekt gesetzt (siehe Abbildung 4c).

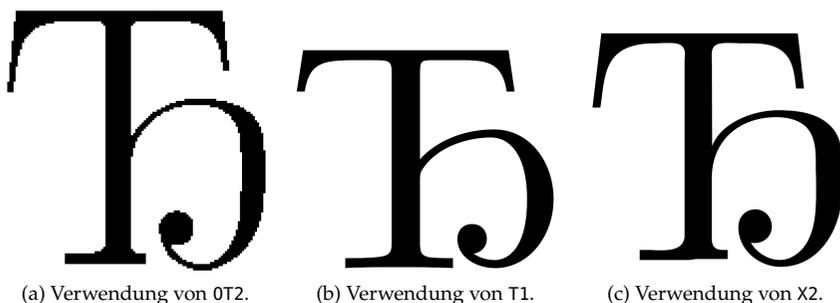


Abbildung 4: Die Ergebnisse dieses Beitrags im Überblick. Es werden zum Einbinden kyrillischer Schriftzeichen in ein lateinisches Dokument die OT2- (in 4a), die T1- (in 4b) und die X2-Schriftkodierung (in 4c) genutzt.

Es wurde bereits erwähnt, dass die X2-Schriftkodierung den vermeintlichen Nachteil mit sich bringt, dass ein gewünschtes Zeichen über dessen oktale oder hexadezimale Kodierung angesprochen werden muss. Daher dient die nachfolgende Tabelle aus [7, Seite 33] als Nachschlagewerk, um die Zeichenkodierungen des Pakets X2 zu entnehmen. In L^AT_EX müssen dann die Zeichen über das Makro `\char'102` (wenn die oktale Darstellung gewählt wird) oder über das Makro `\char"42` (wenn die hexadezimale Darstellung gewählt wird) angesprochen werden. Die hier angegebenen Werte '102 bzw. "42 sind exemplarisch für den Aufruf des Zeichens Т angegeben.

Tabelle 1: Die Tabelle kyrillischer Zeichen.

rxrm1000, X2	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	˘	˙	ˆ	˜	¨	˘	˚	ˇ	~0x
'01x	˘	˘	˘	˘	˘	I	<	>	
'02x	“	”	ˆ	˘	˘	—	—		~1x
'03x	o	˘	˘	˘	δ	δ	И	И	
'04x	˘	!	"	#	\$	%	&	'	~2x
'05x	()	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	~3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	Æ	Ђ	Ђ	€	€	К	К	~4x
'11x	Д	И	Ј	Љ	М	Н	О	П	
'12x	Р	Q	Т	С	Ц	Ц	Ч	W	~5x
'13x	Ђ	Ђ	V		\		ˆ	—	
'14x	‘	æ	ђ	ђ	€	€	к	к	~6x
'15x	д	i	j	љ	м	н	о	п	
'16x	Р	q	т	с	ц	ц	ч	w	~7x
'17x	Ђ	Ђ	v	{		}	˘	-	
'20x	Г	Г	Г	Ђ	ђ	Ж	З	З	~8x
'21x	Ї	К	К	К	Ј	Н	Н	Н	
'22x	Ө	Ç	Ÿ	Y	Y	Х	Х	Ч	~9x
'23x	Ч	€	Ә	ε	Ě	№	□	§	
'24x	г	г	г	ђ	ђ	ж	з	з	~Ax
'25x	ï	ç	к	к	ј	н	н	н	
'26x	ө	ç	Ÿ	Y	Y	х	х	ч	~Bx
'27x	ч	€	ә	ε	ě	„	«	»	
'30x	А	В	В	Г	Д	Е	Ж	З	~Cx
'31x	И	Й	К	Л	М	Н	О	П	
'32x	Р	С	Т	У	Ф	Х	Ц	Ч	~Dx
'33x	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
'34x	a	б	в	г	д	е	ж	з	~Ex
'35x	и	й	к	л	м	н	о	п	
'36x	p	c	t	y	ф	x	ц	ч	~Fx
'37x	ш	щ	ъ	ы	ь	э	ю	я	
	˘	˘	˘A	˘B	˘C	˘D	˘E	˘F	

Literatur

- [1] Hanna-Chris Gast: *Kyrillische Schrift für den Computer – Benennung der Buchstaben, Vergleich der Transkriptionen in Bibliotheken und Standesämtern, Auflistung der Unicodes sowie Tastaturbelegung für Windows XP*; Shaker Verlag, Aachen; 2010.
- [2] Gerrit Kirpal: *L^AT_EX und Fonts – Wie funktioniert das?*; Technische Universität Dresden; Juli 2004.
- [3] LinuxSelfHelp.com: *Cyrillic Wordspacing*; <http://www.linuxselfhelp.com/howtos/Cyrillic/Cyrillic-HOWTO-7.html>.
- [4] Vladim Maslov: *Enthalten im L^AM^S Font-Paket*; <http://www.kovrik.com/sib/russify/tex-latex/ams-cyrillic/cyracc.def>.
- [5] Peter Meyer: *Das Problem für mehrsprachige T_EX-Dokumente*; <http://www.peter-meyer.org/edv/latex/latex.html>.
- [6] Frank Mittelbach: *L^AT_EX₂_ε Encoding Interfaces – Purpose, Concepts, and Open Problems*; <http://www.latex-project.org/papers/encoding-concepts.pdf>; 1995.
- [7] Frank Mittelbach et al.: *L^AT_EX font encodings*; <http://mirror.ctan.org/macros/latex/doc/encguide.pdf>; Januar 2006.
- [8] Achim Schaffrinna: *Anatomie der Buchstaben*; »Die T_EXnische Komödie«; 3, S. 11–15; 2010.
- [9] Walter Schmidt: *Kyrillisch für arme Leute – Setzen russischer Textpassagen mit L^AT_EX*; <http://home.vr-web.de/was/x/pmcyr.pdf>; Mai 2005.

Ein passendes Bewerbungsanschreiben zum ModernCV-Lebenslauf

Kurt Lidwin

In verschiedenen Foren kommt immer wieder die Frage auf, wie man ein passendes Bewerbungsanschreiben zum eigenen Lebenslauf erzeugen kann, der mit der Dokumentklasse ModernCV geschrieben wurde.

Dieser Artikel zeigt Ihnen, wie Sie Ihr Bewerbungsanschreiben (mit der Dokumentklasse `scr\l\tr2` geschrieben) an das Layout Ihres Lebenslaufes anpassen können.

Einführung

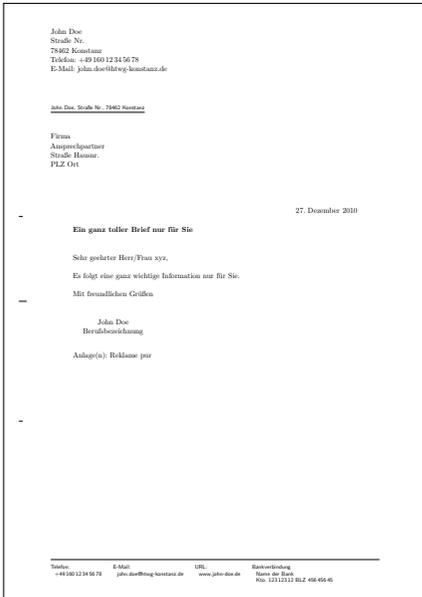
Ein gängiger Standard-Brief – mit der Dokumentklasse `scr\l\tr2` [2, Seiten 173–262, 417–442] geschrieben – ist in Abbildung 1a wiedergegeben. Abbildung 1b zeigt den »Lebenslauf« von JOHN DOE, für den ein passendes Bewerbungsanschreiben verfasst werden soll. In der dazu verwendeten Dokumentklasse ModernCV (*curriculum vitae*) [1] wurde im Beispiel die Grundfarbe Blau eingestellt. Das Bewerbungsanschreiben und der Lebenslauf passen nicht gut zusammen.

Eine kurze Dokumentation der Lebenslauf-Dokumentklasse ist unter [3] zu finden, eine Dokumentation von XAVIER DANAUX, dem Autor der Dokumentklasse `moderncv`, existiert (noch) nicht. Einige Beispiele findet man auf CTAN (<http://mirror.ctan.org/macros/latex/contrib/moderncv/>).

Die notwendigen Anpassungen im Einzelnen

KOMA-Script bietet mit seiner Brief-Dokumentklasse `scr\l\tr2` unter anderem die Möglichkeit, persönliche Daten des Absenders in einer `\lco`-Datei (Letter Class Option, Brief-Klassenoption) zu speichern (Listing 4, Seite 37). Falls Sie einen speziellen Brief-Fuß benötigen, legen Sie diesen ebenfalls als `\lco`-Datei, beispielsweise unter `John-Doe-Fuss.lco`, ab. Je nach Bedarf können die `\lco`-Dateien als Klassenoption oder mit dem Befehl `\LoadLetterOption` geladen werden.

Um einen Brief als passendes Anschreiben verwenden zu können, sind die folgenden Einstellungen sinnvoll, die Sie am besten in einer separaten `\lco`-Datei zusammen fassen. In dieser `\lco`-Datei ersetzen Sie den Befehl `\usepackage` durch `\RequirePackage`, den Befehl `\newcommand*` durch `\providecommand*`



(a) Standard-Brief von John-Doe.



(b) »Lebenslauf« von John-Doe.

Abbildung 1: Standard-Brief und Lebenslauf – passen nicht gut zusammen.

und die Befehle `\newcaptionname` und `\renewcaptionname` durch den Befehl `\providcaptionname`.

Die folgenden Beispiele sind für eine `\co`-Datei gedacht.

Zunächst definieren Sie die im Lebenslauf üblicherweise verwendeten Farben:

```
\definecolor{blau} {rgb}{0.25,0.50,0.75}
\definecolor{hellblau} {rgb}{0.25,0.50,0.75}
\definecolor{orange} {rgb}{0.95,0.55,0.15}
\definecolor{gruen} {rgb}{0.30,0.65,0.15}
\definecolor{rot} {rgb}{0.95,0.20,0.20}
\definecolor{dunkelgrau} {rgb}{0.35,0.35,0.35}
\definecolor{hellgrau} {rgb}{0.45,0.45,0.45}
\definecolor{schwarz} {rgb}{0.00,0.00,0.00}
```

Anschließend definieren Sie in Ihrer `\co`-Datei ein neues Kommando, mit dessen Hilfe Sie Ihr Anschreiben komplett auf die Farbe des Lebenslaufes umstellen können:

```
% Einheitliche Farbe für Anschreiben festlegen
\providecommand*\Dokfarbe}{blau} % Blau ist Standard
```

Im Quelltext Ihres Anschreibens können Sie nun bei Bedarf mit dem Befehl

```
\renewcommand*\Dokfarbe}{orange}
```

Ihr komplettes Anschreiben auf eine andere Farbe, hier Orange, umstellen, wenn Sie diese Umstellung auch im Lebenslauf vornehmen.

Damit das Anschreiben dem Lebenslauf ähnlich sieht, ändern Sie die Farbe der folgenden Symbole (aus dem Paket marvosym):

```
\RequirePackage{marvosym} % Symbole für Telefon, Mobile, Mail, ...
\providecaptionname{ngerman}\emailname{\color{\Dokfarbe}{\Letter}}%
\providecaptionname{ngerman}\phonename{\color{\Dokfarbe}{\Mobilefone}}%
\providecaptionname{ngerman}\faxname{\color{\Dokfarbe}{\Telefon}}%
```

Ist Ihnen aufgefallen, dass in diesem Beispiel `\phonename` für die Handynummer und `\faxname` für die Festnetznummer verwendet wird?

Die im Lebenslauf als Standard verwendete Schrift ist die serifenlose Schrift der Schriftfamilie Latin Modern. Stellen Sie mit den in Listing 1 angegebenen Kommandos in Ihrer `\co`-Datei die Schrift um.

Bitte beachten Sie die Verwendung von `\RequirePackage` in Zeile 1. Die Befehle `\setkomafont` und `\addtokomafont` werden nicht ersetzt.

Listing 1: Definition der Schrift im Anschreiben

```
1 \RequirePackage{lmodern} % Brottschrift Latin Modern
2
3 % Definition der Schriftart für Brottschrift: sans serif lmodern
4 \renewcommand\familydefault{\sfdefault} % auskommentieren für roman!
5
6 % Absender:
7 % ... im Brieffenster:
8 |\setkomafont{backaddress}{\normalsize\rmfamily\mdseries\slshape}
9 \setkomafont{fromaddress}{\normalsize\rmfamily\mdseries\slshape} % Adresse
10 \addtokomafont{fromaddress}{\color{\Dokfarbe}}
11 |\setkomafont{fromname}{\normalsize\rmfamily\mdseries\slshape} % Name
12 |\addtokomafont{fromname}{\color{\Dokfarbe}}
13
14 % Empfänger:
15 |\setkomafont{toaddress}{\normalsize\rmfamily\mdseries\slshape} % Adresse
16 |\setkomafont{toname}{\normalsize\rmfamily\mdseries\slshape} % Name
17
18 \setkomafont{subject}{\normalsize\bfseries\color{\Dokfarbe}} % Betreff
19 |\setkomafont{title}{\normalsize\rmfamily\mdseries\slshape} % Titel
```

Verwenden Sie im Lebenslauf die Serifenschrift, so ist Zeile 4 im Quelltext 1 zu kommentieren. Nun passen Sie noch die persönlichen Textelemente an den Lebenslauf an, wie in Listing 2 gezeigt.

Listing 2: Definition der persönlichen Textelemente im Anschreiben

```

1 % Name in gleicher Größe wie im Lebenslauf
2 \newcommand*\HUGE{\fontsize{34}{36}\sffamily\mdseries\upshape} %
3 % Grußformel, Signatur und Anlagen linksbündig (ansonsten: eingerückt).
4 \renewcommand*\raggedsignature{\raggedright}
5
6 %\setkomavar{fromname}{John-Doe} % Name, mit Titel: {Prof.\,Dr.\,John-Doe}
7 \setkomavar{fromaddress}{Strasse-Nr.\ 78462-Konstanz} % Ihre Adresse
8 \setkomavar{fromemail}{john.doe@htwg-konstanz.de} % E-Mail Adresse
9 \setkomavar{fromphone}{+49\,160\,12\,34\,56\,78} % Mobiltelefonnummer
10 \setkomavar{fromfax}{+49\,7531\,12\,34\,56\,78} % Festnetznummer
11 \setkomavar{signature}{John-Doe\ \LaTeX{}-Trainer} % Unterschrift einbinden
12 \setkomavar{phoneseparator}{-} % Leerzeichen nach "Telefon"
13 \setkomavar{emailseparator}{-} % Leerzeichen nach "EMail"
14 \setkomavar{faxseparator}{-} % Leerzeichen nach "Mobile"
15 \setkomavar{fromlogo}{\HUGE John-Doe}} % Links: Name groß, schwarz
16 \setkomavar{subject}{Bewerbung als Trauarbeitnehmer} % Betreffzeile

```

Es empfiehlt sich, die Zeile 16 aus Listing 2 in das eigene Anschreiben zu übernehmen, da sich der Betreff des Briefes für jeden Brief ändert. Der Rest ist am besten in einer `\lco`-Datei aufgehoben. Es kann sein, dass Ihr Name im Lebenslauf zu lang ist, um ohne Umbruch dargestellt werden zu können. Fügen Sie dann in Ihren Lebenslauf in der Präambel die Zeile

```
\renewcommand*\firstnamefont{\fontsize{34}{36}\sffamily\mdseries\upshape}
```

ein, und variieren Sie die beiden Zahlen 34 (Schriftgröße) und 36 (Zeilenabstand) so lange, bis Ihr Name ohne Umbruch wiedergegeben wird. Ändern Sie anschließend Zeile 2 in Listing 2 entsprechend ab.

