

pro^{ly}sa^{rik}

goLysa

Handbuch

Sisyphos₂

Geschrieben von
Marcus Daniel Cremer

Stand 31. Dezember 2023

Laut **cloc** umfaßte der Quelltext für den Interpreter bei der letzten Zählung 15.510 Zeilen (zuzüglich 1.113 Zeilen mit Kommentaren sowie 1.017 Leerzeilen).

Dieses Handbuch, gleichwohl im Format DIN A5 vorliegend, ist für die Betrachtung am Bildschirm gedacht. Vorzugsweise werden jeweils zwei Seiten nebeneinander dargestellt (ungerade Seiten rechts).

Sowohl das Handbuch als auch die HTML-Ausgabe von *Lysa* verwenden ein auf den Herbstfarben des Zuckerahorns basierendes Farbschema.

© 2020-2023 bei **Marcus Daniel Cremer**, Gelsenkirchen

Diese Dokumentation wird mit **LUA^ATEX** gesetzt. Für die Darstellung werden die **kpfonts** verwendet
(für Quellcode wird **STIX Two Math** 2.13 verwendet).

Inhaltsverzeichnis

Mein Lysa	7	Zeitraum	92
Versionshistorie	19	Untertitel	95
Programmaufruf	20	Taktgeber	96
Fremdpakete	21	Klang	98
Abschweifungen	22	Klangfilter	106
Vokabular	24	Klangprofil	109
Verben	24	Raumklang	112
Lysa	31	Licht	118
Klammerdialekt	33	Remission	122
Stapel	35	Farbe	127
Regal	38	Bild	137
Pronomen	43	Maske	146
Anführung	45	Histogramm	165
Ablauf	48	Maskenfilter	??
Entscheidung	50	Bildfilter	??
Zeichenkette	53	Dekor	??
Ganzzahl	55	Zeichnung	169
Zahl	57	Schrift	179
Dupel	65	Schriftzeichen	182
Tripel	68	Letter	189
Quadrupel	70	Zusammenhang	192
Tripelkarree	71	Zeile	198
Quadrupelkarree	77	Kolumne	202
Sammlung	78	Effekt	204
Verlauf	82	Animation	206
Entwicklung	89	Literaturverzeichnis	208

Mein Lysa

»Der Kampf gegen Gipfel vermag ein Menschenherz auszufüllen. Wir müssen uns Sisyphos als einen glücklichen Menschen vorstellen.« (ALBERT CAMUS [5, S. 160])

Lysa ist eine an Forth¹ angelehnte, interpretierte Programmiersprache, so wie unter anderem z. B. auch Postscript (Seitenbeschreibungssprache, Grundlage des PDF-Formats). Sie entsteht in Zusammenhang mit **proly-sarik**. Ihr Hauptzweck besteht darin, Animationen mit kinetischer Typographie zu erstellen.

Die erste Version von *Lysa* habe ich in Java geschrieben. Der Wechsel von einer zur anderen Sprache geschah nicht aus Abenteuerlust, sondern war jeweils durch die Möglichkeiten begründet, welche die neue Sprache gegenüber der alten bot. Von C++ mit **Qt** zu Java bin ich noch in Zeiten vor *Lysa* gewechselt, da Java durch seine Optimierungen zur Laufzeit schneller arbeitet als C++ mit Qt (jedenfalls mit dem von mir geschriebenen Kode), von Java zu Go bin ich gewechselt, da Go PNGs mit 16 Bit pro Farbkanal laden und speichern kann, C sollte es nun sein, weil ich mich unter Go beim Laden und Speichern der Bilder nur im sRGB-Farbraum bewegen kann. Aber leider bin ich darin gescheitert, eine eigene Speicherverwaltung zu programmieren, die auch nebenläufig funktioniert...

¹**Forth** ist eine »imperative, stackbasierte Programmiersprache«, die in die späten 60er-Jahren des 20. Jahrhunderts von **Charles H. Moore** entwickelt wurde [13].

Auch wenn ich Camus zustimme, daß ein stets neues Stemmen des Steines zum Gipfel ein Menschenherz ausfüllen kann, so wünsche ich mir insbesondere in jenen Momenten, in denen ich den Fels wieder hinabrollen sehe (wenn mir also eine mehr oder minder sinnvolle Änderung einfällt, die bedauerlicherweise die Umformulierung aller Quelltexte erfordert...), er möge doch zumindest nicht ganz bis zur Talsohle gelangen. Tatsächlich bringt mich jede Version ein Stück weiter. Und bei jedem neuen Anlauf zum Gipfel finde ich mich vor immer komfortabler ausgetretenen Pfaden. Zumindest gefühlt starte ich mittlerweile schon mindestens auf halber Höhe.

Andererseits bergen solche ausgetretenen Pfade die Gefahr der Betriebsblindheit. Bloß, weil ich etwas von Anfang an so und nicht anders gemacht habe, heißt das nicht, daß ein Anders nicht von Vorteil sein könnte. Die wieder aufgenommene Version in Go profitiert durchaus von dem ansonsten erfolglosen Ausflug in C. Zumal ich zuvor wie gesagt in Java geschrieben hatte, also einer objektorientierten Sprache, wohingegen C zu den imperativen Sprachen zählt, denen ich mich näher fühle, vielleicht weil ich darin vor langer Zeit zu programmieren begonnen habe. Go steht dazwischen. Jedenfalls haben sich durch die veränderte Perspektive einige Möglichkeiten ergeben, den Quellcode zu straffen. Das ist immer gut.

Solange ich mit und an Computern arbeite, war zu programmieren für mich Teil meines literarischen Schaffens. Ich fühle mich durch jene **Studie** der Universität Washington bestätigt, daß zu programmieren eher sprachliches als mathematisches Talent erfordert. Umso bedauerlicher empfinde ich die selbstverständliche Ausschließlichkeit, mit der Englisch als Grundlage des Programmierens genommen wird. Nichts gegen Englisch, wenn es die eigene Muttersprache ist, aber wieviel Kreativität geht verloren, wenn Gedanken und Konzepte aus der eigenen Muttersprache zunächst ins Englische und dann auch noch in eine Programmiersprache übersetzt werden müssen? Könnte jemand tatsächlich meinen, es würde die Literatur voranbringen, wenn Gedichte zukünftig nur noch auf Englisch verfaßt werden?

Ein Magier spricht Formeln und vollführt Riten, um Kräfte zu beschwören. Dabei appelliert er keineswegs an deren Verständnis, er betet nicht um ihre Gnade, sondern er gebietet ihnen. Das unterscheidet ihn von einem Priester. Damit es gelingt, darf er sich keinen Fehler erlauben. Schon die kleinste Abweichung kann zum Mißlingen des Unterfangens führen (das aus naturwissenschaftlicher Sicht mit hoher Wahrscheinlichkeit eintritt).

Der Programmierer schreibt Code, um eine Maschine zu steuern. Diese versteht seine Anweisungen nicht², befolgt sie aber, wenn sich darin kein Fehler findet. Kleinste Abweichungen im Code können dazu führen, daß ein Programm entweder gar nicht mehr oder aber nicht wie gewünscht abgearbeitet wird. Eigentlich sind Programmierer die Magier unserer Zeit (allerdings haben sie aus Sicht der Naturwissenschaften wesentliche günstigere Erfolgsaussichten). Handbücher wie dieses hier können mächtige Zauberbücher sein, denn sie enthalten die Formeln, den Kräften der Maschine zu gebieten, wenn einer denn gelernt hat, sie richtig zu gebrauchen.

Zu programmieren ist eine andere Art des Schreibens, eher gebieterisch³ als appellierend, doch zweifellos ist es eine Art des Schreibens, auch des Beschreibens. Um die Maschine zu steuern, muß ich Zustände und Handlungen so hinreichend genau beschreiben, daß die Maschine sie herstellen beziehungsweise ausführen kann. In der Regel wird auch nicht für die Maschine allein programmiert. Es ist sinnvoll, Code immer wieder zu verwenden. Darum wird er geteilt, und auch der Autor profitiert von verständlichem Code, wenn er ihn mit zeitlichem Abstand betrachtet, weil etwa Anpassungen notwendig sind.

Hauptsächlich wird Programmcode wie ein Gedicht geschrieben: Funktionen und Blöcke entsprechen den Strophen und die nach rechts ungleichmäßig auslaufenden Zeilen sind die Verse. Ein Text,

²Auch wenn sich das Programm, welches den Code ausführt, Interpreter nennt, findet sich darin nichts, was über Reflexbögen hinausginge.

³Wer allerdings erst einmal ein wenig Zeit mit Fehlersuche verbracht hat, der kommt sich kaum mehr als Gebieter, sondern eher als Knecht der Maschine vor!

in dem es um Verständlichkeit geht, wird jedoch zumeist in Prosa mit umgebrochenen Zeilen geschrieben. Ich finde, Forth und somit auch *Lysa* eignen sich nur eingeschränkt für eine Aufteilung in Verse und Strophen, da Sätze oft noch Nebensätze enthalten und das zum Teil verschachtelt.⁴ Deshalb schreibe ich den Kode wie Prosa. Ob er dadurch verständlicher wird? Er ähnelt dazu noch zu sehr einem frühmittelalterlichem Codex (und die waren nicht gerade berühmt für ihre Leserlichkeit). Eine Interpunktion wäre vermutlich hilfreich.⁵

Zu einer Programmiersprache, die von keinem anderen verwendet wird, bedarf es eigentlich keines Handbuchs, zumindest keines Textes, der eindeutig mehr ist als eine Gedächtnisstütze für den Autor. Daß ich mir dennoch die Arbeit mache, dieses Handbuch in einem möglichst allgemeinverständlichen Zustand zu halten, liegt daran, daß ich die Programme, die ich in *Lysa* schreibe, um mei-

⁴»An average sentence, in a German newspaper, is a sublime and impressive curiosity; it occupies a quarter of a column; it contains all the ten parts of speech – not in regular order, but mixed; it is built mainly of compound words constructed by the writer on the spot, and not to be found in any dictionary – six or seven words compacted into one, without joint or seam – that is, without hyphens; it treats of fourteen or fifteen different subjects, each enclosed in a parenthesis of its own, with here and there extra parentheses, which re-enclose three or four of the minor parentheses, making pens with pens; finally, all the parentheses and re-parentheses are massed together between a couple of king-parentheses, one of which is placed in the first line of the majestic sentence and the other in the middle of the last line of it – after which comes the verb, and you find out for the first time what the man has been talking about« (Mark Twain THE AWFUL GERMAN LANGUAGE, 1880)

⁵Interpunktion, die Auskunft über den Zustand des Stapels gibt:

- Ist nach einem Verb der Stapel leer, wird ein Punkt gesetzt
- Ist nach einem Verb der Stapel leer, weil eine Gegebenheit in ein Register geladen wurde, und modifiziert das nächste Verb diese Gegebenheit, so ist statt eines Punktes ein Doppelpunkt zu setzen
- Müßte üblicherweise nach einem Verb der Stapel leer sein, dort befinden sich jedoch noch eine oder mehrere Gegebenheiten zu einem späterem Gebrauch, wird ein Semikolon gesetzt
- Wird eine Gegebenheit auf den Stapel gelegt, die vom nächsten Verb nicht verwendet wird, wird ein Komma gesetzt

ne Videos zu produzieren, in gedruckter Form mit beiliegendem Datenträger (der das Video nebst allen Einzelbildern enthält) in wenigen, nummerierten und signierten Exemplaren herausgebe.⁶ Und dieses Handbuch findet sich auf dem Datenträger, damit der Programmcode verständlich werden kann.

Warum Forth? Mit der Sprache in Berührung gekommen bin ich erstmals in der ersten Hälfte der 80er Jahre des letzten Jahrhunderts. Ich hatte einen **ZX Spectrum** und Forth hatte ich mir zugelegt, da man damit Grafik programmieren können sollte. Insbesondere die umgekehrte polnische Notation (Postfixnotation) hat mich abgeschreckt.

Bei **proly-sarik** fand ich es erstrebenswert, immer wenn numerische Angaben zu machen sind, dies auch in der Form von Formeln machen zu können, mit der diese Angaben errechnet werden können, wenn sie dann gebraucht werden. In einer solchen Formel können auch Variablen verwendet werden, deren Werte sich je nach Situation ändern können. Folglich brauchte ich einen Formelparser.

Mit dem **mXparser** von Mariusz Gromada steht auch ein mächtiger Parser als freie Software zur Verfügung, nur leider stellte sich dieser für nebenläufige Berechnungen als nur eingeschränkt geeignet heraus. Wenn verschiedene nebenläufig arbeitende Programmteile dieselbe Formel verarbeiten, die darin enthaltenen Variablen aber jeweils etwas anderes bedeuten, funktioniert das nicht mehr richtig. Oder ich habe dabei etwas falsch gemacht. Das würde ich niemals ausschließen.

Jedenfalls habe ich mich dazu entschlossen, selbst einen Formelparser zu schreiben. Der gängige Weg dabei ist, die in der Infixnotation angegebenen Formeln unter Einsatz mehrerer Stapel mit dem **Rangierbahnhof-Algorithmus** in eine Postfixnotation umzuwandeln und dann mit entsprechenden Funktionen zu berechnen. Das erinnerte mich deutlich an Forth.

Nachdem ich das zu meiner Zufriedenheit umgesetzt hatte, wobei ich auch die Möglichkeit schuf, die Formeln direkt in einer Postfixnotation anzugeben, die ich bald nahezu ausschließlich nutzte,

⁶Ob sich dadurch wohl eine Aura bildet?

drängte sich mir zu meiner wachsenden Unzufriedenheit die Frage auf, wenn ich schon zur Berechnung numerischer Angaben eine Art Forth einsetze, warum nicht gleich die gesamte Skriptverarbeitung darauf umstellen? Also ein einziger Parser für das gesamte Skript.

In einem Skript gebe ich dem Anwender gewisse Möglichkeiten, und bemühe mich, Fehler möglichst auszuschließen, indem ich den Eingaben enge Grenzen setze. Jedes Kommando braucht bestimmte Parameter und sowohl dem Anwender als auch dem Entwickler (in diesem Fall ist das dieselbe Person) ist gedient, wenn dabei möglichst nur sinnvolle Eingaben möglich sind.

In einer Programmiersprache hingegen stelle ich dem Programmierer eher allgemein gehaltene Möglichkeiten zur Verfügung, ohne dabei genau zu wissen, was er im Einzelfall damit vorhat. Insofern verbietet sich weitgehend, Möglichkeiten einzuschränken. Dadurch nimmt zwar die Fehleranfälligkeit exorbitant zu, andererseits wachsen die kreativen Möglichkeiten des Anwenders entsprechend. Ein Skript zur Steuerung einer Anwendung ist also rein konzeptionell etwas ziemlich anderes als eine vollwertige Programmiersprache wie Forth. Aber es gibt wenig reizvollere Dinge für einen Schriftsteller, als sprachschöpferisch tätig zu werden. Und darum wird aus *prolysarik* *Lysa*, auch wenn dies einem konzeptionellen Spagat gleichkommt, denn die Sprache muß komfortabel genug sein, ein Ergebnis mit möglichst wenig Schreibarbeit zu erreichen, und zugleich flexibel genug, um Wege zu beschreiben, die zum Zeitpunkt, da Funktionen zur Verfügung gestellt werden, gar nicht absehbar sind. Mein Ansatz, dieses Dilemma zu überwinden besteht darin, sowohl komplexe als auch elementare Handlungen anzubieten.

Allerdings sind dann da noch die drei Grundsätze von Charles M. Moore, dem Schöpfer von *Forth*, zu berücksichtigen: »*Keep it simple!*«, »*Don't speculate!*« und »*Do it yourself!*« — Problematisch im Hinblick auf obiges Dilemma ist insbesondere die mittlere Maxime, denn sie verbietet geradezu alles, was nicht unmittelbar benötigt wird. Nichtsdestotrotz halte ich die Grundsätze für weit-

gehend sinnvoll. Deshalb habe ich Funktionalität ausgespart, die mir in Hinblick auf die Erstellung von Videos mit kinetischer Typographie nicht notwendig erscheint, auch wenn sie allgemein in Programmiersprachen eine Selbstverständlichkeit darstellen. So fehlt z. B. die Möglichkeit, beliebig auf Dateien zuzugreifen.

Infixnotation:	$(1 + 2) \times 3$
Postfixnotation:	$1\ 2\ +\ 3\ \times$

Was die Postfixnotation (abgesehen von der Reihenfolge) von der Infixnotation unterscheidet ist, daß keine Klammern benötigt werden. Genau aus diesem Grund wandelt ein Formelparser die eine Notation in die andere um. Denn in der Postfixnotation hat er die Angaben und Operatoren genau in der Reihenfolge, in der sie vom Rechner abzuarbeiten sind. In der Infixnotation bringen die Klammern die Reihenfolge in der Regel durcheinander. Das macht Formeln zwar für Menschen oft leserlicher, nicht jedoch für Computer.

Wenn etwas nicht mehr benötigt wird, steht es zur anderweitigen Verwendung zur Verfügung. Die Eingabe des originalen Forth besteht aus durch Leerzeichen getrennten Wörtern (obiges Beispiel für die Postfixnotation ist gültiger Forth-Code). In *Lysa* hingegen werden Klammern dazu verwendet, um Eingaben sowohl abzugrenzen als auch semantisch zu markieren.⁷ Erst einmal wird alles ignoriert, was nicht eingeklammert ist. Kommentare oder eine der Verständlichkeit für Menschen dienende Interpunktion lassen sich bedenkenlos zwischen die eingeklammerten Wörter schreiben und bedürfen keiner weiteren Kennzeichnung.

Lysa kennt vier verschiedene Klammernpaare für Substantive, Pronomen, Anführungen und Verben. Die Terminologie des originalen Forth wird also insofern beibehalten, daß eine Eingabe aus Wörtern besteht. Durch die Klammern ist für den Interpreter aber erkennbar, um welche Wortart es sich jeweils handelt:

⁷*Lysa* ähnelt also **Lisp**, aber nur im Sinne von *Lots of Irritating Superfluous Parentheses*.

- Als Substantive können entweder Zahlen oder Zeichenketten angegeben werden, sie stehen also unmittelbar für Gegebenheiten.
- Pronomen sind Platzhalter für Gegebenheiten.
- Anführungen⁸ sind Eingaben, die nicht direkt abgearbeitet werden. Es wird damit also eine gewisse Uneigentlichkeit markiert, die allerdings aufgelöst werden kann.⁹ Anführungen können beliebige Gegebenheiten enthalten (auch weitere Anführungen, welche als Ganzes also auch Gegebenheiten darstellen).
- Verben sind die Wörter, bei denen etwas geschieht. Meist verbrauchen sie Gegebenheiten und/oder schaffen welche.

Von den Klammern gibt es mehrere Sätze, Dialekte genannt (S. 33). Jederzeit kann innerhalb eines Skriptes bestimmt werden, welcher Dialekt ab da erkannt wird. Eigene Dialekte lassen sich hinzufügen.

Grundsätzlich arbeitet *Lysa* mit Unicode-Eingaben (UTF-8). Ob ein Symbol in der Schrift **STIX Two Math** vorhanden ist, entscheidet darüber, ob es verwendet werden kann oder nicht.

Gegebenheiten sind das, was auf den Stapel gelegt oder in einem Regal deponiert wird (wenn auch nur mittelbar). Vermutlich erscheint einem der Begriff erst einmal alles andere als naheliegend,

⁸*PostScript* kennt eine ähnliche Konstruktion mit geschweiften Klammern. Ebenso **Factor** von Slava Pestov, wo sie interessanterweise *quotations* heißen, wovon ich aber erst später etwas erfahren habe.

⁹Wenn man in der gewöhnlichen Weise Worte gebraucht, so ist das, wovon man sprechen will, deren Bedeutung. Es kann aber auch vorkommen, dass man von den Worten selbst oder von ihrem Sinne reden will. Jenes geschieht z. B., wenn man die Worte eines Andern in gerader Rede anführt. Die eigenen Worte deuten dann zunächst die Worte des Andern und erst diese haben die gewöhnliche Bedeutung. Wir haben dann Zeichen von Zeichen. In der Schrift schließt man in diesem Falle die Wortbilder in Anführungszeichen ein. Es darf also ein in Anführungszeichen stehendes Wortbild nicht in der gewöhnlichen Bedeutung genommen werden.« Gottlob Frege ÜBER SINN UND BEDEUTUNG. In: ZEITSCHRIFT FÜR PHILOSOPHIE UND PHILOSOPHISCHE KRITIK, N. F., Bd. 100/1 (1892), S. 25-50

doch möglicherweise wird er verständlicher, wenn ich darauf hinweise, daß Datum auf das lateinische Wort für Gegebenes zurückgeht (und Daten ist die Mehrzahl von Datum; erst in neuerer Zeit hat sich die Bedeutung von Datum in der deutschen Sprache auf das Kalenderdatum verengt).

Winkelklammern sind der voreingestellte Dialekt. Darin sind auch die Beispiele in diesem Handbuch gehalten. Die Verwendung ansonsten unüblicher Klammern erklärt sich vor allem dadurch, daß diese nicht in den Wörtern vorkommen dürfen (es gibt keine Möglichkeit, sie zu maskieren). Aus diesem Grunde verwendete bereits *proly-sarik* die Winkelklammern, um Angaben voneinander abzugrenzen. Aber es gibt genug Gelegenheiten, bei denen die Standardklammern oder die Guillemets ausreichen, weil in den Wörtern weder Klammern noch Anführungszeichen vorkommen.

Wie das originale Forth arbeitet auch *Lysa* mit **Stapelspeichern**. Während beim originalen Forth allerdings Daten direkt auf dem Stapel abgelegt werden, landen bei *Lysa* dort grundsätzlich nur Verweise auf die eigentlichen Daten. Das hat den Vorteil, daß alle Gegebenheiten denselben Platz auf dem Stapel beanspruchen (den des Verweises), gleichgültig, wie umfangreich ihr Inhalt sein mag. Die einzelnen Gegebenheiten hingegen liegen im Hauptspeicher (Heap).

Der Stapel arbeitet nach dem **LIFO-Prinzip**: Die zuletzt abgelegte Gegebenheit wird als erste wieder abgerufen. Die Kapazität aller Stapel, Regale und Lexika nebst den Gegebenheiten in *Lysa* wird ausschließlich durch den zur Verfügung stehenden Hauptspeicher begrenzt.

Neben dem Stapel verfügt jeder Interpreter über ein Regal mit 26 frei nutzbaren und zudem für einige Gegebenheitsarten speziellen Fächern. Letztere dienen insbesondere dazu, an den üblicherweise immutablen Gegebenheiten Änderungen vorzunehmen. Und schließlich sind da noch die Lexika für die Pronomen.

An oberster Stelle steht ein globales Lexikon, auf das von überall zugegriffen werden kann. Jeder Interpreter verfügt dazu noch über ein eigenes, lokales Lexikon. Einige interne Vorgänge werden mit

je eigenen Interpretern ausgeführt, die aber dann keinen Zugriff auf die Stapel, Register und das lokale Lexikon des aufrufenden Interpreters haben. Dafür verfügen sie über eigene Speicher. Nur das globale Lexikon steht wie gesagt überall zur Verfügung. Ebenso überall verfügbar sind die Verben, die in einer eigenen Speicherstruktur global verwaltet werden.

Substantive, Verben und Anführungen werden direkt beim Einlesen interpretiert und liegen ab da in maschinensprachlicher Form vor. Pronomen hingegen werden erst bei ihrer Effektivierung durch die Gegebenheit ersetzt, für die sie einen Platzhalter bilden. Dann wird zunächst im lokalen Lexikon nachgeschaut, ob zu dem Pronomen eine Gegebenheit gespeichert ist. Ist dies der Fall, wird das Pronomen durch diese Gegebenheit ersetzt und die Sache ist erledigt. Findet sich im lokalen Lexikon hingegen keine Gegebenheit, wird die Suche im globalen Lexikon fortgesetzt. Jedes lokale Lexikon hat also den Vorrang gegenüber dem globalen.

Direkt als Substantive eingegeben werden können Zahlen und Zeichenketten. Wenn bei einer Zahl ein Nachkommateil angegeben ist, wird sie als Gleitkommazahl gespeichert, ohne als Ganzzahl (jeweils 64-Bit). Was nicht als Zahl interpretiert werden kann, ist eine Zeichenkette. Ab einer Länge von 18 Zeichen wird allerdings gar nicht erst versucht, etwas als Zahl zu interpretieren, das ist dann von vornherein eine Zeichenkette.

Durch die Angabe eines Anführungszeichens (u0022) an erster Stelle eines Substantivs kann zusätzlich erzwungen werden, daß ein Substantiv als Zeichenkette genommen wird. Das Anführungszeichen zählt dann allerdings nicht mit (für den Fall, daß in der Zeichenkette ebenfalls ein Anführungszeichen an erster Stelle steht, sind deren zwei anzugeben).

Durch die Angabe eines Semikolons an erster Stelle eines Substantivs wird bei einer Gesamtlänge von 17 Bytes die Angabe als hexadezimale Repräsentation des Speicherinhalts einer Gleitkommazahl (64 Bit) interpretiert (**IEEE 754**). Aber diese Möglichkeit ist vor allem für interne Zwecke vorgesehen, um Zahlen verlustfrei einzulesen.

Neben Zahlen und Zeichenketten gibt es noch weitere Arten von Gegebenheiten. Doch diese lassen sich nicht mehr direkt eingeben, sondern werden durch Verben geschaffen. In der Regel – und wann immer praktikabel – sind Gegebenheiten immutabel. Soll eine Zahl verändert werden, wird nicht die sie repräsentierende Gegebenheit verändert, sondern diese wird durch eine neue Gegebenheit ersetzt. Nur bei Gegebenheiten, die zu komplex und/oder umfangreich sind, um sie jedes Mal zu ersetzen, finden Veränderungen in der Gegebenheit statt. Es ist ineffektiv, bei Änderung eines einzelnen Bildpunktes gleich ein neues Bild zu erschaffen, vor allem wenn dies häufiger vorkommt. Hauptsächlich für diese Fälle sind die Register gedacht (das sind die speziellen Fächer in einem Regal).

Pronomen und Verben werden durch Zeichenketten identifiziert, die homonym sein können, da durch die Klammern zu erkennen ist, was jeweils gemeint ist. Verben haben zudem Synonyme. Zum Beispiel ist zum Addieren außer (+) auch (add) oder (addiere) möglich. Exemplarisch zeigt dies zudem die drei Verbarten, die dabei vorkommen: symbolische, mnemonische und ausgeschriebene.

Eine Handlung (das sind die mit den Verben verbundenen Anweisungen für den Computer), kann mit einem oder auch mehreren Verben verbunden sein. Ein symbolisches Verb ist in der Regel kurz und besteht oft aus besonderen Zeichen; ein mnemonisches Verb besteht zumeist aus einer Abkürzung und wenigen Buchstaben; bei ausgeschriebenen Verben handelt es sich um ein oder auch mehrere Wörter. Die mnemonischen Verben bilden eine Zwischenform zwischen symbolischer Darstellung und ausgeschriebenen Wörtern. Ursprünglich stammen sie aus den **Assemblersprachen**. An dieser Stelle habe ich auch fremdsprachliche Bezeichnungen berücksichtigt, insbesondere aus Englisch und **Esperanto**¹⁰.

¹⁰Ich hatte einmal die Idee, statt in Englisch oder in der eigenen Sprache in Esperanto zu **programmieren** (Bezeichnungen und Kommentare), sozusagen um gleiche Bedingungen sowohl für den Schreibenden als auch internationale Leser zu schaffen, allerdings kann ich viel zu schlecht Esperanto (so gut wie gar nicht), um diesen Plan konsequent umzusetzen; immerhin habe ich mir zu dem Zweck ein Wörterbuch besorgt, das ich nun wieder verwenden kann: [14]

Zusätzliche Verben, die während des Programmlaufs definiert werden, haben zunächst nur eine Bezeichnung; allerdings können auch dazu Synonyme gebildet werden.

Pronomen werden gebildet, indem eine Zeichenkette mit einer Gegebenheit in Verbindung gebracht wird (durch ein spezielles Verb); zusätzliche Verben werden gebildet, indem eine Zeichenkette in Verbindung mit einer Anführung gebracht wird (durch ein anderes spezielles Verb). Letzteres entspricht mehr oder minder dem Compilermodus des originalen Forth. Soweit würde ich bei *Lysa* nicht gehen wollen, auch wenn die neu geschaffenen Verben (bis auf die Pronomen) in maschinensprachlicher Form vorliegen. Aber diese Stellung zwischen interpretierter und kompilierter Programmiersprache bewirkte von jeher, daß Forth für seine hohe Ausführungsgeschwindigkeit berühmt ist.

Anführungen können sowohl Pronomen als auch Verben bilden. Durch andere Verben können Anführungen ausgeführt werden, das heißt, sie verlieren ihre Uneigentlichkeit und der Inhalt wird effektuiert. Insofern können Anführungen sowohl als Verben als auch als Pronomen ausgeführt werden.

Nicht nur durch die Klammern, sondern insgesamt erinnert eine Anführung an einen **Anweisungsblock**, wie er in zahlreichen Programmiersprachen vorkommt. Um diese Ähnlichkeit weiterzutreiben, verfügt jede Anführung in *Lysa* über ein eigenes Regal (eine Kopie des Regals der übergeordneten Ebene) und auch ein eigenes lokales Lexikon. Letzteres wird jeweils an die Kette angegliedert, welche durch globales und lokales Lexikon des Interpreters begonnen wurde. Bei der Suche nach dem Ersatz für ein Pronomen wird dann entsprechend verfahren.

Innerhalb eines Lexikons kann es nicht mehrere Verben oder Pronomen unter derselben Zeichenkette geben. Erfolgt eine Verknüpfung mit einer bereits verwendeten Zeichenkette, wird die bestehende ersetzt.

Lysa gibt einige Meldungen auf der Konsole aus. Hauptsächlich schreibt es seine Ausgaben jedoch in eine HTML-Datei, die schließ-

lich mit jedem Browser betrachtet werden kann. Das ermöglicht die Ausgabe auch komplexerer Informationen.

Das sollte als Überblick genügen. Alles weitere erschließt sich bei der Erläuterung der einzelnen Verben.

Versionshistorie

Begonnen	Sprache	Kodezeilen	Besonderheit
13.09.2019	Java	14.369	Letzte Version, <i>Corregio</i> , der Vorläufersoftware proly sarik zur Erstellung von Videos; letzte Fassung vom 5. August 2020
10.10.2020	Java	10.252	Verben und Gegebenheitsarten zur Laufzeit als Services einbinden
14.01.2021	Go	4.277	Erste Version in Go; Umstieg von Java, da in Go PNGs mit 16 Bit pro Kanal gespeichert werden können
17.02.2021	Go	10.051	Farbe als lineares RGB gespeichert
28.04.2021	Go	13.519	Einführung der Maske zur Separierung eines Farbkanals; erste Version mit einer Art Regal, je ein Fach für jede Gegebenheitsart
26.09.2021	Go	15.508	Allgemeines Funktionen jenseits des aktuellen Bedarfs, z. B. für Dateizugriff
07.01.2022	Go	14.083	Konzentration auf den aktuellen Bedarf (<i>»Don't speculate!«</i>); Gegebenheiten für Licht und Remissionen, Farben als CIE-XYZ gespeichert
30.03.2022	Go	19.332	Interaktive Ausführung über einen Port
01.09.2022	C	743	Parallel zur Version in Go begonnen, um mit Konzepten zu experimentieren
11.09.2022	Go	11.816	Aufspaltung der Verben in Stapel- und Modifikatorverben; Regal mit 26 Fächern; Einführung des Wide Gamut-Farbraums gemäß BT.2100-2 und Speicherung als nicht wieder einlesbare PNGs
03.02.2023	C	3.876	Ernsthafter Versuch einer Version des Interpreters in C, um bessere Bibliotheken zum Laden und Speichern von PNGs einbinden zu können
10.03.2023	C	5.160	Aufbau einer Speicherverwaltung, die sich allerdings für nebenläufige Verarbeitung als ungeeignet erwiesen hat
04.04.2023	C	5.992	Gescheiterter Versuch, die Speicherverwaltung für Multithreading zu härten
25.04.2023	Go	12.996	Erster ΣΙΣΥΡΗΟΣ-Anlauf; Regal mit 26 Fächern
03.08.2023	Go	15.510	Aktuelle Version

Mit dieser Version wird erstmals das erreicht, was ich *konzeptionelle Produktionsreife* nennen möchte. Damit endet die erste Experimentierphase zur Entwicklung der Sprache als solcher. In einer neu

begonnenen Version *Sysiphos*₃ wird das bisher Erreichte noch einmal poliert, geglättet und für ein erstes Video genutzt. Damit endet nicht das Programmieren an *Lysa*, doch wichtiger wird das Programmieren in *Lysa*.

Dies ist die letzte Version eines öffentlich zugänglichen Handbuchs von *Lysa*. Der jeweilige Stand wird zukünftig nur noch in Zusammenhang mit erschienenen Videos veröffentlicht.

Programmaufruf

goLysa ist für Linux auf einer AMD64-Architektur kompiliert. Die Programmdatei des Interpreters wird im Verzeichnis `/usr/bin` installiert und steht somit systemweit zur Verfügung. Aufgerufen wird der Interpreter von einer Kommandozeile aus idealerweise in dem Verzeichnis, in dem sich das zu verarbeitende Skript befindet:

```
gL skriptname
```

Das Skript muß entweder auf `.lysa` oder auf `lysa.gz` (komprimiert) enden. Fehlt eine Dateiendung wird `.lysa` automatisch angehängt.

Üblicherweise wird die HTML-Ausgabe in die Datei `/tmp/Lysa.html` geschrieben. Um dafür einen anderen Dateinamen zu verwenden, ist dieser inklusive Pfad vor den Namen der Skriptdatei zu setzen:

```
gL ausgabedatei skriptname
```

Fehlt bei der Angabe der Ausgabedatei die Endung `.html`, wird diese automatisch ergänzt.

Im HTML-Protokoll werden die Eingaben auf der obersten Ebene aufgeführt. Um das zu unterbinden, ist vor der Angabe der Ausgabedatei und des Skriptes ein `-s` (stumm) hinzuzufügen. Um andererseits auch den Inhalt von aufgerufenen Handlungsbeschreibungen protokollieren zu lassen, ist entsprechend ein `-a` (ausführlich) hinzuzufügen.

Insofern nichts anderes angegeben wurde, wird bei weniger als vier nebenläufig ausführbaren Routinen die Nebenläufigkeit ausgeschaltet. Ab vier parallel ausführbaren Prozessen wird die maximal zur Verfügung stehende Anzahl minus Eins dafür vorgesehen. Durch die Angabe `-rn` vor der Ausgabedatei kann allerdings nutzerseitig eine Anzahl von Routinen festgelegt werden (n darf dabei allerdings nicht größer sein als die maximal zur Verfügung stehende Anzahl).

Für eine profilgeleitete Optimierung (PGO) kann der Interpreter angewiesen werden, ein CPU-Profil des Programmlaufs zu schreiben (`goLysa.pprof`, im Verzeichnis, aus dem heraus der Interpreter aufgerufen wurde). Dazu ist vor der Angabe der Ausgabedatei ein `-p` hinzuzufügen.

Fremdpakete

Neben einigen Standardbibliotheken von Go (1.21.5) kommen in *goLysa* noch folgende Fremdbibliotheken zum Einsatz:

<i>Autor</i>	<i>Adresse</i>	<i>Version</i>	<i>Lizenz</i>
Klaus Post	github.com/klauspost/cpuid/v2	2.2.6	MIT License
Nigel Tao	golang.org/x/image/vector	0.14.0	BSD-3-Clause
	↳ golang.org/x/image	0.14.0	BSD-3-Clause
	↳ golang.org/x/sys	0.15.0	BSD-3-Clause
Go Autoren	golang.org/x/text/collate	0.14.0	BSD-3-Clause

Abschweifungen

Anhand von Stichwörtern schiebe ich Informationen zu Hintergründen an mehr oder minder passender Stelle in das Handbuch ein. Um sie einfach finden zu können, sind die Stichwörter an dieser Stelle verzeichnet:

Absatz	200	Glatter Satz	183
Ausschluß	193	hängende Interpunktion	184
Duchschuß	203	Regal	40
Farbraumkonvertierung	72	Schreibfedern	176
Fliegenkopf	180	Transformationsmatrix	74
Geviert	180		

goLysa (Sisyphos₂)

mit 40 Gegebenheitsarten, 230 Handlungen, 6 Protokollen und 5 Effekten

© 2023 bei Marcus Daniel Cremer, Gelsenkirchen

Gestartet am Sonntag, 31. Dezember 2023 (16:44:15 CET) auf einem Intel(R) N95

Verarbeitung des Skriptes hallowelt.lysa: 800 • 150 • **Leinwand erschaffen** • **registrieren** • 95,147153658 • 75,874066799 • 47,573576829 • ‚sRGB‘ • **umrechnen** • **grundieren** • **Bild verwenden** • **ausmessen** • ‚diagonal fallend‘ • *Linear* (Verlauf) • **Maske aus Verlauf generieren** • **registrieren** • **invertieren** • **Maske entnehmen** • ‚Alpha‘ • **maskieren** • **Bild entnehmen** • **duplizieren** • **ausmessen** • **Leinwand erschaffen** • **registrieren** • 90,152993259 • 32,77680183 • 9,637784404 • ‚sRGB‘ • **umrechnen** • **grundieren** • 0 • 0 • **Dupel generieren** • 0 • **plazieren** • ‚usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz‘ • **einbeziehen** • *brandvogel-r-400* (Schrift) • **registrieren** • ‚Alpha‘ • **entlarven** • **Zeichnung generieren** • **registrieren** • 150 • 0,862 • **dividieren** • **duplizieren** • **anpassen** • 0,2 • 0,8 • **versetzen** • ‚H‘ • **greifen** • **setzen** • ‚a‘ • **greifen** • **unterschneiden** • **setzen** • ‚l‘ • **greifen** • **unterschneiden** • **setzen** • ‚l‘ • **greifen** • **unterschneiden** • **setzen** • ‚o‘ • **greifen** • **unterschneiden** • **setzen** • 0,2 • 0 • **versetzen** • ‚W‘ • **greifen** • **setzen** • ‚e‘ • **greifen** • **unterschneiden** • **setzen** • ‚l‘ • **greifen** • **unterschneiden** • **setzen** • ‚t‘ • **greifen** • **unterschneiden** • **setzen** • **Zeichnung entnehmen** • **verwirklichen** • ‚Alpha‘ • **maskieren** • **Bild entnehmen** • **zeigen**

Bild: 800×150 Bildpunkte

Hallo Welt

Verarbeitet: {1502653} {3} {10536} { 401685 }

31. Dezember, 16:44:16 • Laufzeit: 0,576 Sekunden (CPU: 0,806 Sekunden), Speicher: 83328456 Bytes in 1782429 Objekten

⟨800⟩ ⟨150⟩ (cvs) (*). ⟨95,147153658⟩ ⟨64,296936811⟩ ⟨18,996655704⟩ ⟨sRGB⟩ (clr) (Bd.pln). (:Bd) (mez) ⟨diagonal fallend⟩ ⟨Linear⟩ (vlm) (*): (Mk.inv). (.Mk) ⟨Alpha⟩ (Bd.msk). (.Bd) (↔) (mez) (cvs) (*); ⟨90,152993259⟩ ⟨32,77680183⟩ ⟨9,637784404⟩ ⟨sRGB⟩ (clr) (Bd.pln); ⟨0⟩ ⟨0⟩ (D) ⟨0⟩ (Bd.plc).

⟨usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz⟩ (⊞). ⟨brandvogel-r-400⟩ (*). ⟨Alpha⟩ (Bd.sms) (dsg) (*). ⟨150⟩ ⟨0,862⟩ (÷) (↔) (g). ⟨0,2⟩ ⟨0,8⟩ (v). ⟨H⟩ (Sf.ltr) (Zn.kmp). ⟨a⟩ (Sf.ltr) (⊞) (Zn.kmp). ⟨l⟩ (Sf.ltr) (⊞) (Zn.kmp). ⟨l⟩ (Sf.ltr) (⊞) (Zn.kmp). ⟨o⟩ (Sf.ltr) (⊞) (Zn.kmp). ⟨0,2⟩ ⟨0⟩ (v). ⟨W⟩ (Sf.ltr) (Zn.kmp). ⟨e⟩ (Sf.ltr) (⊞) (Zn.kmp). ⟨l⟩ (Sf.ltr) (⊞) (Zn.kmp). ⟨t⟩ (Sf.ltr) (⊞) (Zn.kmp). (.Zn) (efv) ⟨Alpha⟩ (Bd.msk). (.Bd) (⌘)

Vokabular

Verzeichnis aller Verben

!	48	↗	(u219b)	35
*a-z	39	↘	(u219d)	90
+	58	↙	(u21a3)	37
++	56	↘	(u21a6)	35
--	56	↗	(u21b7)	36
.	143	⊂	(u21bb)	47
.a-z, Xx	39	⇌	(u21c4)	37
:a-z, Xx	39	↑↓	(u21c5)	32
<	51	←	(u21d0)	40
>	51	⇒	(u21d2)	46
?	43	↔	(u21dd)	91
??	44	↖	(u21e4)	199
⊃	(u00ac) 51	→	(u21e5)	200
®	(u00ae) 41	∃	(u2203)	81
²	(u00b2) 62	∈	(u2208)	79
³	(u00b3) 60	∉	(u2209)	79
ℳ	(u00b6) 200	∋	(u220b)	80
×	(u00d7) 60	∄	(u220c)	79
÷	(u00f7) 59	−	(u2212)	62
ΔE	(u0394) 129	√	(u221a)	64
—	(u2014) 99	³√	(u221b)	63
Nº	(u2116) 55	∧	(u2227)	51
Q	(u211a) 62	∨	(u2228)	50
←	(u2190) 42	~	(u223f)	102
↑	(u2191) 31	≅	(u224a)	52
→	(u2192) 41	⊗	(u2297)	73

☉	(u2299)	74	add	58
≡	(u22a8)	50	addiere	58
≠	(u22ad)	52	addiere.Vektoren	65
·	(u22c5)	66	adjungiere	50
★	(u22c6)	61	ändere.auf.Ausgangszeile	199
[(u2308)	56	akk	113
]	(u230a)	55	alf	121
⊕	(u2316)	171	anm	206
└	(u2319)	59	annuliere	171
⌨	(u2328)	190	apt	49
⌚	(u2339)	76	aut	51
⊕	(u236c)	53	avc	140
⊠	(u2384)	187	avv	156
⚡	(u2386)	31	Bd.lin	145
∖	(u2387)	45	Bd.msk	163
⌚	(u238b)	46	Bd.plc	142
⊗	(u238c)	174	Bd.pln	141
↩	(u23ce)	46	Bd.pxl	141
}	(u23dc)	196	Bd.rkt	141
}	(u23de)	195	Bd.rtm	143
)	(u23e0)	195	Bd.sms	147
⌚	(u23f2)	93	bebildere	206
▶	(u23f5)	206	befürworte	43
■	(u23f9)	207	beginne.Zeile	182
●	(u23fa)	207	bejahe	50
⚡	(u23ff)	37	beleuchte	123
└	(u2423)	193	berechne.Cosinus	58
☀	(u2600)	123	berechne.Sinus	58
☎	(u260e)	207	berechne.Skalarprodukt	66
☝	(u261d)	53	berechne.Tangens	59
⚙	(u263c)	120	berechne.Vektorprodukt	71
♠	(u2660)	80	bestimme.Bildfrequenz	93
♣	(u2663)	81	beziehe.ein	31
♪	(u266a)	116	beziffere	53
☐ ☐	(u2682 / 3)	63	bilde.aus	138
☐	(u2685)	63	bilde.Dyade	73
☉	(u27f2)	46	bilde.ein	140
→	(u2905)	36	binde.ein	207
→	(u2920)	36	bld	133
↪	(u2934)	87	brt	41
×	(u2a2f)	68	bzr	88
A		173	cbr	63
abl		43	cct	54

cfr	53	erfrage	90
clb	207	ermittle.Bildkoordinaten	171
clp	142	ermittle.Durchschnittsfarbe	140
clr	134	ermittle.Durchschnittswert	156
cmd	48	ermittle.Farbabstand	129
cnm	207	errechne.Kreuzprodukt	68
col	80	erschaffe.Kolumne	202
cos	58	erschaffe.Leinwand	141
cpy	36	erschaffe.Maske	156
cvs	141	erschaffe.Raumklang	113
D	66	erschaffe.Regal	41
d	171	erschaffe.Sammlung	78
dev	91	erschaffe.Schrift	180
dfn	43	erschaffe.Stille	99
dgt	53	erschaffe.Zeile	199
dif	129	erweitere.Entwicklung	90
dig	66	erweitere.Schriftprobe	181
digitize	53	etd	90
div	59	Ew.stt	90
dividiere	59	ewo	45
dlg	60	exe	46
dlk	33	exportiere.Audio	113
dlt	44	exportiere.Klang	99
dot	40	exportiere.Maske	158
dotiere	40	F	130
drehe	171	fbp	125
drp	35	fil	108
dsg	171	filtere	108
dup	35	fls	52
dupliziere	35	fps	93
dya	73	frage.ab	110
dyl	120	frq	100
eat	166	ftc	36
edu	138	füge.Doppelspatium.ein	195
efi	204	füge.Haarspatium.ein	195
efv	176	füge.hinzu	79
elv	123	füge.Spatium.ein	196
end	95	fülle	158
entfalte	123	fülle.Rechteck	141
entferne	35	G	174
entlarve	147	g	172
entscheide	45	gänze	55
eql	52	generiere.Animation	207

generiere.Ausschluß	193	hdm	74
generiere.Dupel	66	hebe	160
generiere.Effekt	204	hebe.auf	43
generiere.Entwicklung	91	icp	207
generiere.Farbe	130	ill	120
generiere.Histogramm	166	ilu	123
generiere.Klammerdialekt	33	img	140
generiere.Klangfilter	108	importiere.Audio	113
generiere.Licht	120	importiere.Klang	100
generiere.Quadrupel	70	ins	31
generiere.Quadrupelkarree	77	installiere	31
generiere.Remission	123	int	55
generiere.Schriftzeichen	184	intensiviere	160
generiere.Tageslicht	120	interpoliere.Frequenzen	100
generiere.Taktgeber	96	interpoliere.kubisch	88
generiere.Tripel	69	invertiere	161
generiere.Tripelkarree	73	isoliere	131
generiere.Untertitel	95	iteriere	46
generiere.Verlauf	87	itg	166
generiere.Verlaufsmaske	159	itr	46
generiere.Zeichnung	171	ivs	76
generiere.Zeitraum	93	jmp	46
gib.ab	79	K	174
gib.Anteil	166	kaj	50
gib.Einzelanteil	166	kbz	60
gib.Einzelwert	87	kehre.Richtung.um	101
gib.Ende	195	kehre.um	59
gib.heraus	79	Kl.amp	104
gib.Intensität	166	Kl.cut	103
gla	110	Kl.mdl	101
glätte	110	Kl.mix	103
gleiche.an	168	Kl.mmf	108
gleiche.aus	190	Kl.osc	102
gleiche.Zeilenanfang.aus	199	Kl.pkn	102
gleiche.Zeilenende.aus	200	Kl.sld	100
glitsche	100	Kl.wrp	111
grd	166	klk	78
greife	179	knl	131
greife.zu	36	kombiniere	161
grt	51	komplementiere	124
grundiere	141	konjungiere	51
Gz.dec	56	kopiere	36
Gz.inc	56	korrigiere	184

Kp.pdm	110	moduliere	101
kpm	124	mov	41
krn	184	mtn	96
kubiziere	60	mul	60
lda	113	multipliziere	60
lds	100	multipliziere.elementweise	74
lies.Bildpunkt.aus	141	multipliziere.Matrizen	74
lies.Maskenpunkt.aus	162	multipliziere.mit.Skalar	66
Ln.agz	199	neg	51
Ln.ara	199	negiere	51
Ln.brs	196	nimm.größeren	61
Ln.cop	195	nimm.kleineren	61
Ln.era	200	nimm.Teilungsrest	61
Ln.led	195	nimm.zurück	172
Ln.spc	193	noi	103
lösche	44	normalisiere.Klang	102
löse.auf	66	nrs	46
log	60	nvt	63
logarithmiere.dekadisch	60	O	173
logarithmiere.natürlich	60	oag	47
lrv	156	opa	134
maskiere	163	oszilliere	102
max	61	ovr	37
mex	158	p	163
mez	142	passe.an	172
min	61	pks	94
mische	80	plaziere	142
mische.an	125	positioniere	114
mische.Farben	133	potenziere	61
miß.aus	142	pow	61
Mk.grz	162	prb	41
Mk.inv	161	prd	93
Mk.its	160	prf	110
Mk.kmb	161	prg	87
Mk.lev	160	prn	44
Mk.niv	168	probiere	41
Mk.plb	158	profiliere	110
Mk.pnk	163	pronominalisiere	44
Mk.rla	163	prt	166
mmp	74	prüfe.auf.Minderung	51
mnt	37	prüfe.auf.Steigerung	51
mod	61	prv	50
modifiziere.Opazität	134		

punktiere	163	sin	58
pzt	120	skaliere	143
qdr	62	skp	66
qpk	77	Sl.ald	79
quadriere	62	Sl.eld	79
r	172	Sl.fdn	79
rausche	103	Sl.occ	81
rechne.um	134	Sl.ord	81
registriere	41	Sl.sfl	80
reguliere	114	smp	66
rev	101	snn	48
richte.aus	115	sondiere	81
Rk.adg	116	sortiere	81
Rk.ext	117	spiele	116
Rk.ldn	114	spr	56
Rk.ori	115	sqr	64
Rk.pan	114	stelle.dar	120
rms	123	stk	95
rnd	63	stv	42
rotiere	36	sub	62
rtt	36	subtrahiere	62
rufe.zurück	207	sva	113
runde.ab	55	svs	99
runde.auf	56	swp	37
sammle	190	sxg	31
sammle.ein	80	sxn	32
sbr	55	synonymisiere	48
schere.horizontal	172	taktiere	97
schere.vertikal	173	tan	59
schlage.an	189	tausche.aus	32
schließe.ab	173	Tg.dir	97
schneide.aus	103	tpk	73
schneide.heraus	142	tpn	75
scl	143	tpr	180
setze	187	transkribiere	95
setze.ab	200	transponiere	75
setze.an	173	trs	91
setze.Bildpunkt	143	Tx.aln	200
setze.zurück	174	Tx.lgn	199
Sf.gly	184	txt	202
Sf.ltr	179	überlagere	103
Sf.scm	181	überspringe	37
sil	99	übertrage	91

umhülle	111	xyz	130
umk	59		
unm	67	Z	199
unterschneide	187	zähle.rauf	56
v	175	zähle.runter	56
vad	65	zdp	66
val	62	zeige	37
verbalisiere	48	Zh.agl	190
verbinde.gerade	174	Zh.asg	189
verbinde.krumm	174	Zh.lst	195
verbinde.würflig	175	Zh.row	190
vereigentliche	46	ziehe.Breitfeder	177
vereigentliche.bedingt	46	ziehe.Kubikwurzel	63
vereinheitliche	67	ziehe.Linie	145
vergleiche	52	ziehe.Quadratwurzel	64
verkehre	76	ziehe.Spitzfeder	178
verknüpfe	54	Zn.adj	187
verlängere	117	Zn.cbc	175
verlasse	46	Zn.ccl	171
verlerne	49	Zn.fin	173
verneine	52	Zn.htv	172
verschiebe	94	Zn.kie	171
versetze	175	Zn.kmp	187
verstärke	104	Zn.krb	174
verstaue	42	Zn.ora	182
vertausche	37	Zn.pos	173
verwerte	62	Zn.rek	174
verwirkliche	176	Zn.rpr	174
vignettiere	163	Zn.skl	172
Vl.vlr	87	Zn.stb	172
vlm	159	Zn.trf	175
vpd	71	Zn.trn	171
W	175	Zn.trt	177
wandle	121	Zn.ttt	178
wiederhole	47	Zn.vtv	173
würfle	63	zqp	70
würfle.normalverteilt	63	ztp	69

Lysa

Eine Instanz des Interpreters enthält sowohl den zentralen Stapelspeicher als auch ein eigenes Lexikon und ein Regal. Sie stellt jedoch keine Gegebenheitsart dar.

