

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

24. Jahrgang Heft 1/2012 Februar 2012

1/2012

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 Autoren wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nicht gestattet. Alle Rechte zur weiteren Verwendung außerhalb von DANTE e.V. liegen bei den jeweiligen Autoren.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse dtk erstellt und per E-Mail oder Datenträger (CD/DVD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, L^AT_EX-Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <http://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit LuaTeX, Version beta-0.70.1-2011061410 (rev 4277) (T_EX Live 2011) erstellt. Als Standard-Schriften kamen T_EX Gyre Pagella, T_EX Gyre Heros, Bera Mono und Latin Modern Math 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: Andreas Eder Rudolf Herrmann Gert Ingold
Rolf Niepraschk Heiko Oberdiek Christine Römer
Gert Seidl Martin Sievers

Redaktionsschluss für Heft 2/2012: 15. April 2012

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

da LuaTeX von einigen L^AT_EX-Nutzern als »eierlegende Wollmilchsau« beschrieben wird, habe ich mich mal mit dem Setzen von mehrsprachigen Texten unter LuaL^AT_EX beschäftigt. Das Ergebnis ist insbesondere dann negativ, wenn man innerhalb eines Dokumentes verschiedene Schreibrichtungen hat. Da bleibt noch einiges an Arbeit für die LuaTeX-Entwickler übrig. Eine Alternative für CJK-Dokumente könnte das upT_EX-System sein, welches Norbert Preining für eine Anwendung innerhalb von T_EX Live aufbereitet hat. Es kann ebenfalls Schriften im Format OpenType verwenden.

Zum Thema Lua finden Sie auch eine Anwendung von Lua-Funktionen für verschiedene Zeit- und Datumsoperationen innerhalb eines L^AT_EX-Dokuments.

Agnieszka Okońska zeigt mit ihrem Beitrag, welche Stolpersteine man als Juristin aus dem Weg räumen muss, wenn man L^AT_EX für seine wissenschaftlichen Arbeiten nutzen will. Wer an der Herbsttagung in Trier teilgenommen hatte, wird sich vielleicht daran erinnern, dass Agnieszka schon dort über die besonderen Anforderungen an juristische Arbeiten berichtet hat. Den »Typografischen Punkt« beleuchtet Wolfgang Beinert, dessen Beitrag wir hier abdrucken, da er den geschichtlichen Hintergrund der verschiedenen Maßeinheiten für Punkte beschreibt. Rolf Niepraschk hat wieder einige »T_EXnische Tricks« beige-steuert.

Dennoch halten Sie ein Heft in der Hand, welches einen eher geringen Umfang hat; zum Redaktionsschluss lag kein einziges Manuskript zur Veröffentlichung vor. Da es primär nicht die Aufgabe der Redaktion ist, auch für den inhaltlichen Stoff zu sorgen, ergeht noch einmal die Bitte an *alle* Mitglieder, uns Beiträge rund um das Thema T_EX zuzusenden, damit wir auch im 24. Jahrgang unsere Zeitschrift regelmäßig erscheinen lassen können.

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,

dies ist nun mein letztes Grußwort als (stellvertretender) Vorsitzender, seit ich in Eichstätt 1998 in eine dieser Funktionen gewählt wurde. Es ist auch das letzte von mir in einer »Quotenbesetzung«, da Adelheid Grob aus beruflichen Gründen entschieden hat, sich nicht zur Wiederwahl zu stellen.

Ich habe in den letzten Tagen viel Zeit aufgewendet, um meine Unterlagen über DANTE e.V., T_EX und Co. aufzuräumen und Platz zu schaffen. Nicht etwa, weil ich in Zukunft kein T_EX mehr machen will, sondern vornehmlich, weil ich die Kisten für meinen bevorstehenden Umzug innerhalb von Darmstadt reduzieren möchte. Bei dieser Aktion sind mir interessante Pre-Prints, Unterlagen von Tagungen und Korrekturfahnen von T_EX-Büchern seit 1990 in die Hände gefallen. So habe ich in Kursunterlagen zu L^AT_EX geblättert (das beim Kurs noch nicht L^AT_EX 2_ε hieß, der Name wurde erst nach der TUG-Tagung in Aston 1992 festgelegt) und mich schweren Herzens von unzähligen Korrektorexemplaren und der Korrespondenz mit Helmut Kopka getrennt.

Allein die Unterlagen von TUG-, EuroT_EX- und DANTE-Tagungen füllten mehr als eine Bananenkiste. Ich habe allerdings mehr Zeit mit dem Lesen als dem Entsorgen zugebracht. Ein A0-Plakat von der Frühjahrstagung DANTE 1990 in Düsseldorf über mathematischen Satz und mit allen Mathematikzeichen gesetzt von Norbert Schwarz wurde *nicht* entsorgt! Es war damals mein ganzer Stolz, eines der wenigen Exemplare, die Norbert zur Tagung mitgebracht hatte, erbeutet zu haben.

Und ich muss sagen, dass ich fasziniert war, wieviele interessante Vorträge und Entwicklungen in T_EX und allen seinen Abkömmlingen es gegeben hat. Beherrschten anfangs mehr die Installation eines T_EX-Systems, Neuerungen bei Zeichensätzen (nur METAFONT!) und T_EX-Makropakete die Beiträge, so kamen später immer mehr die T_EX-Neuerungen (ConT_EXt, Ω, N_TS – New Typesetting System, dvipdf, pdfT_EX, u. a.) zum Tragen. Betrachtet man heute die aktuellen Standardsysteme wie T_EX Live oder MiK_TE_X mit den verfügbaren T_EX-Prozessoren, den

nahezu unzähligen Makropaketen und Schriften, vergisst man leicht die vielen »T_EXniker«, die dies alles geschaffen und auf Tagungen oft selbst darüber berichtet haben.

Ich hoffe deshalb, dass wir bei der Frühjahrstagung von DANTE e.V. in Leipzig wieder so faszinierende Beiträge und Berichte über Neuentwicklungen, Ratschläge und Tipps geboten bekommen, wie ich sie in den mehr als 20 Jahren Tagungsunterlagen bewundern konnte. Und vergesst bitte nicht, unsere Zeitschrift »Die T_EXnische Komödie« braucht auch unsere Beiträge!

Mit freundlichen Grüßen,

Ihr/Euer Volker RW Schaa (Vorsitzender)

*DANTE 2012
in Leipzig*

Bretter, die die Welt bedeuten

L^AT_EX für Juristen: Die Suche nach der perfekten Lösung aus Sicht einer Anwenderin

Agnieszka Okońska

Vor- und Nachteile der Verwendung von L^AT_EX durch Juristen

L^AT_EX wird vorwiegend für die Erstellung wissenschaftlicher Arbeiten in der Mathematik, den Natur- und den Wirtschaftswissenschaften verwendet. Sein Einsatz in anderen Bereichen gilt dagegen immer noch als Seltenheit. Dabei eröffnet ein solches System auch den Vertretern anderer Disziplinen große Vorteile, insbesondere den Juristen.

Neben seinen allgemeinen Stärken bietet L^AT_EX Studierenden und Promovierenden in den Rechtswissenschaften ganz besondere Vorzüge. Zunächst wird durch die Arbeit mit den Gliederungsbefehlen (`\part`, `\chapter`, `\section`, ...) eine genaue Auseinandersetzung mit dem Aufbau der eigenen Arbeit nicht nur unterstützt, sondern sogar erzwungen. Dies ist insofern sehr hilfreich, als diese Frage bei den Juristen viel mehr als eine bloße Formalie darstellt, da i. d. R. nur die korrekte Gliederung die Einhaltung der richtigen Struktur und des richtigen Inhalts der Arbeit erlaubt. Ferner unterstützt L^AT_EX (zusammen mit zusätzlichen Paketen) auch die juristische Zitierweise, wodurch einerseits erhebliche Zeitersparnisse (z. B. bei der Abschlussredaktion), andererseits die Beseitigung einer großen Fehlerquelle bewirkt werden.

Allerdings sind hier auch die Nachteile zu beachten. Neben der längeren Einarbeitungszeit sind auch die Probleme bei der Weiterverarbeitung nicht zu unterschätzen: Handelt es sich um eine Arbeit, die später nicht vom Verfasser selbst gedruckt, sondern extern verlegt wird (wie z. B. ein Aufsatz oder eine Dissertation), werden T_EX-Dateien selten und PDF-Dateien nicht immer angenommen.

Trotz dieser Probleme bietet der Einsatz von L^AT_EX den Juristen deutliche Vorteile, so dass nachfolgend die Möglichkeiten der Erstellung größerer und kleinerer

juristischer Arbeiten am Beispiel einer Dissertation und einer Hausarbeit vorgestellt werden. Dabei wird auf die Erfahrungen der Verfasserin aus ihrem eigenen Jura-Studium, der Arbeit an der Dissertation sowie aus den Versuchen, anderen Juristen L^AT_EX näher zu bringen, zurückgegriffen.

Besonderheiten juristischer Arbeiten

Zunächst sind die typographischen Besonderheiten juristischer Arbeiten zu nennen. Aus der Perspektive eines L^AT_EX-Profis können diese Anforderungen als relativ einfach eingestuft werden. Es werden i. d. R. keine Tabellen, Aufzählungen, Grafiken oder Schaubilder benutzt. Dafür sind aber die folgenden Eigenarten zu beachten:

1. Besonderheiten der Gliederung (Tiefe und Beschreibung),
2. Zitierweise,
3. Korrekturrand,
4. Inhaltsübersicht.

Ad 1.

Für den Aufbau einer juristischen Arbeit, insbesondere einer Dissertation, sind mehrere Gliederungsebenen erforderlich. Ferner gilt die als Standard in L^AT_EX festgelegte numerische Zählung der einzelnen Ebenen unter Juristen als unübersichtlich [5] (was bei mehreren Gliederungsebenen durchaus zutrifft) und wird daher nicht verwendet. Als üblich wird folgende alphanumerische Gliederung bezeichnet: 1., § 1, A., I., 1., a), aa), (1), (a) [5], wobei auch andere ähnliche Systeme verwendet werden.

Ad 2.

Eine weitere Besonderheit bildet die juristische Zitierweise. Entgegen den Gepflogenheiten in anderen Disziplinen befinden sich die Quellenangaben in den Fußnoten und nicht erst am Ende der Arbeit. Ferner handelt es sich hier um genaue Zitate, d. h. unter Angabe der einschlägigen Seite bzw. der Randziffer. Schließlich verfügen Juristen über mehrere Arten der zitierten Quellen, für die jeweils bestimmte Eigenarten zu beachten sind. Hier sind u. a. Aufsätze (in den nach Jahrgängen sowie nach Jahren zitierten Zeitschriften), Monographien, unveröffentlichte Dissertationen und Kommentare zu nennen. Bei den letztgenannten wird in der Fußnote neben dem Herausgeber oder dem Redakteur auch der Bearbeiter zitiert (z. B. Rauscher/*Pabst*, EuZPR Art. 3 EuVTVO Rz. 5), wobei im

Literaturverzeichnis nur der Kommentar als Gesamtwerk erscheint. Zudem werden einige Quellen mit, andere ohne Titel bzw. unter Nennung eines Kurztitels zitiert.

Für die konkrete Umsetzung stehen entsprechende Pakete zur Verfügung (siehe dazu den Punkt »Literatur« unten).

Ad 3.

Eine kleinere Eigenart juristischer Arbeiten ist der Korrekturrand von ca. 7 cm. Dieser ist jedoch nur im eigentlichen Teil der Arbeit, d. h. bei den Ausführungen des Verfassers, nicht aber bei der Gliederung oder dem Literaturverzeichnis einzuhalten.

Ad 4.

Bei Dissertationen ist es oft erforderlich, dem eigentlichen Inhaltsverzeichnis eine Inhaltsübersicht voranzustellen. Diese enthält nicht alle Gliederungspunkte, sondern lediglich die Teile (`\part`) und die Kapitel (`\chapter`).

Somit ist es für die Verwendung von L^AT_EX durch Juristen entscheidend, ob diese vier Besonderheiten ohne unverhältnismäßigen zusätzlichen Arbeitsaufwand umgesetzt werden können. Dies ist separat bezüglich kleinerer und größerer juristischer Arbeiten zu analysieren.

Juristische Dissertationen

Bei den juristischen Dissertationen handelt es sich um umfangreichere Arbeiten (ab etwa 200 Seiten). Für ihre Erstellung können sowohl »allgemeine« Klassen wie `book`, `scrbook` oder `memoir` als auch speziell für Juristen entwickelte Klassen genutzt werden.

Wird beispielsweise die Klasse `scrbook` aus der KOMA-Script-Sammlung verwendet, so müssen zwecks Einhaltung der Formalien einige Anpassungen vorgenommen werden: Die alphanumerische Gliederung kann mithilfe des Pakets `alnumsec` definiert werden, der wechselnde Korrekturrand ist dank `geometry` ohne Weiteres möglich und die Erzeugung einer Inhaltsübersicht stellt aufgrund der Erweiterung `shorttoc` ebenfalls kein Problem dar. Bestimmte Anforderungen können allerdings nicht so leicht umgesetzt werden. So sind die in dieser Klasse angebotenen sieben Gliederungsebenen oft nicht ausreichend. Weitere Feinheiten (z. B. Anpassungen des Fußnoten- und Kopfzeilenlayouts, der Kapitelnummerierung usw.) bleiben ebenso dem Anwender überlassen.