Realisierung des Anschreibens als Brief mit zwei `\lco`-Dateien

Um ein Bewerbungsanschreiben zu bekommen, das problemlos auf einem eigenen Standard-Brief aufsetzt, werden alle notwendigen Anpassungen in eine separate Datei namens `John-Doe-Anschreiben.lco` geschrieben. Die Funktion von `\lco`-Dateien ist in [2, Seite 251–257] genauer beschrieben. Somit besteht das komplette Anschreiben aus den folgenden Dateien:

- `John-Doe-\lco-Anschreiben.tex` (Listing 3), dem eigentlichen Anschreiben,
- `John-Doe.lco` (Listing 4), den persönlichen Daten von JOHN DOE und
- `John-Doe-Anschreiben.lco` (Listing 5) mit den Anpassungen des Anschreibens an den Lebenslauf.

Das Ergebnis des angepassten Anschreibens finden Sie in Abbildung 2 auf Seite 35, zum Vergleich den Lebenslauf noch einmal in Abbildung 3 auf Seite 36.

Die Datei `John-Doe-lco-Anschreiben.tex`

Die Datei `John-Doe-lco-Anschreiben.tex` in Listing 3 enthält das eigentliche Anschreiben, in dem in Zeile 10 die Datei `John-Doe.lco` und in Zeile 25 die Datei `John-Doe-Anschreiben.lco` aufgerufen werden. Beachten Sie bitte, dass Sie gegebenenfalls die richtige Reihenfolge der `lco`-Dateien einhalten müssen. Für das Bewerbungsanschreiben gilt: zuerst die Datei `John-Doe.lco`, dann `John-Doe-Anschreiben.lco` aufrufen.

Listing 3: Datei `John-Doe-lco-Anschreiben.tex`

```

1 %==== Copyright 2010-2011 Kurt Lidwin =====
2 %==== Datei John-Doe-lco-Anschreiben.tex; Projekt: LaTeX-Stammtisch =====
3 % Benötigte Dateien: John-Doe.lco, John-Doe-Anschreiben.lco
4 \documentclass[%
5   draft=false           % Fertiges Dokument; =true Entwurfsstadium
6   ,ngerman              % Neue deutsche Rechtschreibung
7   ,fontsize=11pt       % Groesse der Brotschrift
8   ,parskip=half*       % Absatzabstand, Einrueckung (Buch Seite 91-93)
9   ,version=last        % Nutze neueste KOMA-Klasse
10  ,John-Doe             % lädt Datei "John-Doe.lco" mit persönlichen Daten
11                      % des Brief-Absenders
12 ]{scrlltr2}
13
14 \usepackage[T1]{fontenc} % "Saubere" Schriften in der PDF
15 \usepackage[utf8]{inputenc} % Deutsche Umlaute direkt eingeben
16 \usepackage[ngerman]{babel} % Neue deutsche Rechtschreibung
17
18 %\usepackage{xcolor} % in John-Doe-Anschreiben.lco geladen
19 %\usepackage{lmodern} % in John-Doe-Anschreiben.lco geladen
20 %\usepackage{marvosym} % in John-Doe-Anschreiben.lco geladen
21 \usepackage{microtype} % Optischer Randausgleich
22
23
24 %\LoadLetterOption{John-Doe-Fuss} % Fußzeile erstes Blatt
25 \LoadLetterOption{John-Doe-Anschreiben} % Anschreiben für ModernCV
26
27 % blau, hellblau, orange, gruen, rot, dunkelgrau, hellgrau, schwarz
28 \renewcommand*{\Dokfarbe}{blau} % blau (Standard)
29
30 % Betreffzeile, für jeden Brief unterschiedlich.
31 \setkomavar{subject}{Bewerbung für eine tolle Stelle}
32
33 %==== Beginn des eigentlichen Anschreibens =====
34 \begin{document}
35 \begin{letter}{Firma \ \ Ansprechpartner \ \ Straße Hausnr. \ \ PLZ Ort}

```

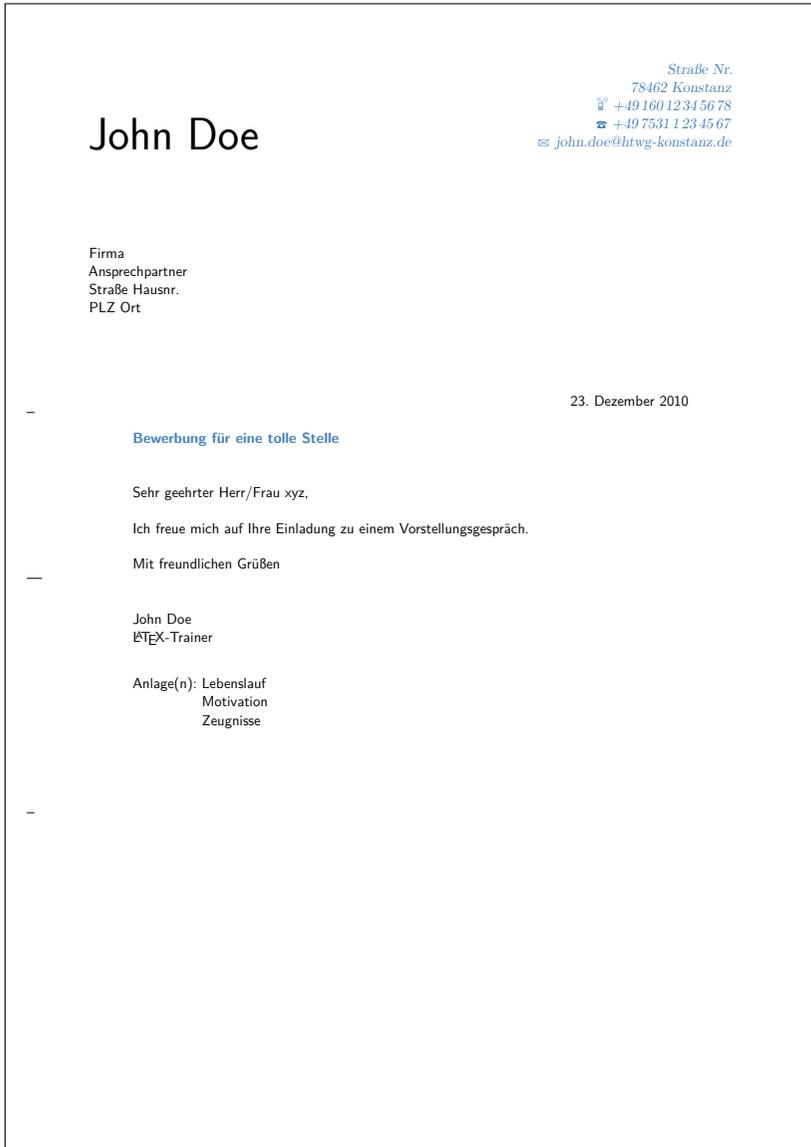


Abbildung 2: Das passende Anschreiben von John Doe.

John Doe

Lebenslauf

Straße Nr.
78462 Konstanz

☎ +49 160 12 34 56 78

☎ +49 7531 1 23 45 67

✉ john.doe@htwg-konstanz.de



Persönliche Daten

Geburtsdatum TT. Monat JJJJ

Geburtsort Geburtsort, Geburtsland

Studium

seit 2010 **Master of Bezeichnung (ausstehend)**, Hochschule Ort, Note (derzeit) Zeugnisnote.
Studiengang Bezeichnung.
Abschluss voraussichtlich Monat JJJJ.

2006–2010 **Bachelor of Bezeichnung**, Hochschule Ort, Note Zeugnisnote.
Studiengang Bezeichnung.
Abschlussarbeit: *Titel der Abschlussarbeit*, Note Zeugnisnote.

2009–2010 **Vertiefungsstudium Bezeichnung**, Hochschule Ort, Note Zeugnisnote.

Erfahrungen

Studium

seit 2010 **Hilfswissenschaftler**, Rechenzentrum Hochschule Ort, Betreuung der Kurse \LaTeX für das Studium.

seit 2010 **Masterprojekt (ausstehend)**, Hochschule Ort, Fachbereich Bezeichnung, *Titel des Masterprojekts*.

2008–2009 **Praktisches Studiensemester**, Firmenname, Sitz der Firma.
Aufgabe im Praktikum.
Projekterfahrung im Ausland:

- Ort1, Land1. Aufenthaltsdauer Zeitangabe.
- Ort2, Land2. Aufenthaltsdauer Zeitangabe.
- Ort3, Land3. Aufenthaltsdauer Zeitangabe.

1/2

Abbildung 3: Der »Lebenslauf« von John Doe.

```

36
37 \opening{Sehr geehrte/r Herr/Frau xyz,}
38
39 Ich freue mich auf Ihre Einladung zu einem Vorstellungsgespräch.
40
41 \closing{Mit freundlichen Grüßen}
42 \encl{Lebenslauf\\Motivation\\Zeugnisse}
43
44 \end{letter}
45 \end{document}

```

Die Datei John-Doe.lco

Die Datei John-Doe.lco enthält die persönlichen Angaben von JOHN DOE und wird in *jedem* Brief von JOHN DOE eingesetzt. Sie ist in Listing 4 wiedergegeben.

Listing 4: Datei John-Doe.lco

```

1 %===== Copyright 2010-2011 Kurt Lidwin =====
2 %===== Datei John-Doe.lco; Projekt: LaTeX-Stammtisch =====
3 % Persönliche Daten und Einstellungen
4 \ProvidesFile{John-Doe.lco}[%
5   2010/12/20 v0.2 scrllttr2 Brief-Klasse-Option kl]
6
7 \KOMAOPTIONS{%
8   fromalign=right      % Kopfzeile Ausrichtung rechts, links, zentriert
9   ,fromrule=false     % Linie unter Kopfzeile
10  ,fromlogo=true       % Logo in Kopfzeile
11  ,addrfield=true      % Adressatenfeld
12  ,backaddress=false   % Absender in Adressatenfeld
13  ,pagenumber=botright % Seitennummer
14  ,foldmarks=BLMT     % Faltmarken abschalten, ausser Lochermarke
15  ,fromphone=true      % Absender Telefonnummer
16  ,fromemail=true      % Absender E-Mail Adresse
17  ,fromfax=true        % Absender Mobiltelefonnummer
18  ,subject=untitled    % "Betreff:" wird nicht gesetzt.
19 }
20
21 \setkomavar{phoneseparator}{~} % Leerzeichen nach "Telefon"
22 \setkomavar{faxseparator}{~}  % Leerzeichen nach "Mobile"
23 \setkomavar{emailseparator}{~} % Leerzeichen nach "EMail"
24
25 \setkomavar{fromname}{John-Doe} % Name, mit Titel: {Prof.\,Dr.\,John-Doe}
26 \setkomavar{fromaddress}{Straße-Nr.\ 78462-Konstanz} % Adresse
27 \setkomavar{fromphone}{+49\,160\,12\,34\,56\,78} % Mobil- statt Festnetznr.
28 \setkomavar{fromfax}{+49\,7531\,1\,23\,45\,67} % Festnetz- statt Faxnummer
29 \setkomavar{fromemail}{john.doe@htwg-konstanz.de} % E-Mail Adresse
30 \setkomavar{signature}{John-Doe\\ LaTeX}-Trainer} % Unterschrift einbinden
31
32 \setkomavar{frombank}{Name der Bank\

```

```

33 | Kto. 123\,123\,12 BLZ 456\,456\,45}
34 | \setkomavar{fromurl}{www.john-doe.de} % Homepage von John Doe
35 |
36 | \setkomavar{subject} {!Es fehlt noch der Betreff!} % Falls Betreff vergessen
37 |
38 | \endinput % Ende Datei John-Doe.lco

```

Einen kleinen Trick finden Sie in Zeile 36. So wird verhindert, dass Sie den Betreff Ihres Briefes vergessen können.

Die Datei John-Doe-Anschreiben.lco

Die Datei John-Doe-Anschreiben.lco (Listing 5) enthält die Anpassungen, die zuvor beschrieben worden sind. Durch die Trennung in zwei lco-Dateien bleibt die ursprüngliche Funktion der Datei John-Doe.lco erhalten.

Listing 5: Datei John-Doe-Anschreiben.lco

```

1 | %==== Copyright 2010-2011 Kurt Lidwin          =====
2 | %==== Datei John-Doe-Anschreiben.lco; Projekt: LaTeX-Stammtisch =====
3 | \ProvidesFile{John-Doe-Anschreiben.lco}[%
4 |   2010/12/15 v0.1 scr\ltxr2 Brief-Klasse-Option kl]
5 |
6 | %==== Farbdefinitionen nach moderncvthemeclassic =====
7 | \RequirePackage{xcolor} % für Farbdefinitionen
8 | \definecolor{blau}      {rgb}{0.25,0.50,0.75}
9 | \definecolor{hellblau} {rgb}{0.25,0.50,0.75}
10 | \definecolor{orange}   {rgb}{0.95,0.55,0.15}
11 | \definecolor{gruen}    {rgb}{0.30,0.65,0.15}
12 | \definecolor{rot}      {rgb}{0.95,0.20,0.20}
13 | \definecolor{dunkelgrau}{rgb}{0.35,0.35,0.35}
14 | \definecolor{hellgrau} {rgb}{0.45,0.45,0.45}
15 | \definecolor{schwarz}  {rgb}{0.00,0.00,0.00}
16 |
17 | % Einheitliche Farbe für Anschreiben festlegen
18 | \providecommand*{\Dokfarbe}{blau}
19 |
20 | %==== Fonts (in Anlehnung an moderncv) Buch KOMA-Script S.71 =====
21 | \RequirePackage{lmodern} % Latin Modern als Brotschrift
22 | % Brotschrift sans serif der lmodern
23 | \renewcommand*{\familydefault}{\sfdefault}
24 |
25 | % Schriftart des Absenders (Brieffenster)
26 | |\setkomafont{backaddress}{\normalsize\rmfamily\mdseries\slshape}
27 | % Schriftart und -farbe des Absenders
28 | \setkomafont{fromaddress}{\normalsize\rmfamily\mdseries\slshape}
29 | \addtokomafont{fromaddress}{\color{\Dokfarbe}}
30 | % Schriftart des Absendernamens
31 | |\setkomafont{fromname}{\normalsize\rmfamily\mdseries\slshape}
32 | |\addtokomafont{fromname}{\color{\Dokfarbe}}

```

```

33 % Schriftart und -farbe der Betreffzeile
34 \setkomafont{subject}{\normalsize\bfseries\color{\Dokfarbe}}
35 % Schriftart des Titels
36 %\setkomafont{title}{\normalsize\rmfamily\mdseries\slshape}
37 % Schriftart der Empfaenderadresse
38 %\setkomafont{toaddress}{\normalsize\rmfamily\mdseries\slshape}
39 % Schriftart des Empfaengernamens
40 %\setkomafont{toname}{\normalsize\rmfamily\mdseries\slshape}
41
42 % Name in gleicher Größe wie im Lebenslauf
43 \providecommand*{\HUGE}{\fontsize{34}{36}\sffamily\mdseries\upshape}
44 \setkomavar{fromname}{~} % Name nicht hier anzeigen, sondern hier:
45 \setkomavar{fromlogo}{\HUGE John-Doe} % Links: Name groß, schwarz gesetzt
46
47 % Absenderadresse
48 \setkomavar{fromaddress}{Straße-Nr.\ 78462-Konstanz}
49
50 %==== Styleanpassung an ModernCV ====
51 \RequirePackage{marvosym} % Symbole für Telefon, Mobile, Mail,...
52 \providecaptionname{ngerman}\emailname{\color{\Dokfarbe}{\Letter}}%
53 \providecaptionname{ngerman}\phonename{\color{\Dokfarbe}{\Mobilefone}}%
54 \providecaptionname{ngerman}\faxname{\color{\Dokfarbe}{\Telefon}}%
55
56 % Grußformel, Signatur und Anlagen linksbündig (ansonsten: eingerückt).
57 \renewcommand*{\raggedsignature}{\raggedright}
58
59 \endinput % Ende Datei John-Doe-Anschreiben.lco

```

Den vollständigen Quelltext für das Anschreiben und den Lebenslauf können Sie unter der URL <http://projekte.dante.de/Stammtische/Konstanz> herunterladen.

