

Die TeXnische Komödie

dante

Deutschsprachige
Anwendervereinigung TeX e.V.

34. Jahrgang Heft 2/2022 Mai 2022

2/2022

Impressum

»Die \TeX nische 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- \LaTeX -Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und per E-Mail oder Datenträger (z. B. CD/DVD) an unten stehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, \LaTeX -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 <https://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit `LuaHBTeX`, Version 1.15.0 (TeX Live 2022) erstellt. Als Standardschriften kamen Libertinus Serif, Libertinus Sans Serif, Anonymous Pro und Libertinus Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2150

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung \TeX e.V.
Postfach 11 03 61
69072 Heidelberg

E-Mail: info@dante.de (DANTE e.V.)
dtkred@dante.de (Redaktion)

Druck: Schleunungdruck GmbH
Eltertstraße 27, 97828 Markttheidenfeld

Redaktion: Luzia Dietsche (verantwortliche Redakteurin)

Mitarbeit: Adelheid Bonnetsmüller Rudolf Herrmann Eberhard Lisse
Ralf Mispelhorn Rolf Niepraschk Stefan Pinnow
Bernd Raichle Christine Römer Herbert Voß

Redaktionsschluss für Heft 3/2022: 15. Juli 2022

ISSN 1434-5897

Die \TeX nische Komödie 2/2022

Editorial

Liebe Leserinnen und Leser,

diese Ausgabe unserer Mitgliederzeitung hat mir wieder einmal gezeigt, was ich an T_EX besonders liebe – die Gemeinschaft, die national und international existiert. Als der Redaktionsschluss nahte, musste ich nämlich zu meinem Leidwesen feststellen, dass ich kein Material für eine neue Ausgabe hatte. Also startete ich einen Hilferuf und bekam ziemlich schnell Beiträge zugesandt. Nicht nur das, der Hilferuf wurde weiter getragen und mir wurde ein Artikel angeboten, der als Übersetzung bereits in den »Cahiers GUTenberg« erschienen war. Und so entstand ein Projekt über mehrere Grenzen hinweg, das in dem Artikel »Druck oder Nichtdruck« auf Seite 64ff mündete. Gibt es eine solche Verbundenheit bei anderen Programmen zum Setzen von Texten auch?

Was Herbert Voß in seinem Artikel auf Seite 30ff zum Einbinden von externen Quellen beschreibt, konnte ich direkt bei einem anderen Beitrag in dieser Ausgabe anwenden. Denn auch wenn sich jeder Artikel einzeln fehlerlos mit der Klasse für »Die T_EXnische Komödie« übersetzen lässt, heißt das noch lange nicht, dass das auch mit allen *zusammen* funktioniert. Der Bericht über hvextern.sty ist nicht nur interessant zu lesen, sondern hat mir unmittelbar sehr gute Dienste geleistet.

Und so wichtig für mich jeder der »einmaligen« Artikel ist, liefern mir auch die immer wiederkehrenden Beiträge wie »Neue Pakete auf CTAN« und – neu – »ConT_EXt kurz notiert« wichtige und interessante Informationen. Vielleicht finden sich ja auch noch für andere Bereiche freiwillige Berichterstatter?

In der Hoffnung, dass »Die T_EXnische Komödie« auch weiterhin so gut und vielfältig befüllt wird, wünsche ich Ihnen und Euch wieder viel Spaß beim Lesen und verbleibe mit T_EXnischen Grüßen
Luzia Dietsche

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

36 Anmeldungen liegen aktuell (Mitte Mai) für die Frühjahrstagung in Magdeburg vor, die coronabedingt vom 23. bis 25. Juni stattfindet und daher eher das Etikett „Sommertagung“ verdient. Der überwiegende Teil der angemeldeten Teilnehmerinnen und Teilnehmer möchte, wie vermutet, vor Ort dabei sein. Ich bin guter Dinge, dass dies möglich sein wird und ich freue mich schon sehr darauf. Es lohnt sich aus ganz unterschiedlichen Gründen, dabei zu sein. Bitte melden Sie sich bei Interesse schnellstmöglich unter <https://www.dante.de/dante2022-anmeldeformular/> an. Sie erleichtern damit die Organisation enorm.

Bis zum Erscheinen dieser Ausgabe von »Die \TeX nische Komödie« haben wir hoffentlich schon einen ersten groben Ablaufplan, aktuell fehlen aber durchaus noch einige Vorträge. Scheuen Sie sich also nicht, auch kurzfristig noch Angebote an dante2022@dante.de zu richten.

Bis zur Tagung, genauer bis zur Mitgliederversammlung, die diesmal am Samstag Vormittag stattfinden wird, sind wir als Vorstand hoffentlich auch schlauer, was die Zukunft unseres Büros angeht. Nachdem wir personell wieder in ruhigeres Fahrwasser gekommen sind, gibt es durch einen (neuerlichen) Eigentümerwechsel des gesamten Wohngebäudes ein wenig Unruhe. Infolgedessen fanden bereits erste Gespräche mit dem neuen Vermieter über die Zukunft für uns als Mieter statt. Wir waren in den vergangenen fast 30 Jahren sehr zufrieden und würden die Büroräume grundsätzlich gerne behalten. Jeder, der sich in den vergangenen Jahren auch nur ein wenig mit dem Wohnungsmarkt in attraktiven Großstadtlagen beschäftigt hat, kann sich jedoch vorstellen, dass es sehr schwierig sein wird, unser günstiges Mietverhältnis dauerhaft zu sichern. Eine schwierige Entscheidung könnte anstehen, es bleibt spannend.

Themenwechsel: Wie manch anderer » \TeX -Enthusiast« versuche ich, in ganz unterschiedlichen Kontexten \TeX einzusetzen. Wie schon das ein oder andere Mal berichtet, hatte ich beruflich in den vergangenen Jahren viele Möglichkeiten dazu.

Aber auch im privaten Umfeld gab es immer wieder interessante Herausforderungen, die ich mit Hilfe von T_EX und Co. und nicht zuletzt den vielen online verfügbaren Dokumentationen und Hilfsangeboten elegant lösen konnte.

So habe ich beispielsweise als Kassenführer eines Fördervereins die Rechnungen, Spendenquittungen, Visitenkarten und auch die allgemeine Mitgliederkorrespondenz selbstverständlich mit L^AT_EX gemacht. Jetzt stehe ich vor dem Problem, dass ich nach vier Jahren dieses Amt aufgegeben habe und für die Übergabe der vorhandenen Dokumente eine »massenkompatible« Lösung wählen muss. Das heißt konkret, Dokumente und Vorlagen in LibreOffice im docx-Format aufzubereiten und zu hoffen, dass neben der Optik (Umbruch, Abstände, OpenType-Fonts mit unterschiedlichen Ziffernsätzen etc.) nicht auch die schönen Ergänzungen wie der CSV-Import (über csvsimple) und die Serienbrieffunktion (in Verbindung mit scr1ttr2) auf der Strecke bleiben.

Zu den wenigen Dingen, die die Coronapandemie nicht erschüttern konnte, gehört die Zusammenstellung und Veröffentlichung der T_EX Collection. Nahezu präzise wie ein Uhrwerk stellt ein eingespieltes Team aus Freiwilligen Jahr für Jahr ab März die einzelnen Bestandteile zusammen, verfeinert und testet sie, damit T_EX-Nutzergruppen weltweit im Mai eine neue Auflage in Form einer DVD erhalten und an ihre Mitglieder verteilen können. In diesem Jahr wurde die mittlerweile veraltete ProT_EXt-Installation durch eine speziell aufbereitete MiK_TE_X-Variante ersetzt. Die DVD des Jahres 2022 kommt in schicker dunkelblauer Verpackung daher und liegt dieser Ausgabe von »Die T_EXnische Komödie« bei. Vielen Dank allen Beteiligten für das routinierte Zusammenwirken.

In diesem Sinne viel Vergnügen beim Stöbern in der T_EX Collection und bei der weiteren Lektüre.

Herzlichst Ihr/Euer
Martin Sievers

Einladung zur Frühjahrstagung 2022 in Magdeburg

Mathias Magdowski

Liebe Mitglieder von DANTE e.V.,

wir laden Sie ganz herzlich zur Frühjahrstagung 2022 vom 23.–25. Juni 2022 in Magdeburg ein. Der Zeitplan sieht wie folgt aus:

Mittwoch, 22. Juni, ab 19 Uhr: Vorabendtreff
Donnerstag, 23. Juni, 9 bis 17 Uhr: Tutorien und Vorträge
ab 19 Uhr: Abendtreff
Freitag, 24. Juni, 9 bis 17 Uhr: Vorträge
ab 19 Uhr: Tagungssessen
Samstag, 25. Juni, ab 9 Uhr: 64. Mitgliederversammlung
anschließend Touristikprogramm (geplant)

Wir möchten als Organisatoren natürlich ein spannendes und vielfältiges Vortragsprogramm anbieten können. Dazu sind wir allerdings auf Eure/Ihre aktive Unterstützung angewiesen. Sollten Sie zu Textsatz, Typographie oder Einsatz von \TeX und Konsorten einen Vortrag halten wollen, freuen wir uns über alle Einreichungen; es darf auch gerne Ihr erster »Auftritt« bei einer Tagung von DANTE e.V. sein.



Unter <https://www.dante.de/veranstaltungen/dante2022/> finden Sie die Tagungsseite mit allen weiteren Informationen rund um die Veranstaltung. Wir bitten diesmal allerdings noch dringender als sonst wegen der besseren Planbarkeit um eine frühzeitige Anmeldung über die Tagungs-Website. Für Fragen, Wünsche und Anregungen schreiben Sie bitte an dante2022@dante.de.

Bericht über Projektförderung von \LaTeX -Servern

Stefan Kottwitz

Seit mehreren Jahren fördert DANTE e.V. den Betrieb zweier Server, die verschiedene \LaTeX -bezogene Dienste bereitstellen. Hier möchte ich einmal darüber berichten, was so dazu gehört und wie der Dienst entstanden ist.

Die beiden Server stehen in zwei Rechenzentren von <https://www.hetzner.de>. Auf ihnen läuft ein Debian-Linux mit Virtualisierung durch Xen: Die in den nächsten Abschnitten genannten Dienste laufen für eine saubere Trennung und Wartbarkeit auf etlichen virtuellen Maschinen. Die Server sichern sich zum Einen gegenseitig, sowie zum Anderen auf einem externen Backup-Bereich. Bei Upgrades bzw. Migrationen von Subservern wird der jeweils andere Server zum Neuaufstellen genutzt.

Hier ein paar Spezifikationen der beiden Server:

- Intel Xeon E5-1650V3 Six Core, 8 Terabyte Festplatten, 256 Gigabyte RAM
- Intel i7-920 Quad-Core, 2 Terabyte Festplatten, 48 Gigabyte RAM

Beide verfügen über eine 1-Gbit-Netzwerkanbindung und unbegrenztes Traffic-Volumen sowie einen Off-Site-Backup-Bereich. Diese Hardware mit allen Diensten im professionellen Rechenzentrumsbetrieb kostet bei der Firma Hetzner Miete, momentan 104,- bzw. 59,- € monatlich, plus Gebühren für IP-Adressen und DNS-Einträge. Etwas ungewöhnlich dabei ist, dass ich den zuerst genannten Server bei einer Rückwärtsauktion bestellt habe. Dabei werden vom Provider gebrauchte und aufgegebene Server zu einer Monatsmiete angeboten, bei der der Preis als Angebot Stück für Stück sinkt, bis jemand zugreift, wodurch es für mich relativ günstig wurde. Das kann man mit »Dedicated Servern« beispielsweise von OVH anstellen, sobald man aber Parameter wie 256 Gigabyte RAM hat, wird es teuer.

DANTE e.V. fördert das Projekt durch die Übernahme der Monatsmiete und die Gebühren der Server. Die Kosten für die Domains sind dabei nicht enthalten.

Und nun zu den Inhalten.

Internet-Foren

<https://LaTeX.org> ist ein klassisches internationales Internet-Forum in englischer Sprache. Der TeXnicCenter-Entwickler Sven Wiegand gründete es als <http://LaTeX-Community.org> im Jahr 2007, 2008 kam ich als Moderator hinzu, übernahm 2012 die Administration und zog es auf einen eigenen Server. 2017 konnte ich die Domain <https://LaTeX.org> sichern und so heißt nun auch das Forum. Mittlerweile gibt es 99 000 Beiträge zu 25 000 Themen von 17 000 Nutzern, die klassische Thread-Diskussionskultur lebt hier weiter.

<https://goLaTeX.de> wurde als deutschsprachiges Forum von Johannes Ähling im Jahr 2008 gegründet. 2015 habe ich zum Zweck der Software-Erneuerung als Admin übernommen und das Forum neu aufgesetzt. Bis heute schrieben hier 8 700 Nutzer zu 21 000 Themen über 108 000 Beiträge.

<https://TeXwelt.de> habe ich 2013 als deutschsprachiges Pendant zu StackExchange gestartet, auch um dieses Q&A-Format auszuprobieren und es für deutschsprachige Anwender anzubieten. Von 1 300 Nutzern wurden 3 700 Fragen und 4 200 Antworten verfasst. Hier möchte ich auf die Askbot-Software oder eine andere Alternative wechseln. Vielleicht kann dabei jemand helfen?

<https://TeXnique.fr> entstand 2016 als gemeinsames Projekt mit Patrick Bideault und Denis Bitouzé und mit Unterstützung der französischen Gruppe GUTenberg, als französischsprachige Frage-Antwort-Seite zu L^AT_EX. Hier schrieben bisher 400 Nutzer 1 200 Fragen und 1 600 Antworten.

Jedes dieser Foren hat einen integrierten Online-Compiler, so dass man Code-Beispiele durch ein Antippen auf dem Smartphone direkt übersetzen und als PDF-

Ausgabe ansehen kann. Zudem kann man Code in den Beiträgen online editieren und so direkt verändert testen.

L^AT_EX-Tools

<https://TeXlive.net> ist der oben erwähnte Online-Compiler. Er wurde von David Carlisle entwickelt und zunächst bei <https://LearnLaTeX.org> auf einer kleinen kostenfreien AWS-Instanz eingesetzt. Als das nicht mehr möglich war, zogen wir es auf einen von DANTE e.V. geförderten Server. Grundlage ist ein aktuelles und vollständiges T_EXLive-System mit allen Paketen und Tools wie B_IB_TE_X, Biber und makeindex. Täglich übersetzen etwa 2 000 Anwender so um die 3 000 bis 4 000 Dokumente, so dass ich auf über 1 Millionen übersetzte Dokumente im Jahr schätze. Das kommt auch daher, weil <https://TeXlive.net> in die obengenannten Foren sowie in die später genannten Blogs, Wikis, Galerien und FAQ-Seiten integriert ist.

<https://TeXdoc.org> ist die online-Version von texdoc. Gestartet wurde es 2014 als <https://TeXdoc.net> mit einem von Paulo Cereda geschriebenen Frontend, damit man auch als Overleaf-Nutzer ohne lokale Doku oder unterwegs auf dem Smartphone beim Forum-Browsen einfach online auf T_EX-Dokumentationen zugreifen kann. Und zwar mit Suche oder durch standardisierte Links und API aus den Foren. Mittlerweile läuft es als Docker-Container, der vom *Island of T_EX project* weiterentwickelt wird.

<https://CTAN.net> habe ich 2018 als CTAN-Mirror eingerichtet, weil die Server nunmal da waren und Kapazität hatten. Da war es naheliegend, CTAN zu spiegeln, um die Installationslast weiter zu verteilen und zu beschleunigen. Und wie Michael Doob sagte: „Having a ... CTAN mirror will change your life.“

<https://LaTeX2e.org> ist ein Spiegel des „L^AT_EX 2_ε Unofficial Reference Manual“ von <https://latexref.xyz>, da eine möglichst weite Spiegelung gewünscht ist.

Wikis und FAQs

<https://TeXfragen.de> wurde von Patrick Gundlach als neue T_EX-FAQ in Form eines Wikis gestartet, etliche Unterstützer steuerten Artikel bei. 2017 habe ich es als Admin übernommen und das Wiki unter anderem um ausgewählte exzellente Antworten von <https://TeXwelt.de> ergänzt und natürlich das Online-Übersetzen wie auf den anderen Seiten hinzugefügt.

<https://goLaTeX.de/wiki> ist ein weiteres Wiki in deutscher Sprache, das L^AT_EX-Befehle und -Pakete dokumentiert und mit online testbaren Beispielen versieht.

<http://LaTeX.net.br> ist eine brasilianisch-portugiesische Version der klassischen T_EX-FAQ. Ich habe sie 2018 anlässlich der TUG-Tagung in Rio de Janeiro mit einem

Partner übersetzt, mangels Unterstützung aber danach nicht weiterentwickelt, sie ist aber weiterhin online.

<https://TeXfaq.org> habe ich bis vor kurzem selbst gehostet, nun ist es auf GitHub umgezogen. Das war die originale www.tex.ac.uk FAQ von Robin Fairbairns und vielen Beitragenden. Diese konnte irgendwann nicht mehr in Cambridge betrieben werden. Daher habe ich 2015 das Hosten mit Joseph Wright und David Carlisle übernommen, später wechselten wir die Domain. www.tex.ac.uk gibt es als Domain (Weiterleitung) nicht mehr, ich konnte jedoch das Mittelstück <https://tex.ac/> bewahren.

Blogs

L^AT_EX-Entwickler, Autoren und Anwender nutzen Blogs, um hin und wieder Artikel über eigene Projekte, Erfahrungen, oder Neuigkeiten aus der T_EX-Welt zu veröffentlichen. Die folgenden Blogs laufen auf den durch DANTE e.V. geförderten Servern.

<https://LaTeX.net> begann als Artikelsammlung in einem KnowHow-Bereich auf <http://LaTeX-Community.org>, stellt die Artikel aus vielen Jahren bereit und dient heute vorwiegend als Blog für Aktuelles.

<https://TeX-Talk.net> ist das Community-Blog von TeX-Stack-Exchange (TeX.SE). Es lief unter der Domain <https://tex.blogoverflow.com> und wurde 2016 von Stack-Exchange eingestellt. Als Moderator bei StackExchange habe ich mich um den Erhalt bemüht, und so wurde mir der Inhalt zum Weiterbetrieb unter neuem Namen übergeben. Interessant ist hier der Teil mit 20 ausführlichen Community-moderierten Interviews von bekannten TeX.SE-Nutzern.

<https://TeX.my> ist das 2009 gestartete Blog einer malaysischen L^AT_EX-User-Gruppe, 2016 hierher umgezogen. Heute wird es vor allem von Lian Tze Lim weiter geschrieben.

<https://www.cnltx.de> ist das Blog von Clemens Niederberger. Weil der Schwerpunkt auf dem Thema Chemie liegt, lief es zuvor unter dem Namen <https://mychemistry.eu>.

<https://TeX.co> ist ein T_EX-Blog, das Mitglieder von <https://TeXwelt.de> gestartet haben und welches später separiert wurde.

<https://TikZ.de> ist ein Blog von einigen TikZ-Freunden einschließlich mir. Es ist momentan eher ruhig darin, weil bei allen wenig Zeit da ist.

<https://TeXblog.net> ist mein T_EX-Blog von 2008, das ich aber gegenüber allem anderen so hintenan gestellt habe, dass ich es momentan kaum nutze.

Galerien

<https://TeXample.net> ist eine TikZ-Beispielsammlung und wurde 2008 von Kjell Magne Fauske aufgebaut. Da Kjell keine Zeit mehr dafür hatte, übernahm ich 2012 als Admin. Darin sind heute 407 eingesandte TikZ-Beispiele von 190 Autoren. Weiterhin liegt hier ein Blog-Aggregator unter <https://TeXample.net/community>, sozusagen ein Newsticker aus der L^AT_EX-Blog-Welt. Von 50 registrierten Blogs gibt es hier 3 600 Einträge.

<https://TikZ.net> ist eine weitere TikZ-Galerie, die ich für <https://TeXample.net> als potentiellen Nachfolger neu aufsetzt habe, da diese schwer wartbar wurde. Die allermeisten der etwa 1 000 Beispiele stammen aus der Feder von Izaak Neutelings, mit Schwerpunkt auf Physik.

<https://TikZ.fr> baue ich gerade mit Patrick Bideault, Denis Bitouzé, und Alain Matthes auf, eine französischsprachige TikZ-Galerie mit einem Schwerpunkt auf tkz-Paketen wie tkz-euclide.

<https://pgfplots.net> hatte ich als begeisterter pgfplots-Anwender eingerichtet als eine Galerie ausgewählter und verschiedenster Plots.

<https://LaTeX-Cookbook.net> ist eine Galerie von vorwiegend grafischen Beispielen aus meinem »L^AT_EX Cookbook« von 2015. Dort liegen alle Code-Beispiele des Buchs online editierbar und übersetzbar.

<https://LaTeXguide.org> ist wie beim Cookbook die Begleitseite zum »L^AT_EX Beginner's Guide« von 2011, anlässlich der Neuauflage in 2021 mit erneuerten Codes und allen Beispielen und Zusatzinformationen.

Dank

Ich danke DANTE e.V. für die Finanzierung der Technik, die das hier Beschriebene möglich macht und eine erhebliche Motivation darstellt.