Unabhängig davon existiert die Klasse `jurabook`, die speziell für juristische Dissertationen und Bücher entwickelt wurde. Sie ist Teil der `juramisc`-Sammlung

von Axel Sodtalbers. [9] Zu ihren Vorteilen gehört vor allem die Unterstützung vieler Gliederungsebenen: Nach dem üblichen `\subsubsection` stehen zusätzliche Ebenen von `\subthreesection` bis `\subeighthsection` zur Verfügung. Die zugehörigen Gliederungspunkte werden entsprechend bezeichnet (Erster Teil, § 1, I., 1., a), aa), (1), (a), (aa), (aaa)). Ferner werden in dieser Klasse bestimmte Einstellungen als Standard definiert, die den juristischen Gewohnheiten entsprechen, wie z. B. die Fußnotenummerierung, die Kapitelnummerierung usw. Auch die Inhaltsübersicht kann mithilfe des Befehls `\overview` auf einfache Weise erzeugt werden. Ferner besteht die Möglichkeit, Vermerke am Seitenrand (z. B. bezüglich erforderlicher Ergänzungen) einzufügen (Befehl `\TODO`). Schließlich werden hier – wenn auch nur rudimentär – die Randziffern und die Indexierung unterstützt. Auf Basis dieser Klasse wurde von Daniel Maier eine Vorlage erstellt, welche den Anforderungen des Nomos-Verlags entspricht. [4] Somit kann diese Vorlage erst dann sinnvoll eingesetzt werden, wenn Nomos als Verlag feststeht – was i. d. R. erst am Ende des Erstellungsprozesses der Fall ist. In allen anderen Fällen wäre eine erneute Anpassung an die Anforderungen eines weiteren Verlags erforderlich.

Juristische Haus- und Seminararbeiten

Juristische Haus- und Seminararbeiten haben einen deutlich geringeren Umfang als Dissertationen (ca. 20–30 Seiten). Er ist oft vorgegeben, so dass in diesen Fällen platzsparendes Arbeiten erforderlich ist. Auch hierfür kann entweder eine allgemeine oder aber eine besonders angepasste Klasse genutzt werden. Wird beispielsweise die Klasse `scartc1` verwendet, sind dieselben Probleme wie bei den Dissertationen zu beachten.

Für die Erstellung (kleinerer) juristischer Arbeiten wurde von Felix Braun die Klasse `jura` entwickelt. [2] Ohne zusätzliche Pakete ermöglicht sie die Einstellung verschiedener Randbreiten, unterstützt mehrere Gliederungsebenen sowie die alphanumerische Gliederung (hier: A., I., 1., a), aa), α $\alpha\alpha$) (1) (a) (aa) (α) ($\alpha\alpha$)). Allerdings leidet diese Klasse an einem Kardinalfehler, der ihre Anwendung erheblich erschwert: Die Gliederungsebenen werden relativ angegeben. Mit dem Befehl `\toc` wird ein Gliederungspunkt auf der aktuellen Stufe eingefügt, mit `\levelup` eine Ebene höher und mit `\sub` eine Ebene tiefer. Dieses System ist kompliziert, fehleranfällig und vor allem bei längeren Texten unhandlich.

Auch `jurabook` kann für die juristischen Haus- und Seminararbeiten nicht problemlos eingesetzt werden. Aufgrund des geringeren Umfangs sind hier bestimmte Anpassungen (z. B. bezüglich der Gliederungsebenen o der zwecks »Platzsparens«) erforderlich.

Aufgrund der o. g. Probleme mit den existierenden Klassen wurde an der Universität Trier von Martin Sievers und der Verfasserin ein Versuch unternommen,

auf Basis von jurabook eine »absolut fertige« und »anfängersichere« Klasse für juristische Hausarbeiten unter dem Namen `hausarbeit-jura` zu erstellen. Ihre Vorteile umfassen z. B. eine »selbsterstellende« Titelseite, alle Gliederungsebenen aus jurabook, alphanumerische Gliederung (hier: A., I., 1., a), aa), (i), (1), (a), (aa), (aaa)), Einbindung und Anpassung des jurabib-Pakets (siehe dazu Punkt »Literaturarbeit«), einige platzsparende Einstellungen u. v. m. Die Klasse wurde im Rahmen einer zweistündigen L^AT_EX-Einführung Erstsemestern vorgestellt (vgl. dazu auch den Vortrag von Martin Sievers [8]). Eine erste Version ist mittlerweile veröffentlicht [6] und soll in Zukunft weiter ergänzt werden – sofern ein entsprechendes Interesse der Anwender da ist.

Literaturarbeit

Ein erheblicher Teil der Vorteile von L^AT_EX für Juristen bezieht sich auf die Vereinfachung und die Automatisierung der Literaturarbeit. Da die unter Juristen verbreitete Zitierweise jedoch stark von den üblichen Vorgaben anderer Disziplinen abweicht, sind hier besondere Pakete erforderlich.

Eine hervorragende Unterstützung der juristischen Zitierweise bietet das jurabib-Paket von Jens Berger. [1] Zunächst werden dort (fast) alle Besonderheiten und Varianten berücksichtigt. Ferner bestehen zahlreiche Anpassungsmöglichkeiten. Diese können komfortabel über Paketoptionen gesteuert werden, nur wenige Dinge müssen durch Änderungen an der Stildatei selbst umdefiniert werden. Nicht zu unterschätzen ist schließlich die erschöpfende und klare Dokumentation. Leider wird das Paket nicht mehr weiterentwickelt, so dass einige (kleine) Probleme (wie z. B. bezüglich der Umsetzung der Angabe *ders./dies.*), wohl keine Lösung innerhalb des Pakets finden werden.

Im Zusammenhang mit dem Makropaket biblatex wurde von Ben E. Hard biblatex-jura entwickelt. [3] Allerdings eignet sich dieses Paket nicht für weniger fortgeschrittene Anwender: Außer einer kleinen README-Datei existiert keine Dokumentation. Darüber hinaus orientiert sich biblatex-jura ausschließlich an den Vorgaben des Nomos-Verlags und verfügt darüber hinaus über keine Änderungsoptionen. Zudem gelang es der Verfasserin trotz zahlreicher Versuche nicht, das Paket fehlerfrei einzubinden.

Eine vielversprechendere Variante ist die Erweiterung biblatex-juradiss von Tobias Schwan. [7] Die Vorteile sind zunächst – wie bei biblatex-jura auch – die Nutzung von biblatex sowie die vielen (juristischen) Eintragsstypen, die der Quellenvielfalt entsprechen. Als Nachteil ist anzusehen, dass die einzige Möglichkeit der Anpassung der vom Autor gewählten Einstellungen (die m. E. nicht immer die »schönsten« sind) in der Änderung des Quellcodes besteht. Trotzdem ist der Einsatz von biblatex-juradiss sehr empfehlenswert, vor allem

für diejenigen Autoren, die mit der Erstellung ihrer Dokumente erst beginnen und mit L^AT_EX bereits ein wenig vertrauter sind.

Weitere besondere Pakete für Juristen

Neben den oben beschriebenen Paketen wie `alnumsec` existieren weitere, speziell für Juristen bestimmte Pakete. Von diesen werden nachfolgend zwei vorgestellt: `juraabbrev` und `jurarsp`.

Das Paket `juraabbrev` von Lucas Wartenburger [10] erlaubt die automatische Erstellung eines Abkürzungsverzeichnisses. Dafür werden die Abkürzungen und ihre Kennungen zunächst mit Hilfe des Befehls `\abbrev` innerhalb der Umgebung `abbreviations` definiert und können anschließend mit dem Befehl `\abb` (in verschiedenen Ausprägungen) im Text ausgegeben werden. Ein Verzeichnis aller verwendeten Abkürzungen wird automatisch erstellt. Dies erspart die mühsame Abschlussredaktion mit der bei Juristen erforderlichen Prüfung, ob alle benutzten Abkürzungen im Verzeichnis »entschlüsselt« sind und umgekehrt, ob alle dort angeführten Abkürzungen wirklich benutzt worden sind. Ein kleiner Nachteil dieses Pakets besteht darin, dass die Spalten des Verzeichnisses nicht automatisch an die Länge der Abkürzungen anpasst werden und daher manuell verbreitert werden müssen.

Sehr aufwendig ist auch die Verwaltung der Rechtsprechung im Rahmen einer – insbesondere größeren – juristischen Arbeit. Daher ist in diesem Bereich jede Automatisierung erwünscht. Diese wird durch die o. g. Erweiterung `biblatex-juradiss` angeboten, allerdings nur teilweise: Dort wird das Zitieren der Rechtsprechung in den Fußnoten automatisiert, indem für gerichtliche Entscheidungen ein spezieller B^BT_EX-Eintragstyp »jurisdiction« definiert wird. Allerdings leidet dieses Paket an einer großen Schwäche: Es besteht keine Möglichkeit, ein separates Rechtsprechungsverzeichnis zu erstellen.

Dies liefert jedoch ein weiteres Paket von Lucas Wartenburger: `jurarsp`. [11] Es erlaubt eine sehr fortgeschrittene Verwaltung der Rechtsprechung. Dazu werden eigene B^BT_EX-Eintragstypen definiert, u. a. »judgement« für die Rechtsprechung und »court« für die Gerichte. Für »judgement« sind 22 Eingabefelder vorgesehen, welche z. B. das Sortieren der Urteile nach Ländern, Gerichten, Gerichtsbarkeit usw. erlauben. Auch ein Rechtsprechungsverzeichnis wird erstellt. Insgesamt erleichtert dieses Paket erheblich die Arbeit, allerdings wurde seine Verwendung von der Verfasserin aus zwei Gründen eingestellt: Erstens wegen der Probleme mit Umlauten in den Namen der Gerichte, zweitens aufgrund der Unmöglichkeit (bzw. Unfähigkeit der Verfasserin) das Rechtsprechungsverzeichnis in Teile (z. B. nach Staaten) aufzuspalten, in denen der Tabelle mit den Urteilen eine kurze Einleitung (z. B. zu Fundstellen und Datenbanken) vorangestellt wird.

Zusammenfassung

Die Anwendung von L^AT_EX bietet auch Juristen erhebliche Vorteile. Für die juristischen Arbeiten existieren spezielle Klassen, die aufgrund der besseren Umsetzung typografischer Besonderheiten den allgemeinen Klassen vorzuziehen sind. Im Bezug auf größere Arbeiten (z. B. Dissertationen) besteht eine (fast) perfekte Lösung in Form der jurabook-Klasse. Die Situation bezüglich kleinerer Arbeiten ist weniger ideal, doch gibt es mit hausarbeit-jura einen vielversprechenden Ansatz, der aber mit Leben, d. h. Anforderungen, gefüllt werden muss. Es existieren zudem zahlreiche Pakete, welche die mühsame Arbeit mit der Literatur, der Rechtsprechung sowie die Verwendung von Abkürzungen erleichtern. Folglich ist L^AT_EX für Juristen bestens geeignet, so dass »nur noch« die Aufgabe bleibt, es ihnen näher zu bringen.

Literatur

- [1] Jens Berger: *jurabib*; CTAN:macros/latex/contrib/jurabib.
- [2] Felix Braun: *jura*; CTAN:macros/latex/contrib/jura.
- [3] Ben E. Hard: *biblatex-jura*; CTAN:macros/latex/contrib/biblatex-contrib/biblatex-jura.
- [4] Daniel Maier: *L^AT_EX-Paket für Nomos*; <http://www.dantown.de/latex>.
- [5] Thomas M. J. Möllers: *Juristische Arbeitstechnik und wissenschaftliches Arbeiten*; Franz Vahlen Verlag; München; 2. Aufl.; 2002.
- [6] Agnieszka Okońska und Martin Sievers: *hausarbeit-jura*; CTAN:macros/latex/contrib/hausarbeit-jura.
- [7] Tobias Schwann: *biblatex-juradiss*; CTAN:macros/latex/contrib/biblatex-contrib/biblatex-juradiss.
- [8] Martin Sievers: *Alles was Recht ist – ein »minimales« L^AT_EX für juristische Hausarbeiten an der Universität Trier*; Vortrag auf der Tagung DANTE 2011; <http://www.dante.de/events/dante2011/programm/vortraege/folienmsi.pdf>.
- [9] Axel Sodtalbers: *juramisc*; CTAN:macros/latex/contrib/juramisc.
- [10] Lucas Wartenburger: *juraabbrev*; CTAN:macros/latex/contrib/juraabbrev.
- [11] Lucas Wartenburger: *jurarsp*; CTAN:macros/latex/contrib/jurarsp.

Datumsfunktionen mit LuaTeX

Herbert Voß

Mit pdfL^AT_EX lässt sich das aktuelle Datum problemlos mit `\today` ausgeben. Für die Uhrzeit kann man auf das Paket `datetime` zurückgreifen. Hierbei wird allerdings die Ausgabe der Sekunden nicht unterstützt. Mit LuaTeX oder LuaL^AT_EX kann man die Datums- und Zeitroutinen von Lua verwenden, die zudem eine einfache Formatierung von Datum und Uhrzeit erlauben.