Zusammenfassung und Ausblick

Es sind nur wenige Änderungen notwendig, um aus einem Standard-Brief ein zu einem Lebenslauf passendes Anschreiben zu erhalten, wie die Abbildung 2 (Seite 35) und Abbildung 3 (Seite 36) zeigen.

Seit Oktober 2010 gibt es das Paket `koma-moderncvc` classic [4], das die Befehle der Klasse `moderncv` nur im Classic-Stil, vorzugsweise mit der Klasse `scrartcl`, zur Verfügung stellt.

Eine andere Möglichkeit wäre, das Anschreiben unabhängig vom DIN-Brief anzufertigen, da es normalerweise in die Bewerbungsmappe eingelegt oder festgeklemmt wird und somit kein Fenster im Briefumschlag berücksichtigt werden muss. In [2, Seiten 496–498] gibt es hierzu ein paar Hinweise.

Schade ist, dass der Rahmen um das Foto nicht mittels Klassen- oder Befehlsoption, beispielsweise `noframe`, abgeschaltet werden kann. Es ist wünschenswert, dass die Klasse `moderncv` entsprechend erweitert wird.¹

Literatur

- [1] Xavier Danaux: *moderncv – a modern curriculum vitae class*; 2007.
- [2] Markus Kohm und Jens-Uwe Morawski: *KOMA-Script: Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε*; Lehmanns Media Verlag; Berlin; 3. Aufl.; 2008.
- [3] Frank Lanitz: *L^AT_EX und der Lebenslauf mit moderncv*; 2010.
- [4] Salome Södergran: *KOMA-ModerncvClassic*; 2010.

Einlesen und Ausführen von Quellcode

Herbert Voß

In den diversen Diskussionsforen oder Mailinglisten tritt immer wieder die Frage auf, ob man bestimmte Bereiche eines Quellcodes nicht nur als Code sondern auch als Endergebnis der Anwendung darstellen kann. Sowohl das Paket `fancyvrb` als auch `listings` unterstützen ein externes Schreiben und partielles Einlesen von Quellcode beliebigen Typs. Weitere Pakete, wie beispielsweise `showexpl`, erlauben das Ausführen von Teilen eines L^AT_EX-Codes. In diesem Artikel wird gezeigt, dass man eine entsprechende Vorgehensweise auch auf beliebige Typen von Quellcode anwenden kann.

Einführung

Bei der Erstellung eines Manuskripts für einen Artikel oder ein Buch wird je nach Fachrichtung der Text durch Beispiele ergänzt, die sich oft auf eine von einem Programm erzeugte Ausgabe beziehen. Dabei kann es vorteilhaft sein, wenn man den Quellcode für diese Programme aus dem Dokument heraus kontrolliert, um auf diese Weise auch sicher zu gehen, dass jede Änderung am Quelltext auch definitiv eine Auswirkung auf die erzeugte Abbildung oder den erzeugten Text

¹ Hinweis für eine sofortige Abschaltung als »Notmaßnahme«, `quick and dirty`: In Zeile 161 der Datei `moderncvthemecasual.sty` den Befehl `\framebox` entfernen.

hat, der automatisch in das Manuskript eingebunden wird. Dadurch können insbesondere bei längeren Manuskripten Fehler vermieden werden, denn der Quellcode korrespondiert immer zur gezeigten Ausgabe.

Einfache L^AT_EX-Sequenzen

Bereiche eines Quellcodes

Insbesondere für L^AT_EX-Beispiele ist oftmals nur der Quellcode zwischen den Befehlen `\begin{document}` und `\end{document}` von Interesse. Sowohl `fancyvrb` als auch `listings` unterstützen die Angabe eines Bereiches, der durch Zeilennummern anzugegeben ist. Dies erfordert jedoch eine Anpassung, falls einzelne Zeilen im Quellcode hinzugefügt oder gelöscht werden. Die Angabe jeweils einer Zeichenkette für den Beginn und das Ende des auszugebenden Bereiches ist daher sinnvoller. Für das Paket `listings` existiert dafür die Option `linerange`, wobei sich die Angaben für das Intervall prinzipiell nicht von der Angabe der Zeilennummern unterscheiden; lediglich Sonderzeichen sind durch einen vorangestellten Backslash zu maskieren: `\\begin\{document\}`. Mit der Angabe `includerangemarker=false` wird die Ausgabe der den Bereich kennzeichnenden Zeichenketten unterbunden; sonst würden auch `\begin{document}` und `\end{document}` in der Ausgabe erscheinen.

```
\lstinputlisting[linerange=\\begin\{document\}-\\end\{document\},
    includerangemarker=false]%
{voss/demo.tex}
```

Die Ausgabe der obigen Befehlszeile liefert dann den folgenden Quelltext eines L^AT_EX-Dokuments, welches hier als Beispiel dienen soll.

```
\begin{tabular}{@{} m{0.5\linewidth}@{}
    >\lstinputlisting[includerangemarker=false,
        rangeprefix=\%,
        linerange=START-STOP]%
    {\jobname.tmp}}m{0.5\linewidth} @{}

\begin{Example}
\pspicture(3,2)
%START
\psframe*[linecolor=blue!30](3,2)
%STOP
\endpspicture
\end{Example}
& \tabularnewline

\begin{Example}
\pspicture(3,2)
%START
\psframe*[linecolor=red!30](3,2)
```

```

\endpicture
%STOP
\end{Example}
& \tabularnewline
\end{tabular}

```

Gleiches kann mit dem Paket `fancyvrb` erreicht werden. Der Bereich ist hier durch die beiden Optionen `firstline` und `lastline` festzulegen. Folgendes kleine Beispiel gibt seinen eigenen Textkörper aus:

```

\documentclass{article}
\usepackage{fancyvrb}
\begin{document}
\VerbatimInput[
  frame=single,
  fontsize=\small,
  firstline=\string\begin{document},
  lastline=\string\end{document},
]{\jobname.tex}
\end{document}

```

Die Optionen `firstline` und `lastline` definieren intern die beiden Makros `\FancyVerbStartString` und `\FancyVerbStopString`. In besonderen Fällen können diese auch direkt manipuliert werden, wobei die Makrodefinition auch führende Leerzeichen enthalten muss, falls diese im Quellcode vorhanden sind. Da die Makros noch nicht existieren, müssen sie beim ersten Mal mit `\newcommand` definiert werden, beziehungsweise mit `\edef`, falls man, wie in diesem Fall, \TeX -spezifische Sonderzeichen verwendet. Das folgende Beispiel gibt die Präambel des Dokumentes aus. Für reine Demonstrationszwecke befinden sich im Quelldokument zwei Leerzeichen vor `\begin{document}`, die hier ebenfalls durch `\space` zu beachten sind:

```

\edef\FancyVerbStartString{\string\documentclass{article}}
\edef\FancyVerbStopString{\space\space\string\begin{document}}% 2 Leerzeichen
\VerbatimInput[frame=single,fontsize=\small]{voss/demo.tex}

```

```

\makeatletter
\let\pc\@percentchar
\makeatother
\usepackage{pstricks,fancyvrb,array,listings}
\lstset{basicstyle=\ttfamily\small}

\def\endExample{\end{VerbatimOut}}
\def\START{}\def\STOP{}\input{\jobname.tmp}}
\newcommand\Example{%

```

```
\VerbatimEnvironment
\begin{VerbatimOut}{\jobname.tmp}}
```

Quellcode und Ausgabe

Ein weiterer oft nachgefragter Anwendungsfall ist die gleichzeitige Angabe von Quellcode und dessen, was nach der Kompilierung mit \LaTeX zu sehen ist, wie in diesem Beispiel:

foo	●	bar	<pre>foo \mbox{% \put(\strip@pt\normalbaselineskip, \strip@pt\dimexpr0.2\normalbaselineskip\relax){% \circle*{\strip@pt\normalbaselineskip}}% \hspace{2\normalbaselineskip}} bar</pre>
-----	---	-----	--

Dabei soll nicht immer der gesamte Quellcode ausgegeben werden, so wie in obigen Beispiel, welches eigentlich zwei weitere Befehlszeilen umfasst.

```
\makeatletter
%START
foo \mbox{%
  \put(\strip@pt\normalbaselineskip,
        \strip@pt\dimexpr0.2\normalbaselineskip\relax){%
    \circle*{\strip@pt\normalbaselineskip}}%
  \hspace{2\normalbaselineskip}%
} bar
%STOP
\makeatother
```

Um die Ausgabe nur auf die eigentlichen Zeilen zu beschränken, wurden die sogenannten Marker %START und %STOP eingefügt, die den Ausgabebereich angeben, jedoch auf den Einfluss des Quellcodes keine Bedeutung haben, da sie durch % für \LaTeX als Kommentarzeile aufgefasst werden. Handelt es sich nicht um einen \LaTeX -Quellcode, sondern beispielsweise um ein kleines Shellskript, so benutzt man das dort gültige Kommentarzeichen #, um die Zeilen für den Code wirkungslos zu halten. Um die beiden Ausgaben nebeneinander gesetzt zu bekommen, wurde hier eine Tabelle benutzt, in deren rechte Spalte explizit nichts eingetragen wird. Der entsprechende Befehl wurde gleich in die Spaltendefinition eingefügt, sodass lediglich auf die Angabe des Spaltentrenners & zu achten ist, auch wenn keine weitere Angabe direkt innerhalb der Tabelle erfolgt.

```
\begin{tabular}{@{} m{0.2\linewidth}@{}
>{\lstinputlisting[includeangemarker=false,
rangeprefix=\\%,
```

```

\lernerange=START-STOP}%
{\jobname.tmp}}m{0.8\linewidth} @{}}
\begin{Beispiel}
\makeatletter
%START
foo \put(12,0){\circle*{\strip@pt\normalbaselineskip}}
\hspace{2\normalbaselineskip}bar
%STOP
\makeatother
\end{Beispiel}
& \tabularnewline
\end{tabular}

```

Die Umgebung `Beispiel` schreibt mit Hilfe von `fancyvrb` alles in eine temporäre Datei, die gleich wieder über `\input` eingelesen und somit ausgeführt wird.

```

\newcommand\Beispiel{%
\VerbatimEnvironment
\begin{VerbatimOut}{\jobname.tmp}}
\def\endBeispiel{%
\end{VerbatimOut}
\input{\jobname.tmp}}

```

Anstelle einer Tabelle hätte man hier auch eine `minipage`-Anordnung verwenden können, in beiden Fällen kann jedoch innerhalb des Beispiels kein Seitenumbruch erfolgen. Soll die Ausgabe untereinander erscheinen, so muss man eine andere Definition vornehmen. Im folgenden Beispiel wird eine Tabelle mit normaler Tabellenüberschrift und wieder teilweisem Quellcode ausgegeben:

Eine Tabelle ohne Anwendung von `tabularx`, die genau so breit ist wie die Zeile, was durch diesen Text gezeigt wird, der deswegen mehrere Zeilenumbrüche aufweist.

Tabelle 1: Ein Beispiel für eine berechnete Tabellenbreite

foo	bar	baz
und jetzt etwas längerer Text, der den Umbruch zeigen soll.	und jetzt etwas längerer Text, der den Umbruch zeigen soll.	und jetzt etwas längerer Text, der den Umbruch zeigen soll.

```

\begin{tabular}{@{>{\RaggedRight}p{3cm}
>{\Centering}p{4cm} |
>{\RaggedLeft}p{\linewidth-7cm-4\tabcolsep}@{}}\hline
foo & bar & baz\\\hline
und jetzt etwas \\"angerer Text, der den Umbruch zeigen soll. &
und jetzt etwas \\"angerer Text, der den Umbruch zeigen soll. &

```

```
und jetzt etwas \\\"angerer Text, der den Umbruch zeigen soll.\\\"\\hline
\\end{tabular}
```

Der ausgegebene Quelltext kann jetzt auch beliebig lang sein, da ein Seitenumbruch möglich ist. Durch die Verwendung von `fancyvrb` können keine UTF8-kodierten Zeichen verwendet werden, da diese weiterhin aktiv bleiben und somit in ihrer Expansion herausgeschrieben würden. Daher sind hier die Umlaute durch die entsprechende \TeX -Notation anzugeben, beispielsweise `\\"u`. Die entsprechende Beispielumgebung `BeispielB` für obiges `Beispiel` sieht wie folgt aus:

```
\newcommand\BeispielB{%
\VerbatimEnvironment
\begin{VerbatimOut}{\jobname.tmp}}
\def\endBeispielB{%
\end{VerbatimOut}
{\centering \input{\jobname.tmp}}
\lstinputlisting[incluserangemarker=false,rangeprefix=\%,
linenrange=START-STOP]{\jobname.tmp}}

\begin{BeispielB}
\begin{table}[!htb]
\hrulefill\par
Eine Tabelle ohne Anwendung von \texttt{tabularx}, die genau so breit
ist wie die Zeile, was durch diesen Text gezeigt wird, der deswegen
mehrere Zeilenumbr\"uche aufweist.
\caption{Ein Beispiel f\"ur eine berechnete Tabellenbreite}
%START
\begin{tabular}{@{}>\RaggedRight}p{3cm} |
>{\Centering}p{4cm} |
>{\RaggedLeft}p{\linewidth-7cm-4\tabcolsep}@{}}\hline
foo & bar & baz\\\"hline
und jetzt etwas \\\"angerer Text, der den Umbruch zeigen soll. &
und jetzt etwas \\\"angerer Text, der den Umbruch zeigen soll. &
und jetzt etwas \\\"angerer Text, der den Umbruch zeigen soll.\\\"\\hline
\end{tabular}
%STOP
\end{table}
\end{BeispielB}
```

Eine umgekehrte Reihenfolge der Ausgabe erreicht man durch entsprechendes Vertauschen der beiden Befehle `\input` und `\lstinputlisting`. Für die Ausgabe des Quellcodes lässt sich grundsätzlich auch der entsprechende Befehl `\VerbatimInput` aus dem Paket `fancyvrb` verwenden. Dass dies hier nicht erfolgt, ist eine willkürliche Entscheidung. Möchte man die Ausgabe von kompletten \LaTeX -Dokumenten oder Quellcode und Ergebnis zeigen, die sich nicht auf \LaTeX beziehen, so muss eine andere Vorgehensweise gewählt werden; ein einfaches `\input` funktioniert dann nicht mehr. Eine allgemeingültige Lösung wäre, das Ergebnis des Quellcodes als Grafik über `\includegraphics` einzubinden. Ver-

wendet man identische Schriften, so wird auch bei reinem Text kein Unterschied gegenüber der `\input`-Variante zu sehen sein. Damit die jetzt extern zu erzeugenden Grafiken auch entsprechend zugeordnet werden können, wird dafür ein eigener Zähler definiert: `\newcounter{BildZaehler}`. Die Dateien werden dann als `\jobname-\theBildZaehler.tex` erzeugt und können dann leicht zugeordnet werden.