Bretter, die die Welt bedeuten

Having Fun with L^AT_EX: Klein- und großkariert

Adelheid Bonnetsmüller

Während meiner Studienzeit war es noch üblich, in Vorlesungen das vom Dozenten auf die Tafel Geschriebene händisch mitzuschreiben. Skripte gab es nur im Ausnahmefall, Präsentationen bei uns gar nicht. Dafür war natürlich viel Papier nötig. Meistens hab ich meine Notizen auf einfachem, weißen Papier verewigt. Manchmal war es aber auch nötig, kariertes oder liniertes Papier zu haben. Das konnte man damals schon selber machen und ausdrucken, wenn mal Not an Papier war. Ob dies heute noch zeitgemäß ist, kann ich schwer beurteilen. Was aber mit Sicherheit auch heute noch gebraucht wird, sind verschiedene Spezialgitterpapiere, wie logarithmische Papiere, Polarkoordinatenpapiere oder das bei Ingenieuren sehr beliebte Smithchart. Hier sollen zwei Möglichkeiten vorgestellt werden, wie man solche Gitter bzw. Papiere einfach mit L^AT_EX gestalten kann.

Übersicht

Es gibt zwei Pakete, um kariertes oder sonstwie strukturiertes Papier zu erzeugen. Das eine, `gridpapers`, ist ein »reguläres« Paket, das andere, `graphpaper`, ist eine Dokumentenklasse.

Das Paket `gridpapers`

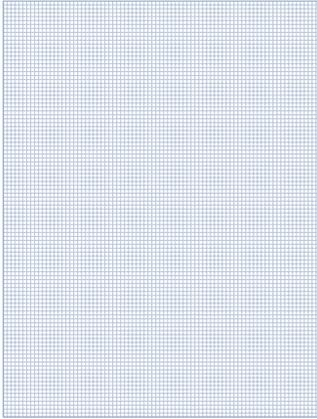
Das Paket `gridpapers` wird wie üblich über `\usepackage{gridpapers}` in der Präambel des Dokuments geladen. Es gibt verschiedene Optionen, die beim Laden angegeben werden können:

Option	Bedeutung
<code>pattern={}</code>	gibt die Art des Hintergrundmusters an.
<code>colorset={}</code>	Definition eines Farbschemas.
<code>majorcolor={}</code>	Hauptfarbe

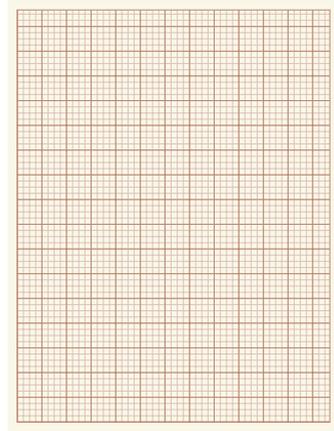
<code>minorcolor={}</code>	Nebenfarbe
<code>bgcolor={}</code>	Hintergrundfarbe
<code>patternsiz={}</code>	Größe des Musters, ergibt unterschiedliche Wert je nach vordefiniertem pattern.
<code>dotsiz={}</code>	Größe der Punkte bei <code>pattern=dot</code> .
<code>fullpage={}</code>	Muster soll über gesamte Seite gehen.
<code>textarea={}</code>	Nur der Textbereich soll mit dem Muster gefüllt werden.
<code>geometry={}</code>	Angabe eines Bereichs, der mit dem Muster gefüllt werden soll.

Im Folgenden werden die möglichen Muster (`pattern`) näher erläutert. Es sei erwähnt, dass sich die Standardwerte auf die englischen Längennormen beziehen (`inch`). Es gibt leider keine `pattern`-Einstellung für den Fall, dass man pro Längeneinheit 10 Kästchen und eine dicke Linie nach 5 Kästchen haben möchte.

Wert für <code>pattern={}</code>	Bedeutung
<code>std</code>	quadratisch, 10 Kästchen pro <code>inch</code> . Mit <code>patternsiz</code> kann die Größe des Kästchens definiert werden, Standard ist <code>0.1inch</code> .
<code>stdeight</code>	quadratisch, 8 Kästchen pro <code>inch</code> . Mit <code>patternsiz</code> kann die Größe des Kästchens definiert werden, Standard ist <code>0.125inch</code> .
<code>majmin</code>	quadratisch, 8 Kästchen pro <code>inch</code> mit einer dicken Linie nach jeweils einem halben <code>inch</code> . Mit <code>patternsiz</code> kann die Größe des Kästchens definiert werden, Standard ist <code>0.125inch</code> .
<code>dot</code>	Punktgitter. <code>dotsiz</code> setzt die Größe der Punkte. + <code>patternsiz</code> die Kästchengröße, Standardwert ist <code>0.125inch</code> .
<code>hexup</code>	Hexagonales Gitter mit 90° -Rotation. <code>patternsiz</code> kontrolliert hier die Seitenlänge der Hexagone, Standard ist <code>0.25inch</code> .
<code>tri</code>	Dreiecksgitter. <code>patternsiz</code> kontrolliert hier die Seitenlänge der Dreiecke, Standard ist <code>0.25inch</code> .
<code>iso</code>	isometrisches Gitter. <code>patternsiz</code> kontrolliert hier die Seitenlänge der Dreiecke, Standard ist <code>0.25inch</code> .
<code>lightcone</code>	Gitter mit zusätzlichen Linien im 45° -Winkel. <code>patternsiz</code> kontrolliert hier die Länge eines horizontalen Kästchens, Standard ist <code>0.25inch</code> .
<code>ruled</code>	liniertes Papier. <code>patternsiz</code> kontrolliert hier den Abstand der Linien, Standard ist <code>0.2inch</code> .



pattern=std,colorset=std



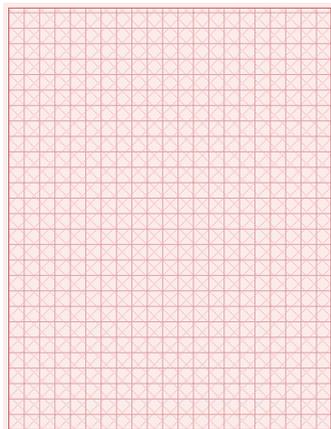
pattern=majmin,colorset=precocious,
patternsize=0.4cm

`doubleruled` liniertes Papier mit alternierenden dicken und dünnen Linien. `patternsize` gibt die Länge zwischen benachbarten Linien an, Standard ist 0.125inch.

Folgende Farbmuster (`colorset`) sind vordefiniert. Diese setzen die drei Farbwerte `majorcolor`, `minorcolor` und `bgcolor`. Wird für einen Wert eine andere Farbe gewählt, bleiben die anderen beiden Werte dennoch bestehen.

Wert für <code>colorset</code>	Farbige Darstellung
<code>std</code>	Hintergrund weiß, Linien blau
<code>precocious</code>	Hintergrund gelb, Linien braun
<code>ghostly</code>	Hintergrund weiß, Linien hellgrau
<code>brickred</code>	verschiedene Rottöne
<code>engineer</code>	verschiedene Grüntöne
<code>plumpad</code>	verschiedene Blautöne

Beispiele für die Verwendung von `gridpapers` sind in den Abbildungen auf dieser und der folgenden Seite zu finden.



`pattern=lightcone,patternsize=1cm,
colorset=brickred`



`pattern=doubleruled,patternsize=1cm,
colorset=engineer`

Das »Paket« graphpaper

Die Dokumentenklasse `graphpaper` kann aber deutlich mehr, als einfach liniertes oder kariertes Papier zu erzeugen. Mit ihr ist es möglich, auch einfach oder doppelt logarithmisches sowie Papier für Polarkoordinaten oder Smithcharts zu erzeugen. Die Paketklasse hat die Optionen `a4paper`, `a3paper` oder `letterpaper`. Das Paket nutzt Querformat – und das lässt sich auch nicht ändern! Nach dem Erzeugen eines grafischen Papiers wird automatisch eine neue Seite angelegt.

Es stehen folgende Kommandos innerhalb der Klasse zur Verfügung:

Kommando	erzeugt was?
<code>\bilinear</code>	Millimeterpapier
<code>\semilogx{Anzahl Dekaden x}</code>	einfach logarithmisches Papier (x -Achse)
<code>\semilogy{Anzahl Dekaden y}</code>	einfach logarithmisches Papier (y -Achse)
<code>\loglog[]{Dekaden x}{Dekaden y}</code>	doppeltlogarithmisches Papier. Die Option gibt an, ob die Größe der Dekade in beide Richtungen gleich (1) oder unterschiedlich (0) sein soll. Im zweiten Fall wird das doppeltlogarithmische Gitter über die gesamte Papierbreite gezogen.

<code>\polar</code>	Polarkoordinatenpapier
<code>\logpolar{Anzahl Dekaden Radius}</code>	logarithmisches Polarkoordinatenpapier
<code>\smith</code>	das legendäre Smithchart

Man kann auch hier das Aussehen der grafischen Darstellung verändern. Die Kommandos müssen (logischerweise) vor dem jeweiligen Papier-Erzeugungskommando stehen. Folgende Möglichkeiten gibt es:

Kommando	erzeugt was?
<code>\setxside{Länge}</code>	horizontale Abmessung des Gitters, im Standard angepasst an das Papierformat
<code>\setyaside{Länge}</code>	vertikale Abmessung des Gitters, im Standard angepasst an das Papierformat
<code>\setminimumdistance{Länge}</code>	minimaler Abstand zwischen zwei Linien (Standard: 1mm). Passt die Untereinheiten automatisch an den Wert an. Greift nicht bei beiden Polarkoordinatenpapieren und dem Smithchart.
<code>\setgridcolor{Farbe}</code>	Farbe des Gitters (Standard: rot-orange)
<code>\setmajorlinethickness{Länge}</code>	Strichstärke der dicken Linien (Standard: 1pt)
<code>\setmediumlinethickness{Länge}</code>	Strichstärke der mittleren Linien (Standard: 0.5pt)
<code>\setminorlinethickness{Länge}</code>	Strichstärke der dünnen Linien (Standard: 0.25pt)
<code>\costumcode[]{ }</code>	Kommandos aus <code>picture</code> -Umgebung können angegeben werden, z.B. um ein Logo zu platzieren. Optionales Argument kann 0 oder 1 sein. Wenn auf 1 gesetzt, wird der Inhalt des Kommandos nach dem ersten Aufruf gelöscht.

Die acht Beispiele für graphpaper wurden folgendermaßen erzeugt. Der hohe Wert von 20 mm für den Abstand zwischen den Linien beim Millimeterpapier und den einfach logarithmischen Papieren dient lediglich der Darstellbarkeit beim Druck in dieser Ausgabe.

```

\documentclass{graphpaper}
\begin{document}
\setminimumdistance{20mm}
\semilogx{10} %%% einfach logarithmisches Papier (x)

\setminimumdistance{20mm}
\setgridcolor{purple} %%% Gitterfarbe = Purpur
\semilogy{10} %%% einfach logarithmisches Papier (x)

\setminimumdistance{1mm}
\loglog{15}{5} %%% doppelt logarithmisches Papier
\setminimumdistance{20mm}
\loglog{15}{5} %%% doppelt logarithmisches Papier

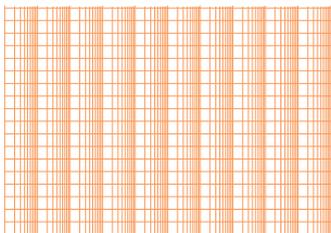
\setminimumdistance{20mm}
\loglog[0]{15}{5} %%% doppelt logarithmisches Papier, Demonstration Option 0
\setminimumdistance{20mm}
\loglog[1]{15}{5} %%% doppelt logarithmisches Papier, Demonstration Option 1

\setminimumdistance{20mm}
\bilinear. %%% Millimeterpapier

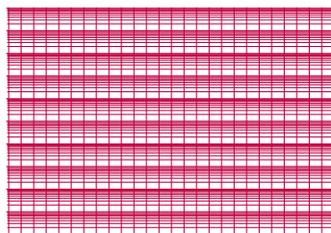
\polar %%% Polarkoordinaten
\logpolar{2} %%% Polarkoordinaten, logarithmisch

\customcode{
  \put(-65,60){{\color{green!40!black!40}\circle*{30}}}
  \put(-65,60){{\color{white}\circle*{12}}}
  \put(-65,60){{\color{green!40!black!40}\circle*{3}}}
}
\smith %%% Smithchart mit Bild über customcode
\end{document}

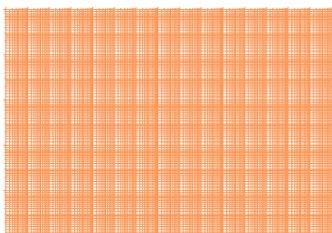
```



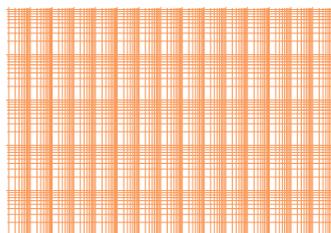
einfach logarithmisches Papier (x)



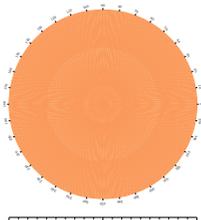
einfach logarithmisches Papier (y), purpurfarbene Gitterlinien



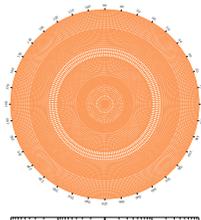
doppelt logarithmisches Papier, minimaler Abstand zwischen den Linien von 1mm



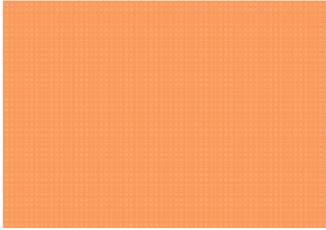
doppelt logarithmisches Papier, minimaler Abstand zwischen den Linien 20mm



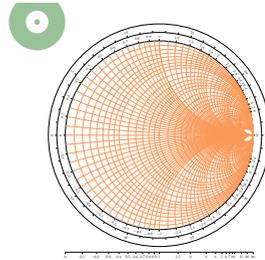
Polarkoordinatenpapier



logarithmisches Polarkoordinatenpapier



Millimeterpapier



Smithchart mit eingefügtem Bild über customcode

Quintenzirkel mit Gitarren-Akkorden

Ralf Mispelhorn

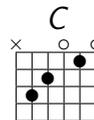
Es gibt mehrere Pakete, mit welchen Gitarren-Akkorde dargestellt werden können. In diesem Artikel wird der Quintenzirkel mit den Dur-Akkorden für Gitarre mit dem Paket `songs` erzeugt.

Akkordsymbole

Zum Darstellen der Akkordsymbole wird das Macro `\gtab` aus dem Paket `songs` verwendet.

Beispiel:

```
\gtab{C}{X32010}
```



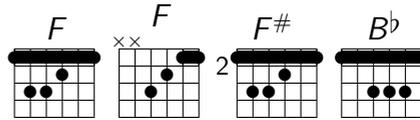
Der erste Parameter ist der Name des Akkords. Der zweite Parameter zeigt die Positionen an, an denen die Saiten gegriffen werden. Ein X bedeutet dabei, dass die Saite nicht verwendet wird und beim Greifen abgedämpft bzw. nicht angeschlagen wird.

Im Symbol ist die tiefe E-Saite ganz links (oben auf dem Griffbrett) und die hohe e-Saite ganz rechts (unten auf dem Griffbrett). Die Positionen innerhalb des zweiten Parameters von `\gtab` sind wie das gezeigte Symbol von links nach rechts.

Barré-Akkorde sind Akkorde, bei denen ein Finger mehrere Saiten greift. Sie werden definiert, indem ein Teil in runde Klammern gesetzt wird. Die kleinste Nummer

bestimmt dabei den Barré-Anteil. Mit einer vorangestellten Ziffer kann der Bund angegeben werden. Außerdem kann mit Hilfe von `\fl` das Zeichen \flat und mit `\shrp` das Zeichen \sharp erzeugt werden.

```
\gtab{F}{(133211)}
\gtab{F}{XX32(11)}
\gtab{F\shrp}{2:(133211)}
\gtab{B\fl}{(113331)}
```

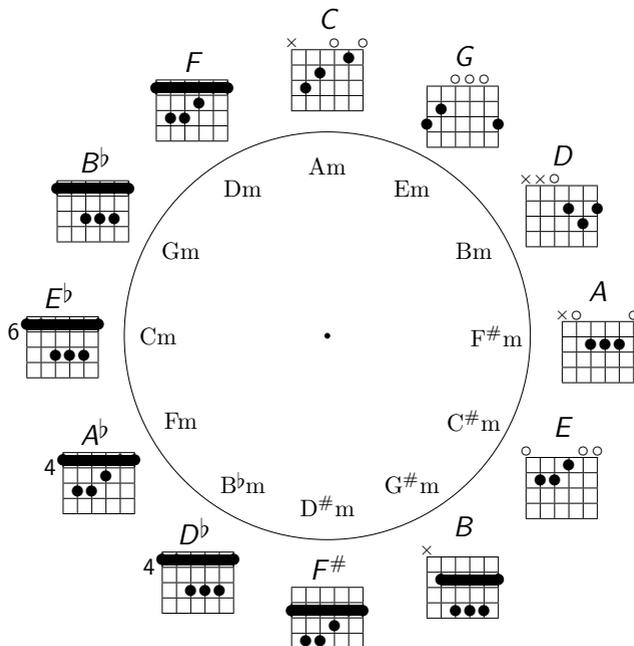


Quinten-Zirkel

Im Quintenzirkel sind alle 12 Tonarten der Reihe nach aufgeführt. Die Anzahl und Art der Vorzeichen bestimmt dabei die Reihenfolge auf dem Quintenzirkel.

Ausgehend von C-Dur, welches kein Vorzeichen hat, werden nach rechts die Tonarten mit dem Vorzeichen \sharp und nach links die Tonarten mit dem Vorzeichen \flat aufgeführt.

Als Gitarrist lohnt es sich diesen Quintenzirkel im Kopf zu haben, weil eine Liedbegleitung sich dabei schneller erschließt. Ein Lied in C-Dur hat in der Regel die Töne c,d,e,f,g,h,c und wird meistens mit dem C-Dur-Akkord (Tonika) begonnen. Der C-Dur-Akkord beinhaltet die Töne c,e und g, welche alle in der Tonleiter enthalten sind. Außerdem finden meistens die benachbarten Akkorde Verwendung. Die Subdominante F-Dur (f,a,c) und die Dominante G-Dur (g,h,d), deren Töne sich ebenfalls alle auf der C-Dur-Tonleiter befinden. Öfter werden auch die Parallel-Moll-Tonarten Am (a,c,e), Dm (d,f,a) und Em (e,g,h) verwendet. Hier sind auch alle Töne in der C-Dur-Tonleiter. Für Lieder in anderen Tonarten gilt entsprechendes.



```

\begin{tikzpicture}
\draw (0,0) circle (3cm);
\draw[fill] (0,0) circle (1pt);
\node at (240:4cm) {\gtab{D\flt}{(446664)}};
\node at (210:4cm) {\gtab{A\flt}{4:(466544)}};
\node at (180:4cm) {\gtab{E\flt}{6:(668886)}};
\node at (150:4cm) {\gtab{B\flt}{(113331)}};
\node at (120:4cm) {\gtab{F}{(133211)}};
\node at (90:4cm) {\gtab{C}{X32010}};
\node at (60:4cm) {\gtab{G}{320003}};
\node at (30:4cm) {\gtab{D}{XX0232}};
\node at (0:4cm) {\gtab{A}{X02220}};
\node at (330:4cm) {\gtab{E}{022100}};
\node at (300:4cm) {\gtab{B}{X(24442)}};
\node at (270:4cm) {\gtab{F\shrp}{(244322)}};

\node at (180:2.5) {Cm}; \node at (150:2.5) {Gm};
\node at (120:2.5) {Dm}; \node at (90:2.5) {Am};
\node at (60:2.5) {Em}; \node at (30:2.5) {Bm};
\node at (0:2.5) {F\shrp{m}}; \node at (330:2.5) {C\shrp{m}};

```

```

\node at (300:2.5) {G\shrp{}}m}; \node at (270:2.5) {D\shrp{}}m};
\node at (240:2.5) {B\flt{}}m}; \node at (210:2.5) {Fm};
\end{tikzpicture}

```

Erstellen eines Songbooks

Ralf Mispelhorn

Neben meiner Tätigkeit als Software-Ingenieur und Lehrbeauftragter spiele ich in der Freizeit Gitarre und hatte auch schon ein paar kleinere Auftritte.

Im Laufe von mehreren Jahren sind viele Liedertexte gesammelt und auch einige Lieder von mir selbst geschrieben worden. Es entstand dabei der Wunsch, diese Liedertexte als Songbook mit einem schönen Seitenlayout zu erstellen.

In diesem Zusammenhang wurde ein Python-Skript geschrieben, das die einzelnen Songtexte, die als Word-Dateien vorlagen, in ein $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Dokument konvertiert und mit Gitarren-Akkord-Symbolen versieht. Hierzu wurde das Paket `songs` verwendet.

Um Copyright-Probleme zu vermeiden, wird im Artikel zur Demonstration ein von mir geschriebener Song verwendet.