Makros und Umgebungen für LuaTeX

Der Befehl `\directlua` steht standardmäßig für LuaTeX zur Verfügung und kann auch im Mathematikmodus verwendet werden: $\pi \approx 3.1415926535898$

```
$\pi\approx\directlua{tex.print(math.pi)}$
```

Hierbei sind allerdings die T_EX-typischen Sonderzeichen zu beachten, wobei `_`, `^` & `{` und `}` problemlos möglich sind. Die drei Zeichen `\`, `~` und `#` können nicht ohne Maskierung mit dem vorangestellten Backslash beziehungsweise `\string` benutzt werden. Lediglich das Prozentzeichen kann nicht verwendet werden (siehe Tabelle 1).

Das Makro `\directlua` wird sofort ausgeführt und seine Ausgabe wird zusammen mit dem normalen Quelltext von T_EX verarbeitet. Daneben gibt es noch das Makro `\lualua`, welches sein Argument erst bei einem »shipout« bearbeitet und daher hier nicht weiter von Interesse ist.

Das Paket `luacode` von Manuel Pégourié-Gonnard unterstützt mit mehreren Makros und Umgebungen die Einbindung von Luacode in den T_EX-Code. [4] `\luaexec` ist prinzipiell ähnlich zu `\directlua`, jedoch können die Sonderzeichen einfacher benutzt werden: `\\` ergibt einen Doppelbackslash (benötigt in Lua-Strings), `%` das Prozentzeichen und `~` das normale `\nobreakspace`. Allerdings ist das Makro nicht voll expandierbar.

Die Umgebung `luacode` entspricht wiederum dem Makro `\luaexec`, wobei Sonderzeichen ohne jegliche Maskierung benutzt werden können, lediglich der einzelne Backslash ist als `\string\` einzugeben. Die Umgebung kann ebenso wie die Sternvariante `luacode*` an beliebiger Stelle im Dokument auftreten, wobei man sie in der Regel wohl in die Präambel des Dokumentes setzen wird. Mit `luacode*` kann auch der einzelne Backslash normal verwendet werden.

Tabelle 1: Zusammenstellung der durch das Paket `luacode` bereitgestellten Befehle und Umgebungen und die Behandlung von Sonderzeichen und Parametern.

	<code>\luadirect</code>	<code>\luaexec</code>	<code>luacode</code>	<code>luacode*</code>
Makros als Parameter	Ja	Ja	Ja	Nein
Backslash	<code>\string\</code>	<code>\string\</code>	<code>\string\</code>	<code>\</code>
Doppelbackslash	<code>\string\\</code>	<code>\\</code>	<code>\\</code>	<code>\\</code>
Tilde	<code>\string~</code>	<code>~</code>	<code>~</code>	<code>~</code>
#	<code>\string#</code>	<code>\#</code>	<code>#</code> oder <code>\#</code>	<code>#</code>
%	Nein	<code>\%</code>	<code>%</code> oder <code>\%</code>	<code>%</code>
\TeX -Kommentar	Ja	Ja	Nein	Nein
Lua-Kommentar	Nein	Nein	Ja	Ja

Ein Beispiel soll die in Tabelle 1 zusammengestellten Möglichkeiten für die Umgebung `luacode` verdeutlichen. Die Variablen `a` und `c` enthalten jeweils eine zusammengesetzte Zeichenkette, die die übergebenen Werte substituiert. Ein *einzelner* Backslash ist daher üblicherweise durch einen vorangestellten `z` zu maskieren, sodass ein Doppelbackslash dann durch `\\` dargestellt wird. Die anderen beiden Variablen verwenden die `[[. .]]`-Syntax, die keine Variablensubstitution erlaubt, andererseits aber nicht die Maskierung des Backslashes benötigt.

```

\newcommand\Verein{DANTE e.V.}   \newcommand\Ort{Heidelberg}
\begin{luacode}
function Sonderzeichen(T1,T2)
  local a = T1 .. "\\\\" .. T2 .. "\\par"  -- T1\\T2\par
  local b = [ [ T1\\T2 ] ]                -- "T1\\"T2"
  local c = T1 .. "\\newline " .. T2      -- T1\n T2
  local d = [ [ T1\string\n T2 ] ]         -- "T1"\n T2"
  tex.print(a,b,c,d)
end
\end{luacode}
\directlua{Sonderzeichen(\Verein,\Ort)}

```

DANTE e.V.

Heidelberg

T1

T2 DANTE e.V.

Heidelberg T1

T2

In der Sprache Lua werden Teilstrings durch die Angabe `» . . «` miteinander verbunden. `» - «` leitet einen Kommentar ein, der bis zum Zeilenende gilt.

Systemfunktionen

Funktionen, die Datum und Uhrzeit unterstützen, sind in der Regel Teil der Systembibliotheken, da sie direkt auf die zugrunde liegende Hardware zurückgreifen. In Lua ist dies die Bibliothek beziehungsweise in Lua-Notation die Tabelle `os`. Dabei interessieren hier nur die folgenden Funktionen:

```
os.clock()
os.date([Format [, Zeit]])
os.difftime(Zeit2, Zeit1)
os.setlocale(Locale [, Kategorie])
os.time([Tabelle])
```

`os.clock()`

Diese Funktion liefert die vergangene Zeit nach dem Start von Lua in Sekunden als Gleitkommazahl zurück und kann für Zeitmessungen benutzt werden, indem sie zweimal aufgerufen wird. Im folgenden Beispiel soll die willkürlich festgelegte Berechnung einer Summe durchgeführt werden:

$$S_k = \frac{1}{1^2} - \frac{3}{2^2} + \frac{6}{3^2} - \frac{10}{4^2} + \dots - \frac{k(k+1)}{k^2}$$

Dazu wird eine Lua-Funktion `AlternierendeSumme` definiert:

```
\begin{luacode*}
function AlternierendeSumme(Max) -- Maximalzahl
  local Start = os.clock()      -- Startzeit holen
  local Summe = 1               -- Startwert
  local k0 = 1                  -- Zähler
  for k=2, Max do               -- For-Schleife
    k0 = k0 + k                 -- Zählersumme
    if k % 2 == 0 then Summe = Summe - k0/(k*k)
    else Summe = Summe + k0/(k*k)
    end
  end
  local Stop = os.clock()       -- Endzeit holen
  tex.sprint(Summe .. " (" .. Stop-Start .. " sek)" )
end
\end{luacode*}
```

Über das Makro `\directlua`, oder alternativ `\luadirect`, kann die Funktion aus dem laufenden Quelltext heraus aufgerufen werden:

```
Die Summe ist für $k=1000001$ gleich \directlua{AlternierendeSumme(1000001)}.
In der Klammer steht die für die Berechnung benötigte Zeit.
```

Die Summe ist für $k = 1000001$ gleich 0.84657384027956 (0.12 sek). In der Klammer steht die für die Berechnung benötigte Zeit.

`os.date()`

Im einfachsten Fall liefert diese Funktion das Tagesdatum mit Uhrzeit in genormter Form unter Beachtung der jeweils eingestellten Sprache des Betriebssystems (Lokalisation): Wed Feb 15 20:53:23 2012

```
\directlua{tex.print(os.date())}
```

Dass diese Ausgabe hier in der englischen Variante erscheint, verwundert etwas, da die entsprechenden Variablen bereits korrekt für die deutsche Sprache gesetzt waren. Eine Änderung innerhalb von Lua \TeX ist mit der Funktion `os.setlocale` möglich. Als Parameter ist die Lokalisation, beispielsweise `de_DE` und optional die Kategorie anzugeben, wobei der Standardwert `all` nicht verwendet werden darf, da dies einen negativen Einfluss auf das Einlesen der Map-Dateien hat und zu einer fehlerhaften PDF-Ausgabe führt. Dieses Problem sollte in einer der nächsten Lua \TeX -Versionen behoben sein. Ein korrekter Aufruf wäre momentan `os.setlocale("de_DE", "time")`. Eine entsprechende Datumsfunktion wäre dann:

```
function Datum()
  os.setlocale("de_DE", "time")
  tex.print(os.date())
end
```

Ein entsprechender Aufruf mit

```
\directlua{Datum()}
```

liefert dann: Mi 15 Feb 2012 20:53:23 CET.

Für das Datum steht eine Großzahl an Parametern für die Ausgabeformatierung zur Verfügung, die alle in Tabelle 2 erklärt sind. Soll beispielsweise die aktuelle Uhrzeit in der Form »HH:MM:SS«, also Stunden, Minuten und Sekunden ausgegeben werden, so ist dies mit

```
\directlua{tex.print(os.date("%H:%M:%S"))}
```

möglich: Es ist gerade 20:53:23 Uhr.

In die allgemeine Datumsformatierung kann auch normaler Text eingefügt werden, wobei das Prozentzeichen auf Lua-Ebene durch sich selbst zu maskieren ist. Das Makro `\directlua` interpretiert dieses Zeichen allerdings als \TeX -Kommentarzeichen, sodass es in der \TeX -üblichen Notation durch einen vorangestellten Backslash zu maskieren ist.

```
\directlua{tex.print(os.date("Heute ist ein \%A im Monat \%B"))}
```

führt zu: Heute ist ein Mittwoch im Monat Februar.

Tabelle 2: Zusammenstellung der Formatierungsparameter für das Datum.

<i>Feld</i>	<i>Bedeutung</i>
%a	Wochentag in Kurzform (z. B. Mon)
%A	Wochentag in Langform (z. B. Freitag)
%b	Monat in Kurzform (z. B. Sep)
%B	Wochentag in Langform (z. B. Februar)
%c	Datum und Uhrzeit (z. B. 09/16/98 23:48:10)
%d	Tagesnummer im Monat [01-31]
%H	Stunde [00-23]
%I	Stunde (am – pm) [01-12]
%M	Minute [00-59]
%m	Monat [01-12]
%p	»am« oder »pm«
%S	Sekunde [00-61]
%w	Wochentag (Numerisch) [0-6=Sonntag-Samstag]
%x	Datum (z. B. 09/16/98)
%X	Zeit (z. B. 23:48:10)
%Y	Jahr, vierstellig (2012)
%y	Jahr, zweistellig (12) [00-99]
%%	Das Prozentzeichen %

Für Unix-ähnliche Betriebssysteme wird die Zeitrechnung formal mit dem 1. 1. 1970 00:00 UTC begonnen. Da Berlin eine Stunde früher liegt, muss die Differenz zu diesem Datum –3600 Sekunden betragen. Die Funktion `date` kann auch dazu benutzt werden, um eine derartige Angabe in Sekunden in eine formatierte Datumsangabe zu wandeln. Die Sekundenzahl wird dabei als zweiter Parameter übergeben:

```
\directlua{tex.print(os.date("%c", -3600))}
```

Die Ausgabe muss jetzt ebenfalls der »Geburtsstunde« der Unix-Zeit entsprechen, denn –3600 Sekunden zu UTC addiert ergibt die CET (Mitteleuropäische Zeit): Do 01 Jan 1970 00:00:00 CET. Sollen die einzelnen Datumsangaben weiterverarbeitet werden, so kann der Formatierungsparameter `*t` benutzt werden, der eine Lua-Tabelle mit den Elementen `year`, `month`, `day`, `yday` (Tagesnummer im Jahr), `wday` (Wochentag), `hour`, `min`, `sec` und `isdst` (*daylight saving time flag* – Sommerzeit) zurückgibt. Dabei handelt es sich bis auf den letzten Wert um numerische Angaben.

Das letzte Feld des Datums ist vom Typ Boolean, daher muss man die Funktion `tostring` verwenden, die allerdings auf alle Elemente angewendet werden kann, sodass keine Fallunterscheidung notwendig ist.

```
function DatumsFelder(Dformat)
  Date = {}
  Date = os.date(Dformat)
  for k,v in pairs(Date) do
    tex.print("\makebox[4em][l]{ " .. k .. " } " ..
      tostring(v) .. "\\\[-2pt]")
  end
end
```

Anwendung von `\directlua{DatumsFelder("!*t")}` führt dann zu:

```
hour  20
min    53
yday   4
day    15
month  2
year   2012
sec    23
yday   46
isdst  false
```

Soll die Ausgabe auf UTC statt CET bezogen werden, so wird dem Formatierungsparameter ein Ausrufezeichen vorangestellt: `!*t`. Als Ausgabe ergibt sich dann mit `\directlua{DatumsFelder("!*t")}`:

```
hour  19
min    53
yday   4
day    15
month  2
year   2012
sec    23
yday   46
isdst  false
```

`os.difftime()`

Die Funktion `difftime` erstellt aus zwei gegebenen Datumsangaben in Sekunden die Differenz, die dann entsprechend in Tage umgerechnet werden kann. Möchte man beispielsweise die Anzahl an Tagen zwischen zwei gegebenen Daten ermitteln, so ist dies mit der folgenden Lua-Funktion möglich:

```
function anzahlTage(d2, d1)
  local Diff = os.difftime(d2, d1)
  tex.print(math.floor(Diff/86400))
  -- 24 h/Tag  60 min/h  60 s/min = 86400 s/Tag
end
```

Eine Anwendung könnte dann beispielsweise wie folgt sein: DANTE e. V. besteht zur Zeit genau 8342 Tage.

```
\directlua{anzahlTage(os.time{year=\the\year,month=\the\month,
day=\the\day}, os.time{year=1989, month=4, day=14})}
```

`os.time()`

Die bereits angesprochene Unix-Zeit kann mit dieser Funktion abgefragt werden. Ohne Parameter wird daher die seit dem 1. 1. 1970 bis heute vergangene Zeit angegeben: 1329335603 Sekunden.

```
\directlua{tex.print(os.time() .. ' Sekunden')}
```

Die Angabe bezieht sich allerdings auf UTC. Als Parameter kann eine Lua-Datums-tabelle angegeben werden, die mindestens die Angaben zu Jahr, Monat und Tag enthalten muss. Ein entsprechendes Beispiel wurde bereits im vorhergehenden Abschnitt gezeigt.