Vollständige Beispieldokumente

Da unter Linux das Arbeiten mit `Makefile` eine große Vereinfachung des gesamten Ablaufs ermöglicht, sollen die Grafiken nicht aus dem Dokument heraus, sondern völlig unabhängig davon erzeugt werden. Nach einem ersten `pdfLATEX`-Lauf, welcher zwecks Geschwindigkeitserhöhung mit der Option `-draftmode` erfolgen kann, werden über das `Makefile` alle Dateien mit dem Namen `\jobname-*` mit dem entsprechenden Programm gestartet, in diesem Fall mit `XgLATEX` um `PSTricks`-spezifischen Code verwenden zu können, aber dennoch eine PDF-Datei als Ausgabe zu erhalten. Damit jeglicher weißer Rand von der Grafik entfernt wird, erfolgt nach dem `XgLATEX`-Lauf noch eine Anwendung von `pdfcrop`. Die Dateieindung der herausgeschriebenen Beispieldateien kann ohne weiteres für die Erkennung des zu startenden Programms benutzt werden, also beispielsweise `cpp` für ein C++-Beispiel. Nachdem die externen Dateien alle erzeugt wurden, erfolgt ein weiterer `pdfLATEX`-Lauf, der jetzt die erzeugten PDF-Dateien als Grafiken einliest.

Damit es keinen Fehler im ersten `pdfLATEX`-Durchlauf wegen der noch nicht erzeugten PDF-Datei gibt, wird deren Existenz über `\IfFileExists` abgefragt.¹ Die entsprechende Umgebung sieht dann wie folgt aus:

```
\newcounter{BildZaehler}
\newcommand\BeispielC{%
\refstepcounter{BildZaehler}%
\VerbatimEnvironment
\begin{VerbatimOut}{\jobname-\theBildZaehler.tex}
\def\endBeispielC{%
\end{VerbatimOut}
\IfFileExists{\jobname-\theBildZaehler.pdf}% PDF schon vorhanden?
    {\includegraphics{\jobname-\theBildZaehler.pdf}}%
    {\fbox{PDF fehlt!}}% nein, dann Hinweis
\lstinputlisting[linerange=\begin\{document\}-\end\{document\},
includerangemarker=false]%
{\jobname-\theBildZaehler.tex}}
```

¹ Alternativ könnte das Paket `graphicx` auch mit der Option `demo` geladen werden; nicht existierende Grafiken werden dann durch einen schwarzen Kasten ersetzt.

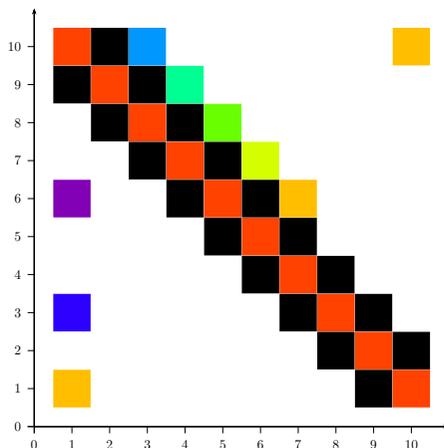
Diese ist in diesem Fall nur für L^AT_EX-, beziehungsweise X_YL^AT_EX-Dokumente geeignet. Damit man als Anwender nur den eigentlichen Beispielcode angeben muss und nicht auch die L^AT_EX-typische Präambel sowie Postambel, werden die Makros `\FVB@VerbatimOut` und `\FVE@VerbatimOut` aus dem Paket `fancyvrb` neu definiert, um zum einen das Einfügen der Präambel und zum anderen der Postambel zu ermöglichen.

```
\renewcommand\FVB@VerbatimOut[1]{%
  \@bsphack%
  \begingroup
  \FV@UseKeyValues%
  \FV@DefineWhiteSpace%
  \def\FV@Space{\space}%
  \FV@DefineTabOut%
  \def\FV@ProcessLine##1{\toks@{##1}\immediate\write\FV@OutFile{\the\toks@}}%
  \immediate\openout\FV@OutFile #1\relax%
  \SchreibePSTricksPraeambel%<<=== Praeambel rausschreiben
  \let\FV@FontScanPrep\relax
  \let\@noligs\relax%
  \FV@Scan}
\renewcommand\FVE@VerbatimOut{%<<=== Postambel rausschreiben
  \SchreibeZeile{\string\end{document}}% <<
  \immediate\closeout\FV@OutFile\endgroup\@esphack}
\makeatother
```

Das Makro `\SchreibeZeile` wird benutzt, um jeweils eine ganz spezifische Präambel benutzen zu können, im folgenden Beispiel für PSTricks-spezifischen Code. Für ein C⁺⁺-Beispiel würde dann eine andere definiert werden.

```
\newcommand\SchreibeZeile[1]{%
  \begingroup%
  \let\protect\@unexpandable\protect%
  \edef\reserved@a{\immediate\write\FV@OutFile{#1}}%
  \reserved@a%
  \endgroup}
\newcommand\SchreibePSTricksPraeambel{%
  \SchreibeZeile{\string\documentclass{article}}%
  \SchreibeZeile{\string\usepackage{pstricks-add}}%
  \SchreibeZeile{\string\pagestyle{empty}}%
  \SchreibeZeile{\string\begin{document}}%
}
```

Mit diesen Definitionen sind die Grundlagen gelegt, um die neue Umgebung `BeispielC` anwenden zu können, was hier an einem Beispiel für die Darstellung eines sogenannten Surface-Plots gezeigt werden soll.



```
\psscalebox{0.5}{%
\begin{pspicture}(-0.5,-0.75)(11,11)
\psaxes[ticks=-5pt 0]{->}(11,11)
\psMatrixPlot[colorType=5,dotsize=1.1cm,xStep=1,yStep=1,dotstyle=square*]{10}{10}{
  ↪matrix1.data}
\end{pspicture}}
```

Möchte man die Präambel flexibler gestalten und eventuell auch ganz bestimmte Teile daraus als Code ausgeben, so lässt sich dies über zwei zusätzliche Makros erreichen, die zum einen den unsichtbaren und zum anderen einen sichtbaren Teil der Präambel festlegen. Hier wird willkürlich festgelegt, dass der unsichtbare Teil zuerst in die externe Datei ausgegeben wird. Das Makro `\FVB@VerbatimOut`, welches bereits oben modifiziert wurde, wird dahingehend geändert, dass jetzt nicht mehr eine Präambel für einen ganz bestimmten Fall eingebunden wird (`\SchreibPSTricksPraeambel`), sondern eine Musterpräambel (`\SchreibPraeambel`). Diese lädt jetzt keine weiteren Pakete, sodass der Anwender diese nun selbst vorgeben muss, damit aber flexibel in der Anwendung bleibt. Das Paket `listings` erlaubt die Angabe nicht nur eines Bereiches, sondern lässt eine sortierte und kommaseparierte Liste zu, die in geschweifte Klammern zu setzen ist:

```
\lstinputlisting[
  linerange={\%PSTART-\%PSTOP, \begin\{document\}-\end\{document\}},
  includerangemarker=false]{\jobname-\theBildZaehler.tex}%
```

In diesem Fall würde alles, was im Quelldokument zwischen `%START` und `%STOP`, sowie zwischen `\begin{document}` und `\end{document}` steht, ausgegeben. Damit Präambel und Textkörper besser voneinander zu unterscheiden sind, sollen sie in unterschiedlicher Hintergrundfarbe und mit einem zusätzlichen kleinen

Abstand dazwischen ausgegeben werden, sodass von der obigen Möglichkeit kein Gebrauch gemacht wird. Der entsprechende Teil für das Einfügen lautet dann:

```
\IfFileExists{\jobname-\theBildZaeher.pdf}%
  {\begin{center}\expandafter\includegraphics\expandafter[\
    ↪GraphicxOptionen]
   {\jobname-\theBildZaeher.pdf}
  \end{center}}%
  {\fbox{PDF fehlt!}}%
\def\GraphicxOptionen{}%
\lstinputlisting[backgroundcolor=\color{black!15},
  linerange=\%PSTART-\%PSTOP,
  includerangemarker=false,]%
  {\jobname-\theBildZaeher.tex}
\lstinputlisting[backgroundcolor={},
  linerange=\begin\{document\}-\end\{document\},
  includerangemarker=false]%
  {\jobname-\theBildZaeher.tex}%
\gdef\Unsichtbarer@Teil{}%
\gdef\Sichtbarer@Teil{}%
```

Das Makro `\GraphicxOptionen` speichert den optionalen Parameter der Beispielumgebung `BeispielD`, der Key/Value-Angaben für `\includegraphics` enthalten kann. Der unsichtbare Teil der Präambel ist als Parameter dem Makro `\PraeambelUnsichtbar` und der sichtbare Teil dem Makro `\PraeambelSichtbar` zu übergeben. Das externe \LaTeX -Dokument erhält dann jeweils die folgende Präambel:

```
\newcommand\SchreibePraeambel{%
  \SchreibeZeile{\string\documentclass{article}}%
  \SchreibeZeile{\string\pagestyle{empty}}%
  \SchreibeZeile{\Unsichtbarer@Teil}
  \SchreibeZeile{\@percentchar PSTART}
  \SchreibeZeile{\Sichtbarer@Teil}%
  \SchreibeZeile{\@percentchar PSTOP}
  \SchreibeZeile{\string\begin{document}}%
}
```

Ein Seitenumbruch ist jetzt nach der Abbildung und innerhalb der Code-Ausgabe möglich, was auch durch das folgende Beispiel gezeigt wird.

Die Bindungsenergie im Tröpfchenmodell setzt sich aus folgenden Teilen zusammen:

- dem Oberflächenanteil,
- dem Volumenanteil,

$$E = a_v A + -a_f A^{2/3} + -a_c \frac{Z(Z-1)}{A^{1/3}} + -a_s \frac{(A-2Z)^2}{A} + E_p \quad (1)$$

- dem Coulomb-Anteil,
- der Symmetrieenergie,
- sowie einem Paarbildungsbeitrag.

```
\usepackage{tgpagella}
\usepackage{pst-node}
```

```
\psset{nodesep=3pt}
Die Bindungsenergie im Tröpfchenmodell setzt sich aus folgenden Teilen zusammen:
\begin{itemize}
\item dem \node{b}{Oberflächenanteil},
\item dem \node{a}{Volumenanteil},\!\!1cm
\def\xstrut{\vphantom{\frac{(A)^1}{(B)^1}}}
\begin{equation}
E =
\node[t]{ae}{\psframebox*[fillcolor=darkyellow,
linestyle=none]{\xstrut a_vA}} +
\node[t]{be}{\psframebox*[fillcolor=lightgray,
linestyle=none]{\xstrut -a_fA^{2/3}}} +
\node[t]{ce}{\psframebox*[fillcolor=green,
linestyle=none]{\xstrut -a_c\frac{Z(Z-1)}{A^{1/3}}}} +
\node[t]{de}{\psframebox*[fillcolor=cyan,
linestyle=none]{\xstrut -a_s\frac{(A-2Z)^2}{A}}} +
\node[t]{ee}{\psframebox*[fillcolor=yellow,
linestyle=none]{\xstrut E_p}}
\end{equation}\!\!0.25cm
\item dem \node{c}{Coulomb-Anteil},
\item der \node{d}{Symmetrieenergie},
\item sowie einem \node{e}{Paarbildungsbeitrag}.
\end{itemize}
\ncurve[angleA=-90,angleB=90]{->}{a}{ae}\ncurve[angleB=45]{->}{b}{be}
\ncurve[angleB=-90]{->}{c}{ce}\ncurve[angleB=-90]{->}{d}{de}
\ncurve[angleB=-90]{->}{e}{ee}
```

Das Makro `\Praelambel`, welches den unsichtbaren und sichtbaren Teil der Präambel speichert, ist etwas aufwändiger, da zum einen alle Sonderzeichen, wie `\`,

\$, &, #, ^, _, % und ~ und zum anderen Zeilenenden gesondert behandelt werden müssen. Bei Angabe eines beliebigen optionalen Arguments wird von einem unsichtbaren Teil der Präambel ausgegangen.

```
\def\MakeVerbatimNewLine{^^}
\begingroup
\catcode'\^^M=\active %
\gdef\obeylines@Preamble{\catcode'\^^M\active \let^^M\MakeVerbatimNewLine}%
\endgroup

\newcommand\Praeambel{%
\par
\begingroup
\makeatother
\let\do\@makeother
\do\ \do\ \do\ $\do\ &\do\ #\do\ ^\do\ _\do\ %|do|~
\obeylines@Preamble
\ifnextchar[\PraeambelUnsichtbar@{\PraeambelSichtbar@[]}]
\long\def\PraeambelUnsichtbar@[#1]#2{\long\xdef\@gtempa{#2}%
\endgroup\let\Unsichtbarer@Teil\@gtempa}
\long\def\PraeambelSichtbar@[#1]#2{\long\xdef\@gtempa{#2}%
\endgroup\let\Sichtbarer@Teil\@gtempa}
```

Damit kann dann jeweils die Ausgabe der Präambel im Beispielcode gesteuert werden, damit nur das ausgegeben wird, was für den Leser von Interesse ist, während andere Dinge zwar frei definiert werden können und auch in die exportierte TeX-Datei geschrieben werden, aber nicht als Quellcode erscheinen. Die entsprechenden Angaben für das obige Beispiel lauten:

```
\Praeambel[Unsichtbar]{\usepackage[T1]{fontenc}
\usepackage{mathpazo}
\usepackage{pstricks}
\newrgbcolor{lila}{0.6 0.2 0.5}
\newrgbcolor{darkyellow}{1 0.9 0}}

\Praeambel{\usepackage{tgpagella}
\usepackage{pst-node}}
```

Beliebiger Quellcodetyp

Es wurde bereits darauf hingewiesen, dass es prinzipiell unerheblich ist, von welchem Quellcodetyp die exportierte Datei ist. Anhand der Dateierweiterung kann über das Makefile eine gezielte Weiterverarbeitung erfolgen. Lediglich der Beispielumgebung muss der zu exportierende Datentyp bekannt sein. Abschließend soll ein Beispiel für ein externes Perlprogramm gezeigt werden, welches aus diesem Dokument heraus geschrieben und dann extern ausgeführt wird. Die Ausgabe des Programms soll unter dem gleichen Basisnamen mit der Dateierwei-

terung .aus gespeichert werden, sodass sie dann in dieses Dokument als reiner Text eingefügt werden kann.

