

| Befehl (Abk) | Erklärung | Beispiel | |
|-------------------------|--|---------------------------|-------------|
| | | Eingabe | Ausgabe |
| alias :neu :alt | Befehle umbenennen | alias 'neuvw 'vw | |
| alle? :wf1 :wf2 | und - Verknüpfung von Aussagen | dz alle? (2>5)(5<8) | falsch |
| alleerstes :liste | gibt das erste von jedem Listenelement | alleerstes[Ute Susi Alf] | [U S A] |
| arctan :zahl | Arcustangens | arctan 1 | 45 |
| ascii :zeichen | ASCII-Wert des Zeichens | ascii 'a | 97 |
| auf :xyliste | Igel auf Punkt :x :y setzen | auf [100 100] | |
| aufkurs :grad (ak) | Igel ausrichten auf Richtung | aufkurs 90 | |
| aufx :xneu | Igel auf Punkt :xneu :y | aufx 100 | |
| aufxy :xneu :yneu | Igel auf Punkt :xneu :yneu | aufxy 0 0 | |
| aufy :yneu | Igel auf Punkt :yneu | aufy 100 | |
| autostart | Automatisch Prg.Start beim Laden | setze 'autostart [prg1] | |
| bb :breite :höhe | Bitblock ab Igelort | bb 100 50 | |
| benutze :fkt :liste | Anwendung von :fkt auf :liste | benutze 'summe [1 2 3] | 6 |
| bild | Igelfenster löschen | bild | |
| bogen :winkel :radius | Kreisbogen Winkel lang mit Radius | bogen 360 100 | |
| cos :grad | Cosinus | cos 90 | 0 |
| def :pname :liste | Definieren von Prozeduren | def 'quad [[a][rg :a*:a]] | |
| definiert? :name | Testen ob eine Prozedur definiert ist | definiert? 'quad | wahr |
| differenz :zahl :zahl | Differenz berechnen | differenz 100 50 | 50 |
| drucke :wort/liste (dr) | Wort oder Liste auf Monitor ohne FF | dr [1 2 3] | |
| druckezeile :w/l (dz) | Wort oder Liste auf Monitor mit FF | dz [4 5 6] | 1 2 3 4 5 6 |
| dzl | Wort oder Liste auf Monitor mit FF | dzl [7 8 9] | [7 8 9] |
| edalle | Editor öffnen | edalle | |
| edit :pname | Prozedur :pname editieren | edit 'quad | |
| egliste | Eigenschaftsliste ausgeben | | |
| eglisten | Eigenschaftslisten zeigen | | |
| eines? :wf1 :wf2 | oder - Verknüpfung von Aussagen | eines? (2<5)(8<5) | wahr |
| element :n :wl (el) | holt das :n-te Element aus Wort/Liste | element 3 [r e g e n] | g |
| element? :d1 :wl | Prüft ob :d1 in Wort/Liste enthalten | element? 'g 'regen | wahr |
| elementab :d1 :wl | Gibt Wort/Liste ab :d1 aus | elementab 'g 'regen | gen |
| ende | Ende einer Prozedurdefinition | | |
| erstes :wl (er) | erstes Element von Wort/Liste | erstes 'Regen | R |
| exp :zahl | gibt :zahl hoch e zurück | exp 2 | 738.905 |
| fange | Abfangen und Behandeln von Programmfehlern sowie eigene Fehlermeldungen (--> wirf) | | |
| fehler | | | |
| fehleraktion | | | |
| fülle | Füllen einer Figur mit Füllfarbe | fülle | |
| ganzzahl :zahl | Ganzzahl-Anteil von :zahl | ganzzahl 8.7 | 8 |
| gibeg :egl :egn | Eigenschaft von :egn in :egl | | |
| gleich? :d1 :d2 | Vergleicht :d1 und :d2 | gleich? [1 2 3] [3 2 1] | falsch |
| größer? :z1 :z2 | GrößerVergleich v. Zahl1 und Zahl2 | größer? 5 8 | falsch |
| grundwort? :befwort | Prüft ob :befwort ein Logobefehl ist | grundwort? 'grundwort | falsch |
| halt | Unterbricht die Programmausführung | | |
| igeltext :wo/li | Textausgabe im Igelfenster | igeltext 'test | |
| kleiner? :z1 :z2 | Kleiner-Vergleich von :z1 und :z2 | kleiner? 5 8 | wahr |
| kurs | Gibt den Kurs des Igels | ak 0 re 90 kurs | 90 |
| länge :wo/li | Länge eines Wortes/einer Liste | länge? 'regen | 5 |
| lb | Lösche Bild (Igel bleibt) | vw 50 | lb |
| leer? :wo/li | Prüft ob Wort/Liste leer ist | leer [] | wahr |
| letztes :wo/li (lz) | letztes Element von Wort/Liste | letztes [a b c [d e]] | [d e] |