Literatur und Software

- [1] Roberto Ierusalimschy: *Programmieren mit Lua*; Open Source Press; München; 2006.
- [2] Roberto Ierusalimschy, Luiz Hernandez de Figueiredo und Waldemar Celes: *Reference Manual 5.1*; lua.org; Rio de Janeiro, Brasilien; 2006.
- [3] *The Lua documentation*; <http://www.lua.org/docs.html>; Jan. 2012.
- [4] Manuel Pégourié-Gonnard: *The luacode package*; CTAN:macros/luatex/latex/luacode/; Dez. 2011.
- [5] LuaT_EX Development Team: *LuaT_EX Reference*; CTAN:systems/luatex/base/manual/luatexref-t.pdf; Mai 2011.

Multilinguale Texte mit Lua \TeX – ein Versuch

Herbert Voß

Die \TeX -Variante Ω (Omega) wurde seinerzeit besonders von Yannis Haralambous entwickelt, um die Beschränkungen von \TeX zu überwinden und einen 16-Bit-Zeichensatz zu ermöglichen. Der Nachfolger von Ω war dann \aleph (Aleph), welches ein erster Schritt auf dem Weg zur allgemeinen Anwendung von Unicode war und mittlerweile nicht mehr weiterentwickelt wird, da mit Lua \TeX ein Nachfolger vorhanden ist, der das Setzen von multilingualen Texten ermöglicht, wie es bislang nicht möglich war. In diesem Artikel werden diesbezüglich erste Erfahrungen mit Lua \TeX vorgestellt.

Einführung

Eine einfache Variante von multilingualen Texten ergibt sich, wenn die Schreibrichtung von links nach rechts und von oben nach unten beibehalten werden kann. In diesen Fällen braucht man zum einen nur die Sprache zu wechseln, um die aktuellen Trennmuster zu laden und zum anderen nur eine entsprechende Schrift zu aktivieren, die auch die Zeichen darstellen kann.

Die im Folgenden verwendeten Beispieltexthe beanspruchen keine grammatikalische und orthografische Korrektheit; sie sind lediglich das Ergebnis des Übersetzungsprogramms von Google, angewendet auf den vorstehenden Absatz. Für Griechisch erhält man:

Μια απλή παραλλαγή των πολύγλωσσων κειμένων που εμφανίζεται όταν η κατεύθυνση γράφουν είναι από αριστερά προς τα δεξιά και από πάνω προς τα κάτω διατηρείται. Σε αυτές τις περιπτώσεις κάποιος πρέπει να μόνο μία γλώσσα για να κατεβάσετε το τρέχον σχέδιο απελευθέρωσης και για να ενεργοποιήσετε μόνο το κατάλληλο σημείο από την άλλη Γραφές, που είναι και οι χαρακτήρες μπορεί.

Nehmen wir jetzt allerdings die hebräische Variante, so müssen wir die Schreibrichtung auf »von rechts nach links« ändern:

גרסה פשוטה של טקסטים לשונית מתרחשת כאשר הכיוון לכתוב הוא משמאל לימין ומלמעלה למטה נשמר. במקרים אלה אחד צריך שפה אחת בלבד על מנת להוריד את התבנית לשחרר הנוכחי כדי להפעיל רק את הנקודה המתאימה על כתיבי אחרים, שגם הם הדמויות יכול.

Die traditionelle japanische Schreibweise ist von oben nach unten und dann von rechts nach links. Aus technischen Gründen muss der Text hier in eine `minipage` gesetzt werden, da er ansonsten nur in einer Zeile am rechten Rand der Seite dargestellt wird, da die grundsätzliche Schreibrichtung durch die Dokumentenklasse `dtk` vorgegeben ist. Der folgende Text ist hier in der gleichen Weise eingefügt worden wie der deutsche Text, es handelt sich also nicht um einen als Grafik eingebunden Text eines externen Dokumentes.

時書き込み方 向多言語テ
キストの単 純な変形 が発生
左から右へ、上から下へ
は維持され ます。これら
のケースで 一つは現在の リリ
スのパター ンをダウンロ
ードするため のみつの言
語が必要と また、文字
である 他の聖書 に対する唯
一の適切なポイント を、有効に
することが できます。

Der Text musste hier zwecks Umbruchmöglichkeit mit Worttrennern eingegeben werden, was eigentlich für japanische Texte nicht unbedingt üblich ist (vergleiche Abb. 1b). Durch die zunehmende Globalisierung findet man japanische Texte verstärkt auch in anderen Varianten, wie nur von rechts nach links:

、へ右らか左 生発が 形変な純 単のトスキ テ 語言多向 方み込書時
ーリリ の在現はつー でスーケの られこ。すま れさ持 維は へ下らか上
、たま と要必が語 言のつみの にめたるすド ーロンウダをン ータパのス
るすに効有、をト ンイポな切 適の一 唯るす対に 書聖の他 るあで 字文
。すまきで がとこるす

Oder sogar in der für uns normalen Anordnung, sodass dann auch nichts weiter zu beachten ist:

時書き込み方 向多言語テ キストの単 純な変形 が発生 左から右へ、
上から下へ は維持され ます。これら のケースで 一つは現在の リリ
スのパター ンをダウンロ ードするため のみつの言 語が必要と また、
文字 である 他の聖書 に対する唯 一の適切なポイント を、有効にする
することが できます。

»Die \TeX nische Komödie« wird normalerweise mit der Schrift Pagella aus dem Projekt \TeX -Gyre gesetzt. Damit lassen sich zwar die obigen griechischen Zeichen, nicht jedoch die hebräischen oder japanischen Zeichen darstellen, da sie einfach nicht in der Schrift enthalten sind. Es sind dann jeweils andere Schriften zu laden, worauf noch auf Seite 23 eingegangen wird.

Schreibrichtungen

Die obigen Beispiele zeigen bereits, wie vielfältig die Anforderungen an Schriften sind. Das Programm \mathfrak{N} ermöglichte 16 verschiedene Schreibrichtungen, die formal unterteilt werden in

- Richtung der Absätze;
- Zeilenbeginn;
- Richtung der Zeichen.

Für jede dieser Richtungen kann einer der Buchstaben L für links, R für rechts oder T für oben (Top) vergeben werden. Unsere traditionelle Schreibweise entspricht dabei TLT; Absätze folgen von oben nach unten, Zeilen werden von links nach rechts gefüllt und die Zeichen stehen aufrecht, zeigen also nach oben (T).

Somit ergeben sich beispielsweise folgende Zuordnungen, die den von Lua^AT_EX unterstützten entsprechen:

- TLT Die traditionelle europäische (lateinische) Schreibweise.
 TRT Die arabische Schreibweise, mit dem einzigen Unterschied, dass von rechts nach links geschrieben wird.
 RTT Die traditionelle CJK-Schreibweise (Chinesisch, Japanisch, Koreanisch), Absätze von rechts nach links, Zeilen von oben nach unten und Zeichen aufrecht.
 LTL Mongolische Schreibweise, Absätze von links nach rechts, Zeilen von oben nach unten und Zeichen um 90 Grad nach links gedreht, was insgesamt der arabischen Schreibweise entspricht, wenn man alles um 90 Grad dreht.

\mathfrak{N} kennt noch weitere 12 Zuordnungen, die in Lua^AT_EX zur Zeit jedoch zu einem Fehler führen.

Die Umschaltung muss, beziehungsweise kann, individuell für die Seiten-, die Textkörper-, Absatz- und Textausrichtung vorgenommen werden, wobei den zugehörigen Makros `\luatexpagedir`, `\luatexbodydir`, `\luatexpardir` und `\luatextextdir` eine der aus drei Buchstaben beschriebenen Schreibrichtungen zugewiesen wird. Dabei kann man zur Vereinfachung entsprechende Makros definieren, beispielsweise

```
\newcommand\setArabic{%
  \luatexpagedir TRT \luatexbodydir TRT
  \luatexpardir TRT \luatextextdir TRT}
\newcommand\setLatin{%
  \luatexpagedir TLT \luatexbodydir TLT
  \luatexpardir TLT \luatextextdir TLT}
\newcommand\setCJK{%
  \luatexpagedir RTT \luatexbodydir RTT
  \luatexpardir RTT \luatextextdir RTT}
```

```
\newcommand\setMongolian{%
\lualatexpagedir LTL \lualatexbodydir LTL
\lualatexpardir LTL \lualatextextdir LTL}
```

Schriften

Theoretisch liegt Unicode die Idee zugrunde, dass eine Schriftfamilie grundsätzlich »alle Zeichen aller Sprachen« enthält. Es dürfte allgemein bekannt sein, dass wir davon noch sehr weit entfernt sind. Wie bereits erwähnt, verfügt die hier verwendete Schrift Pagella nur über einen sehr kleinen Teil an Zeichen. Soll beispielsweise ein arabischer Text ausgegeben werden, so muss neben der Schreibrichtung auch noch die Schrift gewechselt werden, was einen negativen Einfluss auf das Ganze haben kann, wenn beispielsweise die Liniendicke der arabischen Schriftzeichen überhaupt nicht zu denen der Pagella passt. Dies ist bereits oben im Zusammenhang mit dem hebräischen und japanischen Beispiel zu sehen.

Für das folgende Beispiel einer arabischen Schrift wird wieder auf die Schrift Code2000 gewechselt. [1]

```
\begingroup
\setmainfont[Scale=0.85,Script=Arabic]{CODE2000}
\setArabic
...
\endgroup
```

کثیراً یکی از شهرهای بزرگ ایران و مرکز استان آذربایجان شرقی است. این شهر بزرگتری ن شهر منطقه شمال غرب ایران بوده و قطب اداری، ارتباطی، بازرگانی، سیاسی، صنعتی، فرهنگی و نظامی این منطقه شناخته میشود. تبریز به دلیل جای دادن بسیاری از کارخانهای مادر و بزرگ صنعتی در خود و نیز وجود بیش از ۰۰۶ شرکت قطبها و در آن، دومین شهر آلوده و نیز دومین شهر صنعتی ایران پس از تهران بهشمار میرود و به دلیل صنعتی بودن، یکی از مهمترین شهرهای مهاجرپذیر ایران محسوب میشود.

Für Code-Beispiele wird standardmäßig das Paket listings verwendet, welches jedoch keine Zeichen im Format UTF-8 verarbeiten kann, womit auch die arabischen Zeichen nicht darstellbar sind. Das Paket listingsutf8 von Heiko Oberdiek hebt diese Beschränkung teilweise auf, jedoch nur für das Einlesen externer Dateien. Dazu müsste man jedoch erst den arabischen Text aus dem Dokument in eine Datei schreiben. Dies kann mit dem Paket fancyvrb geschehen, welches sich beim Schreiben von Dateien nicht sonderlich für die Kodierung der Zeichen interessiert. Die Umgebung VerbatimOut kann daher in der gewohnten Weise angewendet werden:

```
\begin{VerbatimOut}{\jobname.arab}
...
\end{VerbatimOut}
```

Diese Datei kann ganz normal über `\input` eingelesen werden, wobei hier nur ein verkürztes Textbeispiel genommen wird:

```
\begingroup
\setmainfont[Scale=0.85,Script=Arabic]{CODE2000}
\setArabic
\input{\jobname.arab}
\endgroup
```

تبریز یکی از شهرهای بزرگ ایران و مرکز استان آذربایجان شرقی است. این شهر بزرگتری ن شهر منطقه شمال غرب ایران بوده و قطب اداری، ارتباطی، بازرگانی، سیاسی، صنعتی، فرهنگی و نظامی این منطقه شناخته میشود. تبریز به دلیل جای دادن

Wird sie aber mit `\lstinputlisting` eingelesen, dann ergibt sich unter der Voraussetzung, dass als `basicstyle` die Standardschrift genommen wird (`\rmfamily`) auch der Inhalt der externen Datei in verbatim Darstellung:

```
\begingroup
\setmainfont[Scale=0.85,Script=Arabic]{CODE2000}
%\setArabic
\lstinputlisting[basicstyle=\small\rmfamily,breakatwhitespace]{\jobname.
  arab}
\endgroup
```

تبریز
 یاهرهش زا یک
 نیرتگزیه رهش نیا. تتسا یقرش ناجیابدآ ناتسا زکرم و ناریا گرزیه
 یسلیس، یناگزیه، یطلبترا، یزادا بطق و هوب ناریا بره لامش مقطنه رهش
 ندلیاج لیلده هب زبیرت. دوشیه متخلنش مقطنه نیا یعاطه و یکنهرف یتمدن

Wird `\setArabic` aktiviert, so kann auch für Codedarstellungen die Schreibrichtung beachtet werden:

تبریز
 یکی از شهرهای
 بزرگ ایران و مرکز استان آذربایجان شرقی است. این شهر بزرگترین
 شهر منطقه شمال غرب ایران بوده و قطب اداری، ارتباطی، بازرگانی، سیاسی،
 صنعتی، فرهنگی و نظامی این منطقه شناخته میشود. تبریز به دلیل جای دادن

Gleiches ist auch mit einer der CJK-Sprachen möglich: Rauschreiben, wieder einlesen und als Code darstellen, wobei in diesem Fall für `basicstyle` *nicht* auf `\rmfamily` gewechselt werden darf:

```
\begingroup
\fontspec{Guseul}
\begin{VerbatimOut}{\jobname.cjk}
...
\end{VerbatimOut}
```

```
\setCJK
\minipage[b][77mm][t]{0.35\linewidth}
\input{\jobname.cjk}\par\bigskip
\lstinputlisting[basicstyle=\small]{\jobname.cjk}
\endminipage
\endgroup
```

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100.

