Referenz für mjoy

vom 2020-08-05

Definition von Bezeichnern

bezeichner1 **==** *wort1 wort2 wort3 ...*
bezeichner2 **==** *wort4 wort5 wort6 ...*

Beispiel eintippen:

makelist (... num -- liste) == [] swap [cons] times <Enter>
'anfang 10 20 30 40 50 5 makelist 'ende stack reverse print <Enter>
... anfang [10 20 30 40 50] ende

Befehle für den Stack

der Parameterstapel (stack) ist eine Liste

stack -- *liste*
Schiebt den Stapel als *liste* auf den Stapel.

liste **unstack**
die *liste* wird zum neuen Stapel.

clear
Löscht den Stapel.

xwert **dup** -- *xwert xwert*
Schiebt eine Extrakopie vom *xwert* auf den Stapel.

wert **pop** --
Entfernt *wert* von der Spitze des Stapels.

xwert ywert **swap** -- *ywert xwert*
Vertauscht *xwert* und *ywert* an der Spitze des Stapels.

x y **over** -- *x y x*

x y z **rotate** -- *z y x*
Vertauscht *x* und *z*.

x y z **rollup** -- *z x y*

x y z **rolldown** -- *y z x*

... *num* **index** -- *stackwert*
Pickt eine Kopie vom *stackwert* mit der Position *num* relativ zur Stapelspitze aus dem Stapel und schiebt ihn auf den Stapel;
mit *num* = 1 -> erster Wert, *num* = 2 -> zweiter Wert, ...

xwert [programm] **dip** -- ... *xwert*
Speichert den *xwert*, führt das *programm* aus, schiebt *xwert* auf den Stapel zurück.

id --
Identitätsfunktion, macht gar nix; als Platzhalter für eine Funktion.

.s --
Gibt den Inhalt des Stapels aus. (jetzt Monadenverhalten)

Befehle für Ein/Ausgabe

wert **.** --
Gibt den obersten *wert* vom Stapel aus. (jetzt Monadenverhalten)

liste **print** --
Gibt die *liste* ohne eckige Klammern aus. (jetzt Monadenverhalten)

fname **loadstring** -- *string*
Lädt den Inhalt einer Textdatei und legt ihn als Charliste auf dem Stapel ab.
(jetzt Monadenverhalten)

fname string **savestring** --
Speichert die Charliste (string) als Text in einer Textdatei ab.
(jetzt Monadenverhalten)

fname **run** --
Startet die Datei *fname*. (jetzt Monadenverhalten)

dump -- (jetzt Monadenverhalten)

Befehle für Listenverarbeitung

[*wert1 wert2 wert3 ...***]**

liste **first** -- *wert*
wert ist der erste Wert der nichtleeren *liste*.

liste1 **rest** -- *liste*
liste ist die Restliste der nichtleeren *liste1* ohne den ersten Wert.

wert1 liste1 **cons** -- *liste*
die *liste* entsteht aus der *liste1* mit neuem ersten *wert1*.

liste1 wert1 **swons** -- *liste*
die *liste* entsteht aus der *liste1* mit neuem ersten *wert1*.

liste1 **uncons** -- *wert liste*
Legt den *first* und den *rest* der nichtleeren *liste1* auf den Stapel.

liste1 **unswons** -- *liste wert*
Legt den *rest* und den *first* der nichtleeren *liste1* auf den Stapel.

liste1 **reverse** -- *liste*
Die Reihenfolge der Elemente der *liste1* wird umgekehrt zur neuen *liste*.

liste **size** -- *num*
num ist die Anzahl der Elemente der *liste*.

liste1 num **take** -- *liste*
Eine *liste* mit den ersten *num* Elementen der *liste1*.

liste1 num **drop** -- *liste*
Eine kopierte *liste* ohne den ersten *num* Elementen der *liste1*.

liste1 liste2 **concat** -- *liste*
Die *liste* ist die Verkettung der *liste1* und *liste2*.

liste1 liste2 **swoncat** -- *liste*

num **iota** -- *liste*
Generiert eine *liste* von Zahlen von 1 bis *num*.

liste num **at** -- $element_{num}$
Pickt das $element_{num}$ aus der Liste.

liste1 num wert **set** -- *liste*

matrix1 **trans** -- *matrix*

wert **unit** -- [*wert*]

wert1 wert2 **pair** -- *[wert1 wert2]*

[wert1 wert2] **unpair** -- *wert1 wert2*

Befehle für das Verarbeiten von Dict-Listen

[*key1 wert1 key2 wert2 ... ...***]**

dict key **dictget** -- *wert*
dict key **get** -- *wert*
Holt den *wert* zum *key* aus dem *dict* hervor.

dict1 key wert **dictput** -- *dict*
dict1 key wert **put** -- *dict*
Legt einen neuen *wert* zum *key* in einem *dict* an mit *dict1* als Kopie.