Ein standardisierter Perlcode könnte folgenden Kopf (Präambel) aufweisen:

```
\newcommand\SchreibePerlPraeambel{%
  \SchreibeZeile{\numbersign !/usr/bin/perl}%
  \SchreibeZeile{\numbersign }%
  \SchreibeZeile{\numbersign Herbert Voss 20110201}%
  \SchreibeZeile{use strict;}%
  \SchreibeZeile{\Unsichtbarer@Teil}
  \SchreibeZeile{\numbersign PSTART}
  \SchreibeZeile{\Sichtbarer@Teil}%
  \SchreibeZeile{\numbersign PSTOP}
  \SchreibeZeile{\numbersign }%
  \SchreibeZeile{\numbersign !!Hauptteil-Anfang!!}%
}
```

Die Definition der Beispielumgebung unterscheidet sich prinzipiell nicht von der bereits oben gezeigten \LaTeX -Variante; statt der erzeugten PDF-Datei wird jetzt ebenfalls mit `\lstinputlisting` einfach die vom externen Perlprogramm erzeugte Textausgabe eingelesen. Das optionale Argument der Umgebung `Beispiele` kann für die Formatierung benutzt werden, die Angaben werden an `\lstinputlisting` weitergereicht.

```
\newcommand\Beispiele[1][{}]{%
  \def\lstOptionen{#1}%
  \refstepcounter{BildZaehler}%
  \VerbatimEnvironment
  \begin{VerbatimOut}{\jobname-\theBildZaehler.pl}
  \def\endBeispiele{%
    \end{VerbatimOut}
    \IfFileExists{\jobname-\theBildZaehler.aus}%
      {\expandafter\lstinputlisting\expandafter[\lstOptionen}%
       {\jobname-\theBildZaehler.aus}}%
      {\fbox{Ausgabe fehlt!}}%
  \medskip
  \def\lstOptionen{}%
  \lstinputlisting[backgroundcolor=\color{black!15},
    linerange=\#PSTART-\#PSTOP,
    includerangemarker=false,]%
    {\jobname-\theBildZaehler.pl}
  \lstinputlisting[backgroundcolor={},
    linerange=\#\#\#Hauptteil-Anfang-\#\#\#Hauptteil-Ende,
    includerangemarker=false]%
    {\jobname-\theBildZaehler.pl}%
  \gdef\Unsichtbarer@Teil{}%
  \gdef\Sichtbarer@Teil{}%
}
```

Das folgende Beispiel bestimmt sogenannte Kaprekarkonstanten (<http://de.wikipedia.org/wiki/Kaprekar-Konstante>). Dies sind natürliche Zahlen mit folgender Eigenschaft: Werden die Ziffern auf- und absteigend sortiert, so entsteht eine größte und kleinste Zahl, deren Differenz wieder gleich der Ausgangszahl ist. Um das Beispiel hier nicht zu umfangreich werden zu lassen, wird die einfachste Methode zur Bestimmung verwendet; es werden einfach alle Zahlen durchlaufen und jeweils überprüft.

```
Bestimmung von Kaprekarkonstanten
```

```
1-stellig:
2-stellig:
3-stellig: 495,
4-stellig: 6174,
5-stellig:
```

```
### Bestimmung von Kaprekarkonstanten ###
```

```
my $zahl = 1;
my $anfang = 1;
my $ende = 10;
```

```
print("Bestimmung von Kaprekarkonstanten\n");
while ($zahl < 6) {
  print "$zahl-stellig: ";
  foreach ($anfang..$ende) { # für jede Zeile $_
    my @Zeichen = split(//,$_);
    my $Min = join("",sort(@Zeichen));
    my $Max = reverse($Min);
    my $Dif=$Max-$Min;
    if($_ eq $Dif) { print $_,","; }
  }
  $zahl = $zahl+1;
  $anfang = $anfang*10;
  $ende = $ende*10;
  print "\n";
}
###Hauptteil-Ende!!
```

Die Ausgabe des kompletten Perlprogramms unterscheidet sich ebenfalls nicht von einem L^AT_EX-Beispiel:

```
\Preamble[Unsichtbar]{# Ein Beispiel für eine Kaprekarzahl}
\Preamble{### Bestimmung von Kaprekarzahlen ###}
my $zahl = 1;
my $anfang = 1;
my $ende = 10;}

\begin{Beispiele}[basicstyle=\ttfamily\small,frame=LR]
print("Bestimmung von Kaprekarkonstanten\n");
```

```

while ($zahl < 6) {
  print "$zahl-stellig: ";
  foreach ($anfang...$ende) { # für jede Zeile $_
    my @Zeichen = split(//,$_);
    my $Min = join("",sort(@Zeichen));
    my $Max = reverse($Min);
    my $Dif=$Max-$Min;
    if($_ eq $Dif) { print $_," "; }
  }
  $zahl = $zahl+1;
  $anfang = $anfang*10;
  $ende = $ende*10;
  print "\n";
}
\end{BeispielE}

```

Zusammenfassung

In diesem Artikel wurde gezeigt, wie man externe Quelldateien beliebigen Typs erzeugen kann, die dann über ein Makefile nach einem ersten \LaTeX -Lauf ausgeführt werden. Die Dateiendung der erzeugten Datei sollte den jeweiligen Programmtyp kennzeichnen, der dann zur Ausführung kommt. Die Ausgabe der Programme kann in einem weiteren \LaTeX -Lauf als Grafik oder als Text eingebunden werden. Der Autor behält somit die volle Kontrolle über die Beispielprogramme. Bei sehr vielen Beispielen kann die jeweils erzeugte Datei auch erst in ein temporäres Verzeichnis geschrieben werden, um dann über den Linuxbefehl `diff` mit einer eventuell schon existierenden Datei verglichen zu werden und daraus folgend, nur dann zur Ausführung kommen, wenn sich der Quellcode geändert hat.

Für das Einfügen der Ausgabe in das Dokument kann man noch weitere optionale Parameter vorsehen, um beispielsweise eine wahlweise Links-/Rechtsanordnung zu ermöglichen. Die Grafiken könnte man jeweils mit `pdfcrop` bearbeiten, um einen eventuellen weißen Rand automatisch zu beseitigen.

Literatur

- [1] Carsten Heinz: *The Listings Package*, Version 1.4., Feb. 2007; CTAN:macros/latex/contrib/listings/
- [2] Rolf Niepraschk: *The showexpl Package*, Version 0.3h., Feb. 2007; CTAN:macros/latex/contrib/showexpl/
- [3] Timothy Van Zandt: *The fancyvrb package – Fancy Verbatims in \LaTeX* , Version 2.8, Mai 2010; CTAN:macros/latex/contrib/verbatim/

Tipps und Tricks

biblatex – abgehakt!

Rolf Niepraschk

Einleitung

In zwei früheren Artikeln dieser Zeitschrift zu `biblatex` (siehe [1] und [2]) wurden bereits die vielfältigen Möglichkeiten dieser mittlerweile sehr beliebten Art der Erzeugung von Bibliografien gezeigt. In diesem Beitrag wird beschrieben, wie man die Darstellung der einzelnen Einträge eines Literaturverzeichnisses gezielt ergänzen kann.

Der Wunsch

In einem Literaturverzeichnis sollen die einzelnen Einträge mit Kennzeichnungen versehen werden, die zeigen, ob das Buch oder der Artikel im Original oder als Kopie vorhanden ist.

Die Lösung

Es bietet sich an, das Eintragsfeld »keywords« bei den Datenbankeinträgen zu nutzen. So könnte beispielsweise »exist« für im Original und »copy« für als Kopie vorhanden stehen. Auf diese Weise wäre es sehr einfach, getrennte Literaturverzeichnisse in Abhängigkeit vom jeweiligen Schlüsselwort auszugeben (siehe dazu die Dokumentation zu `biblatex` [4]).

Hier soll ein etwas anderer Weg der Kennzeichnung beschritten werden, da man getrennte Literaturverzeichnisse möglicherweise auch nach anderen Kriterien auswählen will. Stattdessen sollen im Randbereich auf der Höhe eines Eintrages das Zeichen ✓ (Original) oder [✓] (Kopie) erscheinen.

Für einen ersten Test wurde in einem Dokument mit Literaturverzeichnis der folgende Code ergänzt:

```

\usepackage{marginnote,amsfonts}
\newcommand*\copyMarker{[\checkmark]}
\AtEveryBibitem{%
  \marginnote[\parbox{\marginparwidth}{\hfill\copyMarker}]%
    {\parbox{\marginparwidth}{\copyMarker\hfill}}%
}

```

Mit Hilfe der biblatex-Anweisung `\AtEveryBibitem` kann man eigenen Code ausführen, bevor der zu dem Bibliografie-Eintrag gehörende Text ausgegeben wird. Hier wird sie verwendet, um – vorerst testweise – das Zeichen [✓] in den Randbereich eines jeden Bibliografie-Eintrages zu schreiben. Die Anweisung `\marginnote` aus dem Paket `marginnote` (siehe [3]) wurde gewählt, da dessen Positionierung erfahrungsgemäß verlässlicher ist als bei Verwendung von `\marginpar`.

Das Ergebnis sieht schon so wie gewünscht aus, wobei natürlich noch die Abhängigkeit von den Schlüsselwörtern fehlt:

Literatur

- [1] Albert Einstein. »Zur Elektrodynamik bewegter Körper«. In: *Annalen der Physik* [✓] 17 (1905), S. 891–921.
- [2] Donald E. Knuth. *Computers & Typesetting*. Bd. A: *The TeXbook*. Reading, Mass.: [✓] Addison-Wesley, 1984.
- [3] Donald E. Knuth. *Computers & Typesetting*. Bd. C: *The METAFONTbook*. Reading, [✓] Mass.: Addison-Wesley, 1986.

Um zu testen, ob beim aktuellen Bibliografie-Eintrag das »keywords«-Feld entweder »exist« oder »copy« enthält, muss der Code erweitert werden. Als Erstes wird mit der Anweisung `\setBibItemMarker` die Möglichkeit geschaffen, einem bestimmten Schlüsselwort die Definition einer Markierung zuzuordnen.

```

\newcommand*\copyMarker{[\checkmark]}
\newcommand*\existMarker{\mbox{\phantom{[\checkmark]\phantom{]}}}
\newcommand*\@item@marker{}
\newcommand*\setBibItemMarker[2]{%
  \expandafter\gdef\csname @bibItemMarker@#1\endcsname{#2}%
}

```

Nach Aufruf von

```

\setBibItemMarker{exist}{\existMarker}
\setBibItemMarker{copy}{\copyMarker}

```

entstehen die zwei neuen Anweisungen `\@bibItemMarker@exist` und `\@bibItemMarker@copy`. Das Schlüsselwort, also `exist` oder `copy`, ist Bestandteil ihrer Namen, und ihr Inhalt ist im einen Fall `\existMarker` und im anderen `\copyMarker`. Damit ist die erwähnte Zuordnung erreicht. Durch die Namensgebung der beiden Anweisungen ist es später möglich, sie programmtechnisch in Abhängigkeit vom Schlüsselwort zu »konstruieren«.

Der `\AtEveryBibitem`-Aufruf muss zusätzlich zu der Ausgabe in den Randbereich einen Test enthalten, um festzustellen, ob eines der beiden Schlüsselwörter im »keywords«-Feld enthalten ist, um letztendlich die Auswahl des passenden Zeichens für die Ausgabe treffen zu können:

```
\AtEveryBibitem{%
  \def\@item@marker{%
    \def\do#1{%
      \ifcsmacro{@bibItemMarker@#1}{%
        \def\@item@marker{\csname @bibItemMarker@#1\endcsname}%
        \listbreak
      }{%
    }%
  }%
  \edef\blx@tempc{\thefield{keywords}}%
  \expandafter\docsvlist\expandafter{\blx@tempc}%
  \marginnote[\parbox{\marginparwidth}{\hfill\@item@marker}]%
    {\parbox{\marginparwidth}{\@item@marker\hfill}}%
}
```

Im Einzelnen wird Folgendes getan: In einer Schleife (die Anweisung `\docsvlist` mit zugehörigem `\do`) wird aus jedem Eintrag der aktuellen »keywords«-Liste testweise ein Makroname `\@bibItemMarker@???` erzeugt. (`???` symbolisiert hier den jeweiligen Eintrag der Liste.) Existiert ein Makro mit einem solchen Namen (`\ifcsmacro`), so bekommt die Anweisung `\@item@marker` dessen Inhalt zugewiesen, und die Schleife wird abgebrochen. Näheres zu `\docsvlist` und `\ifcsmacro` findet man in der Beschreibung des Paketes `etoolbox` (siehe [5]).

Das Ergebnis sieht nun folgendermaßen aus:

Literatur

- [1] Albert Einstein. »Zur Elektrodynamik bewegter Körper«. In: *Annalen der Physik* [✓]
17 (1905), S. 891–921.
- [2] Donald E. Knuth. *Computers & Typesetting*. Bd. A: *The T_EXbook*. Reading, Mass.: ✓
Addison-Wesley, 1984.
- [3] Donald E. Knuth. *Computers & Typesetting*. Bd. C: *The METAFONTbook*. Reading,
Mass.: Addison-Wesley, 1986.

Die hierzu verwendete BibT_EX-Datei enthält die folgenden Einträge einschließlich der erwähnten »keywords«-Felder:

```
@Book{knuth:ct:a,
  hyphenation = {american},
  author      = {Knuth, Donald E.},
  title       = {The \TeX book},
  sorttitle   = {Computers & Typesetting A},
  maintitle   = {Computers \& Typesetting},
  volume      = {A},
  publisher    = {Addison-Wesley},
  location     = {Reading, Mass.},
  date        = {1984},
  keywords    = {exist}
}

@Book{knuth:ct:c,
  hyphenation = {american},
  author      = {Knuth, Donald E.},
  title       = {The METAFONTbook},
  sorttitle   = {Computers & Typesetting C},
  maintitle   = {Computers \& Typesetting},
  volume      = {C},
  publisher    = {Addison-Wesley},
  location     = {Reading, Mass.},
  date        = {1986}
}

@Article{einstein:elektrodynamik,
  hyphenation = {german},
  author      = {Einstein, Albert},
  title       = {Zur Elektrodynamik bewegter K{"o}rper},
  journaltitle = {Annalen der Physik},
  pages       = {891-921},
  volume      = {17},
  date        = {1905},
  keywords    = {copy}
}
```

Es sei noch erwähnt, dass die Definitionen, falls sie im Dokumentenkopf eingefügt werden sollen und das Zeichen @ enthalten, innerhalb eines durch \makeatletter eingeleiteten und durch \makeatother beendeten Block stehen müssen. Um eine bessere Übersicht zu haben, empfiehlt es sich allerdings, sie in eine eigene Stil- oder Klassen-Datei einzufügen, in der dann \makeatletter und \makeatother entfallen.