Trennmuster

Man kann davon ausgehen, dass für das Programm pdf \TeX mit dem Paket babel ein wesentlicher Teil der verbreiteten Weltsprachen mit Trennungen durch vorhandene Trennmuster gesetzt werden kann. babel lässt sich zwar auch mit Lua \TeX verwenden, man muss jedoch mit Problemen rechnen, sobald eine Sprache benutzt wird, für die babel viele interne Definitionen vornimmt, beispielsweise für Griechisch.

Es muss also zur Zeit in jedem Einzelfall geprüft werden, ob Lua \TeX und babel für eine gewählte Sprache korrekt funktionieren.

Probleme

Lua \TeX ist noch weit davon entfernt, den gleichen zuverlässigen Textsatz liefern zu können, wie er für \mathbb{N} oder up \TeX unter \TeX Live [4] selbstverständlich ist. Setzt man beispielsweise den ersten japanischen Text mit \fbox in eine Rahmenbox, so ergeben sich für Lua \TeX und up \TeX die in Abbildung 1 dargestellten Ausgaben.

Für beide wurde der Quelltext von Seite 21 verwendet. Benutzt man für die Rahmenbox \frame, die den Boxinhalt grundsätzlich an der Basislinie ausrichtet, oder ändert man die Definition der Minipage zu \fbox{\minipage[t][77mm][t]{...}, so ergibt sich für up \TeX das erwartete Ergebnis, wohingegen der Text für Lua \TeX

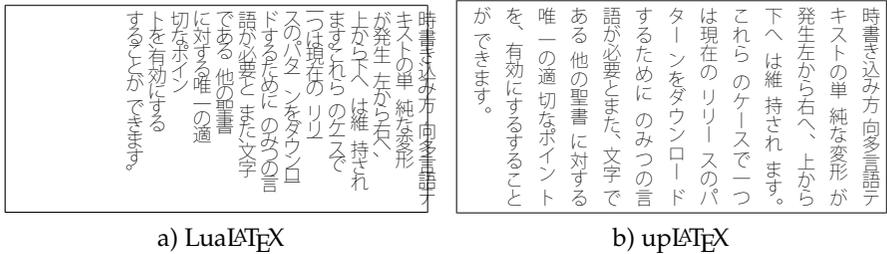


Abbildung 1: Ausgabe einer gerahmten Minipage mit japanischem Text.

teilweise oder völlig außerhalb der Box erscheint. Beim Umschalten in Lua \TeX auf die normale Textausrichtung gibt es außerdem zusätzliche Probleme mit dem vertikalen Abstand.

Texteingabe

Dieser Beitrag wurde mit dem Editor \TeX maker erstellt, der Zeichen im Unicode darstellen kann, sodass die Textbeispiele ohne jegliche speziellen Tastatortreiber einfach aus anderen Dokumenten oder Webseiten kopiert wurden. Für die meisten Sprachen mit einer anderen Textausrichtung wird man auch automatisch spezielle Eingabemöglichkeiten vorsehen müssen, um beispielsweise für Japanisch Kanjis (漢字), Hiragana (ひらがな) und/oder Katakana (カタカナ) über die Tastatur eingeben zu können.

Literatur und Software

- [1] James Kass: *Code2000*; <http://sourceforge.net/projects/code2000/>; 2008.
- [2] A Lee: *ttf-alee - A Lee's free Hangul truetype fonts*; http://pkgs.org/ubuntu-10.10/ubuntu-main-i386/ttf-alee_12_all.deb.html; 2008.
- [3] Lua \TeX Development Team: *Lua \TeX Reference*; CTAN:systems/luatex/base/manual/luatexref-t.pdf; Mai 2011.
- [4] Nobuyuki Tsuchimura: *tlptexlive*; <http://tutimura.ath.cx/ptexlive/?tlptexlive%A5%EA%A5DD%A5B8%A5C8%A5EA>; 2012.

Typographischer Punkt – Point typographique, Didot-Point, Pica Point, PostScript Point¹

Wolfgang Beinert

Der typografische Punkt, ursprünglich als »Point typographique« bezeichnet, ist Teil eines typographischen Maßsystems, der im gewerbespezifischen Sprachschatz auch einfach als »Punkt« abgekürzt wird. Der typographische Punkt ist die kleinste Einheit eines typographischen Punkt-Systems, das Mitte des 18. Jahrhunderts in Frankreich aus einem unmetrischen und duodezimalen Maßsystem zur einheitlichen Größenbestimmung von Buchstaben bzw. Schriftgraden und Abständen hervorgegangen ist. Dieser typographische Punkt hat nichts mit dem orthographischen Punkt zu tun.

Das typographische Punkt-System geht auf den Pariser Typographen Pierre Simon Fournier (1712–1768) zurück. Als Ausgangsmaß wählte er zwölf Cicero, was etwa zwei Zoll des damaligen Landesmaßes, des »Pied de roi« (französisch für Königsfuß), entsprach. Inspiriert von der englischen Zollteilung unterteilte Fournier dieses Maß dann in zwei Teile mit je zwölf Linien. Ein Sechstel einer Linie bestimmte er als kleinste Einheit, den »Point typographique«. Aus Rücksicht auf bereits bestehende Schriftgrade, insbesondere auf die weit verbreiteten Cicero-Schriftvarianten, hatte er nur die Unterteilung des Landesmaßes übernommen. Der »Pied de roi« entsprach 32,49 cm. Fourniers »Point typographique« war somit nicht mit dem »Pied de roi« kompatibel und besaß auch keine amtliche Bezugsgröße.

1790 beschloss der französische Nationalkonvent – nach einer langen öffentlichen Diskussion – das staatliche Messwesen zu reformieren. Er beauftragte deshalb die Akademie der Wissenschaften, ein neues Einheitensystem auf der Grundlage geeigneter physikalischer Größen auszuarbeiten. Joseph Lagrange und Claude de Berthollet empfahlen daraufhin u. a. die Einführung des Dezimalsystems.

Dieser nationale Reformwille inspirierte vermutlich auch um 1789–1795 den Typographen François Ambroise Didot und seinen Sohn Firmin Didot, den »Point typographique« von Fournier dem »Pied de roi« anzugleichen und auf 0,376065 mm festzulegen. Didots Punkt war nun etwas größer als der bisherige; 11 Didot-Punkte entsprachen 12 Fournier-Punkten.

¹ Es handelt sich bei diesem Artikel um einen Nachdruck des auf der Internetseite <http://www.typolexikon.de> erschienenen gleichnamigen Artikels mit freundlicher Genehmigung des Autors.

Das Didot-Punkt-System löste dann in Westeuropa das Fournier-Punkt-System ab. Die kleinste typographische Maßeinheit war fortan der »Didot-Punkt« und das Maßsystem das »Didotsche Maßsystem«. 1879 wurde der Didot-Punkt vom deutschen Typographen Hermann Berthold (1831–1904) und Wilhelm Foerster (1832–1921) im Auftrag der Vereinigung der deutschen Schriftgießereien auf 0,376 mm abgerundet.

In den USA wurde das Fournier-System weiterhin benutzt und verschmolz im 19. Jahrhundert mit dem britischen Caslon-System zum Pica-Point-System. Seit der Etablierung des PostScript-Systems, in Deutschland von 1985 bis 1994, benutzt man in Westeuropa überwiegend auch die angloamerikanische Maßeinheit »Pica Point« (traditionell) und »PostScript Point« (Adobe Systems). Ein Pica Point misst 0,351 mm. Ein Pica-Punkt ist also um 0,025 mm kleiner als ein Didot-Punkt.

Umrechnungstabelle:

1 Didot-Punkt = 0,376 mm

1 Cicero (Mittel) = 12 Didot-Punkte = 4,512 mm (französisches Pica)

1 Pica Point = 0,35147 mm (gerundet 0,351 mm)

1 Pica Point = 1/72,27 Inch (amerikanisches Pica)

1 Pica Point = 0,0138 Inch

1 Pica Point = 0,9335 Didot-Punkt

1,00375 Pica Point = 1 PostScript Point

1 Pica = 4,2176 mm (gerundet 4,217 mm)

1 Pica = 1/6 Inch

1 Pica = 12 Pica Point

1 PostScript Point (DTP-Punkt von Adobe Systems) = 1,00375001 Pica Point

1 Inch = 25,399 mm (gerundet 25,4 mm bzw. 2,54 cm)

1 Zoll = 6 Pica = 25,4 mm

1 mm = 2,659 Didot-Punkt

1 mm = 0,237 Pica

1 mm = 2,849 Pica Points

1 mm = 0,039 Inch

- Schriftgrade können nur relativ sein, da verbindliche Bemessungsgrundlagen für Schriftgrößen im Sinne der Metrologie und der Typometrie nicht existieren.
- Die Divergenz bei Typometern, Software, Peripheriegeräten (RIPs) etc. ist – trotz oft gleich lautender Termini und Größen – gravierend.
- Schriftgrade, die auf Benutzeroberflächen dargestellt werden, sind noch »relativ« als gedruckte Schriftgrade, egal welche Maßeinheit verwendet wird.

- Schriftgrade auf Benutzeroberflächen sind insbesondere systemabhängig, wie beispielsweise das »Windows/Mac Font-Size-Problem«. Windows berechnet die Bildschirmauflösung mit 96 dpi (dot per Inch = Punkte per Zoll) und Apple mit 72 dpi. Als weitere Variable gilt die physische Größe des Bildschirms und die verwendete Auflösung. Weitere Variablen sind u.a. unterschiedliche Browser.
- Pica-, Didot- oder PostScript-Punkt: Alle drei Punkt-Systeme sind heute in Europa gebräuchlich und in jeder professionellen Schriftsatz-Software (z. B. Adobe InDesign®) kalibrierbar, wobei der »PostScript Point« von Adobe® (DTP-Punkt) nicht mit dem klassischen »Pica Point« zu verwechseln ist. Ein traditioneller Pica Point misst $1/72,27$ Inches, ein PostScript Point $1/72$ Inches.
- Die Feststellung, dass der »Pica Point« kleiner als der »Didot-Punkt« ist, ist insofern wichtig, um typographische Maßeinheiten und Lesetests in der kontinentaleuropäischen Fachliteratur im materiellen Schriftsatz korrekt zu interpretieren. Gleiches gilt natürlich auch für typographische Messinstrumente, wie zum Beispiel Typometer oder für Durchschusstabellen. So kann sich zum Beispiel über die Länge einer DIN-A4-Seite der Unterschied zwischen dem typographischen Maß eines Didot-Punkts und dem eines Pica Points mit bis zu fünf Textzeilen bemerkbar machen.
- Während der prädigitalen Schriftsatzperiode gab es in den USA viele unterschiedliche Pica-Point-Systeme, beispielsweise von IBM, welche in ihrer Metrik stark divergierten.
- Der Typograph Jan Tschichold rechnete den typographischen Didot-Punkt wie folgt um: Ein Punkt ist 0,3759 mm; 2660 Punkte sind ein Meter; 12 Punkte bilden ein Cicero; vier Cicero oder 48 Punkte sind eine Konkordanz.
- In Deutschland ist offiziell das metrische System, also die Angaben in Millimeter, Zentimeter, bzw. Meter bindend.
- Obwohl Lese-, Gleitsichtbrillen und Mehrstärken-Kontaktlinen eine Presbyopie (Alterssichtigkeit, beginnend ca. ab dem 40. Lebensjahr) erfolgreich ausgleichen können, sollte bei der typographischen Gestaltung daran gedacht werden, dass in der Hektik des Alltags nicht immer eine Lesebrille parat ist. Kleine Schriftgrade sind zwar hübsch, aber für die meisten »Über40jährigen« nur schwer oder nicht lesbar.

Nachrichten aus der »T_EXwelt«

Jürgen Fenn

Friedrich Forssman über die Neugestaltung von Reclams Universal-Bibliothek

Friedrich Forssman und seine Frau Cornelia Feyll haben eine der traditionsreichsten Buchreihen neu gestaltet, Reclams Universal-Bibliothek (<http://www.reclam.de>). Der Verlag hat hierzu – natürlich – ein eigenes Reclam-Heftchen herausgegeben (http://www.reclam.de/data/media/Die_Welt_in_Gelb.pdf), in dem auch die Geschichte der UB ausführlich beschrieben wird, einschließlich der Ausgaben, die in der DDR erschienen waren. Forssman schreibt darin (S. 31):

Zum ersten Mal seit der Umstellung auf Antiqua wurde die UB-Innentypographie ganz neu angelegt, zum ersten Mal überhaupt im Sinne eines modernen »Durchgestaltens« ohne konzeptionellen Bruch zwischen Außen und Innen. Die Schrift ist die »Documenta« des Niederländers Frank E. Blokland, eine der schönsten neueren Satzschriften, die besonders gut lesbar ist und sich durch eine hervorragende Zeilenbildung auszeichnet. Darüber hinaus wurde die Textmenge pro Seite gegenüber den bisherigen UB-Serien etwas verringert. Haupttitel und Überschriften stehen linksbündig, der lebende Kolumnentitel steht nun am Fuß, was im kleinen Format bessere Satzspiegelproportionen ermöglicht und vermeidet, dass Kolumnentitel und Zwischenüberschriften einander ins Gehege kommen. Bei der gemeinsamen Arbeit mit der Herstellungsabteilung (vor allem mit Martin Butschek) an den ersten Bänden der Sachbuch-Reihe erwiesen sich diese Grundentscheidungen als praktikabel; Verlag und Gestalter freuen sich über die wohl am besten lesbaren UB-Bände seit 1867.