| | | | |
|---------------------------|--|---------------------------|--------------|
| liesliste | Einlesen von Wort, Liste und Zeichen von der Tastatur ... sollte man in einer Variablen speichern! | setze 'l1 liesliste | |
| lieswort | | setze 'w1 lieswort | |
| lieszeichen | | setze 'z1 lieszeichen | |
| links :winkel (li) | dreht den Igel um :winkel nach links | li 90 | |
| liste :w1 :w2 | Macht eine Liste aus den Eingaben | (liste 'a 'b 'c [d e]) | |
| liste? :d1 | Prüft ob :d1 eine Liste ist | liste? [1 2 3] | wahr |
| ln :zahl | Natürlicher Logarithmus (Basis e) | ln 2.718281828459046 | 1 |
| log10 :zahl | Zehnerlogarithmus | log10 10000 | 5 |
| minus :zahl | Negativer Wert von Zahl | minus -3 | 3 |
| miterstem :wo/li :li (me) | Wort/Liste als erstes Element in :li | me 'a [b c d] | [a b c d] |
| mitletztem :wl :l (ml) | Wort/Liste als Letztes in :liste | ml 'a [b c d] | [b c d a] |
| mitte | Igel zur Fenstermitte (0 0) | vw 100 mitte | |
| name? :d1 | Prüft ob :d1 eine Variablenname ist | setze 'alf 1 name? 'alf | wahr |
| namen | Gibt alle Variablennamen aus | namen | [[]alf] |
| nicht :wf | Negation eines Wahrheitswertes | nicht 'wahr | falsch |
| nimmeg :egl :egn :eg | Eigenschaft einem Namen zuweisen | | |
| normalstift | Igelstift auf normal zeichnen stellen | | |
| oder | siehe eines | | |
| oea :liste | alle ohne Erstes | aoe [Das ist toll] | [as st oll] |
| ohneerstes :wo/li (oe) | Wort/Liste ohne erstes Element | oe [Das ist toll] | [ist toll] |
| ohneletztes :wo/li (ol) | Wort/Liste ohne letztes Element | ol [Das ist toll] | [Das ist] |
| ort | Koordinaten des Igels als Liste | mitte ort | [0 0] |
| pause | Prozedur unterbrechen | | |
| potenz :basis :exp | Berechnet die Potenz | potenz 2 10 | 1.024 |
| pr | Beginn einer Prozedurdefinition im Editor: pr quad :a ... | | |
| prliste :pname | Prozedurliste von :pname | prliste 'quad | |
| produkt :z1 :z2 | Produkt zweier Zahlen | produkt 10 2 | 20 |
| prozedur? :pname | Prüft ob :pname eine Prozedur ist | prozedur? 'quad | wahr |
| prozedurliste | Liste aller Prozeduren | prozedurliste | [quad] |
| prtext :pname | :pname in einer einfachen Liste | prtext 'quad | |
| prüfe :wf | Testen einer Aussage (für ww wf) | prüfe (2<3) ww [ok] wf [] | ok |
| quotient :z1 :z2 | Dividiert :z1 durch :z2 | quotient 4 5 | 8 |
| qw :zahl | Quadratwurzel von :zahl | qw 16 | 4 |
| radarctan :zahl | Arcustangens im Bogenmaß | 4*radarctan 1 | 31.415 |
| radcos | Cosinus im Bogenmaß | | |
| radiere | Igelstift auf radieren stellen | | |
| radsin | Sinus im Bogenmaß | | |
| rand | Rand für den Igel | bild rand vw 2000 | |
| randsprung | kein Rand beim Zeichnen | bild randsprung vw 2000 | |
| rechts (re) | Drehung um :w nach rechts | re 90 | |
| rest :divid :divis | Divisionsrest | rest 5 2 | 1 |
| richtung :xyliste (ri) | Peilt den Kurs zum Punkt [x y] an | bild richtung [100 100] | 45 |
| rückgabe (rg) | Rückgabe eines Funktionswertes | | |
| rückkehr (rk) | Abbruch einer Prozedur | | |
| rückwärts :s (rw) | Igel um :s Schritte rückwärts bewegen | rw 100 | |
| runde :zahl | :zahl auf die nächste Genzzahl runden | runde 5.8 | 6 |
| satz :w1 :w2 | Einen Satz aus den Wörtern bilden | (satz 'Alf 'ist 'ok) | [Alf ist ok] |
| setze :vname :wert | Var :vname mit einem Wert belegen | setze 'v1 5 | |
| sichtbar? | Prüft ob der Igel zu sehen ist | vi sichtbar? | falsch |
| sin :zahl | Sinus | sin 0 | 0 |
| startezufall | Zufallszahlgenerierung wiederholen | startezufall 23 | |
| stiftab (sa) | Igelstift aufs Papier zum Zeichnen | sa | |