Fazit

Mit der Anweisung `\AtEveryBibitem` bietet `biblatex` eine sehr vielseitig nutzbare Möglichkeit, die einzelnen Einträge eines Literaturverzeichnisses zu ergänzen. Zumindest für Spezialfälle bietet es sich an, davon Gebrauch zu machen.

Literatur

- [1] Dominik Waßenhoven: *Bibliographien erstellen mit biblatex (Teil 1); Die T_EX-nische Komödie*; 2/08, S. 53–75; Mai 2008.
- [2] Dominik Waßenhoven: *Bibliographien erstellen mit biblatex (Teil 2); Die T_EX-nische Komödie*; 4/08, S. 31–51; Nov. 2008.
- [3] Markus Kohm: *Non-Floating Margin Notes with marginnote Package*; CTAN: `/macros/latex/contrib/marginnote/marginnote.pdf`; Jan. 2010.
- [4] Philipp Lehman: *The biblatex package. Programmable bibliographies and citations*; CTAN: `/macros/latex/contrib/biblatex/doc/biblatex.pdf`; Jan. 2011.
- [5] Philipp Lehman: *The etoolbox Package. An e-TeX Toolbox for Class and Package Authors*; CTAN: `/macros/latex/contrib/etoolbox/etoolbox.pdf`; Jan. 2011.

Makro oder Umgebung?

Heiko Oberdiek, Rolf Niepraschk, Herbert Voß

Man erhält in Newsgruppen und Mailinglisten oder an anderen Orten nicht selten den Tipp, statt eines Makros eine Umgebung mit dem selben Namen zu verwenden, beispielsweise `\begin{small}... \end{small}` anstelle von `{\small...}`. Obwohl diese Vorgehensweise auf den ersten Blick recht elegant wirkt, bringen diese »Pseudoumgebungen« mehr Probleme als Nutzen, was am folgenden Beispiel gezeigt werden soll:

```
\documentclass{article}
\usepackage{longtable}
\setlength{\parindent}{0pt}
\begin{document}
\begin{huge}
Hallo\hfill Welt\
Hallo\hfill Welt
\end{huge}
\begin{small}
```

```
\begin{longtable}{ll}
Hallo & Welt
\end{longtable}
\end{small}
\end{document}
```

`\begin{small}` steht hier im horizontalen Modus, man will ja auch mal ein Wort im laufenden Absatz klein schreiben. Der Absatz wird erst durch ein `\par` in `\begin{longtable}` beendet. Dabei wird der letzte Leerraum, das Zeilenende nach `\begin{small}` entfernt. Die Leerräume durch die Zeilenenden in den zwei Zeilen zuvor bleiben jedoch erhalten. Bewundern kann man dies im Beispiel beim zweiten »Welt«, das dadurch nach links verschoben wird.

Hallo
Hallo

Welt
Welt

Hallo Welt

Wie man weiterhin erkennt, stimmt der Zeilenabstand der mit `\huge` gesetzten Zeilen nicht mehr. TeX nimmt die Einstellung, die am Absatzende aktiv ist. Der Absatz wird aber erst in `\begin{longtable}` beendet. Hier ist aber bereits auf `\small` umgeschaltet worden.

Wenn der Wunsch nach einer Umgebung besteht, kann man sich eine solche in Analogie zu der Umgebung `sloppypar`, der Umgebungsform für `\sloppy`, definieren:

```
\newenvironment{sloppypar}{\par\small}{\par}
```

Das Beispiel wird dann zu:

```
[...]
\newenvironment{smallpar}{\par\small}{\par}
\newenvironment{hugepar}{\par\huge}{\par}

\begin{document}
\begin{hugepar}
Hallo\hfill Welt\
Hallo\hfill Welt
\end{hugepar}
\begin{smallpar}
\begin{longtable}{ll}
Hallo & Welt
\end{longtable}
\end{smallpar}
\end{document}
```

mit der Ausgabe:

Hallo
Hallo

Welt
Welt

Hallo Welt

Die oben gezeigten Unzulänglichkeiten sind damit behoben.

enumerate und Referenzen

Heiko Oberdiek, Rolf Niepraschk, Herbert Voß

Auf die einzelnen Einträge einer Umgebung `enumerate` kann in der üblichen Weise mit `\label` referenziert werden.

1. Erste Zeile	<code>\begin{enumerate}</code>
1A Zweite Zeile	<code>\item \label{Eins} Erste Zeile</code>
2. Dritte Zeile	<code>\item[1A] \label{Eins-A}Zweite Zeile</code>
	<code>\item \label{Drei} Dritte Zeile</code>
	<code>\end{enumerate}</code>
$1 - 1 - 2$	<code>\ref{Eins} -- \ref{Eins-A} -- \ref{Drei}</code>

Wie un schwer zu erkennen ist, gibt es jedoch ein Problem, wenn das optionale Argument von `\item` benutzt wird. In diesem Fall ist die interne Zählung nicht mehr korrekt, da \LaTeX nun nicht mehr das für die Referenzierung notwendige `\refstepcounter` ausführt. Die Referenz bezieht sich daher auf die vorhergehende Marke, sodass `\ref{Eins-A}` hier identisch zu `\ref{Eins}` ist.

Für eine korrekte Referenzierung muss daher ein eigenes Makro definiert werden:

```
\makeatletter
\newcommand*{\itemlabel}[2]{%
  \item[#1]}%
\leavevmode
\rlap{% Setze Anker fuer hyperref
  \raisebox{.7\baselineskip}[0pt]{%
    \kern-\leftmargin\csname phantomsection\endcsname
```

```

    }%
  }%
  \protected@edef\@currentlabel{#1}% Setze Inhalt fuer Label
  \label{#2}%
  \ignorespaces
}
\makeatother

```

1. Erste Zeile	<code>\begin{enumerate}</code>
1A Zweite Zeile	<code>\item \label{Eins1} Erste Zeile</code>
2. Dritte Zeile	<code>\ItemLabel{1A}{Eins-A1} Zweite Zeile</code>
	<code>\item \label{Dreil} Dritte Zeile</code>
	<code>\end{enumerate}</code>
$1 - 1A - 2$	<code>\ref{Eins1} -- \ref{Eins-A1} -- \ref{Dreil}</code>

Sortierung im Index

Heiko Oberdiek, Rolf Niepraschk, Herbert Voß

In einem Index werden die Einträge normalerweise alphabetisch sortiert aufgeführt. Treten Symbole als zu sortierende Begriffe auf, so ist ihre Position innerhalb des Zeichensatzes entscheidend, was aber nicht immer gewünscht ist. Will man, dass sie in der Reihenfolge, wie sie im Dokument auftreten, sortiert werden, muss man ein eigenes Makro (`\SymbolIndex`) verwenden. Es soll beispielsweise für das Komma einen Eintrag der Form

```
\indexentry{"$0000001",@"\verb9",9}{i}
```

in der `.idx`-Datei erzeugen, womit die gewünschte seitenweise Sortierung erreicht werden kann. Vor »@« steht der Sortierschlüssel. Das Dollar-Zeichen erzwingt die Sortierung unter den Symbolen, dann folgt die absolute Seitennummer immer mit sieben Stellen. Die Anführungszeichen dienen als Quote-Zeichen, um dem folgenden Zeichen eine Sonderbedeutung für `makeindex` zu nehmen.

Der folgende Quelltext eines Beispieldokumentes zeigt die Definition des Makros `\SymbolIndex` sowie seine Anwendung:

```

\documentclass[paper=a6,pagesize,open=any]{scrbook}
\usepackage{makeidx}
\makeindex
\usepackage{zref-abspage}
\makeatletter

```

```

\newcommand*\SymbolIndex{%
  \@bsphack
  \begingroup
  \@sanitize
  \@SymbolIndex}
\def\@SymbolIndex#1{%
  \let\TheSymbolNumberPage\relax
  \@wrindex{%
    \@SymbolChar
    \TheSymbolNumberPage
    \@QuoteChar#1%
    \@ActualChar
    \@QuoteChar
    \@backslashchar verb9\@QuoteChar#19}}
\newcommand*{\TheSymbolNumberPage}{%
  \ifnum\value{abspage}<100 0\fi
  \ifnum\value{abspage}<1000 0\fi
  \ifnum\value{abspage}<10000 0\fi
  \ifnum\value{abspage}<100000 0\fi
  \ifnum\value{abspage}<1000000 0\fi
  \ifnum\value{abspage}<10000000 0\fi
  \the\value{abspage}}
\def\@ActualChar{@}
\def\@QuoteChar{"}
\def\@SymbolChar{$}
\makeatother

\begin{document}
\frontmatter
\chapter{Introduction}
\index{hello}
\SymbolIndex{,}
\SymbolIndex{$}
Hello World
\mainmatter
\chapter{Main chapter}
bar \index{bar} X \SymbolIndex{@} \newpage
foo \index{foo} Y \SymbolIndex{:} \newpage
foo \index{foo} bar \index{bar} Z \SymbolIndex{[]} \newpage
\chapter{Next chapter}
X \SymbolIndex{\} foo \index{foo} Y \SymbolIndex{#} \newpage
bar \index{bar} Z \SymbolIndex{!} foo \index{foo}

\printindex
\end{document}

```

Die von `makeindex` mit dem einfachen Aufruf

```
makeindex -o beispiel.ind beispiel.idx
```

erzeugt Datei `beispiel.ind` hat folgendes Aussehen:

```

\begin{theindex}
  \item \verb9$, i
  \item \verb9,, i
  \item \verb9@, 1
  \item \verb9:, 2
  \item \verb9[, 3
  \item \verb9#, 4
  \item \verb9\, 4
  \item \verb9!, 5
  \indexspace
  \item bar, 1, 3, 5
  \indexspace
  \item foo, 2--5
  \indexspace
  \item hello, i
\end{theindex}

```

Die Ausgabe des Indexes ergibt sich dann in der gewünschten Form; die Symbole erscheinen entsprechend ihres Auftretens im Dokument, während die normalen Wörter alphabetisch sortiert ausgegeben werden.

Index

\$, i
 ,, i
 @, 1
 :, 2
 [, 3
 #, 4
 \, 4
 !, 5

 bar, 1, 3, 5

 foo, 2–5

 hello, i

6

Mathematik für »Tippfaule«

Herbert Voß

Insbesondere in längeren Formeln kann die übliche Klammernotation mit der Kombination `\left(...\right)` zu einem ziemlich unübersichtlichen Aufbau führen, was bei einer Fehlersuche erschwerend wirkt. Es kann daher sinnvoll sein, wenn man nicht nur für »Tippfaule« die beiden Klammern (`()`) zu aktiven Zeichen macht, die dann automatisch in entsprechende `\left(...\right)`-Kombinationen umgewandelt werden. Ohne die Verwendung des Paketes `amsmath` würde bereits die folgende Definition ausreichen:

$$\left(\frac{\left(3 - \frac{1}{x} \right)^2}{\left(\frac{4}{x} - 1 \right) x} \right)$$

```

1 \begingroup
2 \catcode'\active \xdef{\left\string{}}
3 \catcode'\active \xdef{\right\string{}}
4 \endgroup
5 \mathcode'("8000 \mathcode')="8000
6
7 \[ (\frac{(3-\frac{1}{x})^2}{(\frac{4}{x}-1)x}) \]
```

Innerhalb einer lokalen Gruppe werden die Klammern zuerst der Kategorie *active* (13) zugeordnet und dann durch `\xdef` global definiert, indem ihnen die jeweilige Links-Rechts-Kombination zugeordnet wird. Durch `\xdef`, was der Kombination aus `\edef` und `\gdef` entspricht, bleiben sie auch außerhalb der Gruppe gültig. Nach der Gruppe werden die beiden Klammern aus der allgemeinen Typendefinition für mathematische Zeichen herausgenommen und durch die Zuweisung des Wertes "8000" auch innerhalb des mathematischen Modus zu einem aktiven Zeichen erklärt.

Benutzt man eine der mathematischen Umgebungen aus dem Paket `amsmath`, dann kommt es zu Problemen, da die öffnende runde Klammer innerhalb des Makros `\resetMathstrut@` benutzt wird. Tauscht man diese gegen die öffnende eckige Klammer aus, ist das Problem gelöst. Alternativ hätte man auch die Zeichenkategorie von » (« lokal wieder herstellen können.

```

\makeatletter
\def\resetMathstrut@{%
  \setbox\z@\hbox{%
    \mathchardef\@tempa\mathcode'\[ \relax
    \def\@tempb##1"##2###3{\the\textfont"##3\char"}%
    \expandafter\@tempb\meaning\@tempa \relax
  }%
  \ht\Mathstrutbox@\ht\z@ \dp\Mathstrutbox@\dp\z@}
\makeatother
\begin{group}
\catcode'\active \xdef{\left\string{}}
```

```

\catcode'\active \xdef{\right\string}
\endgroup
\mathcode'{"8000 \mathcode'}="8000

\begin{align}
f(x) &= (\frac{3-\frac{1}{x}}{\frac{4}{x}-1}x)^2 - 1 \ x^2 \ \backslash
g(x) &= x(1+x(2+x(3+x)))
\end{align}

```

$$f(x) = \left(\frac{\left(3 - \frac{1}{x}\right)^2}{\left(\frac{4}{x} - 1\right)x} - 1 \right) x^2 \quad (1)$$

$$g(x) = x(1 + x(2 + x(3 + x))) \quad (2)$$

Dass es im letzten Beispiel anscheinend nicht klappt, ist nicht ein Problem mit der hier vorgenommenen Klammerdefinition, sondern mit der Parametersetzung innerhalb von \LaTeX , was sich jedoch korrigieren lässt [1, 2]:

```

\setlength\delimitershortfall{-1pt}
\begin{align}
f(x) &= (\frac{3-\frac{1}{x}}{\frac{4}{x}-1}x)^2 - 1 \ x^2 \ \backslash
g(x) &= x(1+x(2+x(3+x)))
\end{align}

```

$$f(x) = \left(\frac{\left(3 - \frac{1}{x}\right)^2}{\left(\frac{4}{x} - 1\right)x} - 1 \right) x^2 \quad (3)$$

$$g(x) = x(1 + x(2 + x(3 + x))) \quad (4)$$

Literatur

- [1] Herbert Voß: *Math mode*, Version 2.47., Dez. 2010, CTAN: info/math/voss/mathmode/Mathmode.pdf.
- [2] Herbert Voß: *Mathematiksatze mit \LaTeX* , DANTE e. V./lehmanns media, Heidelberg/Berlin, 2009.

detexify – Wie findet man ein Symbol?