Wir dürfen gespannt sein.

Neues über die »B_IB_TE_X-Kultur«

Die TIB/UB in Hannover hat eine Online-Umfrage zur Verbreitung und Nutzung von Literaturverwaltungssystemen abgeschlossen. [1] Teilnehmen konnten alle, die von der Umfrage erfahren hatten, also nicht nur die Benutzer der Bibliothek. Es gingen etwa 1000 Antworten ein. Ein Hinweis darauf war u. a. auch in `de.comp.text.tex` gepostet worden. Lambert Heller hat die Ergebnisse der Umfrage zusammengefasst. Sie werfen ein Blitzlicht auf die derzeitige Nutzung von B_IB_TE_X als Datenbankformat im wissenschaftlichen Betrieb:

- Fast 80 Prozent der Befragten arbeiten unter Microsoft Windows, je etwa 10 Prozent unter Linux und Apple Macintosh.

- 35 Prozent haben schon einmal von B_IB_TE_X gehört, aber nur 19 Prozent verwenden es für die wissenschaftliche Arbeit. Führend ist hier Citavi, das von 28 Prozent benutzt wird, während 25 Prozent Zotero verwenden.
- Nach Fächern aufgliedert, ist B_IB_TE_X unter Ingenieuren und Naturwissenschaftlern genauso beliebt wie Citavi, das allgemein führend ist. Häufig wird in diesen Gruppen auch JabRef verwendet (9 Prozent aller Befragten insgesamt). Mendeley setzen 15 Prozent aller Befragten ein. Die Naturwissenschaftler sind mit B_IB_TE_X sogar noch etwas besser vertraut als die Ingenieure, bei denen Citavi und B_IB_TE_X etwa gleich häufig verwendet wird. Sie greifen aber etwa gleichviel auch zum Quellenmanager von Microsoft Word, der »in anderen Fächern (noch?) nahezu unbekannt ist«.
- Die Möglichkeit, PDFs in der Literaturverwaltung verarbeiten zu können, fanden Naturwissenschaftler, von denen die meisten B_IB_TE_X-Benutzer sind, eher weniger interessant.
- Die Naturwissenschaftler zeigen die geringste Neigung, an Online-Schulungen zum Thema Literaturverwaltung teilzunehmen – einen Umstand, den Lambert Heller auf die dort verbreitete »B_IB_TE_X-Kultur« zurückführt: »[...] die meisten Befragten werden hier ohnehin bereits einen gewissen eigenen Lernaufwand hinter sich haben und daher kaum noch erwarten, viel Neues beigebracht zu bekommen.« Immerhin: Ingenieure würden Präsenz-Schulungen an den Universitäten gegenüber Online-Schulungen bevorzugen.
- Obwohl viele Teilnehmer an der Umfrage es begrüßen würden, wenn an den Hochschulen mehr freie Software zum Einsatz käme, sind die meisten der Ansicht, die Universitäten sollten Campus-Lizenzen bereitstellen.

Man sieht, dass es B_IB_TE_X schwer hat, außerhalb der Natur- und Ingenieurwissenschaften Fuß zu fassen, wobei ein Blick in die Auswertung der Umfrage zeigt, dass hier die Mathematiker mit den Naturwissenschaftlern eine gemeinsame Gruppe bildeten. Schade ist auch, dass die vorzügliche Plattform BibSonomy anscheinend so wenig genutzt wird, dass sie hier keine Erwähnung findet. Kritisch wäre anzumerken, dass sowohl Zotero als auch Mendeley das Format B_IB_TE_X verarbeiten können, so dass es interessant gewesen wäre zu erfahren, wie die damit gesammelten Daten von den Benutzern weiter verarbeitet werden.

Literatur

- [1] *Citavi, Endnote, Zotero und Co. – Was benutzen, was brauchen Sie?*; <http://www.tib.uni-hannover.de/de/aktuelles/aktuell/newsdetail/article/citavi-endnote-zotero-und-co/1.html>.

Tipps und Tricks

Favicons mit L^AT_EX

Heiko Oberdiek, Rolf Niepraschk

In der Newsgroup `de.comp.text.tex` kam vor einiger Zeit die Anfrage, wie man mit Hilfe von L^AT_EX ein sogenanntes Favicon erzeugen kann. Es handelt sich dabei um eine innerhalb des Quelltextes von Web-Seiten angegebene kleine Grafik, die ein Web-Browser z. B. in seiner Liste der Lesezeichen anzeigen kann. In Beantwortung dieser Anfrage wurden einige Lösungen gezeigt. Eine davon soll im Folgenden vorgestellt werden. Da es sich bei einem Favicon um eine normale Bitmap-Grafik handelt, kann der beschriebene Weg auch für die Erzeugung beliebiger anderer Grafiken benutzt werden.

Der folgende Quelltext beschreibt, was zu tun ist. Nach dem Laden der Dokumentenklasse wird mit Hilfe des Paketes `geometry` die Seitengröße definiert und sämtliche Ränder werden auf Null gesetzt:

```
7 % favicon.tex
8 \documentclass{article}
9 \usepackage[
10   paperwidth=72bp,% = 1in
11   paperheight=72bp,% = 1in
12   hmargin=0pt,
13   vmargin=0pt,
14 ]{geometry}
```

Wie man sieht, ist die Seitengröße eines L^AT_EX-Dokuments nicht auf die üblichen Papiergrößen beschränkt. Hier ist es stattdessen ein Quadrat mit 1 Zoll Seitenlänge. Davon ausgehend wird es später relativ leicht sein, eine Grafik mit einer bestimmten Auflösung zu erzeugen.

Um innerhalb einer `picture`-Umgebung nicht nur relative Maßangaben nutzen zu können, ist das Paket `picture` nützlich. Dieses Paket und weitere zur Unterstützung von Grafikoperationen und Farbanweisungen geeignete Pakete werden nun geladen, sowie die relative Einheit der `picture`-Umgebung auf 1 bp festgelegt. Da Seitenzahlen natürlich unerwünscht sind, wird der Seitenstil »empty« gewählt:

```

15 \usepackage{picture,graphicx,xcolor}
16 \setlength{\unitlength}{1bp}
17 \pagestyle{empty}

```

Im nun folgenden Hauptteil des Dokumentes wird eine `picture`-Umgebung derart definiert, dass sie genau die Größe der gesamten Seite einnimmt. Alle mittels `\put`-Anweisungen innerhalb der `picture`-Umgebung positionierten Text- oder Grafikelemente werden später auch Bestandteil der zu erzeugenden Bitmap-Grafik:

```

18 \begin{document}
19 \sffamily\bfseries
20 \noindent
21 \begin{picture}(\paperwidth,\paperheight)
22   \put(0,0){%
23     \makebox(\paperwidth,\paperheight){%
24       \resizebox{.9\paperwidth}{.9\paperheight}{\textcolor{blue}{W}}}%
25   }%
26 }%
27 \color{red}%
28 \put(.1\paperwidth,.1\paperheight){\circle*{.1\paperwidth}}
29 \put(.5\paperwidth,.1\paperheight){\circle*{.1\paperwidth}}
30 \put(.9\paperwidth,.1\paperheight){\circle*{.1\paperwidth}}
31 \end{picture}
32 \end{document}

```

Im vorliegenden Fall werden ein großes nahezu blattfüllendes »W« sowie drei rot gefüllte Kreise gleichmäßig im Innern der `picture`-Umgebung positioniert. Genauso gut könnte man auch externe Grafiken an bestimmte Positionen setzen oder Linien und Kurven zeichnen. Auch ist es denkbar, statt der einfachen `picture`-Umgebung die entsprechenden Umgebungen der Pakete »PSTricks« oder »TikZ« zu verwenden, womit die Möglichkeiten der Darstellung noch erheblich erweitert wären.

Um nun aus diesem Quelltext eine für Favicons geeignete Bitmap-Grafik zu erzeugen, sind die folgenden Schritte nötig:

```

pdfflatex favicon.tex
gs -r32 -sDEVICE=png16 -dBATCH -dNOPAUSE -sOutputFile=favicon.png
↪ favicon.pdf

```

Ausgehend von der PDF-Datei »favicon.pdf« (oder im Falle von `PSTricks` »favicon.ps«) wird unter Verwendung des Programms »ghostscript« eine PNG-Datei mit 16 Farben erzeugt. Mit der Angabe »-r32« wird die dabei verwendete Grafikauflösung festgelegt. Hier sind es 32 dpi (dots per inch). Da die Ausgangsdatei eine Größe von genau 1 inch (1 Zoll) hat, führt diese Angabe ohne weiteren Aufwand zu einer Grafik der Größe 32 × 32, was ein für Favicons üblicher Wert ist. Näheres

zu dem ghostscript-Aufruf (z. B. zu weiteren Grafikformaten) findet man in [1]. Falls gewünscht, kann man die png-Grafik mit Hilfe üblicher Grafikprogramme auch in das ebenfalls für Favicons verbreitete Format ».ico« wandeln.

Literatur

[1] *Ghostscript*; <http://www.ghostscript.com/Documentation.html>.

[2] *Favicon*. *Wikipedia*; <http://de.wikipedia.org/wiki/Favicon>.

Seiten drehen mit pdftk

Rolf Niepraschk

Mitunter hat man eine PDF-Datei vorliegen, deren Seiten im Querformat angeordnet sind. Druckt man sie, kann es abhängig vom verwendeten Drucker bzw. Druckertreiber sein, dass das Ergebnis nicht korrekt ist. Beispielsweise kann es geschehen, dass die Seiten nicht automatisch gedreht werden und somit Teile der Seiten abgeschnitten dargestellt werden. Ein weiteres Problem tritt auf, wenn diese Querformat-Datei doppelseitig gedruckt werden soll, was im Interesse des Umweltschutzes wünschenswert ist. In diesem Falle müssten die Seiten abwechselnd um 90 Grad und um -90 Grad gedreht werden. Mit Hilfe des Programms `pdftk` kann man diese Drehungen leicht bewirken:

```
pdftk A=ORIGINAL.pdf shuffle AoddL AevenR output GEDREHT.pdf
```

In der resultierenden Datei `GEDREHT.pdf` sind alle ungeraden Seiten nach links (-90 Grad) und alle geraden Seiten nach rechts (90 Grad) gedreht. Diese Datei kann nun ohne Probleme zweiseitig gedruckt werden und das Resultat ist gut lesbar. Unter Unix-artigen Betriebssystemen könnte man auch auf die Ausgabedatei verzichten und die Daten folgendermaßen direkt an einen Drucker senden:

```
pdftk A=ORIGINAL.pdf shuffle AoddL AevenR output - | lpr
```

Das Programm `pdftk` ist für alle verbreiteten Betriebssysteme verfügbar. Nähere Informationen zu vielen weiteren Möglichkeiten, die `pdftk` bietet, findet man in den angegebenen Literaturstellen.

Literatur und Software

- [1] *pdftk – the pdf toolkit*; <http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/>.
- [2] Stefan Lagotzki: *pdftk - Ein OpenSource-Programm zur Bearbeitung von PDF-Dateien*; <http://www.lagotzki.de/pdftk/>.
- [3] Stefan Lagotzki: *Ein Multifunktionsgerät für die Arbeit mit PDF-Dateien; Die T_EXnische Komödie*; 4/04, S. 19–28; Febr. 2005.

Index mit Fortsetzungsanzeige

Heiko Oberdiek, Herbert Voß

Mit den Paketen fancyhdr oder scrpage2 kann für einen Index in der Kopfzeile der jeweilige Buchstabenbereich der Seite ausgegeben werden. Wenn zusätzlich auf einer Seite angegeben werden soll, dass es eine Fortsetzung des Buchstabens der vorhergehenden Seite ist, so ist einiger Aufwand zu treiben.