↑	(u2191)	31	installiere	31
↑↓	(u21c5)	32		
⇔	(u2386)	31	sxg	31
beziehe.ein		31	sxn	32
ins		31	tausche.aus	32



(Strg+⇧+u2386)
(Esperanto)

sxargi

beziehe.ein

$\langle \text{Dateiname}_{zk} \rangle (\Leftrightarrow \mid \text{sxg} \mid \text{beziehe.ein}): ?$

Liest ein Skript ein, als sei es ein Teil der aktuell abzuarbeitenden Eingabe (allerdings ohne Ausgaben in die HTML-Datei). Es kann sich bei der importierten Datei auch um ein komprimiertes Skript handeln (Endung: .lysa.gz).

Falls der Dateiname nicht bereits auf .lysa oder .lysa.gz endet, wird die Endung .lysa angefügt. Danach wird versucht, die durch die Zeichenkette bestimmte Datei zu öffnen und einzulesen. Es wird derselbe Interpreter verwendet. Gültig ist der installierte Klammerdialekt.



(Strg+⇧+u2191)
(Englisch)

install

installiere

$\langle \text{Gegebenheit}_{Kd \vee Rg} \rangle (\uparrow \mid \text{ins} \mid \text{installiere}): \{ \langle \text{Installation}_{Kd \vee Rg} \rangle \}$

Installiert einen Klammerdialekt oder ein Regal im Interpreter.

↑↓

sxangxi

tausch.e.a.us

(Strg+↑+u21c5)

(Englisch)

$\{\langle Installation_{Kd\vee Rg}^A \rangle\} \langle Gegebenheit_{Kd\vee Rg}^B \rangle (\updownarrow | \text{sxn} | \text{tausch.e.a.us}):$
 $\langle Gegebenheit_{Kd\vee Rg}^A \rangle \{\langle Installation_{Kd\vee Rg}^B \rangle\}$

Tauscht den bisherigen Klammerdialekt oder das bisherige Regal gegen eine entsprechende Gegebenheit vom Stapel aus.

Verarbeitung des Skriptes <i>klammerdialekt.lysa</i> : Standard (Klammerdialekt) • zeigen		
Klammerdialekt:		
Substantiv:	()
	u0028	u0029
Pronomen:	'	'
	u0027	u0027
Anführung:	{	}
	u007b	u007d
Verb:	[]
	u005b	u005d

Enthält die Klammern zur Kennzeichnung der Eingaben. Der installierte Klammerdialekt ist gültig. Voreingestellt sind Winkelklammern.

↔ nein • Q ungültig • Z ungültig • ∞ ungültig

dlk 33 generiere.Klammerdialekt 33

Folgende Standarddialekte sind als Pronomen abrufbar:

Dialekt	Substantive	Pronomen	Anführungen	Verben
Winkel	⟨ ⟩ 27e8 27e9	⟨ ⟩ 2991 2992	« » 27ea 27eb	{ } 2772 2773
Standard	()	' '	{ }	[]
Guillemets	‹ › 2039 203a	, , 201a 201b	„ “ 201e 201c	« » 00ab 00bb
Ecken	┌ ┐ 231e 231f	┌ ┐ 231c 231d	[] 2308 2309	┌ ┐ 230a 230b

dialekto (Esperanto)
generiere.Klammerdialekt

⟨Klammern_{zk}⟩ (dlk | generiere.Klammerdialekt): ⟨Dialekt_{kd}⟩

Erzeugt einen Klammerdialekt und legt ihn auf den Stapel. Die Zeichenkette mit den Klammern muß exakt acht Zeichen lang sein

und vier ineinander geschachtelte Klammerpaare enthalten (von außen nach innen: Anführungen, Pronomen, Substantive, Verben [also: «‹‹()››»]).

Der Dialekt kann direkt verwendet werden, indem er installiert wird (S. 31).

Die Klammern können weitgehend frei gewählt werden. Bis auf die Klammern für Anführungen, ist es zulässig, für beide Seiten eines Klammerpaares dasselbe Zeichen zu verwenden. Anführungen können verschachtelt werden, deshalb muß eindeutig sein, ob eine neue Anführung begonnen wird oder eine bereits begonnene abgeschlossen.

Die Verarbeitung eines Skriptes erfolgt mit einem **endlichen Zustandsautomaten**, den das Auftauchen einer im aktuell gültigen Dialekt vorhandenen Klammer an der richtigen Stelle in einen anderen Zustand versetzt. Außerhalb aller Klammern erfolgt eine Zustandsänderung bei jeder sich öffnenden Klammer aus dem aktuellen Dialekt. Selbiges gilt innerhalb einer Anführung. Wenn der Zustand hingegen durch eine sich öffnende Klammer für Substantive, Pronomen oder Verben verändert wurde, erfolgt eine Zustandsänderung erst wieder bei einer entsprechenden sich schließenden Klammer.

Das einzige Zeichen, das nicht für die Definition eines neuen Klammerdialekts zur Verfügung steht, ist die rechte Klammer für Substantive im Klammerdialekt des Skriptes, in dem die Definition stattfindet. Da empfiehlt es sich dann, für diese Stelle einen Dialekt zu wählen, in dem an dieser Stelle kein Zeichen verwendet wird, welches man benutzen möchte.

Stapel

Ein Stapel ist keine gewöhnliche Gegebenheitsart, sondern stellt als **Stapelspeicher** eine der grundlegenden Strukturen *Lysas* dar.

↗ (u219b)	35	ftc	36
↘ (u21a3)	37	greife.zu	36
↖ (u21a6)	35	kopiere	36
↷ (u21b7)	36	mnt	37
↺ (u21c4)	37	ovr	37
↻ (u23ff)	37	rotiere	36
↠ (u2905)	36	rtt	36
↡ (u2920)	36	swp	37
cpy	36	überspringe	37
drp	35	vertausche	37
dup	35	zeige	37
dupliziere	35			
entferne	35			

↖ (Strg+⤴+u21a6)
duplicate (Englisch)
dupliziere

⟨Gegebenheit⟩ (↖ | dup | dupliziere): ⟨Gegebenheit⟩ ⟨Gegebenheit⟩

Legt die oberste Gegebenheit erneut auf den Stapel (nur eine Referenz, keine echte Kopie [S. 36]).

↗ (Strg+⤴+u219b)
drop (Englisch)
entferne

⟨Gegebenheit⟩ (↗ | drp | entferne)

Entfernt die oberste Gegebenheit vom Stapel.

↦

fetch

greife.zu

(Strg+⇧+u2905)

(Englisch)

$\langle \text{Gegebenheit} \rangle^i \left(\dots \right)^{i-1} \langle \text{Index}_Z \rangle (\mapsto | \text{ftc} | \text{greife.zu}): \langle \text{Gegebenheit} \rangle^i \left(\dots \right)^{i-1} \langle \text{Gegebenheit} \rangle$

Dupliziert eine tiefer liegende Gegebenheit vom Stapel und legt sie obenauf.

Die oberste Gegebenheit vor dem Index ist 1, die zweitoberste 2 usw.

↦

copy

kopiere

(Strg+⇧+u2920)

(Englisch)

$\langle \text{Gegebenheit} \rangle (\mapsto | \text{cpy} | \text{kopiere}): \langle \text{Gegebenheit} \rangle \langle \text{Kopie} \rangle$

Legt eine Kopie der obersten Gegebenheit auf den Stapel, insofern diese kopierbar ist, ansonsten wird auch hier nur die Referenz kopiert.

Außerdem wird bei Gegebenheiten, in denen andere Gegebenheiten referenziert werden (z. B. Regal, Sammlung) auch nur eine Kopie der obersten Ebene (eine Kopie der Liste der Gegebenheiten) erstellt, keine tiefe Kopie (Kopie der referenzierten Gegebenheiten).

↶

rotate

rotiere

(Strg+⇧+u21b7)

(Englisch)

$\langle A \rangle \langle B \rangle \langle C \rangle (\curvearrowright | \text{rtt} | \text{rotiere}): \langle B \rangle \langle C \rangle \langle A \rangle$

Holt die dritte Gegebenheit nach oben auf den Stapel.

↗

over

überspringe

(Strg+⇧+u21a3)

(Englisch)

$\langle A \rangle \langle B \rangle (\rightarrow | \text{ovr} | \text{überspringe}): \langle A \rangle \langle B \rangle \langle A \rangle$

Dupliziert die zweitoberste Gegebenheit und legt sie auf den Stapel

⇄

swap

vertausche

(Strg+⇧+u21c4)

(Englisch)

$\langle A \rangle \langle B \rangle (\rightleftharpoons | \text{swp} | \text{vertausche}): \langle B \rangle \langle A \rangle$

Vertauscht die obersten beiden Gegebenheiten auf dem Stapel.

☞

montri

zeige

(Strg+⇧+u23ff)

(Esperanto)

$\langle \text{Gegebenheit} \rangle (\rightarrow | \text{mnt} | \text{zeige}): \langle \text{Gegebenheit} \rangle$

Gibt Informationen über die oberste Gegebenheit auf dem Stapel in die HTML-Datei aus ohne diese zu entfernen.

Verarbeitung des Skriptes *regal.lysa*: Regal erschaffen • zeigen

Regal:

a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	
m	
n	
o	
p	
q	
r	
s	
t	
u	
v	
w	
x	
y	
z	
Register:	Bd, Ew, Gz, Kl, Kp, Ln, Mk, Rk, Sf, Sl, Tg, Tx, Vl, Zh, Zk, Zn

Ein Regal enthält zum einen Platz für 26 Referenzen auf beliebige Gegebenheiten (a-z). Einige Verben erwarten bestimmte Gegebenheitsarten in bestimmten Fächern des Regals. Dies betrifft insbesondere jene Handlungen, die Gegebenheiten modifizieren. Für solche Gegebenheitsarten steht jeweils noch ein gesondertes Fach zur Verfügung.

↪ ja • Q ungültig • Z ungültig • 𐌹 ungültig

*a-z	39	erschaffe.Regal	41
.a-z, Xx	39		
:a-z, Xx	39	mov	41
®	(u00ae)	prb	41
←	(u2190)	probiere	41
→	(u2192)		
⇐	(u21d0)	registriere	41
brt	41	stv	42
dot	40		
dotiere	40	verstaue	42

*a-z

$\langle \text{Gegebenheit} \rangle (*x): [\langle \text{Gegebenheit} \rangle^x]$

Verschiebt eine Gegebenheit vom Stapel in das bezeichnete Fach des Regals. Sollte das Fach bereits besetzt sein, wird die vorhandene Gegebenheit verdrängt.

.a-z

$[\langle \text{Gegebenheit} \rangle^x] (.x): \langle \text{Gegebenheit} \rangle$

Verschiebt eine Gegebenheit aus einem Fach des Regals auf den Stapel. Das Fach ist danach leer.

Für alle Register gibt es ein entsprechendes Verb, welches eine Gegebenheit daraus auf den Stapel verschiebt, bei der statt eines einzelnen Minuskels das zweistellige Kürzel der Gegebenheitsart anzugeben ist.

:a-z

$[\langle \text{Gegebenheit} \rangle^x] (:x): \langle \text{Gegebenheit} \rangle [\langle \text{Gegebenheit} \rangle^x]$

Kopiert eine Gegebenheit aus einem Fach des Regals auf den Stapel.

Für alle Register gibt es ein entsprechendes Verb, welches eine Gegebenheit daraus auf den Stapel kopiert, bei der statt eines einzelnen Minuskels das zweistellige Kürzel der Gegebenheitsart anzugeben ist.

⇐
dotieren
dotiere

(Strg+⇧+u21d0)

$\langle \text{Regal}_{\text{Rg}} \rangle \langle \text{Gegebenheit} \rangle (\Leftarrow | \text{dot} | \text{dotiere}): \langle \text{Regal}_{\text{Rg}} \rangle$

Füllt ein Register eines Regals auf dem Stapel mit einer geeigneten Gegebenheit. Gegebenenfalls wird dabei der bisherige Inhalt überschrieben.

Regal

Im Handsatz bestand der Arbeitsplatz eines Setzers hauptsächlich aus einem weitgehend genormten Regal (DIN 16501), das in Schubfächern die Schriftkästen enthielt (12-14). Die Höhe bis zur Vorderkante betrug einen Meter, die Tiefe 65 Zentimeter; insofern ein Pultaufsatz vorhanden war (Regel), betrug die hintere Höhe 130 Zentimeter. Die Breite des Regals hing von den Schriftkästen ab. Für große Kästen (Inhalt: 25 kg) betrug sie 105 Zentimeter.^a

Um eine Schrift zu verwenden, wurde der entsprechende Kasten herausgenommen und oben auf das Pult gestellt. An solch einem Regal wurde im Stehen gearbeitet.^b

^a[11, S. 91 f.]

^bDie Idee, die aktuell verwendete Schrift in Analogie zu einem Setzerregal unabhängig vom Stapel zugänglich zu halten, führte letztlich zur Einführung von Regalen in *Lysa*.

(brt | erschaffe.Regal): $\langle \text{Regal}_{\text{rg}} \rangle$

Erschafft ein leeres Regal und legt es auf den Stapel. Um ein Regal nutzen zu können, muß es installiert werden (S. 31).

→

(Strg+⇧+u2192)

probieren
probiere

$\langle \text{Regal}_{\text{rg}} \rangle \langle \text{Fach}_{\text{zk}} \rangle (\rightarrow | \text{prb} | \text{probiere}): \langle \text{Gegebenheit} \rangle$

Kopiert eine Gegebenheit aus einem Fach eines Regals auf dem Stapel. Dieses wird dabei konsumiert. Existiert das angegebene Fach nicht, erfolgt eine Fehlermeldung. Dies ist auch der Fall, wenn ein existierendes Fach leer ist.

®

(Strg+⇧+u00ae)

move
registriere

(Englisch)

$\langle \text{Gegebenheit}_{\text{xx}} \rangle (\text{®} | \text{mov} | \text{registriere}): [\langle \text{Gegebenheit} \rangle^{\text{xx}}]$

Deponiert eine Gegebenheit in einem speziellen Fach des Regals (Register), falls es eines für die entsprechende Gegebenheitsart gibt. Sollte das Register bereits besetzt sein, wird die vorhandene Gegebenheit verdrängt.

Register gibt es für folgende Gegebenheitsarten: **Bd, Ew, Gz, Kl, Kp, Ln, Mk, Rk, Sf, Sl, Tg, Tx, Vl, Zh, Zk, Zn**

←

stivi

verstaue

(Strg+⇧+u2190)

(Esperanto)

$\langle \text{Regal}_{Rg} \rangle \langle \text{Gegebenheit} \rangle \langle \text{Fach}_{Zk}^{[a-z]} \rangle (\leftarrow | \text{stv} | \text{verstaue}): \langle \text{Regal}_{Rg} \rangle$

Füllt das bezeichnete Fach eines Regals auf dem Stapel mit einer Gegebenheit. Gegebenenfalls wird dabei der bisherige Inhalt überschrieben.

Steht für eine andere Gegebenheit. Sie kann direkt eingegeben werden und wird durch eigene Klammern gekennzeichnet. Außerhalb einer Anführung wird ein Pronomen sofort durch die Gegebenheit ersetzt, für die sie steht. Innerhalb einer Anführung hingegen wird ein Pronomen erst dann aufgelöst, wenn die Anführung vereingelicht wird.

↪ nein • Q ungültig • Z ungültig • Å ungültig

?	43	dlt	44
??	44	hebe.auf	43
abl	43	lösche	44
befürworte	43	prn	44
dfn	43	pronominalisiere	44

?

difini

(Esperanto)

befürworte

$\langle \text{Gegebenheit} \rangle \langle \text{Alias}_{zk} \rangle \{ ? \mid \text{dfn} \mid \text{befürworte} \}: \{ \langle \text{Gegebenheit} \rangle^{\text{Alias}} \}^L$

Verbindet eine Zeichenkette mit einer Gegebenheit zu einem Eintrag im lokalen Lexikon.

aboli

(Esperanto)

hebe.auf

$\{ \langle \text{Gegebenheit} \rangle^{\text{Alias}} \}^G \langle \text{Alias}_{zk} \rangle \{ \text{abl} \mid \text{hebe.auf} \}$

Löscht ein Pronomen aus dem globalen Lexikon.

Gibt es keinen entsprechenden Eintrag, passiert nichts.

delete
lösche

(Englisch)

$\{\langle \text{Gegebenheit} \rangle^{\text{Alias}}\}^L \langle \text{Alias}_{zk} \rangle (\text{dlt} \mid \text{lösche})$

Löscht ein Pronomen aus dem lokalen Lexikon.

Gibt es keinen entsprechenden Eintrag, passiert nichts.

??

pronominalisiere
pronominalisiere

$\langle \text{Gegebenheit} \rangle \langle \text{Alias}_{zk} \rangle \{?? \mid \text{prn} \mid \text{pronominalisiere}\}: \\ \{\langle \text{Gegebenheit} \rangle^{\text{Alias}}\}^G$

Verbindet eine Zeichenkette mit einer Gegebenheit zu einem Eintrag im globalen Lexikon.

Eine Anführung enthält eine aus Gegebenheiten zusammengesetzte Ablaufbeschreibung. Sie wird direkt eingegeben und durch ein eigenes Klammerpaar gekennzeichnet. Beim Inhalt kann es sich um Gegebenheiten beliebiger Arten (auch wieder Anführungen) handeln. Solange eine Anführung nicht durch eine abschließende Klammer abgeschlossen wurde, können weitere Gegebenheiten hinzugefügt werden. Diese werden nicht unmittelbar bei der Interpretation effektuert, auch wenn sie danach bereits nur mehr in maschinensprachlicher Form vorliegen. Lediglich Pronomen werden noch nicht aufgelöst.

Bei einer Vereigentlichung erhält jede Anführung ein lokales Lexikon (in Kette mit den übergeordneten Lexika) und eine eigenen Kopie des Regals der übergeordneten Instanz. Der Austausch von Gegebenheiten über die Grenzen einer Anführung hinweg hat über den Stapelspeicher des Interpreters zu geschehen.

↪ nein • Q ungültig • Z ungültig • ⚠ ungültig

⊃ (u21bb) 47	itr 46
⇒ (u21d2) 46	jmp 46
⊗ (u238b) 46	nrs 46
⋈ (u2387) 45	oag 47
↪ (u23ce) 46	vereigentliche 46
⊃ (u27f2) 46	vereigentliche.bedingt 46
entscheide	45	verlasse 46
ewo	45	wiederhole 47
exe	46		
iteriere	46		

⋈ (Strg+⬆+u2387)
entweder oder
entscheide

⟨Bedingung_{Es}⟩ ⟨Entweder_{Af}⟩ ⟨Oder_{Af}⟩ {⋈ | ewo | entscheide): ?

Vereigentlicht abhängig von der Entscheidung eine von zwei Anführungen. Ist diese positiv, ist es die erste (*Entweder*), ansonsten die zweite (*Oder*).

☹ (Strg+⇧+u27f2)
iterate (Englisch)
iteriere

⟨⟨Ablaufbeschreibung_{Af}⟩⟩ ⟨Anzahl_Z⟩ (☹ | itr | iteriere): ?

Vereigentlicht eine Anführung mehrfach unbedingt.

↵ (Strg+⇧+u23ce)
execute (Englisch)
vereigentliche

⟨⟨Ablaufbeschreibung_{Af}⟩⟩ (↵ | exe | vereigentliche): ?

Vereigentlicht eine Anführung vom Stapel unbedingt.

⇒ (Strg+⇧+u21d2)
nur se (Esperanto)
vereigentliche.bedingt

⟨Bedingung_{Es}⟩ ⟨⟨Ablaufbeschreibung_{Af}⟩⟩ (⇒ | nrs | vereigentliche.bedingt): ?

Vereigentlicht eine Anführung bei positiver Entscheidung.

☹ (Strg+⇧+u238b)
jump (Englisch)
verlasse

⟨⟨Bedingung_{Es}⟩ (☹ | jmp | verlasse): ([⟨Signal⟩])⟩⟩

Ist die Bedingung erfüllt, wird ein Abbruchsignal in ein spezielles Fach des Regals gelegt. Die Vereigentlichung einer Anführung wird unmittelbar abgebrochen. Wörter hinter diesem Verb kommen also nicht mehr zur Ausführung.

Außerhalb einer Anführung ist das Verb sinnlos.

⌚

once again
wiederhole

(Strg+⬆+u21bb)
(Englisch)

⟨⟨*Bedingung_{es}*⟩ (⌚ | oag | wiederhole): ([⟨*Signal*⟩])⟩⟩

Ist die Bedingung erfüllt, wird ein Wiederholungssignal in ein spezielles Fach des Regals gelegt. In einer Anführung wird es direkt wieder entfernt und diese wird erneut von Anfang an vereigentlicht. Das heißt, daß alle Wörter, die hinter diesem Verb stehen, nur beim letzten Durchlauf effektuiert werden, wenn kein Wiederholungssignal auf den Stapel gelegt wurde.

Außerhalb einer Anführung ist das Verb sinnlos.

In einem Ablauf wird eine Ausführung als Ablaufbeschreibung mit einer Zeichenkette als Bezeichnung verbunden. Die Bezeichnung wird in das Wörterbuch eingetragen, woraufhin die Ablaufbeschreibung als neue Handlung vereigentlicht werden kann. Insofern stellen Abläufe die Möglichkeit der funktionellen Spracherweiterung dar.

⇒ nein • Q ungültig • Z ungültig • ∆ ungültig

!	48	snn	48
apt	49	synonymisiere	48
cmd	48	verbalisiere	48
		verlerne	49

sinonimo

(Esperanto)

synonymisiere

$\{ \{ \text{Beschreibung}_{Al} \}^{\text{verb}} \} \langle \text{Verb}_{zk} \rangle \langle \text{Synonym}_{zk} \rangle \{ \text{snn} \mid \text{synonymisiere} \} :$
 $\{ \{ \text{Beschreibung}_{Al} \}^{\text{verb}} \} \{ \text{Beschreibung}_{Al} \}^{\text{synonym}} \}$

Legt im Wörterbuch ein Synonym zu einem bereits vorhandenen Verb an.

Handelt es sich beim Alias bereits um ein Verb, so wird dessen Zuordnung überschrieben. Ist es die einzige Verbalisierung einer Handlung, so ist diese nach der Neuordnung nicht mehr ausführbar.

!

command

(Englisch)

verbalisiere

$\langle \text{Verb}_{zk} \rangle \langle \langle \text{Beschreibung}_{Af} \rangle \rangle \{ ! \mid \text{cmd} \mid \text{verbalisiere} \} :$
 $\{ \{ \text{Beschreibung}_{Al} \}^{\text{verb}} \}$

Schafft aus einer Anführung und einer Zeichenkette einen neuen Eintrag im Wörterbuch.

In *Lysa* ist es mit diesem Mechanismus allerdings nicht ohne Tricks (über ein Verb, das vor dem Ablauf definiert wird und hernach als dessen Synonym gesetzt wird) möglich, Funktionen zu programmieren, die sich selbst aufrufen. Rekursion kann jedoch stattdessen über ein Pronomen realisiert werden.

apartigi
verlerne

(Esperanto)

$\{ \{ \text{Beschreibung}_{\text{AI}} \}^{\text{Verb}} \} \langle \text{Verb}_{\text{zk}} \rangle \{ \text{apt} \mid \text{verlerne} \}$

Löscht ein Verb aus dem Wörterbuch. Handelt es sich dabei um die einzige Verbalisierung einer Handlung, so ist diese danach nicht mehr ausführbar.

Entscheidung

Es

Eine Entscheidung repräsentiert ein binäres Urteil, kann also entweder den Wert *Ja* oder *Nein* enthalten.

Entscheidungen können nicht direkt eingegeben werden, sondern werden durch Verben erzeugt.

↪ nein • Q 1, 0 ∨ 0, 0 • Z 1 ∨ 0 • 𐀀 ungültig

<	51	grt	51
>	51		
¬	(u00ac)	kaj	50
∧	(u2227)	konjungiere	51
∨	(u2228)	lsr	51
≅	(u224a)		
⊨	(u22a8)	neg	51
⊭	(u22ad)	negiere	51
adjungiere	50	prüfe.auf.Minderung	51
aut	51	prüfe.auf.Steigerung	51
bejahe	50	prv	50
eql	52	vergleiche	52
fls	52	verneine	52

∨ (Strg+⬆+u2228)
aut (Esperanto)
adjungiere

$\langle A_{Es} \rangle \langle B_{Es} \rangle (\vee \mid \text{aut} \mid \text{adjungiere}): \langle \text{Adjunktion}_{Es} \rangle$

Verknüpft zwei Entscheidungen durch ein logisches Oder miteinander.

⊨ (Strg+⬆+u22a8)
prava (Esperanto)
bejahe

$(\models \mid \text{prv} \mid \text{bejahe}): \langle Ja_{Es} \rangle$

Generiert eine positive Entscheidung.

\wedge (Strg+⤴+u2227)
kaj (Esperanto)
konjungiere

$\langle A_{Es} \rangle \langle B_{Es} \rangle (\wedge \mid \text{kaj} \mid \text{konjungiere}): \langle \text{Konjunktion}_{Es} \rangle$

Verknüpft zwei Entscheidungen durch ein logisches Und miteinander.

\neg (Strg+⤴+u00ac)
negi (Esperanto)
negiere

$\langle \text{Urteil}_{Es} \rangle (\neg \mid \text{neg} \mid \text{negiere}): \langle \text{Gegenteil}_{Es} \rangle$

Verkehrt eine Entscheidung in ihr Gegenteil.

$>$
greater (Englisch)
prüfe.auf.Minderung

$\langle \text{Gegebenheit} \rangle \langle \text{Gegebenheit} \rangle (> \mid \text{grt} \mid \text{prüfe.auf.Minderung}): \langle \text{Urteil}_{Es} \rangle$

Vergleicht zwei beliebige Gegebenheiten miteinander und gibt eine positive Entscheidung zurück, wenn die erste gegenüber der zweiten als vorrangig eingestuft wird. Ist auch nur eine der beiden Gegebenheiten nicht vergleichbar, ist das Ergebnis negativ.

$<$
lesser (Englisch)
prüfe.auf.Steigerung

$\langle \text{Gegebenheit} \rangle \langle \text{Gegebenheit} \rangle (< \mid \text{lsr} \mid \text{prüfe.auf.Steigerung}): \langle \text{Urteil}_{Es} \rangle$

Vergleicht zwei beliebige Gegebenheiten miteinander und gibt eine positive Entscheidung zurück, wenn die erste gegenüber der zweiten als nachrangig eingestuft wird. Ist auch nur eine der beiden Gegebenheiten nicht vergleichbar, ist das Ergebnis negativ.

\approx (Strg+⇧+u224a)
equal (Englisch)
vergleiche

$\langle \text{Gegebenheit} \rangle \langle \text{Gegebenheit} \rangle (\approx \mid \text{eq} \mid \text{vergleiche}): \langle \text{Urteil}_{\text{es}} \rangle$

Vergleicht zwei beliebige Gegebenheiten auf Gleichheit miteinander. Diese kann nur festgestellt werden, wenn es sich entweder um zwei wertvolle Gegebenheiten handelt oder zumindest um zwei gleichartige, vergleichbare Gegebenheiten. Ansonsten wird grundsätzlich keine Gleichheit festgestellt.

$\not\approx$ (Strg+⇧+u22ad)
false (Englisch)
verneine

$(\not\approx \mid \text{fls} \mid \text{verneine}): \langle \text{Nein}_{\text{es}} \rangle$


Generiert eine negative Entscheidung.

Verarbeitung des Skriptes *zeichenkette.lysa*: Φ (Zahl) • digitalisieren • zeigen

Zeichenkette: ‚1.618033989‘

Einerseits die grundlegendste Gegebenheitsart in Lysa, da alle Angaben als Text gemacht werden. Andererseits sind die dazu angebotenen Handlungen begrenzt. Zur Erreichung des Hauptzwecks ist die Manipulation von Zeichenketten nachrangig.

⇒ nein • \mathbb{Q} ungültig • \mathbb{Z} ungültig • \mathbb{A} alphabetisch

Θ	(u236c)	53	cfr	53
	(u261d)	53	dgt	53
beziffere		53	digitize	53
cct		54	verknüpfte	54

Θ (Strg+ $\hat{\cup}$ +u236c)
ciferecigi (Esperanto)
beziffere

$\langle \text{Gegebenheit}_{\mathbb{Q}} \rangle (\Theta \mid \text{cfr} \mid \text{beziffere}): \langle \text{Darstellung}_{\text{zk}} \rangle$

Stellt den Wert einer Gegebenheit als Zeichenkette dar (gegebenenfalls mit einem Komma als Dezimaltrenner).

 (Strg+ $\hat{\cup}$ +u261d)
digitize (Englisch)
digitize

$\langle \text{Gegebenheit}_{\mathbb{Q}} \rangle (\text{hand} \mid \text{dgt} \mid \text{digitize}): \langle \text{Darstellung}_{\text{zk}} \rangle$

Stellt den Wert einer Gegebenheit als Zeichenkette dar (gegebenenfalls mit einem Punkt als Dezimaltrenner).

concatenate
verknüpfe

(Englisch)

$\langle \text{Anfang}_{zk} \rangle \langle \text{Fortsetzung}_{zk} \rangle (\text{cct} \mid \text{verknüpfe}): \langle \text{Zusammenfassung}_{zk} \rangle$

Verknüpft zwei Zeichenketten nahtlos zu einer neuen.

Ganzzahl

Gz

Eine Zahl ohne Nachkommateil. Sie wird als Substantiv eingegeben.

⇒ nein • Q Wert • Z Wert • 𐀓 Wert

++	56	int	55
--	56	runde.ab	55
N ₀	(u2116)	runde.auf	56
[.....	(u2308)	sbr	55
.....	(u230a)	spr	56
gänze	55	zähle.rauf	56
Gz.dec	56	zähle.runter	56
Gz.inc	56		

N₀

(Strg+⤴+u2116)

integra

(Esperanto)

gänze

⟨Gegebenheit_Z⟩ (N₀ | int | gänze): ⟨Wert_{Gz}⟩

Legt den ganzzahligen Wert der obersten Gegebenheit auf den Stapel. Bei wertlosen Gegenständen wird eine Eins abgelegt.

|

(Strg+⤴+u230a)

subronridi

(Esperanto)

runde.ab

⟨Gegebenheit_Q⟩ (| | sbr | runde.ab): ⟨Wert_{Gz}^{↓,0}⟩

Rundet den Wert einer Gegebenheit zur nächsten Ganzzahl ab.

[
superrondigi
runde.auf

(Strg+⤴+u2308)
(Esperanto)

$\langle \text{Gegebenheit}_Q \rangle ([| \text{spr} | \text{runde.auf}]: \langle \text{Wert}_{Gz}^{\uparrow,0} \rangle$

Rundet den Wert einer Gegebenheit zur nächsten Ganzzahl auf.

++
Gz.increment
zähle.rauf

(Englisch)

$[\langle \text{Zähler}_{Gz} \rangle^{Gz}] (++ | \text{Gz.inc} | \text{zähle.rauf}): [\langle \text{Zähler}_{Gz}^{+1} \rangle^{Gz}]$

Erhöht die Ganzzahl im Register um Eins.

--
Gz.decrement
zähle.runter

(Englisch)

$[\langle \text{Zähler}_{Gz} \rangle^{Gz}] (-- | \text{Gz.dec} | \text{zähle.runter}): [\langle \text{Zähler}_{Gz}^{-1} \rangle^{Gz}]$



Erniedrigt die Ganzzahl im Register um Eins.

Verarbeitung des Skriptes *zahl.lysa*: Φ (Zahl) • zeigen

Zahl: 1,618033989 (3ff9e3779b97f4a8)

Eine Zahl mit Nachkommateil. Sie wird als Substantiv eingegeben. Als Dezimaltrennzeichen ist ein Komma anzugeben. Tausendertrennzeichen sind nicht erlaubt. Zur Kennzeichnung negativer Zahlen hat die Angabe mit einem Bindestrich zu beginnen.

\mapsto ja • \mathbb{Q} Wert • \mathbb{Z} | Wert | • Δ Wert

+	58	max	61
²	(u00b2) 62	min	61
³	(u00b3) 60	mod	61
×	(u00d7) 60	mul	60
÷	(u00f7) 59	multipliziere	60
\mathbb{Q}	(u211a) 62	nimm.größeren	61
-	(u2212) 62	nimm.kleineren	61
$\sqrt{}$	(u221a) 64	nimm.Teilungsrest	61
$\sqrt[3]{}$	(u221b) 63	nvt	63
★	(u22c6) 61	potenziere	61
⌊	(u2319) 59	pow	61
	(u2682 / 3) 63	qdr	62
	(u2685) 63	quadriere	62
add	58	rnd	63
addiere	58	sin	58
berechne.Cosinus	58	sqr	64
berechne.Sinus	58	sub	62
berechne.Tangens	59	subtrahiere	62
cbr	63	tan	59
cos	58	umk	59
div	59	val	62
dividiere	59	verwerte	62
dlg	60	würfle	63
kbz	60	würfle.normalverteilt	63
kehre.um	59	ziehe.Kubikwurzel	63
kubiziere	60	ziehe.Quadratwurzel	64
log	60		
logarithmiere.dekadisch	60		
logarithmiere.natürlich	60		

Folgende Zahlen sind als Pronomen im globalen Lexikon gespeichert:

Φ	u03a6	Verhältnis der inneren und äußeren Teilung (> 1), besser bekannt als ›Goldener Schnitt‹.
π	u03c0	Verhältnis von Kreisumfang und -durchmesser.
e	u212f	Der ›natürliche‹ Exponent oder auch Eulersche Zahl.

+
add (Englisch)
addiere

$\langle \text{Augend}_{\mathbb{Q}} \rangle \langle \text{Addend}_{\mathbb{Q}} \rangle (+ \mid \text{add} \mid \text{addiere}): \langle \text{Summe}_{\mathbb{Z}} \rangle$

Bildet die Summe aus den Werten beliebiger Gegebenheiten. Bei wertlosen Gegebenheiten wird mit einer Null gerechnet.

cosinus
berechne.Cosinus

$\langle \text{Winkel}_{\mathbb{Q}}^{\circ} \rangle (\cos \mid \text{berechne.Cosinus}): \langle \text{Verhältnis}_{\mathbb{Z}}^{[0-\pm 1]} \rangle$

Bildet das Verhältnis von Ankathete zur Hypothenuse in einem rechtwinkligen Dreieck zu einem gegebenen Winkel in Grad.

sinus
berechne.Sinus

$\langle \text{Winkel}_{\mathbb{Q}}^{\circ} \rangle (\sin \mid \text{berechne.Sinus}): \langle \text{Verhältnis}_{\mathbb{Z}}^{[0-\pm 1]} \rangle$

Bildet das Verhältnis von Gegenkathete zur Hypothenuse in einem rechtwinkligen Dreieck zu einem gegebenen Winkel in Grad.

tangens berechne.Tangens

$\langle Winkel_Q^{\leq \pm 90^\circ} \rangle$ (tan | berechne.Tangens): $\langle Verh\u00e4ltnis_{Zl}^{[0-\pm 1]} \rangle$

Bildet zu einem gegebenen spitzen Winkel in Grad das Verh\u00e4ltnis von Gegenkathete zur Ankathete in einem rechtwinkligen Dreieck.

\div (Strg+ \uparrow +u00f7)
divide (Englisch)
dividiere

$\langle Dividend_Q \rangle \langle Divisor_Q^{!0} \rangle$ (\div | div | dividiere): $\langle Quotient_{Zl} \rangle$

Bildet den Quotient der Werte zweier beliebiger Gegebenheiten. Bei wertlosen Gegebenheiten wird mit einer Eins gerechnet.

Der Versuch, durch Null zu dividieren, f\u00fchrt zu einer Fehlermeldung nebst Programmabbruch.

\neg (Strg+ \uparrow +u2319)
umkehren
kehre.um

$\langle Vektor_{D \vee T \vee QP} \rangle$ (\neg | umk | kehre.um): $\langle -Vektor_{D \vee T \vee QP} \rangle$
 $\langle Wert_Q \rangle$ (\neg | umk | kehre.um): $\langle -Wert_{Zl} \rangle$

Verkehrt das Vorzeichen des Wertes einer beliebigen Gegebenheit und legt ihn als Zahl auf den Stapel.

Bei Vektorgegebenheitsarten, also Dupeln, Tripeln und Quadrupeln, wird allerdings nicht das Vorzeichen des Wertes, sondern das aller Komponenten umgekehrt.

kubiziere**kubiziere**

$$\langle \text{Basis}_{\mathbb{Q}} \rangle \{^3 \mid \text{kbz} \mid \text{kubiziere}\}: \langle \text{Potenz}_{\mathbb{Z}} \rangle$$

Bildet eine Kubikzahl des Wertes einer beliebigen Gegebenheit.

dekadischer logarithmus**logarithmiere.dekadisch**

$$\langle \text{Numerus}_{\mathbb{Q}} \rangle \{\text{dlg} \mid \text{logarithmiere.dekadisch}\}: \langle \text{Logarithmus}_{\mathbb{Z}} \rangle$$

Bildet den dekadischen Logarithmus des Wertes einer beliebigen Gegebenheit.

logarithmus**logarithmiere.natürlich**

$$\langle \text{Numerus}_{\mathbb{Q}} \rangle \{\log \mid \text{logarithmiere.natürlich}\}: \langle \text{Logarithmus}_{\mathbb{Z}} \rangle$$

Bildet den natürlichen Logarithmus des Wertes einer beliebigen Gegebenheit.

×

(Strg+⇧+u00d7)

multiply

(Englisch)

multipliziere

$$\langle \text{Multiplikator}_{\mathbb{Q}} \rangle \langle \text{Multiplikand}_{\mathbb{Q}} \rangle \{\times \mid \text{mul} \mid \text{multipliziere}\}: \langle \text{Produkt}_{\mathbb{Z}} \rangle$$

Multipliziert die Werte zweier beliebiger Gegebenheiten. Bei wertlosen Gegebenheiten wird mit einer Eins gerechnet.

maximum
nimm.größeren

$\langle A_Q \rangle \langle B_Q \rangle (\max \mid \text{nimm.größeren}): \langle \text{Größerer}_{zl} \rangle$

Wählt aus den Werten zweier beliebiger Gegebenheiten den größeren.

minimum
nimm.kleineren

$\langle A_Q \rangle \langle B_Q \rangle (\min \mid \text{nimm.kleineren}): \langle \text{Kleinerer}_{zl} \rangle$

Wählt aus den Werten zweier beliebiger Gegebenheiten den kleineren.

modulo
nimm.Teilungsrest

$\langle \text{Dividend}_Q \rangle \langle \text{Divisor}_{\mathbb{Q}}^{!0} \rangle (\text{mod} \mid \text{nimm.Teilungsrest}): \langle \text{Rest}_{zl} \rangle$

Ermittelt den Teilungsrest der Division der Werte zweier beliebiger Gegebenheiten. Bei wertlosen Gegebenheiten wird mit einer Eins gerechnet.

Der Versuch, durch Null zu dividieren, führt zu einer Fehlermeldung nebst Programmabbruch.

★ $(\text{Strg} + \hat{+} + \text{u22c6})$
power (Englisch)
potenziere

$\langle \text{Basis}_Q \rangle \langle \text{Exponent}_Q \rangle (\star \mid \text{pow} \mid \text{potenziere}): \langle \text{Potenz}_{zl} \rangle$

Bildet die Potenz der Werte zweier beliebiger Gegebenheiten.

Die Handlung endet mit einer Fehlermeldung, wenn die Basis gleich Null und der Exponent nicht größer als Null, beziehungsweise bei einer Basis kleiner als Null der Exponent nicht ganzzahlig ist.