Word-Datei

Die einzelnen Songs liegen im Word-Format vor. Der Aufbau der Songtexte ist immer gleich. Als Erstes steht dort das Datum der Erstellung und danach der Titel des Songs. Danach folgen die Strophen, bei denen in der ersten Strophe zusätzlich die Akkorde für die Gitarre in der Zeile darüber vermerkt sind. Nach dem Songtext ist eventuell ein Bild, welches hier nicht weiter betrachtet wird. Am Schluß können Hinweise zur Instrumentierung, wie Position des Capos, und zu den Effekten folgen. Diese werden vom Rest durch mehrere Minuszeichen getrennt und bei der Generierung ebenfalls nicht betrachtet.

Songbook-Format

Das Songbook wird mit Hilfe des Paketes `songs` gesetzt. Hierzu gibt es die Umgebung `songs` zur Gruppierung mehrerer Songs, innerhalb werden die Stücke in der Umgebung `song` eingefasst. Strophen werden zwischen `\beginverse` und `\endverse`, der Chorus zwischen `\beginchorus` und `\endchorus` gesetzt.

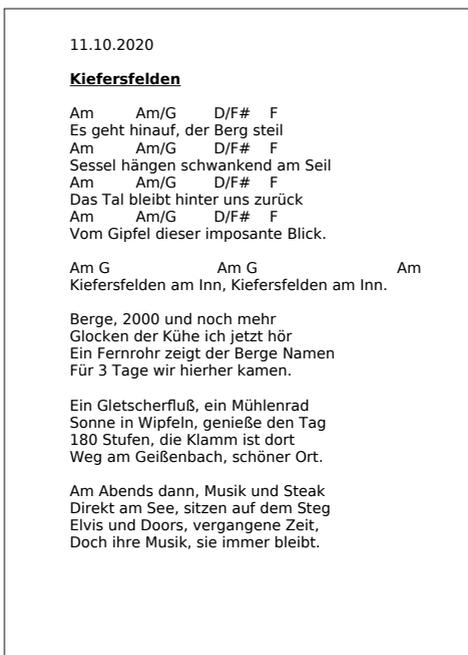


Abb. 1: Word-Dokument

```

1 \beginverse
2 \[Am]Es geht hin\[Am/G]auf, der Ber\[D/F#]g steil \[F] ~~~~~
3 \[Am]Sessel häng\[Am/G]en schwanken\[D/F#]d am Sei\[F]l~~~~~
4 \[Am]Das Tal ble\[Am/G]ibt hinter u\[D/F#]ns zurück\[F]k~~~~~
5 \[Am]Vom Gipfel \[Am/G]dieser impos\[D/F#]ante Bli\[F]ck.~~~~~
6 \endverse
7 \beginchorus
8 \[Am]Kie\[G]ffersfelden am Inn, Kiefe\[Am]rsf\[G]elden am Inn. \[
  ↪Am] ~~~~~
9 \endchorus

```

Mit Hilfe von `\chordson` können die Akkorde über dem Text ein- und mit `\chordsoff` abgestellt werden. Außerdem kann die Seitenaufteilung eingestellt werden. Hier wurde die einspaltige Form mit Hilfe von `\chordscolumns{1}` gewählt. Die Schrift kann für jeden Teil durch Neudefinition einiger Makros geändert werden.

```

1 \usepackage[chorded]{songs}
2 \songcolumns{1}
3 \beforepostludeskip=30pt plus 20fil

```

```

4 \renewcommand{\printchord}[1]{\sffamily\color{red}#1}
5 \renewcommand{\lyricfont}{\sffamily\small}

```

Akkorde im Text

Die Akkorde innerhalb des Textes werden angegeben, indem beispielsweise für den Am-Akkord `\[Am]` geschrieben wird. Dies führt dazu, dass der Akkord-Name oberhalb des Textes erscheint.

```

7 \usepackage{gchords}
8 \usepackage{guitar}
9 \usepackage[chorded]{songs}

13 \begin{songs}{}
14 \begin{song}{Kiefersfelden}[cr={\copyright~Ralf~
  ↳Mispelhorn}]
15 \beginverse
16 \[Am]Es geht hin\[Am/G]auf, der Ber\[D/F#]g
  ↳steil \[F] ~~~~~
17 \[Am]Sessel häng\[Am/G]en schwanken\[D/F#]d am
  ↳Sei\[F]l~~~~~
18 \[Am]Das Tal ble\[Am/G]ibt hinter u\[D/F#]ns
  ↳zurüc\[F]k~~~~~
19 \[Am]Vom Gipfel \[Am/G]dieser impos\[D/F#]ante
  ↳Bli\[F]ck.~~~~~
20 \endverse
21 \end{song}
22 \end{songs}

```

1 Kiefersfelden

1. Am Es geht hinauf, Am/G der Berg $D/F^\#$
 F
 steil
 Am Sessel hängen Am/G
 $D/F^\#$ schwankend F am Seil -
 Am Das Tal bleibt Am/G hinter uns $D/F^\#$
 F
 zurück -
 Am Vom Gipfel dieser Am/G imposante $D/F^\#$
 F
 Blick.

© Ralf Mispelhorn

Akkord-Griff-Bilder

Mit Hilfe des Makros `\gtab{name}{finger}` können Griffbilder dargestellt werden.

```
\gtab{Am}{X02210}
```



Im Artikel »[Quintenzirkel mit Gitarren-Akkorden](#)« auf Seite 18ff wird das genaue Format der Akkorde beschrieben.

Beim Parsen des Songtextes wurden die Akkorde jedes Songs getrennt gesammelt und diese Liste der Akkorde bei jedem Song ausgegeben.

Python-Skript

Das Python-Skript <http://www.mispelhorn.de/mispelsoft/songbook/songbook.py>.txt erstellt das L^AT_EX-Dokument.

Alle Akkorde mit ihrem Fingersatz (Griffbilder) werden in einem Dictionary gespeichert. Die benötigten Akkorde wurden durch das Parsen der Word-Dokumente in einem vorangehenden Schritt ermittelt und dann im Python-Skript definiert. Hier werden nur die für mein Lied verwendeten Akkorde gezeigt. Im vollständigen Skript sind erheblich mehr Akkorde definiert worden.

```

1 # Chord Tabelle
2 chordtable = {}
3
4 def chord_insert ( name, finger ):
5     chordtable[name] = finger
6
7 chord_insert('Am', 'X02210' )
8 chord_insert('Am/G', '302210' )
9 chord_insert('G', '320003' )
10 chord_insert('D', 'XX0232' )
11 chord_insert('D/F#', '2X0232' )
12 chord_insert('C', 'X32010' )
13 chord_insert('F', 'XX32(11)' )

```

Zuerst wird das Erstellen des L^AT_EX-Dokuments vorbereitet, indem die Datei sb.tex zum Schreiben geöffnet wird. Eine Schleife geht rekursiv durch alle Verzeichnisse. Jedes Word-Dokument (*.docx) wird dann durch die Funktion process_file() bearbeitet.

```

1 textfile = open ( 'sb.tex', 'w' )
2
3 start_document()
4
5 for root,dirs,files in os.walk ( '.' ) :
6     for f in files:
7         fn = os.path.join ( root, f )
8         if fn.endswith ( '.docx' ):
9             print ( f'{fn}' )
10            process_file ( fn )
11
12 end_document()
13
14 textfile.close()

```

Zuerst wird die Word-Datei entpackt. Da dieses Skript ausschließlich den Text benötigt, wird nur aus der Datei `word/document.xml` mit einem XML-Parser der Dokumentenbaum erzeugt.

```

1  import zipfile
2  from xml.etree.cElementTree import XML
3  PARA = '{http://schemas.openxmlformats.org/wordprocessingml/2006/main}p'
4  TEXT = '{http://schemas.openxmlformats.org/wordprocessingml/2006/main}t'
5
6  # unzip docx
7  document = zipfile.ZipFile(fn)
8  xml_content = document.read('word/document.xml')
9  document.close()
10
11 # parse XML
12 tree = XML(xml_content)

```

Anschließend wird der Text extrahiert, indem über alle Knoten des Dokumentenbaums iteriert wird und die Paragraphen- und Text-Knoten ausgelesen werden.

```

1  # get text from docx
2  paragraphs = []
3  for para in tree.iter(PARA):
4      texts = [node.text for node in para.iter(TEXT) if node.text]
5      paragraphs.append(''.join(texts))
6  text = '\n'.join(paragraphs)

```

Nun werden alle Zeilen ermittelt, in denen Akkorde stehen. Diese werden in einer Liste gesammelt, wobei jeder Akkord nur einmal in die Liste kommt. Außerdem wird geprüft, ob der Akkord in dem anfänglich erzeugten Dictionary der Akkorde mit den Griff-Bilder enthalten ist. Fehlt er, wird mit einer Fehlermeldung abgebrochen.

Auf diese Weise wurden alle insgesamt benötigten Akkorde ermittelt bzw. unterschiedliche Schreibweisen erkannt und korrigiert.

```

1  # find chords
2  chord = []
3  for para in text.split('\n\n'):
4      for line in para.split('\n'):
5          if is_chord ( line ):
6              for word in line.split():
7                  if word not in chord:
8                      chord.append (word)
9                  if word not in chordtable:
10                     print ( 'chord %s not in table' % word )

```

```
11 sys.exit(1)
```

Nach diesen Vorbereitungen wird der Text eines Songs gelesen. Mit ein paar Hilfsfunktionen werden bestimmte Teile des Textes erkannt und Sonderbehandlungen durchgeführt.

Außerdem werden die Absätze gezählt und innerhalb der Absätze die Zeilen.

Sobald eine vollständige Strophe erkannt wurde, wird das Erzeugen der Akkorde beendet. Damit enthält nur die erste Strophe die Akkorde.

Die Anzahl der Zeilen bestimmt hier, ob es eine Strophe oder ein Refrain ist. Dies ist keine besonders gute Lösung, hat aber in den meisten Fällen ein Ergebnis gebracht, welches für das Songbook in Ordnung war.

Wird eine Zeile mit Akkorden gefunden, dann werden diese und ihre Position innerhalb der Zeile vermerkt. In der darauffolgenden Textzeile werden die Akkorde dann anhand der Position in die Textzeile eingebaut. Damit diese für alle Akkorde stimmt, wird deren Bestimmung vom Ende der Zeile aus durchgeführt.

Weil die Positionierung manchmal nicht funktionierte oder weil die Positionen für den Text zu groß waren, wurden alle Textzeilen um mehrere Leerzeichen ~ erweitert.

```

1  paraNr = 0
2  chordsFlag = True # Akkorde ausgeben
3  verseFlag = 0
4  for para in text.split('\n\n'):
5      lineNr = 0
6      chordpos = []
7      lines = para.split('\n')
8      linesAnz = len ( lines )
9      if linesAnz >= 4 and paraNr > 1:
10         chordsFlag = False # Akkorde nicht ausgeben
11         print_line ( r'\chordsoff' )
12     for line in lines:
13         line = line.strip()
14         if is_date ( line ):
15             continue
16
17         if is_chord ( line ):
18             for i in re.finditer ( r'(\S+)', line ):
19                 chordpos.append ( (i.start(), i.group()) )
20             line = line.strip()
21             ccl = len(line)
22             continue

```

```

23
24     if is_end ( line ):
25         if lineNr:
26             if verseFlag == 1: end_verse()
27             if verseFlag == 2: end_chorus()
28         versFlag = 0
29         end_song()
30         return
31
32     if line:
33         line = line.replace ( '#', '-' )
34         if lineNr==0:
35             if paraNr==0:
36                 start_song (line)
37                 for c in chord:
38                     print_line ( r'\gtab{%s}{%s}' % (c, chordtable[c]) )
39             paraNr += 1
40             if paraNr > 1:
41                 if linesAnz < 3:
42                     start_chorus()
43                     verseFlag = 2
44                 else:
45                     start_verse()
46                     verseFlag = 1
47             if chordsFlag and chordpos:
48                 if len(line) < ccl: line = line.ljust(ccl)
49                 for i in chordpos[::-1]:
50                     pos = i[0]
51                     ch = i[1]
52                     ch2 = r'\[%s]' % ch
53                     line = line[:pos] + ch2 + line[pos:]
54             if paraNr > 1: print_line ( line + '~~~~~' )
55             chordpos = []
56             lineNr += 1
57         if lineNr:
58             if paraNr > 1:
59                 if verseFlag == 1: end_verse()
60                 if verseFlag == 2: end_chorus()
61             versFlag = 0

```

Hilfsprogramme

Die beiden Hilfsfunktionen `start_document()` und `end_document()` schreiben den Anfang und das Ende der L^AT_EX-Dokuments.

```

1  def start_document ():
2      print_line ( r'\documentclass[11pt,a4paper,DIV=15]{scrartcl}' )
3      print_line ( r'\usepackage[ngerman]{babel}' )
4      print_line ( r'\pagestyle{empty}' )
5      print_line ( r'\usepackage[top=3cm,bottom=3cm]{geometry}' )
6      print_line ( r'\usepackage[chorded]{songs}' )
7      print_line ( r'\songcolumns{1}' )
8      print_line ( r'\beforepostludeskip=30pt plus 20fil' )
9      print_line ( r'\renewcommand{\printchord}[1]{\sfamily\color{red}#1}' )
10     print_line ( r'\renewcommand{\lyricfont}{\sfamily\small}' )
11     print_line ( r'\begin{document}' )
12     print_line ( r'\begin{songs}{}' )
13
14     def end_document ():
15         print_line ( r'\end{songs}' )
16         print_line ( r'\end{document}' )

```

Die beiden Hilfsfunktionen `start_song()` und `end_song()` schreiben den Anfang und das Ende jedes Songs.

```

1  def start_song (title):
2      print_line ( r'\begin{song}{%s}[cr={\copyright~Ralf~Mispelhorn}]' % title )
3
4  def end_song ():
5      print_line ( r'\end{song}' )
6      print_line ( r'\sclearpage' )

```

Die beiden Hilfsfunktionen `start_verse()` und `end_verse()` schreiben den Anfang und das Ende jeder Strophe.

```

1  def start_verse ():
2      print_line ( r'\beginverse' )
3
4  def end_verse ():
5      print_line ( r'\endverse' )

```

Die beiden Hilfsfunktionen `start_chorus` und `end_chorus` schreiben den Anfang und das Ende jedes Refrains.

```

1  def start_chorus ():
2      print_line ( r'\beginchorus' )

```

```

3
4 def end_chorus ():
5     print_line ( r'\endchorus' )

```

Weiterhin gibt es ein paar Hilfsfunktionen, um bestimmte Stellen im Text zu erkennen.

`is_date()` erkennt die Datumszeile und verwirft sie.

```

1 def is_date ( line ):
2     return re.search ( r'^\d+\.\d+\.\d+', line)

```

`is_chord()` erkennt die Zeile mit Akkorden.

```

1 def is_chord ( line ):
2     r = r'^[AaBbCcDdEeFfGgHh][b#]?m?[79]?[+]?(/[AaBbCcDdEeFfGgHh])?[#]?$'
3     for word in line.split():
4         if not re.search ( r, word ): return False
5     if line: return True
6     return False

```

`is_end()` erkennt die Zeile, die das Ende des Songtextes kennzeichnet.

```

1 def is_end ( line ):
2     if re.search ( r'Eff', line ): return True
3     return re.search ( r'^[=-]+\s*$', line )

```

Das fertige Songbook:

Zum Schluß

Das Skript hat sich inzwischen sehr gut bewährt und wurde auch noch stark erweitert, um zusätzliche Informationen in das Songbook zu bringen (Capo-Position, CD-Nummer und mehr). Die Anzahl der Songtexte ist mittlerweile ziemlich angewachsen, es sind mehrere tausend Songs, und das Skript erzeugt das gesamte Songbook innerhalb weniger Minuten.

1 *Kiefersfelden*



1. ^{Am} Es geht hinauf, ^{Am/G} der Berg ^{D/F#} ^F steil
^{Am} Sessel hängen ^{Am/G} schwankend ^{D/F#} ^F am Seil
^{Am} Das Tal bleibt ^{Am/G} hinter uns ^{D/F#} ^F zurück
^{Am} Vom Gipfel dieser ^{Am/G} imposante ^{D/F#} ^F Blick.

^{Am} ^G Kie - fersfelden am Inn, ^{Am} ^G Kiefersf - elden am Inn. ^{Am}

2. Berge, 2000 und noch mehr
 Glocken der Kühe ich jetzt hör
 Ein Fernrohr zeigt der Berge Namen
 Für 3 Tage wir hierher kamen.

3. Ein Gletscherfluß, ein Mühlenrad
 Sonne in Wipfeln, genieße den Tag
 180 Stufen, die Klamm ist dort
 Weg am Geißenbach, schöner Ort.

4. Am Abends dann, Musik und Steak
 Direkt am See, sitzen auf dem Steg
 Elvis und Doors, vergangene Zeit,
 Doch ihre Musik, sie immer bleibt.

Abb. 2: Das Ergebnis des Songbooks.

Erstellen, Ausführen und Einbinden der Ausgaben externer Dateien

Herbert Voß

Beim Schreiben von Fachbüchern werden sehr häufig Ausgaben von Programmen angegeben. In vielen Fällen auch mit kompletter oder teilweiser Angabe des jeweiligen Quellcodes. Als Autor hat man dabei das Problem, die externen Beispielprogramme mit dem aktuellen Manuskript immer synchron zu halten. Erstellt man die externen Beispiele jedoch aus dem Manuskript selbst, dann kann man sicher gehen, dass die im Hauptdokument eingefügte Ausgabe des Beispiels auch synchron zum aktuell angegebenen Quellcode ist.

Einführung

Schreibt man mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ein Buch über $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, so lassen sich die Ausgaben der Beispiele einfach direkt in das Hauptdokument einfügen. [2] Eine Gewissheit, dass

die Beispiele auch in kleinen Einzeldokumenten funktionieren, hat man dadurch nicht zwingend. Alle Beispiele in einem größeren Buch verwenden die Präambel des Hauptdokuments, die einem Leser des Buches nicht zur Verfügung steht.

Es erscheint sinnvoller, alle Beispiele als eigenständige und vom Hauptdokument unabhängige Dokumente oder Programme zu erstellen. Dazu wird aus dem Hauptdokument heraus der vollständige Quellcode in eine externe Datei geschrieben, diese dann mit einem vorzugebenden Programm übersetzt und das Ergebnis als PDF, PNG oder auch in Textform wieder in das Hauptdokument eingebunden. Vom gesamten Quellcode des Beispiels kann man, über Marker gesteuert, die *wesentlichen* Codezeilen in das Hauptdokument vor oder neben der Beispielausgabe setzen.

Zum besseren Verständnis soll ein kurzes Beispiel gezeigt werden: Die »Die \TeX -nische Komödie« wird mit $\text{Lua}\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ übersetzt, sodass man mit einem Beispiel, welches zwingend die Anwendung von $\text{X}\mathcal{Y}\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ voraussetzt, Probleme bekommt. Das Beispiel muss dann extern erstellt und die Ausgabe als PDF eingebunden werden. Sinnvoller ist es, wenn man dies gleich aus diesem Dokument heraus bewerkstelligen könnte und die Ausgabe, wie hier zu sehen 美好的一天 direkt einbindet. Ermöglicht wird dies durch das Paket `hvextern`, welches nur eine Umgebung und einen Befehl definiert. [5] Der entsprechende Code für obiges Beispiel lautet:

```
die Ausgabe, wie hier zu sehen \begin{externalDocument}{%
  compiler=xelatex, inline, runs=2, grfOptions={height=0pt},%
  crop, cropmargin=0, cleanup, docType=latex}{voss}
\documentclass{ctexart}
\pagestyle{empty}
\begin{document}
美好的一天
\end{document}
\end{externalDocument}
~direkt einbindet.
```

In diesem Beispiel wurde nur die Ausgabe eingebunden, ohne Angabe des Quellcodes. Jede Änderung in diesem Beispiel läuft immer synchron zum Hauptdokument. Beim nächsten Übersetzungsvorgang wird auch das Hauptdokument aktualisiert, dadurch der Code des Beispiels und somit auch die einzufügende Ausgabe. Je nach Anwendungsfall kann es wünschenswert sein, auch Teile des Quellcodes mit anzugeben, was auf den folgenden Seiten beschrieben wird.

Momentan unterstützt das Paket `hvextern` externe Dokumente für METAPOST , $\mathcal{T}\mathcal{E}\mathcal{X}$, $\text{Con}\mathcal{T}\mathcal{E}\mathcal{X}\mathcal{T}$, $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, $\text{Lua}\mathcal{T}\mathcal{E}\mathcal{X}$, $\text{Lua}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, $\text{X}\mathcal{Y}\mathcal{T}\mathcal{E}\mathcal{X}$, $\text{X}\mathcal{Y}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, Lua, Perl, Java und Python. *Alle* Ausgaben der folgenden Beispiele werden »on-the-fly« beim Übersetzen der DTK erzeugt. Jede Änderung im Beispielcode würde daher automatisch beim nächsten Übersetzungsvorgang zur veränderten Ausgabe führen.