Stildatei für den Index

Normalerweise bekommen Stildateien für die Anwendung von `makeindex` die Endung `.ist`. Handelt es sich jedoch um spezielle Anwendungs- oder Testfälle, so empfiehlt es sich, die Möglichkeit einer `mst`-Variante zu nutzen; existiert im aktuellen Dokumentverzeichnis eine Datei mit gleichem Basisnamen und der Endung `mst` so wird diese automatisch von `makeindex` berücksichtigt, ohne dass diese über den Parameter `-s` explizit beim Aufruf von `makeindex` anzugeben ist. Die hier zu verwendende Stildatei wird daher auch einfach über die Umgebung `filecontents` herausgeschrieben:

```
\begin{filecontents}{Voss0-demo.mst}
headings_flag 1
heading_prefix "\\IndexHead{"
heading_suffix "}"
\end{filecontents}
```

Für den Index ist standardmäßig das Paket `makeidx` zu laden und anschließend der Befehl `\makeindex` auszuführen:

```
\usepackage{makeidx}
\makeindex
```

| Personenverzeichnis A–H | Personenverzeichnis Forts. H–V | Personenverzeichnis Forts. V–Z |
|--|---|--|
| <p>Personenverzeichnis</p> <p>Anton, 1 Emil, 1
 Erasmus, 5</p> <p>Berta, 1 Hans, 2
 Harmin, 2</p> <p>Dora, 1 Clothilde, 3
 Hans, 3</p> <p style="text-align: center;">8</p> | <p>Irmhilde, 3 Klaus, 5
 Irmin, 2 Martin, 6</p> <p>Heiko, 4 Rudolf, 2</p> <p>Heiner, 4 Uwe, 6</p> <p>Heinrich, 4 Valentina, 7</p> <p>Henry, 4 Valerie, 7</p> <p>Holger, 4</p> <p>Horst, 5</p> <p>Hugo, 5</p> <p style="text-align: center;">9</p> | <p>Viko, 6 Zacharias, 6
 Volker, 6</p> <p>Willi, 7</p> <p>Xanthippe, 7
 Xaver, 5</p> <p>Yve, 7
 Yvette, 7</p> <p style="text-align: center;">10</p> |

Kopfzeilen

Das obige Beispiel wurde mit dem Paket fancyhdr realisiert, obwohl Gleiches auch mit dem Paket scpage2 aus dem KOMA-Script-Projekt möglich wäre. Hier wird nur die Definition für die Kopfzeile des Index vorgenommen. Soll sich diese auf eine im normalen Text vorhandene Kopfzeile beziehen, so ist der folgende Code entsprechend zu ändern. Die Fußzeile wird nicht verändert.

```
\cleardoublepage
\pagestyle{fancy}
\fancyhead{}
\fancyhead[C]{\IndexHeadLine}%
\fancypagestyle{plain}{%
  \fancyhead{}%
  \fancyhead[C]{\IndexHeadLine}}
\let\indexname\IndexName
\printindex
```

Index

Die Angabe zum Fortsetzungstext wird in \Contname gespeichert. Die Überschrift für den Index wird ebenfalls in der Kopfzeile verwendet und daher gesondert festgelegt.

```
\newcommand\ContName{Forts. }
\newcommand\IndexName{Personenverzeichnis}

\makeatletter
\newcommand*\IndexHeadWrite{%
  \protected@write\auxout{}{%
    \string\@IndexHead{\@CurrentIndexHead}{\noexpand\the\value{page}}}}
\newcommand*\IndexHead[1]{%
  \def\@CurrentIndexHead{#1}%
  \IndexHeadWrite%
  \def\@CurrentIndexHead{\IndexContinuedName#1}}
\protected\def\IndexContinuedName{\ContName\@}%
\newcommand*\@IndexLastPages{}
```

```

\newcommand*\@IndexLastTestPages{}
\newcommand*\@IndexHead[2]{%
  \@ifundefined{ih@first@#2}{\expandafter\xdef\csname ih@first@#2\
  \endcsname{#1}}{}%
  \@ifundefined{ih@last@#2}{\g@addto@macro\@IndexLastPages{,#2}}{}%
  \expandafter\xdef\csname ih@last@#2\endcsname{#1}}
\newcommand*\@IndexHeadTest[2]{%
  \@ifundefined{ih@first@#2}{%
    \global\@IndexRerunWarningtrue
  }{}%
  \@ifundefined{ih@lasttest@#2}{%
    \g@addto@macro\@IndexLastTestPages{,#2}%
  }{}%
  \expandafter\xdef\csname ih@lasttest@#2\endcsname{#1}%
  }%
}
\newif\if@IndexRerunWarning
\usepackage{atveryend}
\AfterLastShipout{%
  \let\@IndexHead\@IndexHeadTest
  \global\@IndexRerunWarningfalse}
\AtVeryEndDocument{%
  \ifx\@IndexLastPages\@IndexLastTestPages
    \begingroup
      \@for\x:=\@IndexLastPages\do{%
        \expandafter\ifx\csname ih@last@\x\expandafter\endcsname
          \csname ih@lasttest@\x\endcsname
        \else\global\@IndexRerunWarningtrue
        \fi}%
      \endgroup%
    \else\global\@IndexRerunWarningtrue%
    \fi%
  \if@IndexRerunWarning
    \latex@warning@no@line{%
      Index headers have changed.\MessageBreak Rerun LaTeX}%
    \fi%
  }
\g@addto@macro\theindex{%
  \expandafter\def\expandafter\item\expandafter{%
    \item\hspace*{0pt}\IndexHeadWrite}%
  \expandafter\def\expandafter\subitem\expandafter{%
    \subitem\IndexHeadWrite}%
  \expandafter\def\expandafter\subsubitem\expandafter{%
    \subsubitem\IndexHeadWrite}%
}
\usepackage{auxhook}
\AddLineBeginAux{\string\providecommand\string\@IndexHead[2]{} }
\newcommand*\IndexHeadLine{%
  \IndexName~
  \@ifundefined{ih@first@the\value{page}}{\textbf{??}}{\@nameuse{
  ih@first@the\value{page}}}%
  --%
}

```

```

\begingroup%
  \def\ContName{}%
%   \let\IndexContinuedName\relax%
  \ifundefined{ih@last@the\value{page}}{\textbf{??}}{\@nameuse{
↪ih@last@the\value{page}}}%
  \endgroup
\makeatother

```

Die Ausgabe des Beispiels bezieht sich auf

```

Hallo Welt
\index{Anton}\index{Berta}\index{Dora}\index{Emil}\index{Egon}\clearpage\null
\index{Rudolf}\index{Hans}\index{Harmin}\index{Harmin!Irmin}\clearpage\null
\index{Harmin!Hans}\index{Harmin!Irmhilde}\index{Harmin!Clothilde}\clearpage\null
\index{Henry}\index{Heiko}\index{Heiner}\index{Heinrich}\index{Holger}\clearpage\
↪null
\index{Horst}\index{Hugo}\index{Xaver}\index{Erasmus}\index{Klaus}\clearpage\null
\index{Volker}\index{Viko}\index{Zacharias}\index{Uwe}\index{Martin}\clearpage\null
\index{Xanthippe}\index{Yve}\index{Yvette}\index{Valentina}\index{Valerie}\index{
↪Willi}

```

Mehrfache Klammern im Mathematikmodus

Herbert Voß

Mehrfache Klammern in richtiger Höhe sind im Mathematikmodus für \TeX bekanntermaßen kein Problem:

$$y(x) = \left(\pi - \arccos \left(\frac{y}{r} \right) \right) \quad (1)$$

$$y(x) = \left(\left(\pi - \arccos \left(\frac{y}{r} \right) \right) \right) \quad (2)$$

$$y(x) = \left(\left(x + \frac{1}{2} \right) \arccos(\pi) \right) \quad (3)$$

```

y(x) &= \left(\pi-\arccos\left(\frac{y}{r}\right)\right)\
y(x) &= \left(\left(\pi-\arccos\left(\frac{y}{r}\right)\right)\right)\
y(x) &= \left(\left(x+\frac{1}{2}\right)\arccos(\pi)\right)

```

Möchte man bei aufeinander folgenden Klammern zur besseren Unterscheidung unterschiedliche Höhen haben, so kann dies über eine Modifikation der Länge `\delimitershortfall` erfolgen. \TeX macht die Klammern nicht höher als:

(berechnete Höhe) · (delimiterfactor)/1000

und

(berechnete Höhe) – (delimitershortfall).

The general idea is that delimiters are vertically centered with respect to the *axis*; hence, if we want to cover a subformula between `\left` and `\right` that extends y_1 units above the axis and y_2 units below, we need to make a delimiter whose height plus depth is at least y units, where $y = 2 \max(y_1, y_2)$. It is usually best not to cover the formula completely, however, but just to come close; so \TeX allows you to specify two parameters, the `\delimiterfactor` f (an integer) and the `\delimitershortfall` δ (a dimension). The minimum delimiter size is taken to be at least $y \cdot f/1000$, and at least $y - \delta$. Appendix B sets $f = 901$ and $\delta = 5$ pt. Thus, if $y = 30$ pt, the plain \TeX format causes the delimiter to be more than 27 pt tall; if $y = 100$ pt, the corresponding delimiter will be at least 95 pt tall. [1, Seite 446]

Die aktuell vorgegebenen Werte führen allerdings nicht automatisch zu größeren Klammern, wenn diese aufeinander folgen. Setzt man den Wert von `\delimitershortfall` auf -1 pt, wodurch $y - \delta$ größer wird, ändert sich dies zu

$$y(x) = \left(\pi - \arccos \left(\frac{y}{r} \right) \right) \quad (4)$$

$$y(x) = \left(\left(\pi - \arccos \left(\frac{y}{r} \right) \right) \right) \quad (5)$$

$$y(x) = \left(\left(x + \frac{1}{2} \right) \arccos(\pi) \right) \quad (6)$$

Wie insbesondere der Quellcode zu Gleichung (2) zeigt, ist zum einen die direkt aufeinander folgende Anwendung des Makros `\right` etwas umständlich und zum anderen der Abstand zwischen zwei aufeinander folgender Klammern etwas groß. Es macht Sinn, statt `...))` besser `...)))` zu setzen. Es wird daher eine erweiterte Definition vorgenommen, die sowohl für `\left` als auch `\right` eine vereinfachte Mehrfachangabe der Klammern ermöglicht, beispielsweise `\right)))` mit gleichzeitig reduziertem Abstand. Dieser Ausdruck setzt nach wie vor das Vorhandensein entsprechend vieler Makros `\left` voraus; denn intern wird `\right)))` in drei `\right)` umgewandelt.

```
\let\Right\right
\let\Left\left
\makeatletter
\renewcommand\right[1]{\Right#1@ifnextchar}{!\right}{}}
```

```
\renewcommand\left[1]{\Left#1\@ifnextchar{\!\left}{}}
\makeatother
```

Der mathematische Ausdruck kann dann wie folgt eingegeben werden:

```
\begin{align}
y(x) &= \left(\pi - \arccos \left(\frac{y}{r}\right)\right)\!
y(x) &= \left(\left(\pi - \arccos \left(\frac{y}{r}\right)\right)\!
y(x) &= \left(x + \frac{1}{2}\right)\arccos(\pi)\right)
\end{align}
```

$$y(x) = \left(\pi - \arccos\left(\frac{y}{r}\right)\right) \tag{7}$$

$$y(x) = \left(\left(\pi - \arccos\left(\frac{y}{r}\right)\right)\right) \tag{8}$$

$$y(x) = \left(\left(x + \frac{1}{2}\right)\arccos(\pi)\right) \tag{9}$$

Die obige Definition bezieht sich allerdings nur auf die runden Klammern, andere, wie eckige oder geschweifte, werden dadurch nicht erfasst. Weitere Klammern lassen sich aber leicht berücksichtigen, indem die Abfrage auf das nächste Zeichen zu einer Kette gestaltet wird:

```
\makeatletter
\renewcommand\right[1]{\Right#1\@ifnextchar{\!\right}{%
\@ifnextchar{\!\right}{%
\@ifnextchar\{\!\right}{%
\@ifnextchar...\!\right}{}}}
\renewcommand\left[1]{\Left#1\@ifnextchar{\!\left}{%
\@ifnextchar[\!\left]{%
\@ifnextchar\{\!\left}{%
\@ifnextchar...\!\left}{}}}
\makeatother
```

Wie das nächste Beispiel zeigt, kann man das *alte* Verhalten der Makros `\left` und `\right` durch Anwendung der *neu* definierten Makros `\Left` und `\Right` erhalten.

$$y(x) = \left\{ \left\{ \frac{1}{2} \right\} \right\} \tag{ohne Anpassung}$$

$$y(x) = \left\{ \left\{ \frac{1}{2} \right\} \right\} \tag{mit Anpassung}$$

```

\makeatletter
\renewcommand\right[1]{\Right#1@ifnextchar}{!\right}{%
  \@ifnextchar}{!\right}{%
    \@ifnextchar}{!\right}{}}}
\renewcommand\left[1]{\Left#1@ifnextchar}{!\left}{%
  \@ifnextchar}{!\left}{%
    \@ifnextchar}{!\left}{}}}
\makeatother
\begin{gather}
y(x) = \Left\{\Left\{ \frac 12 \Right\} \Right\}\tag*{ohne Anpassung}\
y(x) = \left\{\left\{ \frac 12 \right\}\right\} \tag*{mit Anpassung}
\end{gather}

```

Die hier angegebenen Gleichungen zeigen übrigens ein weiteres Problem mit dem Satz von Gleichungen. Man vergleiche $\arccos(\pi)$ mit $\arccos(\pi)$, wobei der erste Ausdruck ohne und der zweite mit der Anwendung von `\left(` und `\right)` gesetzt wurde. \TeX betrachtet den Inhalt einer solchen Kombination wegen der internen Anwendung von `\mathopen` und `\mathclose` als »math atom«, um das dann automatisch ein entsprechender Abstand gesetzt wird. Möchte man diesen verhindern, so ist ein `\!` vor die Klammer zu setzen: `\arccos\!\left(\pi\Right)` \Rightarrow $\arccos(\pi)$. Dieser Vorgang kann mit dem Paket `mleftright` von Heiko Oberdiek vereinfacht werden; es stellt dafür die Makros `\mleft` und `\mright` bereit. Das Paket wird voraussichtlich demnächst auch auf CTAN zur Verfügung stehen und daraus folgend dann auch für \TeX Live und $\text{MiK}\TeX$.