Herbert Voß

Die Zusammenstellung [1] von Scott Pakin wird vielfach zitiert, wenn es um das Auffinden eines Symbols mit dem entsprechenden T_EX-Code geht. Wem das Durchsuchen dieser sehr umfangreichen Liste zu umständlich ist, der kann sich der Hilfe von <http://detexify.kirelabs.org> bedienen. Hier hat Daniel Kirsch ein grafisches Interface erstellt, welches das Suchen und Finden eines Symbols erleichtern soll.

ⓘ Please read: [Detexify needs help!](#)

Detexify² - LaTeX symbol classifier

What is this?

Anyone who works with LaTeX knows how time-consuming it can be to find a symbol in [symbols-a4.pdf](#) that you just can't memorize. Detexify is an attempt to simplify this search.

Draw here!

In das große Fenster kann mit der Maus das zu bestimmende Symbol gezeichnet werden. Zum Test wurde hier als Rechtshänder mit der linken Hand ein Symbol

gezeichnet, welches man in der Beschriftung zu Abbildung 1 auf Seite 24 findet. In dem Moment, wo die Maustaste losgelassen wird, beginnt im Hintergrund bereits die Suche nach einem entsprechenden Zeichen aus der Liste von [1]. Das Ergebnis wird rechts neben der Eingabe dargestellt.

Please read: [Detexify needs help!](#)

Detexify² - LaTeX symbol classifier

classify
symbols
blog



clear

What is this?

Anyone who works with LaTeX knows how time-consuming it can be to find a symbol in [symbols-ad.pdf](#) that you just can't memorize. Detexify is an attempt to simplify this search.

Draw here!

Share this help?

Hosting Detexify costs money and if it helps you may consider helping to pay the hosting bill.





Score: 0.168878210370481
 \backslash oplus
 mathmode



Score: 0.174907351824683
 \backslash bigoplus
 mathmode



Score: 0.175493830355009
 \backslash usepackage(stmaryrd)
 \backslash inplus
 mathmode



Score: 0.184430687884964
 \backslash usepackage(marvosym)
 \backslash Venus
 textmode



Score: 0.188892975719005
 \backslash usepackage(tipa)
 \backslash textbarrevglotstop
 textmode

Es ist offensichtlich, dass die Vorschläge in keiner Weise zutreffen, was in erster Linie an der Qualität der Eingabe liegt. Für solche Fälle kann man sich die gesamte Liste der möglichen Symbole anschauen, in der man relativ schnell das gesuchte Symbol findet: \backslash dj. Ein notwendiges Paket wird dabei ebenfalls mit angegeben, in diesem Fall fontenc mit der T1-Schriftkodierung. Die Anwendung kann dabei im Textmodus erfolgen: \backslash d.



Score: 0.203636521642652
 \backslash usepackage(wasysym)
 \backslash venus
 textmode & mathmode



Score: 0.20639715207198
 \backslash bigoplus
 mathmode



Score: 0.208116455470117
 \backslash usepackage[T1](fontenc)
 \backslash dj
 textmode



Score: 0.208303616412315
 \backslash oplus
 mathmode

Für den Fall, dass die Symbolbestimmung zu keinem Ergebnis führt, kann man sich die gesamte Liste in alphabetischer Reihenfolge ausgeben lassen. Über das Suchfenster ist es möglich, nach bestimmten Begriffen zu suchen, beispielsweise »diamond«, welches dann zur rechten Ausgabe führt:

Detexify² - Symbol table

[classify](#)
[symbols](#)
[blog](#)

Sort by command — Filter by

$ $	$ $ mathmode
<hr/>	
$!$	$!$ textmode
<hr/>	
$/$	$/$ mathmode
<hr/>	
$[$	$[$ mathmode
<hr/>	
$]$	$]$ mathmode
<hr/>	
$\ $	$\ $ mathmode

Detexify² - Symbol table

[classify](#)
[symbols](#)
[blog](#)

Sort by command — Filter by

\diagdown	\diagdown <code>\usepackage{ amssymb }</code> <code>\diagdown</code> mathmode
<hr/>	
\diagup	\diagup <code>\usepackage{ amssymb }</code> <code>\diagup</code> mathmode
<hr/>	
\oslash	\oslash <code>\usepackage{ wasysyn }</code> <code>\diameter</code> textmode & mathmode
<hr/>	
\diamond	\diamond <code>\diamond</code> mathmode
<hr/>	
\blacklozenge	\blacklozenge <code>\diamondsuit</code> mathmode
<hr/>	
\oslash	\oslash <code>\usepackage{ wasysyn }</code> <code>\invdiameter</code> textmode & mathmode

Die Ausgabe der Gesamtliste kann alternativ nach dem Paketnamen ausgegeben werden. Das folgende Beispiel zeigt die Ausgabe für das Stichwort »triangle«. Die Sortierung erfolgt aber jetzt alphabetisch nach dem Paketnamen, in diesem Fall `amssymb` (siehe Abbildung auf der nächsten Seite).

»Detexify« greift auf Elemente von HTML5 zurück und benötigt daher für die Darstellung einen aktuellen Browser, beispielsweise Firefox 3.5, Opera 9.6 oder Safari 4.

Literatur

- [1] Scott Pakin: *The Comprehensive L^AT_EX Symbol List*; CTAN:info/symbols/comprehensive/symbol-a4.pdf; 2008.

Detexify² - Symbol table

[classify](#) [symbols](#) [blog](#)

Sort by — Filter by



`\usepackage{ amssymb }`
`\blacktriangle`
 mathmode



`\usepackage{ amssymb }`
`\blacktriangledown`
 mathmode



`\usepackage{ amssymb }`
`\blacktriangleleft`
 mathmode



`\usepackage{ amssymb }`
`\blacktriangleright`
 mathmode



`\usepackage{ amssymb }`
`\triangleleft`
 mathmode



`\usepackage{ amssymb }`
`\trianglelefteq`
 mathmode

What is this?

This is the symbol table. Supported symbols are listed here (alphabetically).

Symbols can also be trained here. Just pick a symbol you sometimes need but tend to forget and click it. A canvas will open for your training input.

If there are symbols missing be sure to drop me a line (danishkirel@gmail.com). If you have more specific LaTeX questions though, you will have more success contacting your [TeX user group](#).

Von fremden Bühnen

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss vor. Die Updates können auf der *ctan-ann*-Mailingliste verfolgt werden, die auch über Twitter und Identi.ca als @ctanannounce verfügbar ist.

`bibleref-french` von *Maïeul Rouquette* ist die französische Übersetzung des Pakets `bibleref` zur Formatierung von Bibelzitatzen.

CTAN:macros/latex/contrib/bibleref-french

`randomwalk` von *Bruno Le Floch* dient zum Zeichnen von Zufallsbewegungen.

CTAN:macros/latex/contrib/randomwalk

`beamer2thesis` von *Fiandrino Claudio* ist ein Beamer-Thema für die Vorstellung der eigenen Abschlussarbeit.

CTAN:macros/contrib/beamer-contrib/themes/beamer2thesis

`easy-todo` von *Juan Rada-Vilela* erlaubt es, »Todo«-Notizen (über noch zu Erledigendes) in ein Dokument einzufügen und hiervon auch einen Index zu erstellen.

CTAN:macros/latex/contrib/easy-todo

`pcarl` von *Boris Veytsman* enthält die L^AT_EX-Unterstützung für die Schriftart *Adobe Caslon Open Face*.

CTAN:fonts/pcarl

`ocr-b-outline` von *Zdenek Wagner* enthält OCR-B-Fonts in den Formaten Type 1 und OpenType.

CTAN:fonts/ocr-b-outline

`aomart` von *Boris Veytsman* ist eine Klasse für Autoren der *Annals of Mathematics* (Princeton).

CTAN:macros/latex/contrib/aomart

`turnthepage` von *Luca Merciadri* erzeugt bei mehrseitigen Dokumenten einen Hinweis, die jeweils ebenfalls bedruckte Rückseite zu beachten.

CTAN:macros/latex/contrib/turnthepage

`cprotect` von *Bruno Le Floch* erlaubt Verbatim-Text in »zerbrechlichen« Befehlen.

CTAN:macros/latex/contrib/cprotect

`uothesis` von *Michael Anderson* ist eine Klasse für wissenschaftliche Arbeiten an der *University of Oregon*.

CTAN:macros/latex/contrib/uothesis

`spanglish` von *Luis Rivera* stellt eine vereinfachte Unterstützung der spanischen Sprache für `babel` bereit.

CTAN:language/spanish/babel/contrib/spanglish

`lato` von *Mohamed El Morabity* enthält die \LaTeX -Unterstützung für die serifenlose Schriftart Lato.

CTAN:fonts/lato

`droid` von *Mohamed El Morabity* bietet die \LaTeX -Unterstützung für die Schriftart Droid.

CTAN:fonts/droid

`lpic` von *Rostislav Matveev* dient dazu, Text über Grafiken zu setzen, die in ein Dokument eingebunden worden sind.

CTAN:macros/latex/contrib/lpic

`fldigial` von *D. P. Story* erzeugt in einer PDF-Datei eine Slideshow für PNG-, JPG- und GIF-Grafiken.

CTAN:macros/latex/contrib/fldigial

`ucharclasses` von *Michiel Kamermans* ist ein Paket für $X_{\text{e}}\LaTeX$, mit dem man zwischen Unicode-Blöcken hin und her wechseln kann (z. B. für mehrsprachige Dokumente).

CTAN:macros/xetex/latex/ucharclasses

`productbox` von *Gerd Neugebauer* erzeugt die Abbildung einer dreidimensionalen Verpackungsschachtel.

CTAN:macros/latex/contrib/gene/productbox

`comfortaa` von *Mohamed El Morabity* enthält die \LaTeX -Unterstützung für die Font-Familie Comfortaa.

CTAN:fonts/comfortaa

`nirma` von *Ashish Revar* enthält einen Style für wissenschaftliche Präsentationen mit dem Paket `beamer`.

CTAN:macros/latex/contrib/beamer-contrib/themes/nirma

`msuthesis` von *Alan Munn* ist eine \LaTeX -Klasse für wissenschaftliche Arbeiten der *Michigan State University*.

CTAN:macros/latex/contrib/msuthesis

`enumitem-zref` von *Florent Chervet* ermöglicht es, Referenzen auf die Gliederungspunkte einer Liste zu setzen, die mit dem Paket `enumitem` erzeugt wird.

CTAN:macros/latex/contrib/enumitem-zref

`threeddice` von *Daniel Luecking* erzeugt mit Hilfe von `METAPOST` Würfel, von denen eine, zwei oder drei Flächen zu sehen sind.

CTAN:graphics/metapost/contrib/macros/threeddice

`tucv` von *George Louthan* dient zum Setzen eines Lebenslaufs für eine Bewerbung.

CTAN:macros/latex/contrib/tucv

`fntproof` von *Daniel Luecking* implementiert sämtliche Befehle in Knuths `testfont.tex` so, dass alle notwendigen Eingaben als Argumente beim Aufruf übergeben werden

können.

CTAN:macros/generic/fntproof

`fileinfo` von *Uwe Lück* liest Metadaten von Dateien ein, ohne diese zu laden.

CTAN:macros/latex/contrib/fileinfo

`he-la-na` von *Zoran Filipovic* stellt Trennmuster und Abkürzungen von etwa 5000 serbischen Wörtern bereit, um den Textsatz im Vergleich zu `babel` zu verbessern.

CTAN:macros/latex/contrib/he-la-na

`dirtytalk` von *Moritz Klammler* stellt ein einfaches Makro `\say` bereit, mit dem man die Anführungszeichen bei Zitaten erzeugen kann. Schachtelungen bis zu zwei Ebenen werden unterstützt, und die Anführungszeichen können für jede Ebene in den Paketoptionen einzeln festgelegt werden.

CTAN:macros/latex/contrib/dirtytalk

`bbold-type1` von *Khaled Hosny* ist die Type-1-Version der Schriftart `Bbold`.

CTAN:fonts/bbold-type1

`acroterm` von *Jakob Voss* ist ein weiteres Paket zum Setzen von Akronymen.

CTAN:macros/latex/contrib/acroterm

`luasseq` von *Tilman Bauer* dient zum Zeichnen von Spektrallinien mit `LuaTeX`.

CTAN:macros/luatex/latex/luasseq

`tabu` von *Florent Chervet* ist ein Paket zum Erstellen von ein- und mehrseitigen tabular-Umgebungen.

CTAN:macros/latex/contrib/tabu

`hardwrap` von *Will Robertson* und *Kevin Godby* dient dazu, Text mit einer bestimmten Breite »hart« nach Wörtern (statt, wie `TeX`, nach Zeichen) zu umbrechen.

CTAN:macros/latex/contrib/hardwrap

`luaocode` von *Manuel Pégourié-Gonnard* vereinfacht die Ausführung von Lua-Code innerhalb von `TeX`.

CTAN:macros/luatex/latex/luaocode

`lua latex-doc` von *Manuel Pégourié-Gonnard* ist eine kurze Einführung zu `LuaTeX`.

CTAN:info/luatex/lua latex-doc

`ytableau` von *Ryan Reich* dient zum Zeichnen von Young-Tableaus und Young-Diagrammen.

CTAN:macros/latex/contrib/ytableau

`dfgproposal` von *Michael Kohlhase* ist eine `LATeX`-Klasse zum Schreiben von Anträgen für Fördermittel der Deutschen Forschungsgemeinschaft (mit Mustern).

CTAN:macros/latex/contrib/dfgproposal

`textgreek` von *Leonard Michlmayr* stellt aufrechte griechische Buchstaben als Textsymbole bereit.

CTAN:macros/latex/contrib/textgreek

`catchfilebetween tags` von *Florent Chervet* stellt ein Makro `\catchfilebetween tags` bereit, das aus einer externen Datei einen Teil eines Abschnitts ausliest, der zwischen den üblichen `Docstrip`-Tags steht. Eine Abwandlung des Pakets `catchfile`.

CTAN:macros/latex/contrib/catchfilebetween tags

`parselines` von *Florent Chervet* ist ein einfacher Zeilenparser für Text, der innerhalb einer Umgebung steht.

CTAN:macros/latex/contrib/parselines

`interfaces` von *Florent Chervet* ist eine Sammlung von Makros zu `pgfkeys`, um die Übergabe von Parametern an andere Pakete zu vereinfachen.

CTAN:macros/latex/contrib/interfaces

`koma-moderncvclassic` von *Salome Södergran* stellt den Stil von `moderncv` für KOMA-Script zur Verfügung. Kompatibel mit `biblatex`.

CTAN:macros/latex/contrib/koma-moderncvclassic

`rec-thy` von *Peter M. Gerdes* dient zum Setzen von Aufsätzen zur Rekursionstheorie (*recursion theory, computability theory*).

CTAN:macros/latex/contrib/rec-thy

`babeltools` von *Javier Bezos* behebt einige Fehler des Pakets `babel`.

CTAN:macros/latex/contrib/bezos

`russ` von *Sergey Shashkov* vereinfacht die Verwendung von kyrillischen Schriftzeichen, die direkt an `TEX`-Befehle übergeben werden können.

CTAN:macros/latex/contrib/russ

`cals` von *Oleg Parashchenko* dient zum Setzen von Tabellen, die sich über mehrere Seiten erstrecken.