Syntax

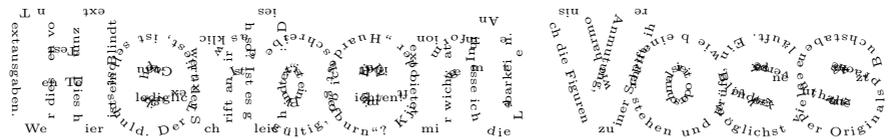
Das Paket wird wie üblich eingebunden: `\usepackage{hvextern}`. Paketoptionen gibt es keine und es wird nur eine einzige Umgebung definiert:

```
\begin{externalDocument}[<Optionen>]{Externer Dateiname ohne Endung}
... Quellcode ...
\end{externalDocument}
```

Das Hauptdokument muss *zwingend* mit der Option `--shell-escape` übersetzt werden, anderenfalls wird es zu keiner korrekten Ausgabe kommen.

```
lualatex --shell-escape <LaTeX-Datei>
```

Das folgende Beispiel muss mit der Sequenz `latex→dvips→ps2pdf` übersetzt werden; denn mit `XYLATEX` oder auch `LuaXLATEX` würde es nicht funktionieren. Mit der Umgebung `hvextern` kann man jedoch den kompletten Code in eine externe Datei schreiben, dann den notwendigen Ablauf wählen und das Ergebnis als PDF einbinden. Wichtig ist lediglich, dass beim Erstellen der Grafik die standardmäßige Ausgabe der Seitennummer unterdrückt und jeglicher Weißraum durch die Option `crop` abgeschnitten wird. Dies entfällt natürlich, wenn eine komplette Seite eingebunden werden soll (siehe Seite 41).



Der entsprechende Code für obige Ausgabe lautet:

```
\begin{externalDocument}[compiler=latex,crop,force=false,
  cleanup={log,aux,ps,dvi},grfOptions={width=\linewidth}]{voss}
\documentclass{article}
\usepackage[ngerman]{babel}
\pagestyle{empty}
\usepackage{pst-text,blindtext}
\begin{document}
\DeclareFixedFont{\SF}{T1}{phv}{b}{n}{2cm}
\pstextpath(0,-1ex){\pscharpath*[linestyle=none]{\SF Herbert Voss}}{%
  \tiny \blindtext}
\end{document}
\end{externalDocument}
```

Die Bedeutung der einzelnen Optionen:

compiler=latex Benutze zum Übersetzen `latex`. Die weitere Sequenz ist durch die interne Definition von `\hv@extern@runLATEX` festgelegt.

crop Den Weißraum mit pdfcrop abschneiden.
force In jedem Fall die Ausgabe neu erstellen, auch wenn sie schon existiert.
cleanup={log,aux,ps,dvi} Lösche am Ende die angegebenen Hilfsdateien.
grfOptions={width=\linewidth} Ausgabe auf Zeilenbreite skalieren.
voss Dateiname, der intern durch eine fortlaufende Nummer ergänzt wird.

Es spielt grundsätzlich keine Rolle, welche Programmiersprache verwendet wird, solange die Minimalkommunikation zwischen dem Hauptdokument und dem externen Programm gewährleistet ist. Diese besteht lediglich darin, dass das externe Dokument seine Ausgabe mit dem demselben Dateinamen versehen muss, mit dem es aufgerufen wurde. Dies ist aber für alle Programmiersprachen kein Problem, was hier an einigen Beispielen gezeigt wird.

Standardmäßig werden Quellcode und Ausgabe übereinander dargestellt, sodass auch ein Seitenumbruch im Quellcode unproblematisch ist. Im folgenden Beispiel wird ein Pythonprogramm erzeugt und ausgeführt und anschließend die Ausgabe als Datei im PNG-Format eingebunden. Der Kopf der Umgebung externalDocument lautet dabei:

```
\begin{externalDocument}[
  compiler=python3, showFilename, code, ext=py, docType=py,
  usefancyvrb, grfOptions={width=\linewidth}]{python}
... Python code ...
\end{externalDocument}
```

In den seltensten Fällen wird man den kompletten Quellcode mit ausgeben wollen. Deshalb können Bereiche über sogenannte Marker definiert werden, die dann ausgegeben werden. Die Marker werden als normale Kommentare in die externe Datei geschrieben, womit sie abhängig von der Programmiersprache sind; das Kommentarzeichen ist nicht einheitlich. Für Python existieren folgende Marker:

```
\hv@extern@ExampleType{py}
  {\NumChar StartVisibleMain}
  {\NumChar StopVisibleMain}
  {\NumChar StartVisiblePreamble}
  {\NumChar StopVisiblePreamble}
```

Und für plainT_EX:

```
\hv@extern@ExampleType{tex}
  {\perCent StartBody}
  {\string\bye}
  {\perCent StartVisiblePreamble}
  {\perCent StopVisiblePreamble}
```

`\perCent` and `\NumChar` sind für \TeX , beziehungsweise Python, die Kommentarzeichen, welche hier beide kodiert dargestellt werden müssen. Intern wird dazu die Kategorie des Zeichens verändert, sodass es über die Befehle `\NumChar` und `\perCent` als normales Zeichen zur Verfügung steht. Nach dem Aufrufen des Pythonprogrammes muss sichergestellt werden, dass der Dateiname ermittelt wird, um die Ausgabe mit demselben Haupt-Dateinamen und anderer Endung zu versehen. In Python ist dies mit folgendem Code möglich:

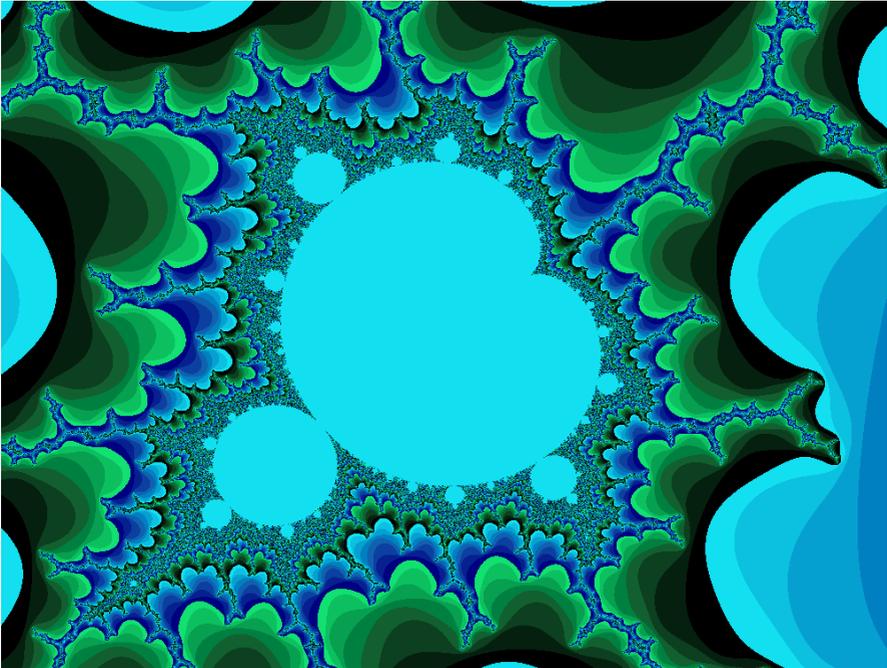
```
fileName = os.path.basename(os.path.splitext(__file__)[0]) # get filename
```

Je nach Ausgabeform wird dieser Dateiname durch `.pdf`, `.png` oder `.txt` ergänzt, sodass die Ausgabe problemlos in das \LaTeX -Hauptdokument eingefügt werden kann. Zusätzlich werden jetzt die Marker benutzt, sodass über das Schlüsselwort `code` die Ausgabe von Teilen des Python-Quellcodes genutzt werden kann.

```
from PIL import Image
import subprocess
# drawing area (xa < xb and ya < yb)
xa = -0.1716
xb = -0.1433
ya = 1.022
yb = 1.044
maxIt = 1024 # iterations
imgx = 1000 # image size
imgy = 750
image = Image.new("RGB", (imgx, imgy))
```

python-7.py

```
for y in range(imgy):
    cy = y * (yb - ya) / (imgy - 1) + ya
    for x in range(imgx):
        cx = x * (xb - xa) / (imgx - 1) + xa
        c = complex(cx, cy)
        z = 0
        for i in range(maxIt):
            if abs(z) > 2.0: break
            z = z * z + c
        r = i % 4 * 6
        g = i % 8 * 32
        b = i % 16 * 16
        image.putpixel((x, y), b * 65536 + g * 256 + r)
```



Rein formal kann die Ausgabe des Quellcodes in Analogie zu L^AT_EX in eine Präambel (allgemeine Festlegungen) und einen Programmkörper (Anwendung) zerlegt werden, wobei zur Unterscheidung zwei leicht unterschiedliche Hintergrundfarben benutzt werden. Die Marker können an jeder beliebigen Stelle des Dokuments genutzt werden. Obiges Beispiel wurde mit folgendem L^AT_EX-Code erzeugt:

```
\begin{externalDocument}[compiler=python3,prospath=/usr/local/bin/,
  force=false,showFilename,runs=1,code,ext=py,docType=py,usefancyvrb,
  grfOptions={width=\linewidth}]{python}
import os
#StartVisiblePreamble
from PIL import Image
import subprocess
# drawing area (xa < xb and ya < yb)
xa = -0.1716
xb = -0.1433
ya = 1.022
yb = 1.044
maxIt = 1024 # iterations
imgx = 1000 # image size
imgy = 750
image = Image.new("RGB", (imgx, imgy))
```

```

#StopVisiblePreamble

#StartVisibleMain
for y in range(imgy):
    cy = y * (yb - ya) / (imgy - 1) + ya
    for x in range(imgx):
        cx = x * (xb - xa) / (imgx - 1) + xa
        c = complex(cx, cy)
        z = 0
        for i in range(maxIt):
            if abs(z) > 2.0: break
            z = z * z + c
        r = i % 4 * 6
        g = i % 8 * 32
        b = i % 16 * 16
        image.putpixel((x, y), b * 65536 + g * 256 + r)
#StopVisibleMain
# now get the filename created by the latex document
imageName = os.path.basename(os.path.splitext(__file__)[0]) # get filename
image.save(imageName+".png", "PNG")
\end{externalDocument}

```

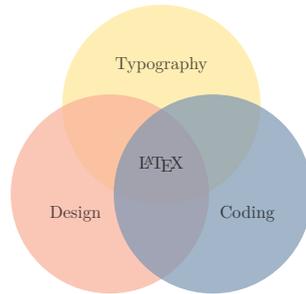
Durch Angabe einer Breite für die linksseitige Ausgabe des Quellcodes kann die Ausgabe von Code und Ergebnis nebeneinander angeordnet werden.

```

\usepackage{tikz}
\usepackage[hks,pantone,xcolor]{xespotcolor}

\SetPageColorSpace{HKS}
\definecolor{HYellow}{spotcolor}{HKS05N,0.5}
\definecolor{HRed}{spotcolor}{HKS14N,0.5}
\definecolor{HBlue}{spotcolor}{HKS38N,0.5}
\begin{tikzpicture}[fill opacity=0.7]
  \fill[HYellow]( 90:1.2) circle (2);
  \fill[HRed] (210:1.2) circle (2);
  \fill[HBlue] (330:1.2) circle (2);
  \node at ( 90:2) {Typography};
  \node at (210:2) {Design};
  \node at (330:2) {Coding};
  \node {\LaTeX};
\end{tikzpicture}

```



Setzen der Marker im Quellcode

Die Marker kennzeichnen die Bereiche des Quellcodes, die im (L^AT_EX-)Hauptdokument ausgegeben werden sollen. Für ein externes Dokument mit L^AT_EX-Code werden die Marker durch folgende Kommentare festgelegt:

```

[... ]
%StartVisiblePreamble
[... listed preamble code ]
%StopVisiblePreamble
[... ]
\begin{document}
[... ]
\end{document}

```

Alles zwischen %StartVisiblePreamble und %StopVisiblePreamble wird mit der Hintergrundfarbe BGpreamble (Vorgabe black!12) ausgegeben. Jeglicher Code zwischen \begin{document} und \end{document} dagegen als Textkörper mit der Hintergrundfarbe BGbody (Vorgabe black!8).

Die Marker sind jeweils durch das interne Makro \hv@extern@ExampleType festgelegt. Dieses Makro erwartet fünf Parameter, beispielsweise für Perl:

```

\hv@extern@ExampleType{py}
  {\NumChar StartVisibleMain}
  {\NumChar StopVisibleMain}
  {\NumChar StartVisiblePreamble}
  {\NumChar StopVisiblePreamble}

```

Das eigentliche Kommentarzeichen für Python ist #, welches aber in L^AT_EX eine Sonderstellung hat und deswegen hier maskiert als \NumChar zu benutzen ist.

Die aktuelle Version von hvextern unterstützt die folgenden Programmiersprachen, beziehungsweise T_EX-Programme: mpost, tex, latex, luatex, python3, perl, lua, java, pdftex, xetex, pdflatex, lualatex, xelatex und context. Für T_EX wird hierbei aber nur zwischen T_EX- und L^AT_EX-Code unterschieden, d. h. der Beispieltyp (Option docType) tex steht auch für pdftex, xetex und luatex und entsprechend latex auch für pdflatex, xelatex und lualatex.

Mit der Option docType wird also der Typ des Quellcodes und mit compiler das Programm festgelegt. Für docType sind folgende Werte möglich: context, lua, pl, tex, latex, mp oder py. Für compiler dagegen jedes auf dem eigenen Rechner installierte Programm. Beide Angaben können auch gleich sein, beispielsweise für die Anwendung von Lua: docType=lua und compiler=lua.

Die Definitionen für Lua, Perl, Java und Python sind alle gleich aufgebaut; sie unterscheiden sich nur in dem jeweils zu verwendenden Kommentarzeichen. Für Lua wird statt \NumChar nur das Kommentarzeichen -- benutzt:

```
\hv@extern@ExampleType{lua}
  {--StartVisibleMain}
  {--StopVisibleMain}
  {--StartVisiblePreamble}
  {--StopVisiblePreamble}
```

Für Lua-Dateien, die auch die Endung .lua haben, muss dreimal der Wert lua zugewiesen werden:

```
ext=lua, compiler=lua, docType=lua, ...
```

Gleiches gilt für das folgende Beispiel mit Java-Code:

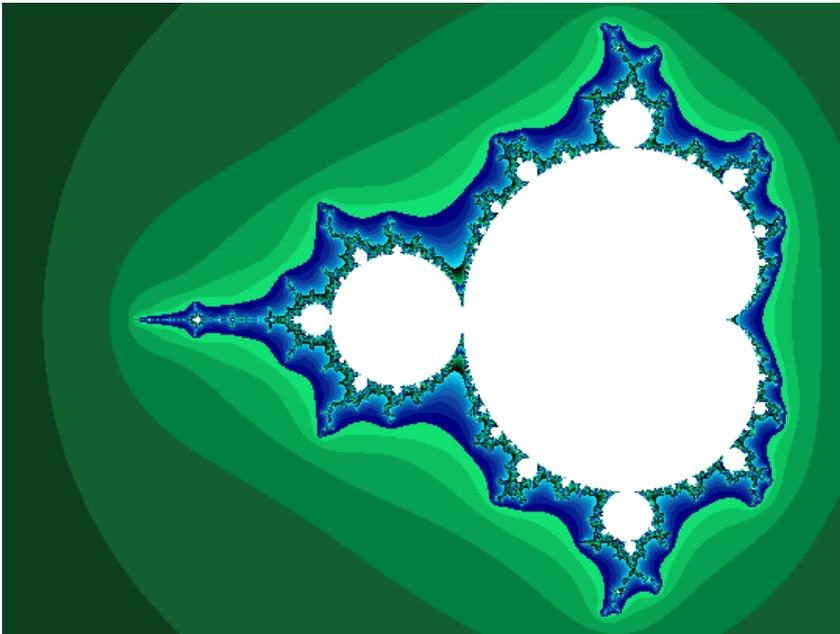
```
ext=java, compiler=java, docType=java, ...
```

```
public static int iterZahl(final double cx, final double cy, int maxIt,
                          final double radius) {
    int zaehler = 0; double zx = 0.0, zy = 0.0, tmp;
    do {
        tmp = zx*zx - zy*zy + cx;
        zy = 2*zx*zy + cy; zx = tmp;
        zaehler++;
    } while (zx*zx + zy*zy <= radius && zaehler < maxIt);
    return zaehler;
}
```

```

double xa = -2.5, xe = 0.7, ya = -1.2, ye = 1.2; // Ratio 20:15
double dx = (xe-xa)/(imageBreite-1), dy = (ye-ya)/(imageHoehe-1);
double cx, cy; int R, G, B; double radius = 10.0; int maxIt = 1024;
cx = xa;
for (int sp = 0; sp < imageBreite; sp++) {
  cy = ye; // von oben nach unten
  for (int ze = 0; ze < imageHoehe; ze++) {
    int zIter = iterZahl(cx, cy, maxIt, radius);
    if (zIter == maxIt) {
      g.setColor(Color.WHITE);
      g.drawLine(sp, ze, sp, ze);
    } else {
      R = zIter % 4 * 6 ; G = zIter % 8 * 32; B = zIter % 16 * 16;
      g.setColor(new Color(R,G,B));
      g.drawLine(sp, ze, sp, ze);
    }
    cy = cy - dy;
  } // for ze
  cx = cx + dx;
} // for sp

```



Das nächste Beispiel hat eine reine Textausgabe, sodass dies der Umgebung externalDocument durch redirect mitgeteilt wird; die Ausgabe wird dann in eine Datei gleichen Namens, jedoch mit der Endung .txt umgeleitet. Diese wird dann vom Hauptdokument Verbatim eingelesen, kann daher auch beliebige Zeichen enthalten.

```
print("1. " .. type("Hello world"))
print("2. " .. type(10.4*3))
print("3. " .. type(print))
print("4. " .. type(type))
print("5. " .. type(true))
print("6. " .. type(nil))
print("7. " .. type(type(X)))

print("8. " .. type(a))
a = 10
print("9. " .. type(a))
a = "a string!!"
print("10. " .. type(a))
a = print
print("11. " .. type(a))
```

```
1. string
2. number
3. function
4. function
5. boolean
6. nil
7. string
8. nil
9. number
10. string
11. function
```

Parameter

Programm und Anzahl der Durchläufe

Im Allgemeinen sollte jedes gewählte Programm auch gefunden werden. Jedoch kann es in seltenen Fällen notwendig sein, einen Pfad für das Programm vorzugeben, was über Zuweisung an progpath geschieht. Am Ende *muss* dabei ein / erscheinen, beispielsweise ./bin/. Als compiler kann jedes auf dem eigenen Rechner vorhandene Programm angegeben werden, wobei pdflatex die Vorgabe ist.

```
\define@key{hv}{progpath}[]{\def\hv@extern@progpath{#1}}
\define@key{hv}{compiler}[pdflatex]{\def\hv@extern@compiler{#1}}
\define@key{hv}{runs}[1]{\setcounter{hv@extern@runs}{#1}}
\define@key{hv}{runsequence}[]{\def\hv@extern@runsequence{#1}}
```

Anstelle der Verwendung von compiler, biber und xindex kann über den Parameter runsequence auch eine individuelle Ablaufsequenz vorgegeben werden. Dabei wird eine kommaseparierte Liste erwartet, beispielsweise für das folgende Beispiel mit zu erstellender Bibliografie und Index.

```
runsequence={lualatex,biber,xindex -l de -c AU,lualatex,lualatex},
cleanup={log,aux,toc,bbl,blg,run.xml,bcf,idx,ilg},
```

```
\usepackage[ngerman]{babel}
\usepackage{libertinus,hvindex}
\usepackage{makeidx}\makeindex
\usepackage{bibtex}\addbibresource{bibtex-examples.bib}
```

```
Sort with xindex \verb|-l DE --config AU|
\blindtext
\Index{Österreich} \Index{Öresund}
\Index{Ostern} \Index{Ober} \Index{Oberin}
\Index{Österreich} \Index{Öresund}
\Index{Ödem} \Index{Oligarch} \Index{Oder}
\Index{Goldmann}
\printindex
\nocite{*}\printbibliography
\blindtext
\blinddocument
```