Mathematische Funktionen

num1 num2 **+** -- *num*
num ist das Ergebnis der Addition von *num1* und *num2*.

num1 num2 **-** -- *num*
num ist das Ergebnis der Subtraktion *num2* von *num1*.

num1 num2 ***** -- *num*
num ist das Produkt von *num1* und *num2*.

num1 num2 **/** -- *num*
num ist der Quotient von *num1* dividiert durch *num2*.

num1 num2 **pow** -- *num*

num1 **pred** -- *num*

num1 **succ** -- *num*

num1 **sign** -- *num*

num1 **abs** -- *num*

num1 **neg** -- *num*
num ist der negative Wert von *num1*.

num1 **int** -- *num*
num ist der ganzzahlige Anteil von *num1*.

num1 **round** -- *num*

num1 **exp** -- *num*

num1 **log** -- *num*

num1 **log10** -- *num*

num1 **log2** -- *num*

pi -- 3.141592653589793

radiwinkel **sin** -- *num*
num ist der Sinus vom *winkel* im Bogenmaß.

radiwinkel **cos** -- *num*
num ist der Cosinus vom *winkel* im Bogenmaß.

radiwinkel **tan** -- *num*

num1 **asin** -- *radiwinkel*

num1 **acos** -- *radiwinkel*

num1 **atan** -- *radiwinkel*

numy numx **atan2** -- *radiwinkel*

num1 **sinh** -- *num*

num1 **cosh** -- *num*

num1 **tanh** -- *num*

num1 **sqrt** -- *num*
num ist die Quadratwurzel von *num1*.

[num1 num2 ... numn] **sum** -- *num*
Summe aller Elemente der Liste.

Logische Funktionen

true -- *true*

Schiebt den Wert true auf den Stapel.

false -- *false*
Schiebt den Wert false auf den Stapel.

bool **not** -- *bool*
Logische Negation für Wahrheitswerte.

bool1 bool2 **and** -- *bool*
Logische Konjunktion für Wahrheitswerte.

bool1 bool2 **or** -- *bool*
Logische Disjunktion für Wahrheitswerte.

*bool1 bool*2 **xor** -- *bool*
Exklusiv-Oder-Verknüpfung für Wahrheitswerte.

wert1 wert2 **=** -- *bool*
charliste1 charliste2 **=** -- *bool*
Prüft, ob *wert1* gleich *wert2* ist, und legt den Wahrheitswert auf den Stapel.

wert1 wert2 **<>** -- *bool*

wert1 wert2 **!=** -- bool

wert1 wert2 **<** -- *bool*
charliste1 charliste2 **<** -- *bool*

wert1 wert2 **>** -- *bool*

wert1 wert2 **<=** -- *bool*

wert1 wert2 **>=** -- *bool*

wert **null** -- *bool*

wert **list** -- *bool*

wert **logical** -- *bool*

wert **consp** -- *bool*

wert **ident** -- *bool*

wert **float** -- *bool*

wert **char** -- *bool*

wert **undef** -- *bool*

wert **user** -- *bool*

wert **type 'null =** -- *bool*
wert **type 'cons =** -- *bool*
wert **type 'ident =** -- *bool*
wert **type 'float =** -- *bool*
wert **type 'char =** -- *bool*
wert **'undef =** -- *bool*

wert liste **in** -- *bool*

Befehle für die Ablaufsteuerung

' *bezeichner* -- *bezeichner*
Der *bezeichner*, der dem Quote folgt, wird auf den Stapel geschoben.

bool [dann] [sonst] **if** -- ...
Wenn *bool* = true -> *dann* wird ausgeführt;
wenn *bool* = false -> *sonst* wird ausgeführt.

bool [dann] [sonst] **branch** -- ... *wie **if***

bool wert1 wert2 **choice** -- *wert*

werti [[wert1 rest1...] [wert2 rest2...] ... [wertn restn...]] **select** -- *[resti...]*

[[bool1 rest1...] [bool2 rest2...] ... [true restn...]] **cond** -- ...

num [programm] **times** -- ...
num-mal wird das *programm* ausgeführt.

[test] [programm] **while** -- ...
Wiederholt, wenn das Ausführen von *test* den Wert true ergibt wird
das *programm* ausgeführt.

[programm] **i** -- ...
Führt das *programm* aus.

[programm] **try** -- ... *[hinweis]*
Führt das *programm* aus und schiebt einen (möglicherweise leeren) *hinweis* auf eine Exception auf den Stapel.

liste [programm] **step** -- ...

liste1 [programm] **map** -- *liste*

liste zero [programm] **fold** -- *querresultat*

liste1 [prädikat] **filter** -- *liste*

num [programm] **'!** -- ...
[monade] [programm] **'!** -- ... (Monadenverhalten)
Erst wird die primitive Monade *num* oder die *[monade]* ausgeführt - also ein Seiteneffekt geschied. Dannach wird das *[programm]* ausgeführt. Die Monade steht am Ende einer Sequenz/eines Programms.