2

(Strg+⇧+u00b2)

quadriere
quadriere

$\langle Basis_Q \rangle \{^2 \mid \text{qdr} \mid \text{quadriere}\}: \langle Potenz_{zl} \rangle$

Bildet das Quadrat des Wertes einer beliebigen Gegebenheit.

—

(Strg+⇧+u2212)
(Englisch)

subtract
subtrahiere

$\langle Minuend_Q \rangle \langle Subtrahend_Q \rangle \{- \mid \text{sub} \mid \text{subtrahiere}\}: \langle Differenz_{zl} \rangle$

Bildet die Differenz aus den Werten beliebiger Gegebenheiten. Bei wertlosen Gegebenheiten wird mit einer Null gerechnet.

ℚ

(Strg+⇧+u211a)
(Englisch)

value
verwerte

$\langle Gegebenheit_Q \rangle \{\mathbb{Q} \mid \text{val} \mid \text{verwerte}\}: \langle Wert_{zl} \rangle$

Legt den Wert der obersten Gegebenheit als Zahl auf dem Stapel ab. Bei wertlosen Gegebenheiten ist das eine Eins.



cuberoot

ziehe.Kubikwurzel

(Strg+⇧+u221b)
(Englisch)

$\langle Radikand_Q \rangle (\sqrt[3]{\mid} \mid \text{cbr} \mid \text{ziehe.Kubikwurzel}): \langle Radix_{zl} \rangle$

Zieht eine Kubikwurzel aus dem Wert einer beliebigen Gegebenheit.



random

würfle

(Strg+⇧+u2685)
(Englisch)

$(\text{random} \mid \text{rnd} \mid \text{würfle}): \langle Zufallszahl_{zl}^{[0-1]} \rangle$

Generiert eine Pseudo-Zufallszahl zwischen Null und Eins (exklusiv) und legt sie auf den Stapel. Der Zeitpunkt des Programmstarts bestimmt, welche Zufallszahlen erzeugt werden.

Insofern eine Zufallszahl aus einem bestimmten Zahlenbereich benötigt wird, könnte z. B. solch ein Makro hilfreich sein:

$\text{von}_Q \text{ bis}_Q \langle Zfz \rangle \langle (\rightarrow) (-) (\text{random}) (\times) (+) \rangle (!): \text{Zufallszahl}_{zl}^{[\text{von-bis}]}$



normalverteilung

würfle.normalverteilt

(Strg+⇧+u2682/3)

$(\text{normal} \mid \text{nvt} \mid \text{würfle.normalverteilt}): \langle Zufallszahl_{zl} \rangle$

Erzeugt eine normalverteilte Zufallszahl mit dem Median Null und einer Standardabweichung von Eins und legt diese auf den Stapel.

99,73% aller so produzierten Zufallszahlen werden sich im Bereich von ± 3 bewegen. Auch wenn dies selten vorkommt, sind Ausreißer

mit extremen Werten möglich. Insofern diese ausgeschlossen werden müssen, ist das Ergebnis zu kontrollieren und gegebenenfalls zu beschränken, etwa mit solch einer Formulierung:

$$\lfloor \text{Wert}_{\text{Q}} \rfloor \lfloor \text{Minimum}_{\text{Q}} \rfloor \lfloor \text{Maximum}_{\text{Q}} \rfloor \langle \text{ } \rangle \langle \langle (\text{ }) (\text{min}) (\text{max}) \rangle \rangle (!):$$

$$\lfloor \text{Wert}_{\text{ZI}}^{\text{[min-max]}} \rfloor$$

$\sqrt{}$	(Strg+⇧+u221a)
sq uare root	(Englisch)
ziehe.Quadratwurzel	

$$\langle Radikand_{\mathbb{Q}}^{\geq 0} \rangle (\sqrt{\mid \text{sqr} \mid \text{ziehe.Quadratwurzel}}): \langle Radix_{z_l} \rangle$$

Zieht eine Quadratwurzel aus dem Wert einer beliebigen Gegebenheit.

Der Versuch, die Wurzel eines negativen Wertes zu ziehen, führt zu einer Fehlermeldung nebst Programmabbruch.

Ein Dupel ist eine aus zwei Zahlen zusammengesetzte Gegebenheitsart (**Tupel**). Es kann nicht direkt eingegeben werden, sondern wird mittels eines Verbs erzeugt.

Als Wert gibt ein Dupel die Quadratwurzel der Summe der Quadrate seiner Komponenten zurück (**euklidische Norm**). Darüber wird es auch verglichen.

Ein Dupel kann unter anderem eine komplexe Zahl, ein Intervall, Flächenkoordinaten oder einen ebensolchen Vektor enthalten.

$$\mapsto \text{nein} \bullet \mathbb{Q} \sqrt{a^2 + b^2} \bullet \mathbb{Z} \left[\text{Wert} \right] \bullet \mathbb{R} \text{Wert}$$

· (u22c5) 66	multipliziere.mit.Skalar 66
addiere.Vektoren 65	skp 66
berechne.Skalarprodukt 66	smp 66
D 66	unm 67
dig 66	vad 65
generiere.Dupel 66	vereinheitliche 67
löse.auf 66	zdp 66

vektoraddition

addiere.Vektoren

$$\langle \text{Vektor}_{D \vee T \vee Qp} \rangle \langle \text{Vektor}_{D \vee T \vee Qp} \rangle \{ \text{vad} \mid \text{addiere.Vektoren} \} : \langle \text{Vektor}_{D \vee T \vee Qp} \rangle$$

Bildet die Summe zweier Dupel, Tripel oder Quadrupel (gleichartig). Die Komponenten werden dabei elementweise zusammengezählt.

Um Vektoren statt zusammenzuzählen voneinander abzuziehen, ist der abzuziehende Vektor vor der Addition umzukehren (S. 59).

(Strg+⇧+u22c5)

skalarprodukt

berechne.Skalarprodukt

$\langle \text{Multiplikator}_{D \vee T \vee Qp} \rangle \langle \text{Multiplikand}_{D \vee T \vee Qp} \rangle (\cdot \mid \text{skp} \mid$
 $\text{berechne.Skalarprodukt}): \langle \text{Skalar}_{zl} \rangle$

Berechnet aus zwei gleichartigen Vektorgegebenheiten das Skalarprodukt.

D

zahlendupel

generiere.Dupel

$\langle a_Q \rangle \langle b_Q \rangle (D \mid \text{zdp} \mid \text{generiere.Dupel}): \langle \text{Zahlenpaar}_{dp} \rangle$

Generiert aus zwei Werten ein Dupel.

disintegrate

(Englisch)

löse.auf

$\langle \text{Dupel}_{dp} \rangle (\text{dig} \mid \text{löse.auf}): \langle A_{zl} \rangle \langle B_{zl} \rangle$
 $\langle \text{Tripel}_{Tp} \rangle (\text{dig} \mid \text{löse.auf}): \langle A_{zl} \rangle \langle B_{zl} \rangle \langle C_{zl} \rangle$
 $\langle \text{Quadrupel}_{Qp} \rangle (\text{dig} \mid \text{löse.auf}): \langle A_{zl} \rangle \langle B_{zl} \rangle \langle C_{zl} \rangle \langle D_{zl} \rangle$
 $\langle \text{Tripelkarree}_{Tk} \rangle (\text{dig} \mid \text{löse.auf}): \langle \text{Komponenten}_{SI}^{Zl|9|} \rangle$
 $\langle \text{Quadrupelkarree}_{Tk} \rangle (\text{dig} \mid \text{löse.auf}): \langle \text{Komponenten}_{SI}^{Zl|16|} \rangle$

Löst ein Tupel oder ein Karree in seine Komponenten auf.

skalarmultiplikation

multipliziere.mit.Skalar

$\langle \text{Skalar}_Q \rangle \langle \text{Vektor}_{D \vee T \vee Qp} \rangle (\text{smp} \mid \text{multipliziere.mit.Skalar}):$
 $\langle \text{Vektor}_{D \vee T \vee Qp} \rangle$
 $\langle \text{Skalar}_Q \rangle \langle \text{Matrix}_{T \vee Qk} \rangle (\text{smp} \mid \text{multipliziere.mit.Skalar}): \langle \text{Matrix}_{T \vee Qk} \rangle$

Multipliziert den Wert einer beliebigen Gegebenheit als Skalar mit einer Vektorgegebenheit oder einem Karree.

ununorma
vereinheitliche

(Esperanto)

$\langle \text{Vektor}_{D \vee T \vee Qp} \rangle$ (unm | vereinheitliche): $\langle \text{Einheitsvektor}_{D \vee T \vee Qp} \rangle$

Generiert zu einer Vektorgegebenheit einen Einheitsvektor.

Verarbeitung des Skriptes *tripel.lysa*: Weißpunkt D65 (Tripel) • zeigen

Tripel: 0,950429662; 1; 1,088800568

Ein Tripel ist eine aus drei Zahlen zusammengesetzte Gegebenheitsart. Es kann nicht direkt eingegeben werden, sondern wird mittels eines Verbs erzeugt.

Als Wert gibt ein Tripel die Quadratwurzel der Summe der Quadrate seiner Komponenten zurück (**euklidische Norm**). Darüber wird es auch verglichen.

Ein Tripel kann unter anderem eine Raumkoordinate oder einen ebensolchen Vektor, eine **homogene** Flächenkoordinate oder auch Farbkomponenten enthalten.

$$\mapsto \text{nein} \bullet \mathbb{Q} \sqrt{a^2 + b^2 + c^2} \bullet \mathbb{Z} \left[\text{Wert} \right] \bullet \mathbb{N} \text{Wert}$$

× (u2a2f) 68 generiere.Tripel 69
errechne.Kreuzprodukt 68 ztp 69

× (Strg+⬆+u2a2f)
kreuzprodukt
errechne.Kreuzprodukt

$\langle A_{Tp} \rangle \langle B_{Tp} \rangle (\times \mid \text{kpd} \mid \text{errechne.Kreuzprodukt}): \langle \text{Kreuzprodukt}_{Tp} \rangle$

Berechnet aus zwei Tripeln als Vektoren das **Kreuzprodukt**, wiederum einen Vektor, der senkrecht zu den beiden anderen steht.

Das Kreuzprodukt (Vektorprodukt) ist nicht kommutativ. Ein Vertauschen der Argumente führt zu einem Vektor, der in die entgegengesetzte Richtung führt.

zahlentripel
generiere.Tripel

$\langle a_{\mathbb{Q}} \rangle \langle b_{\mathbb{Q}} \rangle \langle c_{\mathbb{Q}} \rangle$ (ztp | generiere.Tripel): $\langle \text{Zahlendrilling}_{\text{Tp}} \rangle$

Generiert aus drei Werten ein Tripel.

Quadrupel

qp

Ein Quadrupel ist eine aus vier Zahlen zusammengesetzte Gegebenheitsart. Es kann nicht direkt eingegeben werden, sondern wird vermittelt eines Verbs erzeugt.

Als Wert gibt ein Quadrupel die Quadratwurzel der Summe der Quadrate seiner Komponenten zurück (**euklidische Norm**).

Ein Quadrupel kann unter anderem eine **homogene** Raumkoordinate, ein Quaternion oder auch Farbkomponenten nebst Alphakanal enthalten.

$$\mapsto \text{nein} \bullet \mathbb{Q} \sqrt{a^2 + b^2 + c^2 + d^2} \bullet \mathbb{Z} \left[\text{Wert} \right] \bullet \mathbb{R} \text{Wert}$$

generiere.Quadrupel 70 zqp 70

zahlenquadrupel

generiere.Quadrupel

$\langle a_{\mathbb{Q}} \rangle \langle b_{\mathbb{Q}} \rangle \langle c_{\mathbb{Q}} \rangle \langle d_{\mathbb{Q}} \rangle \{ \text{zqp} \mid \text{generiere.Quadrupel} \}: \langle \text{Zahlenvierling}_{\text{qp}} \rangle$

Generiert aus vier Werten ein Quadrupel.

Verarbeitung des Skriptes *tripelkarree.lysa*: 1 • 0,329 • dividieren • 0,3127 • 0,329 • 0,3583 • Tripel generieren • mit Skalar multiplizieren • Sammlung erschaffen • 0,64 • 0,3 • 0,15 • 0,33 • 0,6 • 0,06 • 0,03 • 0,1 • 0,79 • einsammeln • Tripelkarree generieren • duplizieren • in m deponieren • verkehren • vertauschen • Vektorprodukt berechnen • 1 • 1 • 1 • Tripel generieren • vertauschen • dyadisches Produkt bilden • aus m entnehmen • elementweise multiplizieren • zeigen

Tripelkarree:

```
[ 0,412390799 0,357584339 0,180480788 ]
[ 0,212639006 0,715168679 0,072192315 ]
[ 0,019330819 0,11919478 0,950532152 ]
```

Ein Tripelkarree ist eine aus neun Zahlen zusammengesetzte Gegebenheitsart, die eine 3×3 -Matrix repräsentiert. Es kann nur mittels Verben erzeugt werden.

Der Wert eines Tripelkarrees errechnet sich als Quadratwurzel der Summe der Quadrate aller Komponenten (**Frobeniusnorm**). Darüber werden Tripelkarrees auch verglichen.

↪ ja • \mathbb{Q} Frobeniusnorm • \mathbb{Z} ungültig • Δ Wert

⊗ (u2297)	73	ivs	76
⊙ (u2299)	74	mmp	74
⊠ (u2339)	76	multipliziere.elementweise	74
berechne.Vektorprodukt	71	multipliziere.Matrizen	74
bilde.Dyade	73	tpk	73
dya	73	tpn	75
generiere.Tripelkarree	73	transponiere	75
hdm	74	verkehre	76
			vpd	71

vektorprodukt

berechne.Vektorprodukt

$\langle \text{Multiplikator}_{T_V Q_k} \rangle \langle \text{Multiplikand}_{T_V Q_p} \rangle (\text{vpd} \mid$
berechne.Vektorprodukt): $\langle \text{Produkt}_{T_V Q_p} \rangle$

Multipliziert ein Tripel- oder Quadrupelkarree mit einem entsprechenden Vektor (Tripel oder Quadrupel) zu einem entsprechenden Vektor.

Farbraumkonvertierung

Um eine Farbe in einen anderen Farbraum zu übertragen, wird eine Konvertierungsmatrix mit ihr als Vektor multipliziert. Diese Matrix selbst wird aus den Primärvalenzen und dem Weißpunkt errechnet. Exemplarisch wird hier die Erstellung einer Konvertierungsmatrix zwischen dem linearen Wide Gamut-RGB-Farbraum gemäß **BT.2100-2** und dem CIE-XYZ₁₉₃₁-Farbraum (2°) mit dem Weißpunkt D65 dargestellt:

Der Weißpunkt stimmt in beiden Farbräumen überein. Er ist mit den xy-Koordinaten 0,3127 und 0,329 angegeben.^a Für die z-Koordinate ergibt sich daraus 0,3583 ($z = 1 - x - y$). Diese Angaben werden auf die y-Koordinate normiert:

$$\frac{1}{0,329} \cdot \begin{bmatrix} 0,3127 \\ 0,329 \\ 0,3583 \end{bmatrix}_{D65} = \begin{bmatrix} 0,950455927 \\ 1 \\ 1,089057751 \end{bmatrix}$$

Der Weißpunkt ist das Ergebnis der Multiplikation einer Matrix der Primärvalenzen mit einem Vektor:

$$\begin{bmatrix} 0,950455927 \\ 1 \\ 1,089057751 \end{bmatrix} = \begin{bmatrix} 0,708 & 0,17 & 0,131 \\ 0,292 & 0,797 & 0,046 \\ 0 & 0,033 & 0,823 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Der Vektor kann errechnet werden, indem die Kehrmatrix der Primärvalenzen mit dem Weißpunkt multipliziert wird. Zur weiteren Verarbeitung wird ein Identitätsvektor (1; 1; 1) gebildet und mit dem Ergebnis zu einem dyadischen Produkt multipliziert. Damit kann nun die Matrix der Primärvalenzen elementweise malgenommen werden (Hadamard-Produkt). Heraus kommt die Konvertierungsmatrix aus dem Wide Gamut-RGB-Farbraum in den CIE-XYZ-Farbraum:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,636958048 & 0,144616904 & 0,168880975 \\ 0,262700212 & 0,677998072 & 0,059301716 \\ 0 & 0,028072693 & 1,060985058 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Für die Umwandlung aus dem CIE-XYZ-Farbraum in den Wide Gamut-RGB-Farbraum dient die Kehrmatrix:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1,716651188 & -0,355670784 & -0,253366281 \\ -0,666684352 & 1,616481237 & 0,015768546 \\ 0,017639857 & -0,042770613 & 0,942103121 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

“vergleiche dazu die sich aus den Illuminant D65 ergebenden Werte, S. 118 sowie [9, S. 112]



(Strg+⬆+u2297)

dyade

bilde.Dyade

$\langle A_{TVQP} \rangle \langle B_{TVQP} \rangle (\otimes \mid \text{dya} \mid \text{bilde.Dyade}): \langle Matrix_{TVQk} \rangle$

Bildet zu zwei Zahlentripeln oder -quadrupeln als dyadisches Produkt ein entsprechendes Karree.

tripelkarree

generiere.Tripelkarree

$\langle Sammlung_{SI}^{[9]} \rangle (\text{tpk} \mid \text{generiere.Tripelkarree}): \langle Matrix_{Tk} \rangle$

Generiert aus den Werten einer Sammlung von genau neun Gegebenheiten einer beliebigen Art eine 3×3 -Matrix. Dabei werden zuerst die Zeilen gefüllt. Bei wertlosen Gegebenheiten wird mit einer Eins gerechnet.



(Strg+⬆+u2299)

hadamard-produkt

multipliziere.elementweise

$\langle \text{Multiplikator}_{TVQk} \rangle \langle \text{Multiplikand}_{TVQk} \rangle (\odot \mid \text{hdm} \mid$
multipliziere.elementweise): $\langle \text{Produkt}_{TVQk} \rangle$

Multipliziert zwei Tripel- oder Quadrupelkarrees elementweise.

matrixmultiplikation

multipliziere.Matrizen

$\langle \text{Multiplikator}_{TVQk} \rangle \langle \text{Multiplikand}_{TVQk} \rangle (\text{mmp} \mid$
multipliziere.Matrizen): $\langle \text{Produkt}_{TVQk} \rangle$

Multipliziert jeweils zwei Tripel- oder Quadrupelkarrees. Da eine Matrixmultiplikation nicht kommutativ ist, kommt es auf die Reihenfolge an ($A \times B \neq B \times A$). Zu beachten ist, daß die Zeilen des Multiplikators mit den Spalten des Multiplikanden malgenommen werden.¹¹

Transformationsmatrix

Mit Matrizen lassen sich Transformationen eines Koordinatensystems ausführen. Insbesondere ist es möglich, durch Multiplikation mehrerer Matrizen mehrere Transformationen in einer einzigen Matrix zu bündeln. Bei der Multiplikation ist die Reihenfolge zu beachten, da eine Matrizenmultiplikation nicht kommutativ ist:

$$\text{neu} = \text{aktuell} \times \text{modifikation}$$

Für Transformation eignen sich **homogene Koordinaten**, da damit auch Translationen durch Multiplikationen ausgeführt werden können. In der Fläche ergeben sich somit folgende 3×3 -Matrizen für ein linkshändisches Koordinatensystem:

¹¹[20, S. 10]

Identität: Damit bleibt alles, wie es ist

Translation: Verschiebt den Ursprung des Koordinatensystems

Skalierung: Multipliziert die Maßstäbe der Achsen mit Faktoren

Rotation: Dreht das Koordinatensystem um den Ursprung

horizontale Scherung: Verschiebt die horizontalen Positionen abhängig vom Winkel

vertikale Scherung: Verschiebt die vertikalen Positionen abhängig vom Winkel

Spiegeln am Ursprung: Alles verkehrtherum

horizontales Spiegeln: Verkehrt links und rechts

vertikales Spiegeln: Verkehrt oben und unten

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} F_h & 0 & 0 \\ 0 & F_v & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos(w) & \sin(w) & 0 \\ -\sin(w) & \cos(w) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \tan(w) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ \tan(w) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Um schließlich Koordinaten des neuen in das ursprüngliche Koordinatensystem zu übersetzen, ist die Matrix aus allen Transformationen mit den Koordinaten als Vektor ($z = 1$) zu einem Vektor zu multiplizieren.

Für Raumkoordinaten gilt entsprechendes. Da handelt es sich dann allerdings um 4×4 -Matrizen.

transponieren transponiere

$\langle \text{Matrix}_{T \vee Qk} \rangle$ (tpn | transponiere): $\langle \text{transponierte Matrix}_{T \vee Qk} \rangle$

Transponiert ein Tripel- oder Quadrupelkarree, vertauscht also Spalten und Zeilen miteinander.



$\langle Matrix_{T \vee Qk} \rangle$ (⬆ | ivs | verkehre): $\langle Kehrmatrix_{T \vee Qk} \rangle$

Bildet zu einem Tripel- oder Quadrupelkarree die Kehrmatrix. Das Tripelkarree wird nach der **Regel von Sarrus**¹² invertiert.

Inversionen scheitern, wenn die Determinante des zu invertierenden Karrees Null ist.

¹²[20, S. 22]

Quadrupelkarree

ok

Verarbeitung des Skriptes *quadrupelkarree.lysa*: Sammlung erschaffen • 1 • 2 • 3 • 4 • 5 • 6 • 7 • 8 • 9 • 10 • 11 • 12 • 13 • 14 • 15 • 16 • einsammeln • Quadrupelkarree generieren • zeigen

Quadrupelkarree:

```
[ 1 2 3 4 ]
| 5 6 7 8 |
| 9 10 11 12 |
[ 13 14 15 16 ]
```

Ein Quadrupelkarree ist eine aus 16 Zahlen zusammengesetzte Gegebenheitsart, die eine 4×4 -Matrix repräsentiert. Es kann nicht direkt eingegeben werden, sondern wird mittels Verben erzeugt.

Der Wert eines Quadrupelkarrees errechnet sich als Quadratwurzel der Summe der Quadrate aller Komponenten (**Frobeniusnorm**). Darüber werden Quadrupelkarrees auch verglichen.

↪ ja • \mathbb{Q} Frobeniusnorm • \mathbb{Z} ungültig • Δ Wert

generiere.Quadrupelkarree 77 qpk 77

quadrupelkarree

generiere.Quadrupelkarree

$\langle \text{Sammlung}_{SI}^{16} \rangle$ (qpk | generiere.Quadrupelkarree): $\langle \text{Matrix}_{Ok} \rangle$

Generiert aus den Werten einer Sammlung von genau 16 Gegebenheiten einer beliebigen Art eine 4×4 -Matrix. Dabei werden die Zeilen zuerst gefüllt. Bei wertlosen Gegebenheiten wird mit einem Wert von Eins gerechnet.

Eine Sammlung enthält Gegebenheiten einer beliebigen Art (außer Sammlungen). Als Wert gibt sie die Summe der Werte ihrer Elemente zurück. Als Ganzzahl gibt sie hingegen ihre Mächtigkeit zurück. Darüber ist sie auch mit anderen Sammlungen vergleichbar.

Die erste zu einer Sammlung hinzugefügte Gegebenheit entscheidet über die Art der gesammelten Gegebenheiten. Nachdem eine Sammlung vollständig geleert wurde, kann darin eine andere Gegebenheitsart gesammelt werden.

↪ ja • \mathbb{Q} Summe der Werte aller Elemente • \mathbb{Z} Mächtigkeit • \aleph Mächtigkeit (untereinander)

∃ (u2203)	81	klk	78
∈ (u2208)	79			
∉ (u2209)	79	mische	80
∃ (u220b)	80			
∄ (u220c)	79	sammle.ein	80
♠ (u2660)	80	Sl.ald	79
♣ (u2663)	81	Sl.eld	79
col	80	Sl.fdn	79
erschaffe.Sammlung	78	Sl.occ	81
füge.hinzu	79	Sl.ord	81
			Sl.sfl	80
gib.ab	79	sondiere	81
gib.heraus	79	sortiere	81

kolekto

(Esperanto)

erschaffe.Sammlung

(klk | erschaffe.Sammlung): $\langle \text{Sammlung}_{si} \rangle$

Erschafft eine leere Sammlung und legt sie auf den Stapel.

€
Sl.aldoni
füge.hinzu

(Strg+⬆+u2208)
 (Esperanto)

$$\left[\langle \text{Sammlung}_{Sl}^{Xx|n|} \rangle^{Sl} \right] \langle \text{Gegebenheit}_{xx} \rangle (\epsilon | \text{Sl.ald} | \text{f\ddot{u}ge.hinzu}):$$

$$\left[\langle \text{Sammlung}_{Sl}^{Xx|n+1|} \rangle^{Sl} \right]$$

Fügt der Sammlung im Register eine Gegebenheit vom Stapel hinzu. Diese muß von der Art her denen bereits in der Sammlung gespeicherten Gegebenheiten entsprechen. Ist die Sammlung im Magazin leer, wird die Art deren Inhalts entsprechend gesetzt.

∅
Sl.fdoni
gib.ab

(Strg+⬆+u220c)
 (Esperanto)

$$\left[\langle \text{Sammlung}_{Sl}^{Xx|n|} \rangle^{Sl} \right] (\emptyset | \text{Sl.fdn} | \text{gib.ab}): \langle \text{Gegebenheit}_{xx} \rangle$$

$$\left[\langle \text{Sammlung}_{Sl}^{Xx|n-1|} \rangle^{Sl} \right]$$

Gibt aus der Sammlung im Register die zuletzt hinzugefügte beziehungsweise am Ende stehende Gegebenheit heraus und legt sie auf den Stapel. Sollte es sich dabei um die letzte Gegebenheit in der Sammlung gehandelt haben, kann diese danach wieder jede beliebige Gegebenheitsart außer Sammlungen enthalten.

∉
Sl.eldoni
gib.heraus

(Strg+⬆+u2209)
 (Esperanto)

$$\left[\langle \text{Sammlung}_{Sl}^{Xx|n|} \rangle^{Sl} \right] (\notin | \text{Sl.eld} | \text{gib.heraus}): \langle \text{Gegebenheit}_{xx} \rangle$$

$$\left[\langle \text{Sammlung}_{Sl}^{Xx|n-1|} \rangle^{Sl} \right]$$

Gibt aus der Sammlung im Register die zuletzt hinzugefügte beziehungsweise am Ende stehende Gegebenheit heraus und legt sie

auf den Stapel. Sollte es sich dabei um die letzte Gegebenheit in der Sammlung gehandelt haben, kann diese danach wieder jede beliebige Gegebenheitsart außer Sammlungen enthalten.



Sl.shuffle
mische

(Strg+⇧+u2660)
(Englisch)

$[\langle \text{Sammlung}_{sl} \rangle^{sl}] (\spadesuit \mid \text{Sl.sfl} \mid \text{mische}): [\langle \text{Sammlung}_{sl} \rangle^{sl}]$

Bringt die Gegebenheiten in einer Sammlung im Regal (s) in eine zufällige Reihenfolge.



collect
sammle.ein

(Strg+⇧+u220b)
(Englisch)

$\langle \text{Sammlung}_{sl}^{[n]} \rangle (\langle \text{Gegebenheit}_{sl} \rangle)^{m>0} (\exists \mid \text{col} \mid \text{sammle.ein}):$
 $\langle \text{Sammlung}_{sl}^{[n+m]} \rangle$

Nimmt beliebig viele, gleichartige Gegebenheiten vom Stapel, bis es auf eine Sammlung trifft, zu welcher die aufgenommenen Gegebenheiten hinzugefügt werden. Die erweiterte Sammlung wird dann wieder auf den Stapel gelegt.

Entweder handelt es sich um eine leere Sammlung oder diese muß bereits Gegebenheiten der gleichen Art enthalten, wie hinzugefügt werden sollen. Enthält die Sammlung allerdings Ganzzahlen oder Zahlen, so werden die entsprechenden Werte (unabhängig von ihrer Gültigkeit) hinzugefügt und es kann sich um beliebige Gegebenheiten außer Sammlungen handeln. (**Achtung:** Die letzte Gegebenheit entscheidet bei einer leeren Sammlung über die Art der gesammelten Gegebenheiten!)

∃
Sl.occupied
sondiere

(Strg+⇧+u2203)
 (Englisch)

$[\langle \text{Sammlung}_{sl} \rangle^{sl}] (\exists \mid \text{Sl.occ} \mid \text{sondiere}): \langle \text{IstNochNichtLeer}_{es} \rangle$
 $[\langle \text{Sammlung}_{sl} \rangle^{sl}]$

Prüft, ob sich in der registrierten Sammlung noch Gegebenheiten befinden.

♣
Sl.ordigi
sortiere

(Strg+⇧+u2663)
 (Esperanto)

$[\langle \text{Sammlung}_{sl} \rangle^{sl}] (\clubsuit \mid \text{Sl.ord} \mid \text{sortiere}): [\langle \text{Sammlung}_{sl} \rangle^{sl}]$

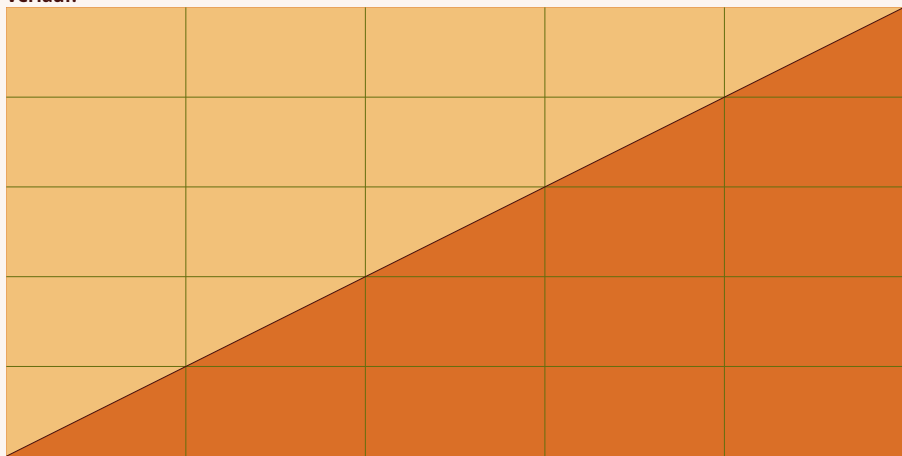
Sortiert eine Sammlung im Register aufsteigend.

Um die Ordnung in einer Sammlung umzukehren (ab- statt aufsteigend), empfiehlt sich solch eine Formulierung:

$\llbracket \text{Sammlung}_{sl} \rrbracket (\mapsto) (\textcircled{R}) \langle\langle (\exists) (\neg) (\textcircled{N}) (\notin) (\models) (\cup) \rangle\rangle (\textcircled{J}) (\text{.Sl}) (\nrightarrow) (\exists)$

Verarbeitung des Skriptes *verlauf.lysa*: Linear (Verlauf) • zeigen

Verlauf:



Ein Verlauf repräsentiert eine frei programmierbare mathematische Funktion mit dem abgeschlossenen Intervall Null bis Eins sowohl für den Definitions- als auch den Wertebereich.

Verläufe können nicht direkt eingegeben werden, sondern werden durch ein Verb erzeugt.

Die Kopie eines Verlaufs enthält gegebenenfalls eine Kopie des Regals, mit dem das Original ausgestattet ist.

↪ ja • \mathbb{Q} ungültig • \mathbb{Z} ungültig • \mathbb{R} ungültig

↗	(u2934)	87	interpoliere.kubisch	88
bzr	88		prg	87
generiere.Verlauf	87			
gib.Einzelwert	87		Vl.vlr	87

Folgender Verlauf ist als Pronomen im globalen Lexikon gespeichert:

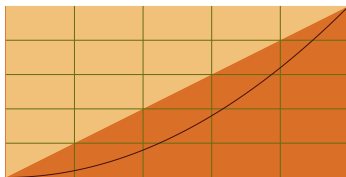
Linear	Ein linearer Verlauf, der den Eingabewert unverändert zurückgibt und aus einer leeren Anführung besteht (siehe oben)
--------	--

Eine Reihe weiterer Verläufe als Vorschläge; viele davon gehen auf **Robert Penner** zurück:

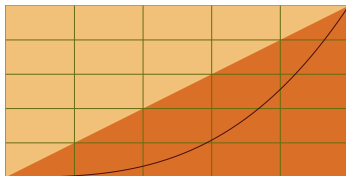
Kode

InQuad: $\langle\langle\{^2\}\rangle\rangle\{\text{↗}\}$

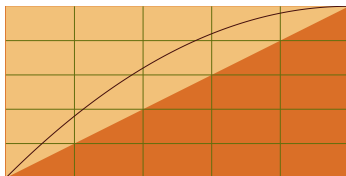
Verlauf



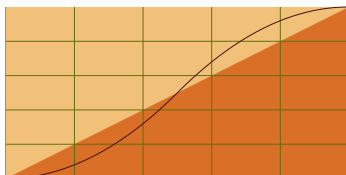
InCubic: $\langle\langle\{^3\}\rangle\rangle\{\text{↗}\}$



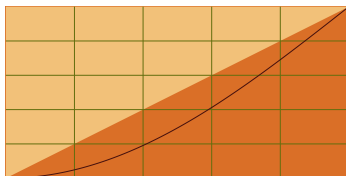
OutQuad: $\langle\langle(\mapsto)\langle 2\rangle\{\rightleftharpoons\}(-)(\times)\rangle\rangle\{\text{↗}\}$



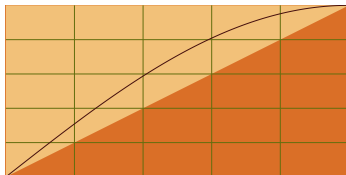
InOutQuad: $\langle\langle(\mapsto)\langle 0,5\rangle\{<\}\langle\{^2\}\langle 2\rangle\{(\times)\}\rangle\rangle\rangle\langle\langle 1\rangle\{\rightleftharpoons\}\langle -2\rangle\{(\times)\}\langle 2\rangle\{(+)\}\langle 2\rangle\{(\div)\}\langle -\rangle\rangle\{(\backslash)\}\rangle\rangle\{\text{↗}\}$



InSine: $\langle\langle\langle 1\rangle\{\rightleftharpoons\}\langle 90\rangle\{(\times)\}\{\cos\}\{-\rangle\rangle\rangle\{\text{↗}\}$



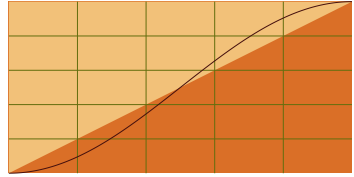
OutSine: $\langle\langle\langle 90\rangle\{(\times)\}\{\sin\}\rangle\rangle\{\text{↗}\}$



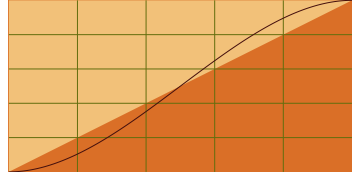
Kode

Verlauf

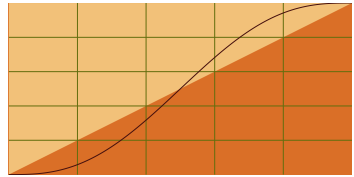
InOutSine: $\langle\langle 0,5 \rangle \{\rightleftharpoons\} \langle 180 \rangle \{\times\} \{\cos\} \langle 2 \rangle \{\div\} \{-\}\rangle\{\uparrow\}$



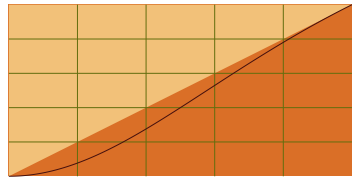
Smoothstep: $\langle\langle (\mapsto) \langle 2 \rangle \{\rightleftharpoons\} \langle 2 \rangle \{\times\} \langle 3 \rangle \{\rightleftharpoons\} \{-\} \{\times\} \rangle\{\uparrow\}$



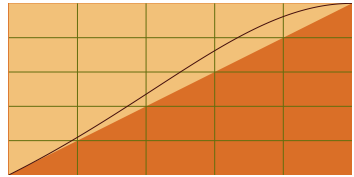
Smootherstep: $\langle\langle (\mapsto) \langle 3 \rangle \{\rightleftharpoons\} (\mapsto) \langle 6 \rangle \{\times\} \langle 15 \rangle \{-\} \{\times\} \langle 10 \rangle \{+\} \{\times\} \rangle\{\uparrow\}$



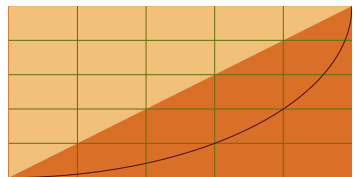
gebrochenrational-a¹³: $\langle\langle \langle 2 \rangle (\mapsto) \langle 1 \rangle \{\rightleftharpoons\} \{-\} \{\rightleftharpoons\} \langle 1 \rangle \{+\} \{\div\} \langle 1 \rangle \{\rightleftharpoons\} \{-\} \rangle\{\uparrow\}$



gebrochenrational-b¹⁴: $\langle\langle \langle 1 \rangle \{\rightleftharpoons\} \{-\} \langle 2 \rangle (\mapsto) \langle 1 \rangle \{\rightleftharpoons\} \{-\} \{\rightleftharpoons\} \langle 1 \rangle \{+\} \{\div\} \rangle\{\uparrow\}$



InCirc: $\langle\langle \langle 1 \rangle \langle 1 \rangle \{\curvearrowright\} \langle 2 \rangle \{-\} \{\sqrt{}\} \{-\} \rangle\{\uparrow\}$

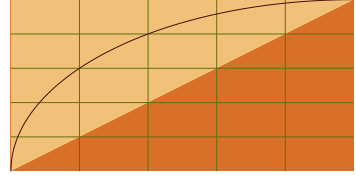


¹³ $1 - \frac{1-x^2}{1+x^2}$
¹⁴ $\frac{1-(1-x)^2}{1+(1-x)^2}$

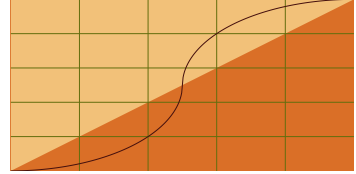
Kode

OutCirc: $\langle\langle 1 \rangle \{\rightleftharpoons\} \langle 1 \rangle \{-\} \langle 2 \rangle \{-\} \{\sqrt{}\} \rangle \{\wr\}$

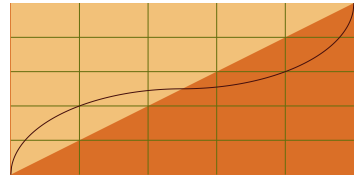
Verlauf



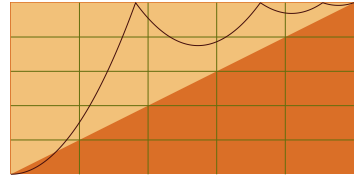
InOutCirc: $\langle\langle 1 \rangle \{\rightleftharpoons\} \{\mapsto\} \langle 0,5 \rangle \{>\} \langle\langle 2 \rangle \{\times\} \langle 2 \rangle \{-\} \{\mapsto\} \{\times\} \{-\} \{\sqrt{}\} \langle 1 \rangle \{+\} \langle 2 \rangle \{\div\} \rangle \rangle \langle\langle 2 \rangle \langle 4 \rangle \{\times\} \{-\} \{\sqrt{}\} \langle 1 \rangle \{-\} \{-2\} \{\div\} \rangle \rangle \{\neg\} \rangle \{\wr\}$



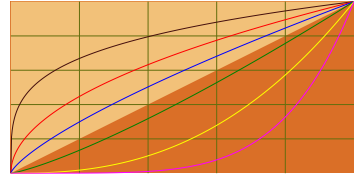
OutInCirc: $\langle\langle \{\mapsto\} \langle 0,5 \rangle \{<\} \langle\langle 0,5 \rangle \langle 1 \rangle \{\neg\} \langle 2 \rangle \{\times\} \langle 1 \rangle \{-\} \langle 2 \rangle \{-\} \{\sqrt{}\} \{\times\} \rangle \rangle \langle\langle -0,5 \rangle \langle 1 \rangle \{\neg\} \langle 2 \rangle \{\times\} \langle 1 \rangle \{-\} \langle 2 \rangle \{-\} \{\sqrt{}\} \langle 2 \rangle \{-\} \{\times\} \rangle \rangle \rangle \{\neg\} \rangle \{\wr\}$



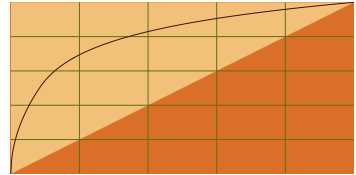
OutBounce: $\langle\langle \{\mapsto\} \{^*p\} \langle 0,36363636 \rangle \{<\} \langle\langle \{^*p\} \langle 2 \rangle \langle 7,5625 \rangle \{\times\} \rangle \rangle \langle\langle \{^*p\} \langle 0,72727272 \rangle \{<\} \langle\langle 0,75 \rangle \{^*p\} \langle 0,54545 \rangle \{-\} \rangle \rangle \langle\langle \{^*p\} \langle 0,90909 \rangle \{<\} \langle\langle 0,9375 \rangle \{^*p\} \langle 0,81818 \rangle \{-\} \rangle \rangle \langle\langle 0,98438 \rangle \{^*p\} \langle 0,95455 \rangle \{-\} \rangle \rangle \rangle \{\neg\} \rangle \{\neg\} \langle 2 \rangle \langle 7,5625 \rangle \{\times\} \{+\} \rangle \{\neg\} \rangle \{\wr\}$



Gammakorrektur¹⁵: $\langle\langle \{^*g\} \{\star\} \rangle \{\text{brt}\} \langle \gamma \rangle \langle g \rangle \{\leftarrow\} \rangle \{\wr\}$



Hybrid logarithmische Gammakorrektur (HLG) gemäß der Empfehlung ITU-R BT.2100-2: $\langle\langle \{\mapsto\} \langle 1 \rangle \langle 12 \rangle \{\div\} \{>\} \langle\langle 0,17883277 \rangle \{^*a\} \langle 1 \rangle \langle 4 \rangle \{^*a\} \{\times\} \{-\} \{^*b\} \langle 0,5 \rangle \{^*a\} \{\mapsto\} \langle 4 \rangle \{\times\} \{\log\} \{\times\} \{-\} \{^*c\} \langle 12 \rangle \{\times\} \{^*b\} \{-\} \{\log\} \{^*a\} \{\times\} \{^*c\} \{+\} \rangle \rangle \rangle \langle\langle 3 \rangle \{\times\} \{\sqrt{}\} \rangle \rangle \{\neg\} \rangle \{\wr\}$

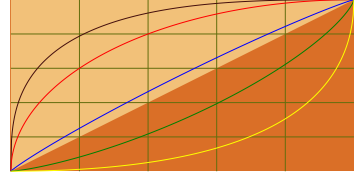


¹⁵Dargestellt sind die Verläufe mit den Werten 0,25 (schwarz) 0,5 (rot) 0,75 (blau) 1,25 (grün) 2,5 (gelb) und 5 (magenta) für γ .

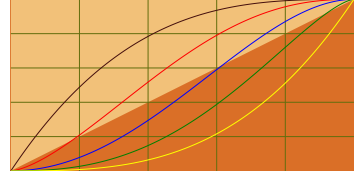
Kode

Superellipse¹⁶: $\langle\langle 1 \rangle \langle 1 \rangle \langle \curvearrowright \rangle \langle - \rangle \langle :e \rangle \langle \star \rangle \langle - \rangle \langle 1 \rangle \langle :e \rangle \langle \div \rangle \langle \star \rangle \rangle \langle \text{brt} \rangle \langle \varepsilon \rangle \langle e \rangle \langle \leftarrow \rangle \langle \text{J} \rangle$

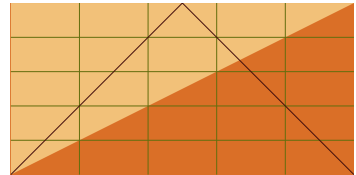
Verlauf



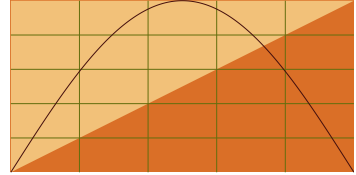
Superelliptische Rampe¹⁷: $\langle\langle 1 \rangle \langle 1 \rangle \langle \curvearrowright \rangle \langle :e \rangle \langle \star \rangle \langle - \rangle \langle :r \rangle \langle \star \rangle \langle - \rangle \rangle \langle \text{brt} \rangle \langle \varepsilon \rangle \langle e \rangle \langle \leftarrow \rangle \langle \rho \rangle \langle r \rangle \langle \leftarrow \rangle \langle \text{J} \rangle$



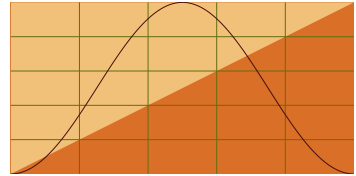
Dreieck: $\langle\langle \langle \mapsto \rangle \langle 0,5 \rangle \langle > \rangle \langle\langle 0,5 \rangle \langle - \rangle \langle 2 \rangle \langle \times \rangle \langle 1 \rangle \langle \neq \rangle \langle - \rangle \rangle \langle\langle 2 \rangle \langle \times \rangle \rangle \langle \text{J} \rangle$



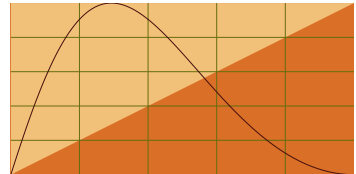
Halber Sinus: $\langle\langle 180 \rangle \langle \times \rangle \langle \sin \rangle \rangle \langle \text{J} \rangle$



Voller Sinus: $\langle\langle 0,25 \rangle \langle - \rangle \langle 360 \rangle \langle \times \rangle \langle \sin \rangle \langle 1 \rangle \langle + \rangle \langle 2 \rangle \langle \div \rangle \rangle \langle \text{J} \rangle$



Kombination aus halbem Sinus und Out-Quad: $\langle\langle (*a) \langle 2 \rangle \langle :a \rangle \langle - \rangle \langle .a \rangle \langle 180 \rangle \langle \times \rangle \langle \times \rangle \langle \sin \rangle \rangle \langle \text{J} \rangle$



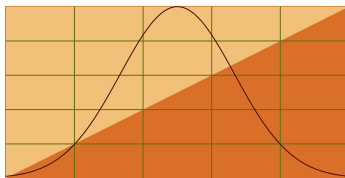
¹⁶[15, S. 104]; dargestellt sind die Verläufe mit den Werten 3 (schwarz) 2 (rot) 1,1 (blau) 0,75 (grün) und 0,5 (gelb) für ε .