Literatur

- [1] Donald E. Knuth: *The \TeX book*; Addison Wesley Professional; 21. Aufl.; 1986.

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 sind.

check-parens von *Jaap Eldering* ist ein C++-Programm, mit dem man Klammern im \LaTeX -Quelltext prüfen kann.

CTAN:support/check-parens

mdsymbol von *Sebastian Schubert* ist ein Font mit mathematischen Symbolen, die zu der Schrift Adobe Myriad Pro passen.

CTAN:fonts/mdsymbol

reotex von *Nuno Oliveira* dient zum Zeichnen von Reo-Channels und -Circuits mithilfe von *pgf/TikZ*.

CTAN:graphics/pgf/contrib/reotex

parnotes von *Michael Hughes* definiert Befehle, mit denen man Fußnoten (automatisch oder händisch) direkt unter einen Absatz setzen kann.

CTAN:macros/latex/contrib/parnotes

happy4th von *Brian Blackmore* erzeugt ein » \TeX -Feuerwerk« zum 4. Juli. Zum Betrachten muss man die Darstellung im Viewer kontinuierlich weiterblättern.

CTAN:macros/plain/contrib/happy4th

pst-tools von *Herbert Voß* ist »ein Hilfspaket für *pstricks*«.

CTAN:graphics/pstricks/contrib/pst-tools

hanoi von *Brian Blackmore* ist ein (übrigens sehr schön formatiertes) Plain- \TeX -Programm, das die Zugfolge der »Türme von Hanoi« ausgibt.

CTAN:macros/plain/contrib/hanoi

romanbar von *H.-Martin Münch* erlaubt die Ausgabe von römischen Zahlen mit abschließenden Balken darüber und darunter.

CTAN:macros/latex/contrib/romanbar

xcookybooky von *Sven Harder* eignet sich zum Setzen umfangreicherer Rezepte, einschließlich Bildern und Nährwertangaben.

CTAN:macros/latex/contrib/xcookybooky

nameauth von *Charles P. Schaum* dient zum Vereinheitlichen von Namen, die in einem Text laufend auftauchen.

CTAN:macros/latex/contrib/nameauth

reverxi von *Bruno Le Floch* ist ein Programm, mit dem man Reversi gegen \TeX spielen kann.

CTAN:macros/plain/contrib/reverxi

libertine-legacy von *Bob Tennent* enthält die Type-1-Version der Linux-Libertine-Fonts, wie sie bis Dezember 2011 verteilt worden waren. Aktuelle Versionen von Libertine unterstützen nur noch das OpenType-Format für \LaTeX und \XeTeX .

CTAN:fonts/libertine-legacy

copyrightbox von *Ives van der Flaas* stellt einen Befehl bereit, mit dem man einen Copyright-Hinweis neben einer Abbildung platzieren kann.

CTAN:macros/latex/contrib/copyrightbox

tcolorbox von *Thomas F. Sturm* stellt eine Umgebung für Textboxen bereit, mit denen \LaTeX -Quelltext und -Ausgabe, aber auch Theoreme gesetzt werden können.

CTAN:macros/latex/contrib/tcolorbox

flipbook von *Olivier Buffet* dient zum Erstellen eines »Daumenkinos«.

CTAN:macros/latex/contrib/flipbook

amiri-font von *Khaled Hosny* ist der Beta-Release eines klassischen arabischen Fonts in Unicode 6.0.

CTAN:fonts/amiri

gates von *Paul Isambert* ermöglicht es, Code modular zu schreiben.

CTAN:macros/generic/gates

sasnrdisplay von *Lars Madsen* dient als ein Frontend zu *listings* und erlaubt es insbesondere Statistikern, Quelltext und die Ergebnisse von Berechnungen oder Simulationen in \LaTeX -Projekte zu importieren.

CTAN:macros/latex/contrib/sasnrdisplay

documentation von *Omar Salazar Morales* dient zum Setzen von Quelltext in C, Java und Assembler mit \LaTeX .

CTAN:macros/latex/contrib/documentation

gamebook von *André Miede* ist ein Paket, mit dem man Spielbücher setzen kann, bei denen sich der Leser aussuchen kann, wie die Geschichte weitergehen soll.

CTAN:macros/latex/contrib/gamebook

fullwidth von *Marco Daniel* erlaubt es, die linken und rechten Seitenränder einfach anzupassen.

CTAN:macros/latex/contrib/fullwidth

philosophersimprint von *Boris Veytsman* ist eine Klasse für die Zeitschrift *Philosopher's Imprint*.

CTAN:macros/latex/contrib/philosophersimprint

apa6 von *Brian Beitzel* setzt die 6. Auflage der *American Psychological Association's Publication Manual* aus dem Jahr 2009 um, das in vielen sozialwissenschaftlichen Fächern verwendet wird.

CTAN:macros/latex/contrib/apa6

bchart von *Tobias Kuhn* ist ein \LaTeX -Paket zum Zeichnen einfacher horizontale Balkendiagramme mit einer numerischen x -Achse. Es setzt auf PGF/TikZ auf.

CTAN:macros/latex/contrib/bchart

diagbox von *Leo Liu* will ein modernerer Ersatz für das Paket *slashbox* sein. Damit kann man diagonal geteilte Tabellenköpfe setzen.

CTAN:macros/latex/contrib/diagbox

kantlipsum von *Enrico Gregorio* ist ein Blindtextgenerator, der auf dem *Kant generator for Python* aufsetzt, den Mark Pilgrim in *Dive into Python* beschreibt. Vorbild war das Paket *lipsum*.

CTAN:macros/latex/contrib/kantlipsum

titleps von *Javier Bezos* ist Teil der Sammlung *titlesec* und dient zum Auswählen von Seitenstilen.

CTAN:macros/latex/contrib/titlesec

biblatex-fiwi von *Simon Spiegel* ist ein *biblatex*-Stil zur Angabe von Fundstellen in geisteswissenschaftlichen Arbeiten, insbesondere in der Filmwissenschaft.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-fiwi

dosepsbin von *Heiko Oberdiek* ist ein Programm, das EPS-Dateien analysiert, die als DOS-EPS-Binaries vorliegen.

CTAN:support/dosepsbin

hletter von *Brian Housley* ist ein neues Paket zum Briefeschreiben mit \LaTeX , das dem Benutzer u. a. die Auswahl zwischen mehreren Varianten für den Briefkopf überlässt und mehrere Unterschriften unterstützt. Serienbriefe können mit *merge* erstellt werden. Eine eingescannte Unterschrift und ein Logo können eingefügt werden.

CTAN:macros/latex/contrib/hletter

cookingsymbols von *Sven Harder* enthält elf Symbole zum Setzen von Rezepten: Herd, Ober- und/oder Unterhitze, Umluft- oder Gasherd, Gedeck und Besteck sowie Handschuhe.

CTAN:macros/latex/contrib/cookingsymbols

biblatex-luh-ipw von *Oliver Heins* ist eine Sammlung von *biblatex*-Stilen nach den Vorgaben des Instituts für Politische Wissenschaft der Leibniz-Universität Hannover.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-luh-ipw

thaifonts-arundina von *Theppitak Karoonboonyanan* enthält thailändische Fonts, die mit *DejaVu* kompatibel sind.

CTAN:fonts/thai/thaifonts-arundina

kgddocs von *Walter Daems* ist eine Klasse für die *Karel-de-Grote-Universität* Antwerpen.

CTAN:macros/latex/contrib/kgddocs

logbox von *Florent Chervet* dient zur Ausgabe von Boxinhalten mit `\logbox` und `\showbox`, ohne die Kompilation zu unterbrechen.

CTAN:macros/latex/contrib/logbox

fcltxdoc von *Florent Chervet* enthält Makros zum Bearbeiten der Dokumentationen des Autors.

CTAN:macros/latex/contrib/fcltxdoc

easyfig von *Martin Scharrer* erleichtert das Einfügen von Bildern in der am häufigsten benötigten Form (zentriert mit einer Bildunterschrift und Label).

CTAN:macros/latex/contrib/easyfig

ipaex von *Norbert Preining* enthält die Fonts IPAex und IPA von der Information-technology Promotion Agency, Japan einschließlich Unterstützung.

CTAN:fonts/ipaex

meetingmins von *Brian Beitzel* dient zum Schreiben von Sitzungsprotokollen, aufbauend auf *mins*.

CTAN:macros/latex/contrib/meetingmins

tablefootnote von *H.-Martin Münch* erleichtert das Setzen von Fußnoten innerhalb einer Tabelle mit dem neuen Befehl `\tablefootnote`.

CTAN:macros/latex/contrib/tablefootnote

ghab von *Vafa Khalighi* definiert den gleichnamigen Befehl, der sein Argument in eine Box mit dekoriertem Rahmen setzt.

CTAN:macros/latex/contrib/ghab

tram von *Vafa Khalighi* erzeugt mit kleinen Punkten versehene Boxen, um den darin enthaltenen Text grau zu hinterlegen.

CTAN:macros/latex/contrib/tram

translation-biblatex-de ist die von *Christine Römer* und elf Studierenden der Universität Jena erstellte Übersetzung der Anleitung zu Version 1.6 des Pakets *biblatex*.

CTAN:info/translations/biblatex/de

mversion von *Michael Schmeing* definiert einen einfachen Zähler für die Versionsnummer des Dokuments, der bei jedem Kompilieren um einen Schritt hochzählt.

CTAN:macros/latex/contrib/mversion

fifinddo-info von *Uwe Lück* enthält drei Pakete mit Hypertexttools für die Pakete *hyperref* und *fifinddo*.

CTAN:info/fifinddo-info

Spielplan

Termine

2012

- 7.3. – 9.3. **DANTE 2012**
und 46. Mitgliederversammlung von DANTE e.V.
Hochschule für Technik, Wirtschaft und Kultur
Lipsius-Bau, Karl-Liebknecht-Straße 145, 04277 Leipzig
<http://www.dante.de/events/dante2012.html>
- 17.3. – 18.3. **Chemnitzer Linuxtage 2012**
Technische Universität Chemnitz
09107 Chemnitz
<http://www.chemnitzer.linux-tage.de>
- 29.4. – 3.5. **20. BachoT_EX-Konferenz 2012**
Bachotek, nahe Brodnica, Polen
<http://www.gust.org.pl/bachotex/2012/>
- 23.5. – 26.5. **Linuxtag Berlin**
Messegelände
14055 Berlin
<http://www.linuxtag.org/2012/>
- 16.7.–18.7. **TUG 2012**
Boston, Massachusetts (USA)
<http://tug.org/tug2012/>
- 8.10. – 12.10. **EuroT_EX 2012**, 6th International ConT_EXt meeting und DANTE-Herbsttagung mit 47. Mitgliederversammlung Breskens (Niederlande)
<http://meeting.contextgarden.net/2012/>

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

Michael-E. Voges, Tel.: (03362) 50 18 35,

mevoges@t-online.de

Ort derzeit wechselnd

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

Erlangen

Walter Schmidt, Peter Seitz, w.a.schmidt@gmx.net

Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen

Dritter Dienstag im Monat, 19.00 Uhr

Frankfurt

Harald Vajkonny,

<http://wiki.lug-frankfurt.de/TeXStammtisch>

Restaurant »moschmosch«, Wilhelm-Leuschner-Straße 78, 60329 Frankfurt

Vierter Donnerstag im Monat, 19.30 Uhr

Hamburg

Lothar Fröhling, lothar@thefroehlings.de

Restaurant Sandstuv, Neue Straße 17, 21073 Hamburg-Harburg

Letzter Dienstag im Monat, 19.00 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,

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

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

nach Vereinbarung

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

| | | | |
|---|----------|---|---------------------|
| Wolfgang Beinert
Berlin
office.2012@beinert.net | [28] | Heiko Oberdiek
Kroppenstück 9
77880 Sasbach
heiko.oberdiek@googlemail.com | [33, 36] |
| Jürgen Fenn
Friedensallee 174/20
63263 Neu-Isenburg
juergen.fenn@gmx.de | [30, 43] | Agnieszka Okońska
Domänenstraße 16a
54295 Trier
agnieszka@okonska.eu | [6] |
| Rolf Niepraschk
Persiusstr. 12
10245 Berlin
Rolf.Niepraschk@gmx.de | [33, 35] | Volker RW Schaa
siehe Seite 50 | [4] |
| | | Herbert Voß
Wasgenstraße 21
14129 Berlin
herbert@dante.de | [3, 13, 20, 36, 39] |

Die T_EXnische Komödie

24. Jahrgang Heft 1/2012 Februar 2012

Impressum

Editorial

Hinter der Bühne

4 Grußwort

Bretter, die die Welt bedeuten

6 L^AT_EX für Juristen

13 Datumsfunktionen mit LuaT_EX

20 Multilinguale Texte mit LuaL^AT_EX – ein Versuch

28 Typographischer Punkt

31 Nachrichten aus der »T_EXtWelt«

Tipps und Tricks

33 Favicons mit L^AT_EX

35 Seiten drehen mit pdftk

36 Index mit Fortsetzungsanzeige

39 Mehrfache Klammern im Mathematikmodus

Von fremden Bühnen

43 Neue Pakete auf CTAN

Spielplan

47 Termine

48 Stammtische

Adressen

51 Autoren/Organisatoren