<p>Sort with xindex -l DE --config AU: Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Gravrater der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich scherze. Dies ist ein Blindtext" oder "Jahrestes geburn"? Kjiff - mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie liest. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie Lorem Ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Österreich Österreich Ostern Ober Österreich Öresund Ödem Oligarch Oder Goldmann</p> <p style="text-align: center;">1</p>	<p style="text-align: center;">Literatur</p> <p>[1] José L. Alameddine u. a. "Elektromagnetisches Signalhorn". EU-297021951 (FR, GB, DE), 1998.</p> <p>[2] Arnold Angenendt. "Im Honorar Salvatoris - Vom Sinn und Unsinn der Patrozinienkunde". In: <i>Revue d'histoire ecclésiastique</i> 97 (2002), S. 431–456, 791–823.</p> <p>[3] Aristotle. <i>De Anima</i>. Hrsg. von Robert Drew Hicks. Cambridge: Cambridge University Press, 1907.</p> <p>[4] Aristotle. <i>Physics</i>. Übers. von P. H. Wicksteed und F. M. Cornford. New York: G. P. Putnam, 1929.</p> <p>[5] Aristotle. <i>Poetics</i>. Hrsg. von E. W. Lucas. Clarendon Aristotle. Oxford: Clarendon Press, 1968.</p> <p>[6] Aristotle. <i>The Rhetoric of Aristotle with a commentary by the late Edward Meredith Cope</i>. Hrsg. und komm. von Edward Meredith Cope. 3 Bde. Cambridge University Press, 1877.</p> <p>[7] Robert L. Augustinus. <i>Heterogeneous catalysis for the synthetic chemist</i>. New York: Marcel Dekker, 1995.</p> <p>[8] Averroes. <i>Drei Abhandlungen über die Conjunction des separaten Intellekts mit dem Menschen. Von Averroes (Ibn Arabi und Ibn Arabi), aus dem Arabischen überetzt von Samuel Ibn Tibbon</i>. Hrsg. und übers. von J. Herz. Berlin: S. Hermann, 1869.</p> <p style="text-align: center;">3</p>	<p style="text-align: center;"><i>Literatur</i></p> <p>[14] Willy W. Chu und We-Min Chow. <i>A Hybrid Hierarchical Model of a Multiple Virtual Storage (MVS) Operating System</i>. Forschungsbber. RC-6947 IBM, 1976.</p> <p>[17] Marcus Tullius Cicero. <i>De natura deorum. Über das Wesen der Götter</i>. Lateinisch und Deutsch. Hrsg. und übers. von Ursula Blank-Sangmeister. Mit einem Nachw. von Klaus Thasme. Stuttgart: Reclam, 1995.</p> <p>[18] Samuel Taylor Coleridge. <i>The collected works of Samuel Taylor Coleridge</i>. Bd. 7, 2: <i>Biographia Literaria, or Biographical Sketches of my literary life and opinions</i>. Hrsg. von Kathleen Coburn, James Engell und W. Jackson Raine. Bollingen Series 75. London: Routledge and Kegan Paul, 1983.</p> <p>[19] <i>Computers and Graphics</i> 35.4 (2013). <i>Semantic 3D Media and Content</i>: issn: 0097-8495.</p> <p>[20] Frank Albert Cotton u. a. <i>Advanced inorganic chemistry</i>. 6. Aufl. Chichester: Wiley, 1999.</p> <p>[21] CTAN. <i>The Comprehensive T_EX Archive Network</i> 2006. url: http://www.ctan.org (besucht am 01. 10. 2006).</p> <p>[22] Terrence Doody. "Hemingway's Style and Jake's Narrator". In: <i>The Journal of Narrative Technique</i> 4.3 (1974), S. 212–225. Excerpt in Roger Martin. Hrsg. <i>Contemporary Literary Criticism</i>. Bd. 61. Detroit: Gale, 1998, S. 204–208.</p> <p>[23] <i>The New Encyclopaedia Britannica</i>. Hrsg. von Warren E. Preece. 13. Aufl. 32 Bde. Chicago, Ill.: Encyclopaedia Britannica, 2003.</p> <p>[24] Dilip Parameshwar Gaonkar. Hrsg. <i>Alternative Modernities</i>. Durham und London: Duke University Press, 2001. issn: 0-822-37714-7.</p> <p style="text-align: center;">5</p>
--	---	---

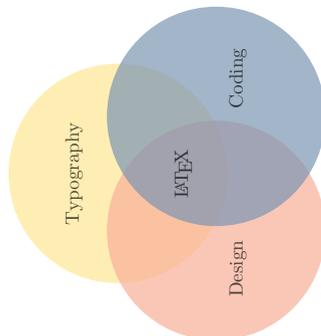
Grafikoptionen

```
\define@key{hv}{grfOptions}[][\def\hv@extern{grfOptions{#1}}
```

Die Angabe wird an das Makro `\includegraphics` weitergereicht, beispielsweise `angle=90,width=\linewidth`, wie im folgenden Beispiel zu sehen. Da Quellcode und Ausgabe hier nebeneinander angeordnet werden, bezieht sich die Angabe `\linewidth` auf die aktuelle Breite der Minipage. Somit hat die Ausgabe die maximal mögliche Größe.

```
\usepackage{tikz}
\usepackage[hks,pantone,xcolor]{xspotcolor}
```

```
\SetPageColorSpace{HKS}
\definecolor{HYellow}{spotcolor}{HKS05N,0.5}
\definecolor{HRed}{spotcolor}{HKS14N,0.5}
\definecolor{HBlue}{spotcolor}{HKS38N,0.5}
\begin{tikzpicture}[fill opacity=0.7]
  \fill[HYellow]( 90:1.2) circle (2);
  \fill[HRed] (210:1.2) circle (2);
  \fill[HBlue] (330:1.2) circle (2);
  \node at ( 90:2) {Typography};
  \node at (210:2) {Design};
  \node at (330:2) {Coding};
  \node {\LaTeX};
\end{tikzpicture}
```



Listingsoptionen

```
\define@key{hv}{lstOptions}[]{\def\hv@extern@lstOptions{#1}}
```

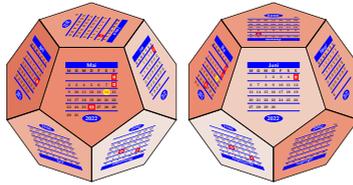
Der zugewiesene Wert wird entweder an `\lstinputlisting` oder, falls `usefancyvrb` gewählt wird, an `\VerbatimInput` aus dem Paket `fancyvrb` übergeben. Zu beachten ist, dass die jeweiligen Optionen unterschiedliche Bedeutungen und auch Namen haben, sodass nicht so einfach zwischen `listings` und `fancyvrb` gewechselt werden kann.

Das folgende Beispiel nutzt das Paket `listings`, was dem Standard entspricht und somit keine Parametersetzung benötigt. Als Optionen für den Code wurde aus reinen Demonstrationsgründen eine leicht »exotische« Festlegung getroffen:

```
lstOptions={basicstyle=\sffamily\itshape\scriptsize},
```

```
\documentclass[ngerman,landscape]{article}
\usepackage[geometry,babel]
\usepackage{pst-calendar}
```

```
\psscalebox{0.5}{%
  \psCalDodecaeder[Jahr=2022,style=Mai]%
}
\hspace{4cm}
\psscalebox{0.5}{%
  \psCalDodecaeder[Jahr=2022,style=Juni]%
}
```



Hintergrundfarbe des Codes

Für den Hintergrund und den Rahmen sind verschiedene Farben möglich. Sie können über die folgenden Parameter gesetzt werden, wobei in eckigen Klammern die Standardwerte angegeben sind.

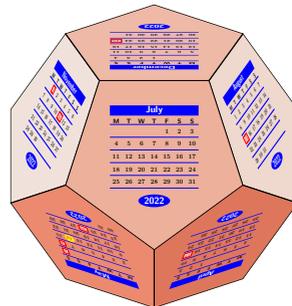
```
\define@key{hv}{BGpreamble}[black12]{\def\hv@extern@BGpreamble{#1}}
\define@key{hv}{BGbody}[black8]{\def\hv@extern@BGbody{#1}}
\define@key{hv}{B0preamble}[black12]{\def\hv@extern@B0preamble{#1}}
\define@key{hv}{B0body}[black8]{\def\hv@extern@B0body{#1}}
```

Die Parameterwerte werden an das Paket `tcolorbox` weitergereicht und dort ausgewertet. [3] Dadurch, dass jeweils Hintergrund und Rahmen dieselbe Farbe haben, bleibt der Rahmen »unsichtbar«. Dies ändert sich bei unterschiedlichen Werten, beispielsweise:

```
BGpreamble=red!10, B0preamble=red, BGbody=blue!8, B0body=blue,
```

Grundsätzlich sollte man auf dezente Farben setzen, um die Ausgabe nicht gegenüber dem Code in den Hintergrund treten zu lassen.

```
\usepackage{pst-calendar}
\psscalebox{0.3}{%
  \psCalDodecaeder[Year=2022,style=july]%
}
```



Typ des Quellcodes

Die aktuelle Version von `hvextern` unterstützt Quellcode in METAPOST, plain \TeX , \LaTeX , Con \TeX t, Python, Lua und Perl. Jeder Typ hat seine eigene Konfiguration, die

zum einen die bereits erwähnten Marker und zum anderen einen Programmablauf vorgibt, falls eine spezielle Behandlung notwendig ist. So erfordert ein Quellcode in L^AT_EX eine Sonderbehandlung, falls das verwendete Programm latex ist. Die entsprechende Konfiguration enthält folgende Definitionen:

```
\hv@extern@ExampleType{latex}%                for _all_LaTeX engines
  {\string\begin\string{document\string}}%
  {\string\end\string{document\string}}%
  {\perCent StartVisiblePreamble}%
  {\perCent StopVisiblePreamble}%

% only for the sequence latex->dvips->ps2pdf
\def\hv@extern@runLATEX#1#2#3#4{% path/compiler/file/extension
  \ifhv@extern@verbose \typeout{>>> running #1#2 #3#4}\fi
  \ShellEscape{#1#2\space #3#4}%
  \ifhv@extern@verbose \typeout{>>> running #1dvips #3}\fi
  \ShellEscape{#1dvips\space #3.dvi}%
  \ifhv@extern@verbose \typeout{>>> running ps2pdf #3.ps}\fi
  \ShellEscape{#1ps2pdf\space -dAutoRotatePages=/None\space
    -dALLOWPSTRANSAPRENCY\space#3.ps}%
}
```

Der zu verwendende Name des Makros muss einer Konvention folgen:

```
\def\hv@extern@run<NAME>#1#2#3#4{% path/compiler/file/extension
...}
```

Die Konfiguration für T_EX-Code sieht dann ähnlich aus. Der Typ des Quellcodes und das verwendete Programm können für T_EX/L^AT_EX und ConT_EXt unterschiedlich sein, beispielsweise Typ latex, aber Programm lualatex.

Ausgabe einer oder mehrerer Seiten

Für den Fall, dass mehrere Seiten oder auch nur eine bestimmte Seite eines Dokuments ausgegeben werden sollen, so kann dies über den Parameter pages gesteuert werden. Er erwartet eine kommaseparierte Liste der auszugehenden Seiten. Mit dem Parameter frame können die einzelnen Seiten gerahmt werden, um so eine bessere Abgrenzung zu erreichen.

```
\define@key{hv}{pages}[1]{\def\hv@extern@pages{#1}}
\define@key{hv}{pagesep}[1em]{\hv@extern@pagesep=#1}
\define@boolkey{hv}[hv@extern@]{frame}[true]{}
```

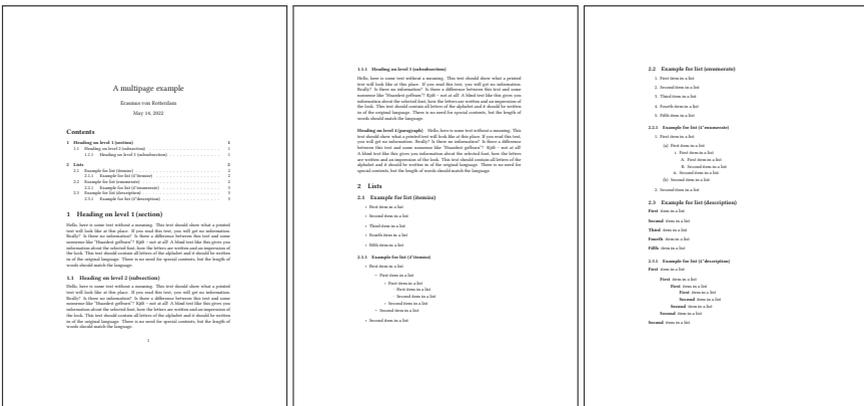
Es obliegt dem Anwender, über den Parameter grfOptions sicherzustellen, dass die Seiten für die Ausgabe in der Regel zu skalieren sind. Für das folgende Beispiel,

welches die ersten drei Seiten eines Dokuments ausgibt, wurde folgende Festlegung getroffen:

```
pages={1,2,3}, pagesep=2pt,
grfOptions={width=0.32\linewidth},
frame, framesep=0pt,
compiler=lualatex, runs=2, % for the TOC
```

```
\usepackage[american]{babel}
\usepackage{libertinus}
\usepackage{blindtext}
```

```
\title{A multipage example}
\author{Erasmus von Rotterdam}
\maketitle
\tableofcontents
\blinndocument
```



Weißraum abschneiden

Bei der Darstellung der Ausgaben von Beispielen interessiert im Prinzip nur der Bereich, der eine Grafik oder Text enthält. An dem Weißraum ist man meistens nicht interessiert und möchte ihn entfernen. Für Dokumente, die nur aus einer Seite bestehen, kann im Allgemeinen dazu die Dokumentenklasse `standalone` benutzt werden, die automatisch jeglichen Weißraum entfernt. Hat man mehr als eine Seite oder möchte eine spezielle Dokumentenklasse benutzen, so ist die Anwendung des Parameters `crop` sinnvoller.

```
\define@boolkey{hv}[hv@extern@]{crop}[true]{}
\define@key{hv}{cropmargin}[2]{\def hv@extern@cropmargin{#1}}% length in pt
```

crop kann auch auf Dokumente mit mehreren Seiten angewendet werden. In diesem Fall sollte man aber darauf achten, dass die Seiten mit Kopf- und Fußzeilen versehen sind, damit der Weißraum, der abgeschnitten werden soll, immer gleich groß ist. Andernfalls haben die Seiten unterschiedliche Höhen, was im folgenden Beispiel an der ersten Seite gezeigt wird. Es wurden unter anderem folgende Parameter gesetzt:

```
pages={1,2,3},
pagesep=2pt,
grfOptions={width=0.3\linewidth},
frame,
crop, cropmargin=5,% entspricht 5pt Rand
compiler=lualatex, runs=2, % for the TOC
```

```
\usepackage[american]{babel}
\usepackage{libertinus}
\usepackage{blindtext}
\pagestyle{headings}
```

```
\title{A multipage example}
\author{Erasmus von Rotterdam}
\maketitle
\tableofcontents
\Blinddocument
```

A multipage example	
Erasmus von Rotterdam	
May 14, 2022	
Contents	
1 Heading on level 1 (section)	1
1.1 Heading on level 2 (subsection)	2
1.1.1 Heading on level 3 (subsubsection)	3
2 Lists	5
2.1 Example for list (description)	5
2.1.1 Example for list (enumitem)	6
2.2 Example for list (enumarater)	8
2.2.1 Example for list (enumarater)	9
2.3 Example for list (description)	10
2.3.1 Example for list (description)	11
1 Heading on level 1 (section)	
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
2	

1 HEADING ON LEVEL 1 (SECTION)	
at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
After this fourth paragraph, we start a new paragraph section. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
1.1 Heading on level 2 (subsection)	
This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
After this fourth paragraph, we start a new paragraph section. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
2	

1 HEADING ON LEVEL 1 (SECTION)	
at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
1.1.1 Heading on level 3 (subsubsection)	
And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
After this fourth paragraph, we start a new paragraph section. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Hundred geberns"? Kjll - not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special contents, but the length of words should match the language.	
3	

Ausgabe als Gleitobjekt

In der Regel erfolgt die Ausgabe im laufenden Text, was durchaus ungünstig sein kann, wenn der Textanteil relativ gering ist. Dann können größere Freiräume auf einer Seite entstehen, was immer ungünstig ist. In solchen Fällen sollte man die Option `float` benutzen, wobei dann wie üblich auch eine Beschriftung über den Parameter `caption` und eine eventuelle Textmarke über `label` festgelegt werden können. Der Gleittyp ist grundsätzlich `figure`.

```
% PSTricks with lualatex (no Ghostscript)
\usepackage{pst-coxeterp}
```

```
\begin{pspicture}(-1,-1)(1,1)\Simplex[dimension=2]\end{pspicture}
\begin{pspicture}(-1,-1)(1,1)\Simplex[dimension=3]\end{pspicture}
\begin{pspicture}(-1,-1)(1,1)\Simplex[dimension=5]\end{pspicture}
\begin{pspicture}(-1,-1)(1,1)\Simplex[dimension=7]\end{pspicture}
```

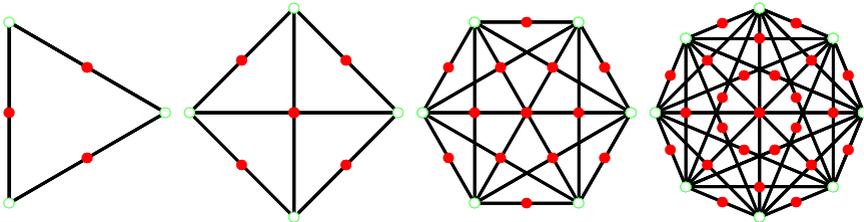


Abb. 1: An example for Coxeter images.

```
\define@boolkey{hv}[hv@extern@]{float}[true]{}
\define@key{hv}{caption}[]{\def\hv@extern@caption{#1}}
\define@key{hv}{label}[]{\def\hv@extern@label{#1}}
```

Die Abbildung 1 zeigt ein Beispiel als gleitendes Objekt, welches unter anderem durch folgende Parameter festgelegt wurde:

```
float,
caption={An example for Coxeter images.},
label=img:cox,
```

Zu beachten ist, dass sich die Angabe `float` nur auf die Ausgabe bezieht, anderenfalls könnte ein vorhergehendes Listing keinen Seitenumbruch haben und der Textsatz wäre erheblich schwieriger. Andererseits kann es durchaus möglich sein, dass anderer Text zwischen Code und Ausgabe eines Beispiels erscheint. Dann ist ein manueller Eingriff notwendig, indem beispielsweise der Befehl `\captionof` aus dem Paket `caption` benutzt wird, der eine Beschriftung ohne Gleitumgebung erlaubt.

Quellcode und Ausgabe nebeneinander

Normalerweise erfolgt zuerst die Ausgabe des Quellcodes und danach die Ausgabe. Diese Reihenfolge kann mit der aktuellen Version von `hvfloot` nicht geändert werden. Der Parameter `mpwidth` bestimmt die Breite einer linken Minipage und wird immer dann ausgewertet, wenn er größer als 0 pt ist. Dann wird für den Rest der Zeile bis auf den Zwischenabstand eine zweite Minipage für die Ausgabe reserviert. Deren Breite beträgt `\linewidth-\hv@extern@mpwidth-1em`. Beide Minipages werden an ihrer Oberkante ausgerichtet.

```
\define@key{hv}{mpwidth}[0pt]{\setlength\hv@extern@mpwidth{#1}}
```

Der Abstand zwischen den beiden Minipages ist 1 em, wie bereits oben angegeben.

Horizontale Ausrichtung der Ausgabe

Der Code wird immer linksbündig ausgerichtet, wogegen die Ausgabe über die Option `align` die bekannten Ausrichtungen annehmen kann. Die Anwendung des Paketes `ragged2e` bringt hier keine Vorteile.

```
%\define@key{hv}{align}[\centering]{\def\hv@extern@align{#1}}
```

Vorgabe mit `align=\centering`

```
\rule{0.5\linewidth}{5mm}
```



Linksbündig mit `align=\raggedright`

```
\rule{0.5\linewidth}{5mm}
```



Rechtsbündig mit `align=\raggedleft`

```
\rule{0.5\linewidth}{5mm}
```



Nebeneinander – Standard mit align=\centering

```
\rule{0.25\linewidth}{5mm}
```


Nebeneinander – linksbündig mit align=\raggedright

```
\rule{0.25\linewidth}{5mm}
```


Nebeneinander – rechtsbündig mit align=\raggedleft

```
\rule{0.25\linewidth}{5mm}
```


Reinen Text als Ausgabe einfügen

Für L^AT_EX-Dokumente ist die Ausgabe als PDF im Allgemeinen der Standard. Nicht jedoch bei Anwendung einer Programmiersprache wie beispielsweise Perl. Dort kann es ebenso eine reine Textausgabe sein, die zwingend in eine Datei umgeleitet (redirect) oder geschrieben werden muss, damit diese Verbatim ins Hauptdokument eingefügt werden kann. Mit den Parametern `includegraphic` und `redirect` kann dies gesteuert werden. Mit `includegraphic=false` obliegt es dem Anwender, dafür zu sorgen, dass innerhalb des externen Dokuments jede Ausgabe in eine Textdatei geschrieben wird. Mit `redirect` erfolgt dies automatisch, wobei dann aber *jede* Programmausgabe in der externen Textdatei landet.

```
\define@boolkey{hv}[hv@extern@]{includegraphic}[true]{}
\define@boolkey{hv}[hv@extern@]{redirect}[true]{}

```

Die Textdatei muss den selben Hauptdateinamen wie die externe Datei haben, jetzt jedoch mit der Endung `.txt`. Wie bereits erwähnt, kann jedes Skript/Programm selbst feststellen, mit welchem Namen es aufgerufen wurde, sodass es problemlos möglich ist, den korrekten Namen für die Datei der Textausgabe zu bestimmen. Für ein Perlprogramm könnte man dies beispielsweise durch folgenden Code erreichen:

```
my $filename = $0;           # the current filename
$filename =~ s/\.pl//;      # without extension .pl
$filename = "${$filename}.txt"; # for the output
open(my $fh, '>', $filename);

```

Im nächsten Beispiel wird jedoch die Umleitung mit `redirect` bevorzugt. Nur der Vollständigkeit halber: Eine Kaprekarkonstante ist eine Zahl A mit $\max(A) - \min(A) = A$. \max und \min sind die sortierten Ziffern der Zahl A : $495 = 954 - 459$.

```
my $zahl = 1;
my $anfang = 1;
my $ende = 9;
my $found = 0;
```

```
print "Bestimmung von\n Kaprekarkonstanten
↳...\n";
while ($zahl < 8) {
  print "${zahl}-stellig: ";
  foreach ($anfang...$ende) { # for every row
    ↳$_
    @Zeichen = split(//,$_);
    $Min = join("",sort(@Zeichen));
    $Max = reverse($Min);
    $Dif=$Max-$Min;
    if($_ eq $Dif) {
      $found = 1;
      print $_, " ";
    }
  }
  if (!$found) { print "---\n"; }
  else { print "\n"; }
  $found = 0;
  $zahl++;
  $anfang = $anfang*10;
  $ende = $ende*10;
}
```

```
Bestimmung von
Kaprekarkonstanten ...
1-stellig: ---
2-stellig: ---
3-stellig: 495,
4-stellig: 6174,
5-stellig: ---
6-stellig: 549945, 631764,
7-stellig: ---
```

Ein anderes Beispiel mit Lua, welches ebenfalls eine Textausgabe erzeugt, diese aber nicht durch Umleitung, sondern durch Definition einer Datei, in die die Ausgabe erfolgt. Der interne Dateiname kann wie folgt ermittelt werden:

```
local filename = arg[0]
local shortFN = str:match("(.)%.+.+") -- delete extension
outFile = io.open(shortFN..".txt","w+") -- open external file
```

```
function nextrow(t)
  local ret = {}
  t[0], t[#t+1] = 0, 0
  for i = 1, #t do ret[i] = t[i-1] + t[i] end
  return ret
end
```

```
function triangle(n)
  t = {1}
  for i = 1, n do
    m = (n - i)
    for j = 1,m do outFile:write("  ") end
    for k = 1,i do outFile:write(string.format("%8s",t[k])) end
    outFile:write("\n")
    t = nextrow(t)
  end
end
```

```
triangle(10)
```

```

                1
              1 1
            1 2 1
          1 3 3 1
        1 4 6 4 1
      1 5 10 10 5 1
    1 6 15 20 15 6 1
  1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
  1 9 36 84 126 126 84 36 9 1
c→1
```

Ausgabe in der Zeile (inline)

Die Ausgabe kann innerhalb einer Zeile im sogenannten Inlinemodus ausgegeben werden. Dies macht immer dann Sinn, wenn man bestimmte Zeichen nicht zur Verfügung hat und diese extern erzeugen lässt. Die Angabe des entsprechenden Quellcodes entfällt sinnvollerweise. Mit `inline` wird daher automatisch `code=false` gesetzt.

```
\define@boolkey{hv}[hv@extern@]{inline}[true]{\hv@extern@codefalse}
```

Ein entsprechendes Beispiel wurde bereits auf Seite 31 gezeigt.