¹⁷[15, S. 104]; dargestellt sind die Verläufe mit den Wertepaaren 1+3 (schwarz) 1,5+2,5 (rot) 2+2 (blau) 2,5+1,5 (grün) und 3+1 (gelb) für ε und ρ .

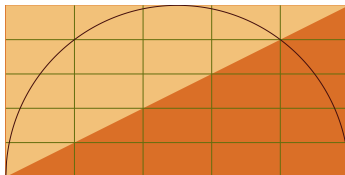
Kode

Glockenkurve¹⁸: $\langle\langle 0,5 \rangle (-) (2) (_) \langle 18 \rangle (\times) \langle e \rangle (\rightleftharpoons) (\star) \rangle \langle \uparrow \rangle$

Verlauf



Halbkreis: $\langle\langle (\mapsto) \langle 0,5 \rangle (>) \langle\langle 0,5 \rangle (-) \langle 2 \rangle (\times) \langle 1 \rangle (\rightleftharpoons) (-) \rangle \rangle \langle\langle 2 \rangle (\times) \rangle (_) \langle 1 \rangle (\rightleftharpoons) \langle 1 \rangle (-) (2) (-) (\surd) \rangle \langle \uparrow \rangle$



$\langle \uparrow \rangle$

progress

generiere.Verlauf

(Strg+ \hat{u} +u2934)
(Englisch)

$\langle\langle Funktion_{Af} \rangle \rangle (\langle Ausstattung_{Rg} \rangle) (\langle \uparrow | prg | generiere.Verlauf \rangle):$
 $\langle Verlauf_{vl} \rangle$

Macht aus einer Anführung und gegebenenfalls einem Regal einen Verlauf.

$\langle x_{Zl}^{[0-1]} \rangle \langle\langle Funktion_{Af} \rangle \rangle \langle \uparrow \rangle: \langle y_{Zl}^{[0-1]} \rangle$

Vl.valoro

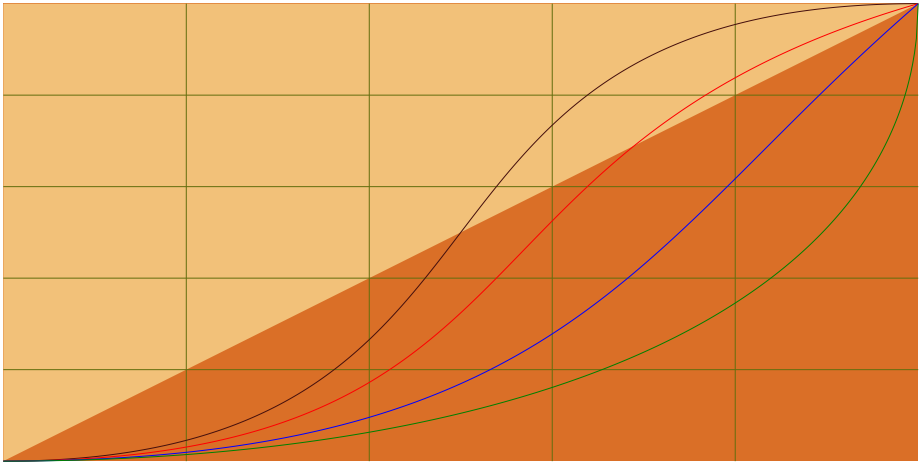
gib.Einzelwert

(Esperanto)

$[\langle Verlauf_{vl} \rangle^{vl}] \langle x_Q^{[0-1]} \rangle (Vl.vlr | gib.Einzelwert): \langle y_{Zl}^{[0-1]} \rangle [\langle Verlauf_{vl} \rangle^{vl}]$

Errechnet mit dem registrierten Verlauf zu einem vorgegebenen den dazugehörigen Wert.

¹⁸Über den Wert hinter der Umkehrung kann die Ausdehnung der Glockenkurve verändert werden (beginnt und endet generell nicht bei Null); durch Abziehen von 1 statt 0,5 am Anfang wird ein rein aufsteigender Verlauf daraus.



Vier kubische Bezierkurven mit einem Ausgangswinkel von durchgängig 90 Grad und Eingangswinkeln von 90, 60, 30 und 0 Grad (schwarz, rot, blau, grün)

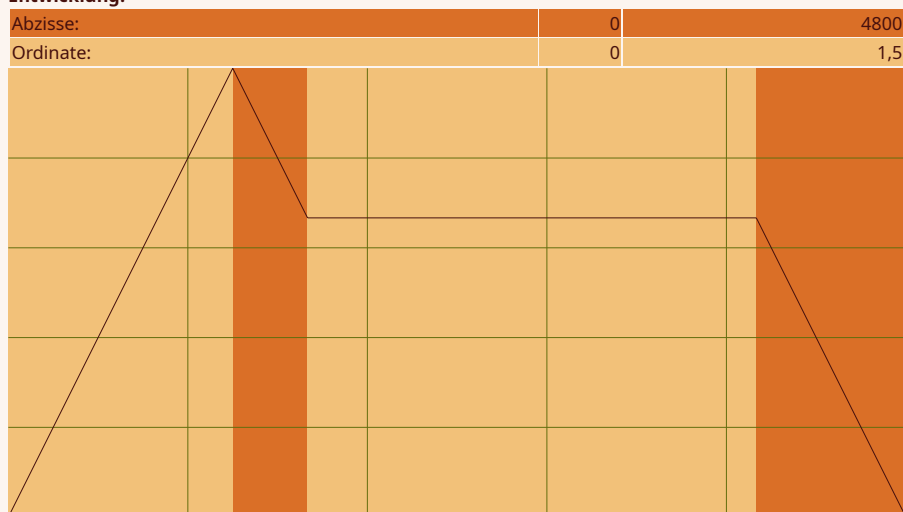
bezier interpoliere.kubisch

$\langle x_{\mathbb{Q}}^{[0-1]} \rangle \langle \text{Ausgangswinkel}_{\mathbb{Q}}^{[0-90]} \rangle \langle \text{Eingangswinkel}_{\mathbb{Q}}^{[0-90]} \rangle$ (bzw |
interpoliere.kubisch): $\langle y_{\mathbb{Zl}}^{[0-1]} \rangle$

Konstruiert aus zwei Winkelangaben (in Grad) die Kontrollpunkte einer kubischen Bézierkurve. Die Abstände der Kontrollpunkte vom Ursprung und Ende der Kurve sind vorgegeben und betragen jeweils $\Phi - 1$. Sowohl Aus- (unten links) als auch Eingangswinkel (oben rechts) bemessen sich von der Senkrechten im Uhrzeigersinn. Zu der vorgegebenen horizontalen Koordinate wird die entsprechende vertikale errechnet. Eine horizontale Positionsangabe kleiner als Null wird auf diesen Wert gesetzt; eine Positionsangabe größer als Eins wird entsprechend gekürzt.

Verarbeitung des Skriptes *entwicklung.lysa*: $0 \cdot 1200 \cdot 0 \cdot 1,5 \cdot \text{Linear (Verlauf)} \cdot \text{Entwicklung generieren} \cdot 1600 \cdot 1 \cdot \text{Linear (Verlauf)} \cdot \text{Entwicklung erweitern} \cdot 4000 \cdot 1 \cdot \text{Linear (Verlauf)} \cdot \text{Entwicklung erweitern} \cdot 4800 \cdot 0 \cdot \text{Linear (Verlauf)} \cdot \text{Entwicklung erweitern} \cdot \text{zeigen}$

Entwicklung:



Die Entwicklung stellt eine Erweiterung des Verlaufs dar. Sie arbeitet sowohl auf der Abzisse als auch der Ordinate mit beliebigen Werten. Zudem können Entwicklungen beliebig oft erweitert werden. Das ändert nichts an ihrer Immutabilität, denn im Falle einer Erweiterung wird eine neue Entwicklung als Anschluß an die bestehende angehängt. Gehandhabt wird das gesamte Konstrukt aber immer als eine einzige Entwicklung, bei welcher die erste maßgeblich ist. Es muß lediglich die Richtung der Wertentwicklung der Abzisse beibehalten werden. Zu einem beliebigen Wert liefert eine Entwicklung einen Wert. Liegt dieser Wert außerhalb des vorgegebenen Intervalls wird entweder der Start- oder Endwert zurückgegeben. Bei innerhalb des gegebenenfalls erweiterten Intervalls liegenden Werten wird mit den übergebenen Verläufen ein entsprechender Wert berechnet.

↪ nein • \mathbb{Q} ungültig • \mathbb{Z} ungültig • \mathbb{R} ungültig

↷ (u219d) 90	etd 90
↷ (u21dd) 91	Ew.stt 90
dev 91	generiere.Entwicklung 91
erfrage 90	trs 91
erweitere.Entwicklung 90	übertrage 91

Ew.stato
erfrage

(Esperanto)

$$\left[\langle \text{Entwicklung}_{Ew} \rangle^{Ew} \right] \langle \text{Position}_Q \rangle (\text{Ew.stt} \mid \text{erfrage}): \langle \text{Wert}_{Zl} \rangle$$

$$\left[\langle \text{Entwicklung}_{Ew} \rangle^{Ew} \right]$$

Ermittelt zu einer gegebenen Position einen Wert aus der registrierten Entwicklung. Bei einer Position außerhalb des Intervalls der Entwicklung wird entweder der Start- oder der Zielwert ausgegeben.

↷
etendi
erweitere.Entwicklung

(Strg+⇧+u219d)
(Esperanto)

$$\langle \text{Entwicklung}_{Ew} \rangle \langle \text{Ende}_Q \rangle \langle \text{Zielwert}_Q \rangle \langle \text{Fortgang}_{vl} \rangle \{ \sim \mid \text{etd} \mid$$

$$\text{erweitere.Entwicklung} \}: \langle \text{Entwicklung}_{Ew} \rangle$$

Verlängert eine Entwicklung. Deren bisheriger Ende und Zielwert werden als Anfang und Startwert der Erweiterung genommen. Die Erweiterung muß in der Abzisse dieselbe Richtung der Wertentwicklung aufweisen.



development
generiere.Entwicklung

(Strg+⬆+u21dd)
 (Englisch)

$\langle \text{Anfang}_Q \rangle \langle \text{Ende}_Q \rangle \langle \text{Startwert}_Q \rangle \langle \text{Zielwert}_Q \rangle \langle \text{Fortgang}_{vi} \rangle (\rightsquigarrow \mid \text{dev} \mid$
 $\text{generiere.Entwicklung}): \langle \text{Entwicklung}_{Ew} \rangle$

Generiert aus den angegebenen Werten eine Entwicklung. Zu beachten ist lediglich, daß Anfang und Ende nicht denselben Wert haben dürfen. Je nachdem, ob der Anfang oder das Ende größer ist, wird die Richtung für die Wertentwicklung auch für eventuelle Erweiterungen festgelegt.

Start- und Zielwert unterliegen keinerlei Beschränkungen.

transigi
übertrage

(Esperanto)


$\langle \text{Muster}_{Ew} \rangle \langle \text{Anfang}_Q \rangle \langle \text{Ende}_Q \rangle (\text{trs} \mid \text{übertrage}): \langle \text{Entwicklung}_{Ew} \rangle$

Nimmt eine Entwicklung (einschließlich Erweiterungen) als Muster und erstellt dazu eine neue mit einer den angegebenen Werten entsprechenden Abzisse. Die Richtung der Entwicklung kann sich dabei umkehren; die Verhältnisse der Glieder einer erweiterten Entwicklung zueinander bleiben erhalten.

Ein Zeitraum repräsentiert eine Zeitspanne in einem zu erstellen- den Video. Generiert wird er aus der Angabe eines Anfangs- und eines Endzeitpunktes. Diese sind anzugeben in 300 stel-Sekunden (genannt lakonische¹⁹ Sekundenbruchteile, abgekürzt Λ) seit dem Anfang des zu erstellenden Videos (respektive des zu erstellenden Abschnitts daraus). Es sind 300 Sekundenbruchteile, da dies das kleinste gemeinsame Vielfache (kgV) von 50 und 60 ist, welche zusammen mit 25 und 30 die hauptsächlichen Bildwiedergabe- frequenzen bei Videos sind. Bei all diesen Frequenzen ergibt sich dadurch eine ganzzahlige Anzahl von lakonischen Sekundenbruch- teilen pro Einzelbild.

Die Hauptaufgabe dieser Gegebenheitsart ist die Umrechnung lakonischer Sekundenbruchteile in Nummern zu generierender Einzelbilder (rechtsoffenes Intervall). Letztere hängen davon ab, wieviele Bilder in der Sekunde dargestellt werden sollen, erstere sind die davon unabhängigen Angaben.

↪ nein • \mathbb{Q} Dauer (Λ) • \mathbb{Z} Einzelbilder • ∇ Anfang (untereinander)

	(u23f2)	93	pks	94
bestimme.Bildfrequenz		93	prd	93
fps		93		
generiere.Zeitraum		93	verschiebe	94

Folgendes Pronomen ist in Zusammenhang mit Zeiträumen im globalen Lexikon gespeichert:

¹⁹Lakonisch leitet sich von der Landschaft Lakonien im Süden des Peloponnes ab, in welcher der Stadtstaat Sparta lag, dessen Einwohner sich auch Lakedaimonier nannten und oft ein Lambda auf ihren Schilden trugen. In der Schlacht bei den Thermopylen (480 v. Chr.) sollen 300 Spartaner (mit weiteren Griechen) das Heer der Perser einige Tage lang aufgehalten haben.

fps	Wieviele Bilder pro Sekunde in einem zu erstellenden Video zu generieren sind. Voreingestellt sind 25 Bilder pro Sekunde (frames per second). Eine Änderung des Pronomens ist bestenfalls wirkungslos. Verändert werden kann und soll der intern verwendete Wert nur über die dafür vorgesehenen Verben. Dies sollte frühzeitig geschehen.
-----	--

frames per second

(Englisch)

bestimme.Bildfrequenz

$\langle \text{Frequenz}_{\mathbb{Q}}^{[0-300]} \rangle (\text{fps} \mid \text{bestimme.Bildfrequenz}): \{ \langle \text{Bildfrequenz} \rangle^{\text{fps}} \}^G$

Setzt die interne Variable für die Bildfrequenz. Zulässig sind Werte größer als Null und maximal 300. Voreingestellt sind 25.

Der Wert des entsprechenden Pronomens wird aktualisiert, damit dieser grundlegende Wert aus dem Skript heraus abgefragt werden kann. Zwar kann der Wert des Pronomens verändert werden, doch dies ändert nichts an den internen Berechnungen. Der dafür maßgebliche Wert kann ausschließlich über dieses Verb verändert werden.



period

(Strg+⬆+u23f2)

(Englisch)

generiere.Zeitraum

$\langle \text{Anfang}_{\mathbb{Q}}^{\Lambda} \rangle \langle \text{Ende}_{\mathbb{Q}}^{\Lambda} \rangle (\text{⌚} \mid \text{prd} \mid \text{generiere.Zeitraum}): \langle \text{Zeitraum}_{\text{zr}} \rangle$

Erstellt aus dem Anfangs- und Endzeitpunkt in lakonischen Sekundenbruchteilen (Λ) einen Zeitraum und legt diesen auf den Stapel.

Es darf sich weder um einen leeren noch um einen negativen Zeitraum handeln.

$\langle \text{Zeitraum}_{\text{Zr}} \rangle \langle \text{Verschiebung}_{\text{Q}}^{\Lambda} \rangle (\text{pks} \mid \text{verschiebe}): \langle \text{Zeitraum}_{\text{Zr}} \rangle$

Generiert einen neuen Zeitraum, indem es einen vorhandenen verschiebt.

Untertitel

ut

Ein Untertitel repräsentiert eine einzelne Untertitelzeile in einem Video. Sie werden gesammelt und in einer eigenen Datei abgelegt (WebVTT).

↔ nein • \mathbb{Q} ungültig • \mathbb{Z} ungültig • \mathbb{A} Anzeigebeginn

end	95	stk	95
generiere.Untertitel	95	transkribiere	95

subteksto

(Esperanto)

generiere.Untertitel

$\langle \text{Anzeige}_{zr} \rangle \langle \text{Zeile}_{zk} \rangle (\text{stk} \mid \text{generiere.Untertitel}): \langle \text{Untertitel}_{ut} \rangle$

Erstellt einen Untertitel und legt ihn auf den Stapel.

endosierigi

(Esperanto)

transkribiere

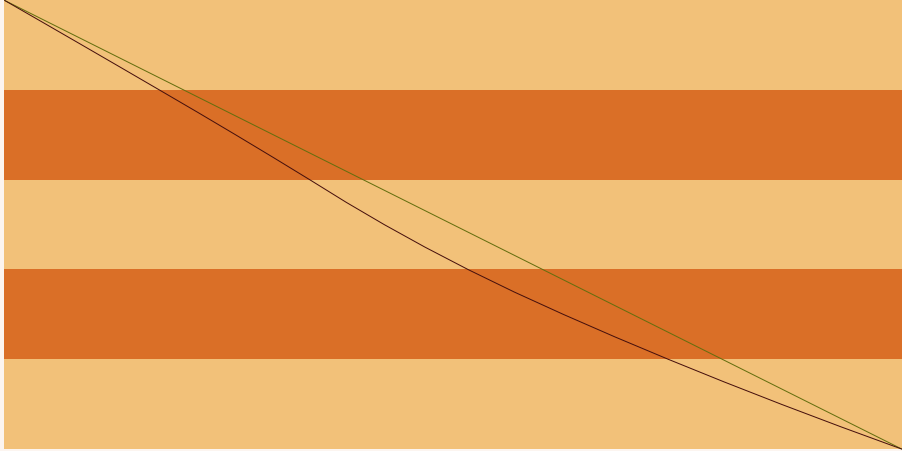
$\langle \text{Untertitel}_{sl}^{ut} \rangle \langle \text{Dateiname}_{zk} \rangle (\text{end} \mid \text{transkribiere})$

Speichert die gesammelten Untertitel in einer **WebVTT**-Datei. Der Dateiname hat den kompletten Pfad zu enthalten (Dateiendung: vtt). Die Sammlung wird vor dem Speichern sortiert.

Insofern bei einem Untertitel eine zeitliche Überlappung festgestellt wird, bricht die Erstellung der Datei mit einer Fehlermeldung ab.

Verarbeitung des Skriptes *taktgeber.lysa*: 25 • 5 • 4 • 1 • 2 • 130 • 132 • *Linear* (Verlauf) • **Entwicklung generieren** • 3 • 140 • *Linear* (Verlauf) • **Entwicklung erweitern** • 4 • 110 • *Linear* (Verlauf) • **Entwicklung erweitern** • 6 • 80 • **Anfang** • *Q.Z* • **potenzieren** • **Ende** • **Verlauf generieren** • **Entwicklung erweitern** • **Taktgeber generieren** • **zeigen**

Taktgeber: 25-3210A, 5 Takte zu 4 Schlägen



Ein Taktgeber unterteilt die Zeit in gleichabständige oder auch unterschiedlich lange Abschnitte (Takte). Das Tempo, welches die Länge jedes einzelnen Taktes bestimmt, wird durch eine Entwicklung vorgegeben.

⇒ nein • Q ungültig • Z ungültig • Δ ungültig

generiere.Taktgeber	96	taktiere	97
mtn	96	Tg.dir	97

metronomo

(Esperanto)

generiere.Taktgeber

$\langle \text{Versatz}_{\text{Q}}^{\Delta} \rangle \langle \text{Takte}_{\text{Z}} \rangle \langle \text{Schläge pro Takt}_{\text{Z}} \rangle \langle \text{Tempo}_{\text{Ew}}^{\text{bpm}} \rangle (\text{mtn} |$
 $\text{generiere.Taktgeber}): \langle \text{Dirigent}_{\text{Tg}} \rangle$

Generiert einen Taktgeber. Zu übergeben sind ein Versatz in lakonischen Sekundenbruchteilen bis zum Beginn des ersten Taktes,

die Gesamtlänge in Takten, die Schläge pro Takt und eine Entwicklung, wo wieviele Schläge pro Minute zu spielen sind (beginnend mit dem ersten Takt).

Tg.direkti
taktiere

(Esperanto)

$$\left[\langle \text{Dirigent}_{\text{Tg}} \rangle^{\text{Tg}} \right] \langle \text{Takt}_Z \rangle \langle \text{Schlag}_Q \rangle \langle \text{Dauer}_Q \rangle (\text{Tg.dir} \mid \text{taktiere}):$$

$$\langle \text{Beginn}_{\text{Zl}}^\Lambda \rangle \left(\langle \text{Dauer}_{\text{Zl}}^\Lambda \rangle \right) \left[\langle \text{Dirigent}_{\text{Tg}} \rangle^{\text{Tg}} \right]$$

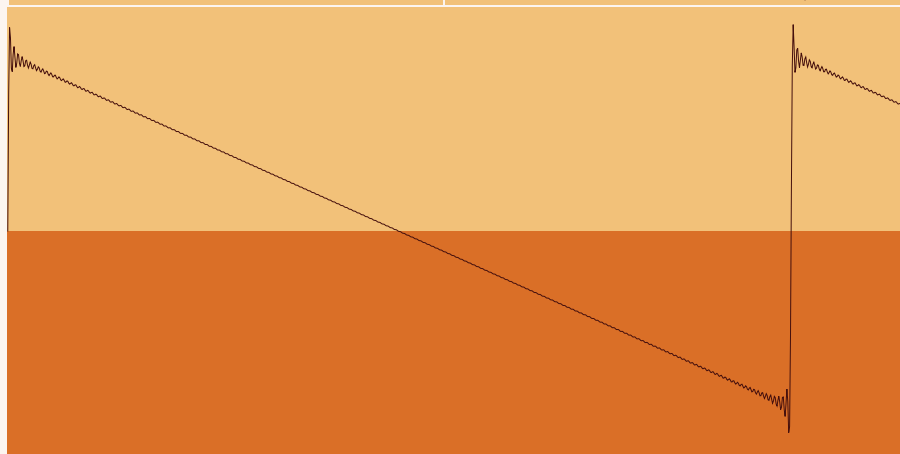
Ermittelt mit dem registrierten Taktgeber einen Zeitpunkt in lakonischen Sekundenbruchteilen anhand der Nummer eines Taktes (wird gerundet) und eines Schlages innerhalb dieses Taktes. Wenn die Dauer (in Schlägen) größer als Null ist, wird diese zudem in lakonischen Sekundenbruchteilen zurückgegeben.

Eine Fehlermeldung erfolgt, wenn die gesuchte Position oder auch die gesuchte Dauer außerhalb des Taktgebers liegen.

Verarbeitung des Skriptes *klang.lysa*: Anfang • 180 • multiplizieren • Sinus berechnen • Ende • Verlauf generieren • in v deponieren • 1000 • Stille erschaffen • registrieren • 0 • gänzen • registrieren • Anfang • raufzählen • 190 • Ganzzahl verwenden • auf Steigerung prüfen • verlassen • 0 • 1000 • 110 • Ganzzahl verwenden • multiplizieren • 0,5 • Ganzzahl verwenden • dividieren • aus v verwenden • duplizieren • oszillieren • bejahren • wiederholen • Ende • vereigentlichen • Klang entnehmen • zeigen

Klang:

Abtastwerte:	1000
Bilder:	0,260416667
Maximum:	0,921831528
Minimum:	-0,896430123



Klang repräsentiert einen einzelnen Klangkanal (beliebig viele davon lassen sich schließlich zu einem Stereo-Raumklang zusammenfassen). Die zeitliche Auflösung ist auf 96 kHz festgelegt (pro lakonischem Sekundenbruchteil ergibt das 320 Abtastwerte). Abtastwerte werden als doppelt genaue Fließkommazahlen (64 Bit) gespeichert (**LPCM**).

Ein Klang gibt seine Länge, also die Anzahl der in ihm enthaltenen Abtastwerte als Wert zurück und ist darüber auch vergleichbar.

Bei einer Länge bis einschließlich tausend Abtastwerten wird deren Verlauf dargestellt (siehe oben), bei längeren Klängen ein Frequenzspektrum.

↪ ja • \mathbb{Q} Abtastwerte • \mathbb{Z} Abtastwerte • \mathbb{R} Wert

—	(u2014)	99	lds	100
~	(u223f)	102	moduliere	101
erschaffe.Stille		99	noi	103
exportiere.Klang		99	normalisiere.Klang	102
freq		100	oszilliere	102
glitsche		100	rausche	103
importiere.Klang		100	rev	101
interpoliere.Frequenzen		100	schneide.aus	103
kehre.Richtung.um		101	sil	99
Kl.amp		104	svs	99
Kl.cut		103	überlagere	103
Kl.mdl		101	verstärke	104
Kl.mix		103		
Kl.osc		102		
Kl.pkn		102		
Kl.sld		100		

— (Strg+⇧+u2014)
silence (Englisch)
erschaffe.Stille

$\langle \text{Abtastwerte}_Z \rangle$ (— | sil | erschaffe.Stille): $\langle \text{Klang}_{kl} \rangle$

Erschafft einen stillen Klang der angegebenen Länge (Anzahl Abtastwerte, 96 kHz).

save sound (Englisch)
exportiere.Klang

$\langle \text{Klang}_{kl} \rangle$ $\langle \text{Dateiname}_{zk} \rangle$ (svs | exportiere.Klang)

Speichert einen Klang als **Au**-Datei. Die Details sind beim Klangimport beschrieben (s. u.).

Insofern der angegebene Dateiname nicht auf .au endet, wird die Endung automatisch hinzugefügt.

$$\left[\langle \text{Klang}_{kl} \rangle^{kl} \right] \langle \text{Start}_Z^{\geq 0} \rangle \langle \text{Länge}_Z^{\geq 0} \rangle \langle \text{Frequenzgang}_{Ew} \rangle \langle \text{Amplitude}_{Ew}^{[0-1]} \rangle \\ \langle \text{Kamm}_{vl} \rangle \langle \text{Tal}_{vl} \rangle (\text{Kl.sld} \mid \text{glitsche}): \left[\langle \text{Klang}_{kl} \rangle^{kl} \right]$$

Mischt eine Welle variierender Frequenz (Glissando) und veränderlicher Amplitude zum registrierten Klang. Für Wellenkämme und Wellentäler sind jeweils eigene Verläufe anzugeben.

Sowohl die Entwicklung für den Frequenzgang als auch für die Amplitude werden an die Start- und Längenangaben angepaßt.

Es wird nicht kontrolliert, ob eine Frequenz höher als 21 kHz erstellt werden soll. Dies ist zu vermeiden.

load sound
importiere.Klang

(Englisch)

$$\langle \text{Dateiname}_{zk} \rangle (\text{lds} \mid \text{importiere.Klang}): \langle \text{Klang}_{kl} \rangle$$

Lädt einen Klang aus einer Datei. Der Dateiname muß den kompletten Zugriffspfad enthalten. Es muß sich um eine **Au**-Datei handeln mit einem einzelnen Kanal, einer Abtastrate von 96 kHz und Werten, die als Gleitkommazahlen (64 Bit) gespeichert sind (Kodierung: 7). Eine fehlende Endung wird automatisch angehängt (.au).

frequencies
interpoliere.Frequenzen

(Englisch)

$$\langle \text{Ton}_{kl} \rangle \langle \text{Anzahl}_Z \rangle \langle \text{Abweichung}_Q^{dB} \rangle (\text{frq} \mid \text{interpoliere.Frequenzen}): \\ \langle \text{Frequenzen}_{SI}^{ZI} \rangle \langle \text{Verhältnisse}_{SI}^{ZI} \rangle$$

Führt eine Fouriertransformation an einem Klang aus, interpoliert die energiereichsten Frequenzen und legt diese ebenso wie die Verhältnisse der Frequenzen zueinander in Sammlungen ab.

Zudem werden die Informationen zu den Frequenzen in die HTML-Datei ausgegeben.

Es wird davon ausgegangen, daß der Klang einen einzelnen Ton enthält. Die anzugebende Anzahl stellt das Maximum der zurückgegebenen Frequenzen dar. Die Abweichung (in dB, negativ) begrenzt die zurückgegebenen Frequenzen vom Energieniveau her.

Um eine lineare Amplitude in Dezibel als Abstand zum Maximum (negativ) umzurechnen, könnte solch eine Formulierung verwendet werden²⁰:

$$\lfloor \text{Amplitude}_{\text{Q}} \rfloor \langle \text{dB} \rangle \langle \langle (\text{dlg}) \langle 20 \rangle (\times) \rangle \rangle (!): \lfloor \text{Abweichung}_{\text{Zl}}^{[0 \dots 90]} \rfloor$$

reverse

(Englisch)

kehre.Richtung.um

$$\langle \text{Original}_{\text{kl}} \rangle (\text{rev} \mid \text{kehre.Richtung.um}): \langle \text{Umkehrung}_{\text{kl}} \rangle$$

Erstellt die Kopie eines Klangs, bei dem die Reihenfolge der Abtastwerte umgekehrt ist.

Kl.moduli

(Esperanto)

moduliere

$$\left[\langle \text{Klang}_{\text{kl}} \rangle^{\text{kl}} \right] \langle \text{Modulator}_{\text{kl}} \rangle (\text{Kl.mdl} \mid \text{moduliere}): \left[\langle \text{Klang}_{\text{kl}} \rangle^{\text{kl}} \right]$$

Multipliziert den registrierten Klang abtastwerteweise mit einem Modulator. Dies entspricht einer Amplitudenmodulation.

Insofern der registrierte Klang länger als der Modulator ist, wird mit diesem immer wieder von vorne begonnen, bis das Ende des registrierten Klangs erreicht ist und dieser vollständig moduliert wurde (Ringmodulation). Es genügt also, wenn der Modulator nicht länger als die Modulationsschwingung ist (oder ein

²⁰nach [2, S. 210]

entsprechendes Vielfaches davon um auf eine ganzzahlige Länge zu kommen). Allerdings darf der Modulator nicht leer sein.

peak normalization normalisiere.Klang

(Englisch)

$\left[\langle \text{Klang}_{kl} \rangle^{kl} \right] \langle \text{Maximum}_{\mathbb{Q}} \rangle (\text{pkn} \mid \text{normalisiere.Klang}): \left[\langle \text{Klang}_{kl} \rangle^{kl} \right]$

Normalisiert einen Klang auf das angegebene Maximum.

Um eine Angabe in Dezibel als Abstand zum Maximum (negativ) in eine lineare Amplitude umzurechnen, könnte solch eine Formulierung verwendet werden²¹:

$\lfloor \text{Abweichung}_{\mathbb{Q}}^{[0 \dots -90]} \rfloor \langle \text{LPMC} \rangle \langle \langle 10 \rangle (\rightleftharpoons) \langle 20 \rangle (\div) (\star) \rangle \langle ! \rangle: \lfloor \text{Amplitude}_{\text{de}_{\text{Z}}} \rfloor$

~ Kl.oscili oszilliere

(Strg+⬆+u223f)
(Esperanto)

$\left[\langle \text{Klang}_{kl} \rangle^{kl} \right] \langle \text{Start}_{\mathbb{Z}}^{\geq 0} \rangle \langle \text{Länge}_{\mathbb{Z}}^{\geq 0} \rangle \langle \text{Frequenz}_{\mathbb{Q}}^{[0 \dots 21000]} \rangle \langle \text{Amplitude}_{\mathbb{Q}}^{[0 \dots 1]} \rangle$
 $\langle \text{Kamm}_{vl} \rangle \langle \text{Tal}_{vl} \rangle (\sim \mid \text{Kl.osc} \mid \text{oszilliere}): \left[\langle \text{Klang}_{kl} \rangle^{kl} \right]$

Mischt eine Welle der angegebenen Frequenz mit der angegebenen Amplitude zum registrierten Klang. Für Wellenkämme und Wellentäler sind jeweils eigene Verläufe anzugeben.

21 kHz liegen weit genug unterhalb der **Nyquist-Frequenz** und oberhalb der höchsten normalerweise von Menschen **wahrnehmbaren Frequenz**.

²¹nach [2, S. 210]

$\langle \text{Länge}_Z^{\geq 0} \rangle \langle \text{Art}_{zk} \rangle (\text{noi} \mid \text{rausche}): \langle \text{Rauschen}_{kl} \rangle$

Erschafft einen Klang (Amplitude $\pm 0,75$) der angegebenen Länge (Anzahl Abtastwerte, 96 kHz) mit Rauschen. Dessen Art kann durch die Angabe einer Farbe bestimmt werden:

weiß Dieses Rauschen wird aus Zufallszahlen generiert und stellt eine energetisch gleichmäßigen Mischung aller Frequenzen unterhalb der Nyquist-Frequenz dar. Sie wird entsprechend durch einen einfachen Tiefpaßfilter begrenzt.

rosa Bei diesem Rauschen nimmt die Energie ab, je höher die Frequenz ist (-3 dB per Verdoppelung).²² Für menschliche Hörer ergibt sich dadurch allerdings der Eindruck, alle Frequenzen gleichmäßig zu hören.

$[\langle \text{Klang}_{kl} \rangle^{kl}] \langle \text{Start}_Z^{\geq 0} \rangle \langle \text{Länge}_Z^{\geq 0} \rangle (\text{Kl.cut} \mid \text{schneide.aus}): \langle \text{Ausschnitt}_{kl} \rangle$
 $[\langle \text{Klang}_{kl} \rangle^{kl}]$

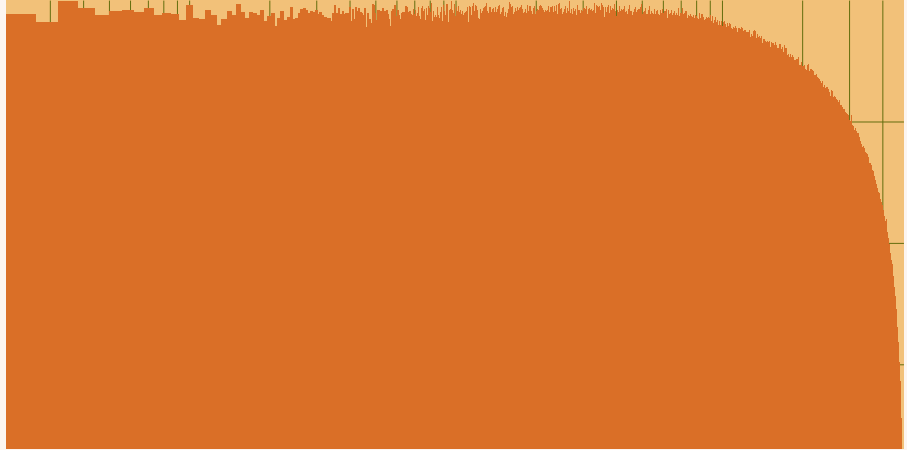
Erstellt einen neuen Klang der angegebenen Länge aus einem Abschnitt eines anderen Klangs. Liegt der angegebene Abschnitt oder ein Teil dessen außerhalb des registrierten Klangs, so enthält der neue Klang an der entsprechenden Stelle Stille.

$[\langle \text{Klang}_{kl} \rangle^{kl}] \langle \text{Schnipsel}_{kl} \rangle \langle \text{Position}_Z \rangle (\text{Kl.mix} \mid \text{überlagere}):$
 $[\langle \text{Klang}_{kl} \rangle^{kl}]$

²²nach cooperbaker.com

Klang:

Abtastwerte:	1920000
Bilder:	500
Maximum:	0,75
Minimum:	-0,749381449



Simple tiefgepaßtes weißes Rauschen (dient auch als Ausgangsmaterial für die Darstellung von Klangfiltern)

Überlagert den registrierten Klang mit einem anderen. Die Abtastwerte der verschiedenen Klänge werden dabei jeweils summiert.

Liegt die angegebene Position außerhalb des registrierten Klangs oder ragt der zu überlagernde Klang darüber hinaus, werden die außerhalb des registrierten Klangs liegenden Teile ignoriert.

Kl.amplify
verstärke

(Englisch)

$$[\langle \text{Klang}_{kl} \rangle^{kl}] \langle \text{Faktor}_Q \rangle (\text{amp} \mid \text{verstärke}): [\langle \text{Klang}_{kl} \rangle^{kl}]$$

Multipliziert alle Abtastwerte des registrierten Klangs mit dem angegebenen Faktor. Ein Faktor größer als Eins führt zu einer größeren, ein Faktor kleiner als Eins zu einer niedrigeren Lautstärke.

Durch das Verstärken können Abtastwerte außerhalb des zulässigen Bereichs (± 1) entstehen. Diese werden nicht automatisch korrigiert.

Verarbeitung des Skriptes *klangfilter.lysa*: 18000 • 96000 • dividieren • 360 • multiplizieren • duplizieren • in a deponieren • Sinus berechnen • 2,9 • dividieren • in b deponieren • aus a entnehmen • Cosinus berechnen • in a deponieren • Sammlung erschaffen • 1 • aus a verwenden • subtrahieren • 2 • dividieren • duplizieren • 1 • aus a verwenden • subtrahieren • vertauschen • einsammeln • Sammlung erschaffen • 1 • aus b verwenden • addieren • -2 • aus a entnehmen • multiplizieren • 1 • aus b entnehmen • subtrahieren • einsammeln • Klangfilter generieren • 'tiefpaß' • befürworten • 1920000 • 'weiß' • rauschen • registrieren • Anfang • tiefpaß • filtern • Ende • 8 • iterieren • 0,75 • Klang normalisieren • Klang entnehmen • zeigen

Klang:

Abtastwerte:	1920000
Bilder:	500
Maximum:	0,747756401
Minimum:	-0,75



Repräsentiert einen Digitalfilter sowohl für endliche (FIR) als auch unendliche Impulsantworten (IIR).

Das Filtern findet in der ersten Direktform²³ statt. Der erste Wert der subtraktiven Koeffizienten dient ausschließlich der Normierung aller Koeffizienten:

$$a_n = \frac{p_0}{m_0} e_n + \frac{p_1}{m_0} e_{n-1} + \dots + \frac{p_x}{m_0} e_{n-x} - \frac{m_1}{m_0} a_{n-1} - \dots - \frac{m_y}{m_0} a_{n-y}$$

Bestehen die subtraktiven Koeffizienten lediglich aus dem Normierungswert, handelt es sich um einen Digitalfilter mit endlicher Impulsantwort.

²³[18, S. 200 ff.]

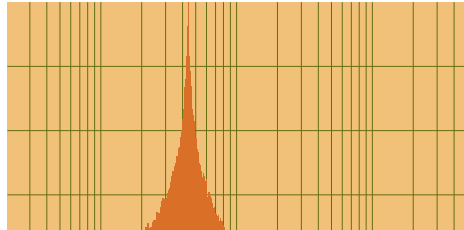
fil	108	Kl.mmf	108
filtere	108	generiere.Klangfilter	108

Rezepte für verschiedene Filter (Biquad) hat **Robert Bristow-Johnson** zusammengestellt.

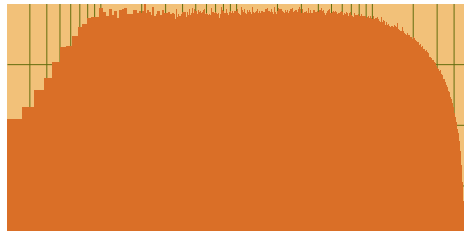
Filter

Biquad-Bandpaßfilter: $\langle 440 \rangle \langle 96000 \rangle (\div)$
 $\langle 360 \rangle (\times) (\mapsto) (*a), (\sin) \langle 100 \rangle (\div) (*b), (\text{klk})$
 $(:b) (0) (:b) (-) (\ni) (\text{klk}) \langle 1,0 \rangle (:b) (+) (-2) (.a)$
 $(\cos) (\times) (1) (.b) (-) (\ni) (\text{fil})$

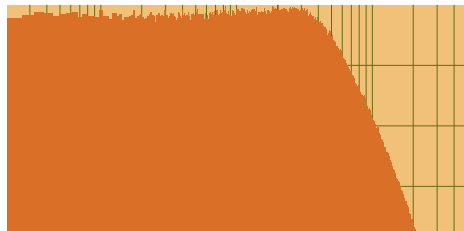
Wirkung



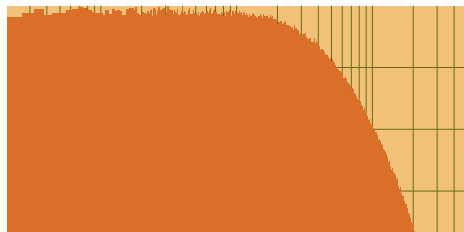
Biquad-Hochpaßfilter: $\langle 80 \rangle \langle 96000 \rangle (\div)$
 $\langle 360 \rangle (\times) (*a) (:a) (\sin) \langle 1,75 \rangle (\div) (*b) (.a)$
 $(\cos) (*a) (\text{klk}) (1) (:a) (+) (2) (\div) (\mapsto) (1) (:a)$
 $(+) (-) (\ni) (\ni) (\text{klk}) (1) (:b) (+) (-2) (.a) (\times)$
 $(1) (.b) (-) (\ni) (\text{fil})$



Biquad-Tiefpaßfilter: $\langle 4000 \rangle \langle 96000 \rangle (\div)$
 $\langle 360 \rangle (\times) (\mapsto) (*a) (\sin) \langle 1,75 \rangle (\div) (*b) (.a)$
 $(\cos) (*a) (\text{klk}) (1) (:a) (-) (2) (\div) (\mapsto) (1) (:a)$
 $(-) (\ni) (\ni) (\text{klk}) (1) (:b) (+) (-2) (.a) (\times) (1)$
 $(.b) (-) (\ni) (\text{fil})$



Butterworth Tiefpaßfilter zweiter Ordnung²⁴: $\langle 4000 \rangle \langle 180 \rangle (\times) \langle 96000 \rangle (\div) (\tan) (1)$
 $(\ni) (\div) (*a) (1) (1) (:a) (2) (\times) (:a) (2) (+) (+)$
 $(\div) (*b) (\text{klk}) (:b) (\mapsto) (2) (\times) (:b) (\ni) (\text{klk})$
 $\langle 1,0 \rangle (2) (:b) (\times) (1) (:a) (2) (-) (\times) (.b) (1) (2)$
 $(:a) (\times) (-) (.a) (2) (+) (\times) (\ni) (\text{fil})$



²⁴[2, S. 484]

Kl.metamorfozi
filtere

(Esperanto)

$[\langle \text{Klang}_{kl} \rangle^{kl}] \langle \text{Filter}_{kf} \rangle (\text{Kl.mmf} \mid \text{filtere}): [\langle \text{Klang}_{kl} \rangle^{kl}]$

Wendet einen Klangfilter auf einen Klang im Regal (k) an.

filtrilo
generiere.Klangfilter

(Esperanto)

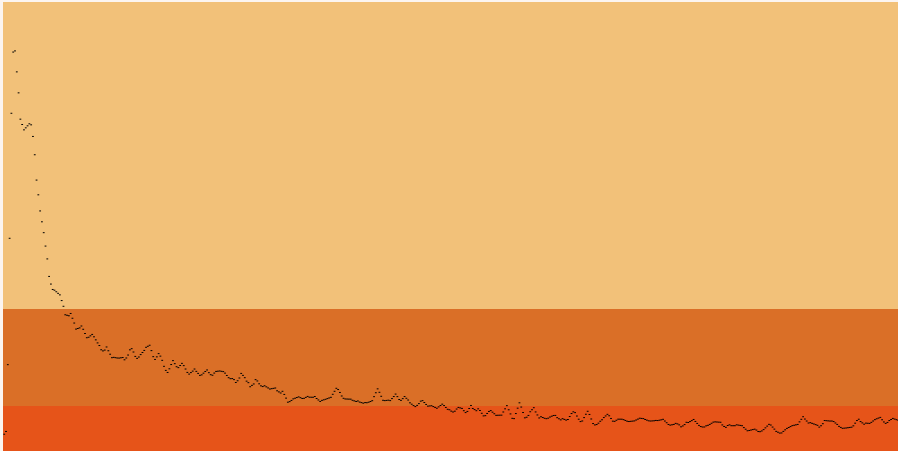
$\langle \text{additiv}_{sl}^Q \rangle \langle \text{subtraktiv}_{sl}^Q \rangle (\text{fil} \mid \text{generiere.Klangfilter}): \langle \text{Filter}_{kf} \rangle$

Generiert aus zwei Sammlungen mit den Koeffizienten einen Filter.

Verarbeitung des Skriptes *klangprofil.lysa*: 100 • Bildfrequenz bestimmen •
'~/Audio/Samples/Piano.pp.A4.au' • Klang importieren • 'Maximum' • 0 • profilieren • 'Maximum' •
glätten • 'Durchschnitt' • glätten • zeigen

Klangprofil:

Bilder:	500
Erstes:	0
Letztes:	499



Ein Klangprofil enthält Informationen über die zeitliche Veränderung der Amplitude eines Klanges. Die zeitliche Auflösung orientiert sich an der Anzahl der Einzelbilder eines zu erstellenden Videos (fps). Insbesondere ermöglicht ein Klangprofil, in Animationen Bild und Klang miteinander zu verknüpfen. Und auch bei der Klangsynthese kann sie eingesetzt werden.