| | | | |
|------------------------|---|--|----------------|
| stiftbreite :liste | Breite des Zeichenstiftes einstellen | stiftbreite [5 5] vw 100 | |
| stifthoch (sh) | Stift hochheben, zeichnet nicht | sh | |
| stiftunten? | Prüft ob der Stift unten ist | stiftunten? | falsch |
| summe :z1 :z2 | Summe von Zahlen | (summe 4 5 6) | 15 |
| tue :pname | Ausführen einer Prozedur im Programm | tue [(dz 1 2 3)] | 1 2 3 |
| tuewert | wie tue, aber Ausgabe in Liste | | |
| umdef | Systemvariable, die das Löschen und Neudefinieren erlaubt --> alias | setze ``umdef ``wahr vg ``vw vw 100 | |
| umkehrstift | Igelstift wird invertiert | | |
| und :wf1 :wf2 | siehe alle? | | |
| versteckigel (vi) | Igel unsichtbar machen | vi | |
| vg :pname | Vergiß die Prozedur :pname | vg ``quad quad 5 | |
| vga(LIB) | Alles löschen (wie NEU) | | |
| vorher? :w1 :w2 | Prüft ob :w1 vor :w2 kommt (ASCII) | vorher? 3 4 | wahr |
| vorwärts :s (vw) | Igel :s Schritte vorwärts | bild vw 100 | |
| warte :t | Wartet :t mit der Programmausführung | wh 10 [vw 10 warte 100] | |
| weiter | Programmausführung weitermachen | nach pause | |
| wenn :wf :liste | auch mit Alternative möglich: (wenn :wf :liste1 :liste2) | wenn 2<3 [dz ``kleiner] | <i>kleiner</i> |
| wennfalsch :liste (wf) | abhängig von --> prüfe | prüfe (2<3) wf [``falsch] | |
| wennsonst | wenn mit Alternative | wennsonst 2>3 [][] | |
| wennwahr :liste (ww) | abhängig von --> prüfe | prüfe (2<3) ww [``wahr] | wahr |
| wert :vname | Gibt den Wert einer Variablen | setze ``v1 10 wert :v1 | 10 |
| whzahl | Gibt die Schleifenzahl aus | wh 10 [dz whzahl] | |
| wiederhole :n :l (wh) | wiederhole :n-mal [diese Liste] | wh 4 [vw 100 re 90] | |
| wirf | Fehlerbehandlung siehe --> fange | | |
| wort :d1 :d2 | Bildet aus :d1 und :d2 ein Wort | wort ``heu `haufen | heuhaufen |
| wort? :d1 | Prüft ob :d1 ein Wort ist | wort? 24 | falsch |
| zahl? :d1 | Prüft ob :d1 eine Zahl ist | zahl? 24 | wahr |
| zahlformat :z :l :g | Zahlformat der Zahl :z mit Länge :l und Genauigkeit :g | zahlformat 2/3 5 5 | 66.667 |
| zeichen :asciir | Zeichen zum ASCII-Wert | zeichen 64 | @ |
| zeigigel (zi) | Zeige das Igelbild | | |
| zg :pname | Zeigt die Prozedur :pname | zg ``quad | |
| zgp | Zeigt alle Prozeduren | zgp | |
| zoom | Vergrößern /verkleinern der Grafik | zoom 2 | |
| zt | Zeigt die Titelzeilen der Prozeduren | | |
| zz :zahl | Zufallszahl (0<=zz<:zahl) berechnen | zz 5 | |