Die Erzeugung von Bibliografie, Index, ...

Die aktuelle Version von `hextern` unterstützt eine Bibliografie mit Biber und einen Index mit `xindex` durch die folgenden Parameter:

```
\define@boolkey{hv}[hv@extern@]{biber}[true]{}
\define@boolkey{hv}[hv@extern@]{xindex}[true]{}
\define@key{hv}{xindexOptions}[]{\def\hv@extern@xindexOptions{#1}}
```

Prinzipiell benötigt der externe Lauf von biber keine weiteren Parameter, wohingegen das Programm xindex beispielsweise Angaben zur Sprache, zur Stildatei usw. benötigt. Das nächste Beispiel wird mit den folgenden Parametern erzeugt:

```
\begin{externalDocument}[
  compiler=lualatex, runs=2,
  pages=2, % nur eine Seite ausgeben
  crop,
  xindex,
  xindexOptions={-l DE --config DIN2},% Deutsch, Österreich
  mpwidth=0.6\linewidth, % nebeneinander
  usefancyvrb=false,
  docType=latex,
  ...
]{voss}
```

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

```
Sort with xindex \verb|-l DE --config AU|
\Index{Österreich} \Index{Öresund}
\Index{Ostern} \Index{Ober}
\Index{Oberin} \Index{Österreich}
\Index{Öresund} \Index{Ödem}
\Index{Oligarch} \Index{Oder}
\Index{Ostern} \Index{Ober}
\Index{Oberin} \Index{Obstler}
\Index{Öl} \Index{ölen}
\Index{Oder|seealso{Fluss}}
\Index{Göbel} \Index{oder}
\index{Fluss!Oder} \Index{Goethe}
\Index{Göthe} \Index{Götz}
\Index{Goldmann}

\printindex
```

Index

F	Öl, 1
Fluss	ölen, 1
- Oder, 1	Öresund, 1
	Österreich, 1
G	Ober, 1
Göbel, 1	Oberin, 1
Göthe, 1	Obstler, 1
Goethe, 1	Oder, 1
Götz, 1	oder, 1
Goldmann, 1	Oder, <i>siehe auch</i> Fluss
O	Oligarch, 1
Ödem, 1	Ostern, 1

Verbatimmodus

Paket listings

Standardmäßig wird der Befehl `\lstinputlisting` aus dem Paket `listings` für die Ausgabe des Quellcodes benutzt.

```
\DeclareNewSectionCommand[
  style=section,
  level=4,
  beforekip=-3.25ex plus -1ex minus -.2ex,
  afterskip=1.5ex plus .2ex,
  font=\normalsize,
  indent=0pt,
  counterwithin=subsubsection
]{subsubsubsection}

\RedeclareSectionCommand[
  level=5,
  toplevel=5,
  tocindent=13em,
  tocnumwidth=5.9em,
  counterwithin=subsubsubsection
]{paragraph}

\RedeclareSectionCommand[
  level=6,
  toplevel=6,
  tocindent=15em,
  tocnumwidth=6.8em
]{subparagraph}

\setcounter{secnumdepth}{\subsubsubsectionnumdepth}
\setcounter{tocdepth}{\subsubsubsectionontocdepth}
```

```
\tableofcontents
\chapter{Einführung}
\section{Ein Abschnitt}
\subsection{Ein Unterabschnitt}
\subsubsection{Ein Unter-Unterabschnitt}
\subsubsubsection{Ein Unter-Unter-Unterabschnitt}
\paragraph{Der normale Paragraph}
\blindtext
\subparagraph{Der normale Unterparagraph}
```

voss-27.tex


```

]{subparagraph}
\setcounter{secnumdepth}{\subsubsubsectionnumdepth}
\setcounter{tocdepth}{\subsubsubsectiontocdepth}

```

```

\tableofcontents
\chapter{Einführung}
\section{Ein Abschnitt}
\subsection{Ein Unterabschnitt}
\subsubsection{Ein Unter-Unterabschnitt}
\paraghraph{Der normale Paragraph}
\blindtext
\subparagraph{Der normale Unterparagraph}
\blindtext
\blinddocument

```

Inhaltsverzeichnis

1	Einführung	3
1.1	Ein Abschnitt	3
1.1.1	Ein Unterabschnitt	3
1.1.1.1	Ein Unter-Unterabschnitt	3
2	Überschrift auf Ebene 0 (chapter)	5
2.1	Überschrift auf Ebene 1 (section)	5
2.1.1	Überschrift auf Ebene 2 (subsection)	5
2.1.1.1	Überschrift auf Ebene 3 (subsubsection)	6
2.2	Listen	6
2.2.1	Beispiel einer Liste (itemize)	6
2.2.1.1	Beispiel einer Liste (itemize)	7
2.2.2	Beispiel einer Liste (enumerate)	7
2.2.2.1	Beispiel einer Liste (enumerate)	7
2.2.3	Beispiel einer Liste (description)	8
2.2.3.1	Beispiel einer Liste (description)	8

1

Kapitel 1

Einführung

1.1 Ein Abschnitt

1.1.1 Ein Unterabschnitt

1.1.1.1 Ein Unter-Unterabschnitt

1.1.1.1.1 Ein Unter-Unter-Unterabschnitt

Der normale Paragraph Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Hundert gefahren“? Kjllt – mütschten! Ein Blindtext bietet nur wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und präde, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Der normale Unterparagraph Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Hundert gefahren“? Kjllt – mütschten! Ein Blindtext bietet nur wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und präde, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3

Unterstützte METAPOST/T_EX-Engines

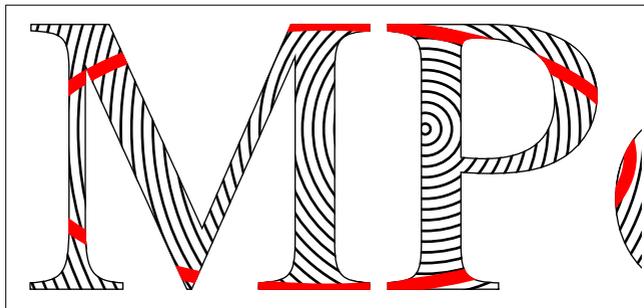
METAPOST-Beispiel

```

defaultfont:= "ptmr8r";
warningcheck:=0;

```

```
draw fullcircle shifted (0.5,0.6) xscaled 8cm yscaled 3.5cm
  withpen pencircle scaled 5bp withcolor red;
special( " /Times-Roman findfont 150 scalefont setfont " &
  " 0 10 moveto (MPost) false charpath clip stroke gsave 150 70 translate " &
  " 2 4 600 {dup 0 moveto 0 exch 0 exch 0 360 arc stroke} for grestore ");
```



plain \TeX -Beispiel

```
\footline={\footsc the electronic journal of combinatorics
  {\footbf 16} (2009), \#R00\hfil\footrm\folio}
```

voss-30.tex

```
\font\bigrm=cmr12 at 14pt
\centerline{\bigrm An elementary proof of the reconstruction conjecture}

\bigskip\bigskip
\centerline{D. Remifa\footnote*{Thanks to the editors of this journal!!}}
\smallskip
\centerline{Department of Inconsequential Studies}
\centerline{Solatido College, North Kentucky, USA}
\centerline{\tt remifa@dis.solatido.edu}
\bigskip
\centerline{\footrm
Submitted: Jan 1, 2009; Accepted: Jan 2, 2009; Published: Jan 3, 2009}
\centerline{\footrm Mathematics Subject Classifications: 05C88, 05C89}
\bigskip\bigskip
\centerline{\bf Abstract}
\smallskip
{\narrower\noindent
The reconstruction conjecture states that the multiset of unlabeled
vertex-deleted subgraphs of a graph determines the graph, provided it
has at least 3 vertices. A version of the problem was first stated
```

by Stanisław Ulam. In this paper, we show that the conjecture can be proved by elementary methods. It is only necessary to integrate the Lenkle potential of the Broglington manifold over the quantum supervacillatory measure in order to reduce the set of possible counterexamples to a small number (less than a trillion). A simple computer program that implements Pipletti's classification theorem for torsion-free Aramaic groups with symplectic socles can then finish the remaining cases.}

\bigskip

\beginsection 1. Introduction.

This is the start of the introduction.



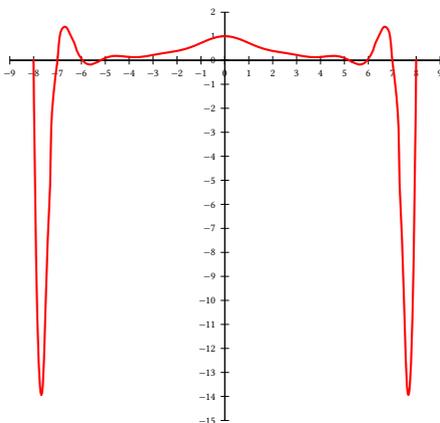
LaTeX-Beispiel (LuaLaTeX)

```
\usepackage{fontenc}\usepackage{libertinus}
\usepackage{pst-all}
```

```

\psset{unit=0.8cm}
\begin{pspicture}(-9,-15)(9,2)
\psaxes(0,0)(-9,-15)(9,2)
\psplot[algebraic,plotstyle=curve,curvature=1 1 0,
linewidth=2pt,linecolor=red]{-8}{8}{
1 - 3876218985722260225*x^2/10892114744073986176
+ 14975974793271450625*x^4/174273835905183778816
- 317095420958296875*x^6/26811359370028273664
+ 194412970920703125*x^8/214490874960226189312
- 2090988251953125*x^10/53622718740056547328
+ 99480224609375*x^12/107245437480113094656
- 7879638671875*x^14/697095343620735115264
+ 152587890625*x^16/2788381374482940461056}
\end{pspicture}

```



ConTeXt-Beispiel (mkIV)

```

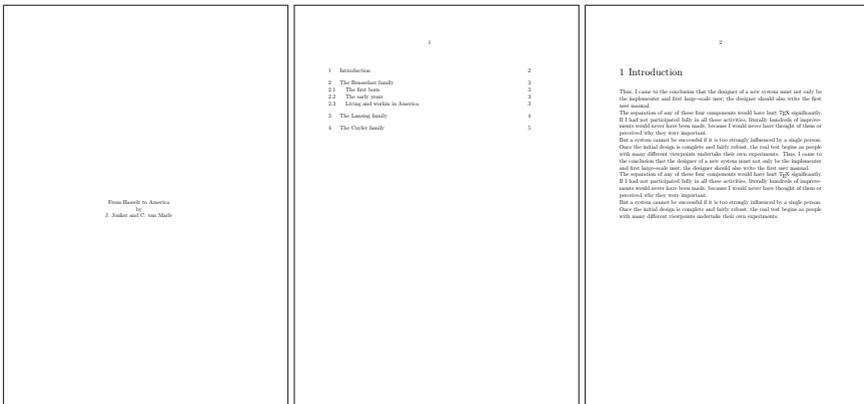
\definehead
[myhead]
[section]
\setuphead
[myhead]
[numberstyle=bold,
textstyle=bold,
before=\hairline\blank,
after=\nowhitespace\hairline]

```

```

\startstandardmakeup
\midaligned{From Hasselt to America}
\midaligned{by}
\midaligned{J. Jonker and C. van Marle}
\stopstandardmakeup
\placecombinedlist[content]
\chapter{Introduction}
\input knuth \input knuth
\chapter[rensselaer]{The Rensselaer family}
\section{The first born}
\input knuth
\section{The early years}
... in those days Hasselt was ...
\section{Living and workin in America}
\input knuth
\chapter[lansing]{The Lansing family}
... the Lansing family was also ...
\chapter[cuyler]{The Cuyler family}
... much later Tydeman Cuyler ...
\myhead[headlines]{And the end}
foo

```



Weitere Parameter

showFilename Der Dateiname des Beispiels wird um 90° gedreht im Rand angegeben. Dies ist insbesondere bei vielen externen Beispielen hilfreich; die Zuordnung zu den externen Dokumenten/Beispielen wird einfacher.

force Kann die Übersetzungsgeschwindigkeit erhöhen, indem das erneute Übersetzen eines Beispiels verhindert wird. Mit `force=false` wird die vorhandene Ausgabedatei benutzt. Diese Option sollte nur in Ausnahmefällen benutzt werden, denn eine Änderung im Hauptdokument beim Quellcode des Beispiels führt dann *nicht* zu einer aktualisierten Beispielausgabe.

cleanup Die TeX-typischen Hilfsdateien eines L^AT_EX-Laufs können zur Förderung der Übersicht gelöscht werden. Standardmäßig sind dies die `.aux`- und `.log`-Datei: `cleanup={aux,log}`.

ExamplesDir Verzeichnis, in das alle Beispieldateien kopiert werden sollen. Das Verzeichnis selbst muss vom Anwender erstellt werden.

copyToExampleDir Alle Beispiele werden in das angegebene Verzeichnis kopiert, was bei vielen Beispielen die Übersicht fördert.

tclobox=false Keine Box-Befehle aus dem Paket `tcolorbox` benutzen.

framesep Wert für `\fboxsep`, falls `frame` gesetzt ist.

mpsep Abstand zwischen Code und Ausgabe (Vorgabe 1 em).

pagesep Abstand zwischen Seiten (Vorgabe 1 em).

eps Die erzeugte PDF-Datei zusätzlich nach EPS konvertieren (historische Gründe).

Literatur

- [1] Jobst Hoffmann, Brooks Moses, Carsten Heinz: The listings package, Typeset source code listings using L^AT_EX, Version 1.8d, 2020, CTAN : <https://ctan.org/pkg/listings> (besucht am 25. 4. 2022).
- [2] Rolf Niepraschk: The showexpl package, Typesetting L^AT_EX source code, Version 0.3s, CTAN : <https://ctan.org/pkg/showexpl> (besucht am 25. 4. 2022).
- [3] Thomas F. Sturm: The tcolorbox package, Coloured boxes, for L^AT_EX examples and theorems, etc, Version 5.0.2, 2022, CTAN : <https://ctan.org/pkg/tcolorbox> (besucht am 25. 4. 2022).
- [4] Timothy Van Zandt u. a.: The fancyvrb package, Sophisticated verbatim text, Version 4.2, 2022, CTAN : <https://ctan.org/pkg/fancyvrb> (besucht am 25. 4. 2022).
- [5] Herbert Voß: The hvextern package, Writing and reading of external source code and insert the output, Version 0.23, 2022, CTAN : <https://ctan.org/pkg/hvextern> (besucht am 8. 5. 2022).

ConT_EXt kurz notiert

Henning Hraban Ramm

In der ConT_EXt-Welt ist Vieles in Bewegung, das eine Erwähnung wert ist, aber keinen ganzen Artikel begründet. (Stand: 12. 5. 2022)

Nach einem sehr aktiven Winter hat sich die Entwicklung wieder etwas verlangsamt.

Internationaler ConT_EXt-»Stammtisch«

Leider habe ich den letzten Stand meines Artikels für die letzte DTK zu spät eingereicht. Seit März haben wir das regelmäßige Online-Treffen von Dienstag auf Mittwoch verlegt: Jeweils am zweiten Mittwoch des Monats, ca. 15–17 Uhr ME(S)Z, gibt es unter <https://lecture.senfcall.de/hen-rbr-rku-oke> ein englischsprachiges Online-Treffen für ConT_EXt-Anwender.



ConT_EXt meeting 2022

Das diesjährige Treffen der ConT_EXt-Entwickler und -Anwender findet vom 12. bis 18. September in Dreifelden/Westerwald statt. Bei Erscheinen dieser Ausgabe ist die Anmeldung eröffnet. Weitere Informationen unter <https://meeting.contextgarden.net/2022>.



Bibliografie

Zur Zeit wird diskutiert, wie sich die Citeproc-Zitierstile (CSL) in ConT_EXt verwenden lassen. Es gibt bereits eine Lua-Bibliothek für L^AT_EX, von der aber nur Teile verwendbar wären.

Serbisch

Serbisch kann in kyrillischer oder lateinischer Schrift geschrieben werden. Ivan Pešić hat die Transliteration verbessert und die Trennmuster zusammengefügt, so dass man die Sprache nicht mehr wechseln muss, wenn man beide Schriftsysteme verwendet. (Für Bulgarisch war das wohl schon so.)

Mathematiksatz

Das Thema ist noch lange nicht abgeschlossen – beispielsweise sollen neue Mathe-Leerzeichenklassen (`mu glue classes`) und Penalties manuellen Weißbraumausgleich unnötig machen. Hans Hagen arbeitet mit dem schwedischen Mathematiker Mikael Sundqvist daran.

Flexiblerer Umbruch

Für den Umbruch von Formeln hat Hans dehnbare Boxen eingeführt, die auch in anderem Zusammenhang einsetzbar sind. Die bisher veröffentlichten Beispiele zeigen, wie sich der Zeilenumbruch verbessern lässt, wenn statt `hbox` die neue `adaptivehbox` eingesetzt wird, die eine Art `emergencystretch` enthält. Dehnbare vertikale Boxen eignen sich auch für Grafiken, deren Größe \TeX für einen besseren Seitenumbruch anpassen darf.

Eine weitere experimentelle Möglichkeit zur Verbesserung des Seitenumbruchs ist der Parameter `adaptive` von `setuplayout`; damit kann man \TeX die Verlängerung des Satzspiegels erlauben, um alleinstehende Zeilen¹ zu vermeiden, zusätzlich zur manuellen Anpassung mit `adaptlayout`.

Auf der Mailingliste wurde das Modul `lua-widow-control` von Max Chernoff diskutiert; es ist auch als Paket für \LaTeX und \PlainTeX verfügbar, natürlich nur mit $\text{Lua}\TeX$. Es sucht auf »schlechten« Seiten nach den Absätzen, wo eine Dehnung des Weißraums am wenigsten auffällt und ist besonders zur Verbesserung des Seitenumbruchs im registerhaltigen Satz gedacht. Bei meinen Tests hat es mich jedoch nicht überzeugt: Für einen Automatismus ist die Qualität passabel, aber mit Zeichendehnung (`hz`) und manuellen Eingriffen erreiche ich einen besseren Umbruch; mir sind im Zweifelsfall alleinstehende Zeilen lieber als löchrige Absätze und unzuverlässige Abstände.

MetaPost

Die MetaPost-Instanz in ConT_EXt LMTX verwendet jetzt größere Integer-Variablen (»DoubleFun«) und ist damit auf den meisten Prozessorarchitekturen sogar schneller.

Eine erhebliche Beschleunigung hat Hans Hagen allerdings durch Code-Optimierungen und ein paar neue Befehle erreicht. Die komplexen MetaPost-Grafiken des Künstlers Ecir Baff² werden bis zu hundertmal schneller berechnet. Das betrifft

¹ Ich verzichte auf die diskriminierenden Fachausdrücke »Schusterjungen« und »Hurenkinder«.