↔ nein • \mathbb{Q} ungültig • \mathbb{Z} Einzelbilder • \mathbb{A} ungültig

frage.ab	110	Kp.pdm	110
gla	110	prf	110
glätte	110	profiliere	110
Kl.wrp	111	umhülle	111

$$\left[\langle \text{Profil}_{kp} \rangle^{kp} \right] \langle \text{Position}_{Gz\vee Q} \rangle (\text{Kp.pdm} \mid \text{frage.ab}): \langle \text{Intensität}_{zl} \rangle$$

$$\left[\langle \text{Profil}_{kp} \rangle^{kp} \right]$$

Befragt das registrierte Klangprofil nach einer Intensität. Handelt es sich bei der Positionsangabe um eine Ganzzahl, wird die Intensität des entsprechenden Bildes direkt zurückgegeben, ansonsten wird zuvor der Wert der Gegebenheit als lakonische Sekundenbruchteile (Λ) in eine Bildnummer umgerechnet.

Liegt die angegebene oder errechnete Bildnummer außerhalb des Klangprofils, wird eine Null als Intensität zurückgegeben.

$$\langle \text{Profil}_{kp} \rangle \langle \text{Methode}_{zk} \rangle (\text{gla} \mid \text{glätte}): \langle \text{geglättetes Profil}_{kp} \rangle$$

Erstellt zu einem Klangprofil eine geglättete Version. An Methoden stehen **Maximum**, **Minimum**, **Median** und **Durchschnitt** (auch alle unerkannten Angaben) zur Verfügung. Diese werden auf jeden einzelnen Wert mit seinem Vorgänger und Nachfolger angewendet.

$$\left\{ \langle \text{Bildwiederholfrequenz}_{zl} \rangle^{fps} \right\} \langle \text{Klang}_{kl} \rangle \langle \text{Art}_{zk} \rangle \langle \text{Versatz}_{\mathbb{Q}}^{\Lambda} \rangle (\text{prf} \mid \text{profiliere}): \langle \text{Profil}_{kp} \rangle \left\{ \langle \text{Bildwiederholfrequenz}_{zl} \rangle^{fps} \right\}$$

Erstellt zu einem Klang ein Profil. Über die Angabe eines Versatzes in lakonischen Sekundenbruchteilen wird es innerhalb eines Videos zeitlich lokalisiert. Für das Profil stehen verschiedene Methoden zur Verfügung:

Maximum: Zu jedem Bild wird der Maximalbetrag der Abtastwerte festgehalten (wird auch bei allen unerkannten Methoden genommen)

Durchschnitt: Zu jedem Bild wird der Durchschnittsbetrag der Abtastwerte festgehalten

Quadratisches Mittel: Zu jedem Bild wird die Wurzel des durchschnittlichen Quadrats des Betrags der Abtastwerte festgehalten

Akustisches Mittel: Zu jedem Bild wird das akustische Mittel der Beträge der Abtastwerte festgehalten

**Kl.wrap
umhülle**

(Englisch)

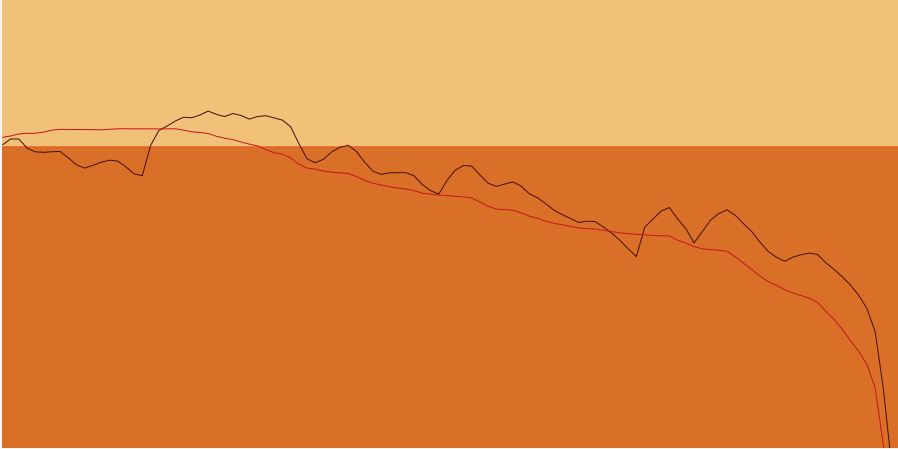
$$\left[\langle \text{Klang}_{kl} \rangle^{kl} \right] \langle \text{Start}_Z \rangle \langle \text{Länge}_Z \rangle \langle \text{Hüllkurve}_{Kp \vee Ew} \rangle (\text{Kl.wrap} \mid \text{umhülle}):$$
$$\left[\langle \text{Klang}_{kl} \rangle^{kl} \right]$$

Hüllt den angegebenen Abschnitt des registrierten Klangs in eine durch ein Klangprofil oder eine Entwicklung beschriebene Hüllkurve. Aus einem Klangprofil wird eine entsprechende Entwicklung generiert. Die Abtastwerte werden mit den Werten der Entwicklung multipliziert. Sollte der angegebene Abschnitt ganz oder teilweise außerhalb des Klangs liegen, bleiben die entsprechenden Stellen unberücksichtigt.

Eine Entwicklung wird lediglich als Muster verwendet, das auf die Angaben für Position und Länge übertragen (S. 91) wird.

Verarbeitung des Skriptes *raumklang.lysa*: '~/prolysarik/Fragezeichentext/Material/titel.au' • Audio importieren • zeigen

Raumklang:	
Abtastwerte:	528000
Bilder:	275
Maximum:	0,387667149
Minimum:	-0,412957549
Lautstärke (LUFS):	-23,000000001%



Raumklang stellt eine Stereoaufnahme dar. Es gibt zwei Kanäle (links und rechts). Die zeitliche Auflösung ist auf 48 kHz festgelegt (160 Abtastwerte pro lakonischem Sekundenbruchteil). Abtastwerte werden intern als doppelt genaue Fließkommazahlen (64 Bit) gespeichert, extern (Datei) jedoch mit einfacher Genauigkeit (32 Bit) gelesen oder abgelegt.

↪ ja • Q Abtastwerte • Z Abtastwerte • 𐀀 Abtastwerte

♪ (u266a) 116	positioniere 114
akk 113	reguliere 114
erschaffe.Raumklang 113	richte.aus 115
exportiere.Audio 113	Rk.adg 116
importiere.Audio 113	Rk.ext 117
lda 113	Rk.ldn 114
	Rk.ori 115

Rk.pan	114	sva	113
spiele	116	verlängere	117

akustiko (Esperanto)
erschaffe.Raumklang

$\langle L\ddot{a}nge_Z^{0\Lambda} \rangle$ {akk | erschaffe.Raumklang}: $\langle Stille_{rk} \rangle$

Generiert einen stillen Raumklang der angegebenen Länge (in lakonischen Sekundenbruchteilen).

save audio (Englisch)
exportiere.Audio

$\langle Aufnahme_{rk} \rangle$ $\langle Dateiname_{zk} \rangle$ {sva | exportiere.Audio}

Speichert einen Raumklang als **Au**-Datei (stereo, 48 kHz, Kodierung 6).

Insofern der angegebene Dateiname nicht auf .au endet, wird die Endung automatisch hinzugefügt.

load audio (Englisch)
importiere.Audio

$\langle Dateiname_{zk} \rangle$ {lda | importiere.Audio}: $\langle Aufnahme_{rk} \rangle$

Lädt einen Raumklang aus einer Au-Datei (stereo, 48 kHz, Kodierung 6). Der Dateiname muß den kompletten Zugriffspfad enthalten.

$$\left[\langle \text{Raumklang}_{rk} \rangle^{rk} \right] \langle \text{Klang}_{kl} \rangle \langle \text{Start}_{\mathbb{Z}}^{\Delta} \rangle \langle \text{Dämpfung}_{\mathbb{Z}}^{-dB} \rangle \langle \text{Position}_{Ew \vee Q}^{\pm 1} \rangle \\ (\text{Rk.pan} \mid \text{positioniere}): \left[\langle \text{Raumklang}_{rk} \rangle^{rk} \right]$$

Mischt den Klang intensitätsstereophonisch mit beiden Kanälen eines Raumklangs unter Erhaltung des Lautstärkeindrucks.²⁵

An welcher Position der Klang einzumischen ist, ist in lakonischen Sekundenbruchteilen anzugeben.

In Dezibel (negativ) ist anzugeben, inwieweit der Klang vor dem Mischen zu dämpfen ist.

Die Entwicklung für die stereophonische Positionsangabe wird auf die Länge des Klangs übertragen (S. 91). Der Wertebereich der Ordinate muß im Intervall ± 1 liegen (wird automatisch darauf begrenzt). Der Minusbereich steht dabei für links, die Null steht für das Zentrum und der positive Wertebereich für rechts.

Handelt es sich bei der Positionsangabe um keine Entwicklung, wird der Wert dieser Gegebenheit als feste Position genommen.

Da ein Klang im Vergleich zum Raumklang mit der doppelten Abtastfrequenz arbeitet, werden die Abtastwerte auf den geraden Positionen für den linken, und auf den ungeraden Positionen für den rechten Kanal des Raumklangs verwendet. Dadurch sind die Kanäle nicht identisch, sondern geringfügig gegeneinander in der Phase verschoben.

$$\left[\langle \text{Raumklang}_{rk} \rangle^{rk} \right] \langle \text{Vorgabe}_{\mathbb{Q}}^{LIFS} \rangle (\text{Rk.ldn} \mid \text{reguliere}): \\ \left[\langle \text{Raumklang}_{rk} \rangle^{rk} \right]$$

²⁵[2, S. 236]

Hebt oder senkt die Lautstärke eines Raumklangs auf das angegebene Niveau (in **LUFS**).

Die Lautstärke wird über die gesamte Länge des Raumklangs ermittelt (gemäß **ITU-R BS.1170-4** und **EBU R 128, Tech 3341**). Dabei wird zunächst in Abschnitten von 0,4 Sekunden in Zehntelsekundenschritten die momentane Lautstärke berechnet. Alle Abschnitte, die leiser sind als -70 LUFS, bleiben unberücksichtigt. Der Durchschnitt wird berechnet und in einem weiteren Schritt daraus die Lautstärke, wobei dann auch noch alle Abschnitte unberücksichtigt bleiben, die mehr als 10 LUFS darunter liegen.

Auf die geforderte Lautstärke wird der Raumklang durch Verstärken oder Dämpfen um die Differenz in Dezibel gebracht.

Rk.orienti
richte.aus

(Esperanto)

$$\left[\langle \text{Raumklang}_{\text{rk}} \rangle^{\text{rk}} \right] \langle \text{Abstand}_{\text{Q}}^{\text{m}} \rangle \langle \text{horizontal}_{\text{SI}}^{\text{Q}[0-1]} \rangle \langle \text{drüber}_{\text{Q}}^{[0-1]} \rangle \\ \langle \text{drunter}_{\text{Q}}^{[0-1]} \rangle (\text{Rk.ori} \mid \text{richte.aus}): \left[\langle \text{Raumklang}_{\text{rk}} \rangle^{\text{rk}} \right]$$

Stattet einen Raumklang mit richtungsabhängigen Empfindlichkeiten für eine Laufzeitstereophonie aus. Die horizontalen Faktoren müssen zumindest eine Gegebenheit enthalten. Deren Wert gilt dann horizontal für alle Richtungen. Bei mehreren Gegebenheiten wird der Vollkreis in entsprechend viele Segmente unterteilt. Der erste Wert gilt stets für einen Winkel von 30 Grad, für das rechte Mikrofon im Uhrzeigersinn (1 Uhr), für das linke entgegen (11 Uhr). Abgesehen davon gelten für beide Mikrophone die gleichen Werte.

Für die vertikale Empfindlichkeit sind nur Werte für die Richtung direkt ober- und unterhalb des Mikrophons anzugeben.

Falls keine Empfindlichkeiten gesetzt sind, gilt in alle Richtungen eine ungedämpfte Empfindlichkeit (Faktor 1).

Die Empfindlichkeit der Mikrophone wird anders als in der Realität nicht durch die Frequenz des aufzunehmenden Klanges beeinflusst.

Der Mikrofonabstand ist in Metern anzugeben. Voreingestellt sind 17,5 cm.



Rk.auxdigi
spiele

(Strg+⇧+u266a)
(Esperanto)

$$\left[\langle \text{Raumklang}_{g_{rk}} \rangle^{rk} \right] \langle \text{Klang}_{kl} \rangle \langle \text{Start}_{\underline{z}}^{\Delta} \rangle \langle \text{Dämpfung}_{\underline{z}}^{-dB} \rangle \langle \text{Abstand}_{Tp}^m \rangle$$

$$\langle \text{Methode}_{zk} \rangle (\text{♩} \mid \text{Rk.adg} \mid \text{spiele}): \left[\langle \text{Raumklang}_{g_{rk}} \rangle^{rk} \right]$$

Fügt einem Raumklang einen Klang an den angegebenen Positionen (räumlich in Metern, zeitlich in lakonischen Sekundenbruchteilen) hinzu. Die Aufteilung auf die Stereokanäle erfolgt lauffzeitstereophonisch. Für jedes der beiden Mikrophone erfolgt die Aufnahme entsprechend der Distanz zur angegebenen Klangquelle (unter Berücksichtigung der richtungsabhängigen Empfindlichkeit).

Die Lautstärke des Klangs kann für eine Standarddistanz von einem Meter angepaßt werden und berechnet sich entsprechend der Distanz zum jeweiligen Mikrophon. Die Position der Klangquelle ist in Metern anzugeben: erster Wert Distanz links (negativ) oder rechts (positiv), zweiter Wert vorne (positiv) oder hinten (negativ), dritter Wert drüber (positiv) oder drunter (negativ). Der Hörer wird also im Ursprung des Koordinatensystems gedacht, die Mikrophone befinden sich gleichabständig links und rechts davon.

Mit hoher Wahrscheinlichkeit muß der jeweils gesuchte Wert des Klanges interpoliert werden, da er zwischen einzelnen Abtastwerten liegt. Dazu stehen mehrere Verfahren²⁶ zur Verfügung:

runden Es wird der Abtastwert an der nächsten gerundeten Position genommen (auch bei allen nicht erkannten Angaben). Bei einer Schallgeschwindigkeit von 343,2 m/s und den 96.000

²⁶[4, S. 539–549]

Abtastwerten pro Sekunde eines Klages hat das Runden eine Genauigkeit von $\pm 1,7875$ mm.

linear Der Wert zwischen den beiden nächsten Abtastwerten wird linear interpoliert (zwei Werte).

skkPz Der Wert wird mit einem kardinalen kubischen Polynomzug mit dem Standardkontrollwert Eins interpoliert (vier Werte).

Catmull-Rom Wieder ein kardinaler kubischer Polynomzug, doch mit dem Kontrollwert 0,5 (vier Werte).

Lanczos2 Gefensterter Kardinalsinus (vier Werte).

Lanczos3 Gefensterter Kardinalsinus (sechs Werte).

Rk.extend

(Englisch)

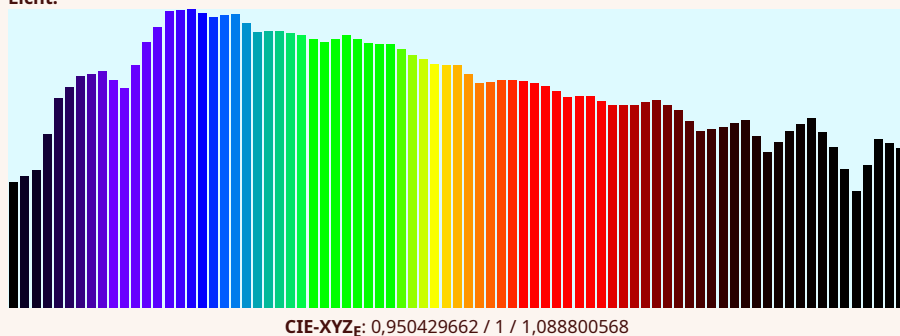
verlängere

$$\left[\langle \text{Raumklang}_{rk} \rangle^{rk} \right] \langle \text{Verlängerung}_{rk} \rangle (\text{Rk.ext} \mid \text{verlängere}): \\ \left[\langle \text{Raumklang}_{rk} \rangle^{rk} \right]$$

Hängt einen Raumklang an einen anderen dran.

Verarbeitung des Skriptes *licht.lysa*: Illuminant D65 (Licht) • zeigen

Licht:



Licht repräsentiert ein relatives Spektrum sichtbarer elektromagnetischer Strahlung (377,5–782,5 nm in 5 nm Intervallen = 81 Werte).

Der Natur der Sache würde es eher entsprechen, ein Spektrum durch Frequenzen statt durch Wellenlängen zu charakterisieren, da sich die Wellenlängen beim Übergang des Lichts in ein Medium anderer Dichte ändern, wohingegen die Frequenz gleich bleibt, da sich die Geschwindigkeit entsprechend verändert, doch gängige Praxis ist Angabe der Wellenlängen.²⁷ Die angegebene Wellenlängen entsprechen der Ausbreitung des Lichts in der unteren irdischen Atmosphäre (Luft).

Die Angaben sind darauf normiert, bei einer Umwandlung in Tristimulus-Koordinaten (CIE 1931 XYZ) einen Y-Wert (proportional zur Lumiananz) von Eins zu ergeben. Insofern entspricht das Spektrum also den Faktoren eines verlustfrei reflektierenden oder transmittierenden Materials. Dies erlaubt einen Umgang mit dem Phänomen Licht unabhängig von der jeweiligen Beleuchtungsstärke.

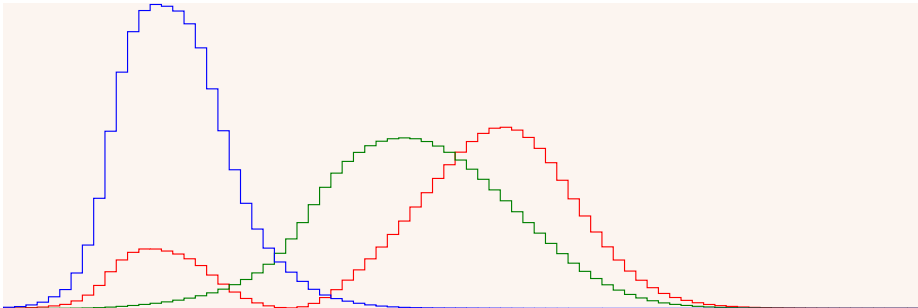
↪ ja • Q ungültig • Z ungültig • Å ungültig

²⁷[9, S. 3]

✱ (u263c) 120	ill 120
alf 121	pzt 120
dyl 120	stelle.dar 120
generiere.Licht 120	wandle 121
generiere.Tageslicht 120	

Folgende Pronomen stehen im Zusammenhang mit Licht im globalen Lexikon:

CMF	Die Spektralwertkurven in 5 nm-Intervallen zur Umwandlung von Spektraldaten in Koordinaten des CIE 1931 XYZ-Farbraums (2°). ²⁸ Die Sammlung enthält 81 Tripel (zu jedem 5 nm-Intervall eines, beginnend mit 377,5 nm) mit entsprechendem $\bar{x}(\lambda)$ an erster, $\bar{y}(\lambda)$ an zweiter und $\bar{z}(\lambda)$ an dritter Stelle.
Illuminant D65	Der CIE-Standardilluminant für durchschnittliches Tageslicht mit einer Farbtemperatur von circa 6.504°K (Daten aus CIE 15, Technical report: Colorimetry, 3rd Edition).
Weißpunkt D65	Tripel mit dem Maximum im durch den Standardilluminant D65 gebildeten Farbraum mit XYZ ₁₉₃₁ -Koordinaten (2°).



Die enthaltenen Spektralwertkurven (Colour Matching Functions) $\bar{x}(\lambda)$ (rot), $\bar{y}(\lambda)$ (grün) und $\bar{z}(\lambda)$ (blau) in 5 nm Intervallen zur Umwandlung von Spektraldaten in Koordinaten des CIE 1931 XYZ-Farbraums (2°) mit dem Weißpunkt S_e sind als Pronomen CMF abrufbar (Daten aus [CIE 15, Technical report: Colorimetry, 3rd Edition](#))

²⁸»... the most important spectral functions in colorimetry.« [9, S. 35]



illuminant

generiere.Licht

(Strg+⇧+u263c)

(Englisch)

$$\{\langle \text{Spektralwertkurven}_{SI}^{Tp} \rangle^{CMF}\} \langle \text{Spektralwerte}_{SI}^{Q[81]} \rangle \{ \text{☼} \mid \text{ill} \mid \text{generiere.Licht} \}: \langle \text{Spektrum}_{Li} \rangle \{ \langle \text{Spektralwertkurven}_{SI}^{Tp} \rangle^{CMF} \}$$

Generiert aus den Werten einer Sammlung von exakt 81 beliebigen Gegebenheiten ein Licht.

daylight

generiere.Tageslicht

(Englisch)

$$\{\langle \text{Spektralwertkurven}_{SI}^{Tp} \rangle^{CMF}\} \langle \text{Farbtemperatur}_Q^{[4.000-25.000]} \rangle \{ \text{dyl} \mid \text{generiere.Tageslicht} \}: \langle \text{Spektrum}_{Li} \rangle \{ \langle \text{Spektralwertkurven}_{SI}^{Tp} \rangle^{CMF} \}$$

Generiert ein Tageslicht der angegebenen Farbtemperatur (°K) entsprechend dem von der CIE vorgegebenen Verfahren.²⁹

prezenti

stelle.dar

(Esperanto)

$$\{ \langle \text{Weißpunkt}_{Tp} \rangle^{\text{Weißpunkt D65}} \langle \text{nach Bradford}_{Tk} \rangle^{\text{xyzNachBradford}} \langle \text{aus Bradford}_{Tk} \rangle^{\text{bradfordNachXyz}} \} \langle \text{Spektrum}_{Li} \rangle \{ \text{pzt} \mid \text{stelle.dar} \}: \langle \text{Farbe}_{fb} \rangle$$

$$\{ \langle \text{Weißpunkt}_{Tp} \rangle^{\text{Weißpunkt D65}} \langle \text{nach Bradford}_{Tk} \rangle^{\text{xyzNachBradford}} \langle \text{aus Bradford}_{Tk} \rangle^{\text{bradfordNachXyz}} \}$$

Stellt ein Licht als Farbe dar. Zur Darstellung eines Spektrums als Farbe wird eine Bradford-Adaption³⁰ vorgenommen, bei der ein Äquivalenzpunkt als Weißpunkt des Ausgangsfarbraums und der Weißpunkt D65 für den Zielfarbraum gesetzt sind.

²⁹Daten aus [9, S. 374 ff.]

³⁰[4, S. 356]

Ziel der Adaption ist nicht, Farbe in anderem Licht erscheinen zu lassen, sondern die Lichtfarbe zu zeigen.

aliformi
wandle

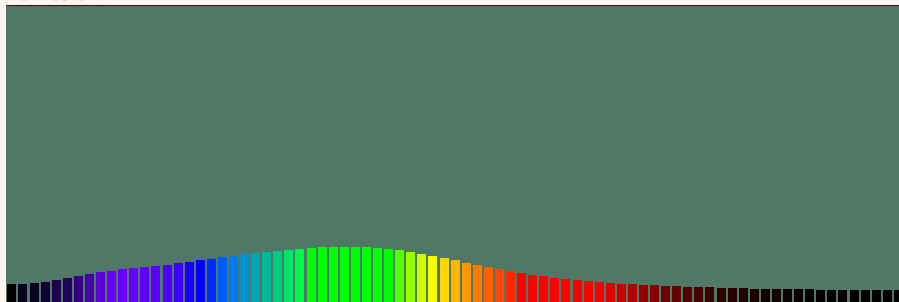
(Esperanto)

$\langle \text{Spektrum}_{Li} \rangle$ (alf | wandle): $\langle \text{Farbe}_{Fb} \rangle$

Wandelt ein Licht unadaptiert in eine Farbe um.

Verarbeitung des Skriptes *remission.lysa*: Sammlung erschaffen • 6,775862069 • 6,710344828 • 7,117241379 • 7,675 • 8,264285714 • 8,953571429 • 9,632142857 • 10,278571429 • 10,853571429 • 11,3 • 11,725925926 • 12,139285714 • 12,544444444 • 12,918518519 • 13,32962963 • 13,762962963 • 14,264285714 • 14,744444444 • 15,253846154 • 15,781481481 • 16,248148148 • 16,718518519 • 17,148148148 • 17,507407407 • 17,892307692 • 18,240740741 • 18,562962963 • 18,826923077 • 19,051851852 • 19,203703704 • 19,3 • 19,277777778 • 19,157692308 • 18,933333333 • 18,596296296 • 18,173076923 • 17,666666667 • 17,018518519 • 16,288461538 • 15,519230769 • 14,722222222 • 13,95 • 13,226923077 • 12,503846154 • 11,803846154 • 11,164 • 10,546153846 • 9,988461538 • 9,464 • 8,965384615 • 8,548 • 8,168 • 7,828 • 7,523076923 • 7,236 • 7 • 6,784 • 6,568 • 6,370833333 • 6,216 • 6,056 • 5,928 • 5,8 • 5,684 • 5,6 • 5,512 • 5,420833333 • 5,32 • 5,266666667 • 5,2 • 5,144 • 5,1 • 5,070833333 • 5,0125 • 5 • 4,972 • 4,9 • 4,9 • 4,9 • 4,9 • 4,9 • 4,9 • **einsammeln** • **Remission generieren** • **zeigen**

Remission:



Eine Remission repräsentiert entweder ein Spektrum der Anteile diffus reflektierten oder transmittierten sichtbaren Lichts einer Körperfarbe (377,5 – 782,5 nm in 5 nm Intervallen = 81 Werte).

↔ ja • Q ungültig • Z ungültig • Å ungültig

*	(u2600)	123	generiere.Remission	123
●	(u262f)	124	ilu	123
beleuchte		123	komplementiere	124
elv		123	kpm	124
entfalte		123	mische.an	125
fbp		125	rms	123



ilumini
beleuchte

(Strg+⬆+u2600)
(Esperanto)

$\langle \text{Licht}_{Li} \rangle \langle \text{Material}_{Rm} \rangle (\text{☼} \mid \text{ilu} \mid \text{beleuchte}): \langle \text{Restlicht}_{Li} \rangle$

Beleuchtet eine Körperfarbe und legt das remittierte Restlicht auf den Stapel. Dieses hat denselben Weißpunkt wie das Ausgangslicht.

elvoli
entfalte

(Esperanto)

$\{ \langle \text{Standardilluminant}_{Li} \rangle^{\text{illuminant D65}} \} \langle \text{Farbe}_{Fb} \rangle (\text{elv} \mid \text{entfalte}):$
 $\langle \text{Spektrum}_{Rm} \rangle \{ \langle \text{Standardilluminant}_{Li} \rangle^{\text{illuminant D65}} \}$

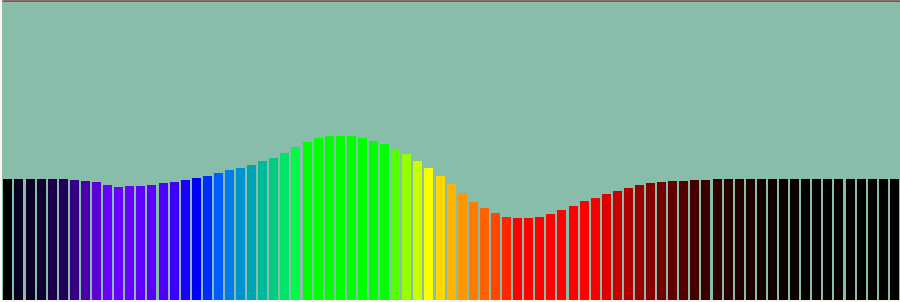
Errechnet unter dem Standardilluminant D65 ein Remissionsspektrum zu einer Farbe. Dieses ist in der Regel mit einem kleinen Fehler (sechste Stelle hinter dem Komma) behaftet. Zudem ist davon auszugehen, daß das Spektrum der **Metamerie** wegen keiner realen Körperfarbe entspricht: Auch wenn sich unter dem Standardilluminanten ein nahezu gleicher Farbeindruck ergibt, ist dies unter einer anderen Beleuchtung mit an Sicherheit grenzender Wahrscheinlichkeit nicht mehr der Fall.

remission
generiere.Remission

$\langle \text{Anteile}_{Sl}^{\mathbb{Q}[81]} \rangle (\text{rms} \mid \text{generiere.Remission}): \langle \text{Spektrum}_{Rm} \rangle$

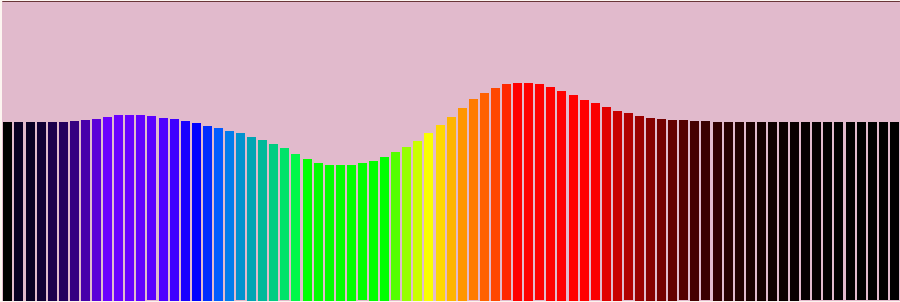
Erstellt aus den Werten einer Sammlung von 81 Gegebenheiten einer beliebigen Art, die als prozentuale Angaben für 5 nm Intervalle (beginnend bei 380 nm) interpretiert werden, ein Remissionsspektrum.

Remission:



• **komplementieren** • **zeigen**

Remission:



Entfalten eines hellen Malachitfarbtens (S. 127) mit anschließender Komplementärbildung



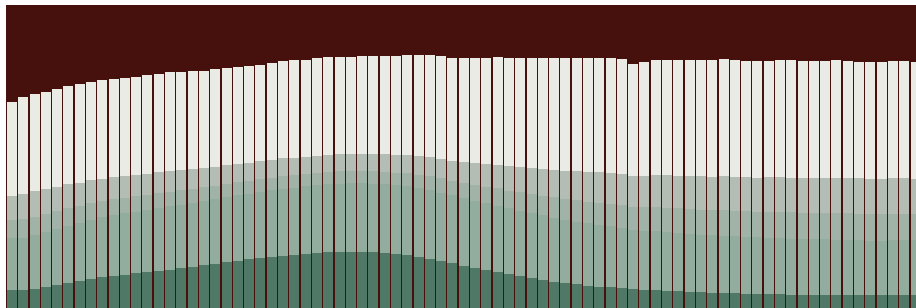
komplemento
komplementiere

(Strg+⬆+u262f)
(Esperanto)

⟨Körperfarbe_{rm}⟩ (👁 | kpm | komplementiere): ⟨Komplementärfarbe_{rm}⟩

Erstellt zu einer Remission das Komplementärspektrum, also eines, das genau den Anteil reflektiert oder transmittiert, der von der Ausgangsremission nicht reflektiert beziehungsweise transmittiert wird. Insofern in der Ausgangsremission bei einer Wellenlänge allerdings mehr als 100 % reflektiert beziehungsweise transmittiert

wird (möglich bei **Fluoreszenz**), wird dieser Wert im Komplementspektrum auf Null gesetzt.



Darstellung des Anmischens von Malachit mit Bleiweiß (hälftig) mit den Verfahren (von hell nach dunkel): 0, 2, 4 (Vgl. dazu [Cavaleri et al. 2013](#), S. 51)

farboplato
mische.an

(Esperanto)

$\langle \text{Primärfarbe}_{rm} \rangle \langle \text{Sekundärfarbe}_{rm} \rangle \langle \text{Anteil}_{\mathbb{Q}}^{[0-1]} \rangle \langle \text{Verfahren}_{\mathbb{Z}}^{[0-6]} \rangle (\text{fbp} | \text{mische.an}): \langle \text{Mischfarbe}_{rm} \rangle$

Mischt zwei Remissionen zu einer neuen. Dazu stehen verschiedene Verfahren zur Verfügung:

0 additives Mischen³¹: $R_{\lambda} = aR_{2\lambda} + (1 - a)R_{1\lambda}$

1 additiv-subtraktiv (nach [Lionel Simonot und Mathieu Hébert](#)):

$$R_{\lambda} = \tau(aR_{2\lambda} + (1 - a)R_{1\lambda}) + \theta R_{2\lambda}^a R_{1\lambda}^{1-a}$$

In diesem Fall (stärker subtraktiv) ist $\tau = 2 - \Phi$ und $\theta = \Phi - 1$.

2 ebenfalls additiv-subtraktiv, nun jedoch ausgeglichen: $\tau = 0,5$ und $\theta = 0,5$.

³¹wird auch bei allen unerkannten Verfahrensangaben angewandt

- 3 ebenfalls additiv-subtraktiv, nun aber stärker additiv: $\tau = \Phi - 1$
und $\theta = 2 - \Phi$.
- 4 subtraktiv-additiv (nach **Lionel Simonot und Mathieu Hébert**):

$$R_{\lambda} = (aR_{2\lambda}^{\tau} + (1 - a)R_{1\lambda}^{\tau}) \cdot R_{2\lambda}^{a\theta} R_{1\lambda}^{(a-1)\theta}$$

In diesem Fall (stärker subtraktiv) ist $\tau = 2 - \Phi$ und $\theta = \Phi - 1$.

- 5 ebenfalls subtraktiv-additiv, nun jedoch ausgeglichen: $\tau = 0,5$
und $\theta = 0,5$.
- 6 ebenfalls subtraktiv-additiv, nun aber stärker additiv: $\tau = \Phi - 1$
und $\theta = 2 - \Phi$.

Bei den Verfahren 1 – 6 werden die Remmissionswerte zur Berechnung auf $[0 - 1]$ normiert und das Ergebnis wieder auf Prozentangaben angehoben.

Verarbeitung des Skriptes *farbe.lysa*: *Illuminant D65 (Licht) • Sammlung erschaffen •* 15,588684211 • 17,631 • 19,806526316 • 22,1155 • 24,422052632 • 26,761947368 • 28,907 • 30,800777778 • 32,458 • 33,832222222 • 35,087157895 • 36,273210526 • 37,428333333 • 38,451611111 • 39,431 • 40,463833333 • 41,541 • 42,706388889 • 43,878833333 • 45,042833333 • 46,175777778 • 47,2625 • 48,312105263 • 49,290444444 • 50,122388889 • 50,860611111 • 51,545055556 • 52,161444444 • 52,588055556 • 52,798333333 • 52,686388889 • 52,397888889 • 51,875736842 • 51,120705882 • 50,147555556 • 48,913882353 • 47,529666667 • 45,972235294 • 44,338388889 • 42,648058824 • 40,941444444 • 39,264823529 • 37,599 • 35,952166667 • 34,333058824 • 32,806235294 • 31,344888889 • 29,986411765 • 28,778777778 • 27,679117647 • 26,669882353 • 25,677611111 • 24,796470588 • 23,946647059 • 23,014166667 • 22,319647059 • 21,677529412 • 21,195888889 • 20,783823529 • 20,330647059 • 19,937333333 • 19,566941176 • 19,250470588 • 18,912388889 • 18,674235294 • 18,4595625 • 18,220058824 • 18,030411765 • 17,801882353 • 17,645705882 • 17,521294118 • 17,345235294 • 17,255 • 17,12725 • 16,988529412 • 16,946588235 • 16,913764706 • 16,922058824 • 17,0058125 • 17,138176471 • 17,3934375 • **einsammeln • Remission generieren • beleuchten • wandeln • zeigen**

Farbe:

CIE-XYZ _{D65}	0,354228983	0,445183797	0,44132937
CIE-Yxy	0,445183797	0,285497662	0,358804444
CIE-L*a*b*	72,573629053	-21,957850406	4,699597517
CIE-L*C*h°	72,573629053	22,455142201	167,919354586
DIN99-Lab	80,600640135	-13,920656429	5,198827417
CIE-L*u*v* _{D65}	72,573629053	-26,665440778	10,525517513
CIE-HLC _{uv}	158,45960851°	72,573629053	28,667616762
LMS	0,421124837	0,458788316	0,405237454
UHDTV _{linear}	33,79239696%	49,042485876%	40,308339216%
UHDTV _{hlg}	51,770547317%	60,331002438%	55,936538381%
sRGB _{linear}	24,356533276%	51,014808062%	39,547920431%
sRGB	53,070047182%	74,200167365%	66,178236941%
HSV _{linear}	154,191380973°	52,255954298%	51,014808062%
HSV	157,221339905°	28,477186688%	74,200167365%
Alpha	100%	Hamburgefonstiv	

Farbe repräsentiert einen Farbton mit Koordinaten des XYZ₁₉₃₁-Farbraums (2°) mit dem Weißpunkt D65 zuzüglich Alphakanal (Opazität).

Bei der Umrechnung in einen anderen Farbraum kann es zu einer Reduktion kommen, wenn eine Farbe ansonsten außerhalb dieses Farbraums läge (Gamut-Mapping). Ziel dabei ist, das Größenverhältnis der drei Kanäle zueinander beizubehalten, denn in diesem Verhältnis liegt der Farbcharakter³². Durch die Reduktion wird also möglichst nur die Helligkeit angepasst.

³²[10, S. 14]

Wurde bei der Umrechnung in den sRGB-Farbraum ein Gamut-Mapping notwendig, wird das in den Informationen zur Farbe durch eine Raute in der Zeile der linearen sRGB-Werte angezeigt. In einem solchen Fall wurden dann auch alle darunter stehenden Angaben angepaßt.

↦ nein • Q Luminanz (y) • Z y (als Prozentangabe) • ∆ Wert

ΔE	(u0394)	129	isoliere	131
bld		133	kn1	131
clr		134	mische.Farben	133
dif		129	modifiziere.Opazität	134
ermittle.Farbabstand		129	opa	134
F		130	rechne.um	134
generiere.Farbe		130	xyz	130

Folgende Pronomen stehen im Zusammenhang mit Farben im globalen Lexikon:

IRGB-XYZ	Tripelkarree mit der Konvertierungsmatrix von linearen sRGB-Werten zu Koordinaten im XYZ ₁₉₃₁ -Farbraum (2°) mit dem Weißpunkt D65 (errechnet aus den Werten des Weißpunktes D65 und den sRGB-Primärvalenzen ³³): {klk} {0,64} {0,3} {0,15} {0,33} {0,6} {0,06} {0,03} {0,1} {0,79} {⇒} {tpk}, {↦} {ivs} {Weißpunkt D65} {vpd} {1} {1} {1} {ztp} {⇌} {⊗} {⊙}.
XYZ-IRGB	Tripelkarree mit der Konvertierungsmatrix aus Koordinaten im XYZ ₁₉₃₁ -Farbraum (2°) mit dem Weißpunkt D65 zu linearen sRGB-Werten: {IRGB-XYZ} {ivs}
UHDTV-XYZ	Tripelkarree mit der Konvertierungsmatrix von linearen RGB-Werten gemäß der Empfehlung ITU-R BT.2020 zu Koordinaten im XYZ ₁₉₃₁ -Farbraum (2°) mit dem Weißpunkt D65 (vgl. S. 72): {klk} {0,708} {0,17} {0,131} {0,292} {0,797} {0,046} {0} {0,033} {0,823} {⇒} {tpk}, {↦} {ivs} {Weißpunkt D65} {vpd} {1} {1} {1} {ztp} {⇌} {⊗} {⊙}.

³³[4, S. 351]

XYZ-UHDTV	Tripelkarree mit der Konvertierungsmatrix aus Koordinaten im XYZ ₁₉₃₁ -Farbraum (2°) mit dem Weißpunkt D65 zu linearen RGB-Werten gemäß BT.2100-2: $\langle \text{UHDTV-XYZ} \rangle$ (ivs)
XYZ-Bradford	Tripelkarree mit der Konvertierungsmatrix aus CIE-XYZ ₁₉₃₁ in den Bradford-Umrechnungsraum ³⁴ : $\langle \text{klk} \rangle \langle 0,8951 \rangle \langle 0,2664 \rangle \langle -0,1614 \rangle \langle -0,7502 \rangle \langle 1,7135 \rangle \langle 0,0367 \rangle \langle 0,0389 \rangle \langle -0,0685 \rangle \langle 1,0296 \rangle \langle \ni \rangle$ (tpk)
Bradford-XYZ	Tripelkarree mit der Konvertierungsmatrix aus dem Bradford-Umrechnungsraum nach CIE-XYZ ₁₉₃₁ : $\langle \text{XYZ-Bradford} \rangle$ (ivs)
XYZ-LMS	Tripelkarree mit einer Konvertierungsmatrix in einen Farbraum, welcher auf den Empfindlichkeiten der Zäpfchen in der Netzhaut beruht. Dafür finden sich verschiedene Konvertierungsmatrizen (z. B. eignet sich auch die Bradford-Matrix dazu). Voreingestellt ist eine als Hunt-Pointer-Estévez bekannte Transformation ³⁵ (normalisiert auf den Weißpunkt D65): $\langle \text{klk} \rangle \langle 0,40024 \rangle \langle 0,7076 \rangle \langle -0,08081 \rangle \langle -0,2263 \rangle \langle 1,16532 \rangle \langle 0,0457 \rangle \langle 0 \rangle \langle 0 \rangle \langle 0,91822 \rangle \langle \ni \rangle$ (tpk)
LMS-XYZ	Tripelkarree mit einer Konvertierungsmatrix aus einem auf den Empfindlichkeiten der Zäpfchen in der Netzhaut basierenden Farbraum in den CIE-XYZ ₁₉₃₁ -Farbraum mit dem Weißpunkt D65 : $\langle \text{XYZ-LMS} \rangle$ (ivs)
Glanzlichter	Der Faktor für den Weißpunkt D65 im XYZ ₁₉₃₁ -Farbraum (2°), der die Grenze der zulässigen Glanzlichter jenseits des Weißpunktes für HDR (UHDTV _{HLG}) beschreibt: $\frac{1}{OETF_{HLN}^{-1}[0,75]} = 3,774118118$

ΔE

difference

ermittle.Farbabstand

(Strg+⬆+u0394)

(Englisch)

$\{ \langle \text{Weißpunkt}_{Tp} \rangle^{\text{Weißpunkt D65}} \} \langle \text{Probe}_{Fb} \rangle \langle \text{Vergleich}_{Fb} \rangle \langle \text{Farbraum}_{zk} \rangle (\Delta E | \text{dif} | \text{ermittle.Farbabstand}): \langle \text{Abstand}_{zi} \rangle \{ \langle \text{Weißpunkt}_{Tp} \rangle^{\text{Weißpunkt D65}} \}$

³⁴[4, S. 356]

³⁵[9, S. 295 f.]

Berechnet den Abstand zweier Farben zueinander (**Empfindungs-delta**). Dies kann in verschiedenen Farbräumen berechnet werden. Zur Auswahl stehen:

LAB CIEL*a*b*

LUV CIEL*u*v*

DIN DIN99-Lab³⁶

Zumeist geht es darum, die Abweichung eines Ist- von einem Sollzustand festzustellen. In diesem Zusammenhang wird ein Abstand wie folgt bewertet³⁷:

ΔE	Bewertung	ΔE	Bewertung
1	kaum wahrnehmbar	0,0 – 0,5	nahezu unmerklich
2	erkennbar	0,5 – 1,0	für geübtes Auge wahrnehmbar
4	mittlere Differenz	1,0 – 2,0	geringer Unterschied
8	große Differenz	2,0 – 4,0	deutlicher Unterschied
16	zu große Differenz	4,0 – 5,0	selten hinnehmbarer Unterschied
		> 5,0	wesentlicher Unterschied

F

xyz

generiere.Farbe

$$\left\{ \langle \text{Weißpunkt}_{Tp} \rangle_{\text{Weißpunkt D65}} \right\} \langle X_Q^{[0-D65_X \times \text{Glanzlichter}]} \rangle \langle Y_Q^{[0-1 \times \text{Glanzlichter}]} \rangle \langle Z_Q^{[0-D65_Z \times \text{Glanzlichter}]} \rangle \langle \text{Alpha}_Q^{[0-1]} \rangle (\text{F} \mid \text{xyz} \mid \text{generiere.Farbe}): \langle \text{Farbe}_{Fb} \rangle \left\{ \langle \text{Weißpunkt}_{Tp} \rangle_{\text{Weißpunkt D65}} \right\}$$

Generiert eine Farbe aus Angaben zum XYZ₁₉₃₁-Farbraum (2°) sowie zur Opazität wie intern gespeichert. Insofern Angaben negativ sind, werden sie auf Null gesetzt.

Eine Anpassung an den Weißpunkt unterbleibt, da HDR (UHDTV) Werte jenseits dessen erlaubt (für Glanzlichter). Für die hybrid

³⁶wird auch im Falle aller unerkannten Angaben angewandt

³⁷z. B. [3, S. 34] (links)

logarithmische Gammakorrektur gilt, daß der Weißpunkt (idealer Reflektor) eine Signalstärke von lediglich 75% ergeben soll.³⁸ Diese erlaubt maximal eine knapp 3,8 fache Leuchtkraft im Verhältnis zum Weißpunkt.

kanal isoliere

$$\left\{ \left\langle \text{Konvertierungsmatrix}_{\text{Tk}} \right\rangle^{\text{XYZ-IRGB}} \left\langle \text{Konvertierungsmatrix}_{\text{Tk}} \right\rangle^{\text{XYZ-LMS}} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{\text{Tk}} \right\rangle^{\text{XYZ-UHDTV}} \left\langle \text{Weißpunkt}_{\text{Tp}} \right\rangle^{\text{Weißpunkt D65}} \right\} \left\langle \text{Farbe}_{\text{fb}} \right\rangle \\ \left\langle \text{Kanal}_{\text{zk}} \right\rangle (\text{knI} \mid \text{isoliere}): \left\langle \text{Wert}_{\text{zl}} \right\rangle \left\{ \left\langle \text{Konvertierungsmatrix}_{\text{Tk}} \right\rangle^{\text{XYZ-IRGB}} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{\text{Tk}} \right\rangle^{\text{XYZ-LMS}} \left\langle \text{Konvertierungsmatrix}_{\text{Tk}} \right\rangle^{\text{XYZ-UHDTV}} \left\langle \text{Weißpunkt}_{\text{Tp}} \right\rangle^{\text{Weißpunkt D65}} \right\}$$

Gibt den Wert eines einzelnen Kanals einer Farbe aus. Folgende Angaben sind möglich:

Alpha Der Grad der Opazität der Farbe ($[0 - 1]$).³⁹

X Der Wert des X-Kanals im CIE-XYZ-Farbraum mit dem Weißpunkt D65 ($[0 - D65_x \times \text{Glanzlichter}]$).