Weitere Hinweise:

Für weitere Infos und Beispiele zu den einzelnen Befehlen benutzt man das Hilfesystem (Hilfe? | Index deutsch). Aus dem Editor startet man Hilfe zu einem Befehlsword durch <Markieren des Wortes> <rechte Maustaste>

Falls ein Befehl im deutschen Teil nicht vorkommt gibt es ihn möglicherweise als englischen Befehl (Hilfe? |Index englisch)

Die Anzahl der Parameter ist bei allen Logoprozeduren festgelegt.

Ausnahmen sind aber z.B.: summe 2 3 (summe 2 3 4 5)

Kommentarzeilen werden mit ; begonnen

Zeilenumbrüche sind signifikant zur Verhinderung dient das ~ Zeichen

Falls Sie Windows-GDI Teile und die Maus nutzen wollen, dann müssen Sie auf die dafür vorgesehenen englischen Befehle (siehe Menue |Hilfe?|Index englisch| zurückgreifen.

| Erklärung | Befehl | Beispiel |
|---------------------------|--|--|
| Fenster einrichten | WINDOWCREATE parent name title xpos ypos width height | windowcreate "main "mywindow "mytitle 0 0 100 100 |
| Fenster löschen | WINDOWDELETE name | windowdelete "mywindow |
| Meldung ausgeben | MESSAGEBOX banner body | messagebox [Titel] [Dies ist die Meldung] |
| Datei öffnen Dialog | DIALOGFILEOPEN filename | show dialogfileopen "c:\\mswlogo\\beispiel*.lg |
| Datei speichern Dialog | DIALOGFILESAVE filename | show dialogfilesave "c:\\mswlogo\\beispiel*.lg |
| Texteditor öffnen | WINDOWFILEEDIT filename callback | windowfileedit dialogfileopen "c:\\mswlogo\\beispiel*.lg [dz "Fertig] |
| Schalter einrichten | BUTTONCREATE parent name label xpos ypos width height callback | windowcreate "main "mywindow "mytitle ~ 0 0 100 100 buttoncreate "mywindow "mybutton ~ "Drücken 25 25 25 25 [print "gedrückt] |
| Schalter löschen | BUTTONDELETE name | windowdelete "mywindow |
| Dialogbox einrichten | DIALOGCREATE parent name title xpos ypos width height setup | pr meindialog buttoncreate "mydialog "myok "OK 25 25 ~ 50 50 [dialogdelete "mydialog] |
| Dialogbox löschen | DIALOGDELETE name | ende dialogcreate "main "mydialog "mytitle 0 0 ~ 100 100 [mysetup] |
| Textausgabe | STATICCREATE parent name text xpos ypos width height | windowcreate "main "mywindow "mytitle ~ 0 0 100 100 staticcreate "mywindow "mystatic ~ [Heading=0] 25 25 50 25 |
| Textausgabe aktualisieren | STATICUPDATE name text | repeat 360 [rt 1 staticupdate "mystatic ~ se [Heading=] heading] |
| Textausgabe löschen | STATICDELETE name | windowdelete "mywindow |
| außerdem gibt es noch: | | |
| | Listenfelder | |
| | Combobox | |
| | Schieberegler | |
| | Checkbox | |
| | Radiobutton | |
| Mausabfragen: | | |
| aktivieren | MOUSEON leftbuttondown leftbuttonup rightbuttondown rightbuttonup move | sh mouseon [auf mousepos sa] [sh] [] [] [auf mousepos] |
| deaktivieren | MOUSEOFF | < Maus bewegen linke Taste gedrückt ! > mouseoff |
| Position abfragen | MOUSEPOS | |
| Multimedia Befehle (MCI) | | |
| | Logo enthält eine große Anzahl von Befehlen für eine Multimedia-Schnittstelle. siehe hierzu Menue Hilfe MCI | |