² <https://art-aleatoire.com>

besonders Konstrukte, bei denen über `arctime` iteriert wird. Ein Artikel dazu ist in der aktuellen Ausgabe der MAPS erschienen.

Tree-Sitter-Parser

Paul Mazaitis hat einen Tree-Sitter-Parser für ConT_EXt vorgestellt: Code auf GitHub³; Wiki-Seite⁴. So etwas kann als Grundlage für Syntax-Highlighting oder Rechtschreibkorrektur dienen.

³github.com/pmazaitis/tree-sitter-context_en

⁴https://wiki.contextgarden.net/pm39_tree_sitter_parser

Von fremden Bühnen

Druck oder Nichtdruck¹

Peter Flynn

Peter Flynn widmet sich im ersten Teil eines Artikels seiner Kolumne »Typographers' Inn« [3] der Frage, wie sich der Druck auf Papier von Veröffentlichungen als Online- und elektronischen Medien wie Web-Seiten oder e-Book-Reader unterscheidet und wie sich dies auf die Typographie, die Dokumentenstruktur und damit den verwendeten Dokumentenklassen auswirken wird.

Seit über 500 Jahren leben wir mit der Vorstellung, dass der letzte Akt bei der Erstellung eines Textes der Druck ist. Danach kann man ihn binden, verkaufen, ausleihen, weitergeben oder was immer man will, denn man hat »ihn« als etwas Greifbares in den Händen: als Buch oder Broschüre oder Handzettel.

Diese Idee führte dazu, dass sich im europäischen Verlagswesen und auch andernorts Konventionen konsolidierten, darüber wie der Satz von Schriftstücken funktioniert, von denen ein Teil aus den Zeiten der mittelalterlichen Handschriften stammen.

- Das Dokument besteht aus rechteckigen Seiten, die zu einem Buch zusammengefügt werden.
- Der Text beginnt am Anfang, in der richtigen Ecke, und wird Symbol für Symbol bis zum Ende fortgesetzt.
- In dieser Abfolge kann der Text nach einem vom Autor festgelegten begrifflichen oder logischen Plan in Abschnitte unterteilt werden, die der Orientierung oder der Information der Leser dienen.
- Es kann weitere Wegpunkte oder Meilensteine geben, um dem Leser zu zeigen, wo er sich im Dokument befindet, und um es ihm zu ermöglichen, anderen mitzuteilen, wie sie etwas von Interesse finden können.
- Als wir von Schriftrollen zu Seiten übergangen, scheint der Wunsch des Menschen nach Ordnung im Chaos einige Vorstellungen hervorgebracht zu haben, wie Dinge üblicherweise aussehen:
 - alle Seiten sollen die gleiche Größe haben;

¹ Übersetzung von Patrick Bideault und Bernd Raichle.

- sie sollen ungefähr gleich aussehen oder einer begrenzten Anzahl von Mustern folgen;
- sie sollen in der Regel die gleiche Zeilenzahl pro Seite haben; und auch wenn anderes Material (Mathematik, Musik, Abbildungen, Tabellen) eingefügt wird, soll die Positionierung der übrigen Zeilen einheitlich sein.

Dies nicht nur, um Schriftstücke leichter binden zu können, sondern auch, um sie leichter lesbar zu machen, und weil die Leute, die die Bücher gedruckt und veröffentlicht haben, schließlich wollten, dass ihre Ausgaben untereinander einheitlich sind, sich aber dennoch von denen der anderen unterscheiden.

Legt man die Idee des Drucks von Schriftstücken beiseite, bleibt die Idee einer PDF- oder Web-Seite auf dem eigenen Bildschirm. Diese können sogar wie eine gedruckte Seite aussehen, aber natürlich ist es nur ein Haufen eingefärbter Punkte. Dennoch behalten wir die meisten der oben aufgeführten Eigenschaften und Konventionen bei, weil sie für die Leser nützlich sind [2] – oder weil wir hoffen, dass sie es sind.

Es gibt eine Vielzahl von Meinungen, teils durch Forschungsergebnisse gestützt, teils aber auch nicht, dass man das PDF-Format *nicht* für andere Zwecke als den Druck verwenden solle (beispielsweise für Web->Seiten<), da es im Vergleich mit klassischem HTML zu ernstern Problemen bei der Benutzerfreundlichkeit kommen könne:

PDFs sind für die Verteilung von Dokumenten gedacht, die Benutzer ausdrucken werden. Sie sind für Papierformate optimiert, nicht für Browser-Fenster oder die Bildschirme moderner Geräte. Wir erleben oft, dass Benutzer in PDFs verloren gehen, weil die druckorientierte Ansicht nur einen kleinen Einblick in den Inhalt bietet. Benutzer können in einer PDF-Datei nicht wie auf einer Web-Seite suchen und blättern. Der Inhalt ist auf mehrere Blätter verteilt, was für gedruckte Dokumente in Ordnung ist, aber online zu erheblichen Problemen bei der Benutzerfreundlichkeit führt. [6]

Als gängige Praxis werden ohnehin mehrere Formate veröffentlicht, wobei zunehmend HTML5 mit CSS3-Paged-Media-Fähigkeiten empfohlen wird [1]. Die Verwendung von PDF ist jedoch in vielen Fällen aus technischen oder kleinkarierten politischen Gründen unvermeidbar, insbesondere wegen der mit L^AT_EX erreichbaren Genauigkeit, die in Browsern selbst mit HTML5/CSS3 oft nicht verfügbar ist. Unabhängig davon, welches Format Sie wählen (oder ob Sie verpflichtet sind, es zu verwenden), ist es wichtig, dass das Dokument gemäß den in Ihrem Fachgebiet vorherrschenden Richtlinien zugänglich gemacht wird.

Seitenzahlen. Als sich das Web und andere Formen des vernetzten elektronischen Publizierens durchsetzten, hatten viele akademische Zeitschriften und Zitierformate, die an die Angabe von Seitenzahlen gewöhnt waren, ernsthafte Bedenken,

denn eine Web-Seite ist überhaupt keine Seite – sie ist im Grunde wie eine endlose Schriftrolle, die ein ganzes Buch oder sogar eine ganze Sammlung von Büchern aufnehmen kann, ohne dass eine Seitenzahl zu sehen ist. Bei EPUB-Büchern ändern sich die Seitenzahlen jedes Mal, wenn Sie die Ansicht vergrößern oder verkleinern, um eine bessere Passform oder Schriftart zu erhalten. Zitierformate, die Seitenzahlen zwingend vorschreiben, wurden sogar von einigen, die inzwischen nur noch elektronisch publizieren, als altmodisch angegriffen. Einige Formate blieben hartnäckig und bestanden auf Seitenzahlen, selbst für seitenlose Dokumente. Diese Panik ist weitgehend vorbei, und viele Zeitschriften übernehmen jetzt Seitenzahlen aus dem PDF-Format nachträglich in die Web-Version (was mit \TeX relativ trivial ist).

Ränder. Gedruckte Bücher und Zeitschriften werden abhängig von der Wahl eines ein- oder doppelseitigen Seitenlayouts am linken oder rechten Rand gebunden, was bedeutet, dass der innere Rand größer sein muss als der äußere, um die Krümmung der Seiten in der Nähe des Buchrückens zu berücksichtigen, wenn das Buch aufgeschlagen ist. In der Vergangenheit wurden die Ränder bei der Buchgestaltung sowohl in den Handschriften als auch im Buchdruck mit großer Sorgfalt und Aufmerksamkeit behandelt, wie das berühmte Diagramm von Tschichold [8] auf Seite 67 zeigt. Die meisten gedruckten Dokumente wurden traditionell im Blocksatz gesetzt, was selbst im Handsatz viel einfacher war als bei Handschriften, so dass die Idee, dass der Text auf jeder Seite ein Rechteck mit festen Abmessungen einnimmt, eine leicht zu befolgende Konvention war.

In einem Format, das für das Lesen am Bildschirm konzipiert ist, sind die ungleichen, aber symmetrischen Ränder wahrscheinlich eine unnötige Ablenkung, es sei denn, man geht davon aus, dass Leser eine Software zur Anzeige von gegenüberliegenden Seiten verwenden. Einige Verlage erstellen neben druckfertigen zusätzlich anzeigeoptimierte PDF-Dateien, damit die Online-Leser die für den Druck vorgesehenen ungeraden und geraden Ränder nicht sehen.

Zeilen. Während die Anzahl der Zeilen pro Seite in einer PDF-Datei gesteuert werden kann, ist dies im Web sinnlos und macht eine EPUB-Datei praktisch unbrauchbar, da diese beiden Formate so konzipiert sind, dass sie vom Leser in der Anzeigegröße angepasst werden können. Ohnehin ist die Ausrichtung der Zeilen über eine aufgeschlagene Doppelseite hinweg in einem Browser oder einem Lesegerät nicht sinnvoll, außer es gibt eine Anzeigemöglichkeit von gegenüberliegenden Doppelseiten. Das Problem des ›Durchscheinens‹, bei dem die ausgerichteten oder ›verschobenen‹ Zeilen einer bedruckten Vorder- oder Rückseite eines Blattes durch dünnes Papier hindurch sichtbar sind, ist ganz klar ein Konzept, das nur für den Druck gilt.

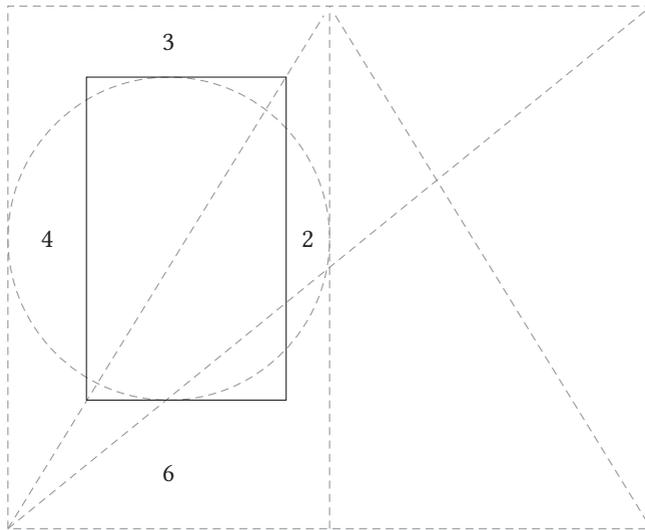


Abb. 1: Entwurf der Seitenproportionen (nach Tschichold [7, 8], zitiert nach Lewis [5] oder Kohm [4])

Fragen. Worauf sollten wir also achten, wenn wir einen Text nur für das Lesen außerhalb von Printmedien formatieren? Vielleicht können folgende Punkte als Anfang dienen:

- Format einer ›Seite‹ (Fenster- oder Bildschirm-Format ist vielleicht ein besserer Begriff): Hochformat wie ein Office-Dokument oder Querformat wie ein moderner Bildschirm?
- Ränder: Wenn sie nicht mehr asymmetrisch sein müssen, wie groß sollten sie dann sein?
- Zeilenlänge: Im Querformat gibt es mehr Platz, aber wir sollten ihn nicht auf Kosten der Lesbarkeit nutzen.
- Schriftgröße und Zeilenabstand/Durchschuss: Wie kann man damit eine größere Zeilenlänge ausgleichen?
- Nummerierung von ›Seiten‹: Ist sie überhaupt notwendig?
- Anzahl der Zeilen pro Seite oder Bildschirm oder Fenster: Ist dies wichtig?
- Konsistenz und Ähnlichkeit: Müssen diese beibehalten werden, wenn man mehr als ein Dokument als Teil einer Serie veröffentlichen wird?
- Struktur des Dokuments: Eine Form der Unterteilung in Abschnitte wird wahrscheinlich weiterhin erforderlich bleiben; wenn es keine Seitennummerierung gibt, wird eine Art von Nummerierungsschema benötigt.

Papier als Medium wird in absehbarer Zeit nicht verschwinden, aber wenn sich unsere Lesegewohnheiten zu ändern beginnen, lohnt es sich nachzudenken, wie sich dies auf unsere Dokumentenklassen auswirken wird.

Literatur

- [1] CSS Paged Media Module Level 3: Working Draft, Boston, Massachusetts, USA, 2018, <https://www.w3.org/TR/css-page-3/>.
- [2] Peter Flynn: Digital Typography, (Hrsg.: K. Norman, J. Kirakowski), Wiley, Hoboken, New Jersey, USA, 2018, S. 89–108, DOI <https://doi.org/10.1002/9781118976005>.
- [3] — »Typographers’ Inn - To print or not to print«, *TUGboat*, 41.3 (2020), 265–268, <https://tug.org/TUGboat/tb41-3/tb129inn.pdf>.
- [4] Markus Kohm: »Satzspiegelkonstruktionen im Vergleich«, *„Die T_EXnische Komödie“*, 14.4 (2002), 28–48, <https://archiv.dante.de/tex/typographie/satzspiegel/>.
- [5] J. Lewis: *Typography: basic principles: influences and trends since the 19th century*, Studio Books, London, 1963.
- [6] J. Nielsen, A. Kaley: *Avoid PDF for On-Screen Reading*, Nielsen Norman Group, 2020, <https://www.nngroup.com/articles/avoid-pdf-for-on-screen-reading/>.
- [7] Jan Tschichold: *Designing Books: Planning a book; a typographer’s composition rules; fifty-eight examples by the author*, Wittenborn, Schultz, New York, 1951.
- [8] — »Willkürfreie Maßverhältnisse der Buchseite und des Satzspiegels« in *Ausgewählte Aufsätze über Fragen der Gestalt des Buches und der Typographie*, Birkhäuser Verlag, Basel, 1987, S. 45–75, ISBN: 3-7643-1946-1.

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss in umgekehrter chronologischer Reihenfolge vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste oder als RSS-Feed auf <https://ctan.org/> verfolgt werden.

xduts von *Qi Xu* ist ein Bundle aus Klassen und Paketen für Arbeiten an der chinesischen *Universität für Elektrotechnik und Elektronik Xi'an*.

CTAN:macros/xetex/latex/xduts

knuth-hint von *Andreas Scherer* enthält eine große Sammlung von HINT-Dokumenten von *Donald Knuth* zu CWEB- und WEB-Quellen, die in \TeX Live enthalten sind. Darunter sind auch die Errata zu *Computers & Typesetting*.

CTAN:info/knuth-hint

hsindex von *Jean-Luc Joulin* ist eine Alternative zu *xindy*, um Indizes zu erzeugen. Grund für die Entwicklung des Programms war, dass *xindy* unter Debian nicht verfügbar bzw. nicht brauchbar ist.

CTAN:indexing/hsindex

commonunicode von *Pablo Pizarro R.* erweitert den Zeichensatz, den man direkt in einem \LaTeX -Dokument eingeben kann, um zahlreiche weitere Unicode-Symbole.

CTAN:macros/latex/contrib/commonunicode

simplenodes von *Bob Vergauwen* erstellt einfache farbige rechteckige »Knoten«, in die man Text setzen und die man untereinander mit Pfeilen verbinden kann.

CTAN:graphics/pgf/contrib/simplenodes

annotate-equations von *ST John* ermöglicht es, Formeln mit erläuternden Beschriftungen zu versehen.

CTAN:macros/latex/contrib/annotate-equations

ieejtran und *jieeetran* von *Haruki Ejiri* sind inoffizielle Bib \TeX -Stile für Aufsätze in Zeitschriften des *Institute of Electrical Engineers of Japan (IEEJ)*.

CTAN:biblio/bibtex/contrib/ieejtran

CTAN:biblio/bibtex/contrib/jieeetran

letgut von *Denis Bitouzé* enthält die Klasse für den Newsletter »La Lettre GUTenberg« der französischen \TeX -Anwendervereinigung GUTenberg.

CTAN:macros/luatex/latex/letgut

seu-ml-assign von *Teddy van Jerry* ist eine Vorlage für wissenschaftliche Arbeiten im Fach Maschinelles Lernen der Southeast University im chinesischen Nanjing.

CTAN:macros/latex/contrib/seu-ml-assign

luaaddplot von *Reinhard Kotucha* ergänzt *pgfplots*, so dass Daten ergänzt werden können, während sie beim Erstellen von Plots eingelesen werden. Das Paket funktioniert mit $\text{Lua}\TeX$ und $\text{Lua}\LaTeX$.

CTAN:macros/luatex/generic/luaddplot

vntex-nonfree von *Reinhard Kotucha* enthält die Schriften URW Classico und URW Garamond Nr. 8 von *Hàn Thê Thành* mit Erweiterungen für Vietnamesisch. Beide Fonts können mit *getnonfreefonts* installiert werden.

CTAN:fonts/vntex-nonfree

jmsdelim von *Jonathan Sterling* ermöglicht eine verbesserte Anpassung von Trennzeichen im Mathematiksatz.

CTAN:macros/latex/contrib/jmsdelim

functional von *Jianrui Lyu* verfolgt das Ziel, den Programmier-Layer `expl3` von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ über ein funktionales $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ -Interface zugänglich zu machen, damit »normale« Anwender einfacher mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ zurecht kommen.

CTAN:macros/latex/contrib/functional

messagepassing von *Martin Vassor* stellt eine Umgebung bereit, um den Nachrichtenaustausch in einem Programm grafisch darzustellen.

CTAN:graphics/pgf/contrib/messagepassing

liftarm von *Matthias Floré* zeichnet »Liftarme«, wie sie von *Lego Technic* her bekannt sind.

CTAN:graphics/pgf/contrib/liftarm

robotarm von *Marcus J. W. Snippe* liefert die technische Zeichnung eines zweidimensionalen mehrgliedrigen Roboterarms mit Beschriftungen.

CTAN:graphics/pgf/contrib/robotarm

hvpymgentex von *Herbert Voß* ist eine weitere Lösung, um Quelltexte schön zu setzen. Es ist kompatibel zu dem Paket `pygmentex` von *José Romildo Malaquias*, benötigt aber keinen separaten Python-Lauf, weil der Formatierer `Pygments` aus dem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Dokument heraus aufgerufen wird.

CTAN:macros/latex/contrib/hvpymgentex

njustthesis von *Wu Zhenyu* ist eine Vorlage für Abschlussarbeiten an der *Nanjing University of Science*.

CTAN:macros/latex/contrib/njustthesis

beamertheme-arguelles von *Michele Piazzai* ist ein schlichtes schwarz-weißes Beamer-Theme, das die Schriftart `alegreya` einsetzt.

CTAN:macros/latex/contrib/beamer-contrib/themes/beamertheme-arguelles

create-theorem von *Jinwen Xu* stellt Theorem-ähnliche Umgebungen bereit, die mit einem Key-Value-Interface vielsprachig konfiguriert werden können.

CTAN:macros/latex/contrib/create-theorem

atendofenv von *Fangyi Zhou* erlaubt es, Umgebungen mit einem beliebigen Symbol abzuschließen; es wird am Ende der Umgebung ausgegeben.

CTAN:macros/latex/contrib/atendofenv

grading-scheme von *Maximilian Keßler* dient zum Setzen eines grafischen Benotungsschemas für Klausuren.

CTAN:macros/latex/contrib/grading-scheme

hexboard von *Peter Rowlett* kann man verwenden, um Stellungen und Züge des Spiels Hex grafisch darzustellen.

CTAN:graphics/pgf/contrib/hexboard

linebreaker von *Michal Hoftich* setzt Lua \LaTeX ein, um übervolle Boxen zu vermeiden. Wenn eine Zeile über die Ränder der Box hinausragt, wird der Text noch einmal mit anderen Größen für `\tolerance` und `\emergencystretch` umbrochen, bis es passt.

CTAN:macros/luatex/latex/linebreaker

luacensor von *Elijah Z. Granet* dient dazu, mit Lua \LaTeX bestimmte Textteile eines Dokuments zu schwärzen, um sie vor der Veröffentlichung zu anonymisieren. Zur Barrierefreiheit wird ein Hinweis für Screenreader eingefügt.

CTAN:macros/luatex/latex/luacensor

andika von *Bob Tennent* ist ein Textfont für Leseanfänger in drei Schnitten samt der dazugehörenden \LaTeX -Unterstützung.

CTAN:fonts/andika

xcharter-math von *Daniel Flipo* ist ein neuer Unicode-Mathematikfont, der als Ergänzung zu *xcharter* von *Michael Sharpe* vorgesehen ist.

CTAN:fonts/xcharter-math

proflycee von *Cédric Pierquet* ist eine Handreichung für Lehrer von 15- bis 18-jährigen Schülern an französischen Gymnasien.

CTAN:macros/latex/contrib/proflycee

byrne von *Sergey Slyusarev* enthält eine MetaPost-Bibliothek und ergänzende Makros für Lua \LaTeX , mit denen man geometrische Beweise ähnlich wie in der Ausgabe von *Euklids* »Elementen« von *Oliver Byrne* setzen kann.

CTAN:graphics/metapost/contrib/macros/byrne

pbibtex-manual von der *Japanese \TeX Development Community* enthält die neue Dokumentation für p \BibTeX and up \BibTeX .

CTAN:biblio/pbibtex/pbibtex-manual

latex-for-undergraduates von *Andrew Lounsbury* ist eine Einführung in \LaTeX für Studienanfänger mit Mi \KTeX und dem Editor Visual Studio Code.