Y Der Wert des Y-Kanals im CIE-XYZ-Farbraum mit dem Weißpunkt D65 ($[0 - 1 \times \text{Glanzlichter}]$).

Z Der Wert des Z-Kanals im CIE-XYZ-Farbraum mit dem Weißpunkt D65 ($[0 - D65_z \times \text{Glanzlichter}]$).

x Der Chromatizitätswert der x-Achse im CIE-XYZ-Farbraum.

y Der Chromatizitätswert der y-Achse im CIE-XYZ-Farbraum.

z Der Chromatizitätswert der z-Achse im CIE-XYZ-Farbraum ($1 - x - y$).

L* Die Helligkeit im CIE-L*a*b*- sowie dem CIE-L*u*v*-Farbraum mit dem Weißpunkt D65 ($[0 - 100]$).

a* Die Position auf der rot-grün-Achse im CIE-L*a*b*-Farbraum mit dem Weißpunkt D65 ($[0 - \pm 127]$).

³⁸ITU-R BT.2408-6, S. 6

³⁹wird auch bei allen unerkannten Angaben zurückgegeben

- b*** Die Position auf der gelb-blau-Achse im CIE-L*a*b*-Farbraum mit dem Weißpunkt D65 ($[0 - \pm 127]$).
- C*** Die relative Farbsättigung im CIE-L*C*h°-Farbraum mit dem Weißpunkt D65 ($[0 - 179, 605122421]$).
- h°** Der Farbwinkel im CIE-L*C*h°-Farbraum mit dem Weißpunkt D65 ($[0 - 360]$).
- u*** Die Position auf der rot-grün-Achse im CIE-L*u*v*-Farbraum mit dem Weißpunkt D65 ($[-134 - 220]$).
- v*** Die Position auf der gelb-blau-Achse im CIE-L*u*v*-Farbraum mit dem Weißpunkt D65 ($[-140 - 122]$).
- Huv** Der Farbwinkel im CIE-HLCuv-Farbraum mit dem Weißpunkt D65 ($[0 - 360]$).
- Cuv** Die relative Farbsättigung im CIE-HLCuv-Farbraum mit dem Weißpunkt D65 ($[0 - 440, 408900909]$).
- L99** Die Helligkeit im DIN99-Farbraum mit dem Weißpunkt D65 ($[0 - 100]$).
- a99** Die Position auf der ersten Bunttonachse im DIN99-Farbraum mit dem Weißpunkt D65.
- b99** Die Position auf der zweiten Bunttonachse im DIN99-Farbraum mit dem Weißpunkt D65.
- L** Der Wert der langwelligen Reizung im LMS-Farbraum ($[0 - 1]$).
- M** Der Wert der mittelwelligen Reizung im LMS-Farbraum ($[0 - 1]$).
- S** Der Wert der kurzwelligen Reizung im LMS-Farbraum ($[0 - 1]$).
- wgR** Der Wert des roten Farbkanals im Wide Gamut-Farbraum gemäß BT.2100-2 (linear, $[0 - 1]$).
- wgG** Der Wert des grünen Farbkanals im Wide Gamut-Farbraum gemäß BT.2100-2 (linear, $[0 - 1]$).
- wgB** Der Wert des blauen Farbkanals im Wide Gamut-Farbraum gemäß BT.2100-2 (linear, $[0 - 1]$).
- hlgR** Der Wert des roten Farbkanals im Wide Gamut-Farbraum mit hybrid logarithmischer Gammakorrektur gemäß BT.2100-2 ($[0 - 1]$).
- hlgG** Der Wert des grünen Farbkanals im Wide Gamut-Farbraum mit hybrid logarithmischer Gammakorrektur gemäß BT.2100-2 ($[0 - 1]$).

- hlgb** Der Wert des blauen Farbkanals im Wide Gamut-Farbraum mit hybrid logarithmischer Gammakorrektur gemäß BT.2100-2 $([0 - 1])$.
- sR** Der Wert des roten Farbkanals im Standard-RGB-Farbraum $([0 - 1])$.
- sG** Der Wert des grünen Farbkanals im Standard-RGB-Farbraum $([0 - 1])$.
- sB** Der Wert des blauen Farbkanals im Standard-RGB-Farbraum $([0 - 1])$.
- sH** Der Farbwinkel aus der HSV-Darstellung des sRGB-Farbraums $([0 - 360])$.
- sS** Die Sättigung aus der HSV-Darstellung des sRGB-Farbraums $([0 - 1])$.
- sV** Die Helligkeit aus der HSV-Darstellung des sRGB-Farbraums $([0 - 1])$.
- lR** Der Wert des roten Farbkanals im linearen sRGB-Farbraum $([0 - 1])$.
- lG** Der Wert des grünen Farbkanals im linearen sRGB-Farbraum $([0 - 1])$.
- lB** Der Wert des blauen Farbkanals im linearen sRGB-Farbraum $([0 - 1])$.
- lH** Der Farbwinkel aus der HSV-Darstellung des linearisierten sRGB-Farbraums $([0 - 360])$.
- lS** Die Sättigung aus der HSV-Darstellung des linearisierten sRGB-Farbraums $([0 - 1])$.
- lV** Die Helligkeit aus der HSV-Darstellung des linearisierten sRGB-Farbraums $([0 - 1])$.

blending mische.Farben

(Englisch)

$\langle \text{Hintergrund}_{fb} \rangle \langle \text{Vordergrund}_{fb} \rangle \langle \text{Verfahren}_{\mathbb{Z}}^{[0-6]} \rangle (\text{bld} \mid \text{mische.Farben}):$
 $\langle \text{Farbe}_{fb} \rangle$

Mischt zwei Farben miteinander. Die zur Verfügung stehenden Verfahren (1 – 6) entsprechen jenen beim Anmischen von Remis-

sionen (S. 125) angewendet statt auf Wellenlängenintervalle auf die Farbkanäle.

Standard (alle Angaben außer 1 – 6) ist das Alpha-Blending (*A over B* nach **Porter-Duff**). Für den Alpha-Kanal wird dieses Verfahren immer angewandt.

opakeco

(Esperanto)

modifiziere.Opazität

$\langle \text{Vorlage}_{Fb} \rangle \langle \text{Faktor}_Q \rangle (\text{opa} \mid \text{modifiziere.Opazität}): \langle \text{Farbe}_{Fb} \rangle$

Bildet aus einer Farbe eine neue mit gleichem Farbton, bei der jedoch der Alphakanal mit einem Faktor multipliziert wird. Ist dieser größer als Eins, wird die Opazität vergrößert, ansonsten verringert.

Sollte dabei ein Alphawert größer als Eins entstehen, wird er automatisch auf diesen Wert begrenzt. Ebenso wird ein Wert kleiner als Null korrigiert.

Dieses Verb dient insbesondere dazu, bei allen Farbangaben ohne Bestimmung des Alphakanals diesen anzupassen.

color

(Englisch)

rechne.um

$\{ \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{\text{RGB-XYZ}} \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{\text{LMS-XYZ}} \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{\text{UHD TV-XYZ}} \langle \text{Weißpunkt}_{Tp} \rangle^{\text{Weißpunkt D65}} \} \langle a_Q \rangle \langle b_Q \rangle \langle c_Q \rangle (\langle d_Q \rangle) \langle \text{Art}_{zk} \rangle (\text{clr} \mid \text{rechne.um}): \langle \text{Farbe}_{Fb} \rangle$
 $\{ \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{\text{RGB-XYZ}} \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{\text{LMS-XYZ}} \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{\text{UHD TV-XYZ}} \langle \text{Weißpunkt}_{Tp} \rangle^{\text{Weißpunkt D65}} \}$

Dieses Verb dient für alle Farbangaben, bei der eine Umrechnung notwendig ist.

Zunächst wird eine Zeichenkette aufgenommen, die Auskunft darüber erteilt, aus welchen Angaben eine Farbe generiert werden soll.

Danach werden die entsprechenden Werte vom Stapel genommen. Möglich ist eine Farbe aus folgenden Angaben zu erstellen (überall dort, wo der Alphakanal unerwähnt bleibt, wird dieser auf 100 % gesetzt):

CIE-Yxy Angaben der Chromatizität mit ergänzender Luminanz im CIE-XYZ-Farbraum⁴⁰ (3 Werte: Y, x, y)

CIEL*a*b* Angaben aus dem CIEL*a*b*-Farbraum mit dem Weißpunkt D65⁴¹ (3 Werte: L*, a*, b*)

CIEL*C*h° Angaben aus dem CIEL*a*b*-Farbraum (D65) angegeben mit Polarkoordinaten (3 Werte: L*, C*, h)

CIEL*u*v* Angaben aus dem CIEL*u*v*-Farbraum mit dem Weißpunkt D65⁴² (3 Werte: L*, u*, v*)

CIE-HLCuv Angaben aus dem CIEL*u*v*-Farbraum mit dem Weißpunkt D65 angegeben mit Polarkoordinaten (3 Werte: H, L C)

LMS Angaben aus dem **LMS-Farbraum** (Standard: nach Hunt-Pointer-Estevez; 3 Werte: L, M, S)

UHDTV Angaben aus dem Wide Gamut-Farbraum gemäß BT.2100-2 (SDR, linear, in Prozent; 3 Werte: r, g, b)

UHDTVhlg Angaben aus dem Wide Gamut-Farbraum gemäß BT.2100-2 mit hybrid logarithmischer Gammakorrektur (in Prozent; 3 Werte: r, g, b)

sRGB Angaben aus dem Standard-RGB-Farbraum (in Prozent; 3 Werte: r, g, b)

sRGBA Angaben aus dem Standard-RGB-Farbraum (s. o.) mit Alphakanal (in Prozent; 4 Werte: r, g, b, a)

IRGB Linearisierte Angaben aus dem sRGB-Farbraum (s. o.) (in Prozent; 3 Werte: r, g, b)

IRGBA Linearisierte Angaben aus dem sRGB-Farbraum (s. o.) mit Alphakanal (in Prozent; 4 Werte: r, g, b, a)

sHSV Angaben aus der HSV-Darstellung des Standard-RGB-Farbraums (3 Werte: h, s, v)

⁴⁰[4, S. 342 ff.]

⁴¹[4, S. 346 ff.]

⁴²[4, S. 348 ff.]

IHSV Angaben aus der HSV-Darstellung des linearisierten Standard-RGB-Farbraums (3 Werte: h, s, v)

Verarbeitung des Skriptes *bild.lysa*: 'Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF.jpg' • **einbilden** • 176 • 870 • **Dupel generieren** • 7500 • **duplizieren** • **herausschneiden** • 750 • 'Durchschnitt' • **skalieren** • **zeigen**

Bild: 750×750 Bildpunkte



Verkleinerter Ausschnitt aus Leonardo Da Vincis *Mona Lisa* als Ausgangsmaterial der Bildbearbeitungsbeispiele

Bild repräsentiert ein digitales Abbild, von dem die Farbinformationen der einzelnen Bildpunkte nebst Informationen zu ihrer Opazität gespeichert sind.

Der Ursprung des Bildkoordinatensystems liegt in der linken, oberen Ecke. Die horizontale Koordinate wächst von links nach rechts,

die vertikale von oben nach unten. Der Drehsinn geht somit mit dem Uhrzeigersinn (linkshändig).

⇒ ja • Q ungültig • Z ungültig • ∅ ungültig

.....	143	fülle.Rechteck	141
avc	140	grundiere	141
Bd.lin	145	img	140
Bd.plc	142	lies.Bildpunkt.aus	141
Bd.pln	141	mez	142
Bd.pxl	141	miß.aus	142
Bd.rkt	141	plaziere	142
Bd.rtm	143	schneide.heraus	142
bilde.aus	138	scl	143
bilde.ein	140	setze.Bildpunkt	143
clp	142	skaliere	143
cvs	141	ziehe.Linie	145
edu	138		
ermittle.Durchschnittsfarbe	140		
erschaffe.Leinwand	141		

educate

(Englisch)

bilde.aus

$\langle Bild_{Bd} \rangle \langle Dateiname_{zk} \rangle \langle Format_{zk} \rangle$ (edu | bilde.aus)

Legt ein Bild als Datei ab. Der Dateiname enthält den kompletten Zugriffspfad. Er muß als Endung je nach gewünschtem Format entweder .png oder .PNG oder .lysa.gz enthalten, ansonsten wird eine entsprechende Endung angefügt. Über das Format entscheidet eine Angabe. Zur Verfügung stehen folgende Optionen:

- PNG32** Speichert das Bild als PNG (mit Alpha) mit 8 Bit pro Kanal (bpc). Die Bilddaten werden in den sRGB-Farbraum übertragen und gegebenenfalls angepaßt (Gamut-Mapping).
- PNG32f** wie oben; bei den Farbkanälen findet zudem eine Diffusion der Rundungsfehler nach dem **Floyd-Steinberg-Algorithmus** statt.

PNG64 Speichert das Bild als PNG (mit Alpha) mit 16 Bit pro Kanal. Die Bilddaten werden in den sRGB-Farbraum übertragen und gegebenenfalls angepaßt (Gamut-Mapping).

PNG64f wie oben; bei den Farbkanälen findet zudem eine Diffusion der Rundungsfehler nach dem Floyd-Steinberg-Algorithmus statt.

PNG64wg Es wird ein PNG (mit Alpha) mit 16 Bit pro Kanal geschrieben. Die Bilddaten werden in den Wide Gamut-Farbraum gemäß BT.2100-2 übertragen (UHDTV) und gegebenenfalls angepaßt (Gamut-Mapping). Ein cICP-Chunk wird geschrieben (9, 8, 0, 1). Zudem werden gAMA- (linear: 1) und cHRM-Chunks angelegt, um den Farbraum ersatzweise zu spezifizieren.

Achtung: Die derart erstellten Bilddateien können leider nicht korrekt wieder eingelesen werden! Sie sind ausschließlich zur externen Verarbeitung vorgesehen.

PNG64hdr Es wird ein PNG (mit Alpha) mit 16 Bit pro Kanal geschrieben. Die Bilddaten werden in den Wide Gamut-Farbraum gemäß BT.2100-2 übertragen (UHDTV) und gegebenenfalls angepaßt (Gamut-Mapping). Ein cICP-Chunk wird geschrieben (9, 18, 0, 1). Zudem werden gAMA- (0,25) und cHRM-Chunks angelegt, um den Farbraum ersatzweise zu spezifizieren.

Achtung: Die derart erstellten Bilddateien können leider nicht korrekt wieder eingelesen werden! Sie sind ausschließlich zur externen Verarbeitung vorgesehen.

EXR Es wird ein **EXR** (gegebenenfalls mit Alpha) mit Angaben des Wide Gamut-Farbraums gemäß BT.2100-2 in 32 Bit Gleitkommazahlen geschrieben. Die Daten werden unkomprimiert gespeichert.

Lysa Legt die Daten des Bildes in einer komprimierten Skriptdatei ab. Die einzelnen Bildpunkte werden als Angaben des CIE-Normvalenzsystems (XYZ) mit dem Standardbeobachter von 1931 (2°) und dem Weißpunkt D65 verlustfrei als hexadezimale Repräsentation des Speicherinhalts abgelegt. Diese Art der Speicherung ist sowohl umfangreicher als auch

beim Einlesen zeitaufwendiger als die Speicherung in einem echten Bildformat. Dafür geht aber auch keine Information verloren.

Wie das Bild eingelesen (S. 31) wird, darüber entscheidet der installierte Klammerdialekt. Zuerst wird im Winkeldialekt ein entsprechendes Bild generiert und registriert. Dann wird aus jedem vordefinierten Dialekt in die Standardklammern umgeschaltet, und die einzelnen Bildpunkte werden je nach ihrem Alphawert entweder gesetzt oder eingemischt (S. 143). Schließlich wird in den zuvor gültigen Dialekt umgeschaltet. Um kein neues Bild zu generieren, sondern die Daten stattdessen in das registrierte Bild einzutragen (schichtweiser Aufbau), ist das Skript in einem der anderen vordefinierten Dialekte außer den Winkelklammern einzubinden.

Diese Option wird auch bei jeder nicht erkannten Formatangabe gewählt.

image
bilde.ein

(Englisch)

$\langle \text{Dateiname}_{zk} \rangle$ (img | bilde.ein): $\langle \text{Bild}_{bd} \rangle$

Lädt ein Bild aus einer Datei (Angabe inklusive Pfad). Eingelesen werden können Bilder im Format PNG (nur sRGB) oder JPG.

average color
ermittle.Durchschnittsfarbe

(Englisch)

$\langle \text{Leinwand}_{bd} \rangle$ (avc | ermittle.Durchschnittsfarbe):
 $\langle \text{Durchschnittsfarbe}_{fb} \rangle$

Ermittelt die Durchschnittsfarbe aller Bildpunkte eines Bildes und legt sie auf den Stapel.

canvas erschaffe.Leinwand

(Englisch)

$\langle \text{Breite}_{\mathbb{Z}} \rangle \langle \text{Höhe}_{\mathbb{Z}} \rangle (\text{cvs} \mid \text{erschaffe.Leinwand}): \langle \text{Bild}_{\text{Bd}} \rangle$

Generiert ein leeres (vollständig durchsichtiges) Bild und legt es auf den Stapel.

Bd.rektangulo fülle.Rechteck

(Esperanto)

$[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}] \langle \text{Koordinaten}_{\text{Dp}} \rangle \langle \text{Breite}_{\mathbb{Q}} \rangle \langle \text{Höhe}_{\mathbb{Q}} \rangle \langle \text{Farbe}_{\text{Fb}} \rangle (\text{Bd.rkt} \mid \text{fülle.Rechteck}): [\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}]$

Füllt ein achsenparalleles Rechteck mit einer Farbe. Ist diese durchscheinend, wird sie mit dem bisherigen Untergrund gemischt (Alpha-Blending).

Bd.plenigi grundiere

(Esperanto)

$[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}] \langle \text{Farbe}_{\text{Fb}} \rangle (\text{Bd.pln} \mid \text{grundiere}): [\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}]$

Füllt das registrierte Bild mit einer Farbe. Befindet sich bereits etwas darauf, wird es durch die Farbe ersetzt, selbst wenn diese durchsichtig sein sollte.

Bd.pixel lies.Bildpunkt.aus

$[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}] \langle \text{Koordinaten}_{\text{Dp}} \rangle (\text{Bd.pxl} \mid \text{lies.Bildpunkt.aus}): \langle \text{Farbe}_{\text{Fb}} \rangle [\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}]$

Liest die Farbe eines einzelnen Bildpunktes aus dem registrierten Bild aus. Die Koordinaten werden gerundet. Liegen sie außerhalb

des Bildes, wird eine vollständig durchsichtige Farbe auf den Stapel gelegt.

mezuri
miß.aus

(Esperanto)

$\langle \text{Leinwand}_{\text{Bd}} \vee_{\text{Mk}} \rangle$ (mez | miß.aus): $\langle \text{Breite}_{\text{Gz}} \rangle$ $\langle \text{Höhe}_{\text{Gz}} \rangle$

Stellt die Ausmaße eines Bildes oder einer Maske fest und legt sie auf den Stapel.

Bd.place
plaziere

(Englisch)

$[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}]$ $\langle \text{Motiv}_{\text{Bd}} \rangle$ $\langle \text{Koordinaten}_{\text{Dp}} \rangle$ $\langle \text{Verfahren}_{\text{Z}}^{[0-6]} \rangle$ (Bd.plc | plaziere): $[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}}]$

Plaziert ein Motiv in einem Bild. Die anzugebenden Koordinaten bezeichnen die obere, linke Ecke des Motivs im aufnehmenden Bild. Das Motiv muß bereits in der richtigen Größe vorliegen. Über das aufnehmende Bild hinausragende Teile werden ignoriert.

Wie die einzelnen Bildpunkte des aufnehmenden Bildes aus des einzufügenden Motivs miteinander kombiniert werden, darüber entscheidet die Nummer des angegebenen Verfahrens entsprechend jenen beim Anmischen von Remissionen (S. 125). Bei allen Angaben außer 1 – 6 ist dies das Alpha-Blending, welches immer auf den Alpha-Kanal angewandt wird.

clip
schneide.heraus

(Englisch)

$\langle \text{Bild}_{\text{Bd}} \rangle$ $\langle \text{Koordinaten}_{\text{Dp}} \rangle$ $\langle \text{Breite}_{\text{Z}} \rangle$ $\langle \text{Höhe}_{\text{Z}} \rangle$ (clp | schneide.heraus): $\langle \text{Ausschnitt}_{\text{Bd}} \rangle$

Kopiert einen Ausschnitt eines Bildes zu einem neuen. Anzugeben sind die Koordinaten der linken oberen Ecke, die Breite und die Höhe in Bildpunkten (Werte werden gerundet). Liegen die Koordinaten außerhalb des Bildes oder ragt der Ausschnitt darüber hinaus, bleibt die entsprechende Fläche leer (durchsichtig).

Bd.rastrumero

(Esperanto)

setze.Bildpunkt

$$\left[\langle \text{Leinwand}_{Bd} \rangle^{Bd} \right] \langle \text{Koordinaten}_{Dp} \rangle \langle \text{Farbe}_{Fb} \rangle (. \mid \text{Bd.rtm} \mid \text{setze.Bildpunkt}) : \left[\langle \text{Leinwand}_{Bd} \rangle^{Bd} \right]$$

Besetzt einen Bildpunkt des registrierten Bildes mit einer Farbe. Ist die zu setzende Farbe durchscheinend, erfolgt ein Mischen mit der vorhandenen Farbe (Alpha-Blending).

Die Koordinaten werden gerundet. Befinden sich diese außerhalb der Bildfläche, geschieht nichts.

scale

(Englisch)

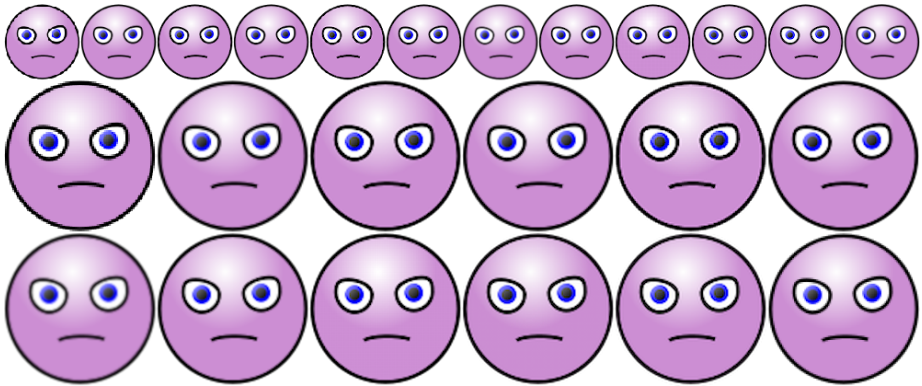
skaliere

$$\left\{ \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{XYZ \rightarrow RGB} \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{IRGB \rightarrow XYZ} \right\} \\ \langle \text{Original}_{Bd} \rangle \langle \text{Höhe}_Z \rangle \langle \text{Verfahren}_{zk} \rangle (\text{scl} \mid \text{skaliere}) : \langle \text{Ergebnis}_{Bd} \rangle \\ \left\{ \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{XYZ \rightarrow RGB} \langle \text{Konvertierungsmatrix}_{Tk} \rangle^{IRGB \rightarrow XYZ} \right\}$$

Skaliert ein Bild. Anzugeben ist die Höhe des Ergebnisses (das Seitenverhältnis wird beibehalten) und die anzuwendende Methode. Zur Auswahl⁴³ stehen:

Nachbar Jeder Bildpunkt des Ergebnisses erhält die Farbe des ihm am nächsten liegenden Bildpunktes des Originals (**Nearest neighbor**).

⁴³[4, S. 560]



Ein Smiley von 103×103 Bildpunkten um Φ raufskaliert (auf 166×166) sowie auf die Hälfte davon runterskaliert (auf 83×83) mit allen zwölf zur Verfügung stehenden Methoden: `<angrySmiley.png> (img) (*a) <83> <12> (x) <83> <5> (x) (cvs) (*) (:a) <83> <Nachbar> (scl) <0> <0> (D) <0> (Bd.plc) (:a) <83> <Bilinear> (scl) <83> <0> (D) <0> (Bd.plc) (:a) <83> <Bikubisch> (scl) <166> <0> (D) <0> (Bd.plc) (:a) <83> <Bikubisch, weich> (scl) <83> <3> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Bikubisch, scharf> (scl) <83> <4> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Catmull-Rom> (scl) <83> <5> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Kubische B-Spline> (scl) <83> <6> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Mitchell-Netravali> (scl) <83> <7> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Lanczos (2)> (scl) <83> <8> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Lanczos (3)> (scl) <83> <9> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Lanczos (4)> (scl) <83> <10> (x) <0> (D) <0> (Bd.plc) (:a) <83> <Durchschnitt> (scl) <83> <11> (x) <0> (D) <0> (Bd.plc) (:a) <166> <Nachbar> (scl) <0> <83> (D) <0> (Bd.plc) (:a) <166> <Bilinear> (scl) <166> <83> (D) <0> (Bd.plc) (:a) <166> <Bikubisch> (scl) <166> <2> (x) <83> (D) <0> (Bd.plc) (:a) <166> <Bikubisch, weich> (scl) <166> <3> (x) <83> (D) <0> (Bd.plc) (:a) <166> <Bikubisch, scharf> (scl) <166> <4> (x) <83> (D) <0> (Bd.plc) (:a) <166> <Catmull-Rom> (scl) <166> <5> (x) <83> (D) <0> (Bd.plc) (:a) <166> <Kubische B-Spline> (scl) <0> <249> (D) <0> (Bd.plc) (:a) <166> <Mitchell-Netravali> (scl) <166> <249> (D) <0> (Bd.plc) (:a) <166> <Lanczos (2)> (scl) <166> <2> (x) <249> (D) <0> (Bd.plc) (:a) <166> <Lanczos (3)> (scl) <166> <3> (x) <249> (D) <0> (Bd.plc) (:a) <166> <Lanczos (4)> (scl) <166> <4> (x) <249> (D) <0> (Bd.plc) (:a) <166> <Durchschnitt> (scl) <166> <5> (x) <249> (D) <0> (Bd.plc) (.Bd) (skalieren.png) (PNG32) (edu)`

Bilinear **Bilineare Interpolation.**

Bikubisch **Bikubische Interpolation** (1,0)

Bikubisch, weich desgleichen, weichzeichnend (0,25)

Bikubisch, scharf desgleichen, schärfend (1,75)

Catmull-Rom Catmull-Rom-Interpolation (0,5)

Kubische B-Spline Kubische B-Spline (0,0; 1,0)

Mitchell-Netravali **Mitchell-Netravali-Interpolation** ($\frac{1}{3}$; $\frac{1}{3}$)

Lanczos (2) **Lanczos-Interpolation** (4×4)

Lanczos (3) desgleichen (6×6)

Lanczos (4) desgleichen (8×8)

Durchschnitt Jeder Bildpunkt des Ergebnisses wird aus der Durchschnittsfarbe der von ihm abgedeckten Fläche des Originals gebildet. Dies ist die Methode, die auch bei allen nicht erkannten Angaben angewandt wird.

Die ersten beiden sowie die letzte Methode finden direkt im CIE-XYZ-Farbraum statt, der Rest wird zuvor in den linearen UHDTV-Farbraum übersetzt.

Bd.linio
ziehe.Linie

(Esperanto)

$\left[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}} \right] \langle \text{Start}_{\text{dp}} \rangle \langle \text{Ende}_{\text{dp}} \rangle \langle \text{Farbe}_{\text{fb}} \rangle \langle \text{Durchsichtigkeit}_{\text{es}} \rangle (\text{Bd.lin} | \text{ziehe.Linie}): \left[\langle \text{Leinwand}_{\text{Bd}} \rangle^{\text{Bd}} \right]$

Zeichnet eine einen Bildpunkt breite Linie. Falls gewünscht, wird sie umso durchsichtiger, je länger sie ist ($\frac{\alpha}{\log(e+l)}$). Derart können in Animationen (2D) Bildpunkte in Bewegung als Linie dargestellt werden.

Verarbeitung des Skriptes *einfärben.lysa*: 'ML.lysa.gz' • einbeziehen • registrieren • 'L*' • entlarven • registrieren • 1,5 • intensivieren • Bild entnehmen • ausmessen • Leinwand erschaffen • registrieren • 24,14 • 7,17 • 13,13 • 'CIEL*a*b*' • umrechnen • grundieren • Maske entnehmen • 'L*' • maskieren • Bild entnehmen • zeigen

Bild: 750×750 Bildpunkte



Die Mona Lisa aufgehellt und in **Sepia** eingefärbt

Eine Maske enthält die Informationen eines einzelnen Farbkanals eines Bildes. Sie ist als digitales Graustufenbild darstell- und speicherbar.

Der Ursprung des Maskenkoordinatensystems liegt in der linken, oberen Ecke. Die horizontale Koordinate wächst von links nach rechts, die vertikale von oben nach unten. Der Drehsinn geht somit mit dem Uhrzeigersinn (linkshändig).

→ ja • Q ungültig • Z ungültig • 𐀀 ungültig

avv	156	maskiere	163
Bd.msk	163	mex	158
Bd.sms	147	Mk.grz	162
entlarve	147	Mk.inv	161
ermittle.Durchschnittswert	156	Mk.its	160
erschaffe.Maske	156	Mk.kmb	161
exportiere.Maske	158	Mk.lev	160
fülle	158	Mk.plb	158
generiere.Verlaufsmaske	159	Mk.pnk	163
hebe	160	Mk.rla	163
intensiviere	160	p	163
invertiere	161	punktieren	163
kombiniere	161		
lies.Maskenpunkt.aus	162	vignettiere	163
lrv	156	vlm	159

Bd.senmaskigi entlarve

(Esperanto)

$$\left\{ \left\langle \text{Konvertierungsmatrix}_{Tk} \right\rangle^{XYZ-IRGB} \left\langle \text{Konvertierungsmatrix}_{Tk} \right\rangle^{XYZ-LMS} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{Tk} \right\rangle^{XYZ-UHDTV} \left\langle \text{Weißpunkt}_{Tp} \right\rangle^{\text{Weißpunkt D65}} \right\} \\ \left[\left\langle \text{Leinwand}_{Bd} \right\rangle^{Bd} \right] \left\langle \text{Farbkanal}_{zk} \right\rangle (\text{Bd.sms} \mid \text{entlarve}): \left\langle \text{Maske}_{Mk} \right\rangle \\ \left[\left\langle \text{Leinwand}_{Bd} \right\rangle^{Bd} \right] \left\{ \left\langle \text{Konvertierungsmatrix}_{Tk} \right\rangle^{XYZ-IRGB} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{Tk} \right\rangle^{XYZ-LMS} \left\langle \text{Konvertierungsmatrix}_{Tk} \right\rangle^{XYZ-UHDTV} \right. \\ \left. \left\langle \text{Weißpunkt}_{Tp} \right\rangle^{\text{Weißpunkt D65}} \right\}$$

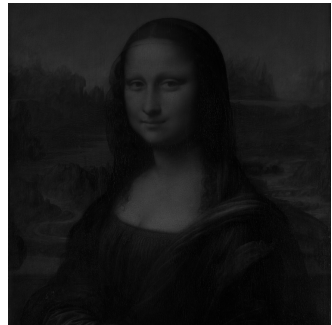
Generiert eine Maske aus einem Kanal eines Bildes. Die Maske hat dieselben Dimensionen wie das Bild. Über eine Zeichenkette wird der Kanal bestimmt. Möglich sind folgende Angaben:

Alpha Die Opazität des Bildes (diese Maske wird auch bei allen nicht erkannten Angaben erstellt)

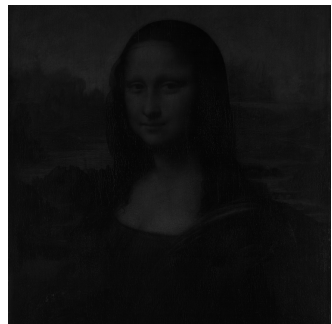
X Der X-Farbkanal aus dem CIE-XYZ-Farbraum mit dem Weißpunkt D65 (Darstellung standardisiert)
Die Darstellung ist dunkel, da der zulässige Wertebereich für Glanzlichter (HDR) erweitert wurde



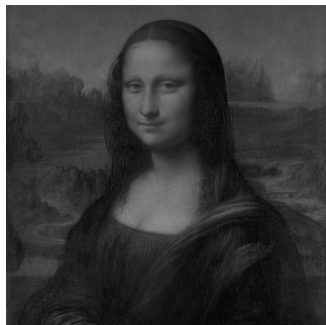
Y Der Y-Farbkanal aus dem CIE-XYZ-Farbraum mit dem Weißpunkt D65 (Darstellung standardisiert)



Z Der Z-Farbkanal aus dem CIE-XYZ-Farbraum mit dem Weißpunkt D65 (Darstellung standardisiert)



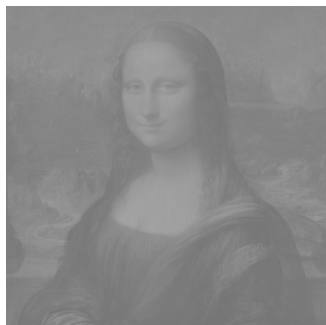
L* Der Helligkeits-Kanal sowohl aus dem CIE- $L^*a^*b^*$ - als auch dem CIE- $L^*u^*v^*$ -Farbraum mit dem Weißpunkt D65



a* Der rot-grün-Kanal aus dem CIE- $L^*a^*b^*$ -Farbraum mit dem Weißpunkt D65



b* Der gelb-blau-Kanal aus dem CIE- $L^*a^*b^*$ -Farbraum mit dem Weißpunkt D65



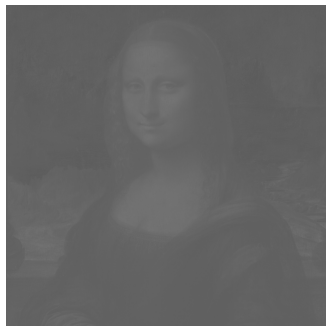
C* Die relative Farbsättigung im CIE- $L^*C^*h^*$ -Farbraum mit dem Weißpunkt D65 (Darstellung standardisiert)



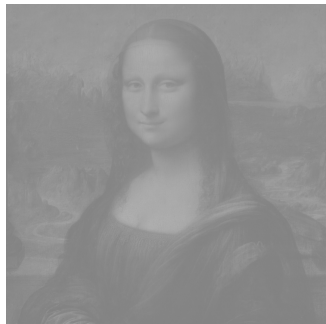
h° Der Farbwinkel im CIE- $L^*C^*h^\circ$ -Farbraum mit dem Weißpunkt D65



u^* Der rot-grün-Kanal aus dem CIE- $L^*u^*v^*$ -Farbraum mit dem Weißpunkt D65



v^* Der gelb-blau-Kanal aus dem CIE- $L^*u^*v^*$ -Farbraum mit dem Weißpunkt D65



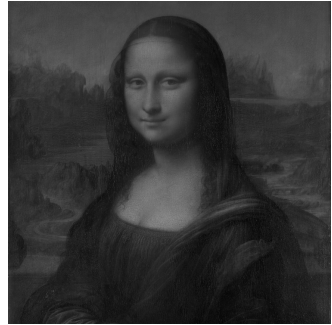
Huv Der Farbwinkel im CIE-HCLuv-Farbraum mit dem Weißpunkt D65



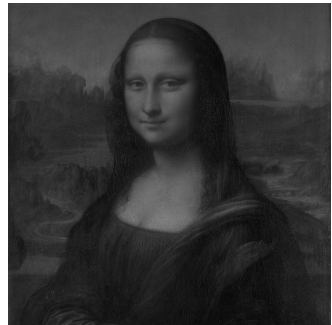
Cuv Die relative Farbsättigung im CIE-HCLuv-Farbraum mit dem Weißpunkt D65 (Darstellung standardisiert)



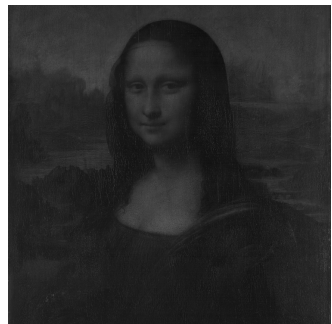
L Der Kanal mit den Werten der langwelligen Reizung im LMS-Farbraum (Standard: nach Hunt-Pointer-Estevez; Darstellung standardisiert)



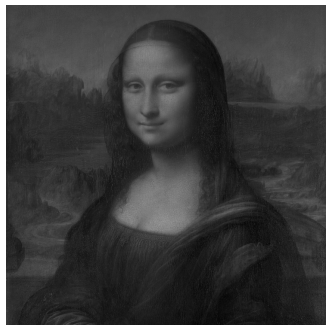
M Der Kanal mit den Werten der mittelwelligen Reizung im LMS-Farbraum (Standard: nach Hunt-Pointer-Estevez; Darstellung standardisiert)



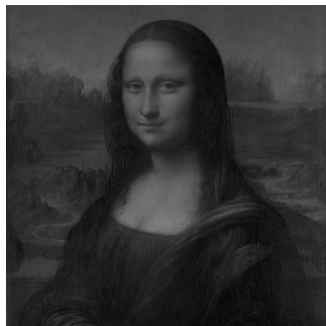
S Der Kanal mit den Werten der kurzwelligen Reizung im LMS-Farbraum (Standard: nach Hunt-Pointer-Estevez; Darstellung standardisiert)



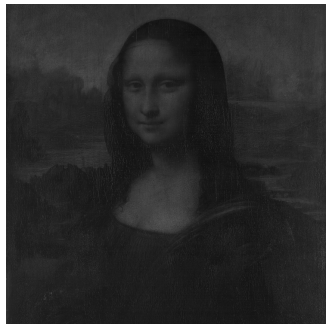
wgR Der rote Kanal (linear) im Wide-Gamut- RGB -Farbraum gemäß BT.2100-2 (Darstellung standardisiert)



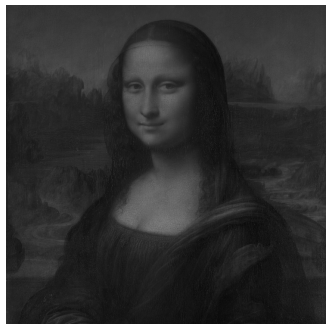
wgG Der grüne Kanal (linear) im Wide-Gamut- RGB -Farbraum gemäß BT.2100-2 (Darstellung standardisiert)



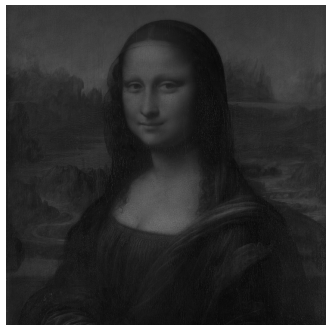
wgB Der blaue Kanal (linear) im Wide-Gamut- RGB -Farbraum gemäß BT.2100-2 (Darstellung standardisiert)



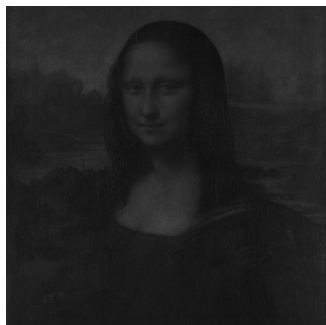
hlgr Der rote Kanal im Wide-Gamut- RGB -Farbraum mit hybrid logarithmischer Gammakorrektur gemäß BT.2100-2



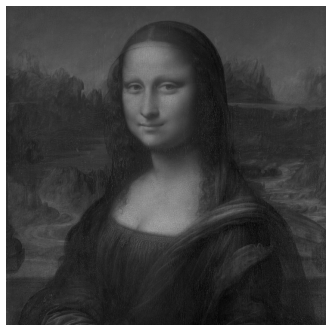
hlG Der grüne Kanal im Wide-Gamut-
RGB-Farbraum mit hybrid logarith-
mischer Gammakorrektur gemäß
BT.2100-2



hlB Der blaue Kanal im Wide-Gamut-
RGB-Farbraum mit hybrid logarith-
mischer Gammakorrektur gemäß
BT.2100-2



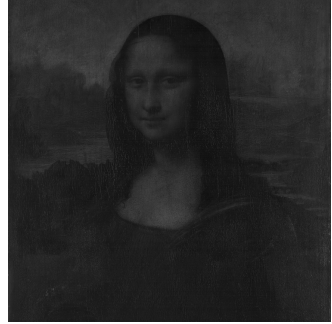
IR Der linearisierte rote Farbkanal im
Standard-RGB-Farbraum (Darstel-
lung standardisiert)



IG Der linearisierte grüne Farbkanal im
Standard-RGB-Farbraum (Darstel-
lung standardisiert)



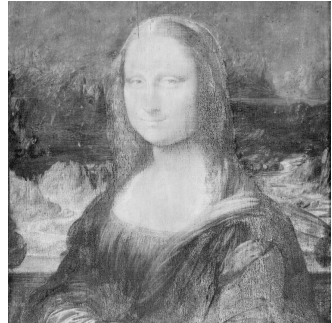
- IB** Der linearisierte blaue Farbkanal im Standard-RGB-Farbraum (Darstellung standardisiert)



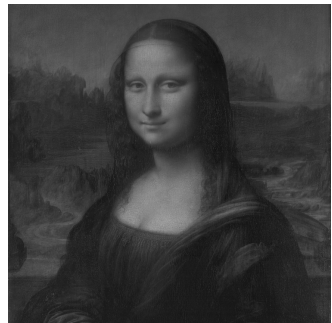
- IH** Der Farbwinkel aus der HSV-Darstellung des linearisierten sRGB-Farbraums



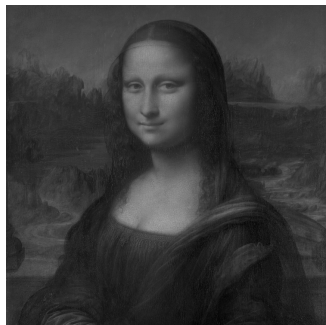
- IS** Die Sättigung aus der HSV-Darstellung des linearisierten sRGB-Farbraums



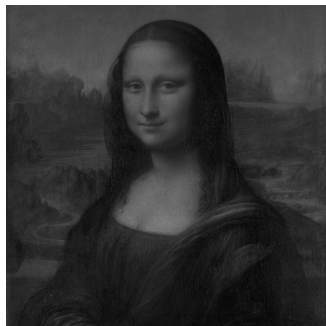
- IV** Die Helligkeit aus der HSV-Darstellung des linearisierten sRGB-Farbraums (Darstellung standardisiert)



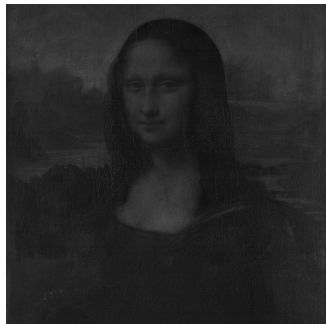
sR Der rote Farbkanal im Standard-
RGB-Farbraum



sG Der grüne Farbkanal im Standard-
RGB-Farbraum



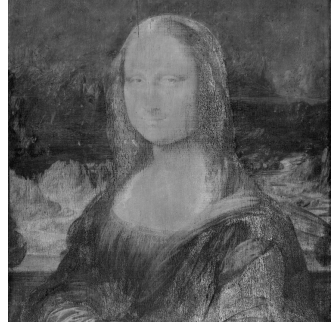
sB Der blaue Farbkanal im Standard-
RGB-Farbraum



sH Der Farbwinkel aus der HSV-
Darstellung des sRGB-Farbraums



sS Die Sättigung aus der HSV-Darstellung des sRGB-Farbraums



sV Die Helligkeit aus der HSV-Darstellung des sRGB-Farbraums



average value
ermittle.Durchschnittswert

(Englisch)

$\langle \text{Maske}_{mk} \rangle$ (avv | ermittle.Durchschnittswert): $\langle \text{Durchschnitt}_{Zl}^{[0-1]} \rangle$

Ermittelt den Durchschnittswert aller Bildpunkte einer Maske.

larva
erschaffe.Maske

(Lateinisch)

$\langle \text{Breite}_Z \rangle \langle \text{Höhe}_Z \rangle$ (lrv | erschaffe.Maske): $\langle \text{Maske}_{mk} \rangle$

Generiert eine leere (vollständig schwarze) Maske und legt sie auf den Stapel. Die Dimensionsangaben werden gerundet.

Verarbeitung des Skriptes *durchschnittswert.lysa*: 'ML.lysa.gz' • einbeziehen • registrieren • 'M' • entlarven • Durchschnittswert ermitteln • 'S' • entlarven • duplizieren • registrieren • Durchschnittswert ermitteln • dividieren • intensivieren • Maske entnehmen • 'S' • maskieren • 'L*' • entlarven • registrieren • 1,5 • intensivieren • Maske entnehmen • 'L*' • maskieren • Bild entnehmen • zeigen

Bild: 750×750 Bildpunkte



Ein erster Versuch, die Mona Lisa farblich aufzufrischen: die kurzweilige Farbmempfindung wird über den Durchschnittswert an das Niveau der mittelweiligen Farbmempfindung angeglichen (LMS-Farbraum nach Hunt-Pointer-Estevéz), zudem wird das Bild aufgehellt

maskenexport exportiere.Maske

$\langle \text{Maske}_{mk} \rangle \langle \text{Format}_{zk} \rangle (\text{mex} \mid \text{exportiere.Maske}):$

Legt eine Maske als Datei ab. Der Dateiname enthält den kompletten Zugriffspfad. Über das Format entscheidet eine Angabe. Zur Verfügung stehen folgende Optionen:

PNG Speichert das Bild als 16 Bit-Graustufen-PNG. Diese Speicherungsart empfiehlt sich ausschließlich zu Dokumentationszwecken.