Misc Befehle

wert **type** -- *datentypbezeichner*

null		[ ]
cons		[x y z ...]
ident		abc
integer	*intern	(123)
float		-3.1415e-100
char		"A"
string	*intern	("abc")

num1 num2 num3 **rgb** -- *num*
Berechnet den Rot-Grün-Blau-Wert.

zeichen1 **upper** -- *zeichen*
zeichen ist der Ansiuppercase von *zeichen1*.

zeichen1 **lower** -- *zeichen*
zeichen ist der Ansilowercase von *zeichen1*.

num **chr** -- *zeichen*

zeichen **ord** -- *num*

wert1 wert2 **min** -- *wert*

wert1 wert2 **max** -- *wert*

ident **name** -- *charliste*

ident **body** -- *num*
-- *liste*

wert **addr** -- *num*
Zellenadresse vom Wert.

charliste **parse** -- *liste*
Wandelt die Stringdarstellung in eine *liste* von internen Darstellungen um.

wert **tostr** -- *charliste*
Wandelt den *wert* in eine Stringdarstellung um.

gc --
Erzwingt eine Garbage Collection, die sonst nur spontan auftritt,
wenn die Freelist erschöpft ist.

wert **out** -- *Seiteneffekt
Möglichkeit, um Fehlern auf die Schliche zu kommen.

quit --
Beendet den Interpreter.

identlist -- *liste*
Liste mit den verwendeten Bezeichnern.

wert **error** -- **exception*

Befehle für Turtlegraphic

die Turtle dreht sich im Bogenmaß (Radiant); eine Umdrehung ist 2pi.

2pi -- 6.283185307179586

degreewert **rad** -- *radiantwert*
Grad-Wert wird in Radiant-Wert umgerechnet.

radiantwert **deg** -- *degreewert*
Radiant-Wert wird in Grad-Wert umgerechnet.

offs == id	*oder*	**offs** == rad
für Radiantwerte	*oder*	für Degreewerte
90 rad turn	*oder*	90 turn

init -- [stack [ ] x 0 y 0 angle 0 pen true color 0 size 1 brush 16777215]

Initialisierung der Turtle. Die Turtle ist ein dict (Dict-Liste).

dict **draw** -- (jetzt Monadenverhalten)
Zeichnet die Spur der Turtle.

dict xwert ywert **moveto** -- *dict*
Bewegt die Turtle auf den (x,y)-Wert.
Aufgepasst: die Bildschirmanzeige liegt im 4. Quadranten also (x,-y).

dict relax relay **moverel** -- dict
Relatives moveto.

dict distwert **move** -- *dict*
Die Turtle wird um den *distwert* in die aktuelle Ausrichtung bewegt.

dict winkel **turnto** -- *dict*
Legt die aktuelle Ausrichtung der Turtle fest.

dict relawinkel **turn** -- *dict*
Verändert die aktuelle Ausrichtung der Turtle um den *relawinkel*.

dict **penup** -- *dict*
Hebt den Zeichenstift der Turtle -> es werden keine Linien gemalt.

dict **pendown** -- *dict*
Senkt den Zeichenstift der Turtle -> es kann wieder gemalt werden.

dict num **pencolor** -- *dict*
Setzt die Zeichenstiftfarbe auf *num*.

dict num **pensize** -- *dict*
Legt die Breite des Zeichenstiftes fest.

dict num **brushcolor** -- *dict*
Legt die Farbe für das Ausfüllen von Flächen fest.

dict radius **circle** -- *dict*
Malt einen ausgefüllten Kreis mit dem *radius* über den letzten Punkt.

dict **rectangle** -- *dict*
Malt ein ausgefülltes Rechteck über die letzten zwei Punkte.

colors -- [red 255 black 0 blue 16711680 white 16777215 green 32768 aqua 16776960 darkgray 8421504 fuchsia 16711935 gray 8421504 lime 65280 lightgray 12632256 maroon 128 navy 8388608 olive 32896 purple 8388736 silver

12632256 teal 8421376 yellow 65535 gold 55295 orange 42495]

liste **showgraph** -- *intern
Die Spur, codiert als Paare in der *liste*, wird gezeichnet. (jetzt Monadenverhalten)

start -- *dict*
Beginn mit einer Turtle in der Bildmitte.

dict **;** -- *dict* (jetzt Monadenverhalten)
Kann für Interaktives Zeichnen mit der Turtle genutz werden (zB: 50 move **;**).