CTAN:info/latex-for-undergraduates

talos von *Antonis Tsolomitis* ist ein griechischer Font, der an der Universität Kreta in den 1980er-Jahren verwendet wurde. Dort entstand damals die erste \TeX -Installation an einer griechischen Universität (und damit wahrscheinlich auch die erste \TeX -Installation mit Unterstützung der griechischen Sprache überhaupt).

CTAN:fonts/greek/talos

zref-vario von *Gustavo Barros* vereint *varioref* und *zref-clever* miteinander.

CTAN:zref-vario

clistmap von *Erwann Rogard* stellt ein Interface für Paketautoren bereit, mit dem man L^AT_EX3 clisten verarbeiten kann.

CTAN:macros/latex/contrib/clistmap

The 28th GUST T_EX conference

Jerzy Ludwichowski

Dear T_EX users and friends,

if the times were normal, a few days ago we would have had returned from BachoT_EX 2022 and would not have lost the BachoT_EX conferences in the years 2020 and 2021. Fortunately, as far as the SARS-CoV-2 virus is concerned the prognosis seems optimistic for the coming months.

Despite yet another danger looming in the east, we decided to organize this years conference in the fall. To make up for the lost time we'll keep the same theme as the cancelled conference in 2020.

Please find at <http://gust.org.pl/bachotex/2022-en> the

Call for Papers for BachoT_EX 2022
from 21st September until 25th September
in Bachotek, Poland

Registrations will be announced separately as soon as possible.

Jerzy Ludwichowski
(for the Organizing Committee)

Bücher

Beyond The Archive – Von der Gießerei zum Klingspor Type Archive

Karl-Heinz Ohnemus

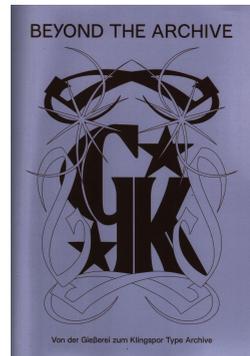
Beyond The Archive – Von der Gießerei zum Klingspor Type Archive

Hrsg.: Klingspor Museum Offenbach & Hochschule für Gestaltung Offenbach

Redaktion und Kuration der Ausstellung: Laura Brunner, Samuel Götschin, Paula Heinrich, Helene Hohmann, Genta Kozaki, Leonie Martin, Merle Schmidt, Franz Weid

2022, Offenbach am Main, broschiert, 132 Seiten, 1. Auflage: 175 Stück

Verkaufspreis: 8,- € plus Versandkosten



Unter dem Titel »*Beyond The Archive – Von der Gießerei zum Klingspor Type Archive*« veröffentlichte das Klingspor Museum Offenbach in Kooperation mit der Offenbacher Hochschule für Gestaltung einen Begleitkatalog zur gleichnamigen Ausstellung. Typografie-Freundinnen und -Freunden sei ein Ausstellungsbesuch sowie der nur in kleiner Auflage erschienene Ausstellungskatalog wärmstens empfohlen.

Das renommierte *Klingspor Museum* und Studierende der *Hochschule für Gestaltung* konzipierten und kuratierten gemeinschaftlich unter dem Titel »*Beyond The Archive – Von der Gießerei zum Klingspor Type Archive*« in Offenbach sowohl die Ausstellung als auch den jetzt erhältlichen, empfehlenswerten gleichnamigen Ausstellungskatalog. In Fachbeiträgen und Interviews wird nachgezeichnet, welche Bedeutung die Schriftgießerei der Gebrüder Klingspor in ihrem Offenbacher Werk in der ersten Hälfte des zwanzigsten Jahrhunderts für die Entwicklung der Typografie hatte. Wir erfahren viel über die einzelnen Arbeitsschritte, die notwendig waren, um eine Schrift zu entwerfen und herzustellen. Umfangreiche Arbeitsproben

aus der Schriftenentwicklung im Hause und ein reichhaltiges Bildmaterial illustrieren exemplarisch die typografische Qualität der Exponate. Zeitweise hatte die Schriftgießerei der Gebrüder Klingspor bis zu 350 festangestellte Mitarbeitende; die im Hause entworfenen, gefertigten Schriften wurden weltweit verkauft. Es werden Exponate der Jugendstilschriften der Schriftkünstler Otto Eckmann und Peter Behrens gezeigt, sowie die kalligrafisch anmutenden Frakturschriften von Rudolf Koch. Typografie ist immer mit dem jeweiligen Zeitgeist verbunden, doch sie weist auch über die Gegenwart hinaus.

Martina Weiß, Bibliothekarin des Klingspor Museums erläutert, was sich in den Archiven des Museums befindet und zeichnet nach, wie die meisten der erhaltenen Schriftproben-Schätze der Schriftgießerei Klingspor ins Archiv kamen. Das *Klingspor Type Archive* will diese Schätze digitalisieren und der Öffentlichkeit zugänglich machen. Historikerin Andrea C. Hansert zeichnet nicht nur die Firmengeschichte der Schriftgießerei Klingspor nach, sondern untersuchte auch die Verbindungen der Gebrüder Klingspor mit der Nazidiktatur in der NS-Zeit. Peter Gerstenberg, einer der letzten noch beruflich tätigen Schriftgießer (Landesmuseum Darmstadt) berichtet in seinem Beitrag über seinen fast ausgestorbenen Beruf. Er beliefert Kund_innen in aller Welt mit Bleilettern. Der frühere Typodesigner Dan Reynolds stellt die wichtigsten Schriftgestalter der Gebr. Klingspor Gießerei vor. Sein Beitrag endet mit dem Typografen Hermann Zapf, dem damaligen künstlerischen Leiter der Frankfurter D. Stempel AG, welche die Gießerei der Gebrüder Klingspor 1956 übernahm. Designhistorikerin Julia Meer reflektiert in einem bemerkenswerten Artikel kritisch über das Ausstellungsmachen und hinterfragt, welches Material den Weg in die Archive findet. Was wird gesammelt und soll der Nachwelt erhalten werden? Und was nicht? Und wer bestimmt mit welcher Erzählung darüber? In den Exponaten in den Archiven der Museen zur Schriftgießerei dominiert der männliche Blick. Das heißt aber nicht, dass es nicht auch viele weibliche Beschäftigte in den Gießerei-Betrieben gegeben hat.

Ein kleines Glossar von Bibliophilie über Fotosatz bis hin zur Variable Font erläutert die gußtechnischen und typografischen Fachbegriffe. In prägnanten, übersichtlichen Definitionen wird erklärt, was man unter Fonts, einer Foundry, Glyphen, Ligaturen, dem Spacing und unter Versalien versteht. Ich habe den auch optisch anregend gestalteten Ausstellungskatalog mit großem Interesse und Gewinn gelesen. Lediglich bei den Textseiten mit roter Schrift auf weißem oder grauen Grund taten sich meine Augen beim Lesen etwas schwer.

Zu den bemerkenswerten Leistungen von Katalog und Ausstellung gehört, dass Lesende und Besuchende dazu animiert werden, sich ein eigenes Bild zu machen. Was steckt im und hinter dem Archiv? Wer erzählt was mit den Exponaten? Ausstellungskataloge erzählen linear. In der Ausstellung selbst wird keine fertige Erzählung

geliefert, die alles dominiert. Für Museumsleiterin Dorothee Ader kommt es darauf an, Fragen zu stellen. Dazu den Anstoß zu geben ist den teilnehmenden, kuratierenden Studierenden der Hochschule für Gestaltung (HfG) in Kooperation mit dem Klingspor Museum rundweg gelungen.

Links:

Klingspor Museum Offenbach: <https://www.klingspor-museum.de/>

International Type Design Archive: <https://www.klingspor-museum.de/Kuenstler.html>

Klingspor-Institut für Schriftgestaltung: <http://www.klingspor-institute.com/>

L^AT_EX Beginner's Guide, 2. Auflage

Uwe Ziegenhagen

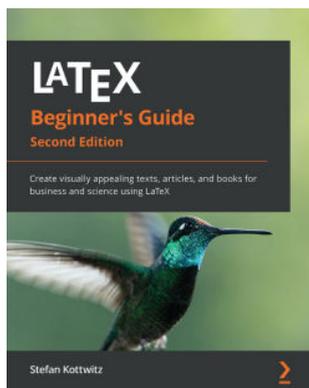
Stefan Kottwitz:

L^AT_EX Beginner's Guide

2. Auflage 2022

ISBN-13: 978-1801078658

32,08 €



In der T_EX-Welt ist der Name Stefan Kottwitz vor allem für die von ihm betriebenen Webseiten und -foren bekannt, Stefan ist jedoch auch Buchautor. Nachdem er 2011 die erste Auflage seines L^AT_EX-Einsteigerbuchs veröffentlichte, auf das 2015 dann das »L^AT_EX Cookbook« folgte, liegt jetzt die zweite Auflage des Einsteigerbuchs »L^AT_EX Beginner's Guide« vor.

Das Buch ist bei Packt erschienen, der Verlag hat mir freundlicherweise ein Exemplar zur Verfügung gestellt, das ich im Rahmen der Vorbereitung mehrerer L^AT_EX-Kurse gleich einem intensiveren Blick unterzogen habe.

Auf etwas mehr als 300 Seiten führt Stefan in die verschiedenen Aspekte von L^AT_EX ein. Auf 25 Seiten werden die Geschichte von T_EX und L^AT_EX kurz umrissen, sowie die Installation von T_EXLive bzw. Nutzung von Overleaf erläutert. Im zweiten

Kapitel geht es dann um die Formatierung von Text. Gut gefällt mir, das in diesem Kapitel bereits die Erstellung eigener Befehle gezeigt wird. Dies mache ich auch in meinen Kursen so, um den Teilnehmerinnen und Teilnehmern zu zeigen, wie leicht sich die Nutzung von komplexeren Auszeichnungen mit selbstdefinierten Befehlen vereinfachen lässt.

In Kapitel 3 wird gezeigt, wie man Seitenformate und Satzspiegel einstellt, gefolgt von der Erläuterung der verschiedenen Listentypen in Kapitel 4. Kapitel 5 und 6 beschreiben den Satz von Abbildungen bzw. Tabellen, gefolgt von Querweisen in Kapitel 7.

Kapitel 8 zeigt, wie man Inhaltsverzeichnisse anpasst und einen Index erstellt. Außerdem geht Stefan in diesem Kapitel auf den Satz von Bibliografien mit der Standard-Bibliografie-Umgebung und Br_BT_EX ein. Bib_LA_TE_X bleibt an dieser Stelle leider außen vor, Stefan beschreibt es aber in seinem »L^AT_EX Cookbook«.

Das 9. Kapitel ist der Mathematik gewidmet, hier werden alle Konstrukte abgedeckt, die einem Studenten oder einer Studentin selbst in einem MINT-Fach begegnen können. Kapitel 10 geht auf die verschiedenen Schriften ein, die man in pdf_LA_TE_X nutzen kann, und zeigt auch, wie mit X_YL_AT_EX oder Lua_LA_TE_X beliebige Systemschriften genutzt werden können. Kapitel 11 zeigt ausführlich, wie längere Dokumente mit `\input` und `\include` in einzelne Teile aufgeteilt werden können.

Das 13. Kapitel dürfte insbesondere für L^AT_EX-Frischlinge hochrelevant sein, denn hier wird gezeigt, wie man im Fehlerfall die Probleme näher eingrenzen und beheben kann.

Das Buch endet mit einer ausführlichen Beschreibung von Webforen und -seiten, auf denen man Unterstützung erhalten kann und stellt auch Usergroups wie T_EX Users Group und DANTE e.V. vor. Die Webseite zum Buch findet man unter <https://latexguide.org/>. Dort gibt es dann neben der FAQ zum Buch auch die Quellcodes mit Online-Kompilierung.

Zusammenfassend kann ich sagen, dass mir das Buch von Stefan sehr gut gefallen hat und ich es auch in meinen Kursen empfehlen werde. Zusammen mit dem »L^AT_EX Cookbook«, in dem neben Bib_LA_TE_X auch das Erstellen von Grafiken mittels L^AT_EX beschrieben wird, bietet es einen guten Einstieg in das Thema L^AT_EX.

Edition dante – Neuauflage

Herbert Voß

**Einführung in L^AT_EX –
pdfL^AT_EX LuaL^AT_EX XeL^AT_EX**

4. Auflage 2022, vierfarbig,
DANTE e.V. und Lehmanns Media,
viii+976 Seiten; ISBN 978-3-96543-296-3;
39,95 € (Ladenpreis) bzw. 35,- € 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 <https://archiv.dante.de/dante/zahlung/zahlung.php>.



Bitte beachten Sie für Bestellungen bei DANTE e.V. folgende Informationen zum Widerrufsrecht: Käufer 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. Der Besteller hat in jedem Fall die Rücksendekosten zu tragen. Bei Verschlechterung der Ware, die über die übliche Prüfung der Ware hinausgeht, hat der Besteller gegebenenfalls Wertersatz zu leisten.

Spielplan

12. 3. – 29. 5. 2022 **Von der Gießerei zum Klingspor Type Archive**
Ausstellung »Beyond the Archive«
Klingspor Museum Offenbach

23. 6. – 25. 6. 2022 **DANTE 2022**
und 64. Mitgliederversammlung von DANTE e.V.
Otto-von-Guericke-Universität
39016 Magdeburg
<https://www.dante.de/veranstaltungen/dante2022/>



22. 7. – 24. 7. 2022 **TUG 2022 – Presentations covering the T_EX world**
The 43rd Annual Conference of the T_EX Users Group
Die Tagung findet erneut online statt.
<https://www.tug.org/tug2022/>



September 2022 **Herbsttagung**
und 65. Mitgliederversammlung von DANTE e.V.
Landesinstitut für Pädagogik und Medien
Beethovenstraße 26
66125 Saarbrücken

12. 9. – 18. 9. 2022 **ConT_EXt Meeting 2022**
Evangelische Freizeitstätte in Dreifelden/Westerwald
<https://www.rittersmorgen.de>



21. 9. – 25. 9. 2022 **28th GUST T_EX conference**
Bachotek (nahe Brodnica), Polen
<http://gust.org.pl/bachotex/2022-en>



Stammtische

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



Aachen

Torsten Bronger

bronger@physik.rwth-aachen.de

Mailingliste:

<https://lists.rwth-aachen.de/postorius/lists/tex-stammtisch.lists.rwth-aachen.de>

»Anvers«, Kockerellstr. 20, 52062 Aachen

Erster Donnerstag im Monat, 20:00 Uhr



Rostock/Barth

Carsten Vogel

texnicer@web.de

auf Anfrage

Berlin

Michael-E. Voges, Tel.: 0 33 62/ 50 18 35,

mevoges@t-online.de

z. Zt. online wegen Corona. Link siehe Mailingliste dante-ev oder auf Anfrage.

Darmstadt

Karlheinz Geyer

geyerk@posteo.de

pausiert z. Zt. wegen Corona

findet voraussichtlich 2022 an jedem ersten Freitag im Monat wieder statt - Planungen laufen

Erlangen

Peter Seitz

p.seitz@KplusS-Ing.de

<https://www.ks-ingenieurconsult.de/TeX/Stammtisch.html>

Gaststätte »Deutsches Haus«, Luitpoldstr. 25, 91052 Erlangen

Dritter Dienstag im Monat, 19:00 Uhr



Frankfurt a. Main

Harald Vajkonny

vajkonny@t-online.de

*zur Zeit inaktiv, Interessenten bitte per Mail melden***Göttingen**

Holger Nobach

holger.nobach@nambis.de

<http://goetex.nambis.de/>*Restaurant »Mazzoni Cucina Italiana«, Hermann-Rein-Straße 2, 37075 Göttingen
Dritter Donnerstag im Monat, 18:00 Uhr***Hamburg**

Günther Zander

guenther.zander@lug-balista.de

*z. Zt. inaktiv. Bei Fragen steht Günther gern per Mail zur Verfügung.***Hannover**

Reiko Kaps

kaps@luis.uni-hannover.de

<http://tex-hannover.de/>*Zweiter Donnerstag im Monat, 18:30 Uhr**z. Zt. online über <https://vc.sonia.de/TexStammtischHannover>***Heidelberg**

Martin Wilhelm Leidig, Tel.: 01 70 41 83 32 9,

moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <https://tinyurl.com/stammtisch-HD>*Physische Treffen bleiben ausgesetzt, solange keine dauerhafte Besserung bzw. Beendigung der gegenwärtigen pandemischen Lage eingetreten ist.***Köln**

Uwe Ziegenhagen

uwe@dante.de

*zur Zeit inaktiv, Interessenten bitte per Mail melden***Leipzig**

Erhard Pross

Erhard.Pross@gmx.de

*halbjährliche Treffen, nächster Termin am 6.10.2022, 18:00 Uhr**le-tex publishing services GmbH, Weissenfeller Str. 84, 04229 Leipzig*

Stralsund

Heiner Richter

Heiner.Richter@hochschule-stralsund.de

z. Zt. inaktiv, Nachfolge in Planung

Stuttgart

Bernd Raichle

bernd.raichle@gmx.de

Zweiter Dienstag im Monat, 19:30 Uhr

zur Zeit online über Jitsi-Meeing, Abstimmung ob Präsenz oder online über Mailingliste.

Wuppertal

Andreas Schrell

as@schrell.de

Zweiter Donnerstag im Monat, 19:30 Uhr

»Restaurant Croatia«, Südstr. 10, 42103 Wuppertal

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 11 03 61
69072 Heidelberg

Tel.: (0 62 21) 2 97 66 (Mo., Mi., Do. von 9.00–11.30 Uhr)
Fax: (0 62 21) 16 79 06
E-Mail: info@dante.de

Konto: VR Bank Rhein-Neckar eG
IBAN DE67 6709 0000 0002 3100 07 SWIFT-BIC GENODE61MA2

Vorstand

Vorsitzender:	Martin Sievers	president@dante.de
stv. Vorsitzender:	Uwe Ziegenhagen	vice-president@dante.de
Schatzmeisterin:	Doris Behrendt	treasurer@dante.de
Schriftführer:	Volker RW Schaa	secretary@dante.de
Beisitzer:	Klaus Höppner Harald König Stephan Lukasczyk Herbert Voß	

Ehrenmitglieder

Peter Sandner	22.03.1990	Klaus Thull († 2012)	22.03.1990
Yannis Haralambous	05.09.1991	Barbara Beeton	27.02.1997
Luzia Dietsche	27.02.1997	Donald E. Knuth	27.02.1997
Eberhard Mattes	27.02.1997	Hermann Zapf († 2015)	19.02.1999
Joachim Lammarsch	12.04.2014	Rainer Schöpf	12.04.2014

Webserver und Mailingliste

DANTE: <https://www.dante.de/> (Erik Braun)
CTAN: <https://mirror.ctan.org/> (Gerd Neugebauer)
DANTE-EV: <https://lists.dante.de/mailman/listinfo/dante-ev>

FAQ

DTK: <https://projekte.dante.de/DTK/WebHome>

T_EX: <https://projekte.dante.de/DanteFAQ/WebHome>

T_EXnische Fragen

beraterkreis@dante.de

ak-schule@dante.de

Autoren/Organisatoren

- Adelheid Bonnetsmüller** [11] **Mathias Magdowski** [5]
bonnetsmueller@icloud.com
Otto-von-Guericke-Universität
Magdeburg
- Luzia Dietsche** [3] mathias.magdowski@ovgu.de
71394 Kernen
dtkred@dante.de
- Jürgen Fenn** [68] **Ralf Mispelhorn** [18,21]
Neu-Isenburg
juergen.fenn@gmx.de
Am Wettbach 12/2
72336 Balingen
mispelsoft@mispelhorn.de
- Peter Flynn** [64] **Karl-Heinz Ohnemus** [73]
Textual Therapy Division
Silmaril Consultants
Cork, Irland
Offenbach
- Stefan Kottwitz** [6] **Henning Hraban Ramm** [61]
stefan@latex.org
Wasgenstraße 21
14129 Berlin
herbert@dante.de
- Jerzy Ludwichowski** [72] **Uwe Ziegenhagen** [75]
Jerzy.Ludwichowski@gust.org.pl
Köln
- Martin Sievers** [4]
siehe Seite 82
- Herbert Voß** [30,77]

Die T_EXnische Komödie

34. Jahrgang Heft 2/2022 Mai 2022

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 5 Einladung zur Frühjahrstagung 2022 in Magdeburg
- 6 Bericht über Projektförderung von L^AT_EX-Servern

Bretter, die die Welt bedeuten

- 11 Having Fun with L^AT_EX: Klein- und großkariert
- 18 Quintenzirkel mit Gitarren-Akkorden
- 21 Erstellen eines Songbooks
- 30 Externe Dokumente/Programme
- 61 ConT_EXt kurz notiert

Von fremden Bühnen

- 64 Druck oder Nichtdruck
- 68 Neue Pakete auf CTAN
- 72 The 28th GUST T_EX conference

Bücher

- 73 Beyond The Archive – Von der Gießerei zum Klingspor Type Archive
- 75 L^AT_EX Beginner's Guide, 2. Auflage
- 77 Edition d_on'te – Neuauflage

Spielplan

- 78 Termine
- 79 Stammtische

Adressen

- 83 Autoren/Organisatoren