Lysa Legt die Daten der Maske in einer Skriptdatei ab (Standardklammern). Der Dateiname (inklusive Pfad) muß auf `.lysa.gz` enden, ansonsten wird diese Endung hinzugefügt. Die Daten werden komprimiert abgelegt wie intern gespeichert (auf neun Stellen hinter dem Komma genau). Diese Art der Speicherung wird empfohlen, wenn eine Maske später weiter verwendet werden soll.

Beim Einlesen (S. 31) wird die Maske angelegt und ins Register geladen (S. 41) und die einzelnen Bildpunkte werden, insofern ihr Wert größer als Null ist, einzeln gesetzt (S. 163). Diese Option wird auch bei jeder nicht erkannten Formatangabe gewählt.

Mk.plombi fülle

(Esperanto)

$[\langle \text{Maske}_{mk} \rangle^{\text{mk}}] \langle \text{Wert}_{\mathbb{Q}}^{[0-1]} \rangle (\text{Mk.plb} \mid \text{fülle}): [\langle \text{Maske}_{mk} \rangle^{\text{mk}}]$

Füllt die registrierte Maske mit einem Wert. Dieser wird automatisch auf das Intervall Null bis Eins korrigiert.

Verarbeitung des Skriptes *farbverlauf.lysa*: 810 • 270 • Leinwand erschaffen • registrieren • 95,147153658 • 75,874066799 • 47,573576829 • 'sRGB' • umrechnen • grundieren • Bild verwenden • ausmessen • 'diagonal fallend' • Linear (Verlauf) • Maske aus Verlauf generieren • registrieren • invertieren • Maske entnehmen • 'Alpha' • maskieren • Bild entnehmen • duplizieren • ausmessen • Leinwand erschaffen • registrieren • 90,152993259 • 32,77680183 • 9,637784404 • 'sRGB' • umrechnen • grundieren • 0 • 0 • Dupel generieren • 0 • plazieren • Bild entnehmen • zeigen

Bild: 810×270 Bildpunkte



verlaufsmaske generiere.Verlaufsmaske

$\langle \text{Breite}_{\mathbb{Z}} \rangle \langle \text{Höhe}_{\mathbb{Z}} \rangle \langle \text{Modus}_{\text{zk}} \rangle \langle \text{Funktion}_{\text{vl}} \rangle (\text{vlm} \mid$
 $\text{generiere.Verlaufsmaske}): \langle \text{Maske}_{\text{mk}} \rangle$

Erstellt aus einem Verlauf eine Maske (vorzugsweise des Alphakanaals). Dafür stehen verschiedene Modi zur Verfügung (Richtungsangabe meint dunkel nach hell):

zentral zentraler Verlauf (aus der Mitte nach außen)

horizontal horizontaler Verlauf (von links nach rechts)

vertikal vertikaler Verlauf (von oben nach unten)

diagonal fallend diagonaler Verlauf (von links oben nach rechts unten)

diagonal steigend diagonaler Verlauf (von links unten nach rechts oben)

Bei allen nicht erkannten Modusangaben wird ein zentraler Verlauf erstellt. Um die Richtungen umzukehren ist die Maske zu invertieren (S. 161).

hebe

(Esperanto)

$$[\langle Maske_{mk} \rangle^{Mk}] \langle Schwelle_Q \rangle \langle Funktion_{vl} \rangle (Mk.lev \mid hebe): [\langle Maske_{mk} \rangle^{Mk}]$$

Korrigiert die Werte einer Maske mit einem Verlauf. Zusätzlich ist eine Schwelle anzugeben, oberhalb der alle Werte der Maske auf Eins gesetzt werden. Der Verlauf modifiziert dann also nur die Werte unterhalb der Schwelle.

Die Schwelle wurde insbesondere in Hinblick auf den Alphakanal hinzugefügt.

Mit Hilfe dieser Handlung lässt sich z. B. die Gammakorrektur des sRGB-Farbraum auf eine Maske anwenden beziehungsweise aus ihr entfernen:

$$\llcorner \text{lineare Maske}_{\text{mk}} \lrcorner \langle \text{standardisieren} \rangle \langle (\text{\textcircled{R}}) \rangle \langle 1 \rangle \langle (\mapsto) \rangle \langle 0,0031308 \rangle \langle > \rangle \langle \langle 1 \rangle \langle 2,4 \rangle \langle \div \rangle \langle \star \rangle \langle 1,055 \rangle \langle \times \rangle \langle 0,055 \rangle \langle - \rangle \rangle \langle \langle 12,92 \rangle \langle \times \rangle \rangle \langle \neg \rangle \rangle \langle \wr \rangle \\ (\text{Mk.lev}) (. \text{Mk}) \rangle \langle ! \rangle \llcorner \text{standartisierte Maske}_{\text{mk}} \lrcorner$$

$\lfloor \text{standardisierte Maske}_{\text{mk}} \rfloor \langle \text{linearisieren} \rangle \langle \langle (\text{®}) \rangle \langle 1 \rangle \langle \langle (\mapsto) \rangle \langle 0,04045 \rangle \rangle \langle \rangle \rangle \langle \langle \langle 0,055 \rangle \rangle \langle + \rangle \langle 1,055 \rangle \rangle \langle \div \rangle \langle 2,4 \rangle \langle \star \rangle \rangle \langle \langle \langle 12,92 \rangle \rangle \langle \div \rangle \rangle \rangle \langle \backslash \rangle \rangle \rangle \langle \wr \rangle \langle \text{Mk.lev} \rangle \langle \text{.Mk} \rangle \rangle \langle ! \rangle \lfloor \text{linearisierte Maske}_{\text{mk}} \rfloor$

intensiviere

(Esperanto)

$\langle \textcolor{brown}{Maske}_{Mk} \rangle^{Mk} \langle \textcolor{brown}{Faktor}_Q \rangle (\text{Mk.its} \mid \text{intensiviere}): \langle \textcolor{brown}{Maske}_{Mk} \rangle^{Mk}$

Verändert die Bildpunktweite einer Maske durch Multiplikation mit einem Faktor. Werte, die dadurch kleiner als Null oder größer als Eins geraten, werden entsprechend korrigiert.

Mk.inversigi invertiere

(Esperanto)

$[\langle \text{Maske}_{mk} \rangle^{mk}] \text{ (Mk.inv | invertiere): } [\langle \text{Maske}_{mk} \rangle^{mk}]$

Invertiert⁴⁴ die Bildpunkte der registrierten Maske.

Eine neue, leere Maske (S. 156) z. B. ist durchgängig schwarz (als Bild betrachtet), als Alphakanal gesehen also vollständig durchsichtig. Durch das Invertieren kann sie opak gemacht werden (weiß als Bild betrachtet).

Mk.kombini kombiniere

(Esperanto)

$[\langle \text{Ziel}_{mk} \rangle^{mk}] \langle \text{Quelle}_{mk} \rangle \langle \text{Position}_{dp} \rangle \langle \text{Methode}_{zk} \rangle \vee \langle \langle \text{Methode}_{af} \rangle \rangle \text{ (Mk.kmb | kombiniere): } [\langle \text{Ziel}_{mk} \rangle^{mk}]$

Kombiniert zwei Masken miteinander. Die Methode wird entweder durch eine Zeichenkette oder eine Anführung bestimmt.

Bei einer Zeichenkette stehen folgende Methoden zur Verfügung:

additiv Die Summe aus Quelle und Ziel wird gebildet (Überschreitung des zulässigen Intervalls wird korrigiert)

subtraktiv Die Quelle wird vom Ziel abgezogen (Unterschreitung wird korrigiert)

multiplikativ Das Ziel wird mit der Quelle multipliziert

dividierend Das Ziel wird durch die Quelle dividiert (ist die Quelle Null, wird das Ziel auf Eins gesetzt; eine Überschreitung wird korrigiert)

durchschnittsbildend Der Durchschnitt von Quelle und Ziel wird gebildet

differenzierend Der Abstandsbetrag zwischen Quelle und Ziel wird gesetzt

⁴⁴ $p = 1 - p$

maximierend Das Maximum von Quelle und Ziel wird gesetzt
minimierend Das Minimum von Quelle und Ziel wird gesetzt
übernehmend Die Quelle überschreibt das Ziel, insofern die Quelle größer als Null ist
kopierend Die Quelle überschreibt das Ziel (gleichzeitig der Modus für alle unerkannten Angaben)

Handelt es sich bei der Methode um eine Anführung (gedacht für alle Fälle, in denen die andere Methode nicht passen), wird diese für jeden Bildpunkt der registrierten Maske vereigentlicht. Dabei findet sie folgendes Umfeld vor:

$$\left[\langle \text{Quellwert}_{Zl}^{[0-1]} \rangle^a \langle \text{Zielwert}_{Zl}^{[0-1]} \rangle^b \langle \text{Quellkoordinate}_{Dp} \rangle^c \langle \text{Zielkoordinate}_{Dp} \rangle^d \right] \\ \langle \langle \text{Anführung}_{Af} \rangle \rangle \leftarrow \text{J} : \langle \text{Ergebnis}_Q^{[0-1]} \rangle$$

Es wird eine beliebige Gegebenheit auf dem Stapel erwartet, deren Wert gegebenenfalls auf das Intervall Null bis Eins begrenzt und in die Maske im Regal eingetragen wird.

Mk.grizo (Esperanto)
lies.Maskenpunkt.aus

$$\left[\langle \text{Maske}_{Mk} \rangle^{Mk} \right] \langle \text{Koordinaten}_{Dp} \rangle \{ \text{Mk.grz} \mid \text{lies.Maskenpunkt.aus} \} : \\ \langle \text{Wert}_{Zl} \rangle \left[\langle \text{Maske}_{Mk} \rangle^{Mk} \right]$$

Liest den Wert eines bestimmten Maskenpunktes aus einer Maske aus. Liegen die Koordinaten außerhalb der Maske, wird ein Nullwert zurückgegeben.

Bd.maski maskiere

(Esperanto)

$$\left\{ \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{XYZ-IRGB} \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{IRGB-XYZ} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{XYZ-LMS} \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{LMS-XYZ} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{XYZ-UHDTV} \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{UHDTV-XYZ} \right. \\ \left. \left\langle \text{Weißpunkt}_{T_p} \right\rangle^{\text{Weißpunkt D65}} \right\} \left[\left\langle \text{Bild}_{B_d} \right\rangle^{B_d} \right] \left\langle \text{Maske}_{M_k} \right\rangle \left\langle \text{Farbkanal}_{Z_k} \right\rangle \left(\text{msk} \mid \right. \\ \left. \text{maskiere} \right): \left[\left\langle \text{Bild}_{B_d} \right\rangle^{B_d} \right] \left\{ \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{XYZ-IRGB} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{IRGB-XYZ} \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{XYZ-LMS} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{LMS-XYZ} \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{XYZ-UHDTV} \right. \\ \left. \left\langle \text{Konvertierungsmatrix}_{T_k} \right\rangle^{UHDTV-XYZ} \left\langle \text{Weißpunkt}_{T_p} \right\rangle^{\text{Weißpunkt D65}} \right\}$$

Ersetzt einen Kanal in einem Bild durch eine Maske. Dazu müssen diese allerdings dieselben Dimensionen aufweisen (kongruent sein). Möglich sind dieselben Kanäle wie beim Entlarven (S. 147).

p Mk.punkti punktieren

(Esperanto)

$$\left[\left\langle \text{Maske}_{M_k} \right\rangle^{M_k} \right] \left\langle \text{Koordinaten}_{D_p} \right\rangle \left\langle \text{Wert}_{\mathbb{Q}}^{[0-1]} \right\rangle \left(p \mid \text{Mk.pnt} \mid \text{punktieren} \right): \\ \left[\left\langle \text{Maske}_{M_k} \right\rangle^{M_k} \right]$$

Setzt einen Punkt in einer Maske auf den angegebenen Wert. Liegen die angegebenen Koordinaten außerhalb der Maske, wird nichts verändert.

Mk.randlichtabfall vignettieren

$$\left[\left\langle \text{Maske}_{M_k} \right\rangle^{M_k} \right] \left\langle \text{Abfall}_{\mathbb{Q}}^{\%} \right\rangle \left\langle \text{Stauchung}_{\mathbb{Q}} \right\rangle \left\langle \text{Funktion}_{v_l} \right\rangle \left(\text{Mk.rla} \mid \right. \\ \left. \text{vignettieren} \right): \left[\left\langle \text{Maske}_{M_k} \right\rangle^{M_k} \right]$$

Modifiziert eine Maske (vorzugsweise ein Helligkeitskanal), indem es die Bildpunkte in Abhängigkeit von der Distanz zum Zentrum

Verarbeitung des Skriptes *vignettieren.lysa*: 960 • 540 • Leinwand erschaffen • registrieren • 99 • 96,036656711 • 94,05 • 'sRGB' • umrechnen • grundieren • 'L*' • entlarven • registrieren • 15 • ϕ (Zahl) • Anfang • *kubizieren* • Ende • Verlauf generieren • vignettieren • Maske entnehmen • 'L*' • maskieren • Bild entnehmen • zeigen

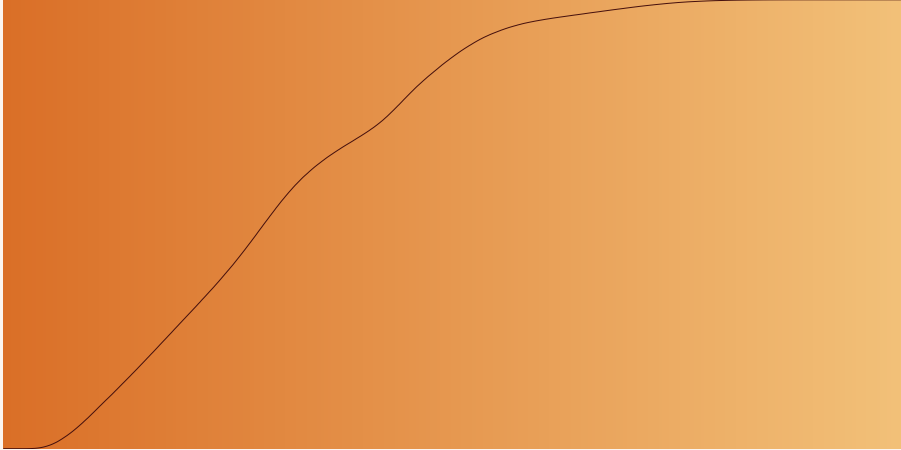
Bild: 960×540 Bildpunkte



abdunkelt. In Prozent ist das Maß der Abdunklung in den Ecken anzugeben (zwischen diesen und dem Zentrum regelt der Verlauf die Abdunklung). Außerdem kann bestimmt werden, wie die vertikale Distanz zum Zentrum zu werten ist (Angaben größer als Eins gestaucht, kleiner als Eins gestreckt). Um zum Beispiel bei einem Seitenverhältnis von 16 : 9 an den vertikalen Kanten eine entsprechende Abdunklung wie an den horizontalen zu erreichen, ist eine Stauchung von $\frac{16}{9}$ also 1,777778 anzugeben.

Verarbeitung des Skriptes *histogramm.lysa*: 'ML.lysa.gz' • einbeziehen • registrieren • 'L*' • entlarven • registrieren • 1,5 • intensivieren • Maske entnehmen • Histogramm generieren • zeigen

Kumulatives Histogramm:



Ein Histogramm repräsentiert das kumulative Histogramm einer Maske. Kumulativ heißt in diesem Zusammenhang, daß sich zu jedem Wert im Histogramm der Anteil der Punkte findet, die dunkler oder gleich hell wie dieser Wert sind. Aus den 1.001 Einzelwerten ergibt sich somit eine Linie, die bei Null beginnt und bei Eins endet. In dieser Hinsicht entspricht ein Histogramm einem Verlauf.

↔ nein • Q ungültig • Z ungültig • ∅ ungültig

eat	166	grd	166
generiere.Histogramm	166	itg	166
gib.Anteil	166	Mk.niv	168
gib.Einzelanteil	166	prt	166
gib.Intensität	166		
gleiche.an	168		

istogramma
generiere.Histogramm

(Italienisch)

$\langle Kanal_{Mk} \rangle$ (itg | generiere.Histogramm): $\langle Histogramm_{Hg} \rangle$

Erstellt zu einer Maske ein kumulatives Histogramm.

parto
gib.Anteil

(Esperanto)

$\langle Histogramm_{Hg} \rangle \langle Intensität_Q^{[0-1]} \rangle$ (prt | gib.Anteil): $\langle Anteil_{Zl}^{[0-1]} \rangle$

Legt den einer vorgegebenen Intensität entsprechenden Anteil aus einem kumulativen Histogramm auf den Stapel.

einzelanteil
gib.Einzelanteil

$\langle Histogramm_{Hg} \rangle \langle Intensität_Q^{[0-1]} \rangle$ (eat | gib.Einzelanteil): $\langle Anteil_{Zl}^{[0-1]} \rangle$

Legt den einer vorgegebenen Intensität entsprechenden Anteil aus einem flachen Histogramm (nicht kumulativ) auf den Stapel.

grado
gib.Intensität

(Esperanto)

$\langle Histogramm_{Hg} \rangle \langle Anteil_{Zl}^{[0-1]} \rangle$ (grd | gib.Intensität): $\langle Intensität_{Zl}^{[0-1]} \rangle$

Legt die einem vorgegebenen Anteil entsprechende Intensität aus einem kumulativen Histogramm auf den Stapel.

Verarbeitung des Skriptes *monalisa.lysa*: 'Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF.jpg' • einbilden • duplizieren • 176 • 870 • Dupel generieren • 7500 • 10000 • herausschneiden • registrieren • 'M' • entlarven • Histogramm generieren • vertauschen • registrieren • 'S' • entlarven • registrieren • angleichen • Maske entnehmen • 'S' • maskieren • 'L*' • entlarven • registrieren • 1,5 • intensivieren • Maske entnehmen • 'L*' • maskieren • Bild verwenden • 'MonaLisa.png' • 'PNG32f' • ausbilden • Bild entnehmen • 176 • 870 • Dupel generieren • 7500 • duplizieren • herausschneiden • 750 • 'Durchschnitt' • skalieren • zeigen

Bild: 750×750 Bildpunkte



Ein zweiter **Versuch einer Farbauffrischung der Mona Lisa**: die kurzweilige Farbmempfindung wird nun über ein Histogramm an die mittelwellige angeglichen (LMS-Farbraum nach Hunt-Pointer-Estevez), zudem wird das Bild in gleichem Maße aufgehellt

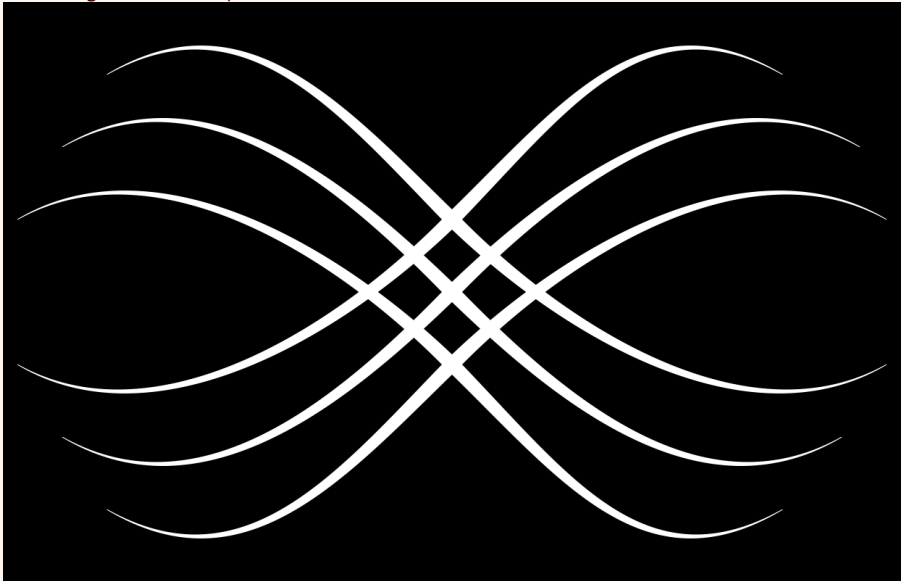
$\left[\langle \text{Maske}_{Mk} \rangle^{Mk} \right] \langle \text{Histogramm}_{Hg} \rangle (\text{Mk.niv} \mid \text{gleiche.an}): \left[\langle \text{Maske}_{Mk} \rangle^{Mk} \right]$

Gleicht eine Maske an ein kumulatives Histogramm an.⁴⁵

⁴⁵[4, S. 73]

Verarbeitung des Skriptes *zeichnung.lysa*: 0 • 1 • 1 • 20 • *Linear* (Verlauf) • Entwicklung generieren • in b deponieren • 0 • 1 • 20 • 1 • *Linear* (Verlauf) • Entwicklung generieren • in e deponieren • 'B' • Anfang • -30 • 220 • 45 • 220 • aus b verwenden • Spitzfeder ziehen • Ende • verbalisieren • 'E' • Anfang • 45 • 220 • -30 • 220 • aus e verwenden • Spitzfeder ziehen • Ende • verbalisieren • 1240 • 800 • Maske erschaffen • registrieren • invertieren • Maske verwenden • Zeichnung generieren • registrieren • 0 • 600 • versetzen • 143,6 • -500 • Dupel generieren • 620 • -300 • Dupel generieren • B • Länge: 566,972844276 • 620 • -300 • Dupel generieren • 1220 • -100 • Dupel generieren • E • Länge: 676,431327187 • 81,8 • -400 • Dupel generieren • 620 • -200 • Dupel generieren • B • Länge: 620,980653499 • 620 • -200 • Dupel generieren • 1158,2 • 0 • Dupel generieren • E • Länge: 620,980653499 • 20 • -300 • Dupel generieren • 620 • -100 • Dupel generieren • B • Länge: 676,431327187 • 620 • -100 • Dupel generieren • 1076,4 • 100 • Dupel generieren • E • Länge: 549,891606968 • 'B' • Anfang • 30 • 220 • -45 • 220 • aus b verwenden • Spitzfeder ziehen • Ende • verbalisieren • 'E' • Anfang • -45 • 220 • 30 • 220 • aus e verwenden • Spitzfeder ziehen • Ende • verbalisieren • 20 • -100 • Dupel generieren • 620 • -300 • Dupel generieren • B • Länge: 676,431327187 • 620 • -300 • Dupel generieren • 1076,4 • -500 • Dupel generieren • E • Länge: 549,891606968 • 81,8 • 0 • Dupel generieren • 620 • -200 • Dupel generieren • B • Länge: 620,980653499 • 620 • -200 • Dupel generieren • 1183,2 • -400 • Dupel generieren • E • Länge: 643,263122997 • 143,6 • 100 • Dupel generieren • 620 • -100 • Dupel generieren • B • Länge: 566,972844276 • 620 • -100 • Dupel generieren • 1220 • -300 • Dupel generieren • E • Länge: 676,431327187 • Zeichnung entnehmen • zeigen

Zeichnung: 1240×800 Bildpunkte



Eine Zeichnung repräsentiert eine Umgebung, in der geschlossene Flächen in Umrissen gezeichnet werden, um schließlich nicht mit Farbe gefüllt, sondern aus einer bei Erstellung der Zeichnung

übergebenen Maske ausgeschnitten zu werden. Dabei ist zu beachten, daß es auf die Richtung des Umrandungspfades ankommt, wenn sich Flächen überlappen. Ist der Pfad einer unteren Fläche im Uhrzeigersinn beschrieben und derjenige der oberen Fläche gegen den Uhrzeigersinn, dann macht die obere Fläche die untere unwirksam (**non-zero**).

Das alles hört sich zugegebenermaßen ein wenig nach dem Motto an: »warum einfach, wenn es auch kompliziert geht«, ist aber mächtig. Das Verfahren entspricht dem Zeichenmodell der Grafikbibliothek *Cairo*. Zumeist handelt es sich um Alphamasken, die derart behandelt werden, es geht aber auch mit jedem anderen Farbkanal.

⇒ nein • Q ungültig • Z Anzahl der Modifikationen am Koordinatensystem • ∞ ungültig

⊕ (u2316)	171	v	175
∞ (u238c)	174	verbinde.gerade	174
A	173	verbinde.krumm	174
annuliere	171	verbinde.würflig	175
d	171	versetze	175
drehe	171	verwirkliche	176
dsg	171	W	175
efv	176	ziehe.Breitfeder	177
ermittle.Bildkoordinaten	171	ziehe.Spitzfeder	178
G	174	Zn.ccl	171
g	172	Zn.fin	173
generiere.Zeichnung	171	Zn.hvt	172
K	174	Zn.kie	171
nimm.zurück	172	Zn.krb	174
O	173	Zn.pos	173
passe.an	172	Zn.rek	174
r	172	Zn.rpr	174
schere.horizontal	172	Zn.skl	172
schere.vertikal	173	Zn.stb	172
schließe.ab	173	Zn.trf	175
setze.an	173	Zn.trn	171
setze.zurück	174	Zn.trt	177
			Zn.ttt	178
			Zn.vtv	173

Zn.cancel
annuliere

(Englisch)

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \text{ (Zn.ccl | annuliere): } [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Beseitigt alle Modifikationen des Koordinatensystem der registrierten Zeichnung.

d
Zn.turn
drehe

(Englisch)

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Winkel}_Q \rangle \text{ (d | Zn.trn | drehe): } [\langle \text{Skizze}_{zn} \rangle^z]$

Dreht das Koordinatensystem der registrierten Zeichnung um den angegeben Winkel (Grad) im Uhrzeigersinn.

Modifikationen des Koordinatensystems ersetzen einander nicht, sondern bauen aufeinander auf. Zudem ist es möglich, sie einzeln oder auch insgesamt zurückzunehmen.

\oplus
Zn.kie
ermittle.Bildkoordinaten

(Strg+ \hat{u} +u2316)
(Esperanto)

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Zeichnungskoordinaten}_{dp} \rangle \text{ (} \oplus \text{ | Zn.kie | } \\ \text{ermittle.Bildkoordinaten): } \langle \text{Bildkoordinaten}_{dp} \rangle [\langle \text{Skizze}_{zn} \rangle^z]$

Wandelt Koordinaten der Zeichnung in Bildkoordinaten der zugrundeliegenden Maske um.

desegno
generiere.Zeichnung

(Esperanto)

$\langle \text{Material}_{mk} \rangle \text{ (dsg | generiere.Zeichnung): } \langle \text{Zeichnung}_{zn} \rangle$

Erstellt eine Zeichnung mit einer Maske als Ausgangsmaterial. Diese wird in ein eigenes Format überführt, spätere Änderungen an der aufgenommenen Maske bleiben also unberücksichtigt.

r

Zn.set back

(Englisch)

nimm.zurück

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \text{ (r | Zn.stb | nimm.zurück): } [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Macht die letzte Änderung am Koordinatensystem der registrierten Zeichnung rückgängig.

g

Zn.skali

(Esperanto)

passe.an

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{horizontal}_Q \rangle \langle \text{vertikal}_Q \rangle \text{ (g | Zn.skl | passe.an)} \\ [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Verändert den Maßstab der registrierten Zeichnung jeweils mit Faktoren für die horizontale und die vertikale Achse.

Zn.horizontale transvektion

schere.horizantal

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Winkel}_Q \rangle \text{ (Zn.htv | schere.horizantal): } [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Schert das Koordinatensystem der registrierten Zeichnung horizontal um den angegebenen Winkel.

Zn.vertikale transvektion schere.vertikal

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Winkel}_Q \rangle (\text{Zn.vtv} \mid \text{schere.vertikal}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Schert das Koordinatensystem der registrierten Zeichnung vertikal um den angegebenen Winkel.

O

Zn.fini

(Esperanto)

schließe.ab

$[\langle \text{Skizze}_{zn} \rangle^{zn}] (\text{O} \mid \text{Zn.fin} \mid \text{schließe.ab}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Schließt einen Umriß in der registrierten Zeichnung ab, indem eine gerade Linie vom letzten zum ersten Punkt gezogen wird.

A

Zn.position

(Englisch)

setze.an

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Position}_{dp} \rangle (\text{Zn.pos} \mid \text{setze.an}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$
 $[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle x_Q \rangle \langle y_Q \rangle (\text{A}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Beginnt einen neuen Umriß an der angegebenen Position in der registrierten Zeichnung.

Von diesem Verb gibt es zwei Versionen, von denen die erste die Position als Dupel, die weitere als Einzelwerte vom Stapel nimmt. Nur bei der ersten Version wird zudem beim ausführlichen Protokoll die aktuelle Position ausgegeben.

⌘

Zn.repreni
setze.zurück

(Strg+⬆+u238c)
(Esperanto)

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Position}_Z \rangle (\text{⌘} \mid \text{Zn.rpr} \mid \text{setze.zurück}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Stellt die Änderungen am Koordinatensystem der registrierten Zeichnung auf einen früheren Zustand zurück.

G

Zn.rekto
verbinde.gerade

(Esperanto)

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Ende}_{dp} \rangle (\text{Zn.rek} \mid \text{verbinde.gerade}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$
 $[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle x_Q \rangle \langle y_Q \rangle (\text{G}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Erweitert den Umriß in der registrierten Zeichnung um eine gerade Strecke von der aktuellen Position zu den genannten Koordinaten, die dann zur aktuellen Position werden.

Von diesem Verb gibt es zwei Versionen, von denen die erste die Position als Dupel, die weitere als Einzelwerte vom Stapel nimmt. Nur bei der ersten Version wird zudem beim ausführlichen Protokoll die aktuelle Position ausgegeben.

K

Zn.kurbo
verbinde.krumm

(Esperanto)

$[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle \text{Kontrolle}_{dp} \rangle \langle \text{Ende}_{dp} \rangle (\text{Zn.krb} \mid \text{verbinde.krumm}):$
 $[\langle \text{Skizze}_{zn} \rangle^{zn}]$
 $[\langle \text{Skizze}_{zn} \rangle^{zn}] \langle k-x_Q \rangle \langle k-y_Q \rangle \langle \text{end}-x_Q \rangle \langle \text{end}-y_Q \rangle (\text{K}): [\langle \text{Skizze}_{zn} \rangle^{zn}]$

Erweitert den Umriß in der registrierten Zeichnung beginnend an der aktuellen Position um eine quadratische Bézierkurve, deren Endpunkt zur neuen aktuellen Position wird.

Von diesem Verb gibt es zwei Versionen, von denen die erste die Positionen als Dupel, die weitere als Einzelwerte vom Stapel nimmt. Nur bei der ersten Version wird zudem beim ausführlichen Protokoll die aktuelle Position ausgegeben.

W

Zn.cubic bézier curve
verbinde.würflig

(Englisch)

$$\begin{aligned} & \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle 1. \text{Kontrolle}_{dp} \rangle \langle 2. \text{Kontrolle}_{dp} \rangle \langle \text{Ende}_{dp} \rangle \{ \text{Zn.cbc} \mid \\ & \text{verbinde.würflig} \}: \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \\ & \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle k1-x_Q \rangle \langle k1-y_Q \rangle \langle k2-x_Q \rangle \langle k2-y_Q \rangle \langle \text{end-}x_Q \rangle \langle \text{end-}y_Q \rangle \{ \text{W} \}: \\ & \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \end{aligned}$$

Erweitert den Umriß in der registrierten Zeichnung beginnend an der aktuellen Position um eine kubische Bézierkurve, deren Endpunkt zur neuen aktuellen Position wird.

Von diesem Verb gibt es zwei Versionen, von denen die erste die Positionen als Dupel, die weitere als Einzelwerte vom Stapel nimmt. Nur bei der ersten Version wird zudem beim ausführlichen Protokoll die aktuelle Position ausgegeben.

v

Zn.transfer
versetze

(Englisch)

$$\begin{aligned} & \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle \text{Versatz}_{dp} \rangle \{ \text{Zn.trf} \mid \text{versetze} \}: \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \\ & \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle \text{horizontal}_Q \rangle \langle \text{vertikal}_Q \rangle \{ \text{v} \}: \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \end{aligned}$$

Versetzt den Ursprung des Koordinatensystem der registrierten Zeichnung.

Von diesem Verb gibt es zwei Versionen, von denen die erste die Position als Dupel, die weitere als Einzelwerte vom Stapel nimmt. Nur bei der ersten Version wird zudem beim ausführlichen Protokoll die aktuelle Position ausgegeben.

⟨Entwurf_{zn}⟩ (efv | verwirkliche): ⟨Realisierung_{mk}⟩

Verwirklicht eine Zeichnung in einer Maske. Dabei werden alle als zu füllend gezeichneten Flächen aus der beim Anlegen der Zeichnung übergebenen Maske ausgeschnitten und in die zu schaffende Maske übertragen. Ansonsten bleibt diese schwarz.

Schreibfedern

Breit- oder auch Bandzugfedern werden hauptsächlich in der Kalligraphie verwendet und ermöglichen durch ihre Form Striche richtungsabhängiger Stärke. Die jeweils maximal erreichbare Strichstärke heißt Federbreite. Diese wird auch als Maß für die Schriftgröße verwendet. Als Grundregel gilt, daß die Majuskelhöhe acht Federbreiten und die Minuskelhöhe deren fünf beträgt.^a

Breitfedern waren ursprünglich breitkantig zugeschnittene Federkiele. Die seit Mitte des 19. Jahrhunderts erhältlichen **Stahlfedern** gibt es mit gerader oder schräger Kante, wobei sowohl die geraden als auch die rechts abfallenden Kanten vor allem für Rechtshänder, die mit nach links abfallenden Kanten vor allem für Linkshänder geeignet sind.^b

Beim Schreiben liegt die Kante voll auf dem Papier auf, was scharfkantige Striche ergibt. Zur Zeilenrichtung bildet der Federhalter einen Winkel zwischen 30 und 40 Grad. In der Regel wird dieser Winkel während des gesamten Schreibvorgangs beibehalten, wodurch sich breitere senkrechte als waagrechte Linien ergeben.^c

Striche werden stets von oben nach unten sowie von links nach rechts gezogen. Die meisten Buchstaben werden darum bei Verwendung einer Breitfeder mit mehreren Strichen geschrieben. Demgegenüber ermöglicht die im Barock aufgekommene Spitzfeder das Schreiben in einer einzigen Bewegung. Auch sie ermöglicht verschiedene Strichstärken, allerdings nicht durch ihre Form, sondern durch variierenden Druck auf die Feder. Wird eine Spitzfeder beim Schreiben stärker auf das Papier gedrückt, spreizt sich die gespaltene Spitze, mehr Tinte gelangt auf das Papier, und der Strich wird breiter. Dies

wird insbesondere bei geraden Abstrichen genutzt. Bei Aufstrichen darf nur leichter Druck auf die Feder ausgeübt werden, da sie sich ansonsten in das Papier bohren kann.^d

^a[8, S. 38]

^b[8, S. 23]

^c[8, S. 22 f.]

^d[8, S. 82 f.]

Zn.trait ziehe.Breitfeder

(Französisch)

$$\left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle \text{Startpunkt}_{dp} \rangle \langle \text{Endpunkt}_{dp} \rangle \langle \text{horizontal}_{vl} \rangle \langle \text{vertikal}_{vl} \rangle$$

$$\langle \text{Federbreite}_{ew} \rangle \langle \text{Federwinkel}_{ew} \rangle \{ \text{Zn.trt} \mid \text{ziehe.Breitfeder} \}:$$

$$\left[\langle \text{Skizze}_{zn} \rangle^{zn} \right]$$

Simuliert in der registrierten Zeichnung einen Strich mit einer Schreibfeder. Die Entwicklungen werden auf eine Abzisse von Null bis Eins übertragen (S. 91).

Die Drehung der Feder ist in Grad anzugeben: 0 Grad gehen waagrecht von links nach rechts (3 Uhr), 90 Grad senkrecht von oben nach unten (6 Uhr). Wenn der Endwinkel größer ist als der Anfangswinkel geht die Drehung im Uhrzeigersinn, ansonsten dagegen.

Das Verb ist insbesondere zum Ziehen kalligraphischer Striche gedacht. Da sowohl der Winkel der Feder als auch die Dicke des Strichs anzugeben sind, lassen sich mit ihm sowohl eine Breit- als auch eine Spitzfeder nachahmen. Insbesondere ist aber an erstere gedacht.

$\left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle \text{Startpunkt}_{dp} \rangle \langle \text{Endpunkt}_{dp} \rangle \langle \text{Ausgangswinkel}_Q^{[0-360]} \rangle$
 $\langle \text{Distanz zum Startpunkt}_Q \rangle \langle \text{Eingangswinkel}_Q^{[0-360]} \rangle \langle \text{Distanz zum}$
 $\text{Endpunkt}_Q \rangle \langle \text{Federbreite}_{ew} \rangle (\text{Zn.ttt} \mid \text{ziehe.Spitzfeder}): \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right]$

Simuliert in der registrierten Zeichnung einen Strich mit einer Spitzfeder. Die Bewegung wird bei dieser Variante über eine kubische Bézierkurve gesteuert. Die Kontrollpunkte sind zum einen über den Ausgangs- beziehungsweise Eingangswinkel und die Distanz zum Start- respektive Endpunkt anzugeben. Zudem steht der Federwinkel bei dieser Variante stets senkrecht zur Richtung des Strichs.

Zu beachten ist, daß die angegebenen Winkel die Richtungen angeben, in der die Feder gezogen wird, beziehungsweise in der sie ankommt.

Verarbeitung des Skriptes *schrift.lysa*: `'/usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz'` •
einbeziehen • *brandvogel-r-400* (Schrift) • **zeigen**

Schrift:

Glyphen:	10536
Maximale Höhe:	0,936□
Maximale Tiefe:	0,551□
Minuskelhöhe:	0,45□
Majuskelhöhe:	0,662□
Unterlänge:	0,2□
Leerzeichen:	0,25□

Hamburgefonstiv

Informationen zum regulären Schnitt der Schrift *FreeSerif* (© 2002 – 2012 bei GNU; GNU General Public License 3); wird in allen Beispielen in diesem Handbuch verwendet.

Schrift repräsentiert einen einzelnen Schriftschnitt. Sie enthält sowohl allgemeine Informationen über die Schrift als auch die einzelnen Schriftzeichen.

Wenn eine solche angefordert wird, liefert eine Schrift keine Kopie von sich, sondern nur einen Verweis auf sich selbst.

↔ nein • \mathbb{Q} ungültig • \mathbb{Z} Anzahl der Glyphen • \mathbb{N} ungültig

erschaffe.Schrift	180	<i>Sf.ltr</i>	179
erweitere.Schriftprobe	181	<i>Sf.scm</i>	181
greife	179	<i>tpr</i>	180

Sf.litero
greife

(Esperanto)

$\left[\langle \text{Schrift}_{sf} \rangle^{sf} \right] \langle \text{Bezeichnung}_{zk} \rangle (\text{Sf.ltr} \mid \text{greife}): \langle \text{Glyph}_{sz} \rangle \left[\langle \text{Schrift}_{sf} \rangle^{sf} \right]$

Liefert aus der registrierten Schrift ein bestimmtes Schriftzeichen. Wird es nicht gefunden, landet ein Verweis auf den *Fliegenkopf* auf dem Stapel. Sollte auch dieser nicht gefunden werden können, gibt es eine Fehlermeldung nebst Programmabbruch.

Fliegenkopf

Fehlte im Handsatz ein Buchstabe, wurde ein in der Dichte entsprechender anderer verkehrtherum gesetzt, um später gegen einen richtigen ausgetauscht zu werden. Dieses Vorgehen wurde Blockade genannt. Im Andruck ergaben solche Buchstaben einen Klecks (in etwa ■), der *Fliegenkopf* genannt wurde.^a Diese Bezeichnung erscheint mir passend für das in jeder Schrift enthaltene Zeichen für nicht vorhandene Buchstaben.

^a[7, S. 71]

tiparo

(Esperanto)

erschaffe.Schrift

⟨maximale Höhe_Q⟩ ⟨maximale Tiefe_Q⟩ ⟨Minuskelhöhe_Q⟩ ⟨Majuskelhöhe_Q⟩
⟨Unterlänge_Q⟩ ⟨Leerzeichenbreite_Q⟩ (tpr | erschaffe.Schrift): ⟨Schrift_{sf}⟩

Erzeugt eine leere Schrift und legt eine Referenz auf den Stapel. Alle Angaben beziehen sich auf das Geviert.

Geviert

Das Geviert ist die grundlegende typografische Größenangabe. Ursprünglich entsprach es dem Kegel^a (Kegelstärke^b) und diente als ein somit der Schriftgröße sowohl in der Höhe als auch in der Dichte entsprechendes Blindmaterial lediglich als Breitenangabe. Wenn es um den Grad (Größe) einer Schrift ging, war der Kegel gemeint (Abweichungen wurden als solche angegeben: etwa Borgis auf Korpus oder Kolonel auf Petit^c).

Im Computersatz entspricht der Kegel (also die tatsächliche Schriftgröße) der maximalen vertikalen Ausdehnung (bounding box), wo-

hingegen das Geviert nur mehr eine vom Schriftdesigner gesetzte Größe darstellt. Dennoch wird im Computersatz nun allgemein bei der Angabe der Schriftgröße das Geviert bestimmt. Der Kegel spielt keine Rolle mehr. Das geht soweit, daß das Wissen um seine Bedeutung verloren zu gehen droht.^d

Zumindest im englischsprachigen Satz leitet sich das Geviert (quad, abgekürzt em) traditionell von der Breite des Großbuchstabens M ab, was aber auf die meisten Schriften auch schon lange nicht mehr zutrifft.

^a[11, S. 76]

^b[19, S. 53]

^c[11, S. 79]

^d»Die Dichte, auch als Schriftkegel bezeichnet...« [1, Nr. 18 v.]

Sf.specimeno

(Esperanto)

erweitere.Schriftprobe

$$\left[\langle \text{Schrift}_{sf} \rangle^{sf} \right] \langle \text{Zeichen}_{zk} \rangle \{ \text{Sf.scm} \mid \text{erweitere.Schriftprobe} \} : \left[\langle \text{Schrift}_{sf} \rangle^{sf} \right]$$

Fügt zum Probetext der registrierten Schrift ein Zeichen hinzu. Er wird in der HTML-Datei ausgegeben, wenn Informationen über eine Schrift gezeigt werden.

Verarbeitung des Skriptes *schriftzeichen.lysa*: '/usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz' •
einbeziehen • *brandvogel-r-400* (Schrift) • registrieren • 'Fliegenkopf' • greifen • zeigen

Schriftzeichen: Fliegenkopf (Dicke: 0,6□)



Repräsentiert ein einzelnes Schriftzeichen. Ein Schriftzeichen steht stets in Zusammenhang mit einer Schrift (s. o.), die es enthält. Identifiziert wird es über seine Bezeichnung (Zeichenkette).

Wenn eine solche angefordert wird, liefert ein Schriftzeichen keine Kopie von sich, sondern nur einen Verweis auf sich selbst.

↔ nein • Q Dicke • Z ungültig • ∞ Dicke

⊞ (u2384) 187	setze 187
beginne.Zeile 182	Sf.gly 184
generiere.Schriftzeichen 184	unterschneide 187
korrigiere 184	Zn.adj 187
krm 184	Zn.kmp 187
	Zn.ora 182

Zn.optischer randausgleich
beginne.Zeile

$[\langle Skizze_{zn} \rangle^{zn}] \langle Glyphes_z \rangle \{Zn.ora \mid beginne.Zeile\}: [\langle Skizze_{zn} \rangle^{zn}]$

Überprüft, ob bei einem Zeichen ein linker Randausgleich gespeichert ist, und versetzt gegebenenfalls das Koordinatensystem der registrierten Zeichnung entsprechend.

Glatte Satz

Beim glatten Satz wird die Zeilenlänge gleichmäßig ausgeschlossen. Davon ausgenommen ist die letzte Zeile eines Absatzes (Ausgang). Fälschlicher-, doch heute geradezu üblicherweise^a wird beim glatten Satz auch gerne von Blocksatz gesprochen. Diesen zeichnet gegenüber dem glatten Satz aus, daß ausnahmslos alle Zeilen ausgeglichen werden, um mit dem Text den Satzspiegel auszufüllen.^b

Um den Text einer Zeile auf gleiche Länge zu bringen, ist der Ausschluß (S. 193) zwischen den Wörtern entweder zu verringern oder zu erweitern. Da dies im Handsatz nicht in beliebigen Bruchteilen erfolgen konnte, galten folgende Prioritäten für das Verringern (für das Erweitern einfach die Reihenfolge umkehren):

<i>Situation</i>	1935	1956
Vor Großbuchstaben	2	1
Nach einem Komma	5	2
Vor Wörtern, deren Anfangsbuchstaben (Minuskeln)	3	3
Ober- oder Unterlängen aufweisen		
Vor Wörtern mit Anfangsbuchstaben (Minuskeln)	4	4
ohne Ober- oder Unterlängen		
Nach einem Doppelpunkt, Semikolon und nichtsatzschließenden Frage- oder Ausrufezeichen	6	5
Am Satzschluß	1	6

Anzumerken ist hierzu allerdings, daß 1935 von einem doppelten Zwischenraum nach einem Satzschluß ausgegangen wurde,^c wohingegen 1956 nur mehr ein normaler Ausschluß zu setzen war.^d

Für das Verringern des Ausschlußes findet sich auch die Bezeichnung einbringen, wie für das Erweitern ausbringen;^e an anderer Stelle wird allerdings als Einbringen das Verringern des Ausschlußes bezeichnet, um noch ein Zeilenstück in die Zeile zu bringen, als Ausbringen dementsprechend das Erweitern, um ein Zeilenstück aus der Zeile herauszubringen (meist um einen Ausgang zu schinden).^f

^a[12, S. 16]

^b»Blocksatz«, bei dem Satzgruppen aus rein dekorativen Gründen zu starren Rechtecken geformt werden, hat damit nichts zu tun.« [11, S. 160]
^c[7, S. 66]
^d[11, S. 120]
^e[11, S. 117], [17, S. 41]
^f[7, S. 70]

Sf.glyphe generiere.Schriftzeichen

$\left[\langle \text{Schrift}_{sf} \rangle^{sf} \right] \langle \text{Bezeichnung}_{zk} \rangle \langle \text{Dicke}_Q \rangle \langle \text{linker Randausgleich}_Q \rangle \langle \text{rechter Randausgleich}_Q \rangle \langle \text{Zeichnung}_{Af} \rangle \langle \text{Sf.gly | generiere.Schriftzeichen} \rangle$
 $\langle \text{Schriftzeichen}_{sz} \rangle \left[\langle \text{Schrift}_{sf} \rangle^{sf} \right]$

Erzeugt ein Schriftzeichen in der registrierten Schrift. Alle Zahlenangaben beziehen sich ebenfalls auf das Geviert.