CTAN:macros/latex/contrib/cals

TUGboat 31:3 – Inhaltsverzeichnis

General Delivery	158	From the president / Karl Berry
	158	Editorial comments / Barbara Beeton
		Matthew Carter named MacArthur Fellow;
		Indie Excellence Awards for self-published books;
		City maps made entirely of type U&L on line;
		Some »under-the-covers« uses of \TeX ;
		Beyond literate programming
	160	Hyphenation exception log / Barbara Beeton
Fonts	161	A story of kpfonts: Reaching the limits of NFSS
		/ Christophe Caignaert
Publishing	175	Giving it away / Jim Hefferon
	177	Glisterings: Meandering miniature books
		/ Peter Wilson
Software & Tools	184	Three things you can do with Lua \TeX that would
		be extremely painful otherwise / Paul Isambert
\LaTeX	191	Some misunderstood or unknown \LaTeX 2e tricks II
\LaTeX 3	194	\LaTeX 3 news, issue 3 / \LaTeX Project Team
	195	From <code>\newcommand</code> to <code>\DocumentNewCommand</code> with <code>xparse</code>
		/ Joseph Wright
ConTeXt	197	Tagged PDF in ConTeXt / Hans Hagen
	203	Introduction to colours in ConTeXt MKIV
		/ Luigi Scarso
Electronic Documents	208	Generate \TeX documents using pdfscript
		/ Oleg Parashchenko
	213	illumino: An XML document production system with
		a \TeX core / Matteo Centonza and Vito Piserchia
	219	Managing printed and online versions of large
		educational documents / Jean-Michel Hufflen
Problems	223	Aligning text in diagrams exported by Mathematica:
		A question about the PostScript infrastructure
		/ Michael Barnett
Hints & Tricks	227	The treasure chest / Karl Berry
Abstracts	229	Ars \TeX nica: Contents of issue 9 (October 2010)
	230	MAPS : Contents of issue 40 (2010)
	231	The Prac \TeX Journal : Contents of issue 2010-1
	233	Zpravodaj : Contents of issues 20(1–2), 20(3) (2010)
	236	Die \TeX nische Komödie: Contents of issue 2010/3
News	237	Calendar
Advertisements	238	\TeX consulting and production services
TUG Business	239	TUG institutional members
	240	TUG 2011 election

Bücher und Rezensionen

Das insbesondere für Anfänger geeignete Buch von Christine Detig, welches nicht mehr im regulären Buchhandel erhältlich ist, kann weiterhin über DANTE e. V. bestellt werden.

Christine Detig:

Der L^AT_EX-Wegweiser; 2. Auflage 2004,

mitp-Verlag, Bonn

268 Seiten; ISBN 3-8266-1414-3;

9 € für Mitglieder von DANTE e. V., versandkostenfrei.



Edition dante – Neuerscheinung

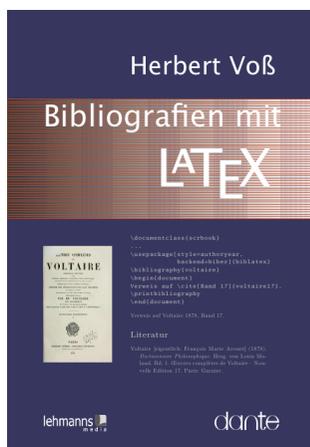
Herbert Voß:

Bibliografien mit L^AT_EX; 1. Auflage 2011,

DANTE e. V. und Lehmanns Media

240 Seiten; ISBN 978-3-86541-415-1;

14,95 € (Ladenpreis) bzw. 12,- € für Mitglieder von DANTE e. V., jeweils versandkostenfrei.



Bestellung

Bitte schicken Sie eine E-Mail an office@dante.de mit Angabe von *Name*, *Anschrift*, *Mitgliedsnummer* und *Anzahl der Exemplare*, und überweisen Sie den Betrag auf das Konto von DANTE e.V. oder bezahlen Sie per PayPal. Die Kontonummer finden Sie am Ende dieses Heftes und Informationen zu PayPal auf <http://www.dante.de/index/Intern/Zahlung.html>.

Bitte beachten Sie für Bestellungen bei DANTE e. V. folgende Informationen zum Widerrufsrecht: Verbraucher können bei Bestellungen per E-Mail, Internet, Brief oder Telefon den Kaufvertrag innerhalb einer Frist von 14 Tagen ab Erhalt der Ware per Brief, Fax oder E-Mail oder durch Rücksendung der Ware widerrufen (siehe Kontaktadresse). Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs oder der Ware. Bei einem Bestellwert bis 40,- € hat der Besteller die Rücksendekosten zu tragen. Bei Verschlechterung der Ware, die über die übliche Prüfung der Ware hinausgeht, hat der Besteller gegebenenfalls Wertersatz zu leisten.

Leserbriefe

Unicode-Zeichen in \LaTeX nutzen

Dominik Wagenführ

Nachtrag zum Artikel »Unicodezeichen in LaTeX nutzen« in »Die \TeX nische Komödie 2/2010

Nicht für jeden ist offensichtlich, wie man die Unicodezeichen, die man in \LaTeX direkt nutzen könnte, überhaupt mit einem Editor unter Linux eingibt. Auf Tastaturen mit normalem Qwertz-Layout kann man die Unicode-Zeichen leider nur eingeschränkt direkt eintippen. Eine kleine Auswahl an Sonderzeichen erhält man, wenn man $\text{AltGr}+\langle\text{BUCHSTABE}\rangle$ bzw. $\text{AltGr}+\text{Umschalt}+\langle\text{BUCHSTABE}\rangle$ drückt. Also beispielsweise $\text{AltGr}+S$ für ein f und $\text{AltGr}+\text{Umschalt}+S$ für ein β (Übersicht <http://wiki.ubuntuusers.de/Sonderzeichen#bersicht-der-Sonderzeichen>).

Daneben gibt es auch ein Feature in den meisten Desktop-Umgebungen bzw. Distributionen, um Unicode-Zeichen direkt zu tippen. Dazu drückt man $\text{Strg}+\text{Umschalt}+U$. Es erscheint dann ein unterstriches »u« auf dem Schirm. Dahinter gibt man den Unicode, beispielsweise »3b1«, ein. Nach einem weiteren Leerzeichen wird daraus das gewünschte Zeichen, im Beispiel ein α . Man muss dazu *nur* den Unicode der Zeichen wissen.

Komfortabler geht es natürlich mit einem anderen Tastaturlayout, wie mit Neo (freiesMagazin 05/2010, <http://www.freiesmagazin.de/freiesMagazin-2010-05>, Die \TeX nische Komödie 2/2010). Natürlich sind solche Alternativ-Layouts am Anfang anspruchsvoll zu lernen, angeblich tippt man damit aber schneller und hat mehr Möglichkeiten.

Spielplan

Termine

2011

- 19. 3. – 20. 3. **Chemnitzer LinuXTage 2011**
Technische Universität Chemnitz
09107 Chemnitz
<http://www.chemnitzer.linux-tage.de>

- 30. 3. – 1. 4. **DANTE 2011**
und 44. Mitgliederversammlung von DANTE e.V.
Universität Bremen
<http://www.dante.de/events/dante2011.html>

- 29. 4. – 3. 5. **EuroT_EX und 19. BachoT_EX-Konferenz 2011**
Bachotek, nahe Brodnica, Polen
<http://www.gust.org.pl/bachotex/bachotex2011-en>

- 11. 5. – 14. 5. **LinuXTag Berlin**
Messegelände
14055 Berlin
<http://www.linuxtag.org/2011/>

- 19. 9. – 24. 9. **5th International ConT_EXt meeting**
Porquerolles (Île de Porquerolles), Frankreich
<http://meeting.contextgarden.net/2011/>

- 1. 10. **Herbsttagung**
und 45. Mitgliederversammlung von DANTE e.V.
Garmisch-Partenkirchen
<http://www.dante.de/events/mv45.html>

- 19. 10. – 21. 10. **TUG 2011**
Trivandrum, Indien (Kerala)
<http://tug.org/tug2011/>

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-Anwendern statt, die für jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://projekte.dante.de/Stammtische/WebHome>.

Aachen

Torsten Bronger, bronger@physik.rwth-aachen.de
 Gaststätte Knossos, Templergraben 28, 52062 Aachen
 Zweiter Donnerstag im Monat, 19.00 Uhr

Berlin

Rolf Niepraschk, Tel.: (030) 3 48 13 16, Rolf.Niepraschk@gmx.de
 Gaststätte Gambirinus, Lilienstraße 133, 10115 Berlin Mitte
 Zweiter Donnerstag im Monat, 19.00 Uhr

Bremen

Winfried Neugebauer, Tel.: 0176 60 85 43 05, tex@wphn.de
 Wechselnder Ort, Erster Donnerstag im Monat, 18.30 Uhr

Darmstadt

Karlheinz Geyer, geyer.k.fv.tu@nds.tu-darmstadt.de, <http://www.da-tex.org>
 Wechselnder Ort, Erster Freitag im Monat, ab 19.30 Uhr

Dortmund

Martin Schröder, martin@oneiros.de, Tel.: (0231) 1 20 65 74
<https://www.xing.com/net/texdortmund>
 Wechselnder Ort, Zweiter Mittwoch im Monat, 18.30 Uhr

Erlangen

Walter Schmidt, Peter Seitz, w.a.schmidt@gmx.net
 Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen
 Dritter Dienstag im Monat, 19.00 Uhr

Hamburg

Lothar Fröhling, lothar@thefroehlings.de
 Zum Schwarzenberg, Schwarzenbergstraße 80, 21073 Hamburg
 Letzter Dienstag im Monat, 19.30 Uhr

Hannover

Mark Heisterkamp, heisterkamp@rrzn.uni-hannover.de
 Seminarraum RRZN, Schloßwender Straße 5, 30159 Hannover
 Zweiter Donnerstag im Monat, 18.30 Uhr

Heidelberg

Martin Wilhelm Leidig, Tel.: (06203) 40 22 03, moss@moss.in-berlin.de
 Anmeldeseite zur Mailingliste: <http://mailman.moss.in-berlin.de/mailman/listinfo/stammtisch-hd-moss.in-berlin.de>
 Wechselnder Ort
 Letzter Freitag im Monat, ab 19.30 Uhr

Karlsruhe

Klaus Braune, Tel.: (0721) 608-4 40 31, klaus.braune@kit.edu,
<http://projekte.dante.de/Stammtische/Karlsruhe>
 SCC (Steinbuch Centre for Computing) des KIT (vormals Universität Karlsruhe, Rechenzentrum),
 Zirkel 2, 2. OG, Raum 203, 76131 Karlsruhe
 Erster Donnerstag im Monat, 19.30 Uhr

Köln

Helmut Siegert

Institut für Kristallographie, Zülpicher Straße 49b, 50674 Köln

Letzter Dienstag im Monat, 19.30 Uhr

Konstanz

Kurt Lidwin, kurt.lidwin@web.de,

Restaurant Radieschen, Hohenhausgasse 1, 78462 Konstanz

Zweiter Donnerstag im Monat, 19.00 Uhr

München

Uwe Siart, uwe.siart@tum.de, <http://www.siart.de/typografie/stammtisch.xhtml>

Erste Woche des Monats an wechselnden Tagen, 19.00 Uhr

Stuttgart

Bernd Raichle, bernd.raichle@gmx.de

Bar e Ristorante »Valle«, Geschwister-Scholl-Straße 3, 70197 Stuttgart

Zweiter Dienstag im Monat, 19.30 Uhr

Trier

Martin Sievers, stammtisch-trier@texberatung.de

Fetzenkneipe (Haus Fetzenreich), Sichelstraße 36 (beim Sieh-Um-Dich), 54290 Trier

Dritter Montag des Monats, 20.15 Uhr

Wuppertal

Andreas Schrell, Tel.: (02193) 53 10 93, as@schrell.de

Restaurant Croatia »Haus Johannisberg«, Südstraße 10, 42103 Wuppertal

Zweiter Donnerstag im Monat, 19.30 Uhr

Würzburg

Bastian Hepp, LaTeX@sning.de

nach Vereinbarung

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

Tel.: (0 62 21)2 97 66 (Mo., Mi.–Fr., 10.00–12.00 Uhr)
Fax: (0 62 21)16 79 06
E-Mail: dante@dante.de

Konto: VR Bank Rhein-Neckar eG
BLZ 670 900 00
Kontonummer 2 310 007
IBAN DE67 6709 0000 0002 3100 07
SWIFT-BIC GENODE61MA2

Präsidium

Präsident: Volker RW Schaa president@dante.de
Vizepräsident: Adelheid Grob vice-president@dante.de
Schatzmeister: Klaus Höppner treasurer@dante.de
Schriftführer: Manfred Lotz secretary@dante.de
Beisitzer: Bernd Raichle
Martin Sievers
Herbert Voß
Uwe Ziegenhagen

Server

CTAN: <http://mirror.ctan.org/>
DANTE: <http://www.dante.de/>

FAQ

DTK: <http://projekte.dante.de/DTK/WebHome>
T_EX: <http://projekte.dante.de/DanteFAQ/WebHome>

Autoren/Organisatoren

Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[71]	Günter Rau Im Wengert 14 72127 Kusterdingen tagdigall@web.de	[17]
Adelheid Grob siehe Seite 82	[5]	Christine Römer	[7]
Kurt Lidwin Hardtstrasse 20 78464 Konstanz	[30]	Institut für germanistische Sprachwissenschaft FSU Jena	
Rolf Niepraschk Persiusstr. 12 10245 Berlin Rolf.Niepraschk@gmx.de	[55, 59, 61, 62]	Christine.Roemer@uni-jena.de	
Heiko Oberdiek Kroppenstück 9 77880 Sasbach heiko.oberdiek@gmail.com	[59, 61, 62]	Volker RW Schaa siehe Seite 82	[5]
Patrick Oßmann Hardtstraße 13 78467 Konstanz patrick.ossmann@htwg-konstanz.de	[22]	Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[3, 40, 59, 61, 62, 64, 66, 74, 76]
		Dominik Wagenführ dwagenfuhr@freiesmagazin.de	[78]

Die T_EXnische Komödie

23. Jahrgang Heft 1/2011 Februar 2011

Impressum

Editorial

Hinter der Bühne

5 Grußwort

Bretter, die die Welt bedeuten

7 Gewichten – Teil 2: Auszeichnungen

17 SageT_EX

22 Kyrillische Schriftzeichen im L^AT_EX-Dokument

30 Ein passendes Bewerbungsanschreiben zum ModernCV-Lebenslauf

40 Einlesen und Ausführen von Quellcode

Tipps und Tricks

55 bibl_atex – abgehakt!

59 Makro oder Umgebung?

61 enumerate und Referenzen

62 Sortierung im Index

65 Mathematik für »Tippfaule«

67 detexify – Wie findet man ein Symbol?

Von fremden Bühnen

71 Neue Pakete auf CTAN

75 TUGboat 31:3 – Inhaltsverzeichnis

Bücher und Rezensionen

76 Edition dante – Neuerscheinung

Leserbriefe

78 Unicode-Zeichen in L^AT_EX nutzen

Spielplan

79 Termine

80 Stammtische

Adressen

83 Autoren/Organisatoren