Die Angaben für den Randausgleich spielen beim glatten Satz und dem optischen Randausgleich eine Rolle.

kerning korrigiere

(Englisch)

$\langle \text{Schriftzeichen}_{sz} \rangle \langle \text{Vorgänger}_{zk} \rangle \langle \text{Korrektur}_Q \rangle \langle \text{krn | korrigiere} \rangle$
 $\langle \text{Schriftzeichen}_{sz} \rangle$

Fügt einem Schriftzeichen eine Information zur Unterschneidung bei einem bestimmten Vorgängerzeichen hinzu. Die Zahlenangabe bezieht sich auf das Geviert.

hängende Interpunktion / optischer Randausgleich

Als hängende Interpunktion wird bezeichnet, wenn Satzzeichen zwecks optischem Randausgleich teilweise oder sogar ganz über das eigentliche Zeilenende hinausragen. Derart soll beim glatten Satz

(S. 183) eine Unruhe am rechten Textrand vermieden werden. Der optische Randausgleich bildet dabei sozusagen den Oberbegriff und die Unruhevermeidung betrifft dann auch den linken Rand.

Das erste Mal habe ich in der Druckerwerkstatt der Kunstakademie Düsseldorf von deren damaligen Leiter Hans Möller (Hofhaus Presse) von hängenden Satzzeichen gehört (1988). Das war etwas, was Desktop-Publishing-Programme nicht beherrschten, was aber zum alten Handwerkswissen der Setzerei gehören sollte und bei einem guten Bleisatz nicht fehlen durfte (laut Hans Möller).

Als dann **Hàn Thé Thành** mit pdfT_EX das *protruding* zum optischen Randausgleich einführte, war mir das zwar hochwillkommen, denn es ermöglichte mir endlich eine hängende Interpunktion, aber weitergehend habe ich mich damit nicht beschäftigt.

Doch als ich nun in meinen Quellen nach dem *alten Handwerkswissen* suchte, stellte sich das als eher schwierig dar. Es muß sich dann doch um altes Geheimwissen handeln, denn weder im DAS BUCH DES SETZERS^a noch im DAS SETZERLEHRBUCH findet sich ein Hinweis darauf, wobei letzteres immerhin beim Gedichtsatz oder bei Aufzählungen, wo gewöhnlich alle Zeilen mit einem Großbuchstaben beginnen, einen optischen Randausgleich für die linke Kante fordert.^b Lediglich Jan Tschichold spricht in einem der mir vorliegenden Bücher davon, daß er die Trennstriche um 2 Punkt über das eigentliche Zeilenende hat hinausragen lassen.^c Von anderen Satzzeichen (insbesondere Punkt und Komma) ist da keine Rede und sie ragen auch, soweit ich das erkennen kann, nicht über das Zeilenende hinaus.

Einige digitale Setzsysteme, die Ende der achtziger, Anfang der neunziger Jahre des 20. Jahrhunderts erschienen (namentlich von Linotype und Berthold) sollen neben dem Kerning auch Funktionen zum optischen Ausrichten sowohl des linken und rechten Textrandes zur Verfügung gestellt haben. Das erste Computerprogramm, das eine hängende Interpunktion unterstützt haben soll, soll Adobes INDESIGN gewesen sein (erschieden 31. August 1999).^d

Wie weit hängen aber denn nun die einzelnen Satzzeichen über das Zeilenende hinaus? Trennstriche sollen komplett außerhalb des Textblocks positioniert werden, Gedankenstriche ragen zur Hälfte aus diesem heraus. Punkte und Kommas werden außerhalb des rechten Randes positioniert (wie weit?).^e

Präziser ist demgegenüber Hàn Thé Thành in seiner Dissertation^f:

Zeichen	Überhang
Punkt	0,70×Zeichenbreite
Trennstrich	0,70×Zeichenbreite
Komma	0,70×Zeichenbreite
quoteright	0,70×Zeichenbreite
quotedblright	0,50×Zeichenbreite
Semikolon	0,50×Zeichenbreite
Doppelpunkt	0,50×Zeichenbreite
Gedankenstrich	0,30×Zeichenbreite
Geviertstrich	0,20×Zeichenbreite
Ausrufezeichen	0,20×Zeichenbreite
Fragezeichen	0,20×Zeichenbreite
Klammer.zu	0,05×Zeichenbreite
A	0,05×Zeichenbreite
F	0,05×Zeichenbreite
K	0,05×Zeichenbreite
L	0,05×Zeichenbreite
T	0,05×Zeichenbreite
V	0,05×Zeichenbreite
W	0,05×Zeichenbreite
X	0,05×Zeichenbreite
Y	0,05×Zeichenbreite
k	0,05×Zeichenbreite
r	0,05×Zeichenbreite
t	0,05×Zeichenbreite
v	0,05×Zeichenbreite
w	0,05×Zeichenbreite
x	0,05×Zeichenbreite
y	0,05×Zeichenbreite

Zugleich gibt er als optischen Ausgleich für den linken Rand folgende Werte an:

Zeichen	Überhang
Anführungszeichen.einfach.oben	-0,70×Zeichenbreite
Anführungszeichen.oben	-0,50×Zeichenbreite
Klammer.auf	-0,05×Zeichenbreite
A	-0,05×Zeichenbreite
J	-0,05×Zeichenbreite
T	-0,05×Zeichenbreite
V	-0,05×Zeichenbreite
W	-0,05×Zeichenbreite
X	-0,05×Zeichenbreite
Y	-0,05×Zeichenbreite
v	-0,05×Zeichenbreite
w	-0,05×Zeichenbreite
x	-0,05×Zeichenbreite
y	-0,05×Zeichenbreite

^a[7]

^b[11, S. 180]

^c[19, S. 16]

^d[16, S. 203]

^e[16, S. 204]

^fMicro-typographic extensions to the T_EX typesetting system, S. 43

$$\left[\langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle \text{Glyphe}_{sz} \rangle (\text{Zn.kmp} \mid \text{setze}): \left[\langle \text{Bezeichnung}_{zk} \rangle^{zk} \right. \\ \left. \langle \text{Skizze}_{zn} \rangle^{zn} \right]$$

Setzt ein Schriftzeichen am Ursprung des Koordinatensystems in die registrierte Zeichnung und verschiebt hernach den Ursprung um die Breite des Zeichens.

Die Bezeichnung des gesetzten Zeichens wird ins Zeichenkettenregister gelegt.

Ⓢ

Zn.adj
just
unterschneide

(Strg+⤴+u2384)
(Englisch)

$$\left[\left(\langle \text{Bezeichnung}_{zk} \rangle^{zk} \right) \langle \text{Skizze}_{zn} \rangle^{zn} \right] \langle \text{Glyphe}_{sz} \rangle (\text{Ⓢ} \mid \text{Zn.adj} \mid \\ \text{unterschneide}): \langle \text{Glyphe}_{sz} \rangle \left[\langle \text{Skizze}_{zn} \rangle^{zn} \right]$$

Schaut sich ein Schriftzeichen auf dem Stapel an und gegebenenfalls die Bezeichnung eines Vorgängers im Zeichenkettenregister. Insofern dazu eine Unterschneidung bei dem Schriftzeichen gespeichert ist, wird der Ursprung des Koordinatensystems der registrierten Zeichnung entsprechend versetzt.

Befindet sich keine Bezeichnung im Regal oder ist keine Unterschneidung gespeichert, geschieht nichts.

Diese Funktion hätte auch im Verb **setze** (S. 187) mit erledigt werden können (und tatsächlich platziert dieses im passenden Regalfach die für diese Funktion notwendige Information). Da aber so wenig wie möglich automatisch erledigt werden soll, handelt es sich um getrennte Handlungen. Nichts spricht allerdings dagegen, selbst die Schritte zusammenzufassen:

$\left[\left[\text{Schrift}_{sf} \right]^{sf} \left(\left[\text{Vorgänger}_{zk} \right]^{zk} \right) \left[\text{Skizze}_{zn} \right]^{zn} \right] \left[\text{Bezeichnung}_{zk} \right] \langle \text{!} \rangle$
 $\langle \langle (\text{Sf.ltr}) (\text{!}) (\text{Zn.kmp}) \rangle \rangle \left[\left[\text{Schrift}_{sf} \right]^{sf} \left[\text{Bezeichnung}_{zk} \right]^{zk} \left[\text{Skiz-} \right. \right.$
 $\left. \text{ze}_{zn} \right]^{zn} \left. \right]$


Verarbeitung des Skriptes *test.lysa*: '/usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz' •
einbeziehen • *brandvogel-r-400* (Schrift) • registrieren • 0 • 0 • Dupel generieren • 'linksbündig' • 30 • 30 •
1,33 • Kolumne erschaffen • registrieren • Zeile erschaffen • 0 • 60 • Sammlung erschaffen •
verstummen lassen • 'r' • 'o' • sammeln • letzte Letter herausgeben • zeigen

Letter: o (Dickte: 0,491)



Repräsentiert ein Schriftzeichen im Zusammenhang (S. 192) einer Kolumne. Die Gegebenheitsart kann nicht für sich allein bestehen, sondern muß zwingend einem inneren Zusammenhang zugeordnet sein, der also keinen Wortzwischenraum repräsentieren darf. Es ist erforderlich, daß sich ein solcher Zusammenhang im Register befindet, wenn eine neue Letter erschaffen wird. Diese wird dem Zusammenhang dann automatisch zugeordnet.

↪ nein • Q ungültig • Z ungültig • Œ ungültig

 (u2328)	190	Zh.agl	190
gleiche.ans	190	Zh.asg	189
sammle	190	Zh.row	190
schlage.an	189		

Zh.anschlagen schlage.an

$$\left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \langle \text{Schrift}_{sf} \rangle^{sf} \left(\perp \text{Effekte}_{sl}^{ef} \perp^{sl} \right) \right] \langle \text{Zeichen}_{zk} \rangle \{ \text{Zh.asg} | \text{schlage.an} \} : \left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \langle \text{Schrift}_{sf} \rangle^{sf} \left(\perp \text{Effekte}_{sl}^{ef} \perp^{sl} \right) \right]$$

Fügt dem registrierten Zusammenhang eine Letter hinzu. Sollte es sich dabei um einen Wortzwischenraum handeln, wird das Programm mit einer Fehlermeldung abgebrochen. Aus der registrierten Schrift wird das angegebene Zeichen (falls nicht vorhanden ein Fliegenkopf) für die Letter genommen. Zudem wird eine registrierte Sammlung von Effekten erwartet. Falls das Register leer sein sollte, wird eine leere Sammlung geschaffen und verwendet, was keinen Effekten entspricht. Insofern eine Sammlung zwar vorhanden ist, jedoch keine Effekte enthält, wird sie ignoriert, was ebenso zu keinen Effekten führt.

Zh.ausgleichen
gleiche.aus

[$\langle \text{Zusammenhang}_{zh} \rangle^{zh}$] (Zh.agl | gleiche.aus): [$\langle \text{Zusammenhang}_{zh} \rangle^{zh}$]

Gleicht gegebenenfalls den Abstand des zuletzt zum registrierten Zusammenhang hinzugefügten Letters zum Vorgängerzeichen aus. Gibt es keine solchen Zeichen in dem Zusammenhang, bleibt alles unverändert.



Zh.row
sammler

(Strg+↑+u2328)
(Englisch)

$$\left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \langle \text{Schrift}_{sf} \rangle^{sf} \left(\perp \text{Effekte}_{sl}^{Ef} \right)^{sl} \right] \left(\langle \text{Zeichen}_{zk} \rangle \right) \left(\text{Zeichen}_{zk} \right) \mid \text{Zh.row} \mid \text{sammle} \rangle: \left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \langle \text{Schrift}_{sf} \rangle^{sf} \left(\perp \text{Effekte}_{sl}^{Ef} \right)^{sl} \right]$$

Liest Zeichenketten ein, solange auf dem Stapel solche unterbrechungsfrei zu finden sind, sucht zu jeder von ihnen ein Schriftzeichen aus der registrierten Schrift und ordnet dieses als Letter dem registrierten Zusammenhang zu, wobei gegebenenfalls die registrierte Sammlung die Effekte für jede einzelne Letter enthält. Eine Unterschneidung findet dabei automatisch bei jeder einzelnen Letter statt.

Das Verb ist eine Zusammenfassung von anschlagen (S. 189) und ausgleichen (S. 190), welche die Eingabe mehrerer Zeichen vereinfacht. Eigentlich müßten die Bezeichnungen der einzelnen Zeichen in umgekehrter Reihenfolge angegeben werden, da das aber umständlich und fehlerträchtig ist, wird das Verkehren der Reihenfolge intern erledigt.

Verarbeitung des Skriptes *test.lysa*: '/usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz' •
einbeziehen • brandvogel-r-400 (Schrift) • registrieren • 0 • 0 • Dupel generieren • 'linksbündig' • 30 • 30 •
1,33 • Kolumne erschaffen • registrieren • Zeile erschaffen • 0 • 60 • Sammlung erschaffen •
verstummen lassen • 'E' • 'r' • 's' • 'c' • 'h' • 'r' • 'o' • 'c' • 'k' • 'e' • 'h' • sammeln • Zusammenhang
verwenden • zeigen

Stumm:

Breite:	4,895□
Minimum:	98%
Maximum:	102%
Erscheinungsbeginn:	0 Λ
Erscheinungsende:	60 Λ

Erschrocken

Ein Zusammenhang repräsentiert entweder eine Ansammlung von Drucktypen (innerer Zusammenhang) oder (exklusiv) einen Zwischenraum zwischen anderen Zusammenhängen in einer Zeile (äußerer Zusammenhang). Auch diese Gegebenheitsart kann nicht für sich bestehen, sondern muß zwingend einer Zeile zugeordnet sein. Insofern ist es erforderlich, daß sich eine solche im Register befindet, wenn ein neuer Zusammenhang erschaffen wird. Der Zusammenhang wird dieser dann automatisch zugeordnet.

↪ nein • Q ungültig • Z ungültig • ⚡ ungültig

(..... (u23dc) 196	generiere.Ausschluß 193
} (u23de) 195	gib.Ende 195
(..... (u23e0) 195	Ln.brs 196
_ (u2423) 193	Ln.cop 195
füge.Doppelspatium.ein 195	Ln.led 195
füge.Haarspatium.ein 195	Ln.spc 193
füge.Spatium.ein 196	Zh.lst 195

$\left[\langle Zeile_{Ln} \rangle^{Ln} \right] \langle normale\ Breite_Q \rangle \langle Faktor\ für\ Minimum_Q^{\leq 1} \rangle \langle Faktor\ für\ Maximum_Q^{\geq 1} \rangle \{ _ | Ln.spc | generiere.Ausschluß \} : \\ \left[\langle Zusammenhang_{zh} \rangle^{zh} \langle Zeile_{Ln} \rangle^{Ln} \right]$

Erschafft einen Wortzwischenraum als Zusammenhang in der registrierten Zeile. Anzugeben ist die normale Breite in Geviert, für das Minimum und Maximum Faktoren für die normale Breite. Über die letzten beiden Werte wird gesteuert, ob es sich um einen variablen (mindestens ein Wert $\neq 1$) oder um einen festen (beide Werte = 1) Wortzwischenraum handelt.

Um die Schreibarbeit zu reduzieren, empfiehlt es sich, für den Standardwortzwischenraum ein Makro zu erstellen (z. B. unter u23b4):

$\langle _ \rangle \langle \langle \langle 0,33 \rangle \langle 0,67 \rangle \langle 1,5 \rangle \{ _ \} \rangle \langle ! \rangle$.

Ausschluß

Als Ausschluß wird der Zwischenraum zwischen Wörtern bezeichnet.

Welcher Zwischenraum als angemessen erachtet wird, unterliegt modischen Strömungen. Schätzte der frühe Druck ein enges und geschlossenes Satzbild, wandelte sich der Geschmack um 1800 hin zu einer weiten und lichten Erscheinung. Ausschluß von $\frac{1}{2}$ Geviert, bei festlichen Gelegenheiten gar von einem ganzen Geviert (S. 180) waren nicht unüblich.^a

Als normaler Wortzwischenraum galt um die Mitte des letzten Jahrhunderts und somit in der Endzeit des Bleisatzes, $\frac{1}{3}$ Geviert.^b Diesen empfahl auch Jan Tschichold 1960,^c obgleich er in den 1947 für die *Penguin Books* aufgestellten Satzregeln zu einen geringeren Wortabstand geraten hatte.^d

Heute üblich ist eher ein Wortabstand von $\frac{1}{4}$ Geviert, wenn nicht

sogar weniger.^e

Daß heute wieder ein enges und geschlossenes Satzbild bevorzugt wird, könnte auch daran liegen, daß im Computersatz ein Unterschneiden üblich ist, welches dem Bleisatz schlicht nicht oder nur unter unverhältnismäßig hohem Aufwand möglich war (Abfeilen der Lettern). Im Handsatz beschränkte sich das Ausgleichen innerhalb eines Wortes in der Regel auf ein Hinzufügen von Fleisch,^f wohingegen im Computersatz die Schriftbilder der Wörter kompakter geworden sind. Dem entspricht natürlich auch eher wieder ein geringerer Wortzwischenraum.

Andererseits empfahl Tschichold eventuell nicht umsonst den Viertelsatz für englischsprachigen und den Drittersatz für deutschen Text. Ein Text, in dem Großbuchstaben seltener vorkommen, trägt einen engeren Satz umso besser.

Letztlich kommt es auf den Charakter der verwendeten Schrift an. Wirkt diese kompakt, empfiehlt sich ein Viertelsatz, bei weiter Schrift hingegen eher ein Drittersatz.

Die bei einer Schrift vom Designer angegebene Leerzeichenbreite ergibt üblicherweise einen zu engen Satz. Die angegebene Größe taugt meines Erachtens allenfalls als Angabe eines minimalen Wortabstands.

^a[11, S. 143 f.]

^b»Als normalen Wortzwischenraum nimmt man zumeist etwa ein Drittel der Kegelgröße. ›Drittersatz‹, vor Jahrzehnten noch von anspruchsvollen Verlegern in Ausnahmefällen gefordert, ist heute beim Buchsatz eine Selbstverständlichkeit geworden.« [11, S. 116]

^c»Tadelloser Ausschluß (sogenannter Drittersatz) ist ein Prüfstein der Qualität einer Satzarbeit.« [19, S. 66]

^d»Im durchschnitt soll mit viertelgevierten ausgeschlossen werden« [6, S. 26]

^e»Der normale Wortzwischenraum »ist in den meisten Schriften so breit wie ein Viertel der gewählten Schriftgröße« [12, S. 15]

^f»Auch das Ausgleichen bedeutet im allgemeinen eine Vergrößerung der Zwischenräume zwischen den Buchstaben. Es ist notwendig, weil in gewissen Buchstabenverbindungen, besonders bei Versalien, der gleichmäßige Fluß gestört ist, auch wenn die Schrift noch so sorgfältig zugerichtet ist.« [11, S. 178]

Zh.last
gib.Ende

(Englisch)

$$\left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \right] (\text{Zh.lst} \mid \text{gib.Ende}): \left(\langle \text{Zeichen}_{lt} \rangle \right) \\ \left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \right]$$

Legt die letzte Letter eines nicht leeren registrierten inneren Zusammenhangs auf den Stapel. Ansonsten geschieht nichts.

⌘

(Strg+⌘+u23de)

Ln.lead

(Englisch)

füge.Doppelspatium.ein

$$\left[\langle \text{Zeile}_{ln} \rangle^{ln} \right] (\text{⌘} \mid \text{Ln.led} \mid \text{füge.Doppelspatium.ein}): \\ \left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \langle \text{Zeile}_{ln} \rangle^{ln} \right]$$

Erschafft einen festen Zwischenraum von $\frac{1}{16}$ Geviert als äußeren Zusammenhang in der registrierten Zeile.

⌘

(Strg+⌘+u23e0)

Ln.copper

(Englisch)

füge.Haarspatium.ein

$$\left[\langle \text{Zeile}_{ln} \rangle^{ln} \right] (\text{⌘} \mid \text{Ln.cop} \mid \text{füge.Haarspatium.ein}): \\ \left[\langle \text{Zusammenhang}_{zh} \rangle^{zh} \langle \text{Zeile}_{ln} \rangle^{ln} \right]$$

Erschafft einen festen Zwischenraum von $\frac{1}{24}$ Geviert als äußeren Zusammenhang in der registrierten Zeile.

⌞

Ln.brass

füge.Spatium.ein

(Strg+⌞+u23dc)
(Englisch)

$\left[\langle Zeile_{Ln} \rangle^{Ln} \right] (\lceil \mid Ln.brs \mid füge.Spatium.ein): \left[\langle Zusammenhang_{zh} \rangle^{zh} \langle Zeile_{Ln} \rangle^{Ln} \right]$

Erschafft einen festen Zwischenraum von $\frac{1}{16}$ Geviert als äußeren Zusammenhang in der registrierten Zeile.

<

Ln.deklam

laß.verlauten

(Strg+⌞+u227a)
(Esperanto)

$\left[\langle Zeile_{Ln} \rangle^{Ln} \right] \langle Erscheinen_{zr} \rangle \lceil Effekte_{sl}^{Ef} \lceil (< \mid Ln.dkm \mid laß.verlauten): \left[\langle Zusammenhang_{zh} \rangle^{zh} \langle Zeile_{Ln} \rangle^{Ln} \right]$

Erschafft ein Zeilenstück für gesprochenen Text. Der Zeitraum beschreibt, wann das Zeilenstück gesprochen wird und dabei erscheint. Die Sammlung muß nicht zwingend Effekte enthalten. Bei etwas anderem werden dann eben keine Effekte auf den Zusammenhang angewendet. Zum Einsatz kommen ausschließlich Koordinierungseffekte.

★

Ln.muta

laß.verstummen

(Strg+⌞+u2280)
(Esperanto)

$\left[\langle Zeile_{Ln} \rangle^{Ln} \right] \langle Beginn_{\mathbb{Q}}^{\Delta} \rangle \langle Dauer_{\mathbb{Q}}^{\Delta} \rangle \lceil Effekte_{sl}^{Ef} \lceil (★ \mid Ln.mut \mid laß.verstummen): \left[\langle Zusammenhang_{zh} \rangle^{zh} \langle Zeile_{Ln} \rangle^{Ln} \right]$

Erschafft ein Zeilenstück für stummen Text. Statt eines Zeitraums ist hier der Beginn und die Dauer der Erscheinung anzugeben. Die Sammlung muß nicht zwingend Effekte enthalten. Bei etwas anderem werden dann eben keine Effekte auf den Zusammenhang

angewendet. Zum Einsatz kommen ausschließlich Koordinierungseffekte.

Verarbeitung des Skriptes zeile.lysa: ./usr/share/goLysa/Schriften/brandvogel-r-400.lysa.gz •
einbeziehen • brandvogel-r-400 (Schrift) • **registrieren** • 0 • 0 • **Dupel generieren** • ,linksbündig' • 30 • 19 •
 1,33 • **Kolumne erschaffen** • **registrieren** • **Zeile erschaffen** • 0 • 33 • **Sammlung erschaffen** •
verstummen lassen • ,E' • ,r' • ,s' • ,c' • ,h' • ,r' • ,o' • ,c' • ,k' • ,e' • ,n' • **sammeln** • 0,33 • 1 • 1 • **Ausschluß**
generieren • 36 • 51 • **Sammlung erschaffen** • **verstummen lassen** • ,liga.st' • ,a' • ,u' • ,n' • ,t' • **sammeln**
 • 0,33 • 1 • 1 • **Ausschluß generieren** • 54 • 63 • **Sammlung erschaffen** • **verstummen lassen** • ,d' • ,e' • ,r'
 • **sammeln** • 0,33 • 1 • 1 • **Ausschluß generieren** • 66 • 81 • **Sammlung erschaffen** • **verstummen lassen**
 • ,H' • ,e' • ,i' • ,d' • ,e' • **sammeln** • 0,33 • 1 • 1 • **Ausschluß generieren** • 84 • 99 • **Sammlung erschaffen** •
verstummen lassen • ,s' • ,c' • ,h' • ,a' • ,f' • **sammeln** • 0,33 • 1 • 1 • **Ausschluß generieren** • 102 • 114 •
Sammlung erschaffen • **verstummen lassen** • ,m' • ,i' • ,c' • ,h' • **sammeln** • 0,33 • 1 • 1 • **Ausschluß**
generieren • 117 • 126 • **Sammlung erschaffen** • **verstummen lassen** • ,a' • ,n' • ,Komma' • **sammeln** •
Zeile verwenden • **zeigen**

Zeile:

Nummer:	1
Breite:	18,194 von 19□
Abstand:	1,33□
Satz:	linksbündig
Randausgleich (links):	0 Geviert

Erschrocken staunt der Heide Schaf mich an,

Eine Zeile repräsentiert eine Textzeile in einer Kolumne (S. 202). Die Gegebenheitsart kann nicht für sich bestehen, sondern muß zwingend einer Kolumne zugeordnet sein. Insofern ist es erforderlich, daß sich eine solche im Register befindet, wenn eine neue Zeile erschaffen wird. Die Zeile wird dieser dann automatisch zugeordnet.

↪ nein • Q ungültig • Z ungültig • ΔΔ ungültig

¶ (u00b6) 200	Ln.agz 199
← (u21e4) 199	Ln.ara 199
→ (u21e5) 200	Ln.era 200
ändere.auf.Ausgangszeile 199	Tx.aln 200
erschaffe.Zeile 199	Tx.lgn 199
gleiche.Zeilenanfang.aus 199	setze.ab 200
gleiche.Zeilenende.aus 200	Z 199

Ln.ausgangszeile ändere.auf.Ausgangszeile

$[\langle \langle Zeile_{Ln} \rangle^{Ln} \rangle] (Ln.agz \mid \text{ändere.auf.Ausgangszeile}): [\langle Zeile_{Ln} \rangle^{Ln}]$

Verändert bei der registrierten Zeile die Satzart auf linksbündig, wenn sie zuvor auf glatt stand. Sollte dies nicht der Fall sein, bleibt alles wie es ist.

Z Tx.ligne erschaffe.Zeile (Französisch)

$[\langle Kolumne_{Tx} \rangle^{Tx}] (Z \mid Tx.lgn \mid \text{erschaffe.Zeile}): [\langle Zeile_{Ln} \rangle^{Ln} \langle Kolumne_{Tx} \rangle^{Tx}]$

Erstellt eine neue, leere Zeile und stellt sie ins Register. Sie wird automatisch der registrierten Kolumne zugeordnet.

← (Strg+⬆+u21e4) Ln.anfangsrandausgleich gleiche.Zeilenanfang.aus

$[\langle Zeile_{Ln} \rangle^{Ln} \langle Schrift_{sf} \rangle^{sf}] \langle Zeichen_{zk} \rangle (\leftarrow \mid Ln.ara \mid \text{gleiche.Zeilenanfang.aus}): [\langle Zeile_{Ln} \rangle^{Ln} \langle Schrift_{sf} \rangle^{sf}]$

Setzt in der registrierten Zeile den linken optischen Randausgleich anhand der Bezeichnung eines Referenzzeichens der registrierten Schrift.

→

(Strg+⇧+u21e5)

Ln.endrandausgleich
gleiche.Zeilenende.aus

$$[\langle Zeile_{Ln} \rangle^{Ln} \langle Schrift_{sf} \rangle^{sf}] \langle Zeichen_{zk} \rangle (\rightarrow | Ln.era | \text{gleiche.Zeilenende.aus}): [\langle Zeile_{Ln} \rangle^{Ln} \langle Schrift_{sf} \rangle^{sf}]$$

Setzt in der registrierten Zeile den rechten optischen Randausgleich anhand der Bezeichnung eines Referenzzeichens der registrierten Schrift, wenn dort ein glatter Satz angegeben ist.

¶

(Strg+⇧+u00b6)

Tx.alineo
setze.ab

(Esperanto)

$$[\langle Kolumne_{Tx} \rangle^{Tx}] \langle \text{Zusätzlicher Durchschuß}_Q \rangle (\¶ | Tx.aln | \text{setze.ab}): [\langle Zeile_{Ln} \rangle^{Ln} \langle Kolumne_{Tx} \rangle^{Tx}]$$

Erstellt eine neue Zeile und legt diese im Register ab. Anzugeben ist der zusätzliche Durchschuß zwischen Absätzen in Gevierten.

Absatz

Ein neuer Absatz wird im glatten Satz zum einen indirekt durch die darüberliegende, linksbündig gesetzte Ausgangszeile gekennzeichnet, zum anderen durch einen Einzug der ersten Zeile.

Die erste Zeile des ersten Absatzes bleibt allerdings in der Regel uneingerückt (auch der erste Absatz nach einer Überschrift). Wenn eingezogen wird, sollte der Einzug zumindest die Breite eines Geviertes haben.^a

Jan Tschichold empfahl in seinen Regeln für die *Penguin Books* durchgängig einen Einzug von einem Geviert.^b Ansonsten wurde die Größe des Einzugs auch von der Zeilenlänge abhängig gemacht: Pro Konkordanz (48 Punkt) wurde ein Halbpetit (4 Punkt) Einzug empfohlen.^c Demnach macht der Einzug also etwa ein Zwölftel der Zeilenlänge aus (mindestens aber ein Geviert). Ein einmal berech-

netter Einzug gilt für das gesamte Dokument, auch wenn sich darin die Zeilenlänge geändert hat.

Eine besondere Form des Einzugs stellen Absätze dar, bei denen die erste Zeile stumpf (also ohne Einzug) beginnt und dann alle folgenden Zeilen eingezogen sind.

Keinesfalls darf eine Ausgangszeile kürzer sein als ein Einzug.^d

Insofern ein Absatz durch zusätzlichen Durchschuß (S. 203) gekennzeichnet wird, verliert der Einzug seinen Sinn. Folglich kann auf ihn verzichtet werden. Allerdings wird durchgängig davon abgeraten.^e Sachlich ist das dadurch begründet, daß Buchseiten beidseitig bedruckt werden und das Papier oft nicht so undurchsichtig ist, daß die Zeilen der anderen Seite nicht durchscheinen. Verlaufen diese aber aufeinander, wird das Durchscheinen minimiert. In Dokumenten, die lediglich auf dem Monitor betrachtet werden (oder bei Videos mit kinetischer Typographie), spielt das allerdings keine Rolle. Dort entscheidet dann ausschließlich die Ästhetik.

^a[7, S. 67]

^b[6, S. 27] ebenso [11, S. 183] und [1, Nr. 44]

^c[7, S. 67]

^d[7, S. 67], [12, S. 17]

^e»Zu glauben, man könne in solchen Fällen mit Hilfe von Durchschuß den Zeilenabstand vergrößern und damit den neuen Absatz sichtbar machen, ist ganz abwegig. Im guten Buch darf man das überhaupt nicht...« [11, S. 184]

»Die Andeutung der Gliederung durch vermehrten Durchschuß über den Abschnitten anstatt durch Einzüge ist nicht nur rhythmisch mangelhaft und daher störend, sondern auch entweder undeutlich oder überdeutlich.« [19, S. 92 f.]

»Falsch ist es, Absätze durch vermehrten Durchschuß zu trennen, weil dann die Zeilen nicht mehr ›passen‹ (Register halten).« [17, S.54]

Kolumne:	
Position:	0:0 Bildpunkte
Satz:	linksbündig
Geviert:	30 Bildpunkte
Breite:	19□
Zeilenabstand:	1,33□
Erscheinungsbeginn:	0 Λ
Erscheinungsende:	1002 Λ

Erschrocken staunt der Heide Schaf mich an,
als sah's in mir den ersten Menschenmann.
Sein Blick steckt an; wir stehen wie im Schlaf;
mir ist, ich sah zum ersten Mal ein Schaf.

Christian Morgenstern GEBURTSAKT DER PHILOSOPHIE

Eine Kolumne repräsentiert eine Umgebung für einen unumbrochenen Text (Paketsatz), der animiert werden kann.

Für ergänzende Gegebenheiten (Zeilen) ist notwendig, daß das Register mit der Kolumne besetzt ist, zu der sie gehören.

↪ nein • \mathbb{Q} ungültig • \mathbb{Z} ungültig • \mathbb{A} ungültig

erschaffe.Kolumne 202 txt 202

text

erschaffe.Kolumne

$\langle \text{Anfang}_{\text{dp}} \rangle \langle \text{Satz}_{\text{zk}} \rangle \langle \text{Geviert}_{\mathbb{Q}} \rangle \langle \text{Breite}_{\mathbb{Q}} \rangle \langle \text{Zeilenabstand}_{\mathbb{Q}} \rangle \{ \text{txt} \mid$
erschaffe.Kolumne): $\langle \text{Kolumne}_{\text{Tx}} \rangle$

Erstellt eine abstrakte Umgebung zur Zusammenstellung eines Textes in Zeilen. Anzugeben ist die Position des Anfangs der

Grundlinie der ersten Zeile in Bildkoordinaten, sodann die normale Schriftgröße (Geviert) in Bildpunkten, die Zeilenbreite und der Zeilenabstand in Gevierten (1+Durchschuß). In einem Tripelkarree können weitere Transformationen bestimmt werden wie Rotation, Scherung oder Spiegelung des Textes. Soll es keine derartigen Transformationen geben, kann die Angabe auch entfallen. Dann wird automatisch eine Identitätsmatrix (S. 74) eingesetzt.

An Satzarten stehen folgende Angaben zur Verfügung:

glatt Glatter Satz (S. 183)

mittig Zeilen werden auf Mitte gesetzt

linksbündig Zeilen werden nach rechts hin aufgefüllt (wird auch genommen, wenn keine andere Option erkannt wurde)

rechtsbündig Zeilen werden nach links hin aufgefüllt

Durchschuß

Als Durchschuß wird der Zwischenraum zwischen Zeilen bezeichnet. In der Regel empfiehlt es sich, einen größeren Zeilenabstand zu nehmen als die Schriftgröße. Was dabei über die Schriftgröße hinausgeht, ist der Durchschuß.

Kompreß nennt man das Setzen unter Verzicht auf einen Durchschuß. Der Zeilenabstand entspricht dann der Schriftgröße (dem Geviert).

Ausschluß (S. 193) und Durchschuß entsprechen idealerweise einander.^a Der Abstand zwischen Zeilen sollte mindestens so groß sein wie der normale Wortabstand.^b Bei einem Drittersatz ist also ein Zeilenabstand von mindestens $1\frac{1}{3}$, bei einem Viertelsatz $1\frac{1}{4}$ Geviert (S. 180) zu verwenden.

Umgekehrt kann man natürlich auch zuerst den Zeilenabstand wählen und den Ausschluß dann daran bemessen.

^a»Unter keinen Umständen aber sollte der Raum, der zwischen den Zeilen liegt, kleiner wirken als der, der die Wörter einer Zeile voneinander trennt.« [11, S. 146]

^b[12, S. 18]

Ein Effekt stellt eine Möglichkeit dar, ein Bild oder eine Zeichnung in einer Animation über eine Anführung zu beeinflussen. Je nach Einsatz gibt es verschiedene Effekte.

Effekte werden mit eigenen Interpretern erzielt und haben dabei lediglich Zugriff auf das globale Lexikon. Zudem wird die Vereigentlichung eines Effektes nicht protokolliert.

↔ nein • Q ungültig • Z ungültig • ∇ ungültig

efi 204 generiere.Effekt 204

efiko

(Esperanto)

generiere.Effekt

$\langle \text{Gelegenheit}_{zk} \rangle \langle \langle \text{Wirkung}_{Af} \rangle \rangle$ (efi | generiere.Effekt): $\langle \text{Effekt}_{Ef} \rangle$

Folgende Gelegenheiten sind möglich:

Koordinierung Bevor in einer Zeichnung Umrisse erstellt und gefüllt werden, werden alle Effekte dieser Art je in folgendem Umfeld vereigentlicht:

$\left[\langle \text{Zeichnung}_{zn} \rangle \langle \text{Bildnummer}_{Gz} \rangle^a \langle \Lambda_{zl} \rangle^b \langle \text{Fortschritt}_{Zl}^{[0-1]} \rangle^c \right.$
 $\langle \text{Bildausmaße}_{Dp}^{B,H} \rangle^d \langle \text{Parameter}_{Rg} \rangle^e \langle \text{Erstes Bild im Erscheinen}_{Gz} \rangle^f$
 $\langle \text{Erstes Bild im Verharren}_{Gz} \rangle^g \langle \text{Erstes Bild im Verschwinden}_{Gz} \rangle^h$
 $\left. \langle \text{Erstes Bild nach dem Verschwinden}_{Gz} \rangle^i \right] \langle \langle \text{Koordinierung}_{Af} \rangle \rangle \curvearrowright$
 $\langle \text{Zeichnung}_{zn} \rangle$

Erscheinen **Verharren** **Verschwinden** Als Bildbearbeitung der jeweiligen Phase werden zu jedem Bild als Schicht eines Einzelbildes einer Animation alle Effekte der entsprechenden Phase je in folgendem Umfeld vereigentlicht:

$$\left[\langle \text{Bild}_{\text{bd}} \rangle^{\text{Bd}} \quad \langle \text{Bildnummer}_{\text{Gz}} \rangle^{\text{a}} \quad \langle \Lambda_{\text{zl}} \rangle^{\text{b}} \quad \langle \text{Phasenfortschritt}_{\text{zl}}^{[0-1]} \rangle^{\text{c}} \right. \\ \left. \langle \text{Bildaussma\ss e}_{\text{Dp}}^{\text{B,H}} \rangle^{\text{d}} \quad \langle \text{Parameter}_{\text{rg}} \rangle^{\text{e}} \right] \ll \text{Bildbearbeitung}_{\text{Af}} \gg \llcorner : \\ \langle \text{Bild}_{\text{bd}} \rangle$$

Bildbearbeitung Vor dem Speichern eines Bildes als Schicht eines Einzelbildes einer Animation werden alle Effekte dieser Art je in folgendem Umfeld vereigentlich:

$$\left[\langle \text{Bild}_{\text{bd}} \rangle^{\text{Bd}} \quad \langle \text{Bildnummer}_{\text{Gz}} \rangle^{\text{a}} \quad \langle \Lambda_{\text{zl}} \rangle^{\text{b}} \quad \langle \text{Fortschritt}_{\text{zl}}^{[0-1]} \rangle^{\text{c}} \right. \\ \left. \langle \text{Bildaussma\ss e}_{\text{Dp}}^{\text{B,H}} \rangle^{\text{d}} \quad \langle \text{Parameter}_{\text{rg}} \rangle^{\text{e}} \right] \ll \text{Bildbearbeitung}_{\text{Af}} \gg \llcorner : \\ \langle \text{Bild}_{\text{bd}} \rangle$$

Eine Animation produziert eine fortlaufende Bilderserie und kann durch andere Animationen erweitert werden. Solche Erweiterungen dürfen aber zeitlich nicht über die erweiterte Animation hinausragen und müssen dasselbe Bildformat produzieren.

Animationen funktionieren mit verschiedenen Protokollen. Diese werden über eine Zeichenkette bestimmt und erhalten über ein Regal die von ihnen benötigten Gegebenheiten. Die Protokolle sind vorgegeben, werden jedoch an anderer Stelle beschrieben.

↦ nein • \mathbb{Q} ungültig • \mathbb{Z} ungültig • \mathbb{A} ungültig

► (u23f5) 206	clb 207
■ (u23f9) 207	cnm 207
● (u23fa) 207	
☎ (u260e) 207	generiere.Animation 207
anm 206	icp 207
bebildere 206	rufe.zurück 207
binde.ein 207	

► (Strg+⬆+u23f5)
animate (Englisch)
bebildere

$\langle Animation_{Am} \rangle \langle Kennung_{zk} \rangle \langle Ausgabeverzeichnis_{zk} \rangle \langle Format_{zk} \rangle \langle Versatz_{\mathbb{Z}} \rangle$
(► | anm | bebildere)

Generiert die Bilder einer Animation. Ist die aktivierte Animation in eine andere eingebettet, wird stattdessen diese aktiviert.

Die Bilder werden in dem angegebenen Verzeichnis abgelegt.

Die Angabe des Ausgabeformats entspricht abgesehen von der nicht möglichen Ausgabe als Skript jener beim Ausbilden (S. 138). Insofern die Angabe nicht erkannt wird, werden die Bilder als PNG64f gespeichert.



incorporate
binde.ein

(Strg+⬆+u23fa)
(Englisch)

$\langle Elter_{Am} \rangle \langle Kind_{Am} \rangle (\bullet \mid icp \mid binde.ein): \langle Elter_{Am} \rangle$

Fügt eine Animation zu einer anderen hinzu.



cinema
generiere.Animation

(Strg+⬆+u23f9)
(Englisch)

$\langle Protokoll_{zk} \rangle \langle Breite_Z \rangle \langle Höhe_Z \rangle \langle horizontal_Z \rangle \langle vertikal_Z \rangle \langle Dauer_{zr} \rangle$
 $\langle Parameter_{rg} \rangle (\blacksquare \mid cnm \mid generiere.Animation): \langle Animation_{Am} \rangle$

Erstellt eine Animation und legt sie auf den Stapel.



callback
rufe.zurück

(Strg+⬆+u260e)
(Englisch)

$(\langle \dots \rangle) (\text{☎} \mid clb \mid rufe.zurück): ?$

Beim Erzeugen der Animationen werden teilweise Zwischenergebnisse in *Lysa*-Skripte ausgelagert. Um diese weiterzuverarbeiten, werden sie wieder eingelesen und das Protokoll kann über diese Rückruffunktion wieder aktiv werden. Außerhalb dieses speziellen Falls bewirkt das Verb nichts (NOP).

Literaturverzeichnis

- [1] Reinhard Albers **TYPOCARDS**. Regeln und Begriffe zur Mikrotypographie. Ein Kompendium auf 62 Karten. Sulgen, Zürich 2009; 181, 201
- [2] Richard Boulanger (Ed.), Victor Lazzarini (Ed.) **THE AUDIO PROGRAMMING BOOK**. Foreword by Max V. Mathews. Cambridge (Massachusetts), London 2011; 101, 102, 107, 114
- [3] Peter Bühler, Patrick Schlaich, Dominik Sinner **DIGITALE FARBE**. Farbgestaltung – Colormanagement – Farbverarbeitung. Bibliothek der Mediengestaltung. Berlin 2018; 130
- [4] Wilhelm Burger, Mark J. Burge **DIGITAL IMAGE PROCESSING**. An Algorithmic Introduction Using Java. Second Edition. London 2016; 116, 120, 128, 129, 135, 143, 168
- [5] Albert Camus **DER MYTHOS DES SISYPHOS**. Deutsch und mit einem Nachwort von Vincent von Wroblewsky. 8. Auflage. Reinbek bei Hamburg 2006; 7
- [6] Gerd Fleischmann **TSCHICHOLD – NA UND?** Göttingen 2013; 194, 201
- [7] Fritz Genzmer, Walther Großmann **DAS BUCH DES SETZERS**. Kurzgefaßtes Lehr- und Handbuch für den Schriftsetzer. Berlin o. J. [1935?]; 180, 184, 186, 201

- [8] Julius de Goede KALLIGRAPHIE FÜR EINSTEIGER. Schönschreiben lernen, Lehrbuch. Aus dem Niederländischen von Sebastian Holz. München 2003; 177
- [9] Robert W. G. Hunt, Michael R. Pointer MEASURING COLOUR. Fourth Edition. o. O. 2011; 73, 118, 119, 120, 129
- [10] Johannes Itten KUNST DER FARBE. Studienausgabe. 8. Auflage. München 2020; 127
- [11] Josef Käufer DAS SETZERLEHRBUCH. Die Grundlagen des Satzsetzes und seiner Gestaltung. Stuttgart 1956; 40, 181, 184, 186, 194, 201, 203
- [12] Franz W. Kuck, Christian Stang DAS TÜPFELCHEN AUF DEM I. Gebrauchsanweisung für Mikrotypographie. München 2013; 183, 194, 201, 203
- [13] Charles H. Moore PROGRAMMING A PROBLEM ORIENTED LANGUAGE. Forth – how the internals work. Edited by Juergen Pintaske. o. O. 2018; 7
- [14] Yves Nevelsteen KOMPUTEKO. Prikomputila terminokolekto. Dua, plibonigita eldono. o. O. 2012; 17
- [15] Herbert Oertel Jr., Eckart Laurien NUMERISCHE STRÖMUNGSMechanik. 2. Auflage. Braunschweig/Wiesbaden 2003; 86
- [16] Joep Pohlen LETTERFONTÄNE. Deutsche Übersetzung von Jeannette Hark. 4. Auflage, 2. Druck. Köln 2015; 186
- [17] Paul Renner DIE KUNST DER TYPOGRAPHIE. Berlin 1940 (Reprint MaroVerlag, Augsburg 2003); 184, 201
- [18] Julius O. Smith III INTRODUCTION TO DIGITAL FILTERS with Audio Applications. Third printing. Stanford 2012; 106
- [19] Jan Tschichold ERFREULICHE DRUCKSACHEN DURCH GUTE TYPOGRAPHIE. Konstanz 1960 (Reprint MaroVerlag, Augsburg); 181, 186, 194, 201

- [20] Christian Voigt, Jürgen Adamy FORMELSAMMLUNG DER MATRIZENRECHNUNG. München 2007; 74, 76